Wirtschaftsinformatik – Theorie und Anwendung

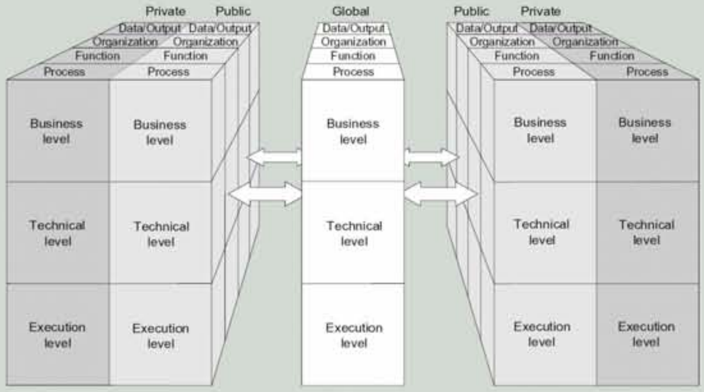
Herausgegeben von Peter Loos

23

Jörg Ziemann

Architecture of Interoperable Information Systems

An Enterprise Model-Based Approach for Describing and Enacting Collaborative Business Processes



λογος

In der Schriftenreihe “Wirtschaftsinformatik – Theorie und Anwendung” werden Beiträge aus Wissenschaft und Praxis publiziert, die sich mit der Modellierung, Konzeption und informationstechnischen Realisierung innovativer Lösungen in Industrie, Dienstleistung und Verwaltung beschäftigen.

The automation of cross-organizational business processes is one of the most important trends of the information age. Instead of a tight integration however, collaborating organizations rather strive for a loose coupling of their information systems. Supporting this objective, the Architecture of Interoperable Information Systems (AIOS) represents a means for the comprehensive description of loosely coupled, interoperating information systems and for the systematic, model-based enactment of collaborative business processes. To this aim, it combines concepts from the areas of enterprise modeling, collaborative business and Service-oriented Computing. At the core of the architecture lies the Business Interoperability Interface, which describes the information system boundaries of one organization to its collaboration partners and connects internal and external information systems. Detailed procedure models specify the usage of the AIOS; its application to an example scenario as well as prototypes that implement core aspects of the AIOS exemplify the method. This book addresses researchers as well as practitioners interested in the areas of organizational interoperability and the modeling and enactment of collaborative business processes.

ISBN 978-3-8325-2414-2

Logos Verlag Berlin

ISSN 1863-4141

Foreword by Peter Loos

Integration and interoperability are information systems research topics that have been investigated since several decades. Recently these research activities gained new momentum when newly developed network-based technologies opened novel possibilities for executing cross-organizational business processes. On the demand side, the prevailing practical importance of information systems interoperability was exemplified only a couple of weeks ago, when a lack of inter-agency interoperability led to a major security incident in the USA. Also in the private sector, the need for concepts that enable flexible automated collaborations between autonomous enterprises is commonly agreed upon.

Existing methods for establishing interoperability between organizations usually approach this objective either from a predominantly technical or from a very abstract, high-level perspective. In this book, Jörg Ziemann describes an intermediate way, which accomplishes a comprehensive yet operational means of developing interoperable information systems. Different from previous approaches, his method is based on concepts that have been used successfully for decades with the systematic development of information systems and the automation of business processes; these were combined with current concepts from collaborative business, supporting the correlation of internal information system elements with those of collaboration partners. As such, he has specified an interface layer between internal and external elements, which allows an organization to control its degree of information sharing and to provide partner-specific interfaces. In this vein, Jörg Ziemann has developed a convincing concept for a so-called Business Interoperability Interface, which was recommended in the European Interoperability Framework (EIF) as a means of interconnecting large organizations.

In essence, this book methodically joins concepts for the comprehensive development of information systems with concepts for the loose coupling of such systems. The result is an Architecture of Interoperable Information Systems (AIOS) that comprehensively describes interoperating information systems and enables the systematic enactment of collaborative business processes. The architecture is based on well-known and in themselves complete concepts, and thus represents a coherent and comprehensible solution. Due to its integrative character, readers from different domains can benefit from the AIOS: From an *enterprise modeling* perspective, it describes how existing frameworks and tool suites for modeling internal information systems can be extended for use in collaborative business. From the perspective of *collaborative business*, the AIOS can be used as a reference for eBusiness suites and eGovernment interoperability frameworks. From the perspective of *Service-oriented Computing*, the described concepts can be used as a reference for service-based process standards and their connection to adjacent enterprise dimensions. Furthermore, the developed Business Interoperability Interface can be used as a reference for comprehensively describing services as needed for the automated discovery of services.

Saarbrücken, March 2010 *Peter Loos*

Foreword by August-Wilhelm Scheer

Organizations have to adapt quickly and efficiently to new business environments to ensure their business success and their long-term survival. Apart from optimizing internal processes, they have to form value chains with complementary partner organizations and optimize their cross-organizational processes. On the one hand, the cross-organizational activities have to be described and optimized on the conceptual level. On the other hand, the cross-organizational business processes need to be enacted via interoperable information systems. In consequence, collaboration partners not only have to ensure complementary business-level concepts, but they also have to agree on complementary technical and execution level concepts.

With systems such as ARIS (Architecture of Integrated Information Systems), it was possible to describe comprehensively the different dimensions of business processes and information systems at the business, technical and execution level. However, approaches dedicated to cross-organizational business processes usually do not follow such a comprehensive approach. They rather support a more narrow selection of information system elements, like for example business documents or technical processes. With the possibilities created by Service-oriented Computing, expectations on flexible cross-organizational business processes have increased and in turn intensified the demand for a more comprehensive representation of cross-organizational business processes. In this context, it is thus a logical step to use established, comprehensive concepts for business process modeling as the basis for cross-organizational business process development.

By consequently combining concepts known from the area of business process modeling with a view concept that supports the peculiarities of collaborative business, Jörg Ziemann has created a coherent system for describing interoperating information systems. The resulting Architecture of Interoperable Information Systems (AIOS) is a compelling solution both for the comprehensive description of interoperating information systems and the systematic enactment of collaborative business processes. Its thorough review of related concepts in different degrees of granularity and the deduction of novel, broadly applicable theories makes this book relevant to all researchers and practitioners interested in organizational interoperability and collaborative business processes. Both groups will benefit from the illustrative and detailed description of architectural elements, the procedure models for the systematic development of interoperable information systems, the application of the concepts to a use case and the description of prototypes that support and implement the architecture.

Saarbrücken, March 2010 *August-Wilhelm Scheer*

Acknowledgements

In November 2004, I started working at the Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI). Some weeks later, I was assigned to a research project whose topic was going to stay with me until today: namely the interoperability of organizations. Over these years, this book developed, and I was happy to work with wonderful people who greatly supported me in this endeavor. I am especially grateful to Peter Loos, who apart from acting as my supervisor, provided the creative working environment in which I could develop this book and enabled me to present my ideas at many national and international conferences. I also want to thank Prof. Günter Schmidt, who agreed to co-supervise the work and provided me with useful criticism of the developed concepts. Furthermore, I am indebted to Prof. August-Wilhelm Scheer, who five years ago employed me at the IWi and opened the door to the academic world; his Architecture of Integrated Information Systems (ARIS) was an important reference for the development of the AIOS.

At the IWi, many co-workers gave me valuable feedback for which I am most grateful. Special thanks go to Florian Kupsch, Torben Hansen, Michael Fellmann, Tobias Dumont, Dima Panfilenko, Peter Fettke and Peter Busch for their great support and advice. In general, I would like to thank all of my IWi-colleagues which I have (not only) worked with during the last five years, for making this a fascinating and joyful time. Furthermore, I want to thank the excellent people with whom I could work on joint publications: Mathias Born, Florian Dreifus, Klaus Fischer, Jörn Freiheit, Ulrike Greiner, Christian Hahn, Frank-Walter Jäkel, Timo Kahl, Thomas Knothe, Katrina Leyking, Sonia Lippe, Peter Loos, Andreas Martin, Thomas Matheis, Jan Mendling, Ansgar Mondorf, Jörg Müller, Oddrun Ohren, Stephan Roser, Rainer Ruggaber, Daniel Schmidt, Dominik Vanderhaeghen, Philipp Walter, Dirk Werth, Maria Wimmer, Andreas Winter and Ingo Zinnikus.

Thanks go to the European Union for co-financing the research projects I worked on (Interop, ATHENA, R4eGov and Commius) as well as to the brilliant partners of these projects. Most of the time I worked on the R4eGov project, which was the inspiration for many ideas described here. Among the R4eGov partners, special thanks go to Europol and Eurojust, whose collaboration I could utilize as the basis for the use case described in the thesis, to the technical project lead (SAP), and to the partners of the Interoperability work package. In this context it should be said that corresponding to the mostly virtual character of the described use case, the models developed to illustrate the concepts introduced in the thesis were not implemented in any real-life organizational setting, and the contents described in this book do not represent the view of the European Union or the view of any R4eGov partner organization. In case of any mistakes, of course, I am the only one responsible for them.

Last but not least I want to thank my family; without their support, I would not have been able to finish this book. My parents Arnhild and Axel Ziemann have encouraged and supported me unconditionally over the years. My wife Julia exchanged California for the Saarland and accompanied me throughout the whole process of developing the thesis; I am looking forward to everything we will experience together.

Saarbrücken, March 2010 *Jörg Ziemann*

Brief Contents

Contents

List of Figures

List of Tables

List of Acronyms

AIOS	Architecture of Interoperable Information Systems
AML	ARIS Markup Language
ARIS	Architecture of Integrated Information Systems
ArKoS	Architektur Kollaborativer Szenarien
ATHENA	Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and Their Application
B2B	Business-to-Business
BII	Business Interoperability Interface
BPEL	Business Process Execution Language for Web Services
BPMN	Business Process Modeling Notation
BPSS	Business Process Specification Schema
CAS	Case Analysis System
CBP	Collaborative Business Process
CIM	Computational Independent Model
CIMOSA	Computer Integrated Manufacturing Open System Architecture
DFDL	Data Format Description Language
EAW	European Arrest Warrant
ebXML	Electronic Business Using XML
EDI	Electronic Data Interchange
ECA	Enterprise Collaboration Architecture
EDOC	Enterprise Distributed Object Computing
EIF	European Interoperability Framework
EJ NM	EJ National Member
ELO	EP Liaison Officer
EPC	Event-Driven Process Chain
EPML	EPC Markup Language
FEAF	Federal Enterprise Architecture Framework
GEA	Government Enterprise Architecture
GERAM	Generalised Enterprise Reference Architecture and Methodology
GPD	Global Process Demonstrator
IDEAS	Interoperability Development for Enterprise Applications and Software
IEEE	Institute of Electrical and Electronics Engineers
ISR	Information Systems Research
IT	Information Technology
MDA	Model-Driven Architecture

MDD	Model-Driven Development
MS	Member State
OASIS	Organization for the Advancement of Structured Information Standards
OM	Organization Model
OMG	Object Management Group
OSI	Open Systems Interconnection
OSML	Organization Structure Markup Language
PAP	Policy Administration Point
oWFN	Open Workflow Net
PIM	Platform Independent Model
PIP	Partner Interface Process
PSM	Platform Specific Model
RBAC	Role Based Access Control
RfQ	Request for Quote
SHAPE	Semantically-Enabled Heterogeneous Service Architecture and Platforms Engineering
SOA	Service-Oriented Architecture
TEAF	Treasury Enterprise Architecture Framework
TOGAF	The Open Group Architecture Framework
UBL	Universal Business Language
UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
UMM	UN/CEFACT Modeling Methodology
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
UN/EDIFACT	United Nations/Electronic Data Interchange for Administration, Commerce and Transport
VPD	View Process Demonstrator
W3C	World Wide Web Consortium
WfMC	Workflow Management Coalition
WS-CDL	Web Service Choreography Definition Language
WSCI	Web Service Choreography Interface
WSDL	Web Service Definition Language
WSMO	Web Service Modeling Ontology
XACML	Extensible Access Control Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XPDL	XML Process Definition Language
XSD	XML Schema Definition Language
XSLT	Extensible Stylesheet Language Transformation

1 Introduction

The necessity for organizations to collaborate with each other is generally accepted, since the goals of individual organizations can be reached faster, more efficiently and with less risk when working jointly with other organizations.[1] Not only enterprises, but also public administrations have to concentrate on their core competencies, and likewise have to rely on the complementary competences of partner organizations.[2] Accordingly, the inter-connection of organizations is regarded as one of the most important trends of the information age.[3]

In this vein, in the previous decades a development *from intra-department to inter-organizational process optimization* could be observed: While around the year 1985 process optimization focused on intra-department processes, around 1995 business process re-engineering was applied to optimize enterprise-wide processes.[4] Around the millennium, inter-organizational processes were implemented, but these mostly treated organizations as black boxes, inhibiting a fine-grained process optimization. To improve cross-organizational process optimization, in the following years the black boxes were transformed into gray boxes, also taking into account components inside the various enterprises for the optimization of the overall process.[5] However, the traditional independence of enterprises and public administrations inhibits a tight integration ("white box approach") of partner organizations: Although they collaborate, the organizations involved remain economically autonomous and occasionally competing entities. Thus, instead of integration, they rather strive for a loose coupling of organizational systems that requires only minor changes of internal systems – but still enables the automation of cross-organizational processes.

The question of how heterogeneous, loosely coupled systems can interact efficiently both on a business and technical level is being investigated by interoperability research.[6] Since the upcoming of information technology (IT), interoperability has been a major concern in information systems research; and, due to the increasing permeation of IT in society, interoperability between the many resulting information systems remains a challenge. Besides a vast amount of scientific publications on the topic,[7] the prevailing importance of interoperability research is indicated by large governmental investments.[8] Concepts to support cross-organizational processes with information technology have existed for several decades; for example, Electronic Data Interchange (EDI) and Value

---

1 Compare SCHMOLL (2001), p. 9.

2 Compare PRAHALAD & HAMEL (2006), GULATI, NOHRIA & ZAHEER (2006) and WERTH (2005).

3 Compare ÖSTERLE (2000).

4 Compare HAMMER & CHAMPY (1993), DAVENPORT (1993) or SCHEER (1990).

5 Compare for example HOFER (2007), pp. 17, pp. 61 and WERTH (2006), pp. 2.

6 Compare for example LEGNER & WENDE (2006), FORD ET AL. (2007) or RAY ET AL. (2008).

7 For an overview of literature on interoperability, refer for example to PERISTERAS AND TARABANIS (2006) or SCHMIDT ET AL. (2007).

8 For example, in the USA, a $1 billion program was recently initiated to resolve interoperability problems among and between public safety entities (compare PUBLIC SAFETEY & HOMELAND SECURITY BUREAU, 2008). Examples of recent projects on interoperability research funded by the European Commission include INTEROP, ATHENA and R4eGov (compare also Footnote 37 on p. 6).

Added Networks (VAN) were used to implement rigid, cross-organizational processes, using protocols like UN/EDIFACT for defining interchange and message structures.[9] With the rise of the internet, XML-based eBusiness protocol suites were created, focusing on the technical realization of processes.[10] To better connect business with IT and to increase process flexibility, the concept of Service-oriented Architecture (SOA) was developed. Though SOA does not explicitly aim at inter-organizational scenarios and is often propagated for intra-organizational usage,[11] SOA concepts support the description and the loose coupling of distributed services and thus constitute a good basis for the implementation of cross-organizational scenarios. Accordingly, in recent years, SOA has given new impetus to research on collaborative business processes.[12]

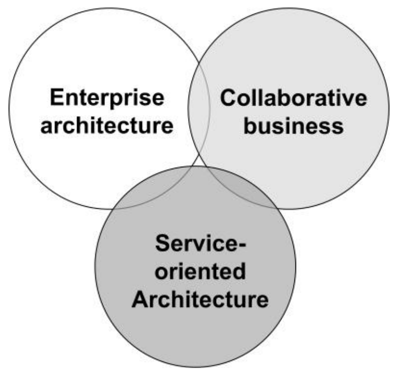


Figure 1: Development of collaborative business processes based on three research fields

1.1 Deficits in the Development of Collaborative Business Processes

Despite these developments, the existing approaches for the development of collaborative business processes have significant shortcomings:[13] They tackle interoperability from various sides, but no integrated, comprehensive approach exists for the systematic development of interoperable enterprise information systems; thus, collaborative business is still inhibited by difficulties stemming

---

9 Compare UNITT & JONES (1999). UN/EDIFACT stands for "United Nations/Electronic Data Interchange For Administration, Commerce and Transport".

10 XML stands for Extensible Markup Language (compare W3C, 2008). An example of such a protocol suite is ebXML (Electronic Business using XML, compare UN/CEFACT, 2003). The technical nature of this eBusiness suite is illustrated by the fact that the design of ebXML processes usually starts with software engineering languages like the Unified Modeling Language (UML) – compare for example HOFREITER, HUEMER & KIM (2006) – and not with genuine languages for capturing business requirements (refer also to the discussion of eBusiness Protcols and ebXML on pp. 82).

11 Compare for example WOODS (2006), CAMPBELL & MOHUN (2007) or HACK & LINDEMANN (2007).

12 For example, concepts stemming from SOA like orchestration, choreography or interface-orientation were picked up in business process modeling. Compare ZIEMANN, KAHL & WERTH (2007).

13 Note that in this introductive chapter, intentionally only a coarse-grained overview of the state of the art and corresponding gaps is provided. However, the gaps declared in this section are re-visited (and confirmed) in Chapter 3, pp. 77, where a detailed literature review is executed.

from the lack of interoperability between organizations.[14] To achieve a comprehensive, business-driven development of collaborative business processes that takes into account the possibilities offered by current technologies, the strength of the three fields illustrated in Figure 1 should be combined:

Enterprise architectures: A comprehensive development of collaborative business processes should cover all relevant enterprise dimensions, incorporating distinct views on data, process, function and organization elements. It should also support systematic process automation and describe processes not only on the technical but also on the business level.[15] Systems that comprehensively describe business processes and support their enactment are known as enterprise architectures.[16] Regarding the question of which enterprise dimensions are relevant for cross-organizational processes, it is important to notice that an inter-organizational business process represents a specialization of a business process,[17] which indicates that the dimensions normally used to describe business processes are also relevant for describing inter-organizational processes.[18]

Collaborative business: To cope with the peculiarities of inter-organizational processes, different concepts were developed: eBusiness protocol suites, interoperability frameworks, approaches for Model-driven Interoperability and enterprise modeling-related approaches to develop collaborative business processes.[19] Most importantly for this work, in research on cross-organizational workflows a concept for collaborative views on processes was developed that can be extended to provide a comprehensive support of interoperability between collaborating organizations.

Service-oriented Architecture: While the previous two fields (partly) incorporate business-level concepts, SOA provides concepts to implement business processes. However, since SOA is based on coarse-grained, business-oriented components (e.g. services), it supports a connection to the conceptual level. Apart from that, SOA is appropriate for implementing cross-organizational processes since it enables the access of services via – possibly cross-organizational – networks, SOA-related standards are well accepted in industry, and SOA supports interface-orientation and loose coupling.

---

[14] Compare for example PANETTO, SCANNAPIECO & ZELM (2004), DIEDRICH ET AL. (2007) or LEGNER & LEBRETON (2007).

[15] Thus, SCHECKERMANN (2004), p. 110, indicates that a lack of business-orientation led to the failure of many efforts for establishing enterprise architectures.

[16] Prominent examples of enterprise architectures are ARIS and the Zachman framework; compare also pp. 33 of this thesis.

[17] This is nicely expressed by the statement “a process is a process is a process” by SCHEER (1996; aligned to the quote that a “rose is a rose is a rose” by STEIN, 1922).

[18] That is also indicated by recent developments in SOA standards, where – going beyond typical elements of cross-organizational processes like documents or security mechanisms – the need for a stronger support of model types formerly used mainly intra-organizationally is acknowledged (e.g. a dedicated representation of organization elements, compare also pp. 147). On the other hand, inter-organizational processes can have requirements going beyond those of intra-organizational processes – for example, a greater need for information hiding, security mechanisms and a greater need to avoid misinterpretations.

[19] Compare for example UN/CEFACT (2003), EEUROPE (2005), ATHENA (2007 IOP) and GREINER ET AL. (2006).

Until now, systems for developing collaborative business processes are inclined too much toward one of these fields, and thus neglect the advantages of adjacent systems. More specifically:[20]

Enterprise architectures lack support for cross-organizational processes and for SOA: Enterprise architectures were created to systematically describe and enact *internal* business processes, thus they lack concepts to support cross-organizational processes.[21] While SOA gave new inspiration towards the enactment of processes, enterprise modeling is only slowly catching up in extending and adapting existing enterprise modeling concepts towards these developments. Thus, enterprise modeling concepts have to be extended to exploit the possibilities offered by SOA.[22] Besides enterprise architectures, other approaches for model-driven development of software systems exist, most prominently the Model-driven Architecture (MDA) from the Object Management Group (OMG).[23] However, compared to enterprise architectures, these take a narrower scope by focusing on technical modeling levels and fewer enterprise dimensions (e.g. functions/classes and processes). And like enterprise architectures, most MDA approaches lack support for cross-organizational system development.

Interoperability frameworks lack constructive aspect: Various interoperability frameworks were created, especially in the area of eGovernment. However, these usually focus on the descriptive and not on the constructive aspects of interoperability,[24] e.g. by providing standards and technical recommendations for selected layers of IT systems. They only offer limited support for the systematic development of cross-organizational processes, since the concepts comprised in them do not support a model-driven development and procedure models for process development are missing.[25] Moreover, a lacking support for business-level interoperability and a lacking interface-orientation of these frameworks can be observed.[26] This raises the necessity for extending existing interoperability frameworks regarding the constructive aspects that support the business-level design and the (SOA-based) enactment of collaborative business processes.

Business protocol suites lack comprehensiveness and SOA support: Business protocol suites focus on the data dimension and technical aspects of process execution, disregarding other enterprise dimensions and conceptual layers needed in a comprehensive process development; they were created before SOA, and thus lack interface-orientation and flexibility. Though techniques like EDI are suitable for large enterprises that establish "hard-wired"

---

20 As mentioned above (Footnote 13), this list represents only an overview of research gaps; the fields and their deficiencies are described in detail in Chapter 3.

21 Pointing out the lacking support for collaborative business in current enterprise architectures, LEGNER & WENDE (2007), p. 3, for example, state that "in order to allow for basic coordination among business partners and clarify interdependencies, the future process architecture needs to reflect external process integration".

22 For example, SOA concepts like the separation of internal and external behavior descriptions have to be picked up by enterprise modeling, enabling increased interoperability and flexibility; compare also LANKHORST (2007), p. 44.

23 Compare for example BAUER & MÜLLER (2004).

24 Descriptive aspects define for example which specific eGovernment data specifications, interfaces, process models or open standards ought to be used on different local, regional and international public administrations.

25 Compare SCHMIDT ET AL. (2007).

26 Thus, recently in the European Interoperability Framework the development of "Business Interoperability Interfaces" was explicitly recommended. Compare EUROPEAN COMMISSION (2004), p. 18.

business collaborations over a long time frame, they are not suitable for flexible collaborations where processes and partners change more frequently.[27]

Frameworks for cross-organizational workflows lack comprehensiveness: These frameworks usually concentrate on executable processes and their formal validation, neglecting the business-level and a comprehensive coverage of enterprise dimensions.[28] Though only few frameworks for cross-organizational workflows explicitly support SOA, they support the interface-orientation typical for SOA by describing external views on a system. However, these frameworks do not provide a comprehensive, business-driven development of cross-organizational processes.

SOA lacks comprehensiveness: Though SOA should serve as a bridge between the business and implementation layer, most work on SOA focuses on technical instead of business issues; related business concepts are created only afterwards in a bottom-up manner.[29] This organic development led to a variety of complex standards,[30] which are often poorly connected or overlapping, hard to understand and to maintain.[31] A result of this development is that SOA standards concentrate on the function (e.g. service) and the process dimension, while other enterprise dimensions are neglected, for example, dimensions to represent business documents and organization elements.[32] Thus, in order to support a business-driven systems development,[33] established knowledge from the enterprise modeling domain needs to be integrated in SOA development methods.[34]

In conclusion, it can be said that the different strengths of these approaches complement each other, but a method is lacking that would integrate their strengths into one coherent system enabling the systematic description and enactment of collaborative business processes.

1.2 Research Background and Objectives

This thesis was written at the Institute for Information Systems at the German Research Center for Artificial Intelligence. At the institute, a research group dedicated to collaborative business and SOA was founded ("Competence Center for Business Integration"); the discussions and publica-

---

27 Compare LIEGL (2008), p. 1.

28 Frameworks for cross-organizational workflows are further described in Chapter 3 (p. 82) and, in more detail, in Chapter 4 (pp. 120).

29 This applies for example to service choreography and orchestration, where, after the technical solutions were created, a lack of methods to conceptually complement them can be observed. Compare DECKER (2008), p. 166.

30 As also described on pp. 82, the decentral, bottom-up development of SOA also had positive aspects, since it enabled a fast development of innovative standards; the fact that SOA standards are (supposedly) complementary and loosely coupled allowed for a best-of-breed approach, where unsuccessful standards could be ignored without hindering the overall development of the web service stack.

31 HOLMES ET AL. (2008) stated in this context that the lack of "abstracting and conceptualizing" while developing SOA standards makes the "evolution of process-driven SOAs a costly and error-prone undertaking".

32 These deficiencies are described in detail in Chapter 4, compare pp. 147 and pp. 164.

33 The need for a business-based SOA development method is further explained in Chapter 3, pp. 86.

34 Accordingly, ZIMMERMANN, KROGDAHL & GEE (2004) pointed out that "the level of abstraction" in SOA development has to be raised again in order to establish a connection to business-level concepts.

tions produced in this environment provided valuable input for this thesis.[35] Moreover, this thesis is closely related to the European Research Community. On the one hand, because the author participated in many (mostly European) conferences,[36] on the other hand, because the author participated in four European research projects aiming at the improvement of interoperability among organizations based on Service-oriented Architectures: INTEROP, ATHENA, R4eGov and SHAPE.[37] INTEROP focused on exchange of interoperability knowledge in the European research community and SHAPE on the development of Service-oriented Architectures. ATHENA and R4eGov on the other hand tackled real life scenarios provided by practice partners in the project consortium. In the case of ATHENA, business use cases were investigated, including eCommerce scenarios from the furniture and the automotive industry. In R4eGov, the collaboration among large European public administrations was investigated.[38] *Both in the eCommerce and the eGovernment scenarios, organizations required methods for a systematic, sustainable development of interoperable information systems.* This method had to keep their internal processes protected and mostly unchanged, but still enable a seamless collaboration with pre-defined partner organizations.[39] In this vein, the overall objective of this thesis is the creation of:

> *A holistic method for the development of interoperable information systems that enables the enactment of collaborative business processes between autonomous organizations.*

A method can be understood as a codified series of steps taken to reach a certain objective.[40] In the context of software development – and especially in the context of developing cross-organizational software systems – these steps need to take into account many different model types.[41] To define these model types and describe the relationships among them, an architecture is required.

---

35 Recently finished PhD theses from the Institute for Information Systems on the topic of collaborative business include WERTH (2006), KUPSCH (2006), HOFER (2007), THELING (2008), and VANDERHAEGHEN (2009).

36 Compare the literature index of this thesis; papers presented on conferences include for example ZIEMANN, KAHL & MATHEIS (2007), ZIEMANN & MENDLING (2005), ZIEMANN, MATHEIS & FREIHEIT (2007) and ZIEMANN, MATHEIS & WERTH (2008).

37 All research projects were supported under the Information Society Technologies program of the Sixth Framework Program of the European Commission. INTEROP started in 2004 and ended in 2007 (project number IST-2004-508011), ATHENA likewise went from 2004 to 2007 (the acronym stands for "Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application", project number IST-2004-507849). R4eGov went from 2006 to 2009 (project number IST-2004-026650) and SHAPE started in December 2007 ("Semantically-enabled Heterogeneous Service Architecture and Platforms Engineering", project number IST-2007-216408). Compare also PANETTO, SCANNAPIECO & ZELM (2004), RUGGABER (2006), www.r4egov.eu and www.shape-project.eu.

38 During the first two years after its start, the author led the work package for "Model-driven Interoperability" in the R4eGov project and in this role could develop and discuss various important concepts that contributed to this thesis.

39 The need for a "holistic B2B methodology", covering various enterprise dimensions, conceptual and technical levels, was also recently confirmed by scientists involved in the development of UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) modeling methodology; compare LIEGL (2008), p. 4.

40 The Greek work "méthodos" originally means to follow a way; compare Duden (2006), entry "methode". Today it is used to describe "a particular way of doing something"; compare CAMBRIDGE (2005), entry "method". It is also defined as a systematic procedure to obtain scientific or practical results; compare DUDEN (2007), entry "method".

41 Accordingly, SCHULZ & ORLOWSKA (2004), p. 113, state that in the context of cross-organizational processes, modeling and architecting are always closely interweaved.

Figure 2 illustrates that the research objectives of the thesis can be seen as four parts of an architecture: *First*, the overall structure of the architecture. *Second*, a Business Interoperability Interface, which describes the means offered by one organization to collaborate with other organizations, and represents an essential part of the overall architecture. *Third*, the fine-grained specification of the individual dimensions of the architecture. *Fourth*, representing the dynamic part of the architecture, a procedure model that describes how the static elements of the architecture are stepwise developed in the enactment of a collaborative business process. Going beyond the architectural elements, a *fifth* research objective is to validate and operationalize the previous concepts with the development of a prototypical tool, which supports the description and enactment of collaborative business processes.

Architecture Describing the Overall System A comprehensive development of collaborative business processes requires many different model types, taking into account different enterprise dimensions, different levels of technical granularity and different levels of information hiding. To support this endeavor, an architecture should be developed that describes the elements needed in the development of collaborative business processes. The corresponding research objective is the creation of *an interoperability architecture describing the model types required in the development of collaborative business processes and their relationships.*

In the following, this architecture will be referred to as *Architecture for Interoperable Information Systems* (AIOS). In order to support interoperability (and not integration), the architecture should enable a loose coupling between organizations. Thus, the internal processes of the collaboration partners should not be connected directly with each other, but via interfaces that act as proxies between internal and external processes.

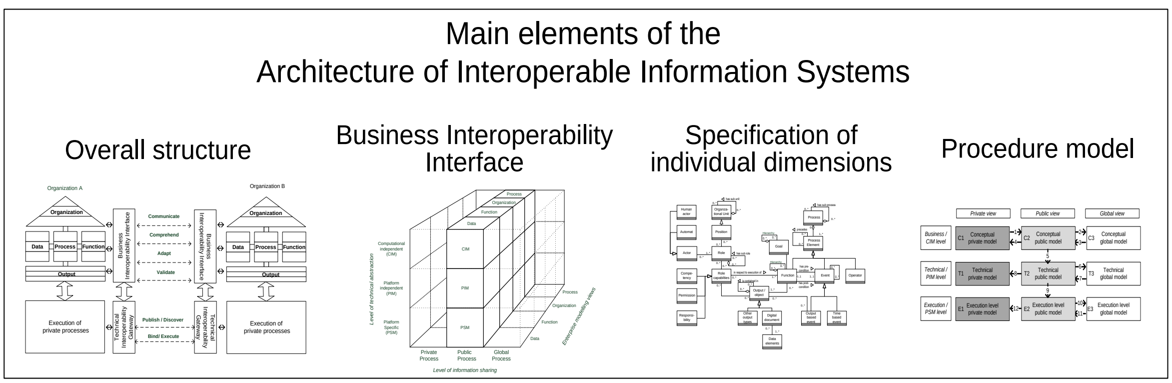


Figure 2: Overview of AIOS elements[42]

Business Interoperability Interface This organizational interface should take a central position in the overall architecture. Other than connecting internal and external processes, it also should provide collaboration partners with a description of the interaction capabilities of an organization and enable a loose coupling between the information systems of collaborating organizations.[43] The

42 The graphics comprised in the figure are also illustrated on pages 117, 115, 177 and 186.

43 Note that interface-orientation is also a key characteristic of SOA, aiming at the loose coupling of services. The close connection between interface description, loose coupling and interoperability is also indicated by NEWCOMER & LO-

need for such an interface was also recently confirmed by the EUROPEAN COMMISSION, which realized that public administrations are independent organizations that will harmonize internal processes only to a limited degree with each other. Therefore, they expressed the need for concepts, which ensure that "processes that are internal to a particular Member State can remain unchanged provided that 'entry and exit points' to these processes are made transparent"[44]. Moreover, they formulated this recommendation: "Public administrations that consider setting up eGovernment services with a pan-European dimension should analyze the related business processes and actors to be involved. They should agree on the necessary Business Interoperability Interfaces (BII) through which their business processes will be able to interoperate at pan-European level and the definition of common BII standards should be studied"[45]. Following this terminology, the interface at the center of the AIOS in the following is also referred to as *Business Interoperability Interface* (abbreviated as BII). To support different stakeholder groups and a model-driven process implementation, it should not only display all relevant enterprise dimensions, but also the different technical levels needed in the automation of collaborative business processes, e.g. conceptual as well as technical levels. This research objective can be summarized as follows: *A comprehensive interface describing the interaction potential of an organization from a business and a technical perspective, serving as a connection point for internal and external processes.*[46]

Fine-Grained Specification of Elements Comprised in the AIOS Dimensions After the overall structure of the AIOS is defined, the individual elements comprised in dimensions of the architecture should be specified as well. This specification should facilitate the description of collaborative business process on the business level, but should also support their enactment. Correspondingly, research goal three is *the development of metamodels that describe the AIOS dimensions in detail and support the specification of collaborative business processes on different levels of technical granularity.*

Procedure Model The overall structure of the AIOS and the related metamodels describe the elements needed in the development of collaborative business processes as well as relationships among them. The procedure model on the other hand, should describe in which sequence these elements are developed, and which implications result from certain development paths. The creation of a procedure model should also demonstrate the applicability of the overall architecture. The corresponding objective is *a procedure model that describes the sequence in which the elements of the AIOS should be developed in order to enact collaborative business processes.*

Development and Application of Prototype To validate and operationalize the AIOS, a prototype should be developed that supports the modeling of collaborative business processes with the AIOS. This prototype should also illustrate the ability of the AIOS to extend mechanisms used in current tools for business process modeling. As an additional proof of concept, both the procedure

---

MOW (2005), p. 113, who state that the interface description of a service represents "the means by which interoperability and integration are achieved".

44 EUROPEAN COMMISSION (2004), p. 18.

45 EUROPEAN COMMISSION (2004), p. 18 (the original is written in British English and no highlights were used).

46 In this work, the interface will be mainly seen as a contract applicable in the design and enactment of collaborative business processes. However, it can also be understood as a means to identify services. Though the multi-perspective description of the interface certainly represents a good basis for discovering business services, the creation of ad-hoc processes via run time discovery and the binding of services are not tackled in this work.

model and the prototype should be applied to a use case. Thus, the fifth research objective is *the conception of a tool suite that demonstrates how the models comprised in the AIOS can be developed and enacted.*

1.3 Research Design

The objective of information systems research (ISR) is the explanation and design of information systems used in business and public administrations.[47] Information systems research is situated between business administration research and computer sciences; or, as SCHEER puts it: "Business information science serves as a facilitator between business applications and information technology"[48]. Comparing ISR with other sciences, FRANK describes the following distinguishing characteristics of ISR:[49]

- Contingent subject: Since the neglect of one element or its slight variation (after the examination) can result in a significantly changed context, the requirements of complex organizational systems are difficult to describe in a significant, sustainable way. In the same vein, HEVNER ET AL. count "unstable requirements and constraints based upon ill-defined environments context" as one of the *wicked problems* that ISR has to cope with.[50] On the other hand, even if the requirements are described in a precise way and could be reused in many different contexts, they do not necessarily lead to an optimal solution.[51]
- Design and analysis of possible worlds: Different from other sciences, ISR does not only aim at studying existing systems or designing single artifacts. Instead, it results in new ways of organizing work, of cooperation and coordination. Since an evaluation of this solution would require its previous implementation, a scientific evaluation is usually not feasible.
- Insufficient comparability of results: Due to the difficulty to comprehensively describe the context of a solution as well as the difficulty of comparing the usually very complex solutions, results from ISR are hard to compare.

To cope with these peculiarities, two complementary approaches are followed in information systems research: design-science and behavioral science.[52] Behavioral science has its roots in natural

---

47 Compare MERTENS ET AL. (2005), p. 3 or FERSTL, SINZ & AMBERG (1996), p.4. The term "information systems research" is translated from the German word "Wirtschaftsinformatik", and is also translated as "business information science" (SCHEER, 1994, p. VII), or "science of information systems" (FERSTL, SINZ & AMBERG, 1996, p.4).

48 SCHEER (1994), p. VII.

49 Compare FRANK (2006), pp. 11.

50 HEVNER ET AL. (2004), p. 81.

51 FRANK (2006), p. 12, expresses this by the following statement: "patterns of successful action, which might be discovered by empirical research, are contingent themselves: There may well exist other, more successful patterns of action that are enabled by other artifacts and contexts". This effect is also illustrated by the citation attributed to HENRY FORD, who pioneered car mass production: if he would have asked the people what they wanted, they would have answered "faster horses".

52 Compare for example SIMON (1996), GOEKEN (2003), HEVNER ET AL. (2004), FRANK (2006) or WILDE & HESS (2007).

sciences;[53] following their model, it aims at theories for explaining observable behavior. It focuses on the analysis and evaluation of systems and therefore requires the collection of representative data sets and the evaluation of hypothesis through empirical studies.[54] Since this rigorous analytical approach only partly matches the peculiarities of information systems development, it is criticized for leading to insufficient results: The difficulty of measuring meaningful system properties, its retrograde perspective and the lack of support for innovative developments qualify it only for limited areas of ISR.[55]

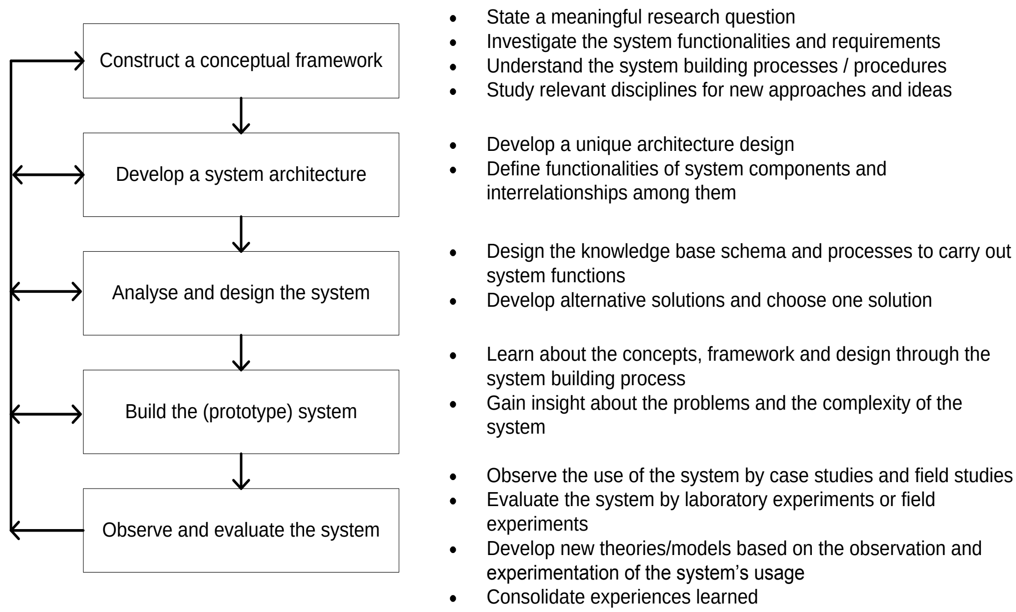


Figure 3: Design-science research method from NUNAMAKER ET AL.[56]

Design-science, on the other hand, has its roots in engineering, seeking to solve problems and to create innovations.[57] While behavioral science focuses on inductive, empirical methods, design-science is a construction-oriented approach, focusing on the engineering of innovative artifacts.[58] Thus, compared to the analytical, reactive behavioral sciences, design-sciences are rather proactive aiming at the development of new solutions.[59] The complementary character of both approaches is also expressed in the statement of HEVNER ET AL. that "*the goal of behavioral science research is truth*" while *"the goal of design-science research is utility*"; which is followed by the declaration

[53] More specifically, it stems from the field of psychology, where it was introduced as a counter reaction to research methods referring exceedingly to mental processes that were not directly verifiable. In consequence, KALAT (1999), p. I-13, defines behaviorism as "a field of psychology that concentrates on observable, measurable behaviors and not on mental processes".

[54] Compare HEVNER ET AL. (2004), p. 77, FRANK (2006), p. 22.

[55] Compare FRANK (2006), pp. 24.

[56] NUNAMAKER, CHEN & PURDIN (1999), p. 98.

[57] Note that the term construction (-oriented) science is used synonymously with the term design science; compare WILDE & HESS (2007), p. 281 or BECKER & PFEIFFER (2005).

[58] Compare SIMON (1996) and HEVNER ET AL. (2004).

[59] Compare WILDE & HESS (2007), p. 281.

that “*truth informs design and utility informs theory*”.[60] While in English-speaking countries in ISR the behavioral science paradigm prevails, in German-speaking countries the design-science approach is used more frequently.[61]

NUNAMAKER, CHEN & PURDIN described a design-science approach for the development of IT systems. Following the design-science paradigm, they state that if the solution of a research problem “proposes a new way of doing things, researchers may elect to develop a system to demonstrate the validity of the solution, based on the suggested new methods, techniques, or design”[62]. In the same vein they state that systems development does not necessarily have to be guided by (empirically gathered) requirements from an organizational setting, but can also follow “new functionalities envisioned by the researcher”[63]. We have chosen their research method as a reference because it resembles the research objectives of this thesis: *The development of an innovative system (for describing and enacting collaborative business processes), guided by envisioned functionalities (e.g., loose coupling of organizational systems and support of process automation), based on existing technologies (e.g. Service-oriented Architecture, enterprise modeling and Business Interoperability).* As illustrated in Figure 3, their method comprises five steps. In this thesis, these steps are realized as follows:

Construct a conceptual framework: Here, the research question is formulated and justified. In addition, the research method is described and relevant research disciplines are described. This first step of their method is tackled in *Chapter 1* of this thesis. In addition, *Chapter 2* describes fundamental concepts for the AIOS and thus represents a part of the conceptual framework.

Develop the system architecture: In this step, the structural relationships and the dynamic interactions among the system components are specified. Here the constraints imposed by the environment, the objectives of the development efforts and the functionalities of the resulting system to achieve the stated objectives are also described. Additionally, requirements should be defined that can be validated in the evaluation stage. This second step is realized in *Chapter 3*, where requirements are described, a state-of-the-art review is carried out and the overall structure of the AIOS is specified.

Analysis and design of the system: Now the previously defined architecture is refined to such a granularity that it can serve as a blueprint for the implementation of the system. Therefore, the tackled domain has to be known in depth; furthermore, different alternatives should be discussed and evaluated before a final design decision is made. Regarding the methods applied in this phase, it is stated “design should be based on theory and abstraction (model-

---

60 Compare HEVNER ET AL. (2004), p. 80. In a similar vein, FETTKE (2006), p. 17, distinguishes between two complementary objectives of ISR: The explanation vs. the development of information systems.

61 Compare also GOEKEN (2003), p. 10 or FRANK (2006), p. 1. Reconfirming their statements, a recent survey showed that in German information systems research “argumentative-deductive” methods are used most frequently (35%), followed by case study (15%) and prototyping based approaches (13%) while methods typical for behavioral research are used to a much smaller extend; compare WILDE & HESS (2007).

62 NUNAMAKER, CHEN & PURDIN (1999), p. 98.

63 NUNAMAKER, CHEN & PURDIN (1999), p. 99.

ing)"[64]. This step is realized in *Chapters 4 and 5* of this thesis, where – after another state-of-the-art review – the elements comprised in the AIOS are specified with metamodels and a procedure model is described that explains how the elements can be used in a model-driven development.

Build the system: Here, a prototype of the system is implemented to demonstrate the feasibility of the design and the usability of the system functionalities. In this vein, in *Chapter 5* a tool suite and a modeling prototype are described, which implement essential parts of the AIOS. Moreover, the procedure model for the AIOS (also developed in *Chapter 5*) can be seen as part of this step as well: though it does not represent a prototype, it shows that the AIOS metamodels can be used to derive proven execution level concepts.

Observe and evaluate the system: In the last step, the system is evaluated by applying it to a use case. Results gained from the implementation of the solution are evaluated against the requirements defined in previous steps. These measures are realized in *Chapter 5*. In this step, NUNAMAKER ET AL. also allocated the development of new theories based on the observations. However, in this thesis, the fifth step mainly confirms the validity of previously developed theories. Nevertheless, new theories going beyond the AIOS are described in *Chapter 6* as future research.

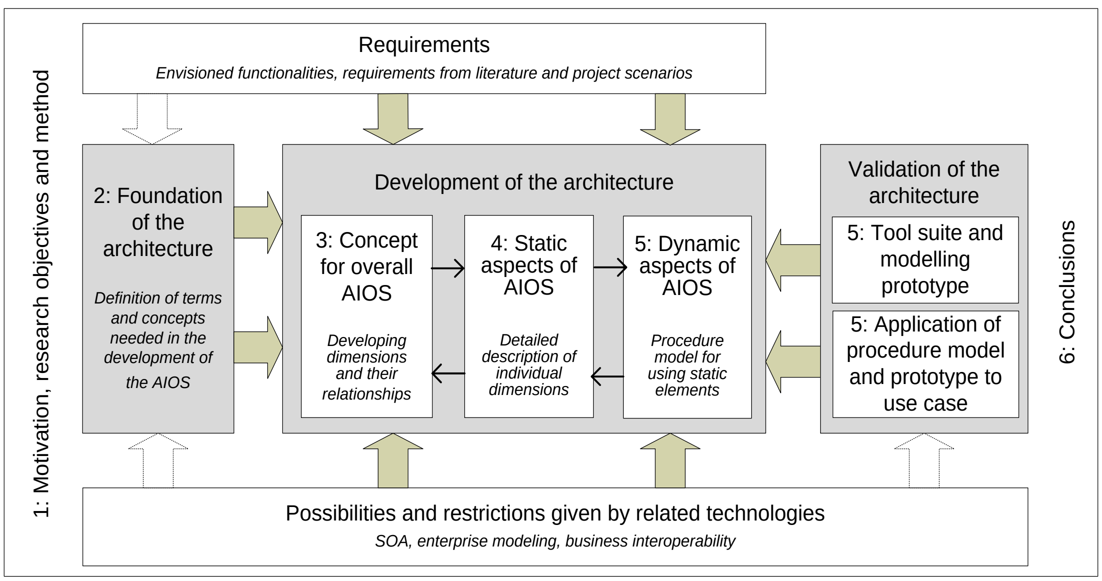


Figure 4: Research method followed in the thesis

The resulting research method is illustrated in Figure 4. In the middle, three major blocks are visible, representing the foundation, development and validation of the AIOS. Inside the development block, the black arrows illustrate the close relationship between the overall structure, the description of the AIOS dimensions and the procedure model. On the one hand, a left-to-right relationship can be observed, where the steps on the left provide a basis for the steps on the right (e.g. the overall

64 NUNAMAKER, CHEN & PURDIN (1999), p. 100.

structure describes the dimensions that are refined in the next step). On the other hand, a right-to-left relationship can be observed, where the steps on the right validate the steps on the left (e.g., the procedure model in Chapter 5 proves the validity of the metamodels developed in Chapter 4, and the refinement of the dimensions validates the feasibility of the overall structure developed in Chapter 3).

Figure 4 also shows that the development process is controlled from two sides: From the top, requirements consisting of envisioned functionalities as well as requirements from literature and related research projects guide the development. From below, possibilities and restrictions provided by concepts from the field of SOA, enterprise modeling and Business Interoperability influence the shape of the AIOS and ensure that the concepts (created based on the envisioned functionalities) are grounded, e.g. their applicability is ensured by their mapping to technical concepts. The solid arrows coming from the requirements block on the top indicate that in the development part requirements are explicitly described, while in the foundational part no explicit requirements definition takes place.[65] The solid arrows from the bottom block indicate that the possibilities and restrictions of existing technologies are discussed predominantly in the development part, where a state-of-the-art overview of concepts related to the AIOS is provided, while in Chapter 2 and in the application of the use case, no explicit state-of-the-art review is carried out. This research method is implemented in the following chapters:

Chapter 1 – Introduction: In this chapter, the motivation of this thesis is laid out, the research method and its position in information systems research is described.

Chapter 2 – Foundational terms and concepts: Here, the conceptual foundation for the thesis is laid, describing and defining essential concepts to be used in the following chapters. Apart from clarifying key terms, in this chapter concepts needed in the automation of collaborative business processes are explained, for example, the concept of developing private, public and global views on processes.

Chapter 3 – Developing the AIOS structure: In this chapter, the overall structure of the AIOS is developed. Based on the perceived gaps in existing solutions, requirements for the architecture are described and consolidated; afterwards a preliminary concept for the architecture is provided. In a second step, a detailed state-of-the-art analysis is carried out investigating to what extent the defined requirements are covered by existing solutions. Finally, the axes of the architecture and its overall structure are consolidated.

Chapter 4 – Specifying individual AIOS dimensions: Here, the concepts described in Chapter 3 are refined. To this purpose, the individual elements of the architecture are described with metamodels, providing private, public and global views on each enterprise dimension of the AIOS. At the end of the chapter, the integration of the elements comprised in the different AIOS dimensions in one metamodel is described.

Chapter 5 – Procedure model and application: The goal of this chapter is twofold: First, to describe the dynamic aspects of the architecture, i.e. a procedure model for the stepwise development of the model types comprised in the AIOS. Second, to provide a proof of concept by

[65] However, the requirements for the AIOS imply the selection of concepts for the foundational chapter.

applying the architecture to a use case and by describing a prototype that implements the concepts described before. In the last section of the chapter, the results of applying the procedure model to the use case are discussed. Apart from the integrated outcome, here also the effort to develop the results is described. Finally, the fulfillment of the objectives formulated in Chapter 3 is assessed.

Chapter 6 – Conclusions: In the first section of this chapter, the quality of the results gained in the previous chapter is reviewed based on the criterions of novelty, validity and usefulness. The second section concludes the thesis with an outlook on future research.

2 Foundational Terms and Concepts

In this chapter, the foundation for the development of the AIOS and its later refinement is laid. Note that the objective of the chapter is *not* to provide a comprehensive state-of-the-art overview or a description of all concepts that are part of the AIOS; instead, it describes selected core concepts needed in the following chapters in order to provide a backbone for the thesis. Thus, Chapter 2 focuses on the process dimension – representing the core of a collaborative business process – and leaves the detailed description of complementary dimensions (organization, data, function etc.) to the following chapters.

Since several concepts needed in the AIOS are defined inconsistently in literature, in the first section of this chapter, terms related to collaborative business processes are defined. In the second section, concepts for the modeling of collaborations are described. Representing a core element of the AIOS, here also the distinction of private, public and global views on collaborative business processes is introduced. In the third section, concepts and techniques are described to transform the business process models to the execution level and to execute them as SOA-based workflows.

2.1 Defining Collaborative Business Processes

2.1.1 Business Process

2.1.1.1 Business Process-Orientation

Following the principles of Taylorism, enterprises in the past century were divided into separate departments that specialized on the execution of individual business *functions.* This separation not only led to the optimization of individual business functions, but also to the development of local optima. The optimization of the overall output of an enterprise on the other hand, was neglected.[66] The principle of optimizing the complete value chain inside an enterprise was picked up by industry only in the 1980's.[67] At this time, essential concepts for business process management were described, including the work from HAMMER & CHAMPY, PORTER, DAVENPORT and SCHEER.[68] PORTER, for example, propagated the concept of interoperability across the value chain as a means to achieve competitive advantage, and highlighted the importance of IT for an efficient interlinkage of value chain activities.[69] Benefits associated with business process-orientation include a higher cus-

---

66 Compare HAMMER AND CHAMPY (1993), p. 52, or BECKER & KAHN (2005), p. 4.

67 Compare BECKER & KAHN (2005), p.5.

68 HAMMER & CHAMPY (1993), PORTER (1985), DAVENPORT (1993), SCHEER (1990).

69 In this vein, he stated "Exploiting linkages usually requires information or information flows that allow optimization or coordination to take place. Thus, information systems are often vital to gaining competitive advantages from linkages". PORTER (1985), p. 50.

tomer satisfaction, a better possibility to control enterprise performance, as well as a higher employee satisfaction.[70]

2.1.1.2 Defining Business Processes

A business process is made up of various activities; aside from activity, also the term *function* is used. In information systems research, a *function represents the transformation of input into output objects*, where this output production is an explicit objective. Thus, SCHEER states, “functions are defined as operations applied to objects for the purpose of supporting one or more goals”[71]. The term *task* implies a stronger focus on the objective, while the term function or activity focuses on the act itself.[72] Similar to a process, a function fulfills a predefined goal. Thus, on a lower level of detail, a business process can also be seen as a business function; however, in difference to the description of a process, the description of a function does not describe the sequence, in which subfunctions that might be comprised in the function are executed.

Corresponding to the viewpoints of the many authors writing about business processes, many different definitions of the term business process exist.[73] Probably one of the most cited ones stems from DAVENPORT, who defines a business process as:

“simply a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus’s emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action.”[74]

HAMMER & CHAMPY further specify that the output of a business process represents a *value for a customer*.[75] SCHEER agrees with the previous definitions, stating, “generally speaking, a business process is a continuous series of enterprise tasks, undertaken for the purpose of creating output. The starting point and final product of the business process is the output requested and utilized by corporate or external customers”[76]. Concordant with these definitions, in the thesis the term business process will be used as follows:

Definition 1 *A* Business Process *is a sequence of business functions aiming at the creation of an output for internal or external customers.*

Since the term business processes is used in many different contexts, many different systems for classifying business processes exist. In the context of process automation, the following criteria are useful to describe and classify business processes:

---

70 Compare SCHEER (2001), p. 9 and VAN DER AALST & VAN HEE (2002), p. 2.
71 SCHEER (2000), p. 22.
72 Compare FINK, SCHNEIDEREIT & VOß (2005), p. 117.
73 Compare for example GADATSCH (2008), SMITH & FINGAR (2002), HAMMER & CHAMPY (1993), DAVENPORT (1993) or SCHEER (1999).
74 DAVENPORT (1993), p. 5.
75 HAMMER & CHAMPY (1993), p. 52, define a business process as “a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer”.
76 SCHEER (1999), p. 4.

- Business value: This criterion displays the importance a certain process type has for an organization.
- Duration: This criterion can by applied both to process instances, e.g. long running vs. short running processes, and to the duration a process type is executed (e.g. only in one month or during many years).
- Rate of repetition: Usually only processes are automated that are instantiated frequently.
- Degree of automation: This dimension expresses the degree of human participation in a process; LEYMANN & ROLLER for example, distinguish three levels:[77] 1. Fully automated, e.g. a batch program which runs in the background of a database, 2. Run & gun process, which are (directly) initiated by humans and run independently afterwards and 3. Human-oriented processes, which have just a very small part of automated activities.[78]
- Degree of formalization: Often processes are not described at all, but are only restricted by implicit knowledge of actors. However, they can also be described informally (e.g. in prose), semi-formally, for example, in diagrams that leave room for interpretation, or formally, in a machine-interpretable way.
- Security: The sensitivity of information transmitted in processes can vary. For example, the data transferred between police departments is usually subject to higher security standards than a hotel booking process.
- Flexibility/Late binding: While some processes are always executed in the same manner and can be completely specified at design time, other processes change often and can only be specified completely at run time (the latter are also called ad hoc processes).
- Scope: This criterion tackles the expansion of a process, which could, for example, be intra-department, inter-department or intra-organizational.

Due to the many different environments in which processes are used, a list of criterions for classifying processes can hardly be complete.[79] However, using the criteria describe above, different process types can now be defined.[80] Since the AIOS focuses on the automation of collaborative business processes, the criteria of formalization/degree of automation and the scope of a process are of special importance. The term workflow for example, can be defined based on the degree of formalization and automation of a process:

Workflows are usually understood as a certain type of business process; though many different definitions of the term workflow exist, most authors agree that a workflow represents the automa-

[77] LEYMANN & ROLLER (2000), pp. 10.

[78] In the context of SOA, MATHAS (2008), pp. 207, is drawing a similar distinction between "human-based workflows" and "technical processes".

[79] For further criteria to characterize business processes, compare for example SCHMIDT (2002), pp. 11, HOFER (2007), p. 19 or WERTH (2006), pp. 21.

[80] Again, many different process types are defined in literature. Different from the conceptions in this work, LEYMANN & ROLLER (2000), pp. 10, for example, distinguish four types of processes: *Collaborative workflows* (with low repetition and high business value), *Ad hoc workflows,* which are rarely repeated and have a small value for the enterprise, *Administrative workflows* that also have a small value for the enterprise and are only indirectly engaged at the value creation, and *Production workflows*, which have a high value for the enterprise and are often repeated.

tion of a business process.[81] Thus, the Workflow Management Coalition (WfMC) defines a workflow as “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules”[82]. Interestingly, it can be observed that in SOA-related publications the word workflow is avoided.[83] However, in the following it is used as defined by the WfMC, inferring that a workflow is a special (e.g. automated) type of a process. Workflows are executed by a workflow engine or a Workflow-Management-System, where the latter is defined as: *“A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications*”[84].

2.1.2 Collaborative Business Processes

2.1.2.1 Collaborative Business Process-Orientation

As sketched out in the motivation of this thesis, the concepts for optimization of intra-organizational processes can be transferred to the optimization inter-organizational processes.[85] Accordingly, the aim of a cross-organizational business process management is not only the coordination and optimization of intra-enterprise processes, but to optimize the sequence of functions stemming from different, collaborating enterprises in a fine-grained manner.[86]

A coarse-grained approach would take into account only the overall role of an enterprise and the core competencies of the enterprises participating in a collaboration. Transferring the concept of process re-engineering to cross-organizational scenarios, collaborative business management aims at a cross-organizational process optimization that does not treat organizations as black boxes, but regards components inside the enterprises for the optimization of the overall process.[87] Thus, different from such a coarse-grained, cross-organizational value chain construction, cross-organizational business process-orientation should result in a fine-grained optimization, which also takes into account the different functions contained *inside* the collaborating organizations.

Figure 5 illustrates the principle underlying both intra-organizational and cross-organizational process optimization: A process optimization based on coarse-grained components, as displayed in

[81] Compare for example SCHEER (1999), p. 87 or GADATSCH (2008), p. 59. An exception is provided by VAN DER AALST, who, instead of focusing on the automated character, named the following three defining characteristics of a workflow: “First of all, a workflow process is case-driven. Secondly, the process itself is considered to be essential. Thirdly, the process can be defined in an explicit manner” (VAN DER AALST, 1998, p. 30).

[82] WfMC (1999), p. 8.

[83] Thus, the probably most prominent SOA standard for executing processes (OASIS, 2007) does not contain the term workflow – despite the participation of WfMC members in its specification. In the same vein, partners in current research projects were skeptical about the “workflow” concept, since they associated negative attributes like rigidity with it, inferring the inability to support complex, real-life processes.

[84] WfMC (1999), p. 9. For a description of the usage of Workflow-Management-Systems in the context of business process management refer for example to LOOS (1998).

[85] Interestingly, PORTER stated already in 1985 that the interlinkages inside the intra-enterprise value chain are of similar nature and importance as the interlinkages in cross-enterprise value chains; compare PORTER (1985), p. 50.

[86] Compare for example WERTH (2006), pp. 2 or HOFER (2007), pp. 61.

[87] Compare for example HOFER (2007), pp. 61.

the middle of the graphic, corresponds to the interactions between organization departments or collaborating enterprises, where both are treated as *black boxes*. A process optimization based on fine-grained components, as displayed on the right-hand side of the graphic, is possible when departments or organizations are treated as *gray boxes* and the components inside of them are at least partly visible. In the coarse-grained optimization, the process consisting of activities of enterprises A, B, C and D is improved by the parallelization of the activities in B and C. In the fine-grained optimization, a higher degree of parallelization and in consequence a faster process execution is possible; for example, the functions B_1, C_1 and C_2 can start before A_2 ends.

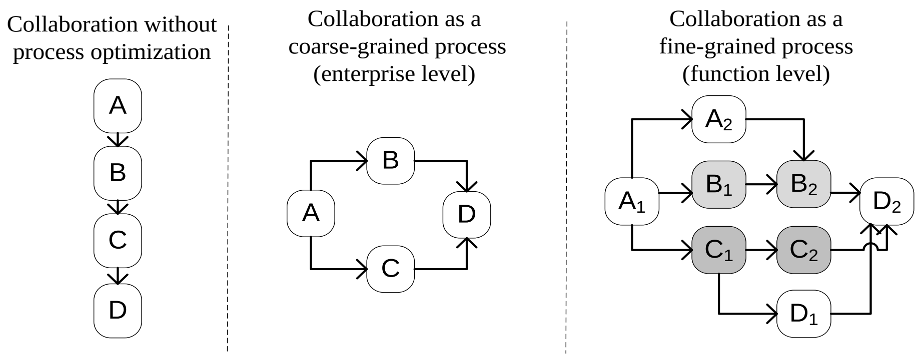


Figure 5: Collaborations based on different levels of process granularity

In information technology, process-orientation can be seen as one approach alongside competing approaches like data-, function- or object-orientation.[88] The 70's and 80's of the last century were dominated by the data-oriented approach. At this time, IT concentrated on storing and retrieving data. Thus, data modeling was the first step in the development of an application system, while process modeling was neglected. In consequence, business processes had to be modified to fit with data-oriented application systems. Moving away from data-orientation, since then, software development was subject to the following changes:[89]

- From programming to assembly: Different from the early days of IT where no generic or domain specific software was available, today many libraries for software functions exist. The challenge is no longer the coding of individual modules, but rather their orchestration.
- From data to processes: Motivated by the need for process-oriented approaches from the business area, system engineers are moving from data-oriented to process-oriented approaches.
- From design to redesign and organic growth: Away from carefully planned designs, software development shifts towards organic growth and the dynamic composition of software assembled from the internet.

The confirmation of these developments by recent IT concepts is obvious, since they resemble the principles of SOA: Coarse-grained services describe and implement business functions, which

88 Compare JABLONSKI, BÖHM & SCHULZE (1997), p. 17.
89 Compare VAN DER AALST, TER HOFSTEDE & WESKE (2003), pp. 2.

are supposed to be composed automatically into workflows – or in SOA terms, into orchestrations and choreographies. SOA and *Software as a Service* (SaaS) explicitly support inter-organizational processes, since services from partner companies can be integrated over networks.[90] Thus, similar to the development of business concepts, also for IT concepts a development towards (cross-organizational) process-orientation can be observed.

On the Similarities between Intra- and Inter-Organizational Business Processes From a business perspective, differences between intra- and inter-organizational business processes seem easy to identify. In inter-organizational scenarios, the degree of shared implicit and explicit knowledge is lower as well as the trust level; heterogeneous enterprise cultures, spatial distribution, possible competition, less possibilities to control actors, divergent objectives and a limited time of working together make the inter-organizational business processes generally more brittle. However, in modern enterprises, similar situations can be found in intra-organizational scenarios: profit centers that compete with each other, spatial distribution among departments of global enterprises, and short project durations can lead to comparable process characteristics.

From an IT perspective, differences between intra- and inter-organizational processes certainly are decreasing. Though collaborating enterprises have separate internal IT systems (for example databases and workflow engines), today standardized network protocols and concepts like SOA minimize differences between intra- and inter-organizational workflows. Workflows in SOA, for example, rely on the invocation of web services, where the invoked web services can be physically allocated *inside* as well as *outside* of an organization. Additionally, SOA focuses on the description of the externally visible interfaces of services. Thus, also inside of organizations that use SOA, IT components have to be described in a similar level of detail as required in cross-organizational scenarios.

In summary, it can be said that in cross-organizational scenarios the need to *explicitly* describe and restrict the parameters of working together is higher than in intra-organizational scenarios. This applies to the business level, where, for example, legal contracts are specified, and to the IT level, where interfaces have to be defined that describe unambiguously and comprehensively the concepts needed in a collaboration.

2.1.2.2 Defining Collaborative Business Processes

Due to the importance of cross-organizational processes a vast amount literature on this subject exists, and the upcoming of the internet, eCommerce and SOA again fostered academic and industrial work in this area. Thus, the IT-supported realization of cross-organizational business processes is discussed under various designations, for example, *Collaborative Business, Business Collaboration, E-Collaboration, Collaborative Commerce, B2B-Integration,*[91] *Business Networking and Business Interoperability*.[92]

---

[90] SOA is explained below (pp. 58); for a description of Software as a Service refer for example to NITU (2009).

[91] B2B stands for Business-to-Business.

[92] Compare SILBERBERGER (2003), SCHUBERT (2008), SPATH, RENNER & WEISBECKER (2005), HARTONO & HOLSAPPLE (2004), LINTHICUM (2001), ÖSTERLE, FLEISCH & RAINER (2001) and GREINER ET AL. (2007).

Cross-Organizational Business Processes Correspondingly, a great number of cross-organizational process types are defined, and even if the designations of the process types are the same, the definitions often are not consistent with each other.[93] However, for describing and classifying cross-organizational business processes often the following criteria are named:[94]

- Number and topology of partners: Obviously, the nature of a collaboration comprising hundreds of organizations differs from a collaboration, where only two or three partners work together. Also depending on the number of partners, the topology of the collaboration can change. For example, a collaboration can be hub-and-spoke shaped, having one central partner, or have the shape of an intermeshed network, where all collaboration partners interact directly with each other.
- Proximity of partners, intensity of coupling: If organizations trust each other and know each other very well, the collaboration will be more open than in a setting, where the partners do not know each other. Thus, dependent on factors like the duration of the collaboration, the trust level among the partners, the heterogeneity of organizational cultures and the degree of spatial distribution, different measures have to be taken to describe and implement collaborative business processes.
- Distribution of coordination: Closely related to the previous point, the coordination of cross-organizational processes can be executed by just one central partner, or by more collaboration partners.
- Distribution of power: If one partner clearly dominates others, the parameters of the collaboration differ from scenarios, where partners have similar strengths. For example, a large car manufacturer can force its suppliers to implement a certain IT infrastructure, while in the case of two large public administrations a joint solution has to be found that takes into account existing IT systems in both organizations.

In addition to these criteria, THELING mentions the *cooperation direction* and distinguishes between horizontal and vertical cooperation.[95] The combination of the criteria listed above allows the definition of many of collaboration types. Often cited forms comprise strategic alliances, consortia, joint ventures, franchising, and virtual enterprises for example;[96] however, since the boundaries between these different types are fluent, they are frequently barely distinguishable.[97] Thus, in the following, we will abstract from such types and focus on the broader term of collaborative business process.

Collaboration Not surprisingly, also for the term collaboration many different interpretations exist. HARTONO & HOLSAPPLE compared 14 definitions of collaboration and subsequently defined

---

93 Forms and definitions of cross-organizational business processes have been discussed intensively in various recently published doctoral theses, e.g. WERTH (2006), HOFER (2007), THELING (2008) and VANDERHAEGHEN (2009).

94 Compare for example HOFER (2007), p. 58, and THELING (2008), pp. 20.

95 Compare THELING (2008), pp. 20. Vertical cooperation refers to relationships along the supply chain (e.g. supplier, manufacturer and retailer), while horizontal cooperation refers to relationships between partners located in the same industrial sector and located on the same position in the supply chain.

96 Compare WERTH (2006), p. 25, and HOFER (2007), p. 60. MASAK (2007), pp. 38, additionally distinguished between four forms: coalition, confederation, federation and virtual enterprise.

97 Compare HOFER (2007), p. 59.

it as "*an interactive, constructive, and knowledge-based process, involving multiple autonomous and voluntary participants employing complementary skills and assets, with a collective objective of achieving an outcome beyond what the participants' capacity and willingness would allow them individually accomplish*"[98].

Closely related to the term collaboration is the term cooperation, which is also used synonymously to collaboration.[99] More specifically, the following conceptions and corresponding relationships of both terms can be distinguished:

- Cooperation and collaboration are synonymous: Some authors treat both terms as synonyms.[100] Especially in colloquial speech, cooperation and collaboration seem to be synonymously; thus, CAMBRIDGE defines cooperation as "to act or work together for a particular purpose, or to help someone willingly when help is requested"[101]. The same dictionary describes collaboration as "to work with someone else for a special purpose", which is synonymous to its definition of cooperation cited above.[102] But also in science both concepts are used synonymously. Thus, HOFER and THELING wrote about *cooperation* while WERTH and VANDERHAEGHEN wrote about *collaboration*; however, in the course of their work, all four tackle the same process type, representing an IT supported "working together" of loosely coupled, autonomous organizations.[103]
- Collaboration is a special case of cooperation: In the context of business information systems, cooperation normally is understood as the joint work of different organizations. Some authors see collaboration as a specialization of cooperation, e.g. HOFER states, that collaboration can be understood as a cooperation that is realized by electronic means.[104]
- Cooperation and collaboration are different concepts: WERTH states that – distinct from cooperation – collaboration is based on autonomous organizations with equal rights.[105] THELING & LOOS distinguish between the two terms as follows: in a cooperation, tasks are distributed among the partners, executed decentrally and the output resulting from the different tasks is integrated only after the tasks have been completed. In a collaboration, each task (as well as the corresponding output object) is tackled jointly by the various partners, thus at the end of the process no integration of the various output ele-

98 HARTONO & HOLSAPPLE (2004), p. 20.
99 Compare for example GERTH (1971), pp. 9, STUKE (1974), pp. 14, and more recently ZENTES, SWOBODA & MORSCHETT (2005), pp. 22, HOFER (2006), pp. 46, and THELING (2008), pp. 13.
100 Compare WERTH (2006), p. 24.
101 CAMBRIDGE ADVANCED LEARNER'S DICTIONARY, entry "cooperation".
102 CAMBRIDGE ADVANCED LEARNER'S DICTIONARY, entry "collaboration". In addition to the cited meaning, the dictionary provides a second meaning: "to work with an enemy who has taken control of your own country". Due to this martial connotation, the use of the term collaboration is discussed critically in information systems research, e.g. in THELING (2008), p. 14. In the following, the term collaboration will be used referring only to the non-belligerent meaning described in the text.
103 Compare HOFER (2007), THELING (2007), WERTH (2006), VANDERHAEGHEN (2009). The synonymous use of both terms is also illustrated in the definition of cooperation from HOFER (2007, p. 51) and the definition of collaboration from WERTH (2006), pp. 39, which basically both express that autonomous organizations work together to reach a shared goal.
104 Compare HOFER (2007), p. 65.
105 Compare WERTH (2006), p. 24.

ments is necessary.[106] We agree with this definition in so far that cooperation in general seems to represent a looser form of working together than collaboration does: if *someone is cooperating* it means at least that he is helping, but normally it does not mean that he is an equal partner executing tasks separately and self responsibly. If *someone is collaborating* this normally means, that he is actively involved in a joint work, having a more important part in this than someone who is only cooperating. Correspondingly, collaboration is usually understood as a long-term endeavor.[107]

Therefore, in the following both collaboration and cooperation are interpreted as "to act or work together for a particular purpose"; however, compared to cooperation, collaboration is understood as a closer form of working together, where the involved partners are equally committed to the joint objective and can work separately on it.

Collaborative Business Process Based on this understanding of collaboration, in the following the term collaborative business process (CBP) is attributed with the following characteristics:

- Cross-organizational: This characteristic is not self-evident, since the term collaboration is not only used to describe interaction between different organizations, but also to describe weakly structured interaction processes between human beings. Thus, NASTANKSY & HILPERT[108] distinguished between three classes of workflows: Structured, semi-structured and collaborative, where the latter class comprises "ad-hoc workflows" and "team-based workflows". Similarly, LEYMANN & ROLLER[109] define collaborative workflows as difficult to describe formally, as typically found in joint creative processes like brand management. However, in recent publications the term collaboration often focuses on the cross-organizational aspect, not judging whether the process is weakly-structured or not. Correspondingly, GUO & LIN state, "Business collaboration is about coordinating the flow of information among organizations and linking their business processes".[110] A comparable understanding of collaborations is provided by WERTH, who defines a collaborative business process as "the assemblage of mutually coordinated activities of coequal, autonomous organizations for the purpose of a distributed production of an output for external customers"[111].[112]
- Autonomous, loosely coupled organizations: The involved parties act as autonomous organizations with similar rights.[113] They do not share internal knowledge with each

[106] Compare THELING & LOOS (2004), p. 2.

[107] Compare for example WERTH (2007), p. 3.

[108] NASTANKSY & HILPERT (1994), pp. 473-479.

[109] LEYMANN & ROLLER (2000), p. 10, compare also the description above of their four workflow classes.

[110] GUO & LIN (2006), p. 463. Compare also WERTH (2006), p. 30.

[111] WERTH (2006), p. 40. The source is in German, originally stating "Der kollaborative Gechäftsprozess ist die Menge von gegenseitig abgestimmten Verrichtungen gleichgestellter, autonomer Organisationen zum Zweck der (arbeitsteiligen) Leistungserstellung für externe Kunden".

[112] For a list of further authors who interpreted collaborations as cross-organizational business process, refer also to SCHUBERT (2008).

[113] Thus, it should be assumed that no partner can dictate conditions to another partner. This excludes, for example, a parent company forcing a child company to open its internal data to the parent company; or the case of a car manufacturer, who can force its supplier to harmonize unilaterally its processes and IT infrastructure with the systems of the manufacturer.

other, except from that knowledge necessary to realize the collaboration. Nonetheless, the contractual coupling between the partners is high enough to justify a substantial investment in the infrastructure necessary to implement the collaboration. This implies a certain value and/or a certain repetition rate of collaborative business processes as well as a longer enduring cooperation.

Based on the definition of business process given above,[114] these characteristics can be summarized in the following definition:

Definition 2 *A* Collaborative Business Process *is a business process whose activities are executed by two or more autonomous organizations.*

Note, that the characteristics of a business process also apply to a collaborative business process: Since a business process must have a clearly defined goal, this implies the joint creation of an output for a customer, not distinguishing if this customer is internal (e.g. belonging to one organization participating in the cross-organizational process) or if it belongs to third parties. It implies also, that the sequence, in which the activities are executed by the collaboration partners, is specified; which, on the other hand, implies coordination between the organizations to describe and implement collaborative business processes.

2.1.3 Interoperability, Business Interoperability and Integration

Due to the need to connect the growing variety of technical systems, the term interoperability was heavily used in the last decades. As stated in the introduction of this thesis, a vast number of publications and research projects indicate the prevailing importance of interoperability research. Corresponding publications can be traced back at least to the year 1967; however, the number of publications on interoperability rose significantly after the year 2000.[115] Not surprisingly, many different definitions of interoperability exist.[116] FORD ET AL. identified 34 definitions for interoperability and found that the definition most often cited in literature describes interoperability as "*the ability of systems, units, or forces to provide services to and accept services from other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together*"[117]. Interoperability is often associated with the following characteristics:

- Possibly heterogeneous systems: Interoperability of systems is usually discussed only in cases, where the accomplishment of interoperability is non-trivial, e.g. when *heterogeneous* systems are supposed to interoperate. In this vein, KLISCHEWSKI & SCHOLL state that interoperability refers to systems and applications with the following characteristics: independency, heterogeneity, and control by different jurisdictions/administrations or by external actors.[118]

114 Compare p. 16.
115 Compare FORD ET AL. (2007).
116 Comparisons of interoperability definitions can be found for example in PERISTERAS & TARABANIS (2006), SCHMIDT ET AL. (2007), RAY ET AL. (2008).
117 Cited from FORD ET AL. (2007). According to them, this definition was probably used already in 1967 in the US Department of Defense (DoD), where it is still being used today.
118 KLISCHEWSKI AND SCHOLL (2006), p. 2.

- Limited need to modify internal systems: Often, interoperability is associated only with the communication layer of a system, but not with its core elements.[119] Thus it is assumed, that the internal structure of a system can remain unmodified and only the systems' interface has to be modified to enable interactions with other systems.[120] However, the establishment of interoperability might require (small) modifications of the involved systems; "business process interoperability", for example, also requires the processes of the interacting partners to be compliant with each other.[121]
- Interlinkage of systems: The EUROPEAN COMMISSION understands interoperability as "the means by which the *inter-linking* of systems, information and ways of working, whether within or between administrations, nationally or across Europe, or with the enterprise sector, occurs".[122]
- Exchange and comprehend information: The notion of "interlinkage" can be refined by specifying that interoperable systems are able to exchange information and to interpret this information correctly. Accordingly, an often-cited definition describes interoperability as "the ability of two or more systems or components to *exchange* information and to *use* the information that has been exchanged"[123].
- Consume and provide services: Going beyond the exchange of information, many definitions state that interoperability means to *provide and consume services* across system boundaries, or, with a similar meaning: the ability to access the functionality of peer systems.[124]
- Work together: If systems are exchanging information in a meaningful way and exchange services as well, it can be said that they work together; they *inter-operate*. Thus, many definitions of interoperability state that systems should be able to "work together" or to "operate together".[125]

Since the notion of possibly heterogeneous systems that work together implies their interlinkage, their ability to exchange and comprehend information as well as the ability to invoke the services provided by each other, in the following interoperability is understood as follows:

Definition 3 Interoperability *is the capability of autonomous systems to work together.*

---

[119] Thus, the UNIVERSITY CONSORTIUM FOR GEOGRAPHIC INFORMATION SCIENCE (1996) stated, "the term interoperability refers to a bottom-up integration of existing systems and applications that were not designed to be integrated when they were built".

[120] In the same vein HANSON, NANDI & KUMARAN (2002), p.68, state "Interoperability means connecting up the business processes with the economy – not turning the business over to someone else" and "interoperability technology shouldn't place constraints on how the core of the business works". Similarly, the European Commission states that it is unrealistic to expect a harmonization between European public administrations and proposes the creation of Business Interoperability Interfaces. Compare EUROPEAN COMMISSION (2004), p. 18.

[121] Compare ZDRAVKOVIC & KABILAN (2005), p. 87.

[122] EUROPEAN COMMISSION (2003), p. 6.

[123] Compare FORD ET AL. (2007).

[124] Compare for example the interoperability definitions listed in FORD ET AL. (2007).

[125] Compare for example JANSEN & SCHOLL (2007); similar definitions can be found in FORD ET AL. (2007) as well as in the definitions cited by RAY ET AL. (2008).

Obviously, the concepts of interoperability and collaborative business processes are closely related. For example, (technical) interoperability is a pre-condition for the automation of CBP: If partners in a CBP cannot exchange meaningful information with each other, the CBP cannot be executed. This close relationship is also confirmed by LEGNER & WENDE, who argue that previous interoperability definitions focused too much on technical aspects. In consequence, they propose the term Business Interoperability, which they defined as "the organizational and operational *ability of an enterprise to cooperate* with its business partners and to efficiently establish, conduct and develop IT-supported business relationships with the objective to create value"[126]. Another indicator for the close relationship between the concepts of interoperability and CBP is that concepts from the area of cross-organizational workflow are often intertwined with interoperability concepts.[127] Building on the interoperability definition provided above and the definition of collaborative business processes, Business Interoperability in the following is understood as follows:

Definition 4 Business Interoperability *is the capability of organizations to execute a collaborative business process among them.*

Note, that this definition leaves open how complex the collaborative business process is, e.g. it could be as simple as "enterprise A sends message X to enterprise B". Thus, Business Interoperability does not necessarily refer to complex interaction sequences, but – in case of simple forms of "working together" – it can also refer to one-step interactions. The notion of *organizational interoperability* used in the European Interoperability Framework (EIF) is closely related to our understanding of Business Interoperability.[128] However, on the one hand, the definition of organizational interoperability leaves more room for interpretation (since it covers also strategic, high-level issues); while on the other hand, it excludes elements necessary to enact a collaborative business process, like its technical description or the specification of exchanged data.[129]

Integration vs. Interoperability The beginning of information systems in the 1960's was characterized by isolated application systems, which inhibited an efficient information flow between individual workplaces and departments. The reduction of such barriers by integrating these application systems is a major topic of information systems research.[130] In general, integration is defined as the "*process of combining two or more things so that they work together*"[131]. A prominent example for information systems integration is the development of databases, which integrate data from different departments into one consistent, enterprise-wide data repository.[132]

Usually different integration dimensions are distinguished, for example integration of data, functions, processes and objects.[133] Another often-used distinction is between *horizontal* and *vertical*

---

[126] LEGNER & WENDE (2006), p. 3.

[127] Thus, a classification of cross-organizational workflow types designates each of them as a different type of interoperability. Compare VAN DER AALST (1999). A slightly modified classification is also used by CHEBBI, DUSTAD AND TATA (2006).

[128] The European Interoperability Framework distinguishes between three types of interoperability: Organizational, semantical and technical interoperability (compare EUROPEAN COMMISSION, 2004); refer also to pp. 84.

[129] Compare EUROPEAN COMMISSION (2004).

[130] Compare MERTENS ET AL. (2005), p. 8, and FERSTL & SINZ (2008), pp. 232.

[131] OXFORD (2005), entry "integration".

[132] Compare SCHEER (1989).

[133] Compare for example ROSEMANN (1996), pp. 157, MERTENS ET AL. (2005), pp. 7 or FERSTL & SINZ (2008), pp. 232.

integration,[134] where vertical integration refers to the integration of elements having the same level of technical granularity, like for example the integration of value chain activities. Vertical integration spans different levels of technical granularity, e.g. the integration of a decision support system with fine-grained, technical data. In addition, *connecting* and *merging integration* can be distinguished.[135] A connecting integration is the creation of a system out of hitherto unconnected, though logically related subsystems. In the connecting integration, these subsystems are explicitly related to each other, but not altered themselves. In the merging integration on the other hand, elements of the subsystems may be altered; for example, in case that two subsystems contain the same element, redundant elements can be deleted.[136]

This implies that integration is closely related to the definition of interoperability as the *ability of systems to work together*. Correspondingly, interoperability and integration are often treated as adjacent concepts, where – compared to integration – interoperability refers to a looser coupling of systems.[137] Different from integration, interoperability only refers to a connection of systems where the connected systems themselves are left intact.

2.2 Modeling Collaborative Business Processes

Building on the definitions of the previous section, in this section essential concepts and terms related to the modeling of collaborative business processes are described. The first two sub-sections describe basic concepts like information system, model, enterprise architecture or interface. Relating these generic terms to concepts needed in collaborations, concepts like the Business Interoperability Interface and ontologies are also discussed. Representing a main pillar of the AIOS, in the following three sub-sections the concept of modeling private, public and global views on collaborative business processes models is described.

2.2.1 Modeling of Information Systems and Related Concepts

The terms described in the following represent core concepts of information systems research. On the one hand, this implies an exhaustive number of perspectives, characterizations and definitions. On the other hand, due the variety of researchers that use them, often no generally accepted definition of these terms exist. In the following, the understanding of these terms as used in this thesis is briefly described.

2.2.1.1 System, Model and Metamodel

A *system* is usually understood as a set of elements, the characteristics of the elements and the relationships among the elements. Since the elements of a system can comprise elements and relationships, hierarchical subsystems can be developed.[138] *Models* are an important instrument for the

---

[134] Compare for example MERTENS ET AL. (2005), p. 8, or ROSEMANN (1996), pp. 160.
[135] Compare ROSEMANN (1996), pp. 167.
[136] Compare ROSEMANN (1996), p. 155.
[137] Compare for example CARNEY ET AL. (2005), p. 9.
[138] Compare FERSTL & SINZ (2008), p. 13. For a detailed description of different system types and a discussion of systems in the context of collaborative business refer to WERTH (2006), pp.53.

analysis and the design of complex systems. In consequence, they take a central role in many sciences, also in information systems research.[139] A model is often defined as an abstracted representation of a system, where the way of the abstraction follows a certain purpose.[140] In information systems research, a model is a representation of an enterprise or parts of it.[141] Thus, a "model" represents an umbrella term for different model types, including organization structure models, business process models and data models.[142] *Modeling* is understood as the creation of a model.[143]

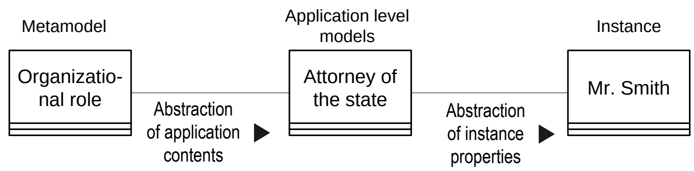


Figure 6: Relationships between metamodel, model and instance[144]

A *metamodel* is a model of a model, i.e. it abstracts from a model.[145] A metamodel should serve as a frame, which describes the possibilities and restrictions of model construction. To this purpose, a metamodel specifies the available model building blocks and the possible relationships among them; it can as also describe rules for the usage of these building blocks and their relationships.[146] Often metamodels are abstracted again, thus hierarchies of metamodels can be developed. For example, a hierarchy of meta-metamodel, metamodel, model and instance is used by the OMG as well as in the Architecture of Integrated Information Systems (ARIS).[147] Here, the instance level displays the concrete entities represented by the models, e.g. executed functions, persons or concrete data values.[148] Figure 6 exemplifies three of these four abstraction levels. For the development of the AIOS, the metamodel and the model level are of most relevance. Thus, in Chapter 4 the metamodels of the AIOS dimensions are described, while in Chapter 5 the usage of the metamodels will be exemplified by describing models used in a real-life scenario. To describe the metamodels of the AIOS dimensions, UML Class Diagrams are used; hence, UML Class Diagrams act as meta-metamodels.

---

139 Compare FETTKE (2006), p. 26.
140 Compare for example FERSTL, SINZ & AMBERG (1996) or OMG (2003).
141 Compare FRANK (1994), pp. 11.
142 Compare FETTKE (2008), pp. 17. For a more detailed description of the characteristics of models in the context of information systems and collaborative business, refer for example to THOMAS (2005) and WERTH (2006), pp. 109.
143 Compare SINZ (2001).
144 The abstraction levels used in the graphic are similar to those used in ARIS; compare SCHEER (1999), p. 29.
145 Compare STRAHRINGER (1996), p. 23.
146 Compare SINZ (2001 Model). In a similar vein, FINK, SCHNEIDEREIT & VOß (2005), pp. 93, stated that, apart from describing the syntax and semantics of model elements (static-structural metamodel), a metamodel can also describe the development steps for constructing a model (processoral metamodel). Other authors refine this understanding, by stating that a metamodel describes also the *notation*, e.g. the symbols that can be used in modeling (compare for example WESKE, 2007 and BALZERT, 2000, p. 38).
147 Compare WESKE (2007), pp. 75, and SCHEER (1999), p. 29. ARIS is described in detail in Chapter 3 (compare pp. 78).
148 Compare WESKE (2007), pp. 75.

2.2.1.2 Modeling Language, Syntax and Semantics

A modeling language is an artificial language used to create models. Different from programming languages, modeling languages are usually defined by a metamodel, and not by a grammar.[149] A modeling language comprises a set of symbols, a syntax specification and semantics. The concrete syntax of a modeling language describes the visualization of the symbols, e.g. the notation. The abstract syntax describes the set of available symbols and rules for their arrangement. The semantics of a modeling language describe the meaning of the symbols; complementing the syntactical rules, the semantics can declare certain symbol arrangements as incorrect, even if these are syntactically correct.[150] Modeling languages used in this thesis comprise for example the Event-driven Process Chain (EPC), the Business Process Modeling Notation (BPMN) and the Unified Modeling Language (UML).[151]

2.2.1.3 Semantics and Ontologies in the Context of CBP

As described above, two aspects of models can be distinguished: syntax, describing the form of a model, and semantics, describing the meaning of the form. Thus, semantics map the syntax of a model to the object that is described by the model, ensuring that different stakeholders associate the same meaning with a model. Since among different enterprises the chances that stakeholders have a different conception of a model is higher than in an intra-organizational setting (where a higher amount of shared conceptions can be assumed), here the need for mechanisms to support a correct understanding of models is particularly high.

In this vein, syntactic and semantic interoperability can be distinguished, where syntactic interoperability ensures that exchanged information is in compatible formats, while *semantic interoperability* ensures that exchanged information has the same meaning for both the information sender and receiver.[152] The EUROPEAN COMMISSION describes semantic interoperability as one of three types of interoperability (organizational, technical and semantic interoperability), and states that semantic interoperability "is concerned with ensuring that the precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose"[153].

Ontologies are also supposed to enable a common understanding of models among different stakeholders. In information systems research, several interpretations of the term ontology exist,[154] though often the definition of GRUBER is followed: "*A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. ... An ontology is an explicit specification of a conceptualization*". This definition illustrates the proximity of the terms model and ontology: both are an "abstract, simplified view" used to represent a selected part of the world in order

---

149 Compare ENGELS (2008).
150 Compare FRANK & VAN LAAK (2003), pp. 20.
151 Compare KELLER, NÜTTGENS & SCHEER (1992), OMG (2006 BPMN) and OMG (2000). These languages are described below (pp. 68).
152 Compare RAY ET AL. (2008).
153 EUROPEAN COMMISSION (2004), p. 16.
154 Compare also FETTKE (2006), p. 54, who even states that the term ontology is often defined so poorly, that it is not possible to distinguish it from any arbitrary term.

to fulfill a certain purpose using a specified language ("explicit specification").[155] In this vein, an enterprise metamodel can also be seen as an ontology.[156]

Nevertheless, going beyond the usual definition of a model, ontologies are often seen as a vocabulary that is developed by a group of human beings, and is jointly used and accepted in a group as well. Furthermore, an ontology is not only used for one purpose, but can be used in different contexts.[157] GÓMEZ-PÉREZ, FERNÁNDEZ-LÓPEZ & CORCHO distinguish ontology types according to the area described by the ontology, for example, general ontologies and domain ontologies. Further, they distinguish between lightweight and heavyweight ontologies, where the latter are formally specified and enable logical reasoning, e.g. the creation of new correlations among concepts in the ontology.[158] Obviously, an ontology that could map the concepts used in different enterprises to each other and enable a shared understanding of them, would be quite useful for collaborative business.[159]

2.2.1.4 Information System Architecture

Information systems are systems that process information, i.e. they capture, transport, transform, store and offer information within an organization or between an organization and its environment.[160] Often an information system is seen as a computer-based application system, for example a program system that aims at the fulfillment of business objectives. However, in a conception prevailing in information systems research, an information system comprises not only the hardware and software of an enterprise, but also the related human actors, business functions and processes as well as organization structures.[161] Following this understanding, only that part of an information system that comprises automated elements is called an application system.[162] Information systems are often process-oriented systems, realized, for example, as workflow-systems that are based on networks and SOA.[163] This process-centric understanding of information systems is also implied by the popular Architecture of Integrated Information Systems,[164] which only tackles elements directly related to business processes, e.g. functions, data, organization and process output.[165]

---

155 For a comparison of the characteristics of models and ontologies compare also FETTKE (2006), p. 107.

156 Compare ZOUGAR, CHEN & VALLESPIR (2008).

157 Compare FETTKE (2006), p. 107.

158 Compare GÓMEZ-PÉREZ, FERNÁNDEZ-LÓPEZ & CORCHO (2004), pp. 28.

159 In this vein, DING ET AL. (2004), p. 613, even state "*... eCommerce has to deal with serious problems ... only ontology technology can promise to provide at least a partial solution*". The problems they refer to are that in eCommerce different standards will always exist, that probably no standard can cover all aspects needed in eCommerce, that the concepts used in eCommerce need to be described beyond a syntax level, and that standards need to be adapted and extended on a frequent basis.

160 Compare BERNUS & SCHMIDT (2006), p. 2, and FINK, SCHNEIDEREIT & VOß (2005), p. 1.

161 Compare BECKER & SCHÜTTE (2004), p. 33, and GABRIEL (2008). This broad understanding is also confirmed by the often-referenced "Framework for Information Systems Architecture" (ZACHMAN, 1987).

162 Compare FERSTL & SINZ (2008), p. 4.

163 Compare GABRIEL (2008).

164 Compare SCHEER (1999).

165 For a discussion of the term information system and a comparison of its different interpretations refer for example to FERSTL & SINZ (2008), pp. 9.

Architecture Similar to terms like "model" or "information system", everyone roughly knows what an architecture is, but there is also no universally accepted definition.[166] According to SCHEER, in information technology, an architecture "describes the type, the functional properties and the interrelationships among the individual building blocks of the information system" [167]. In a similar vein, BASS, CLEMENTS & KAZMAN state, "the software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements and the relationships among them"[168]. They complement their definition with the statement that "architecture is high-level design", which is also indicated by the often-cited definition of the IEEE 1477 standard: Architecture is the "*fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution*"[169]. Information system architectures can be domain specific or domain independent: While ARIS is an example for a domain independent architecture that can be applied in sectors as different as manufacturing, banking or eGovernment,[170] the Y-CIM (manufacturing), Handels-H (service industry) or the TEAF (eGovernment) are examples for domain specific architectures.[171]

The definition of the IEEE already indicated that an architecture can comprise both *descriptive* (or static) and *constructive* (or dynamic) aspects: The descriptive aspect tackles static elements of an architecture, e.g. the system elements and their relations, comprising the various levels and views contained in architecture models. The constructive aspect, on the other hand, provides methods for the development of the system described by the architecture.[172] SINZ shares this judgment, and states that an information system architecture represents the building plan of an information system in the sense of a specification and documentation of its components and their *relationships covering all relevant viewpoints* as well as the *construction rules* for the creation of the building plan.[173] In comparison to the previous definitions, here it also mentioned that architectures comprise various viewpoints. Since this separation of concerns helps to describe complex systems, most information system architectures provide various views (also called dimensions) to describe complementary aspects of a system.[174] VAN DER AALST ET AL. even describe such views as a constituent element of an architecture.[175]

In this work, the information system architecture conception of SCHEER is followed. It is complemented with the conception of VAN DER AALST ET AL. that views constitute elements of an architecture and SINZ that an architecture consists both of descriptive and constructive aspects.

---

166 Compare also VAN DER AALST ET AL. (2007), p. 142, or BASS, CLEMENTS & KAZMAN (2006), pp. 23.
167 SCHEER (1999), p. 1.
168 Compare BASS, CLEMENTS & KAZMAN (2006), p. 21.
169 IEEE (2007). IEEE stands for "Institute of Electrical and Electronics Engineers".
170 Compare for example SCHEER ET AL. (2004), where the application of ARIS in these and other domains are described.
171 Compare SCHEER (1994 CIM), BECKER (1996) and DEPARTMENT OF THE TREASURY (2001). TEAF stands for "Treasury Enterprise Architecture Framework".
172 Compare also HEUTSCHI (2007).
173 Compare SINZ (2002), p. 1055. This is also confirmed by SCHMIDT (1999), p. 2, who writes that an extended architecture notion comprises also the procedure model for the development of a system.
174 Compare for example SCHEER (1999), ZACHMANN (1987) or BEECKMAN (1989).
175 Hence, they define an architecture of a system as "a set of descriptions that present different views of the system. These views should be consistent and complete. Each view models a set of components of the system, one or more functions of each component, and the relationships between these components" (VAN DER AALST ET AL., 2007, p. 95).

2.2.1.5 Enterprise Models vs. Business Process Models

Though the term enterprise model indicates a broader scope than the term business process model, the usage of both terms is strongly overlapping and sometimes synonymous. One indicator for the proximity of them is provided by VERNADAT, who states, that enterprise models describe the various aspects of an enterprise (process models, data models, resource models etc.) on different abstraction levels (e.g. business level and engineering level).[176] This description of enterprise models fits exactly with business process frameworks like ARIS, where the history of ARIS also exemplifies the close relationship between both terms: Based on enterprise-wide data models, ARIS was first described as an enterprise modeling framework but today is usually seen as a business process framework.[177]

The proximity of enterprise models and business process models is not surprising: Enterprise models usually are supposed to describe those parts of an enterprise that are relevant for information technology and represent the basis for automating business processes.[178] Accordingly, in a recent survey among 219 researchers on the objects that should be described in enterprise architectures, business processes were ranked as the most important object.[179] On the other hand, a business process describes those parts of an enterprise that – directly or indirectly – produce value. This implies that elements of an enterprise model, which are *not* related to a business process, are not relevant for the value generation and thus are redundant in an enterprise model. Note that this argument is based on the assumption that a business process does not only comprise the control flow, describing the sequence of functions, but also other enterprise dimensions related to the control flow. Apart from the data consumed or produced by functions, such dimensions cover for example organization elements related to the functions of a business process.

Thus, it is important to distinguish between a *business process model* and a *control flow model*. Accordingly, in ARIS, a business process model comprises different enterprise modeling dimensions, where the control flow dimension is only one besides the organization, data, output, and function dimension.[180] This comprehensive notion of business process modeling is confirmed for example by WESKE, who subsumes process, information, function, IT landscape, and organization modeling under the term business process modeling.[181] However, also narrower conceptions of the term business process (model) exist; e.g. in the context of SOA the understanding of "business process" is sometimes restricted to the control flow.[182]

---

[176] Compare VERNADAT (1992), p. 1.

[177] Compare SCHEER (1989), describing "Enterprise-Wide Data Modeling", SCHEER (1992), where ARIS is designated as a basis for "Enterprise Modeling", and SCHEER (2000), where ARIS is described as a means for business process modeling.

[178] Compare also FINK, SCHNEIDEREIT & VOß (2005), p. 91. This was confirmed by FRANK (2008), who stated that enterprise models originally were focusing on the development of information systems, though in recent years this notion has been extended and some authors demand that enterprise architectures should also cover high-level concepts, not directly related to the development of information systems.

[179] Compare AIER, RIEGE & WINTER (2008), pp. 293.

[180] Compare SCHEER (1999).

[181] Compare WESKE (2007), pp. 77.

[182] The Web Services Business Process Execution Language (BPEL), for example, focuses on the sequence of functions comprised in a process but lacks an organization dimension. Compare OASIS (2007).

In this work, the enterprise model conception implied by ARIS is followed, that an enterprise model describes the different dimensions (e.g. process, data, organization) of an enterprise on different levels of granularity (e.g. business, technique and execution) that are relevant for the description or the enactment of business processes.

2.2.1.6 Enterprise Architecture vs. Information System Architecture

THE OPEN GROUP describes an enterprise architecture as the fundamental organization of a government agency or a corporation, either as a whole, or together with partners, suppliers or customers ("extended enterprise"), or in part (e. g. a division, a department), as well as the principles governing its design and evolution.[183] Prominent examples of enterprise architectures comprise the Zachman framework, TOGAF and ARIS.[184]

A similar proximity to that between enterprise and business process models exists between the terms enterprise architecture and information system architecture. Traditionally, enterprise models and enterprise architectures focus on IT related artifacts and thus capture only elements relevant for the development of information systems.[185] Correspondingly, AIER states, "*enterprise architecture is the combination of organizational, technical, and psychosocial aspects during planning and development of socio-technical business information systems*"[186]. WINTER & FISCHER, on the other hand, propagate a distinction between information systems architecture and enterprise architecture. They argue that – going beyond information system architecture – enterprise architecture should also cover "purely" business-related artifacts, like markets, organizational goals and performance indicators.[187] However, with this incorporation of strategic aspects, they deviate from most other enterprise architecture conceptions, which usually focus on the description and enactment of business processes and organization structures.[188] It can be summarized, that usually the conception of enterprise architectures is *synonymous* to information system architecture. Thus, both ARIS and the AIOS can be understood not only as enterprise architectures, but – since they cover only elements that are necessary to describe and enact business processes – also as information system architectures.

Conclusions Figure 7 recaptures the concepts discussed above and the relationships among them. The term *enterprise architecture* implies a higher-level description of an enterprise. Since the enterprise model represents an operational description of selected enterprise elements, the relationships illustrated in Figure 7 can be inferred: An enterprise architecture describes the principles and

---

183 THE OPEN GROUP (2003).

184 TOGAF stands for "The Open Group Architecture Framework". TOGAF and other enterprise architectures are described in Chapter 3, pp. 78.

185 Compare WINTER & FISCHER (2007), p. 8 or FRANK (2008).

186 AIER & SCHÖNHERR (2006), p. 3. A similar definition stems from LANKHORST (2005, p. 3). Though he does not refer explicitly to information systems, his definition clearly focuses on information systems and the enactment of business processes: an enterprise architecture is "*a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure*".

187 Compare WINTER & FISCHER (2007), pp. 8.

188 Compare AIER, RIEGE & WINTER (2008), p. 294.

the structure of an enterprise model.[189] An enterprise model represents real world elements of an enterprise, displaying either the as-is or the to-be state. In this work, *enterprise architecture* and *information system architecture* are treated as being synonymous – the terms *information systems model* and *enterprise model* are used synonymously as well. Both, the architecture and the model, describe only elements relevant for the enactment of business processes. The information system – comprising for example workflow engines, databases and human actors – on the other hand, executes the business processes. Paying tribute to the broader enterprise architecture conception of other authors, "elements not related to information systems" are included on the lower left side of the graphic.

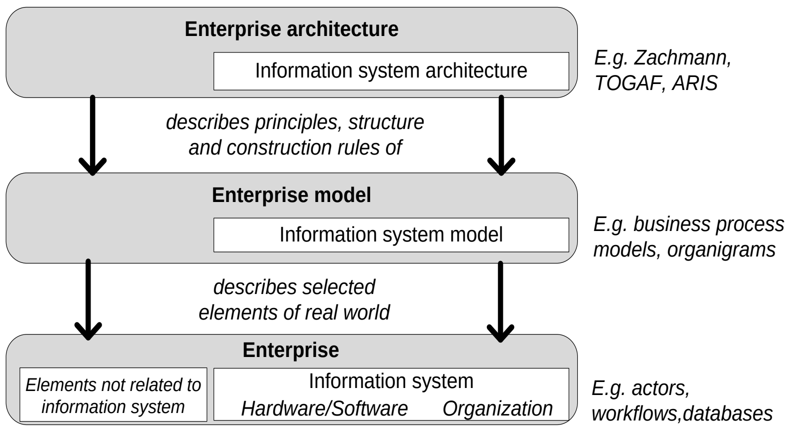


Figure 7: Enterprise architecture, enterprise model and information system

2.2.2 Interfaces of Information Systems

Information systems usually consist of many subsystems; to connect these subsystems, interfaces are used to describe those characteristics of a system relevant for the adjacent systems. Modern software architectures like SOA confirm the importance of interfaces in information systems.[190] While interfaces in information systems formerly concentrated on technical aspects, e.g. by describing Application Programming Interfaces, SOA shifted the attention to conceptual interfaces aiming at a business-level description of services. Since collaborative business always comprises interactions between systems, interfaces are an essential element in the development of collaborative business processes.[191]

2.2.2.1 Objectives and Types of Interfaces

Two closely related conceptions of the term interface can be found in literature:[192]

---

[189] In a similar vein, SCHMIDT (1999), p. 2, writes that an architecture determines the required model types and the relationships among them.
[190] As described below, interface-orientation is a major principle of SOA, see pp. 58.
[191] Compare also HOFER (2007), pp. 68, or WERTH (2006), pp. 146.
[192] Compare also BELLMANN (2002), p. 363.

First, the conception of an interface as the *accessible surface of a system*; thus in Object-oriented programming and SOA, an interface is often defined as the set of all public methods of an object (or a service).[193] In the same vein, FERSTL & SINZ speak of an interior and an exterior view on a system, where the latter comprises the interfaces of a system.[194] Also in the context of SOA, interfaces do not only describe how to interact with a system (e.g. a service), but are also used to *identify systems*; thus, KARAKOSTAS & ZORGIOS state that "an interface is capable of fully and completely defining the type of service the provider is offering"[195].

Second, the conception of an interface as a means to enable *interaction between systems*. Thus, CAMBRIDGE defines an interface as "a connection between two pieces of electronic equipment, or between a person and a computer"[196]. In the following, an interface is understood as the surface of a system that enables interactions with other systems. In this function, an interface can fulfill several functions:

- Describe interactions: An interface describes how other systems can interact with a system. For example, all messages that a system can send to or receive from other systems in a defined context (like a specific collaboration) can be described in an interface.
- Contract: An interface can be understood as a binding contract, in which the provider and the user of a system describe the behavior to be fulfilled in a collaboration.[197]
- Control information dispersal: Interfaces represent only those characteristics of a system that should be communicated to adjacent systems, while other characteristics are held back. Apart from protecting internal systems from unwanted information disclosure, it also protects adjacent systems from irrelevant information.
- Represent overall system: The interface description can enable the identification of the described system, e.g. the discovery of a specific system in a set of other systems.
- Intermediate between systems: During run time, the interface can be implemented as a proxy, where other systems direct their messages to. In this function, an interface can also translate heterogeneous communication formats used inside and outside the system.

These functionalities also imply that, depending on the requirements and the trust relationship between organizations, one system can be represented via different interfaces to different partners.

Technical vs. Business Interfaces Two types of information system interfaces can be distinguished: technical and business-oriented approaches.[198] While the latter type takes a holistic perspective on information systems, the first type concentrates on technical aspects. Taking a technical perspective, VOIGTMANN & ZELLER distinguish between three different types of interfaces: (Graph-

---

[193] Compare HANSEN & NEUMANN (2001), p. 265, and JURIC, MATHEW & SARANG (2006), p. 13.
[194] Compare FERSTL & SINZ (2008), p. 21.
[195] KARAKOSTAS & ZORGIOS (2008), p. 26.
[196] CAMBRIDGE (2005), entry "Interface".
[197] In the context of SOA for example, JURIC, MATHEW & SARANG (2006, p. 13) state that an "interface is a contract between the service provider and a service consumer".
[198] A similar distinction is drawn for example by HOFER (2007), p. 68, who separates technical and organizational interfaces.

ical) user interfaces, application interfaces and data interfaces.[199] In the context of SOA, usually no distinct data interfaces are offered, only service interfaces as described, for example, by the Web Service Description Language (WSDL).[200] These interfaces often serve as components wrappers, where a wrapper represents a software layer that encapsulates legacy data and logic.[201] Further descriptions of technical interfaces can be found in literature on Object-oriented or Service-oriented approaches.[202]

Compared to the amount of literature on technical interfaces, relatively few approaches tackle interfaces on an organizational, business-oriented level.[203] In the context of business administration, interfaces are discussed on a rather abstract level. BELLMANN, for example, investigates interfaces between organizational departments.[204] Nevertheless, he concentrates on organizational problems, for example, cases where departments separated through interfaces pursue conflicting objectives, but he does not tackle information system interfaces. ZELEWSKI explicitly tackles interfaces in business information systems, but stays on a high level of abstraction; he distinguishes, for example, between data, function and process-oriented approaches for the design of interfaces, but does not explain the contents of these views.[205] Based on the information systems conception of FERSTL & SINZ,[206] WERTH proposes describing interfaces based on three orthogonal axes: type of information system, transaction object and actor.[207] The first dimension refers to a distinction of business information systems according to their "vertical" position in an enterprise: At the highest level, planning systems are used, at the intermediate level controlling systems and at the lowest level, operative systems are used. The second dimension refers to the object exchanged via the interface; a distinction is made between material, information and energy flows.[208] The third dimension represents the actor executing the interaction, which can be either human or automated; accordingly, three interaction patterns are described: machine-machine, human-human and human-machine.

Static vs. Dynamic Interfaces Orthogonally to their scope, interfaces can be distinguished into static and dynamic interfaces: *Static interfaces*, also called "structural interfaces",[209] do not describe sequences of interactions but only static elements, as for example organization units, business documents or functions comprised in one interaction. The interface diagram described by KLEIN, KUPSCH & SCHEER is static,[210] as well as most other interface types, like for example Java Application Programming Interfaces or service interfaces described with WSDL. They describe certain

---

199 VOIGTMANN & ZELLER (2003), pp. 18.

200 Compare W3C (2001),

201 Compare PAPAZOGLOU & VAN DEN HEUVEL (2007), p. 402.

202 Refer for example to MEYER (1997), who describes in detail the concept of "design by contract" in the context of Object-oriented programming. Examples for interface descriptions in SOA are provided for example by KELLER (2007), pp. 357, ERL (2008), pp. 139 or KARAKOSTAS & ZORGIOS (2008), p. 26.

203 This was already stated by ZELEWKSI (1987), pp. 551, and despite the upcoming of SOA still holds true today.

204 Compare BELLMANN (2002), pp. 363.

205 Compare ZELEWSKI (1987).

206 Compare FERSTL & SINZ (2008), pp. 2.

207 Compare WERTH (2006), pp. 147. In the German original, the three dimensions are named "Aufgabenphase", "Aufgabenobjekt" and "Aufgabenträger".

208 VANDERHAEGHEN (2009) describes similar exchange objects; however, instead of "material", he uses the term "output", and instead of an energy flow, he describes a monetary flow.

209 Compare BARROS ET AL. (2007), p. 246.

210 Compare KLEIN, KUPSCH & SCHEER (2004).

characteristics of a system that can be used to access it, for example, operation names and required input/output documents, but they do not describe in which sequence these operations have to be invoked. *Dynamic interfaces* describe the sequence in which systems can interact with each other. WERTH calls such interfaces "process interfaces" and defines them as a relationship between functions that specifies either their timely sequence or their logical correlation.[211] The public processes and choreography interfaces described in the next sections represent such dynamic interfaces.

2.2.2.2 Business Interoperability Interface

The research objective of developing a Business Interoperability Interface was already introduced in Chapter 1. As mentioned there, the development of a BII was recommended by the EUROPEAN COMMISSION to improve the interoperability of large, decentral organizations that internally use different standards. Though the BII specification of the EUROPEAN COMMISSION remains vague, the following requirements are described: The BII should describe entry and exit points to the internal business processes of an organization and it should describe the contribution and commitments the collaboration partners require from each other in a formal way. Furthermore, the BII should enable the enactment of collaborative business processes, e.g. "interconnect the 'national' business processes which might be completely different"[212]. This description implies that the main objective of the BII is to describe to collaboration partners how they can interact with an organization; in other words, how they can enact a collaborative business process with this organization. Correspondingly, a Business Interoperability Interface is defined as follows:

> **Definition 5** *The* Business Interoperability Interface *of an organization represents the sum of all models this organization has to provide to collaboration partners to enact a collaborative business process with them. It describes those elements of a collaborative business process, which are provided by the organization as well as the elements, the organization expects from collaboration partners.*

Note that this definition does not restrict the BII contents to the process dimension: in order to enact a collaborative business process, the collaboration partners also have to comprehend other enterprise dimensions complementing the process dimension, for example, organizational roles, individual business functions/services or document types. Likewise, the description of the BII is not limited to one level of technical granularity, since the systematic enactment of CBP requires models on both the business and the technical level.

Having the interaction description as a primary objective, a secondary objective of the BII is to describe the parts (for example services) of an organization in such a way that they can be easily discovered by collaboration partners. In respect to the generic interface characteristics described in the previous section, the BII should represent a comprehensive interface covering organizational, technical as well as static and dynamic aspects. Resorting to the classes described by WERTH, the BII should focus on information exchange and on machine-to-machine interactions – however, since the organigrams and business process models needed for process automation can be also used for purely human-based processes, the BII could also describe human-based interactions.

---

[211] Compare WERTH (2006), p. 155.
[212] Compare EUROPEAN COMMISSION (2004), p. 18.

2.2.2.3 On the Disadvantages of Interface Creation

No matter whether referring to organizational systems (e.g. a large enterprise) or computer systems: The complexity of modern systems usually demands their decomposition into various, abstracted parts in order to understand, optimize and control the overall system. The success of approaches like Object-orientation or SOA confirms the usefulness of modularizing information systems into different components. Since the use and reuse of software components implies a description of those characteristics of the components being relevant and accessible for other components, the necessity of interfaces is hardly arguable. Also in the context of collaborative business, the usefulness of describing the interfaces of the collaborating organizations to specify possible interactions is accepted.[213] Naturally, costs apply for the design and implementation of the additional layer encapsulating a system.

Apart from that, disadvantages associated with interfaces are often not caused by the interfaces, but rather by the fact that different (sub)systems exist, e.g. the concept of splitting a system into subsystems. This applies, for example, to the disadvantages described by MELZER-RIDLINGER, who states that interfaces foster delays and information losses. As a further disadvantage of organizational interfaces, she notes that members of separated business functions strive to reach department goals instead of enterprise goals.[214] In general, the decomposition of a system bears the risk that the resulting component-focus produces local instead of global optima, and – due to the focus of stakeholders on only one system element – an optimal outcome for the overall system can be missed. On the other hand, a finer-grained analysis and decomposition of a system can improve the possibilities of optimizing it. Thus, Figure 5 (p. 19) illustrated that process models based on fine-grained components allow for a better process optimization. More specifically, the graphic showed that the higher granularity has two effects: the overall system appears more complex and harder to understand, but it also leads to an increased process efficiency.

It has also been argued that the technical interfaces of web services imply a *black-box* approach that would contradict the white-box approach of process-orientation.[215] However, the existence of interfaces implies neither a black- nor a white-box approach, since not the mere existence of an interface determines the transparency of a system, but the degree to which this interface discloses system characteristics.[216] In summary, it can be said that the question is not whether to use interfaces or not, but to what degree a system should be disaggregated into subsystems and how closely these subsystems should be correlated with each other. While the latter aspect is tackled in this thesis, the question on the granularity of subsystems is discussed intensely in the context of SOA, where services have to be defined in an optimal granularity.[217]

---

[213] Compare for example WERTH (2006), pp. 146, HOFER (2007), pp. 68 or VANDERHAEGHEN (2009).

[214] Compare MELZER-RIDINGER (2007), p. 9.

[215] Compare KAHL, VANDERHAEGHEN & WERTH (2005).

[216] If an interface is understood as the form in which a system appears to an observer, it can even be said that everything is perceived only by its interface (as expressed in Plato's allegory of the cave), e.g. every (observed) thing has an interface.

[217] Refer for example to von HENNING (2007), pp. 343, STÖRRLE & GLOCK (2007) or HAESEN ET AL. (2008).

2.2.3 Private, Public and Global Views on Processes

Requirements for CBP Modeling As mentioned above,[218] in cross-organizational scenarios generally the need to explicitly describe and restrict the parameters of working together is higher than in intra-organizational scenarios. Thus, CBP modeling is torn between two conflicting demands:

- Trust – Provide unambiguous and sufficient information: To communicate, optimize and execute cross-organizational activities, collaborative business processes should be described precisely and in detail, allowing collaboration partners to adjust their processes optimally to each other. In this vein, in the literature the following requirements for modeling of CBP are named*: support semantic annotation, describe processes on different levels of technical granularity, provide a shared language, display semantic conflicts, and use formal specifications*.[219]
- Mistrust – Partner specific restriction of information flow and definition of responsibilities: On the other hand, organizations are reluctant to disclose (internal) information. In the private sector, this is because processes usually represent a competitive factor and the publishing of such knowledge would weaken the position of an organization. Also in public administrations, often highly sensitive data being subject to elaborated security regulations is transferred, e.g. data on citizens or data related to police work.[220] Here, two aspects of internal processes can be distinguished that need to be protected: first, individual items contained in internal processes, like a document covering citizens' information; second, the process logic, in case its' publishing would decrease the competitive position of an organization.[221] In this vein, in the literature, the following requirements for the modeling of CBP are named: *distinguish areas of responsibility, display roles of partners, enable a scalable exposition of internal processes, display interfaces, keep private information private*.[222]

In summary it can be said, that collaborative business process models should provide partners with the information necessary for a joint process execution, but should not reveal more internal information than necessary. From a practical standpoint, further requirements on collaborative business process modeling are that models should be re-usable and easy to connect with internal models.

---

[218] Compare p. 20.
[219] Compare ZIEMANN, MATHEIS & FREIHEIT (2007), pp. 28, and WERTH (2006), pp. 75.
[220] Compare DIEDRICH ET AL. (2007).
[221] Apart from the risk that a competitor could copy processes, another risk of disclosing internal processes is represented by criminal organizations, which can use process knowledge to identify vulnerabilities enabling them to temper the system.
[222] Compare ZIEMANN, MATHEIS & FREIHEIT (2007), pp. 28, WERTH (2006), pp. 75, and GREINER ET AL. (2006).

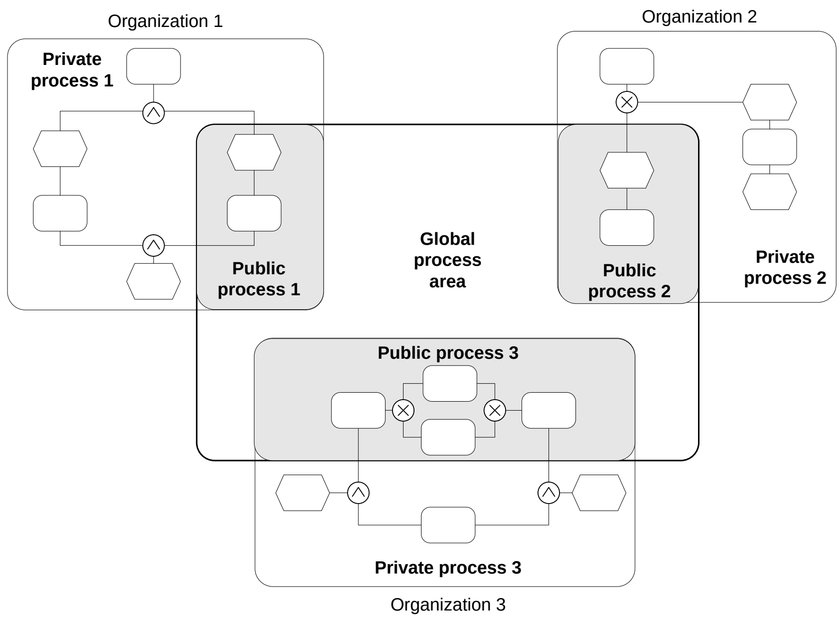


Figure 8: Private, public and global views on processes[223]

Three Different Views on Collaborative Business Processes Addressing these requirements, the distinction between private, public and global views on processes has been discussed for a couple of years in the context of cross-organizational processes. This approach distinguishes three types of processes:

- Representing the overall collaboration, a global process comprises all organizations involved in a collaborative business process and their (potential) interactions.
- Private processes are executed inside an organization and may contain non-public information. However, they can also contain interaction activities that should be published to the collaboration partners.
- Public processes comprise only those parts of the private processes that must be known by collaboration partners and thus are also part of the global process. Public processes can also be used to present a private process in different forms to different partners.

The concept of separating internal and external information representations was inspired by views used in databases, which allow publishing selected contents of a database to specific users. Building on database views, corresponding concepts were created for workflows, where information abstraction was achieved by the introduction of process views.[224] The idea of separating internal and external process views resulted in the private-public model as introduced by SCHULZ.[225] Alluding to their different degrees of information disclosure, the three process types described above are also referred to as white box, gray box and black box approaches.[226] Since it supports the

---

[223] The process annotation is aligned with EPC as defined in KELLER, NÜTTGENS & SCHEER (1992).
[224] Compare SCHULZ (2002) and SCHULZ & ORLOWSKA (2004).
[225] SCHULZ (1998).
[226] Compare SCHULZ & ORLOWSKA (2004), p. 121.

SOA concept of describing the observable behavior of services, it was implemented in standards for displaying service choreographies.[227] Building on workflow views, recently this technical concept was transferred to conceptual, business-oriented processes.[228] While workflow views focus on technical issues, the latter approaches focus on business concepts and organizational boundaries. Building on these approaches, in the following the terms private, public and global process are defined.

2.2.3.1 Private and Public Processes

As mentioned above, private processes are executed inside an organization. They represent best practice and thus a considerable intellectual property.[229] However, these internal activities can also represent interactions with collaboration partners. This is illustrated in Figure 8, where parts of the private process of one organization should be visible to partner organizations and thus are published as public process. Thus, private processes can be defined as follows:

> **Definition 6** *A* private process *is a business process created for internal use, representing only internally executed activities. In general, private process models should not be disclosed to external parties.*

Note that an interaction with other organizations, e.g. the sending or receiving of messages, is also seen as an internal activity.

A public process abstracts from a private process by describing only those parts that are relevant for the global process. Corresponding to the background of the different authors, many terms are used for public processes, including *abstract process, workflow view, choreography interface, local choreography, behavioral interface, virtual process, process skeleton, view process, process stub and process template*.[230] A public process can be derived from a private process (that contains cross-organizational interactions), by abstracting all confidential elements from it, leaving mainly interaction activities in the public process. Some authors strictly confine a public process to the communication activities of a private process.[231] Such a narrow definition applies for public processes on the execution level, where only the exchange of messages matters. However, on a business level, it can be useful to enrich a public process with additional information, e.g. an internal activity that helps collaboration partners to understand the overall process. While technical conceptions of public processes focus on digital message exchanges,[232] on a higher level, these models can also describe material exchanges, as well as the place and time of such exchanges.[233]

---

[227] Compare W3C (2002) and W3C (2006).

[228] The transfer of the workflow view concept to the business design level was realized in the ATHENA project; compare for example GREINER ET AL. (2006) or BORN ET AL. (2009). Today the terms private, public and global processes are established in the field of technically oriented business process management; compare for example WESKE (2007), pp. 219.

[229] Compare also SCHULZ & ORLOWSKA (2004), p. 123.

[230] Compare OASIS (2007), SCHULZ & ORLOWSKA (2004), W3C (2002), MENDLING & HAFNER (2005), BARROS, DUMAS & OAKS (2006), LIU & SHEN (2003), DANIEL & PERNICI (2006), GREINER ET AL. (2006), MENDLING & HAFNER (2005) and HOHPE (2007).

[231] Compare WESKE (2007), p. 221.

[232] Compare for example BUSSLER (2002) or WESKE (2007).

[233] Compare KLEIN, KUPSCH & SCHEER (2004).

Thus, a public process can be seen as an interface of one organization, which comprises only the information necessary for the interaction with one or more partners. Unlike static interfaces, it describes not only the operations offered by an organization, but also the sequence in which operations are executed. A public process can contain interactions with different partners. However, since it is described from the viewpoint of one partner, it only describes interactions between this partner and one or more of its partners (but not the interactions between its partners).[234] This leads to the following definition:

Definition 7 *A* public process *is a business process, which comprises those elements of a private process that are relevant and public for one or more collaboration partners. Its purpose is to describe the interactions of an organization A with its partner organizations (B, C, D…) from the viewpoint of A.*

2.2.3.2 Global Processes

While a public process describes an interaction sequence from the perspective of one organization and can omit other activities in a collaborative business process, a global process describes the activities of several organizations participating in a CBP from a neutral perspective. Thus, while the public process displayed in Figure 8 only captures interactions between the organizations 3 and 2 as well as the interactions between 3 and 1, the global process model comprises also the interactions between organization 1 and 2. As this description already indicates, the term global process is closely related to the term collaborative business process. The close relation between the concept of global process and collaborative, cross-organizational processes is also confirmed by the names that other authors use in place of global process: *coalition workflow* and *cross-organizational business process*.[235] However, while the term collaborative business process is used in many different contexts,[236] the term global process is normally confined to the context of process enactment, and, being closely related to execution level concepts like service choreographies,[237] has a more operational connotation than the term collaborative business process. Private elements, which are not relevant or not public (e.g. classified) for collaboration partners, are not part of the global process.[238] Hence, a global process can be defined as follows:

Definition 8 *A* global process *is a collaborative business process comprising only those elements that are public and relevant for all organizations in the CBP; its purpose is to describe the allowed interactions between these organizations.*

On the Extent of a Global Process Figure 9 illustrates that the scope of a global process depends on the objectives of the collaboration. Note that other than for the design, optimization and implementation of interactions, a global process can also depict the scope of collaborative monitor-

---

234 Protocols describing not only such "1:n" relationships but also "n:n" relationships are be specified by global processes. Note however that the public process of one partner may suffice to describe the interactions between all collaborating enterprises. This is the case, if this partner is in a central position ("1:n"), e.g. all interactions in the CPB are executed only between him and the other partners, but not directly among the partners.

235 Compare SCHULZ & ORLOWSKA (2004) and GREINER ET AL. (2006).

236 Compare the previous discussion of the different interpretations of collaboration (pp. 20).

237 Compare also the definition of choreography on p. 64.

238 This is also illustrated in Figure 8.

ing. In a supply chain scenario for example, it would make sense to include the organizations A, B, C and D in one global process; then all partners involved in the supply chain would be aware of mutual dependencies and – at run time – could be informed about the state of the process. For example, organization D could be notified early enough for it to take countermeasures if problems in organization A occur. However, to avoid unnecessary complexity and information dispersal, in general, global process models should be reduced to indispensable interactions with adjacent organizations. In the scenario displayed in Figure 9, it might suffice if only the adjacent units are connected in the form of global processes, resulting in the three global models [A, B], [B, C] and [C, D]; in this case, unit A would not have to deal with the collaboration [C, D]. Thus, usually a global process comprises only those organizations, which are relevant for the majority of the collaboration partners.[239]

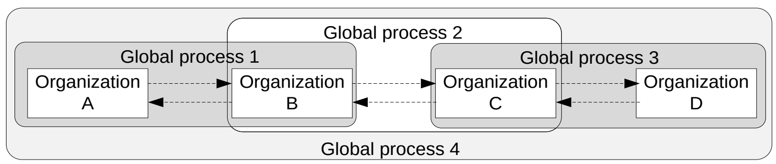


Figure 9: The extent of a global process depends on the objectives of a collaboration

Two Ways of Representing Global Processes Note that the definition provided above leaves open the question whether the model of the global process is created for central or decentral enactment. A model created for central enactment would describe the global process as seen from a neutral perspective, while a model created for decentral enactment would describe the global process as the sum of the public processes contained in it. Accordingly, two ways of representing global processes can be observed in standards for enacting global processes: First, the connection of public processes to a global process, e.g. *interface-based models* as illustrated on the left-hand side of Figure 10. Second, the representation of a global model as a sequence of interactions without (explicit) public processes, e.g. *interaction-based models* as illustrated on the right-hand side of Figure 10.[240] Note that these different representation forms do not imply differences in the interaction semantics.[241] The discrepancy is similar to the difference between a "normal" process model and a process model divided into swimlanes: The first focuses on activities and their sequence, but not on the allocation of activities to organizations. In the swimlane diagram, on the other hand, activities are positioned according to their organizational affiliation, e.g. all activities designated to organization A are situated only in the swimlane of organization A.[242]

[239] For example, if the organizations A, B and C form a global process, it might be that A and B are not interested in the collaboration of C and D. Hence, D should not be comprised in this global process but form a separate global process [C, D].

[240] These differences can be observed in BPEL, the Web Service Choreography Interface (WSCI) and WS-CDL: While in the case of BPEL Abstract Processes and WSCI, various public processes are combined into one global process, WS-CDL represents a global process as an interaction sequence without referring to the public processes comprised in it. Compare OASIS (2007), W3C (2002) and W3C (2006).

[241] E.g. the same global process can be represented either as a collection of connected public processes (like WSCI does) or as a sequence of interactions, like WS-CDL does. In case of WS-CDL, the roles responsible for a transaction are explicitly related to the transaction. Refer also to the descriptions of WSCI and WS-CDL in Chapter 4 (pp. 120).

[242] Interestingly, UML Activity Diagrams and BPMN support interface-based models (since both modeling languages support swimlanes), while interaction-based models are not supported – since individual functions cannot be corre-

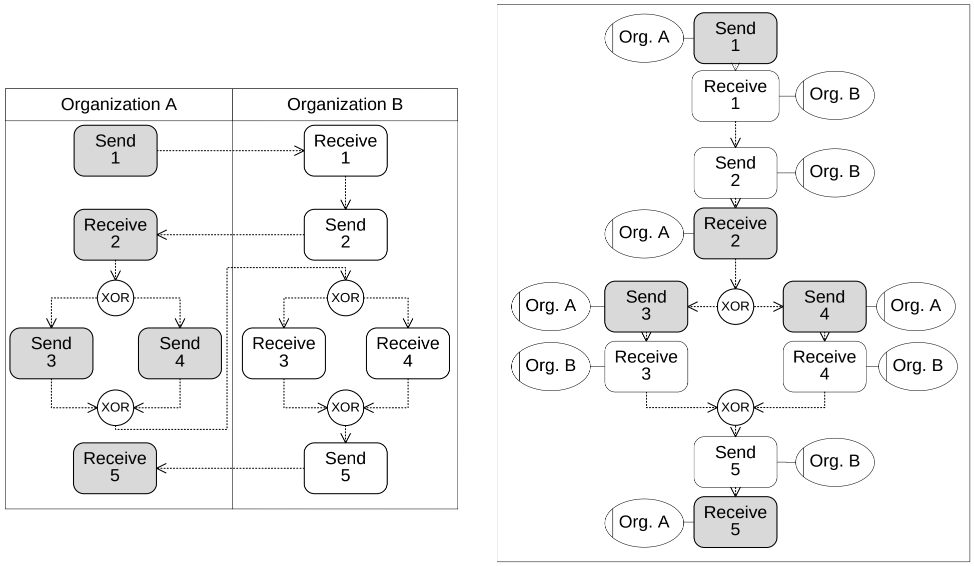


Figure 10: Interface-based vs. interaction-based representation of a global process

2.2.3.3 On the Operations Between Private, Public and Global Views

To refine the specification of the relationships between private, public and global process models, in the following it is described which operations are available to transform and correlate the models to each other. Resorting to the distinction between horizontal and vertical integration, in the following only horizontal operations are discussed, e.g. the source and target model of an operation have the same level of technical granularity.

In relation to the amount of literature on public and private processes,[243] little work exists on the exact relationship between these process types, e.g. how public processes can be derived from and correlated to private processes. This applies especially for the business level, where authors only describe vague or arbitrary transformation rules between private and public processes; or they neglect the consequences of such operations on the ability to execute processes. WERTH, for example, describes operations between private and public processes, but does not explain how these operations could be realized on the execution level or if the resulting process types could be implemented and related at run time. HOYER, BUCHERER & SCHNABEL on the other hand describe six transformation rules to derive public from private EPC models.[244] Though some of these rules make sense (e.g. to discard non-interaction functions in the public process), other seem arbitrary, like for example, the elimination of “trivial” events or organizational positions in the public process.[245]

lated to organization elements. The EPC, on the other hand, is well suited to display interaction-based models – since organization elements can be attached to functions – but originally did not support swimlanes. Advantages of the interface and the interaction-based representation of global process models are described in Chapter 4, pp. 132.

243 For example SCHULZ & ORLOWSKA (2004), SHEN & LIU (2001), CHEBBI, DUSTDAR & TATA (2006), SHAN ET AL. (2006), CHIU ET AL. (2002), ZIEMANN, KAHL & WERTH (2007), BORN ET AL. (2009) or OASIS (2007).

244 Compare HOYER, BUCHERER & SCHNABEL (2008).

245 They argue that trivial events are only needed for “business people”; however, since private and public EPC processes aim at an audience with a similar technological background, those would be needed in the public processes as well.

Models of private, public and global processes are closely related, complementary parts needed in CBP development. Figure 11 provides an overview of these model types and their relations: Organization A and B each derive a public process from a private process by hiding information not relevant for the collaboration partner. The resulting public processes are then correlated to form a valid global process. In general, these operations can have two directions: An inside-out development of collaborative business processes starts with private processes, from which public processes are derived and subsequently assembled into global processes. In an outside-in approach, first public processes are derived from global processes and then, private processes are derived from public processes.[246] Since the operations in the outside-in development equal the inverse operations of the inside-out approach, they are discussed only briefly in the following.

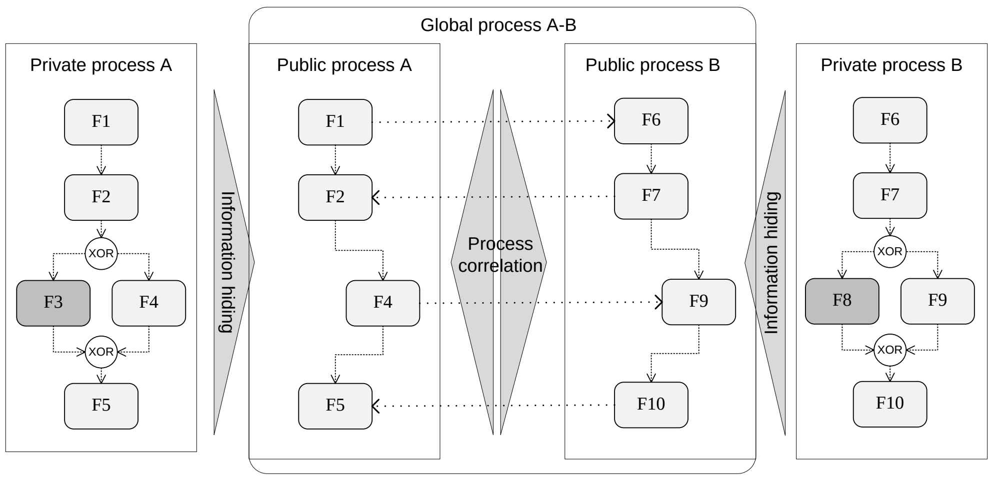


Figure 11: Overview of relationships between private, public and global processes

Eligible Operations In order to achieve abstraction and foster the re-usability of artifacts, in software engineering often the operations of information hiding and generalization/specialization are used.[247] Partly coinciding with these, in the construction of reference models the following operations are common:[248]

- Configuration: Here the resulting model consists of selected parts of the original model; in other words, parts of the original model are omitted. In the following, the operation will be referred to as *abstraction* and is used to derive public from private process models.
- Instantiation: In general, the instantiation of a model can be understood as the enhancement of a generic original model with additional elements.[249] However, in the context of software engineering, instantiation is understood as the creation of a tangible

Moreover, the knowledge about organization units may be relevant for collaboration partners; thus, instead of completely omitting organization elements in public processes, a mechanism should be used to only communicate those organization elements to the partner, which are relevant (and public) to it.

246 Both approaches are described in detail in Chapter 5, pp. 181.
247 Compare for example FRANK & VAN LAAK (2003).
248 Compare VOM BROCKE (2008).
249 Compare VOM BROCKE (2008).

run time object (e.g. an instance) from a more abstract model (e.g. a class in Object-oriented programming). Thus, instantiation is seen as a vertical operation and is not tackled further here.

- Aggregation/Integration: In an aggregation, different original models are integrated into one resulting model. Similar to integration, a merging and a connecting aggregation can be distinguished. In the following, the loss-less aggregation of different models into one model is referred to as *integration*; thus, different public process models can be integrated into a global process model. The merging composition of models or model elements into an aggregated model (or model element) is referred to as *aggregation* and is used to derive public from private process models.
- Specialization/Generalization: Specialization, as opposed to generalization, means to create a model that inherits all the characteristics of the original model but can have additional characteristics. The applicability of this operation to create public processes is discussed below.
- Analogy: A resulting model is called analogue if, in respect to selected criterions, it is similar to the original model.[250] This definition leaves room for interpretation; interpreted as creating a process model that comprises the same process structure and elements as the original model, the construction of an analogous model is not useful for the development of collaborative views. However, the creation of *complementary* process models can be useful: Thus, the creation of a process model complementary to a given public process can be used in the public process of a collaboration partner and thus to the forming of global processes.

Scope of the Operations Since the operation type implies a certain relationship between the original and resulting model, it should be distinguished if an operation is applied to the overall model, to a group of model elements or to an individual model element. For example, a sub-process A_1 of the process A can be a specialization of the sub-process B_1 of the process B. However, this does not imply that A is a specialization of B (since the other parts of A might be completely different from B). On a higher level of granularity, it can also be distinguished, whether an operation is applied to a group of model elements, or to the *attributes of a model element*. An example for an alteration of model element attributes would be that an event $E_{1,}$ contained in the source model, is called G_1 in the target model.[251] While the alteration of process model elements results in changed process logic, the alteration of model element attributes does not change it.

These three cases are visible in Figure 12: On the left-hand side of the graphic, a process model is abstracted and the target model represents an abstraction of the source model. On the right-hand side, groups of model elements are altered, e.g. F1 and F2 are aggregated into F6. The change of the name F5 to F5' represents an alteration of an individual model element, which – as opposed to the previous operations – does not infer a change of the process logic.

---

[250] Compare VOM BROCKE (2008).

[251] In a similar vein, WERTH (2006), p. 142, distinguishes two ways to alter process models: alienation of the process logic and functional alienation, where the latter only alters characteristics of individual functions.

2.2.4 Operations Between Private and Public Processes

2.2.4.1 Derivation of Public from Private Processes

Public processes are created to inform collaboration partners about those parts of internal processes, which are relevant for the collaboration, describing both the process logic and process elements. The public process must describe to direct interaction partners at least in which sequence what type of messages are exchanged. However, as mentioned before, also non-interaction elements might be described in the public process, to provide the collaboration partner with a better understanding of the collaboration. Thus, to derive a public process from a private process, elements that are not relevant for the collaboration partner are removed from the private process.

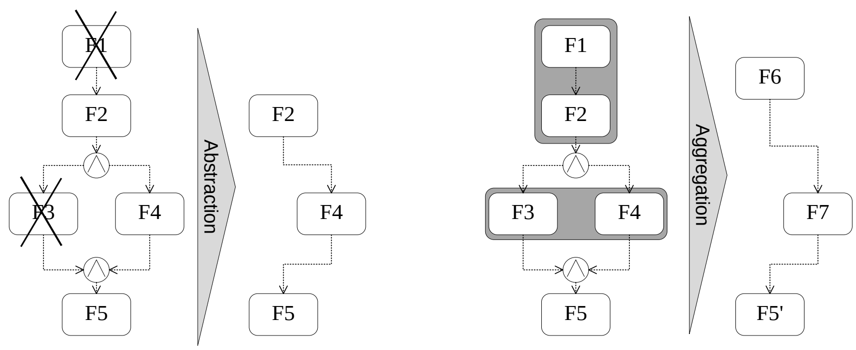


Figure 12: Abstraction and aggregation of process elements

Abstraction of a process model means to leave out some of its elements. As a result, the abstracted process model does not contain all the information comprised in the original model. Other authors argue that an abstraction does not consist of leaving out model characteristics, but in making them variable.[252] We do not follow this conception; certainly, one process model can be abstracted in various ways; for example, the function “Check Credit of Problem Customer using third party Agency” used in a private process, could be published under different names to two different partners (e.g. “Check Credit” and “Check Credit of Problem Customer”). However, the creation of different variants belonging to one original is just one possible outcome of the abstraction process and not the distinguishing characteristic of it. Note, that any element in the private process can be abstracted in order to derive a public process, no matter if it is a control flow element, function or other. However, it has to be ensured that the public process still possesses valid process logic. For example, the abstraction of F3 in Figure 12 results in the redundancy of the AND operators in the public process, thus they also have to be abstracted.

Aggregation In case of model element aggregation, a group of source model elements is subsumed in one new element of the target model. Since not all characteristics of the source model elements are contained in the target model, an aggregation could be seen as a special case of an abstraction. However, going beyond a mere abstraction, here new model elements that are not part of the source model, are added to the target model. For example, on the right-hand side of Figure

[252] Compare KLAUS (1963). For a discussion of further understandings of (model) abstraction, refer to WERTH (2006), p. 139.

12, the functions F1 and F2 are aggregated into the function F6, while F3 and F4 together with the related AND operators are aggregated into the function F7.

On the Possibility to Implement the Derived Processes The main objective of developing private, public and global process models is the enactment of collaborative business processes. It should be clear that the way in which private and public processes are derived and related to each other, affects the possibility to implement them.[253]

Collaborative Business processes for example, which are based on public processes derived only by *abstracting* elements of the private process, are comparably easy to implement. The advantage of this approach is that the public functions described to the partner are equally comprised in the private process. Thus, the partner can directly invoke the functions described to him, and no run time proxy is needed to intermediate between messages from partners and private processes (for example, a collaboration partner could directly invoke the function F4 in the public process illustrated in Figure 12 since exactly this function is implemented in the private process). In case a public process contains elements that represent *aggregations* of private process elements on the other hand, a collaboration partner could not directly access function F4; instead, he would access the function F7, which is not part of the private process. Therefore, in case a public process comprises aggregations of private process elements, a proxy is needed to forward messages directed to an aggregated function to the different functions comprised in it.[254]

From Public to Private Process Models As mentioned above, the operations to develop private from public processes are inverse to those operations needed to come from private to public processes. Since a public process represents an abstraction of a private process, its private counterpart can be created by complementing the public process with private elements. On a technical level, the "completion" of a public process to form an executable process is described, for example, in the specification of BPEL.[255] On a formal level, this enrichment of private processes was recently described by VAN DER AALST ET AL.[256]

On the Generalization of Private Process Elements Besides abstraction and aggregation, WERTH counts generalization as a third operation to achieving information hiding in business processes and applies generalization – similarly as abstraction and aggregation – to *individual process elements*.[257] However, the outcome of deriving a public from a private process *can* be the generalization of the private process, but the operation applied to derive the public process is still abstraction. Only on the *process model* level, can it be useful to identify generalization relationships, e.g. when it becomes clear that various private processes might use the same public process as an interface. In other words, abstraction and aggregation are operations applicable to individual

[253] In Chapter 4, individual approaches to enacting collaborative business processes based on collaborative views are discussed in depth; see pp. 119.

[254] For example, the public visible function "book hotel" with the input parameters "credit card number" and "date of reservation" could internally be split up into the functions "withdraw fee from credit card" and "update room availability database".

[255] Compare OASIS (2007), pp. 152.

[256] Compare VAN DER AALST ET AL. (2008).

[257] Compare WERTH (2006), pp. 136.

elements of a process, while generalization and specialization should be seen as relationships between processes; the latter case is discussed in the following section.

2.2.4.2 Public Processes as Generalizations of Private Processes

In the context of software engineering, generalization means that a specialized class (sub-class, derived class) has the characteristics, the behavior and the associations of one or more general classes (super class, base class). A sub-class is completely consistent with its base class but normally has additional attributes. An object of the sub-class can be used everywhere where an object of the base class can be used.[258] The concept of generalization is well-known and defined for static structures like classes. However, processes are harder to compare than classes,[259] and no established definition for generalization of processes exists. The difficulty in this comparison lies in the fact that not only individual elements, but also a large number of possible process sequences have to be compared. Correspondingly, only partial solutions for process generalization exist and only few modeling languages support concepts for generalization – and those are regarded as being “not very convincing”[260].

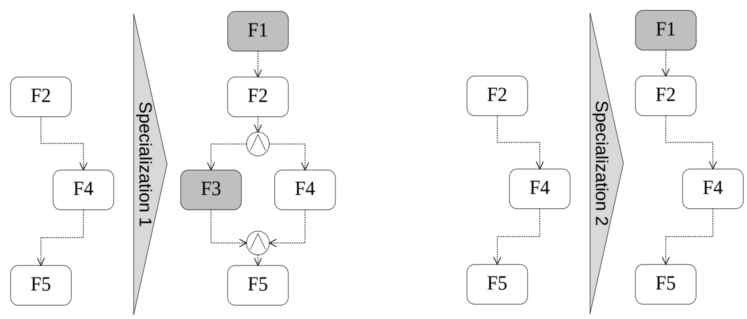


Figure 13: Two private process models as specializations of one public process model

Nevertheless, as Figure 13 illustrates, a public process can be understood as the generalization of a private process:[261] A public process describes a fixed set of characteristics that any private process (which uses the public process as a stub) must offer, and each of the private processes can possess additional, differentiating characteristics. For example, a public process could describe the process “Request for Quote (RfQ) is sent to manufacturer”. Then both the private process of a car retailer and the private process of a bike retailer could implement this public process, and for each the relationship “is a” “process that sends RfQ to manufacturer” would apply. Since a generalization always implies an abstraction, a generalization can be seen as a special case of an abstraction. As the left-hand side of Figure 13 shows, the first specialization of the private process is the result of the abstraction illustrated in Figure 12. Naturally, it only makes sense to speak of generalization if the base class contains more (or is expected to contain more) than one sub-class. In the derivation of public from private processes, this is not necessarily the case since a major objective of this deriva-

---

[258] Compare BALZERT (2000), p. 200.
[259] Compare BASTEN & VAN DER AALST (2001).
[260] FRANK & VAN LAAK (2003), p. 28. Compare also VAN DER AALST (2002), p. 15.
[261] VAN DER AALST (2002 B2B) also describes private processes as specializations of public processes.

tion is the disclosure of internal information and not always the increase of reusability through generalization.

2.2.5 Operations Between Public and Global Processes

In an inside-out development,[262] the public process models created by each collaboration partner are synchronized and integrated into one global process model. In this vein, in the following section the integration of public process models into global process models is discussed. Here, it is presumed that the public process models are complementary to each other, i.e. steps possibly necessary to synchronize them with each other were already executed.

In an outside-in development, the inverse operations of an inside-out development are used: global processes are developed first; afterwards the public processes comprised in these models are disaggregated and distributed to the individual collaboration partners. Since the aim of a public process is to represent the externally visible activities of one organization, the criterion to distribute functions of a global process to public processes is their organizational affiliation.[263] For a more detailed description of the disaggregation of global into public processes, refer for example to ZAHA ET AL., who describe a formal algorithm for deriving public process models from global process models.[264]

2.2.5.1 Message-Based vs. Control Flow-Based Integration of Public Processes

The connection of distributed models to form one valid model is an often needed functionality in the context of collaborative business. Global process models, for example, can be developed by the (connecting) integration of various complementary public process models.

To form a global process based on public processes both the comprised functions – for example, send and receive activities – and the logic of the public process have to be complementary. For example, if organization A can either send the message C *or* D, organization B must equally be able to receive either C *or* D but should not expect to receive both C *and* D. As illustrated in Figure 14, two different ways for integrating such complementary process parts into one global process can be distinguished:

- Connection via message flow: Figure 14 displays a global process in the form of two interacting public processes. On the left-hand side, the public processes comprised in the global processes are connected via *message flows* (displayed as solid arrows), while the control flow (dashed arrows) is modeled only intra-organizationally. In other words, the control flow is modeled explicitly only inside public processes, while the control flow between the public processes is modeled only implicitly, via message exchanges. This form of connecting integration represents a comparatively loose coupling, and re-

[262] The individual steps and possible transformation paths of inside-out and outside-in procedure models in the enactment of collaborative business processes are discussed in Chapter 5 (pp. 181).

[263] For example, each function of a global process that is executed by organization A should be allocated in the public process of organization A (compare also Figure 10); if more than one collaboration partner is designated to a function, the function must be represented in a finer granularity to be split up among the organizations.

[264] Compare ZAHA ET AL. (2006). Also tackling the execution level, MENDLING & HAFNER (2005) describe how a global process annotated in WS-CDL can be transformed into a public process annotated in BPEL.

sembles the interface-based representation of global processes as described above (p. 43). Since the information about the correlation of the different process parts is not comprised in a central global model, it must be contained in the various public processes. This approach is realized for example in BPEL, where a number of correlated public processes can be implemented without the need for an explicit global model.

- Connection via control flow: On the right-hand side, the inter-organizational control flow is modeled explicitly. This connection form results in a tighter coupling, where the public processes are merged into one global process model. It resembles the interaction-based representation of global processes described above. Theoretically, it can be used to realize either a merging or a connecting integration, though the tight character of the integration suggests a merging integration. This approach is implemented for example by the Web Service Choreography Description Language (WS-CDL),[265] where all elements of the CBP – including information on the correlation of the public processes – are stored in one global model.

Corresponding to these two connection types, two different ways to develop global models exist: First, the integration of different public process models into one explicit global model, second, the correlation of different public process models with each other in order to form a CBP.

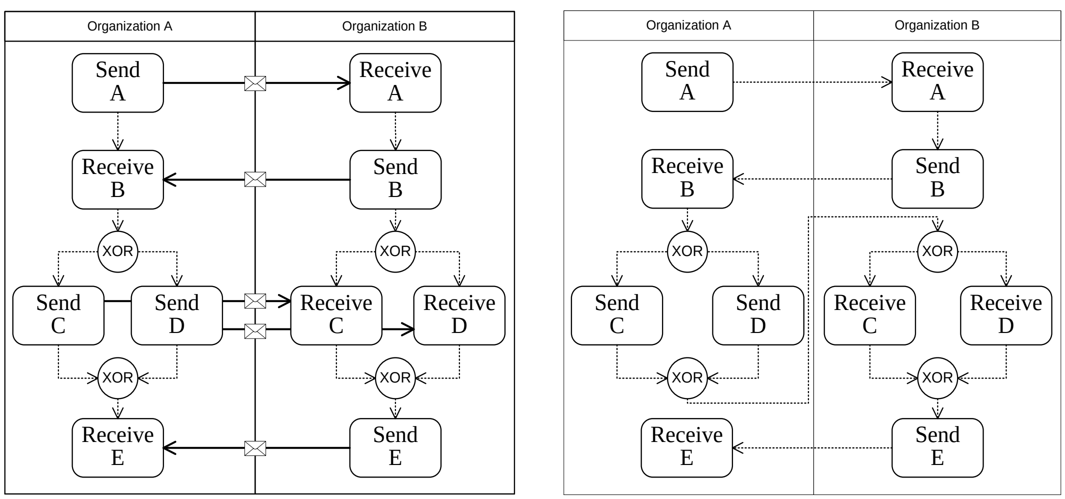


Figure 14: Message-based vs. control flow-based integration of public processes

In this context, WERTH points out that if conceptual modeling languages like the EPC are used to represent the control flow implied by message exchanges in CBP, ambiguities could occur. More specifically, he writes that if a flow of messages or goods is transformed into control flow, the time/logical relation among the interacting functions (consisting of emitting and receiving functions) must be described explicitly. For example, it should be clear, if the emitting function must terminate before the receiving function can be initiated. In consequence, he proposes to divide an

[265] Compare W3C (2006).

interaction function in a pre-, a main- and a post-phase.[266] However, this extended annotation is only necessary if the granularity of the original model capturing the process is too imprecise to capture the interaction logic. Therefore, as an alternative to using an extensional annotation, the level of detail of the process model can be augmented.[267] Nevertheless, the model must specify exactly, which emitting function is correlated with which receiving function. On the left-hand side of Figure 14, a corresponding global process is shown that consists of two public processes related via message flows where each sending function is matched to the corresponding receiving function.[268]

2.2.5.2 On the Advantages of Each Approach

WERTH also distinguishes between connecting and merging integration of process models in the context of collaborations, and proposes to use both integration types making use of a control flow-based integration.[269] Since model-checking techniques normally rely on control flow and not on message flow, it can be argued that the public process in global models should be connected via a control flow.[270]

However, in case an explicit control flow is needed for model checking purposes, the transformation of message flow into control flow is simple: In a valid (message-based) global process, a clear predecessor-successor relation exists between activity couples that realize cross-organizational transactions. More specifically, a *send* must precede a correlated *receive* activity; and the control flow follows these interaction pairs. Thus, a simple algorithm can be used to transform a message-based into a control flow-based global process:

1. If the process starts with a receive activity, add a dummy activity that precedes the receive activity. If the process ends with a send activity, add a dummy activity that succeeds the Send activity.[271]
2. After each send activity, add an AND-split in such a way that a new control flow edge goes to the corresponding receive activity of the adjacent public process; the other edge leaving the AND-split corresponds to the previously existing control flow.
3. Before each receive Activity, add an AND-join, which joins the previously existing control flow with the control flow edge coming from the correlated send activity.

---

[266] Compare WERTH (2006), pp. 157.

[267] Which is also indicated by WERTH, since the semantic of “pre-phase”, “main-phase” and “post-phase” is defined by a more fine-grained EPC representing one (coarse-grained) interaction activity. Compare WERTH (2006), p. 164.

[268] In the EPC, this would be displayed via pairs of events, for example, “RfQ was sent” and “RfQ is received”. In BPEL, such an interaction would be represented as a pair of functions (“send RfQ” and “receive RfQ”); or, alternatively, the receive activity could be represented as an event; compare OASIS (2007).

[269] Compare WERTH (2006), pp. 246.

[270] Compare FU, BULTAN & JIANWEN (2003).

[271] The syntax used corresponds to a simplified variant of Event-driven Process Chains, focusing on interaction activities and omitting events that are implied by the functions (e.g. “message was received”, or “message was sent”). In the traditional EPC as defined by KELLER, NÜTTGENS & SCHEER (1992) each process starts and ends with an event. Thus, applied to the EPC, no dummy elements at the beginning and end of the process would be needed, and Step 1 of the algorithm would be omitted.

The result of applying this algorithm to the global process illustrated on the left in Figure 14 is illustrated in Figure 15, where the new AND-operators are illustrated in gray and the additional control flow edges in bold.

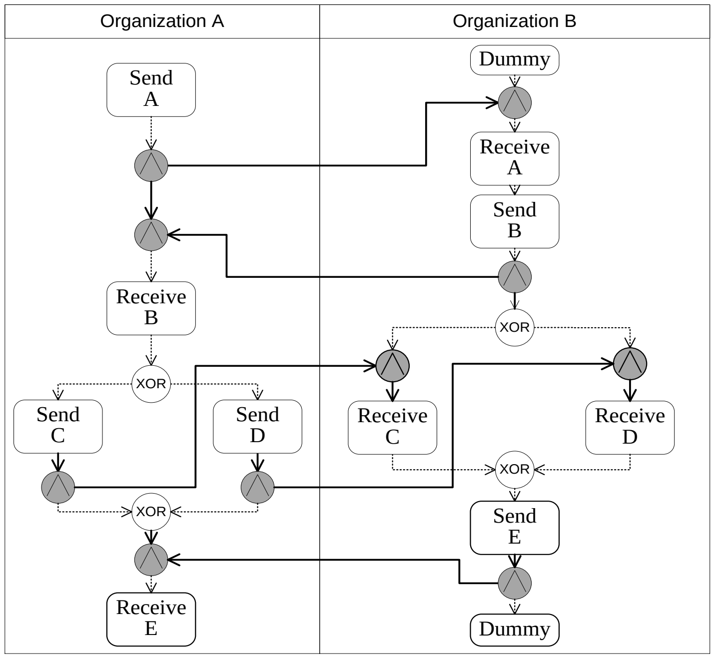


Figure 15: Message flow converted into control flow of a global process

A connecting integration via message flows, on the other hand, supports a loose coupling between independent, decentralized models as needed in collaborations.[272] It enables a decentral execution via service choreographies, where public processes are correlated via explicit transactions, e.g. exchanges of digital messages or material, and not by implicit information exchange contained in control flow. The suitability of each approach for the AIOS is further discussed in Chapter 4 (pp. 129).

2.2.5.3 Message-Based Integration on the Instance Level

The description above focused on the integration of public process models that display information on the generalized model level. To support the execution of processes, global process models also need to contain information on how the comprised public process are to be related during run time: Not only the process models, but also the instances of the public process need to be integrated. For example, if organization A sends a message to organization B, it is not sufficient to direct this message to a specific process of B, since B could have multiple instances of the same process (compare Figure 16). Thus, A also needs to specify the process instance the message is directed to. For this aim, the specification of the static interface – e.g. the receiving business function of a collaboration partner, or, on a technical level, the WSDL port of the recipient service – is not sufficient.

[272] Compare also Figure 10; the integrating connection would result in a process model, which would optimize the overall model; however, the original public process can hardly be identified anymore, inhibiting a distribution of the global process to the collaboration partners.

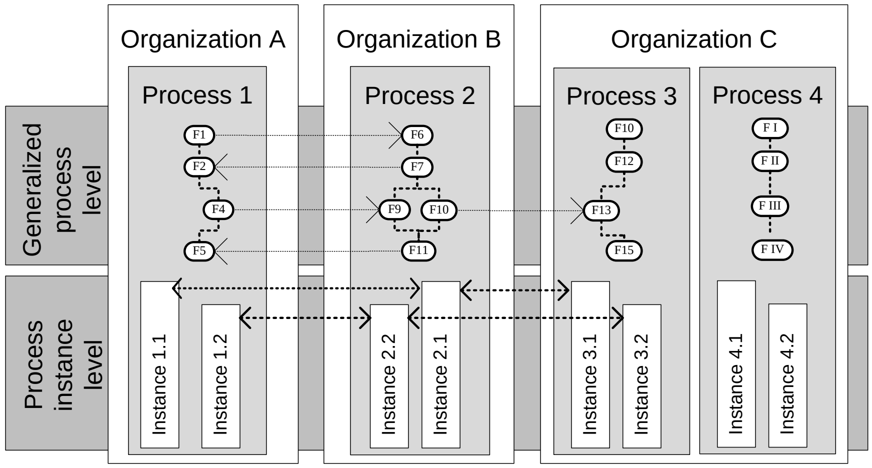


Figure 16: Correlation of public process models and instances

To relate messages to a CBP instance, the concept of *message correlation* is used. According to the World Wide Web Consortium (W3C), “Message correlation is a means of associating a message within a specific conversational context”[273]. One way of realizing message correlation, is to provide each message that is part of a certain CBP instance with a set of properties unique for that instance. Thus, all messages with the same tokens are identifiable as belonging to the same group. Note, that if this concept is applied in the context of CBP, a message is automatically correlated with three process instances: the process instance emitting the message, the receiving process instance, and in consequence, the collaborative business process in which both parties are engaged. This way of message correlation is implemented by BPEL,[274] where messages are associated with correlation tokens derived from the process.[275] Naturally, a corresponding correlation mechanism could be implemented by the middleware executing the CBP. For example, the engines of the different organizations could agree on an identifying token that should be attached to each message that is part of a certain CBP instance. However, this would require a harmonization on the middleware level, which contradicts the loose coupling paradigm. Instead of creating implementation dependencies, in the context of SOA this correlation should be based on business data independent of the execution environment.[276] Note, that the concept of routing is related to correlation, but only describes the way from a sender to a target process, leaving out the way from the engine (that executes the target process) to the process instance. Or, in other words, finding the way to the engine that implements a process is part of the routing, finding the process instance running on that engine is achieved via message correlation.

BARROS ET AL. describe two variations of the principle to correlate messages using identical tokens: Correlation based on the comparison of tokens and time-interval-based correlation. In the first case, logical operators like “greater” or “not” are applied to the tokens in order to evaluate if a mes-

---

[273] W3C (2004).
[274] Compare ANDREWS ET AL. (2003). Here we refer to BPEL 1.1., however, a similar correlation mechanism is implemented in BPEL 2.0, compare also BARROS ET AL. (2007).
[275] In a purchasing process for example, the elements “Name of the Buyer”, “Date of Transaction” and “Purchased Object” could constitute a set that correlates each message comprised in it to an instance of the “Purchasing Process”.
[276] Compare ANDREWS ET AL. (2003), p. 45.

sage belongs to a certain process instance. In the second case, the correlation token is extended with a time stamp. Additionally they propose a chained correlation mechanism, where, instead of one unique correlation token, a chain of correlation tokens is used throughout a CBP instance.[277] However, since the benefits of these extensions do not seem to rectify the added complexity implied by them, in the following a message correlation based on identical correlation tokens is presumed.

2.3 Enacting Collaborative Business Processes

In the previous sections of this chapter, it was described how collaborative business processes are defined and modeled. Since the AIOS should also tackle the execution level, in the following, essential concepts are described to transfer business level models to the execution level. These concepts focus on SOA, since SOA can be seen as the state-of-the-art technology for enacting loosely coupled, cross-organizational business processes. Therefore, traditional systems used in collaborative business, like for example EDI, are not tackled in this section; however, they are discussed in Chapter 3, pp. 82. Representing a bridge between business-level models and execution concepts, in the following section the model-driven development of information systems is described.

2.3.1 Model-Driven Development

To support a methodical software development, a number of initiatives propagate the model-based generation of software, where execution level models are derived from conceptual models. In software engineering – e.g. rather technology than business-oriented approaches – such methods are known as model-driven engineering or model-driven development (MDD).[278]

Modern software development is confronted with a number of challenges, including an increased software complexity, higher requirements on compliance, performance, reliability and quality, shorter lifecycles for software, the frequent need for adaptations of processes and an increased need for cost efficiency due to global competition.[279] The aim of MDD is a systematic software development that tackles these problems. To reduce software complexity, abstract, coarse-grained models are used to represent complex, fine-grained models. To support a precise correlation between these models, they are defined via formal metamodels, which enables the automatic transformation of models and reduces the risk of errors from manual model transformations. Apart from efficiency, MDD also aims at the effectiveness of software development, e.g. to ensure that organizational requirements are met by software development (as explained below, this second objective is also pursued by Business-IT alignment). Interestingly, and though model-driven development is not a new concept and is supported by many tools and concepts, the maturity of this approach is still under

[277] Compare Barros et al. (2007), pp. 250.
[278] Compare Schmidt (2006) and Hailpern & Tarr (2006).
[279] Compare Gruhn, Pieper & Röttgers (2006), pp. 9. According to Weigert et al. (2007), p. 36, these challenges are aggravated by problems in the practice of software development: the informality and imprecision of notations used in software development leads to misunderstandings between developers, the manual transformation of design model to code is error-prone and, if code is changed, it is often not synchronized with design models.

discussion.[280] However, for the description and enactment of collaborative business processes, the following characteristics of MDD are useful:

- Description of systems on different vertical levels: To enable interoperability on both the business and the execution level, the corresponding MDD levels can be used.
- Comprehensiveness and precision: By providing both conceptual and technical models for a system, a comprehensive description is delivered that is understandable to different stakeholders. This multi-perspectivity also supports precision, since elements of the system are described from different sides and in different granularities.
- Compliancy: A model-driven development ensures that the implementation of a (cross-organizational) IT system follows the goals specified on the business-level.
- Evaluation possibilities: If execution level elements are correlated with conceptual elements, technical events can be evaluated and illustrated on the business level.

Model-Driven Architecture In 2001 the OMG created the probably best known framework for model-driven engineering, the Model-driven Architecture.[281] The objective of MDA is to improve software development. In particular, the speed of software development is to be increased by using automated transformations between models, for example, by using generators that transform models into code. Another objective is to reduce the complexity of software development through abstraction layers, e.g. to use coarse-grained, conceptual models for structuring and fine-grained models for adding technical details. The MDA stems from the software engineering area; compared, for example, with approaches from the area of enterprise modeling,[282] it is a rather technical approach that offers only limited possibilities for business requirements modeling. Thus, business-IT alignment is not a primary objective of MDA; instead, MDA focuses on the efficiency of software development.[283] In MDA, three vertical modeling levels are distinguished:[284]

- Computational Independent Model (CIM): This model type describes the business logic and the domain model from a computational independent viewpoint. The primary users of CIM are domain practitioners, who describe requirements in the model that are to be realized by the experts for designing and constructing the envisioned artifacts.
- Platform Independent Model (PIM): The platform independent model formally describes the structure and functionality of a system.[285] It is defined at a high level of abstraction and is independent of any implementation technology.
- Platform Specific Model (PSM): The platform specific model enriches the PIM with detail needed by a specific platform in order to execute it. Thus, the PSM is tailored to specify a system in terms of the implementation constructs available in one specific implementation technology, like for example web services.

---

280 SCHMIDT (2006), p. 30, for example states, “although a great deal of publicity on model-driven topics has appeared in the trade press, it’s surprisingly hard to find solid technical material on Model-driven Engineering technologies”.
281 For details, see OMG (2008) or GRUHN, PIEPER & RÖTTGERS (2006).
282 Compare for example ZIEMANN & MENDLING (2005).
283 Compare also GRUHN, PIEPER & RÖTTGERS (2006), p. 43.
284 Compare OMG (2003).
285 Compare also GRUHN, PIEPER & RÖTTGERS (2006), p. 27.

Related Approaches The alignment of IT to the needs of an organization is a major objective of information systems research. Thus, it is not surprising that this goal is tackled by various, often-unclearly defined and overlapping approaches. Under the term *IT governance*, for example, principles and procedures are captured, which ensure that IT fulfills the business objectives that resources are used responsible and that risks are monitored.[286] While IT governance is a part of corporate governance, *convergent engineering* stems from the area of software engineering where it is understood as the holistic treatment of business and software systems.[287] More specifically, it propagates that the structure of IT systems should follow the structure of business processes,[288] and that each business object should be represented by one corresponding software component.[289] The term *business-IT alignment* is described as the connection of business requirements with corresponding IT-based solutions.[290] To narrow the gap between organization and technology is also a major objective of *business process management*, which includes "concepts, methods and techniques to support the design, administration, configuration, enactment, and analysis of business processes"[291]. *Business process automation* focuses on the enactment of business processes, and thus can be seen as a part of business process management.[292]

Model Transformations Since the main idea of model-driven software development is to transform conceptual models into fine-grained, executable models, model transformations are an essential element of model-driven development.[293] In general, a model transformation consists of applying transformation rules to a source model in order to create a modified target model. A transformation can tackle different modeling levels: *Transformations on the metamodel-level* only change the syntax of a model; i.e. the content of the models stays the same, while the metamodel in which the model is displayed, changes. For example, a process model could be transformed from an annotation in EPC to an annotation in BPMN. *Transformations on the model level* are directed to specific model elements, altering the content of a model. For example, a private process model could be transformed to a public process model by applying the transformation rule "hide all elements annotated as private in the source model". Both transformations types can also be intertwined, for example, by transforming an EPC private process into a BPMN public process.

Two further types of transformations can be distinguished: *Vertical transformations* take place between models of different levels of technical detail (for example, transformations between EPC and BPMN, or BPMN and BPEL). In cases with *horizontal transformations*, the technical level of detail of the source and target model is the same. This transformation type is especially relevant in the context of interoperability, where the syntax of models has to be harmonized or private model elements have to be disclosed from partners. An example of a *horizontal metamodel transformation*

---

286 Compare RÜTER, SCHRÖDER & GÖLDNER (2006), p. 28.
287 Compare GRUHN, PIEPER & RÖTTGERS (2006), p. 41.
288 Compare TAYLOR (1995).
289 Since services in SOA also should correspond to business functions rather than technical functions, convergent engineering and corresponding architectures (e.g. HUBER, 2008) can be seen as predecessors of SOA.
290 Compare BAUMÖL (2006), p. 314. According to BAUMÖL, in this context the "business perspective" is mainly constituted by business processes and the organization structure. For further discussion of the term IT-alignment, refer to MASAK (2006), p. 11.
291 WESKE (2008), p. 5.
292 In Chapter 3 (pp. 102), various approaches for business process automation are discussed.
293 Compare for example BROWN, CONALLEN & TROPEANO (2005), p. 8.

is the transformation from technically enriched EPC to BPMN models that does not alter the semantics of the model.[294] An example of a *horizontal model transformation* is the transformation of a public process modeled as EPC into an EPC public process model.[295]

2.3.2 Service-Oriented Architecture

Traditional Understanding of SOA In the last decade, SOA has received a great amount of attention both in research and in industry. The term Service-oriented Architecture was coined in the late 1990s from the company Sun to describe Jini, an environment enabling the dynamic discovery and binding of services in a network.[296] The main elements of a SOA are illustrated in Figure 17: A service provider offers a software component that can be found and executed via a network. To enable service consumers to find a service, the provider publishes a description of the service in a directory. Potential service consumers can search through the service descriptions, and if they find a service that matches their requirements, they can bind and execute this service.[297] SOA does not replace Object-oriented, procedural or data-centered approaches but complements them, since services in a SOA are still developed with these traditional approaches. A major difference between SOA and previous approaches is in the coupling mechanism: The coupling in classic approaches relies on the technical descriptions of fine-grained operations, while SOA aims at describing the functionalities of services on a coarse-grained business level.[298]

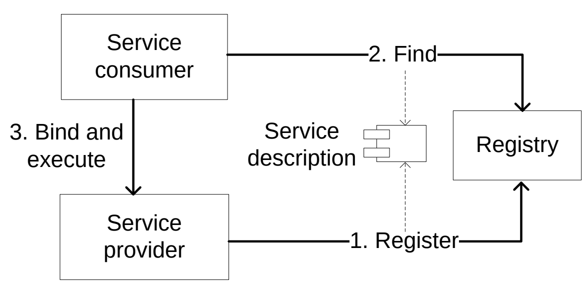


Figure 17: Main elements of a Service-oriented Architecture

Different Understandings of SOA Many different understandings of the term SOA and its characteristics exist. Nonetheless, the basic SOA elements as displayed in Figure 17 imply a number of core SOA characteristics, e.g. loose coupling, comprehensive service descriptions and the possibility to search and bind services via a network. HEUTSCHI evaluated nine different sources regarding their view on SOA design principles, the results are displayed in Table 1. As displayed in the table, he distinguished four main principles of SOA:[299]

---

[294] Another example of a horizontal metamodel transformation in the context of interoperability is provided in ZIEMANN ET AL. (2007), where EPC process models are transformed into IEM process models using POP* as an intermediate format.

[295] As described for example by ZIEMANN, MATHEIS & WERTH (2008).

[296] Compare MCGOVERN ET AL. (2003), p. 36. Jini was supposed to realize “Plug-and-Play” networks, for example, to dynamically add printers to the network and automatically download and activate the corresponding driver.

[297] Compare for example MCGOVERN ET AL. (2003), pp. 35.

[298] Compare for example FERGUSON ET AL. (2003).

[299] Compare HEUTSCHI (2007), pp. 46.

- Interface-orientation: To foster the use of services in different contexts only the interfaces of services are described, but not their implementation. This interface description resembles a contract, describing service characteristics on both a technical and a conceptual level.
- Interoperability: A service provider and service consumer have to understand each other; most importantly, they have to be able to describe and interpret service descriptions correctly. Thus, both the technical and the business level aspects of a service have to be described in a format understandable by potential service consumers.
- Autonomy and modularity: Modularity is a traditional software engineering concept, requesting a high internal cohesion of modules and a low inter-module dependency.
- Business-orientation: The functionality of a service should follow business objects or activities of business processes. Thus, services should represent coarse-grained, well-defined business functions.[300]

As the table also shows, HEUTSCHI further refined these categories into sub-categories. The sub-category with the highest level of agreement illustrates the fact that SOA abstracts from service implementations; which is not surprising, since a white-box approach would question the overall SOA idea of describing and invoking services based on comprehensive service descriptions. Accordingly, the category “comprehensive, unified service specification” was also mentioned seven times. The only other category that found similar acceptance was “technical interface description”, which reconfirms the interface focus as the most important common denominator in the understanding of SOA characteristics. After these categories, only “loosely coupled communication” and a business-oriented service granularity were referred to more than four times.

Interestingly, these characteristics do not directly display the (ambitious) goal of the traditional SOA to enable the run time discovery of business functionalities. While MCGOVERN ET AL. in the early days of SOA explicitly described this discovery and dynamic binding of services as a main characteristic of SOA,[301] today this notion of SOA seems to be more relaxed. This can be seen as a tribute to reality, since to our knowledge the automated discovery of services in a directory based on functional descriptions remains a research topic. Note though, that the current notion of SOA supports such an automated discovery, for example, through principles like requiring a comprehensive interface description.

Another reason for the different conceptions of SOA lies in its popularity and the fact that different research communities can relate to it. In this vein, VÖLTER distinguishes three different conceptions of SOA: A *component-centered view* that interprets SOA as a well-made component architecture, an *Enterprise Application Integration*-based viewpoint, focusing on the realization of asynchronous communication, and a *business process management-based* SOA concept, which focuses on the transformation of conceptual business processes and their realization with SOA. As he also concludes,[302] these three viewpoints are not contradictory but can be combined well. However, there

---

[300] HEUTSCHI (2007), p. 31, uses the term “Bedarfsorientierung”, which also translates to “requirements-orientation”; however, in the explanation of this category he only refers to business requirements.
[301] Compare MCGOVERN ET AL. (2003), pp. 40.
[302] Compare VÖLTER (2007), pp. 423.

is an agreement that SOA services in general should rather be coarse-grained and business-oriented than fine-grained and technical; and since SOA, on the other hand, is a technical means to execute processes, SOA represents a bridge between business-oriented and technique-oriented approaches. Accordingly, this work focuses on the third viewpoint, seeing SOA as a means to realize flexible, collaborative business processes.

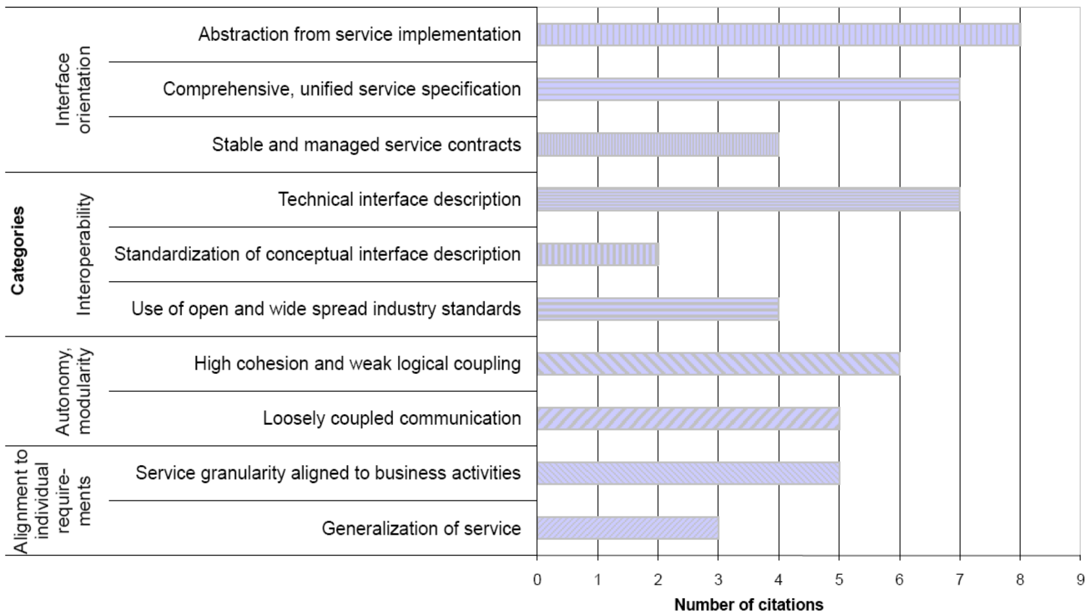


Table 1: SOA characteristics referenced in literature[303]

Components as Predecessors of Services Similar to the various interpretations of SOA, different conceptions of services in the context of SOA exist. However, services are successors of software components, which have since long been discussed in software engineering. BALZERT defines a component as a "closed software building block offering an application-oriented, semantically coherent functionality, which can be accessed via an interface"[304]. The objective of component-based software engineering is to easily re-assemble existing components into new programs. Therefore, the components should be language and platform independent as well as being accessible via networks. Furthermore, it would be beneficial if these components could be assembled using visual and generative programming environments.[305] ANDRESEN highlights the fact that each component inherently possesses knowledge representing a certain business area, which it can offer to other components present in an enterprise.[306]

Web Services Different from traditional components, web services should support SOA characteristics; for example, they should represent coarse-grained (business) functionalities, support an

303 This chart is based on data described by HEUTSCHI (2007), pp. 46.

304 BALZERT (2000), p. 856. This citation is translated from German ("abgeschlossenen, binären Software-Baustein, der eine anwendungsorientierte, semantisch zusammengehörende Funktionalität besitzt, die nach außen über Schnittstellen zur Verfügung gestellt wird").

305 Compare BALZERT (2000), p. 857.

306 Compare ANDRESEN (2003), pp. 19.

easy discovery and binding within a network. In this vein, O'SULLIVAN, EDMOND & TER HOFSTEDE name four activities that should be supported by a service description: discovery, substitution, composition and management of services.[307] However, corresponding to the variety of SOA definitions, in literature many different service conceptions from technical and business viewpoints exist. An example for a technical conception is: "A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL)"[308]. The definition of the UDDI consortium refers to business applications and focuses on the interface aspect, describing web services as "self-contained, modular business applications that have open, internet-oriented, standard-based interfaces"[309]. From a business perspective, a service is defined as a collection of tasks to fulfill certain business goals.[310] In the following, a service is understood as follows:

> **Definition 9** *A* service *is a software component that implements a business function, is described via a comprehensive, machine-readable interface and can be found and invoked via a network.*

Figure 18 illustrates the relation of services to business processes and to the execution environment: services implement and are described by business functions, and a business function is part of a business process. A service can also be part of a service composition, which represents the technical counterpart of a business process. Service compositions can be executed via process engines, e.g. the service composition description controls how the process engines acts. In the course of executing a service composition, the process engine invokes the service comprised in the service composition. The service itself can be realized by various software applications.

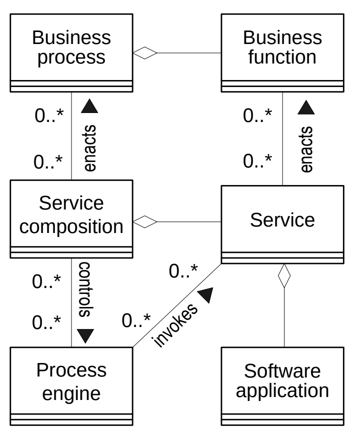


Figure 18: Relations of business processes, services and applications

[307] Compare O'SULLIVAN, EDMOND & TER HOFSTEDE (2002), p. 118.
[308] W3C (2004).
[309] UDDI CONSORTIUM (2001), p. 1. UDDI stands for Universal Description, Discovery and Integration.
[310] Compare JONES (2005), p. 89.

2.3.3 Service-Based Workflows

In the context of SOA, processes are automated as service flows. Since the distinction between private, public, and global processes meets the requirements of SOA for describing external views on services, corresponding forms of service-based flows were implemented: Service choreography and service orchestrations. More specifically, similar to private, global and public processes, the concepts of orchestration, choreography and choreography interface are used to display processes on the execution level.[311] Note though, that private, public and global processes refer to organization elements and their definition requires domain experts that specify how organizations should interact. Orchestration and choreography on the other hand are usually seen as means to implement business processes in a central or decentral way, e.g. technical concepts.[312]

2.3.3.1 Orchestration

The terms orchestration and choreography are sometimes used synonymously or at least in a hardly distinguishable way, like for example in the definition of ALONSO ET AL.: "*Orchestration deals with how different services are composed into a coherent whole. In particular, it specifies the order in which services are invoked*"[313] – which also applies to choreographies; both concepts describe how services are composed into a process and hence specify the order in which those are invoked. The following characteristics constitute orchestration:

Central Execution Most authors define orchestration as the execution of one process via one central (workflow) engine.[314] Orchestrations are also called "executable processes"[315] since they are intended to be executed by a workflow engine.[316] We follow this conception: Similar to an orchestra where each member is directed only by a conductor, one central party interacts with all participants of an orchestration, but the parties do not interact directly with each other.[317] Referring to the same principle, HINTERHUBER describes a central form of coordination performed by orchestrator firms.[318] In the same context, SCHULZ & ORLOWSKA distinguish between mediated and unmediated communication, where a central "mediator" orchestrates the communication between workflow participants.[319] Referring to the same concept, MILES & SNOW use the term "broker".[320] SCHULZ & ORLOWSKA distinguish two types of orchestra-

---

311 Compare BARROS, DUMAS & OAKS (2006).

312 MASAK (2007), p. 104, even states that the service composition form is nearly always chosen without taking into consideration organizational issues.

313 ALONSO ET AL. (2004), p. 257.

314 Thus MASAK (2007), p. 107, states that an Orchestration describes the executable aspects of one service from the viewpoint of an orchestrator. Similar definitions can be found in PELTZ (2003) and BURGHARDT ET AL. (2003).

315 Compare OASIS (2007).

316 Compare BARROS, DUMAS & OAKS (2006), p. 66. KRALLMANN ET AL. (2007), p. 16, also distinguish between orchestration and choreography based on the criterion of centrality/decentrality: a service composition is called orchestration if invocations are centrally coordinated and choreography, if the coordination is decentral.

317 This means for example, that if party A is the orchestrating party who runs the workflow engine, party B and C do not interact directly inside of this orchestration. C can only know of activities from B if A transfers such information.

318 HINTERHUBER (2002).

319 SCHULZ & ORLOWSKA (2004), p. 16. More specifically, they state, "In a mediated environment, there is one central participant that is able to route information to the communication partners, which do not have to know each other. Prior knowledge about the mediator is sufficient. All or some communication is routed through the mediator".

320 MILES & SNOW (1986).

tion: stateless and stateful.[321] In the former case, the orchestrator only passes the messages between the participants of the orchestration, in the latter case, the interaction is logged and the orchestration process may contain activities going beyond message passing. A recent definition from OASIS confirms the focus on a central execution, defining orchestration as "a technique used to compose hierarchical and self-contained Service-oriented business processes that are executed and coordinated by a single agent acting in a 'conductor' role"[322].

Technical Focus The terms orchestration and choreography stem from a technical background: They do not primarily describe business processes between organizations, but originally describe compositions of coarse-grained software components which are invoked by one or more process engines. Accordingly, MATHES states that a service orchestration represents the description of a technical process.[323] In addition, KRALLMANN ET AL. refer to orchestrations and choreographies as "technically executable service combinations".[324]

Rather Internal than Cross-Organizational LEYMANN ET AL. use the term private flow for orchestrations, indicating that orchestrations are usually executed within the boundaries of one organization.[325] A close relation between private process and orchestration is also indicated by WESKE ("a private business process realizes a process orchestration")[326] and by MASAK ("orchestration is normally used for private business processes")[327]. However, orchestrations could also be used in cross-organizational scenarios. Thus, RODON, BUSQUETS & CHRISTIAANSE propose to use orchestrations in business networks, while HINTERHUBER describes orchestrations in value chain scenarios.[328]

Building on the definition of business processes given above, orchestration can be defined as follows:

Definition 10 *An* orchestration *is a business process intended for execution by one process engine. The functions comprised in the business process are realized as services invoked by that engine.*

2.3.3.2 Choreography and Choreography Interface

Similar to orchestrations, choreographies originally describe service compositions and thus are of a technical nature.[329] As Figure 19 illustrates, a business process can be represented either as an orchestration or as a choreography. The difference between them is that in the case of orchestration

321 SCHULZ & ORLOWSKA (2004), p. 125.
322 OASIS (2008), p. 69. OASIS stands for "Organization for the Advancement of Structured Information Standards".
323 Compare MATHAS (2008), p. 211.
324 KRALLMANN ET AL. (2007), p. 16.
325 Compare LEYMANN ET AL. (2002), p. 207.
326 WESKE (2007), p. 221.
327 MASAK (2007), p. 107.
328 Compare RODON, BUSQUETS & CHRISTIAANSE (2005) and HINTERHUBER (2002).
329 Thus AUSTIN ET AL. (2004) define a choreography description as "a multi-party contract that describes from a global view point the external observable behavior across multiple clients (which are generally web services but not exclusively so) in which external observable behavior is defined as the presence or absence of messages that are exchanged between a web service and its clients".

the process is designed for a central execution, while in case of the choreography, the same process is decomposed into various process fragments to be executed by distributed, interacting process engines. While orchestration is a central approach, choreography is a peer-to-peer approach where no central party executes a process; instead, each actor executes one part of the process. In this vein, WUTKE, MARTIN & LEYMANN distinguish between the enactment of workflows via a central navigator unit and workflows that are decentralized, in which case no central Workflow-Management-System exists.[330] Thus, in the orchestration scenario one process model exists which is executed by a central engine and correspondingly invokes the services participating in the orchestration. In the choreography scenario, no central orchestration engine is used; instead, services interact directly with each other. But to replace the process logic captured in the central model, each service participating in the choreography has to store its process logic in a decentral manner.[331] Or, as MASAK states: a choreography is composed of a number of peer-to-peer interactions of the participating services.[332] OASIS also focuses on the interaction aspect, defining a choreography as "a technique used to characterize and to compose Service-oriented business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal"[333]. Based on these characteristics and the above-described definition of business process,[334] the term choreography can be defined as follows:

Definition 11 *A* choreography *is a business process intended to be executed by interacting engines that are each responsible for the execution of designated parts of the process.*

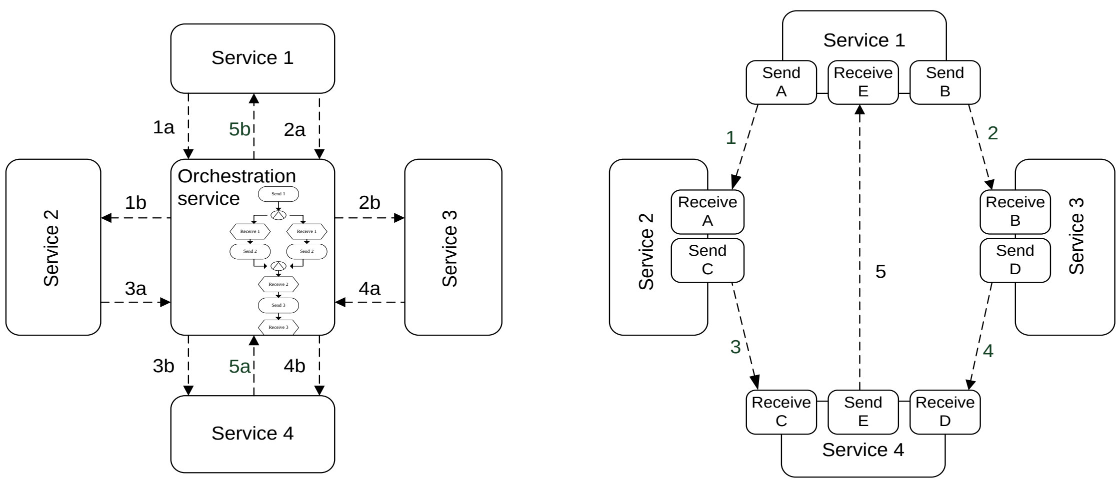


Figure 19: Orchestration vs. choreography

Choreography Interface is the third SOA-related concept that enables views on collaborative business processes. Corresponding to public processes, a choreography interface captures the beha-

---

[330] WUTKE, MARTIN & LEYMANN (2008).

[331] This is illustrated in Figure 19, where services publish the operations they offer. Note though, that choreography interfaces do not only publish the individual operations of services, but also the time/logical relationships between the operations.

[332] MASAK (2007), p. 106.

[333] OASIS (2008), p. 104.

[334] Compare Definition 1, p. 16.

vioral aspects of interactions in which a service can engage in to achieve a specific goal.[335] Since choreographies are the technical counterparts of global processes, they can also be described in two ways: either as one process that plainly describes the sequence between the interaction activities of the partners (interaction-oriented) or by explicitly modeling and interconnecting the choreography interfaces of each participating service (interface-oriented).[336] Based on the definition of a public process, the term choreography interfaces can be defined as follows:

> **Definition 12** *A* choreography interface *is a public process intended to describe the externally visible interaction activities of a service.*

Choreography vs. Conversation Protocols REICH argues that the concepts of choreography and orchestration have been since long known in the more technically oriented branch of computer science under the names of protocol and process.[337] In fact, these terms have been discussed in the context of orchestration and choreography since the upcoming of these service-based process types, e.g.:

- Business Protocol: To describe the choreography interfaces offered by BPEL, LEYMANN & ROLLER state, "a business protocol specifies the potential sequencing of messages exchanged by on particular partner with its other partners to achieve a business goal. That is, a business protocol defines the ordering in which a particular partner sends messages to and expects messages from its partners based on actual business context".[338] This definition obviously is congruent with that of public processes or choreography interfaces as well as the definition of global processes or choreographies.
- Conversation: ALONSO ET AL.[339] use the term conversation protocol for a similar definition, describing a "conversation as sequences of operations (i.e., message exchanges) that could occur between a client and a service as part of the invocation of a web service". Based on this, they define coordination protocol as a specification of a set containing all correct and acknowledged conversations. HOHPE uses the synonymous term *conversation policy*, which represents a "description of all allowed conversations"[340], and relates it to choreographies and orchestrations. Again, both public and global process models can be used to describe allowed conversations between collaborating parties, and thus represent conversation protocols.

As Figure 20 illustrates, WESKE also uses the term conversation to specify choreographies. In concordance with the authors cited above, he states, that one choreography can describe many conversations and that one conversation consists of many interactions. As the metamodel shows, he

---

335 Compare the definition of behavioral interface in BARROS, DUMAS & OAKS (2006), p.63.
336 Compare the respective ways to describe global processes on p. 43.
337 Compare REICH (2008), p. 360.
338 LEYMANN & ROLLER (2006), p. 280.
339 ALONSO ET AL. (2004).
340 HOHPE (2007), p. 442.

limits interactions to two "communication activity instances", implying that one interaction consists out of one *send* and one *receive* activity.[341]

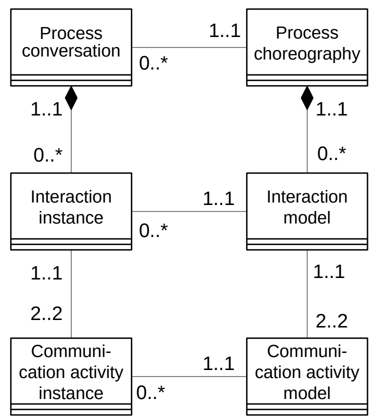


Figure 20: Choreography metamodel from WESKE[342]

2.3.3.3 Relation to Private, Public and Global Process Views

Other than the generally higher level of technical detail, choreographies and choreography interfaces differ from their conceptual counterparts (global and public processes) by their limitation on executable, externally visible process parts,[343] since the internal activities are not relevant on the execution level. Public or global processes, on the other hand, aim at illustrating process logic on the conceptual level. Therefore, in case this helps partners to better understand the process and does not disclose classified information, public and global process models can also comprise internal activities beyond interactions. The relations between the organizational and the technical view can be summarized as follows (compare also Figure 21):

Theoretically, *private processes* can be implemented either as a service orchestration or as choreography; an example for an intra-organizational choreography would be the enactment of a global process by engines situated in different organization units. Nonetheless, in this work we will constrain the enactment of private processes to service orchestrations. Choreography interfaces can be seen as technically detailed *public processes*. Similar to a private process, a *global process* can also be enacted either as a service orchestration or as a service choreography. However, following our conception of CBP where collaborating organizations are autonomous and reluctant to rely on third parties that execute their processes, in the following, the enactment of global processes via service choreographies is assumed.

[341] Note, that this conception does not cover broadcasting of messages, where one sender sends messages to many receivers, since here an interaction would comprise [number of receivers + 1] activities.
[342] WESKE (2007), p. 230.
[343] These comprise interaction activities (e.g. send and receive) as well as the logic that describes the sequence in which interactions are executed.

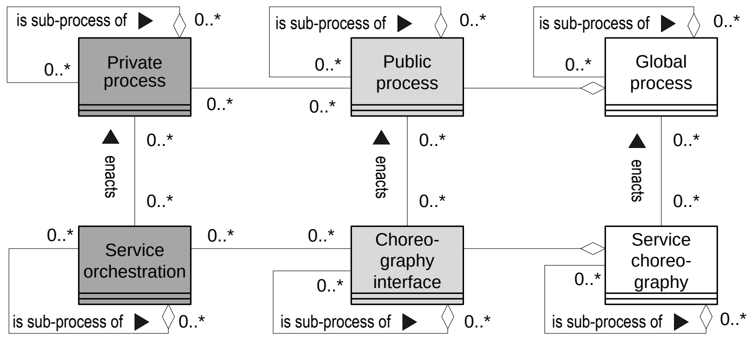


Figure 21: Private, public and global processes related to their technical counterparts

2.3.3.4 Relation of Service-Based to Traditional Workflows

The concepts of orchestration and choreography were created along with the upcoming of SOA and represent relatively coarse-grained workflow classes that distinguish between central and decentral workflow coordination. Before the spreading of SOA concepts, other types of cross-organizational workflows were described; thus, VAN DER AALST differentiated between the following types of cross-organizational workflows, which can be mapped to either orchestration or choreography: [344]

- Capacity sharing: Here a centrally controlled workflow is executed making use of distributed resources; the routing of the workflow is under the control of one so-called workflow manager, while the execution of tasks is outsourced to business partners. This form is closely related to service orchestration where one orchestration engine invokes all services comprised in an executable processes. However, different from orchestration, here the task executed by the partner resembles a “white box”; the central controller has direct access to the resources and can control the way the task is executed.
- Subcontracting: Like in capacity sharing, one business partner has control over the top-level process and “subcontracts” tasks to other business partners. Different from capacity sharing, here the tasks are black boxes for the central actor, allowing partners to realize their tasks with another workflow, which can lead to hierarchical process structures. In SOA terminology, this case would be called orchestration.
- Chained execution: In this solution, the process is split into a number of disjoint sub-processes, which are executed by different among business partners in a sequential order, e.g. the control of the workflow is distributed among the business partners. It requires that a partner transfers or initiates the flow for a case after completing his tasks. This workflow type is equivalent to service choreographies.
- Loosely coupled: This form resembles a choreography where each partner offers a choreography interface (“only the protocol which is used to communicate is public”[345]).
- Case transfer: Here, one global process description is transferred among the collaboration partners, allowing them to execute specific tasks in it. Therefore, each process in-

[344] VAN DER AALST (1999).
[345] VAN DER AALST (1999), p. 644.

stance resides at any given time at exactly one location. A process instance can be transferred to balance the workload or because tasks are not implemented at all locations.[346] Since no central orchestration engine coordinates the tasks and the global process description is not partner specific, this workflow type also resembles a choreography (which is realized not via distributed choreography interfaces but only via one global choreography description, e.g. a global process).

CHEBBI, DUSTDAR & TATA described the same types and added the "public-to-private" type.[347] This refers to the approach described by VAN DER AALST,[348] where a global process is specified for a collaboration and each collaboration partner can alter the parts assigned to him following rules for process inheritance. Thus, it is closely related to the loosely coupled type since it also describes how choreography interfaces of collaboration partners can be derived from a global process model. HAN, SHIM & KWAK distinguished between centralized, decentralized, distributed and completely distributed workflow systems.[349] As distinguishing criterion, they use the allocation of the "task managing instance", which, in SOA terms, represents the service responsible for executing certain tasks. However, their "centralized workflows" do not correspond to orchestrations, since tasks are executed on the coordinating server and cannot be distributed over a network. Their "decentralized workflow", on the other hand, represents an orchestration, since here the tasks are outsourced to other servers. In addition, the workflow types "distributed" and "completely distributed" apparently represent orchestrations, since they rely on a central workflow coordination. Furthermore, more coarse-grained classes of distributed workflows can be found at SCHULZ & ORLOWSKA (outsourced vs. distributed workflows) and WERTH (output pools, output nets, value chain and outsourcing).[350]

2.3.4 On the Usage of Standards for Modeling and Executing CBP

The creation of a new modeling language or the comprehensive extension of an existing language for modeling or enacting collaborative business processes in not in the scope of this work. Rather than describing *how* things are modeled by providing a syntax including symbols and rules for the arrangement of symbols, the aim is to provide metamodels describing *what* has to be modeled. However, to model examples and as a reference for the metamodels of the AIOS, selected modeling languages are used in the following.

Apart from being widespread and state-of-the-art, business process modeling languages that can serve as references for the AIOS should support the automation of business processes in the context of SOA and the modeling of collaborations. However, all current modeling languages (e.g. UML, Petri Nets, BPMN and EPC) show deficiencies in the modeling of collaborative business processes.[351] UML and Petri Nets, for example, concentrate too strongly on technical aspects. BPMN likewise has gaps in business level modeling, but in contrast to EPC, supports swimlanes.

---

346 VAN DER AALST (1999) also describes a derivation of this type called "extended case transfer". There the collaboration partners can modify tasks they are responsible for by deriving specializations of the general descriptions of their task described in the global process.

347 Compare CHEBBI, DUSTDAR & TATA (2006) pp. 145.

348 Compare VAN DER AALST (2002 B2B).

349 Compare HAN, SHIM & KWAK (2000).

350 Compare SCHULZ & ORLOWSKA (2004) and WERTH (2006), pp. 41.

351 Compare ZIEMANN, MATHEIS & FREIHEIT (2007).

While the lacking support of the organization dimension is a deficiency of BPMN, the inclusion of swimlanes and pools supports the modeling of SOA concepts, e.g. displaying the interfaces of services and processes, or transactional spheres.[352] However, recently, collaborative extensions of the EPC were developed that describe the usage of swimlanes in the EPC.[353]

Though the EPC focuses rather on the business design level and the BPMN focuses rather on technical processes and process automation,[354] a direct mapping to BPEL processes is possible for both EPC and BPMN. For BPMN, a mapping to BPEL is already contained in the language specification,[355] and BPMN elements match well with BPEL concepts; for example, both languages support elements for distributed transactions. The EPC is also a suitable basis for BPEL generation and the core elements of EPC (functions and events) map to the core elements of BPEL (web service invocations and various types of events). KRUCZYNSKI evaluated the usability of BPMN and EPC for modeling technical business processes and found no significant differences between the languages.[356] Due to its proximity to Petri nets as well as the large amount of respective research, the EPC is currently better suited for a formal analysis of business processes than BPMN. Thus, a large amount of literature and corresponding tool support exists on the formalization and analysis of EPC models.[357] Another advantage of the EPC is its foundation in a comprehensive framework (ARIS) that ensures that the EPC covers all aspects of enterprise modeling, including the organization dimension.

In summary, it can be said that the EPC allows for a more comprehensive business process modeling while the BPMN is closer aligned with service-based workflows. Since the first aspect is more important for the AIOS, most process models illustrated in this thesis are based on EPC. Due to its dominant role as a standard for executing SOA-based workflow, BPEL is used to display processes on the execution level, while UML Class Diagrams are used to illustrate metamodels.[358]

---

[352] Thus, KRUCZYNSKI (2008) recommends using BPMN for SOA modeling rather than EPC.
[353] Compare KLEIN, KUPSCH & SCHEER (2004) and WERTH (2006). The EPC is described in more detail in Chaper 4, compare p. 135.
[354] Compare KRUCZYNSKI (2008).
[355] Compare OMG (2006 BPMN).
[356] Compare KRUCZYNSKI (2008).
[357] Compare for example MENDLING (2007) or KINDLER (2003).
[358] Compare OASIS (2007) and OMG (2000). In Chapter 4, the suitability of BPEL and the implied usage of choreography interfaces (instead of WS-CDL global processes) are further discussed, compare for example Footnote 585.

3 Developing the AIOS Structure

In the previous chapter, definitions and basic concepts needed in the development of collaborative business process were described. In this chapter, the structure of the AIOS is developed that builds on these concepts and extends them into a comprehensive system for describing and enacting collaborative business processes. While this chapter focuses on the overall structure of the AIOS, in the following Chapters 4 and 5, static and dynamic aspects of the AIOS are described in detail.

Method for Developing the AIOS Various procedure models for the development of architectures exist, for example, the TOGAF Architecture Development Lifecycle or the approach from REIJERS & VAN DER TOORN.[359] However, due to the broad application range of architectures in the context of information systems, a (generic) procedure model for developing information system architectures must stay on a very abstract level. Accordingly, most procedure models for developing information systems architectures resemble generic lifecycle phases, as known, for example, from software development.[360]

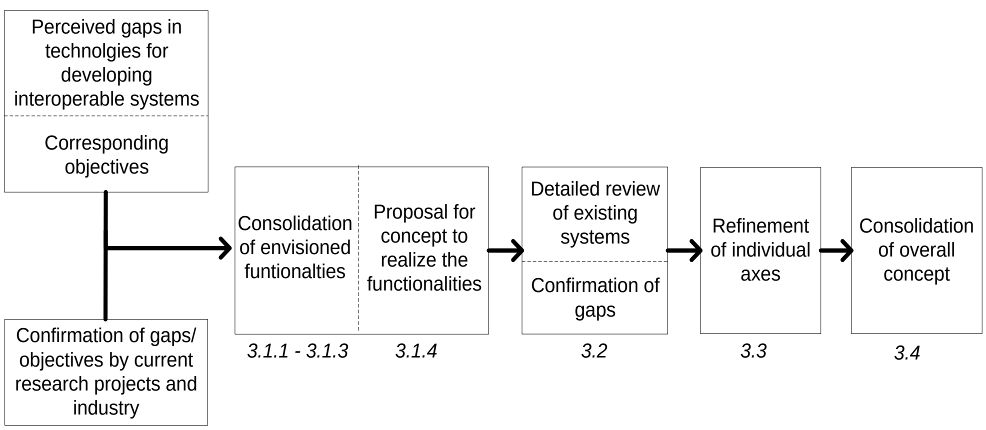


Figure 22: Research method followed in this chapter

Figure 22 illustrates the method to develop the overall structure of the AIOS. In a first step, following the state-of-the-art overview and the gaps proclaimed in Chapter 1, functionalities to be fulfilled by the AIOS are specified; here, it is also explained, how far these requirements were confirmed in recent research projects. Based on the envisioned functionalities, a first AIOS concept is described that complies with these requirements. In a second step, a more detailed state-of-the-art review is executed, where individual frameworks and architectures are reviewed and compared to the AIOS concept. After the confirmation of the initially proclaimed gap in the state-of-the-art and the novelty of the AIOS, the individual axes of the architecture are refined; finally, the overall structure of the AIOS is consolidated.

[359] Compare THE OPEN GROUP (2003) and REIJERS & VAN DER TOORN (2002).
[360] Compare also THELING (2008), p. 60.

3.1 Requirements Specification

ABRAN & MOORE described the phases *requirements elicitation* and *requirements analysis* inside the requirements gathering process. Further, they explained, that this process is not a discrete front-end activity, but rather a process initiated at the beginning of a project that continues to be refined throughout the development life cycle.[361] While the phase of requirements elicitation directly tackles the sources from which the requirements are gathered, in the phase of requirements analysis, requirements are classified; afterwards, an architectural design is created based upon them.

However, in design science research, requirements for the design of an artifact do not necessarily have to be gathered empirically, for example, by questioning potential users of the solution. On the contrary, design decisions can be based on the perceived gaps in existing solutions as well as allegedly soft elements like "creative inspiration or gut instinct".[362] The requirements guiding the AIOS development can be classified into two categories: Envisioned functionalities based on gaps perceived in the state-of-the-art and requirements stated by industrial stakeholders, stemming from individual use cases. The envisioned functionalities are described in the following; afterwards requirements from industrial stakeholders are used to confirm the validity of these design goals. After a further confinement of the requirements, the concept (or, in the words of ABRAN & MOORE, the architectural design) for a solution fulfilling the requirements is described.

3.1.1 Envisioned Functionalities

This thesis follows a design science approach, where new knowledge is created in a deductive manner. Thus, ensuring a broad applicability of the designed artifacts, the requirements are not based on the needs of individual enterprises, but on envisioned functionalities and corresponding gaps of current solutions. The deduction of these gaps is based on a literature review,[363] while the judgment of existing concepts and gaps was influenced by insights gained in recent European research projects in the area of collaborative business and SOA.[364] For example, the participation in three different European research projects on interoperability as well as the participation in different conferences on the subject,[365] led to an exchange of knowledge with many researchers as well as industrial stakeholders on the topic, resulting in an overview of existing solutions and gaps in interoperability research.

---

[361] Compare ABRAN & MOORE (2004), pp. 2-2.

[362] HEVNER (2009), p. 127. In this context, HEVNER (2009), p. 127, stated, "Design science research is grounded on existing ideas drawn from the domain knowledge base. Inspiration for creative design activity can be drawn from many different sources to include rich opportunities/problems from the application environment, existing artifacts, analogies/metaphors, and theories".

[363] In Chapter 1, an overview of major research areas and their strengths and weaknesses regarding a comprehensive development of interoperable information systems was given. This overview is refined in the coming sections, where the individual approaches comprised in the research areas are reviewed.

[364] Compare also the description of the research background in Chapter 1, pp. 5.

[365] From 2005 until 2008, the author was working on several European research projects, compare pp. 5. Conferences attended on the subject include for example: Interoperability for Enterprise Software and Applications (I-ESA), Multi-Konferenz Wirtschaftsinformatik (MKWI), Internationale Tagung Wirtschaftsinformatik (WI) and the Hawaii International Conference on System Sciences (HICCS); compare ZIEMANN ET AL. (2007), ZIEMANN ET AL. (2006), ZIEMANN, KAHL & MATHEIS (2007) and ZIEMANN, MATHEIS & WERTH (2008).

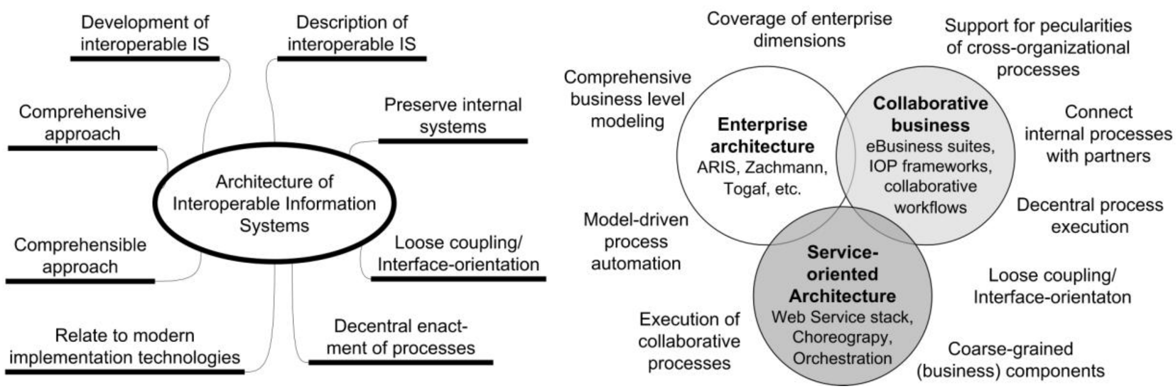


Figure 23: Objectives of the AIOS and characteristics of related approaches

In the motivation for this thesis, the goal of developing a holistic method for the description and the enactment of collaborative business processes was described. Further, the insufficiencies of current methods in fulfilling this goal were sketched out, most importantly, a lack of integration among the fields of enterprise architectures, collaborative business and SOA. More specifically, it was described that only a combination of them can provide a *comprehensive, business-driven development method to describe and enact collaborative business processes*. Figure 23 illustrates the strength of the three fields as well as the objective of the AIOS to combine their strengths.[366] Aiming at the closure of the gap in existing solutions for a systematic development of interoperable information systems, we define the functionalities to be fulfilled by the AIOS as follows:

1. Construction of interoperable information systems: The AIOS should incorporate a method for the development of interoperable information systems. On the one hand, it should enable an individual organization to prepare its information systems for the participation in collaborative business processes. On the other hand, it should also support a multi-lateral development of collaborative systems, where collaborating enterprises already know that they want to collaborate and for this purpose want to extend their systems to set up a specific collaboration context.
2. Description of interoperable information systems: Apart from the constructive aspects, the AIOS should also support the description of interoperable information systems. It should explain how the different aspects of the information system of one organization, which are necessary to connect it with other information systems, could be efficiently described.
3. Comprehensive approach: The AIOS should comprehensively cover the aspects needed in the enactment of collaborative business processes, starting on the business requirements level and reaching down to the execution level; not only describing interaction activities, but also relating those to internal processes.
4. Comprehensible approach: Practitioners find many enterprise architectures difficult to comprehend and to apply.[367] To ensure its applicability, a major requirement of the AIOS is comprehensibility. Two ways of fostering comprehensibility can be distinguished: First, re-

[366] The characteristics of the three fields including their weaknesses and strengths were summarized in Chapter 1 (pp. 2); below these fields and the individual approaches comprised in them are described in detail (compare pp. 77).
[367] Compare WEERAKKODY, JANSSEN & HJORT-MADSEN (2007).

ducing the size of the modeled system, second, structuring the system elements and their relations in an easy to understand manner.

5. Relate to modern implementation technologies: The AIOS should exploit and support the possibilities for enacting collaborative business processes offered by modern execution technologies like SOA.
6. Preserve internal systems: Instead of modifying internal systems to the needs of each collaboration, internal (legacy) systems should be exposed to as few changes as possible. Thus, instead of being modified, internal processes should be *connected* with external processes.
7. Interface-orientation: To enable a separation of internal and external processes, interfaces should be used to overcome heterogeneities among systems and to protect private information. Interfaces should be provided both on a conceptual and technical level and describe the behavioral possibilities and constraints of an organization to collaboration partners. This supports a decentral approach, where partners first publish their possibilities and expectations and then can optionally adapt their interfaces to optimize a collaboration with partners.
8. Loose coupling: Further, the interoperating systems should not be tightly integrated but rather, the connection among the systems should be easy to establish and abort.
9. Decentral enactment of processes: Since collaborations consist of autonomous organizations with equal rights, no central party should enact the collaborative business processes. Instead, each party should enact its part of the collaborative business process, based on previously made agreements among the organizations.

3.1.2 Confirmation of Requirements by Industrial Stakeholders

The requirements for the AIOS – representing a holistic architecture for the development of interoperable information systems – were also expressed in recent research projects: stakeholders from industrial enterprises as well as from public administrations were lacking a comprehensive system for the preparation and enactment of collaborative business processes.[368] Though separate solutions for interoperability among enterprise information systems have been created from various sides, a holistic system that integrates these different approaches is missing, as perceived by both scientists and industrial stakeholders.[369] Thus, in a survey among the public administrations participating in the R4eGov project, the practice partners expressed the need for a comprehensible and comprehensive interoperability method that connects high-level requirements with the execution level. For example, they were interested in interoperability methods that would take into account existing enterprise architectures like the ZACHMANN framework or TOGAF,[370] and that would reduce the gap between the strategic and the execution level, ensuring the compliancy of IT solutions with legal obligations.[371]

---

[368] The projects referred to are INTEROP, ATHENA and R4eGov. Detailed requirements for developing interoperable information systems based on the use cases of these projects can be found in ATHENA (2005), BOUJRAF, AYDOGMUS & VERMER (2006), BOUJRAF & NOBLE (2007), MATHEIS ET AL. (2007), SPADONE (2006) and DIEDRICH ET AL. (2007).

[369] Compare also PANETTO, SCANNAPIECO & ZELM (2004), BOUJRAF, AYDOGMUS & VERMER (2006), p. 23 and LIEGL (2008), p. 4.

[370] Compare ZACHMANN (1987), THE OPEN GROUP (2003).

[371] Compare MATHEIS ET AL. (2007), pp. 53.

3.1.3 Delimitations and Premises of the AIOS

Tackling the requirement of comprehensibility and applicability, the AIOS does not try to cover every aspect of interoperability: it concentrates on those aspects of interoperability, which are needed for enacting collaborative business processes. More specifically, the delimitations and premises of the AIOS are:

High Level Interoperability Barriers The AIOS does not tackle strategic interoperability barriers; for example, concepts for the finding of (strategically) suitable collaboration partners are not part of the AIOS. Neither are "soft" interoperability barriers that might hinder the enactment of collaborative business processes, like the existence of different organizational cultures.

Agreement on Syntax Presumed If different parties want to communicate, at some point of time a common understanding of the interaction means has to be established. This can either happen by using the same language (e.g. two collaborating enterprises use the same syntax and semantics to describe business process elements) or by mapping heterogeneous languages to a commonly understood intermediate language.[372] Thus, certain agreements must exist among collaboration partners on a shared syntax and semantics. In the AIOS, standards for the different enterprise dimensions and vertical levels are named; however, these are rather seen as a proof of concept instead of binding standards. A mechanism to map the syntax of heterogeneous languages to each other is not in the scope of the AIOS, either. Therefore, it is presumed, that the collaboration partners agree on a common syntax to specify elements used in the collaboration. Nevertheless, the AIOS supports an agreement on shared collaboration semantics, since internal concepts are mapped to concepts (global models) known to all collaboration partners.

Ability to Specify Collaboration Type and Corresponding Interfaces In general, three scenarios for developing interoperable information systems can be distinguished: First, an organization does not know the type of collaboration it will engage, but wants its information systems to be *generally prepared* for future collaborations. Second, though an organization does not refer to concrete collaboration partners, it can *describe exactly the type of collaborations* it wants to realize, and is able to describe precisely what the organization itself would contribute and likewise what it expects from (future) collaboration partners. Third, a group of organizations wants to establish a collaboration among its members, and for this purpose requires interoperable information systems. The AIOS only supports the second and the third case: An organization that uses the AIOS must be able to specify its Business Interoperability Interface, comprising the services and processes it offers to potential collaboration partners as well as the corresponding services it expects from them.

No Explicit Support for Automated Composition of Services In a recent research project, user partners required the support of so-called flexible processes. In the corresponding use cases, many actors with various behavioral options were involved.[373] In order to avoid a potentially large amount of completely defined end-to-end processes, it was proposed to describe process chunks, which can be automatically connected at run time. On a technical level, this corresponds to the automatic

---

[372] An example for the second case is provided by ZIEMANN ET AL. (2007), where two enterprises using different enterprise modeling languages inside their organizations are able to enact a collaborative business process by mapping their languages to the POP* format, which served as an intermediate format.

[373] Compare MONDORF ET AL. (2008).

composition of web services into processes.[374] A prerequisite for such a mechanism is that pre- and post-conditions of the process chunks are described in a way that allows their concatenation, and it can be assumed that a multi-dimensional Business Interoperability Interface is a good basis for a comprehensive description of such pre- and post-conditions.[375] However, flexible processes and the automated composition of components are not tackled explicitly in the following.

3.1.4 Proposition of a Solution Compliant with Requirements

A major requirement of the AIOS is *comprehensibility*. The large amount of models and the equally great amount of interdependencies among these models imply a high complexity of architectures in the context of collaborations. To reduce the complexity of information systems (and thus to increase their comprehensibility), it is useful to divide them into different *areas of concern*.[376] Accordingly, many architectures and frameworks for developing information systems are based on orthogonal axes, where each axis represents one concern.[377] Orthogonal axes ensure that the model types resulting from the combination of axes are disjoint and complementary. Thus, to realize the design goals described above, three orthogonal axes are proposed as constituting elements of the AIOS:

- Enterprise dimensions: To describe business processes comprehensively, various dimensions known from enterprise modeling should be covered, e.g. distinct views on processes, data, and organization elements. This will ensure a thorough description of business requirements and increase the re-usability of process elements; for example, functions comprised in one process can also be of another process. Since enterprise dimensions are well known from frameworks like ARIS, their inclusion in the AIOS mainly supports the functionalities 2, 3 and 4 described above.
- Collaborative views: Similar to private, public and global processes, corresponding private, public and global views should be created for all enterprise dimensions. Thus, an external view is created on (private) information systems, as also done in SOA. The distinction between internal systems and external view also protects internal systems and enables interoperability without the need for a significant change to the internal systems. The functionalities supported by collaborative views are 2, 3, 5, 6, 7, 8 and 9.
- Levels of technical granularity: The description of system elements on different levels of technical granularity supports a systematic development of CBP, starting with the business requirements definition and going all the way down to the code level. Thus, the compliancy of the implemented system with business concepts and regulations is provided; additionally, the correlation of different vertical levels supports monitoring functionalities since it enables the visualization of run time activities via conceptual models. Apart from the construction aspect, a multi-dimensional interoperability description is also provided, since the interacting systems are described on different levels

---

374 Compare for example BERARDI ET AL. (2005).

375 Compare also MONDORF ET AL. (2008). In order to provide a good basis for a service discovery, the service interface should describe both business and technical concepts and cover different enterprise dimensions. In addition, a reference of interface elements to globally known concepts is useful to comprehend the characteristics of a service. As explained below, all these functionalities are offered by the AIOS.

376 Compare DIJKSTRA (1982).

377 Examples of such frameworks and architectures are described in the following literature review (pp. 77).

of technical granularity, enabling the synchronization of the collaborating systems on each level. Since it can be assumed that such levels are well known from frameworks like MDA, the functionalities supported by different levels of technical granularity include not only 1, 2, and 3, but also functionality 4 (comprehensible approach).

In addition, the AIOS will exhibit the following characteristics:

Domain-Independent Architecture Comparable to ARIS,[378] the AIOS will be domain independent, i.e. not bound to specific industries but suitable for developing collaborative business processes in the eGovernment domain as well as in the private sector. The generic applicability of the AIOS is supported by the deductive method used in this work and its organization around widespread, domain independent concepts that are incorporated in the three axes described above (vertical levels of technical granularity as known from software engineering, enterprise dimensions, and collaborative views as known from – amongst others – SOA).

SOA as Guiding Principle The AIOS resorts to SOA concepts both for the description of interoperable systems and for the enactment of collaborative business processes. This is already implied by the collaborative view axis, which enforces private, public and global views as implemented by SOA concepts as well as the interface-orientation of SOA. As in SOA, in the AIOS the interface of services and organizations will be published and discovered via repositories accessible to potential collaboration partners. In addition, the principle of forming coarse-grained components that correspond to business functions is followed. In general, the SOA principle of loosely coupled services is well suited for implementing interoperable information systems, since these are supposed to be loosely coupled as well (instead of tightly integrating internal systems of organization A with the internal systems of organization B). In addition to the ones described above, principles associated with a loose coupling include event-based communication and asynchronous messaging, which are also followed in the AIOS. Finally, due to their great acceptance in practice, it is recommendable to use web service-related standards (WSDL, XSD,[379] BPEL etc.) to implement collaborative business processes. Therefore, the AIOS supports a compliance with these standards.

3.2 Review of Approaches Related to the AIOS

Above, a coarse-grained review of existing concepts based on the distinction of existing work relating to the areas *SOA*, *enterprise architecture* and *collaborative business* was executed. In consequence, a gap was described that lead to the definition of the AIOS axes. To confirm the lack in existing approaches, in this section individual frameworks and architectures of the afore-mentioned research areas are reviewed and compared to the AIOS. Note that due to the vast number of architectures and frameworks, it is not within the scope of this work to provide a complete description of all approaches developed in the three areas;[380] instead, the objective is to describe representative

[378] ARIS is domain-independent and is used for example both in eGovernment and in eCommerce. Compare SCHEER & JOST (2002), SCHEER ET AL. (2004) or SCHEER, KRUPPKE & HEIB (2003).

[379] XSD stands for XML Schema Definition Language, compare W3C (2009).

[380] For an exhaustive state-of-the-art overview of approaches for SOA development, development of interoperable systems and enterprise architectures refer, for example, to BASTIDA ET AL. (2009), ATHENA (2007), SCHMIDT ET AL. (2007).

approaches, which illustrate the possibilities and limitations of current approaches in each area. Apart from a general introduction of the different approaches, the following aspects of the approaches are discussed: how strong is the support for model-driven development, for interoperability, and for a comprehensive modeling of enterprise elements. Further, it is of interest, whether these approaches are based on orthogonal axes similar to those of the AIOS.

3.2.1 Enterprise Architectures

Building on the generic description of enterprise architectures in Chapter 2,[381] in the following, a selection of representative enterprise architectures is reviewed:[382]

Zachman The Zachman framework is probably the best known enterprise architecture framework. It is based on a 2-dimensional matrix: The first axis of the matrix represents different stakeholder roles (planner, owner, designer, builder, programmer, and user) and the second axis represents different perspectives: what (data), how (function), where (network), who (people), when (time) and why (motivation). An advantage of the framework is that it is easy to understand. Disadvantages are its large number of cells – representing an obstacle for the practical applicability of the framework – and the fact that the relations between the cells are not well specified.[383] The macro-level of the framework does not fit with SOA development, as it is too coarse-grained to develop services; due to the strategic perspective of the framework as well as the lack of operational information it is not well suited for the design of the execution level.[384] Thus, the applicability of the framework for a model-driven development is limited.

ARIS The Architecture of Integrated Information Systems distinguishes five enterprise dimensions: organization, function, output, information and control views. For each enterprise dimension, three levels of technical granularity are described, which resemble the MDA-levels CIM, PIM and PSM: requirements definition, design specification and implementation description. Thus, ARIS is based on two orthogonal axes (compare also Figure 31).[385] In comparison to the other frameworks, ARIS supports a systematic, model-based software development based on a genuine business requirements specification.[386]

CIMOSA The objective of the "Computer Integrated Manufacturing Open System Architecture" (CIMOSA) is to provide a framework for analyzing the requirements of an enterprise and to translate these into a system, which enables and integrates the functions that match these requirements. The framework is based on three orthogonal dimensions: First, a dimension of genericity. Indicating the genericity/specificity of the comprised elements, three levels are distinguished: a generic level consisting of basic building blocks, a partial level comprising a library of models applicable to particular purposes and a particular level consisting of models for a particular enterprise. Second, a dimension of technical granularity, consisting of a requirement, a design and an implementation

---

381 See pp. 33.
382 For overviews and evaluations of different enterprise architectures see also ATHENA (2004), SCHÖNHERR (2004), NORAN (2004), SCHEKKERMANN (2004), GREEFHORST, KONING & VAN VLIET (2006) or ZIEMANN ET AL. (2008).
383 Compare LANKHORST (2005), pp. 24.
384 Compare BASTIDA ET AL. (2009), p. 47.
385 Compare SCHEER (2000).
386 Details about the ARIS dimensions are described below.

level. Third, a dimension for different enterprise views, comprising a function, an information, a resource and an organization view.[387]

GERAM In response to an evaluation of existing enterprise architectures, the "Generalised Enterprise Reference Architecture and Methodology" was developed as a superset of these architectures. The central element of GERAM is the "Generalised Enterprise Reference Architecture" which provides general concepts for the description of enterprises. The concepts in this architecture are divided into three categories: 1. Concepts to describe the role of humans (e.g. in the form of organization models describing the roles and functions of actors), 2. Concepts for the description of the business processes, 3. Concepts for the description of the technology for modeling and executing business processes.[388]

FEAF The Federal Enterprise Architecture Framework was first released in 1999 and has the objective to improve interoperability within the United States government by creating one Federal Enterprise Architecture (FEA) to integrate the separate architectures of the various federal agencies. The framework comprises a collection of reference models, which represent a common taxonomy and ontology for describing IT resources among the agencies. These include the Performance Reference Model, the Business Reference Model, the Service Component Reference Model, the Data Reference Model and the Technical Reference Model.[389] FEAF divides the elements in the architecture into four fields: business, data, application and technology.[390] In summary, the FEAF covers a broader area than AIOS; instead of concentrating on the model-driven development (and corresponding description) of collaborative business processes, it provides coarse-grained reference models for the structure and content of information systems needed in federal agencies.

TOGAF The Open Group Architecture Framework was developed in 1995 and is based on an architecture framework from the US Department of Defense. It consists of three main parts: An architecture development method, a collection of reference models (The Enterprise Continuum) and a set of resources (TOGAF Resource Base). Like FEAF, TOGAF also distinguishes four different subsets of an enterprise architecture: A business, a data, an application and a technology architecture.[391]

GEA Similar to TOGAF, the Queensland Government Enterprise Architecture (GEA) is an enterprise architecture for the public domain. It was released in 2005 replacing the formerly used Government Information Architecture to improve interoperability of IT systems and the sharing of information resources across Queensland government agencies. It offers the same views on information systems as TOGAF by providing distinct layers for business, information, application and technology.[392]

TEAF Another example of a public domain specific framework is the Treasury Enterprise Architecture Framework. The purpose of TEAF is to support architecture development and management

---

[387] Compare Beeckman (1989).
[388] Compare IFIP-IFAC (2000).
[389] Compare FEA (2007).
[390] Compare Scheckermann (2004), pp. 105.
[391] Compare The Open Group (2003).
[392] Compare Queensland Government Chief Information Office (2007), p. 9.

with regard to requirements in the field of treasury, supporting the implementation of the architectures in the respective Treasury Bureaus and offices. To classify “work products”, the TEAF describes a 2-dimensional matrix consisting of four architectural views (functional, information, organizational, and infrastructure) and four perspectives (Planner, Owner, Designer, and Builder).[393]

Summary The reviewed enterprise architectures offer modeling support for various dimensions of an enterprise. Interestingly, many share the distinction between business, information, application and technology layers. They combine different user perspectives and allow modeling on different levels of abstraction; however, they focus on the representation of intra-organizational elements and lack support for developing collaborations. The reviewed enterprise architectures from the eGovernment domain (FEAF, TOGAF, GEA, TEAF) cover a broader area than AIOS, focusing rather on coarse-grained, strategic issues instead of operational issues like model-driven development. Moreover, many enterprise architectures lack clarity: blurry and varying definitions of essential terms and a plethora of elements often rather lead to confusion instead of providing a comprehensible and applicable development guide.[394] Since most enterprise architectures were created before the rise of SOA, they only partially support SOA concepts like for example loose coupling or interface-orientation. Thus, the weaknesses and strengths of enterprise architectures proclaimed in Chapter 1 and in the first section of Chapter 3 are confirmed.

3.2.2 Approaches in the Context of Collaborative Business

3.2.2.1 CBP Modeling Frameworks

While enterprise architectures provide a comprehensive overview of intra-organizational elements, the frameworks reviewed in the following concentrate on inter-organizational business processes on conceptual and technical levels.

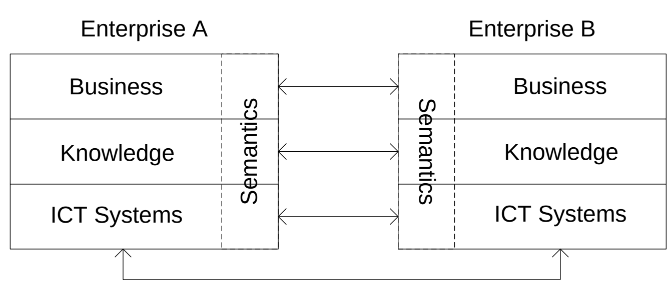


Figure 24: IDEAS Framework[395]

IDEAS The project “Interoperability Development for Enterprise Applications and Software” (IDEAS) [396] was supported by the European Commission and defined the framework illustrated in Figure 24, which also served as a basis for the ATHENA frameworks described below. The framework contains four layers: The business layer comprises “all issues related to the organization and

---

[393] Compare DEPARTMENT OF THE TREASURY (2001), p. 15.
[394] This judgment is shared by GREEFHORST (2006), p. 107.
[395] BLANC (2005), p. 2.
[396] Project Nr. IST-2001-37368.

the operations of an enterprise"[397]. The knowledge layer is also described imprecisely, but seems to cover rather static elements (e.g. products), while processes are described on the business layer. The IT layer, responsible for the execution of the interaction, is allocated at the bottom. Orthogonal to these layers is positioned the semantic layer, which describes the elements of the other layers in a (multi-laterally) understandable way.[398] These layers are very similar to the distinction between business, data, application and technology layers used in eGovernment frameworks that were described above. Like those frameworks, they do not support a model-driven development of interoperable systems nor do they distinguish between internal and external views on organizations.

ATHENA – Process-Centric Framework Two distinct frameworks were developed in the ATHENA project: A framework focusing on model-driven development in the context of SOA (described in the next section, pp. 87), and a framework focusing on the automation of collaborative business processes. The latter framework focuses on the process dimension and uses the public-private-global concept for the automation of CBP.[399] To this end, two orthogonal axes are used; the first axis comprises three vertical modeling levels (business, technical and execution levels), the second axis comprises the three collaborative views (private, public and global).[400] For each vertical level, modeling notations were proposed and transformations between the levels were sketched out.[401] Apart from the process view, no other enterprise dimensions like for example organization or function, are supported. The approach does not focus on the interface (e.g. the public process) description, but rather on the overall CBP establishment methodology, where different procedure models like inside-out or outside-in development were briefly discussed.

INTEROP/Model-Driven Interoperability In the INTEROP project, a framework for model-driven interoperability was developed.[402] This interoperability framework is separated into three parts: conceptual integration (focusing on models and their relationship), technical integration (focusing on software development and execution environment), and applicative integration (focusing on standards and domain models). For this work, the conceptual integration part is relevant, which is based on so-called system aspects that resemble enterprise dimensions. These aspects comprise service aspects, information aspects, process aspects and non-functional aspects. All four aspects are closely related to SOA concepts (service aspects resemble web services; process aspects facilitate service-based workflows, etc.). The system aspects axis can be combined with various other axes, most importantly an axis describing the MDA-levels CIM, PIM and PSM. Apart from the four aspects and the MDA-levels, they propose five other dimension to harmonize CBP models: Genericity (from generic patterns to specific products), model abstraction (instance, model, metamodel), time (state of maturity), degree of composition/granularity (elementary object or coarse-grained service) and viewpoint. The latter seems to refer also to different stages in the software development cycle.

---

[397] ATHENA (2007 IOP), p. 18.
[398] Compare ATHENA (2007 IOP).
[399] Compare for example ATHENA (2007), BORN ET AL. (2009) or ZIEMANN, KAHL & WERTH (2007).
[400] The axes are also illustrated in Figure 31, p. 93.
[401] The notations comprised EPC, Mo^{2}Go (a proprietary notation for technical collaborative workflows, compare MERTINS & JAEKEL, 2006), and BPEL.
[402] Compare ELVESÆTER ET AL. (2005) and INTEROP (2007).

ArKoS In the ArKoS research project an architecture to support collaborative scenarios was developed.[403] Amongst others, a view concept was specified that distinguishes between global and local knowledge (illustrated in Figure 25) and a procedure model describing how to construct collaborations.[404] Similar to this work, they also used ARIS to describe private and global data. However, their view concept is more coarse-grained than the one proposed in this thesis: instead of differentiating inside each ARIS dimension between private, public and global knowledge, they declare individual ARIS dimensions as being either global or local, the only exception being the process dimension, where "process modules" serve as a connection between global and local views.[405]

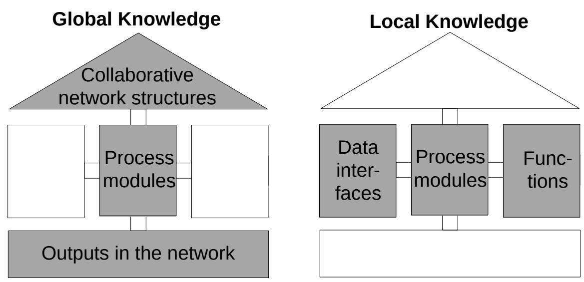


Figure 25: Distinction between local and global knowledge in ArKoS[406]

3.2.2.2 Frameworks for Collaborative Workflows

Some of the collaborative business process frameworks described above are based on concepts stemming from approaches used in the development of collaborative workflows. Compared to the business process frameworks, which rather focus on the requirements level, the workflow frameworks focus on the execution level, where more formal, machine interpretable process models are used. The frameworks focus on the technical description of the process dimension, while other enterprise modeling dimensions are neglected.[407] However, in an effort to generalize the public process concept, first attempts were made to cover additional enterprise dimension in collaborative workflows; CHIU ET AL., for example, suggested applying the Private-Public scheme to different types of flows needed in system integration: data, control, semantic, exception and security flow.[408] Closely related to the frameworks for collaborative workflows and especially workflow views, frameworks for so-called eContracts emerged.[409] Similar to the AIOS, here the externally visible behavior of collaboration partners is described in multiple dimensions. Various approaches for workflow views are described in Chapter 4.

3.2.2.3 eBusiness Protocol Suites

While SOA development in the context of collaborations is a comparatively new research field, collaborative business processes have been implemented via EDI and corresponding networks for

403 Compare LOOS & VANDERHAEGHEN (2007). The acronym ArKoS stands for „Architektur Kollaborativer Szenarien".

404 The procedure model is illustrated Figure 75, p. 183.

405 A discussion of the view concept used in ArKoS is provided below, see p. 110.

406 Adapted from HOFER ET AL. (2005), p. 5.

407 Refer also to Chapter 4, where the view concepts of such frameworks are compared.

408 Compare CHIU ET AL. (2005).

409 Compare KARLAPALEM, DANI & KRISHNA (2001), ZDRAVKOVIC & KABILAN (2005) or CHIU ET AL. (2002).

more than two decades, along with standards for defining interchange and message structures.[410] This kind of interchange description is also called protocol, or (e)Business protocol.[411] Various standards exist to describe protocols; prominent examples are the Business Process Specification Schema (BPSS) of ebXML and the Partner Interface Processes (PIP) of RosettaNet. While ebXML represents a family of XML-based standards standardized by OASIS and UN/CEFACT aiming to provide an open, XML-based infrastructure for electronic business, RosettaNet is an industry-driven consortium aiming at creating, implementing, and promoting open eBusiness process standards.[412]

In general, B2B protocols can be seen as predecessors of web service-based solutions. ebXML, for example, is the most recent eBusiness protocol suite; however, it was initiated before the rise of web services, resulting in a limited support of ebXML for SOA. RosettaNet and ebXML represent complete, self-contained solution stacks developed by a standardization body. The web service stack on the other hand represents a modular, open system, where standards can be developed independently from each other, allowing for a selection of best-of-breed standards.[413] In addition, standards of the web service stack are better accepted by industry than closed systems, like for example ebXML.[414]

However, compared with SOA-based solutions, an advantage of the traditional eBusiness protocols is that they are praxis proven and represent a great body of knowledge, especially regarding the definition of document types. On the other hand, traditional message-oriented systems like EDI have a limited functional scope and are frequently tailored to proprietary needs, but are inadequate to support heterogeneous applications.[415] In general, they lack concepts for loose coupling and interface-orientation; instead of public processes, they concentrate on global processes. Moreover, they do not provide a comprehensive support for all enterprise dimensions but clearly focus on the data dimension.[416]

---

410 Compare UNITT & JONES (1999).

411 MASUD (2003) described the elements of such protocols, comprising interfaces published in a network, choreography descriptions and partner roles, a standard vocabulary and mechanisms to ensure security and trust.

412 Compare CLARK ET AL. (2001) and DAMODARAN (2004). ebXML and RosettaNet are further described in Chapter 4, compare pp. 127.

413 The flexibility of the SOA standards is also enabled by the SOAP messaging format, which allows placing arbitrary elements in the message header; for example, a message header could be created comprising elements of the standards WS-Addressing, WS-Security and WS-ReliableMessaging (compare FERGUSON ET AL., 2003). Note, that originally SOAP was the acronym for Simple Object Access Protocol; however, since version 1.2, officially it is not used as an acronym anymore (and hence not listed in the list of acronyms of this thesis); compare W3C (2007 SOAP).

414 "Closed" in the sense that the standard is owned and updated by one single organization, whereas any party can develop standards complementary to other standards in the web service stack.

415 Compare SCHROTH (2008), p. 74.

416 This is also illustrated by the functionality of *EDI-gateways*, which represent architectures specialized on data exchange. Thus, the messages of various senders can be collected in such a gateway to be analyzed, modified and forwarded to internal recipients; compare SCHNEIDER (2007), pp. 24. In this context, he also writes that EDI-Gateways do not have to be investigated anymore since they are outdated, indicating that EDI solutions will be replaced by modern, SOA-based solutions. For further descriptions and comparisons of eBusiness protocol standards refer to ATHENA (2006 Protocols) or BERNAUER, KAPPEL & KRAMLER (2003).

3.2.2.4 Interoperability Frameworks

As mentioned before, a vast number of interoperability frameworks exist. A recent evaluation is provided by SCHMIDT ET AL.;[417] their selection of interoperability initiatives illustrates the broadness of the term interoperability: Amongst others, they evaluate eGovernment frameworks from various countries, enterprise architectures and standardized approaches for process automation, for example, the workflow reference model of the Workflow Management Coalition. In the following, the Business Interoperability framework and two prominent examples from eGovernment are described: The EIF and a framework from WIMMER for the Austrian government, which builds on the EIF.

Business Interoperability Framework The Business Interoperability Framework was also developed in the context of ATHENA.[418] While the other ATHENA frameworks focused on their construction, this framework focuses on the assessment of interoperable systems. Therefore, it describes constituents of Business Interoperability and outlines how an enterprise may assess and improve its Business Interoperability. To this purpose, it distinguishes four categories: Management of external relationships, employees and culture, collaborative business processes and information systems. To assess the maturity of an organization, five levels of Business Interoperability are used:[419]

1. No interoperability: No awareness of external relationships; interaction with external partners is not planned or performed ad-hoc.
2. Minimum: No previsions for interoperability; individual design of each external relationship.
3. Moderate: Relevance of Business Interoperability is “understood”; Measures for improving interoperability have been taken, but substantial room for improvement remains.
4. Qualified: External relationships are designed for improved Business Interoperability; only few factors are missing on the way to full interoperability.
5. Fully interoperable: Maximum level of Business Interoperability; external relationships can be established with no or few cost involved.

In difference to the AIOS, this framework does not focus on the development of collaborative business processes but rather on a strategic interoperability assessment of individual organizations. Thus, it covers a broader area than the AIOS (e.g. by tackling soft interoperability barriers like organizational culture) but is harder to operationalize.

EIF The European Interoperability Framework was developed within the “Interchange of Data between Administrations” program as a framework for a common understanding of interoperability.[420] Its objective is to support the delivery of pan-European eGovernment services to citizens and enterprises. Therefore, the EIF provides recommendations and guidelines for interoperability among eGovernment services. In the EIF, three levels of interoperability are defined: *organizational* intero-

---

[417] SCHMIDT ET AL. (2007).
[418] Compare ATHENA (2007).
[419] Compare ATHENA (2007), p. 43.
[420] Compare EEUROPE (2005).

perability targets business processes, information architecture and policies in the interaction of different administrations. *Semantic* interoperability aims to establish a common understanding of the exchanged data, information, or process models. The linking of systems and the definition of corresponding technical standards in order to enable seamless communication is addressed by *technical* interoperability. [421] The EIF is based on the assumption that interacting public administrations represent independent organizations that will *not* harmonize (internal) processes with each other, and in consequence recommends the creation of Business Interoperability Interfaces. In this vein, it is also stated that administrations need means to describe, to which globally accessible ("pan-European") services and to which (global) business processes they contribute.[422]

WIMMER ET AL. Based on the EIF, WIMMER ET AL. developed an interoperability framework for Austria.[423] As Figure 26 illustrates, their framework is based on three orthogonal dimensions: 1. Interoperability, 2. Extent and 3. Location. The interoperability dimension comprises six categories: 1. Organization, 2. Process, 3. Content, 4. Data, 5. Components and 6. Protocols. Compared to the EIF, the framework is finer grained: The EIF semantic layer is split up into content (targeting semantics) and data (targeting the syntax of data), the EIF organization layer is split up into organization and process, while the EIF technique layer is split into component and protocol. The second dimension ("Seamless Administration") specifies a certain domain to which the other dimensions apply, like for example eDelivery or ePayment. The third dimension specifies the scope of the collaboration. Here, three categories are specified: local, national and international.

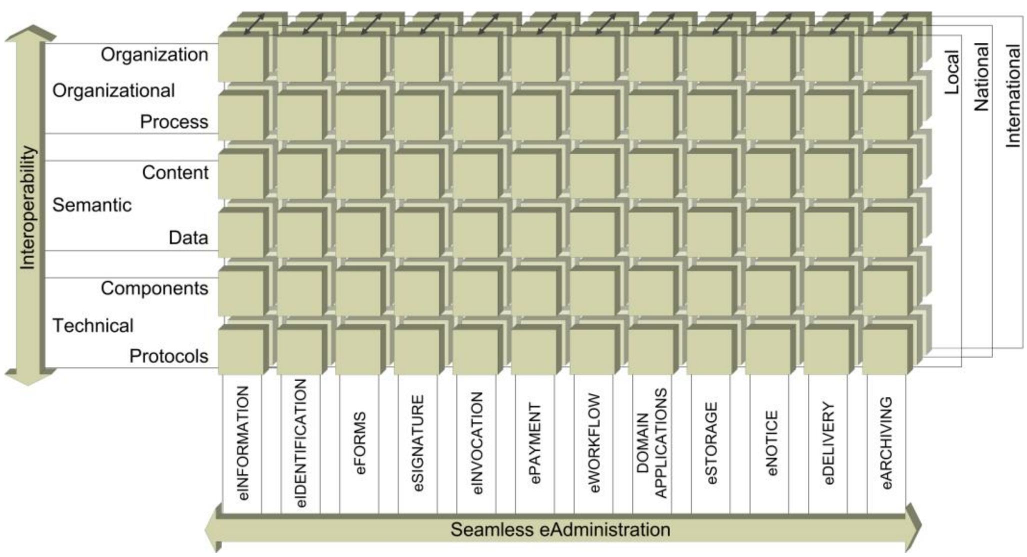


Figure 26: eGovernment interoperability framework from WIMMER ET AL.[424]

The advantage of the EIF is that its three levels are easy to understand. Nonetheless, as also visible in WIMMER'S Framework, the EIF layers do not aim at a method for the development of interoperable systems. Thus, the three EIF levels – organization, semantic and technique – are not

421 EUROPEAN COMMISSION (2004).
422 EUROPEAN COMMISSION (2004), pp. 18.
423 Compare WIMMER, LIEHMANN & MARTIN (2006).
424 DIEDRICH, SCHMIDT & WIMMER (2006), p. 58.

grouped according to increasing levels of technicality; e.g. a message in the semantic dimension can be described on the same level of granularity as a business process in the process dimension. Moreover, the levels seem unrelated or even inconsistent with each other; for example, if the technical level comprises components, then these components should have a conceptual counterpart on the design level (i.e. business functions described on the organizational interoperability level), which does not seem to be the case.

3.2.2.5 Summary

It can be concluded, that the previously proclaimed gaps in current approaches in the context of collaborative business could be confirmed: *eGovernment interoperability frameworks* focus on the descriptive aspects of interoperability by providing standards and technical recommendations for selected aspects of IT system development but lack support for constructive aspects, e.g. suitable vertical levels and procedure models that would support the systematic implementation of collaborative business processes.[425] *Frameworks for collaborative workflows* and *eBusiness protocol suites*, on the other hand, focus too strongly on the execution of processes and therefore neglect conceptual levels and – in the case of eBusiness protocols – interface-orientation. *CBP modeling frameworks* come closest to fulfilling the requirements of the AIOS, but their support for collaborative views is limited to the process dimension.

3.2.3 SOA-Related Approaches

The core idea of SOA is the loose coupling of services, based on descriptions that enable their discovery in a repository. To enable this discovery, a suitable service interface description is essential. Correspondingly, OVERHAGE & TUROWSKI[426] state that a specification framework to describe the externally visible characteristics of services must be the central element of a service development method. In this vein, HEUTSCHI names interface-orientation as a design principle of SOA.[427] This interface-orientation is supported by different SOA concepts and standards, for example, the description of externally visible behavior via choreography interfaces, the description of global processes as choreographies or the description of private processes as orchestrations. Thus, SOA inherently supports collaborative views and the loose coupling needed in collaborations. The weakness of current SOA development approaches is the lacking connection with business requirements modeling and a corresponding business-based SOA development method.[428] Frameworks for a systematic software development in the context of SOA comprise:

SOAD ZIMMERMANN, KROGDAHL & GEE developed an interdisciplinary approach for SOA development called Service-Oriented Analysis and Design (SOAD).[429] Based on the gaps they experienced in SOA implementation projects, they argue that Object-oriented design and analysis, enterprise architectures and business process management should be integrated into a coherent service modeling approach. Figure 27 illustrates the scope of SOAD. As the graphic shows, it distinguishes

---

[425] This is also indicated by MATHEIS ET AL. (2007) and SCHMIDT ET AL. (2007).
[426] OVERHAGE & TUROWSKI (2007).
[427] HEUTSCHI (2007).
[428] In response to these missing aspects, in 2008 the EU-supported SHAPE project started, whose overall aim is to provide a comprehensive method for supporting the engineering process for SOA. Compare BASTIDA ET AL. (2009), p. 13.
[429] Compare ZIMMERMANN, KROGDAHL & GEE (2004).

between business process modeling (e.g. UML models), enterprise architectures (e.g. the Zachman framework), technical solution architectures and programming techniques used on the application level. Similar to the AIOS, SOAD argues in favor of combining approaches from enterprise modeling and technical architectures to establish a comprehensive development method. However, SOAD takes a more narrow approach: While the AIOS aims at supporting different enterprise dimensions equally, SOAD focuses on the development of services to answer the question "what makes good services"[430]. Despite the argumentation for an integration of business-level concepts, the approach remains rather technical and does not cover a comprehensive business requirements modeling. While the AIOS only goes down to the service level, in SOAD also the implementation of services with classes is described. Another difference is the lacking support for collaborative views in SOAD.

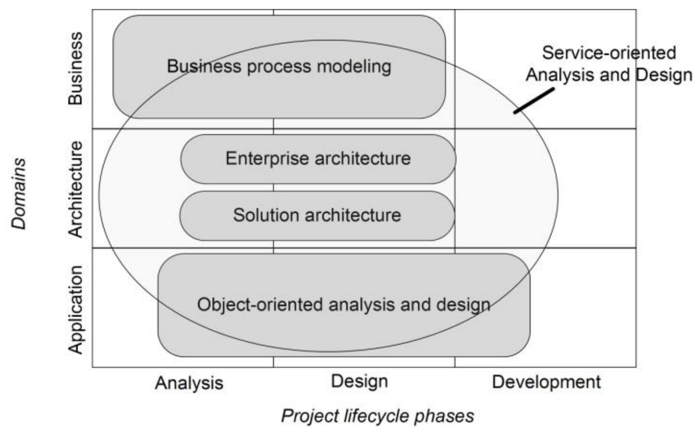


Figure 27: Scope of SOAD[431]

ATHENA – SOA-Centric framework The second major ATHENA framework, illustrated in Figure 28, does not focus on processes but on model-driven development in the context of SOA. Based on the IDEAS framework, it also divides interactions between enterprises into various levels: The *Collaborative enterprise modeling* layer contains models describing the processes, organizations, products and systems in the collaboration context. The second layer, called *modeling of cross-organizational business processes*, focuses on process modeling using the public/private/global concept. The third layer concerns the *flexible execution and composition of services*. The lowest layer, information interoperability, describes documents and messages.[432] Orthogonal to these layers are situated one dimension for model-driven development of interoperable systems and one dimension to describe the syntactic models in the four layers with computer processable semantics. In comparison to our proposal, again, the interface focus is lacking. Moreover, the four axes are not convincing as a basis for model-driven development, since they are not disjoint and do not cover all enterprise dimensions: the first and the second layer are both tackling processes; on the other hand, an organization dimension is missing (though it may implicitly be contained in the collaborative enterprise modeling layer).

430 ZIMMERMANN, KROGDAHL & GEE (2004), p. 1.
431 Adapted from ZIMMERMANN, KROGDAHL & GEE (2004).
432 Compare ATHENA (2007 IOP).

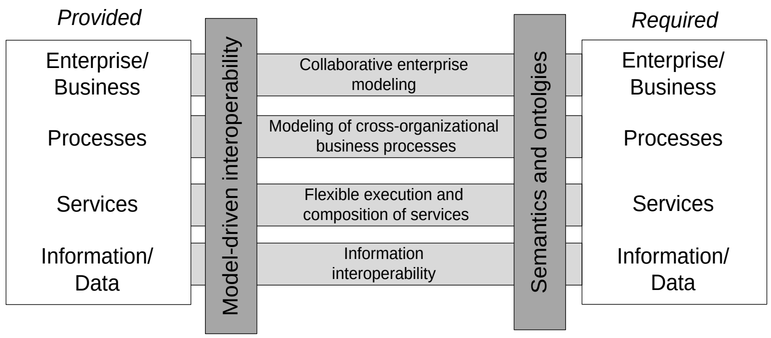


Figure 28: ATHENA SOA-centric framework[433]

Development of Business Components Since SOA is originating from a technical background, the majority of these approaches neglect the business level.[434] An approach for modeling services from a business perspective is provided by ACKERMANN ET AL., who described the seven aspects of a business component illustrated in Figure 29. Recently HEUTSCHI proposed a slight modification[435] of this approach to describe services in the context of SOA, resulting in the description of six levels:[436]

1. The interface level, which technically enables a service to communicate with other services and hence describes the syntax of protocols, messages, etc.
2. The behavior and information level complements the interface level with semantic aspects and describes pre- and post-conditions of service operations.
3. The coordination level describes sequences in which service operations can be used and in which sequences the service can interact with others. Thus, this level describes choreography aspects.
4. The task and terminology level relates elements of the previously described levels with terms used on the business level.
5. The quality level describes technical, non-functional characteristics and pre-conditions of a service.
6. The marketing level describes non-functional service characteristics relevant for the organizational/business level, e.g. people's responsibility for its maintenance, its release status or the domain that the service belongs to.

In difference to the AIOS, their system for describing interfaces does not support a model-driven approach. The three vertical levels of the AIOS allow for a systematic requirements modeling covering all enterprise dimensions and a corresponding model-driven refinement. The six levels described above, on the other hand, do not support such a development, since they represent a mixture

---

433 ATHENA (2007 IOP), p. 13.
434 For example, all widespread SOA standards comprised in the web service stack (like WSDL, SOAP, and BPEL) are situated on the execution level and do not take into account enterprise elements like organization units or business goals.
435 The modification consists of the joining of the "task level" with the "terminology level", the extension of the "behavior level" regarding information aspects, and the renaming of the "interaction level" to "coordination level".
436 Compare HEUTSCHI (2007), pp. 32.

between levels of different technical detail (e.g. MDA-levels) and enterprise dimensions. For example, level 1 and 2 both describe operations and thus correspond to the functional enterprise dimension, on different levels of technical granularity. Level 3 corresponds to the process dimension, though the level of technical detail is unclear, i.e. if this process is on the technical level (e.g. in form of a UML Activity Diagrams) or on the code level (e.g. BPEL).

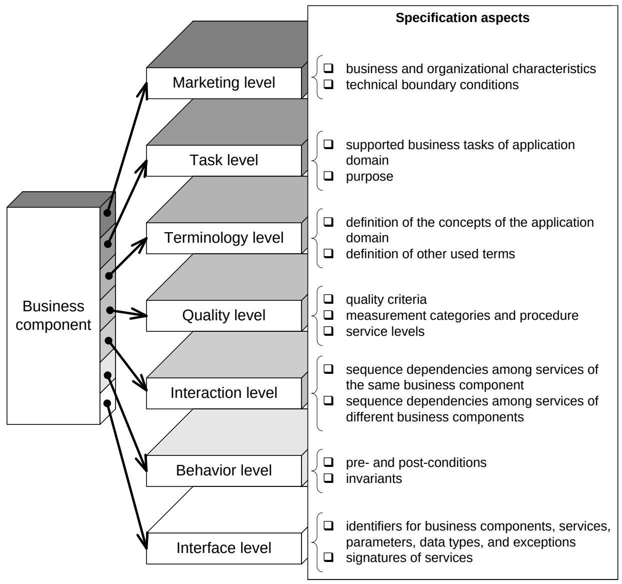


Figure 29: Business component levels according to ACKERMANN ET AL.[437]

Enterprise Collaboration Architecture (ECA) was published by the Object Management Group (OMG) in 2004. At first glance, the ECA bears many resemblances to the AIOS. Thus, the ECA also represents a “collaboration-based modeling approach that can be used at different levels of granularity and different degrees of coupling”[438]. Coming from the OMG, ECA supports a model-driven development based on the MDA model types, using for example metamodels to describe the model semantics. However, the enterprise dimensions needed in collaborations are only partly covered in ECA; more specifically, four types of UML models are used: business process, components, entities and events. The first three model types are similar to the process, function and data dimension known from enterprise modeling; an organization dimension is missing. ECA also supports interface-orientation and SOA concepts, e.g. choreography, choreography interfaces and orchestrations (the latter are referred to as compositions). Nevertheless, such collaborative views are specified only for the process dimension, while in the AIOS it is described explicitly for all enterprise dimensions. In summary, ECA represents a rather technical approach, focusing on the specification of technical components, and lacks a comprehensive business requirements modeling as well as the coherent application of collaborative views.

---

437 ACKERMANN ET AL. (2002), p. 4. Following the illustration from FETTKE & LOOS (2007), p.630, the description of the specification aspects where slightly adapted.

438 Compare OMG (2004), p. 2-2.

Summary Though the importance of a connection to the business level is acknowledged in recent SOA frameworks, the declaration that such frameworks focus on technical aspects and lack a comprehensive support of enterprise dimensions, could be confirmed, as well as a lacking support for systematic system development. The approach for describing services from a business perspective (e.g. business components) on the other hand, exhibits a lacking connection to SOA execution level concepts as well as a lack of collaborative views. Further, it was confirmed, that some SOA frameworks support collaborative views, though these view are limited to the process dimension.

3.2.4 Conclusions

In the previous three sections, individual approaches in the fields of SOA, enterprise architecture and collaborative business as well as their relation to the AIOS were described. Based on this review, now it can be specified, how the AIOS relates to the landscape of existing approaches and to what length the reviewed approaches support the three AIOS axes (enterprise modeling dimensions, model-driven development levels and collaborative views).

3.2.4.1 Support of Enterprise Modeling and Model-Driven Development

Figure 30 provides an overview of the work reviewed above, ordered along the axes of covered enterprise dimensions and their coverage of different levels of technical granularity.[439] *Enterprise architectures* like ARIS cover many enterprise dimensions and comprise different levels of technical granularity: Usually they start on the strategic level and are less detailed on the execution level. The reviewed *interoperability frameworks* are incomplete regarding the coverage of enterprise dimensions; they cover a broad area taking into account also soft interoperability barriers, but are not well suited for systematic enactment of processes. *eBusiness protocol suites* usually focus on the data dimension and rather support the implementation of processes instead of their business level design. *CBP modeling frameworks* as provided by ATHENA or ArKoS focus clearly on the process dimension, other dimensions are mentioned but described in low detail. Starting at the business level, they intend to automate processes, but the implementation level is not described precisely. Surprisingly, some *frameworks for cross-organizational workflows* capture more enterprise modeling dimensions: Besides the data dimension especially the organizational roles/security dimension is tackled;[440] nevertheless, the majority of them are restricted to the execution level. Approaches for *model-driven development of SOA* usually start from a technical design level and focus on processes and components (e.g. classes). In the context of SOA, various approaches for the *modeling of services* were created which concentrate on few enterprise dimensions and are rather technical; if approaches for service choreographies are added, apart from the functional also the process dimension is covered by service descriptions. On the business side, comparable work exists for the description of *business components*, which cover more aspects of business requirements modeling but are difficult to relate to execution level concepts.

---

[439] For the sake of readability, this illustration abstracts from deviating members of the shown groups; for example, there might be MDA-approaches, which start at a business-level – but this does not change the fact that in general MDA approaches start at a rather technical level, e.g. UML diagrams.

[440] For example in the approach of CHIU ET AL. (2005).

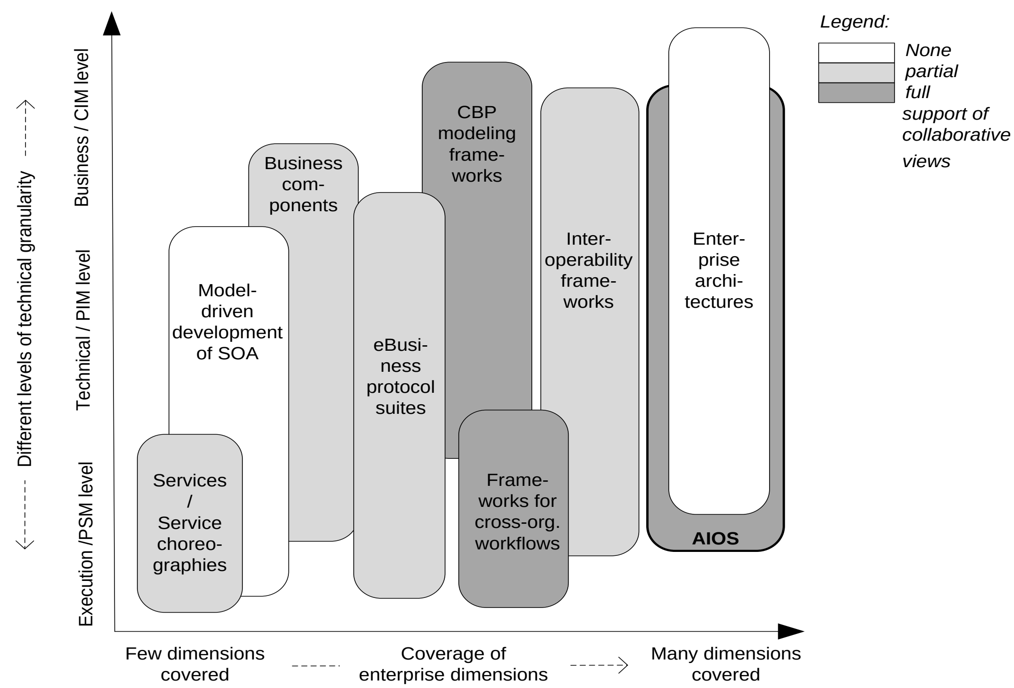


Figure 30: Position of the AIOS in relation to existing work

3.2.4.2 Coverage of Collaborative Views

Corresponding to their support of collaborative views, the different approaches can be grouped in three categories (Figure 30 illustrates the different groups in black, gray and white):

- Support of private, public and global views: In this group, all three collaborative views are supported explicitly, as in the case of the *CBP modeling frameworks* and *frameworks for cross-organizational workflows.*
- Partial support of collaborative views: Though in SOA concepts for all three viewpoints exist (orchestration, choreography interface and choreography), not all SOA development methods support these. However, since interface-orientation is a main paradigm of SOA, the separation between private and public views is covered by *service modeling* and *business components*; both describe the interface of a private system. Though the EIF describes the need for a Business Interoperability Interface, the investigated *eGovernment interoperability frameworks* did not describe concepts explicitly supporting the creation of private, public or global views; in the Business Interoperability Framework, public processes are supported. *eBusiness protocol suites* focus on global processes, while corresponding public and private processes are not described explicitly.
- No collaborative views: *Model-driven Architecture* approaches are usually not CBP specific and do not support concepts for private, public and global views; likewise, in the SOA development methods no or weak support for collaborative views was present. The same goes for *enterprise architectures*, which concentrate on intra-organizational processes.

Thus, it can be observed that only CBP modeling frameworks and the frameworks for cross-organizational workflows support all three types of collaborative views; but these frameworks are incomplete regarding their coverage of MDA-levels and their coverage of enterprise dimensions.

3.2.4.3 Position of the AIOS

Figure 30 illustrates that in relation to existing approaches, AIOS is situated close to enterprise architectures: Like these, the AIOS aims at a comprehensive coverage of enterprise dimensions and a business-driven development of executable processes. However, different from high-level enterprise architectures, the AIOS does not take into account strategic aspects; on the other hand, the AIOS is closer related to execution technologies than strategy-oriented enterprise architectures. A major difference between enterprise architectures and the AIOS is that only the latter supports interoperability by offering collaborative views on business processes.

	ARIS	ZACHMAN	ATHENA CBP	ATHENA SOA	INTEROP/MDI	CHIU	IDEAS	CIMOSA	TEAF	WIMMER CUBE
Enterprise views	●							●	●	
Collaborative views			●			●				
MDD levels	●		●	●	●			●		
Stakeholder perspectives		●			●				●	
SOA views					●					
Metamodel, model, instance					●					
Genericity/Specificity of model					●			●		●
Specific use cases										●
Semantic annotation				●			●			
Non-linear axes		●		●		●	●			●

● / Blank: Approach incorporates / does not incorporate axis

Table 2: Orthogonal axes covered by the different approaches

3.2.4.4 Orthogonal Axes Used in the Approaches

The literature review has also shown, that approaches exist that use orthogonal axes similar to those proposed for the AIOS. Thus, in Figure 31 four approaches are illustrated that support enterprise dimensions (ARIS, CHIU ET AL.), different levels of technical granularity (INTEROP, ATHENA, ARIS) and public views on internal processes (CHIU ET AL.). Table 2 shows an evaluation of those frameworks described above being based on orthogonal axes. The table illustrates that a surprisingly high quantity of different axes exist. In addition to the AIOS axes (enterprise dimensions, MDD-levels, collaborative views) the following axes were used:

- Stakeholder perspective: In the axis *stakeholder perspective* roles like “process designer” or “programmer” are used to describe a modeling level required by these roles. Thus, this axis is closely related to the *technical granularity/MDD* axis, since both describe different stages of the software development lifecycle to support a systematic software development.

- SOA views, as used in the INTEROP/MDI framework, are related to enterprise views, but describe only views that are often used by SOA standards (e.g. processes, functions and data).
- Modeling level: Other frameworks distinguish between the modeling level, e.g. if models on the instance, the type or the metamodel level is tackled.
- Genericity/specificity: Also tackling the abstraction level, a *genericity/specificity* axis is used by the CIMOSA framework, designating the scope in which the models can be used. For example, generic building blocks can be used in different industries, while certain process models might be usable only in one industry or only in one specific enterprise. In a similar vein, the WIMMER framework distinguishes between local, national and international scope of the described models.
- Use case: The WIMMER framework contains an axis where each unit represents a different *use case*, like for example "ePayment" or "eIdentification".
- Semantic annotation: A number of frameworks contain an axis that distinguishes between models and *semantically annotated* models.

Some frameworks contain axes that cannot be combined with other orthogonal axes, because the units on the axis are non-linear, i.e. they combine aspects comprised in several other axes. For example, the workflow views described by CHIU ET AL. comprise generic process views like control flow and data flow (as known from the enterprise dimension), but they also comprise a "semantic view".[441] The latter should be standing orthogonally to the previous dimensions, since semantic annotations are used to describe elements from the data and process view.

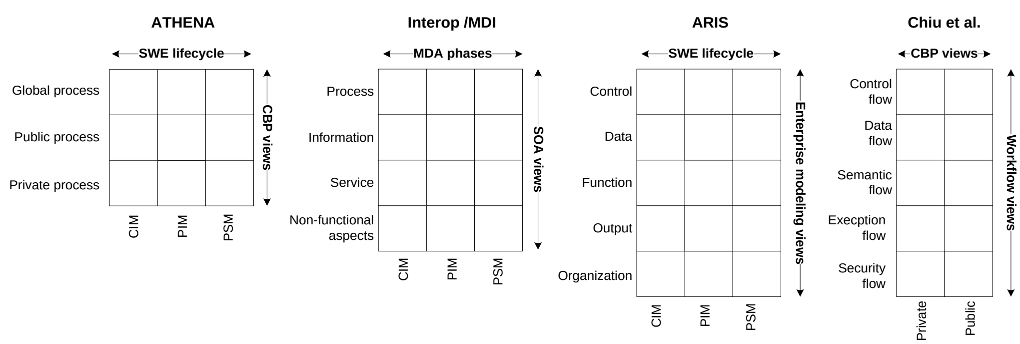


Figure 31: Examples of orthogonal axes used in frameworks related to the AIOS

On the Possibility to Replace or Complement Axes of the AIOS The literature review has shown that other axes would be available to constitute the AIOS or to extend the existing choice of axis (which is displayed in gray in Table 2). However, none of the other axes is as suitable for the fulfillment of the AIOS goals as the current axes are; SOA views, for example, have a too narrow

[441] Compare CHIU ET AL. (2005). Other frameworks that comprise such non-linear axes are: The ZACHMAN framework, comprising a dimension that is described by the questions what, how, where, who, when and why; the ATHENA SOA framework, possessing an axis that comprises the units enterprise/business, processes, services and information/data; the IDEA framework, with an axis that comprises the units business, knowledge, and IT systems; and the WIMMER framework, which incorporates the EIF levels organization, semantic and technique.

view; they concentrate on technical process dimensions and neglect business requirement modeling dimensions, thus providing an incomplete picture of collaborative business processes. The other axes (model/metamodel, genericity/specificity, use case and semantic annotation) are not as essential for a model-based development of collaborative business processes as the current axes are; however, they could be used to complement the AIOS. For example, AIOS elements could be described for a selection of use cases. Nevertheless, in order to restrain its complexity, no further axis is added to the AIOS.

3.3 Consolidation of Axes

The previous section showed that none of the existing approaches supports all three AIOS axes – hence, the gap analysis and the corresponding choice of the AIOS axes, as described at the beginning of this chapter, was confirmed. However, since only the nature of the axes but not their concrete instantiation was tackled by the review, in the following, the definition of the axes is refined by specifying the units that constitute each axis.

3.3.1 Specifying the Enterprise Dimension Axis

In this section, the choice of enterprise dimensions to be supported by the AIOS is determined. Therefore, different approaches for describing enterprise dimensions are shortly reviewed; afterwards, a set of enterprise dimensions for the AIOS is defined.

3.3.1.1 Preliminaries

On the Applicability of (Intra-Organizational) Enterprise Dimensions for Collaborative Business The literature review above indicated that current approaches for developing interoperable information systems focus on the process, the functional and the data dimension.[442] However, as sketched out previously,[443] a central assumption of this thesis is that those *enterprise dimensions needed for the description of intra-organizational business processes are also needed for the description of inter-organizational business processes*. In the past decades, when techniques for process automation and digital networks were less developed, cross-organizational activities were reduced to essential elements (e.g. business documents) while the processes surrounding the document exchange had to be manual. Today, network-based concepts like SOA are also used intra-organizationally and the gap between intra- and inter-organizational processes is decreasing. Instead of covering fewer aspects, due to higher security concerns and a lower degree of implicit, shared understanding, inter-organizational process models rather should describe more aspects than models of intra-organizational processes do.

Different Methods to Form Enterprise Dimensions To reduce their complexity, enterprise models are divided into different dimensions, also called views. SCHEER defines enterprise views based on the criterion *semantic similarity*, meaning that model elements with similar characteristics

[442] For example, eBusiness protocols like UN/EDIFACT concentrate on defining widely accepted business documents, collaborative workflows as well as collaborative business process frameworks focus on the process dimension, and web services or similar component-based approaches facilitate functions in collaborative business processes.

[443] Compare pp. 2 and p. 20.

are grouped into one view.[444] The dissection of processes into a control flow and a data view is another example for building views on the criterion of semantic similarity.[445] Another way of dividing process models is their disassembly into subsystems using the criterion of *position in process*. Thus, a process can be divided into various sub-processes, but in each subsystem, all elements of the upper level process could be used. Splitting a process model into semantically similar views has the advantage of avoiding redundancies, which could occur when objects in a process model are used more than once. For example, the same document type, event or organization unit might be applied to several functions.[446]

Business Process-Centric Views The ARIS approach from SCHEER focuses on business processes, thus, the ARIS views are centered on the output producing business functions,[447] and the process dimension is used to integrate other enterprise dimensions.[448] Other systems for splitting up IT systems into different views do not follow this business process-centric approach; Object-oriented approaches, for example, focus on objects instead of functions and do not offer genuine views on organizational and output elements.[449]

Requirements on the AIOS Enterprise Dimensions Foremost, the set of enterprise dimensions in the AIOS has to support a comprehensive business requirements modeling, covering all aspects relevant for internal process as well as the cross-organizational activities connected to internal processes. For this purpose, the enterprise dimensions should be complementary and disjoint; they also should be positioned on one level of technical granularity, e.g. the business/PIM level. However, to support a model-driven development and a description of interoperable systems on various levels of technical detail, the enterprise dimension on the business level should have counterparts on the technical/PIM and the execution/PSM level.

3.3.1.2 Using the ARIS Enterprise Dimensions as Reference

Due to the different objectives of modeling and different ways of structuring enterprise models, a great variety of metamodels exists, representing different sets of enterprise dimensions.[450] In order to create a reference point for a comparison with further approaches, here in a first step, the dimensions used in ARIS are assumed suitable for the enterprise dimension axis of the AIOS. Later, these dimensions will be compared with other approaches.

The ARIS House (illustrated in Figure 32) represents a concept to model the different aspects of an enterprise necessary for the development of an integrated information system. In ARIS, enter-

444 Compare SCHEER (1999), p. 34. For example, all model elements that represent functions are allocated in the function view.

445 Though this distinction is useful, for example, in the verification of control flows in Petri Nets, these two views are insufficient to display all enterprise elements relevant on the business level.

446 Compare SCHEER (1999), p. 33. Note that both ways of abstractions in practice are often combined, e.g. EPC models distinguish various views (data, process, organization etc.) but can also contain sub-processes.

447 This is also illustrated in Figure 32.

448 Compare SCHEER (1999).

449 Compare GADATSCH (2008), p. 80 and SCHEER (1999), p. 146.

450 Compare for example GADATSCH (2008), p. 78.

prise dimensions are created according to the criterion of semantic correlation, resulting in the following dimensions:[451]

Function Dimension In this dimension, the activities comprised in a process are described as well as the goals related to a function. While in comparable approaches functions and the IT resource executing the function are described in different views,[452] the ARIS function dimension comprises also the software applications responsible for the execution of a function.

Organization Dimension This dimension describes the hierarchical organization structure and its elements. Organizational entities are created in order to represent groups of humans or devices which are responsible for the same work object.

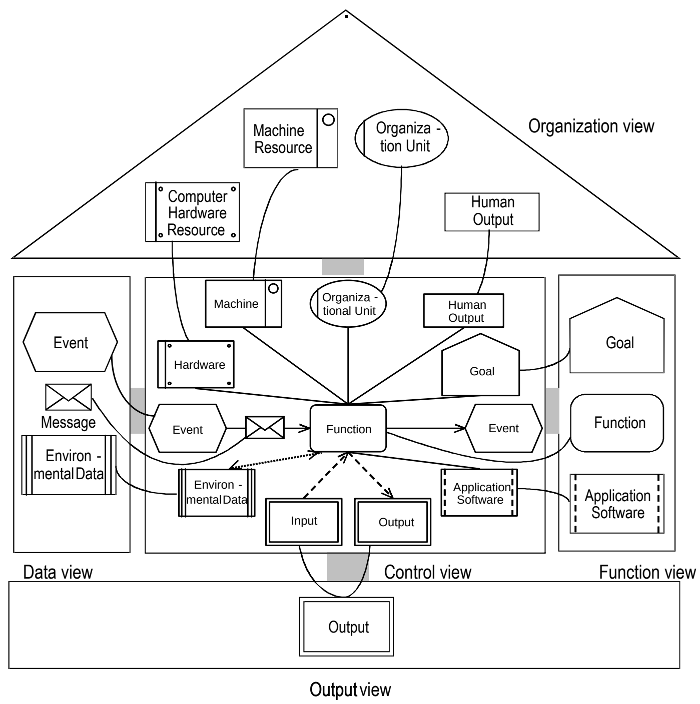


Figure 32: The five enterprise dimensions of ARIS and comprised elements[453]

Output Dimension Here all physical and non-physical input and output of functions is described. Interestingly, the early versions of ARIS comprise no distinct output dimension;[454] it was added, after experiences in the appliance of ARIS suggested its need. While the other ARIS dimensions are represented on the business requirements, the technical and the execution level, the ARIS output dimension concentrates on the requirements level.[455] Thus, it is used

451 Compare SCHEER (1999), p. 36. Instead of "enterprise dimension", SCHEER uses the term "ARIS view".
452 For example in the approach of LEYMANN & ROLLER (2000), processes and the process-executing IT resources are displayed in two different dimensions.
453 SCHEER (1999), p. 37.
454 Compare for example SCHEER (1994), p. 16.
455 Compare SCHEER (2000), p. 93.

mainly in the design phase, while on the operation level the elements of the output dimension are represented as data objects comprised in the data dimension.

Data Dimension This dimension comprises the data flowing through the process, e.g. messages triggering functions or being produced by functions. Three data types are distinguished: 1. Data constituting the control flow and 2. Environmental data, which has no direct influence on the control flow. The third category is called information services objects and comprises information, which serves as input or output of a function; this kind of data is allocated in both the output and the data dimension.[456]

Control/Process Dimension In this dimension the elements of the previous dimensions are related to form a business process. While the other four ARIS dimensions describe the static aspects of a process, the control dimension describes the process dynamically. Thus, functions and related elements are concatenated into sequences of business functions.

3.3.1.3 Comparable Sets of Enterprise Dimensions

GADATSCH compared different systems of enterprise modeling dimensions and defined five dimensions on his own: *Process, organization, application structure*, *activity structure*, and *information structure*.[457] The process, organization and information dimensions correspond to the ARIS control, organization and data dimensions; the application structure and activity structure dimensions are comprised in the ARIS function dimension.

BECKER, ALGERMISSEN & FALK described the PICTURE modeling method, which uses four dimensions: *Organization dimension* ("who executes?"), *resource* dimension ("which means are used to execute?"), *business object* dimension ("what is being produced/transformed?") and *process dimension* ("how is something executed?").[458] Compared to ARIS, they focus rather on the description of individual process building blocks and neglect the sequence between different processes. Thus, their process dimension comprises both the functional and the process dimension of ARIS. They do not explicitly cover the ARIS data dimension; nonetheless, since in CBP data can be seen as the most important object being transformed, their "business object dimension" covers parts of the ARIS data dimension. Their resource dimension is covered by ARIS' organization dimension (human resources) and function dimension, since the latter describes also which IT resources are executing a function.

AMFIBIA Aiming at the formalization of essential concepts of business process modeling, KINDLER, AXENATH & RUBIN described a metamodel for business processes called AMFIBIA, based on four dimensions:[459] The *control dimension* describes the order in which functions are executed. The *organization aspect* describes the organization structure, e.g. actors and the way they are involved in the business process. The *information aspect* describes how data is represented in a business process, and how it is propagated among the functions. The fourth dimension, referred to

---

456 The differences and intersections of the data and the output dimension are described in more detail on pp. 175.
457 Compare GADATSCH (2008), pp. 78.
458 Compare BECKER, ALGERMISSEN & FALK (2007), pp. 87.
459 Compare KINDLER, AXENATH & RUBIN (2006) and AXENATH, KINDLER & RUBIN (2005). Instead of "dimension", they use the term "aspect".

as the *core of the business process*, describes individual functions and at the same time is used to integrate the previous dimensions into a business process. The authors explicitly omit the goal modeling comprised in the ARIS function dimension, arguing that their approach aims only at enactment of processes, where goal modeling is not needed. In addition to these four constituting dimensions, three further dimensions are described: *Assignment*, where the correlation of tasks to organization elements is described, *Authorization*, where the rights of organization elements to access information items are described, and *Transaction*, where allowed sequences of functions and corresponding modifications of information items are described. Though this choice of separation as well as their explicitly loose coupling is suitable for the enactment of processes, the design view needed in the AIOS is neglected.

LEYMANN & ROLLER distinguished three orthogonal dimensions of workflows: *Process* logic ("what dimension"), *organization* ("who dimension") and *IT infrastructure* ("which dimension"). The first dimension describes activities and their sequences; the second dimension describes the organization dimension and correlates activities with departments, roles and persons.[460] The IT dimension describes which IT resources are used in the workflow. While the first two dimensions correspond to the ARIS dimensions of the same name, the IT infrastructure dimension is covered by the function dimension of ARIS.

CHIU ET AL. developed different workflow views to describe interaction between large-scale workflow systems comprising *data* flow, *control* flow, *semantic* flow, *exception* flow and *security* flow. [461] These views are closely related to existing web service standards, e.g. the semantic layer is specified in OWL, the security layer using WS-Security and Security Assertion Markup Language and the control flow in BPEL. Different from the ARIS dimensions, their views are orthogonal, e.g. data and control flow describe what kind of elements a process contains while the semantic view explains how such elements are described. The semantic view is the only view that cannot be mapped to ARIS.

LEGNER ET AL. described an approach for transforming inter-organizational business processes into Service-oriented Architectures. For this purpose, they defined a metamodel for modeling inter-organizational business processes consisting of four dimensions: The *organization* dimension corresponds roughly to the ARIS dimension of the same name, the *information objects* dimension is similar to ARIS' data dimension, though LEGNER ET AL. put a stronger emphasis on business documents. The *process dimension* incorporates elements from ARIS function and control dimension; additionally, private and public processes are distinguished. The *terminology* dimension matches the syntax of elements used in a collaboration to a shared semantic. In difference to ARIS, no output dimension is defined.[462]

CAMARINHA-MATOS & AFSARMANESH developed a modeling framework for "collaborative networked organizations".[463] The framework is based on four modeling dimensions: in the (misleadingly named) *structural* dimension, similar to ARIS' organization dimension, actors that participate in a CBP are described. Their *componential* dimension represents static elements like resources

[460] LEYMANN & ROLLER (2000).
[461] Compare CHIU ET AL. (2005).
[462] Compare LEGNER ET AL. (2007).
[463] Compare CAMARINHA-MATOS & AFSARMANESH (2007).

and information; thus, it also covers elements described in the data dimension of ARIS. Further, they described a *functional* and a *behavioral* dimension, resembling the function and process dimension of ARIS.

ZDRAVKOVIC & KABILAN described the obligations of collaboration partners using so-called contract workflow models, which cover aspects of business process models relevant for the collaboration. To this purpose, they distinguish five dimensions: function, behavior, information, organization and transaction. The behavior dimension corresponds to ARIS' process dimension. In addition to the dimensions known from ARIS, the transaction dimension describes the transaction types used in the collaboration, for example, the atomic transaction model or the long-running transaction model.[464]

BERNAUER ET AL. used seven workflow views to compare several B2B protocols. These are based on the workflow views described by RAUSCH-SCHOTT, though one dimension irrelevant for CBP is left out and an additional security view is added, which they motivate by the fact that CBP would require more security measures than intra-organizational processes.[465] The *security* view describes confidentiality, non-repudiation, integrity, authorization and authentication. The other views are *function*, *behavior* (corresponding to the ARIS control dimension), *data*, *organization* and *transaction*; where the transactional aspect describes how transactions can be used to ensure the consistent execution of various functions. Additionally, they described a "causal aspect" view, which defines "why a certain B2B protocol is specified in a certain way and why it is being executed".[466]

3.3.1.4 Conclusions

Table 3 illustrates the results of the review. In addition to the approaches described in this section, the table also illustrates the CIMOSA dimension, which was described in the section on enterprise architectures (p. 80). The column "means to execute functions" refers to resources related to the execution of functions that are not comprised in the organization dimension. The results confirm the assumption that current approaches for modeling inter-organizational business processes cover similar dimensions as those in intra-organizational approaches. Going beyond these dimensions, some of the inter-organizational approaches offered dimensions for describing security aspects, transactions and the semantics used in the collaboration.

Process, Function, Organization, Data and Security Dimension Despite the different backgrounds of the approaches, a nearly unanimous support of four core dimensions can be observed: Process, function, organization and data. Accordingly, these four views will be supported by the AIOS. Issues tackled usually in security dimensions, will be described in the organization dimension of the AIOS.

---

464 Compare ZDRAVKOVIC & KABILAN (2005).

465 Compare RAUSCH-SCHOTT (1997). The "historical aspect", which defines which data should be logged at which point in time, was left out because it is only relevant inside individual organizations but not cross-organizational; compare BERNAUER, KAPPEL & KRAMLER (2003), p. 229.

466 Compare BERNAUER, KAPPEL & KRAMLER (2003), p. 229. KRAMLER ET AL. (2003) describe a similar set of dimensions for inter-organizational workflows that – instead of a security and a causal aspect view – comprises an *operational* view (that describes how individual activities are implemented) and an *interaction view*, describing details of interaction activities.

Output Dimension Few approaches incorporated a distinct output dimension for the modeling of elements that are consumed or produced by business functions. Also in ARIS, this view is mainly used in the requirements modeling phase, while on the operational levels output elements are described in the data dimension. This thesis focuses on enterprise dimensions that are necessary to describe interoperable information systems on all vertical levels needed for process automation; since the output dimension is restrained to the business level, this thesis focuses on the other enterprise dimensions, and tackles the output dimension mostly implicitly in the data dimension. However, in order to stay consistent with ARIS – representing a wide spread and accepted framework for intra-organizational business process modeling – and to support an explicit, business-level modeling of service input and output, the output dimension is regarded as part of the AIOS.

	ARIS	CIMOSA	Gadatsch	Leymann et al.	PICTURE	AMFIBIA	Chiu et al.	Legner et al.	Camarihna-Matos et al.	Zdravkovic et al.	Bernauer et al.
Process	●	○	●	●	●	●	●	●	●	●	●
(Individual) function	●	●	●	○		○		○	●	●	●
Means to execute function	○	●	○	●	●						
Organization elements	●	●	●	●	●	●	○	●	●	●	●
Data elements	●	●	●		○	●	●	●	○	●	●
Input and output objects	●	●			●						
Security dimension						●	●				●
Transaction dimension						●	●			●	●
Semantic dimension							●	●			
CBP specificity							●	●	●	●	●

● / ○ / Blank: Explicit / implicit / no support for enterprise dimension

Table 3: Enterprise dimensions used to display intra- and inter-organizational processes

Transaction Dimension To ensure the consistency of the various information systems comprised in a collaboration, the explicit support of transaction mechanisms can be useful. The corresponding solution proposed in AMFIBIA to position a transaction dimension between the process, function and data dimensions appears to be an elegant solution. Nevertheless, the introduction of such a dimension in the AIOS would exaggerate the relevance of transaction mechanisms in comparison to the other dimensions constituting a business process. Already in ARIS database transactions are described; there, they are positioned in the technical level of the control/process dimension.[467] Nevertheless, due to the static, nested character of long-running business transactions, these

[467] Compare Scheer (2000), pp. 141.

could also be situated in the function dimension.[468] However, the integration of genuine mechanisms to support long-running, distributed transactions in the AIOS is left for future work.

Note, that dimensions tackling transactional integrity or security are also subsumed under the term *non-functional* characteristics. Non-functional features do not represent the core business functionality, but rather side aspects of it, e.g. the conditions under which these functionalities should be executed. Therefore, non-functional requirements are also called constraints or qualities.[469] However, since the distinction between "core functionalities" and "constraining functionalities" seems to be context dependent, the AIOS does not incorporate these categories.

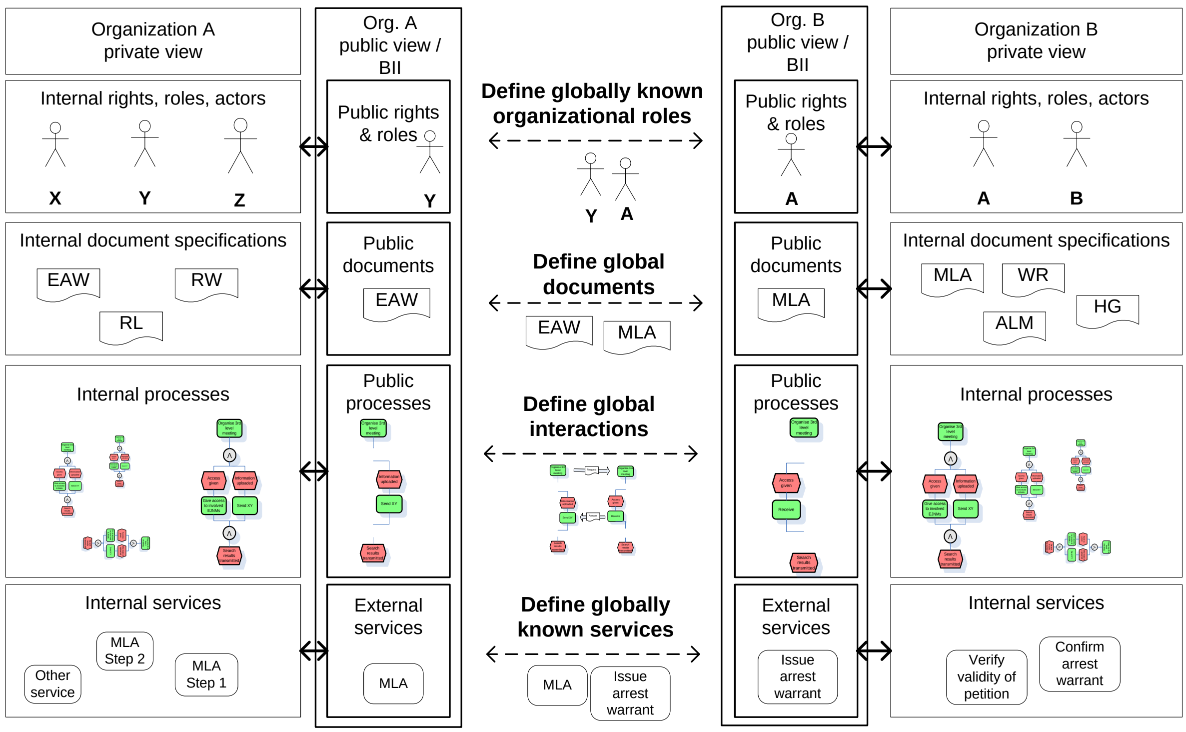


Figure 33: AIOS enterprise dimensions - organization, data/output, process and function

Semantic Dimension Several approaches include a semantic dimension, which is clearly useful to ensure a common understanding of concepts among collaboration partners. However, instead of describing enterprise elements to be modeled ("what has to be described") they describe the manner, in which all elements have to be modeled ("how does it have to be described"). Thus, semantics are not part of the AIOS enterprise dimensions. The support of a common understanding of concepts will be tackled in the collaborative views axis of the AIOS, which is orthogonal to the enterprise dimensions axis.

---

[468] In this vein, PAPAZOGLOU (2003), p. 52, defines business transactions as a "consistent change in the state of the business that is driven by a well-defined business function". In the context of SOA, long-running business transactions are implemented as open nested transactions (compare BEERI ET AL., 1989 and OASIS, 2007).

[469] Compare ABRAN & MOORE (2004), p. 2-2, or BIEBERSTEIN (2006), p. 123, who classifies non-functional requirements in four categories: business constraints, technology constraints, run time qualities and non-run time qualities.

To summarize: Though ARIS was developed before the rise of SOA and originally aims at intra-organizational processes, the enterprise dimensions used in ARIS comply with the requirements of modeling intra- and inter-organizational business processes. The ARIS enterprise dimensions can also be used in the development of Service-oriented Architectures.[470] Since with ARIS a practice proven,[471] coherent framework exists that describes in detail how the dimensions are related and how they can be transferred to technical levels, the AIOS enterprise dimensions will be based on ARIS. However, due to the focus on process automation, the output dimension receives less attention than the process, function, organization and data dimension.

Accordingly, Figure 33 illustrates that the AIOS enables collaboration partners to adjust their information systems along four enterprise dimensions: In the *organization dimension*, roles, units and other organization elements relevant for the collaboration are described and related to internal elements. This ensures for example, that the collaboration partners have a common understanding of the interacting roles. In the *data dimension*, document types used in the collaboration are defined and related to internally used document types. In the *function dimension*, services offered in the collaboration are described. In the *process dimension*, the processes that each organization offers are described as well as how these public processes are related to adjacent processes of partner organizations.

3.3.2 Specifying the Model-Driven Development Axis

In this section, the levels of different technical granularity to be supported in the AIOS are defined. The aim of this axis is it to support the model-driven development of processes and to describe interoperable information systems on different levels of technical granularity, ensuring the compliancy of the executed processes with conceptual specifications and supporting controlling functionalities.

On the Suitability of Process Automation Levels to Describe Interoperable Systems A foundational assumption of the AIOS is, that the different levels of technical granularity used in the model-driven development of systems (or more specifically: in the automation of business processes) are suitable to describe interoperable systems. This assumption is based on the close relationship between cross-organizational business processes and interoperability: if enterprises are able to enact collaborative business processes among themselves, it is implied that the information systems of the enterprises are interoperable.[472] In consequence, the different development stages of collaborative business processes (e.g. business design, technical design and implementation) can be used to describe different aspects of interoperable systems, e.g. interoperability on the business, the technical and the execution level.

3.3.2.1 Proposal of Three Vertical Levels

Apart from technical software engineering approaches like MDA, a model-based development of information systems is also supported by business-oriented approaches, for example, Business-IT

---

[470] Compare for example ZIEMANN ET AL. (2006).

[471] ARIS is one of the few enterprise modeling methodologies proven applicable in practice, both in the private sector and in public administrations. Compare for example SCHEER, KRUPPKE & HEIB (2003) or SCHEER & JOST (2002).

[472] Compare also Chapter 2, p. 26.

alignment and (collaborative) business process management.[473] To provide a basis for the selection of the corresponding AIOS levels, in the following, different vertical levels supported by approaches from the areas of enterprise modeling, MDA and collaborative business are compared.

ARIS To enable a business-driven development of information systems with ARIS, SCHEER distinguished the following five phases:[474]

1. IT strategy: In this phase, the IT-oriented initial strategy is formulated. A strategy determines long-term corporate goals, enterprise activities and resources. The IT-oriented strategy of an enterprise describes and plans the long-term use of IT, taking into account restrictions and possibilities of current technologies. Examples for such IT strategies would be the implementation of SOA to create loosely coupled enterprise networks, or the reliance of portal systems to improve customer relations. At this level, only coarse-grained processes are depicted, using for example value chain diagrams.
2. Requirements definition: At this level, the individual business processes are modeled in detail and divided into the ARIS views to describe the content of the planned application system. Models here should be sufficiently formal to be a suitable starting point for an IT implementation, yet be understandable from a business point of view. General IT objects, such as databases or programs, are also described at this level.
3. Design specification: Here the business models are adapted to the detailed requirements of the IT concepts. For example, Entity Relationship models are used to describe the detailed content of a database, or a process is refined in such a way that it could be executed as a workflow. Nevertheless, since this is still a platform independent modeling level and not an implementation level, the workflow would not be specified as code.
4. Implementation description: In this phase, the elements described in the design specification are implemented in physical data structures, hardware components and deployed to execution platforms. For example, programming languages like C++ or Java are used here to describe information systems on the code level.[475]
5. Operation and maintenance: Since these four phases cover only build time aspects, this phase covers the run time aspects.

ARIS focuses on the levels of requirements definition and design specification, while the implementation description is of less detail; IT strategy and the run time phases are described only scarcely. Though the levels 2, 3, and 4 are described for all ARIS enterprise dimensions, no transformation method between the vertical levels is described. In this context, SCHEER remarks that the five phases do not imply a rigid sequence in the development process as in waterfall models, but rather an evolutionary prototyping procedure.[476]

Model-Driven Architecture As described in Chapter 2, MDA is a well-known framework for model-driven software development; it focuses rather on technical aspects and less on business re-

[473] Compare also Chapter 2, pp. 55.
[474] Compare SCHEER (1999), pp. 38.
[475] Compare also SCHEER (2000), p. 50.
[476] Compare SCHEER (1999), p. 40.

quirements modeling.[477] The fact, that an automated transformation between levels is only foreseen among the PIM and PSM level models illustrates the proximity between both levels. The transformation of the CIM to the PIM on the other hand requires human judgment, thus, here no automated transformation is foreseen.[478]

CBP Modeling Frameworks MDA aims at software development in general but does not take into account peculiarities of cross-organizational processes. Since MDA and SOA concepts reached a certain maturity, various authors took the logical step of combining both to create approaches for model-driven development of interoperable systems: following the rationality that cross-organizational business processes are just a special case of business processes, the concept of dividing a process into different vertical layers is transferred to cross-organizational processes. In consequence, the approach of ELVESÆTER ET AL.[479] follows the OMG levels to obtain interoperability on three different levels: CIM, PIM and PSM level processes are designed and implemented. In the ATHENA framework for designing and implementing cross-organizational business processes, also three levels of technical detail were used:[480] the business level, technical level and execution level. *Business level* models describe organizational aspects as a prerequisite for the technical integration of IT systems or their configuration. Accordingly, models on this level allow for analyzing business aspects, for example, costs or involved resources. The *technical level* provides a more detailed view on the CBP though it remains platform independent, and thus enables the (re)use of the models in different execution environments. On the *execution level* models are platform specific, e.g. machine interpretable and executable by business process engines.

Protocol Layers and Web Service Stack To divide the different tasks necessary for a successful communication, it is useful to segregate them into different layers. Apart from a reduced design complexity, another advantage is the reduction of dependencies and increase of stability: Especially for large and complex systems that are constantly being adapted, the ability to restrict changes to a delimited area of the system is important. In protocol layers, each layer provides a collection of related functions to the layer above it, meaning that the layer itself can receive services from the layer below it. Following the protocol layering concept, different protocol stacks were created, one well-known being the Open Systems Interconnection (OSI) Basic Reference Model, which comprises seven layers of protocols to enable communications in computer networks.[481] Today, the most often-used protocol stack is probably the internet protocol stack, which, like the OSI reference model, has the application layer on top, and from there refines the layers technically to the bottom, the physical layer.[482] Similar to protocol layers, standards used in SOA are supposed to build on each other or at least to complement each other. In this vein, a variety of different web service stacks emerged to display various SOA standards and their relationships.[483] Due to the central role of networks and the internet, these were inspired by the layering of network protocols. However, in

---

477 Compare OMG (2008) or GRUHN, PIEPER & RÖTTGERS (2006).
478 Compare OMG (2003).
479 ELVESÆTER ET AL. (2005).
480 Compare GREINER ET AL. (2006) or BORN ET AL. (2009).
481 Compare ISO (1994). The seven layers are (from top to bottom) the Application, Presentation, Session, Transport, Network, Data Link, and Physical layer.
482 Compare KUROSE & ROSS (2002), p. 55.
483 Characteristics of the web service stack and the bottom-up creation of its standards were already described in Chapter 1, p. 5.

difference to protocol layering, where a clear top-down relationship among the standards exists, no clear principle for forming web service stack exists, and levels of technical granularity are mixed with enterprise/business process dimensions. For example, in web service stacks, services are usually positioned below the process level, displaying the fact that a service can be a part of a service composition; nevertheless, the service description might be of the same technical granularity as the process description. Moreover, a couple of authors introduced categories orthogonally to the vertical layers of the web service stack. For example, the discovery standard UDDI, which does not primarily describe web services but serves as the basis for a discovery service.

In consequence, many different web service stacks exist in parallel, stemming from research, industry and standardization organizations.[484] Though the AIOS aims at an implementation of processes with standards described in the web service stack (e.g. WSDL and BPEL), the stack itself is not suitable for a model-driven development: it is clearly technology focused, providing standards for the execution of service-based processes, neglects business level standards and thus does not support a systematic refinement of process models.

Conclusions Though coming from different areas (enterprise modeling, technical software development and collaborative business), the approaches described above support the same three vertical levels: On the first level, the processes to be automated are described from a technique independent level. This level will be referred to as the *business* or *CIM level*. On the second level, the IT concept is described. Therefore, the models from the first level are technically enriched, for example, instead of business functions now components are described, but still on a coarse-grained, conceptual level. Since the models on the second level represent the basis for an automated generation of executable code, they might have to be further adapted to fit implementation level constraints.[485] The second level will be referred to as the *technical* or PIM level. On the third level, the models are machine interpretable and can used during run time in the execution of processes. This level will be referred to as the *execution* or PSM level.

AIOS	ARIS	MDA	ATHENA	Example standards
Business level	Requirements definition	CIM	Business level	EPC
Technical level	Design specification	PIM	Technical level	BPMN, UML
Execution level / code	Implementation description	PSM	Execution level	BPEL, WSDL

Table 4: Three vertical levels used in the AIOS and related approaches

These three levels and their counterparts in related approaches are displayed in Table 4. Note that the last level can be further split up into logical and physical execution level models:[486] Logical models describe processes and components on the execution level (e.g. BPEL or WSDL) independently of the physical instantiation of services and processes. Physical models enrich the abstract models by correlating them with physical services that instantiate the functions and processes.[487] In the AIOS, only the logical level is tackled.

---

484 Compare for example VAN DER AALST (2003), FERGUSON ET AL. (2003) and O'RIORDAN (2002).

485 For example, ambiguous logical operators like the "OR" operator might have to be replaced by operators that are interpretable on the execution level, for example "XOR" or "AND" operators.

486 Compare also NEWCOMER & LOMOW (2005).

487 This resembles the difference of WSDL or BPEL models and their deployment into executable services/processes.

3.3.2.2 Reconfirmation of Vertical Levels in Process Automation Approaches

Besides the description of interoperable information systems, a major requirement for the vertical AIOS levels is the support of process automation; more specifically, the development of SOA-based processes as implemented, for example, by BPEL. Therefore, in the following, the applicability of the vertical levels proposed above is validated by reviewing recent approaches for a business-driven development of BPEL processes.

ZIEMANN & MENDLING described how the elements of the EPC can be mapped to BPEL elements. They also described the reverse direction, e.g. the visualization of BPEL models with EPC.[488] Focusing more strongly on the procedural aspect, KOPP, UNGER & LEYMANN developed a detailed algorithm for the generation of BPEL from EPC.[489] STEIN & IVANOV described a procedure model to transform EPC to BPEL based on workflow patterns. Interestingly, they only use BPEL's block-oriented structured activities (which is surprising, since the EPC is graph-oriented and BPEL offers graph-oriented elements as well). Their transformation approach is implemented in the ARIS toolset, which also offers a function to verify the correctness of the models to be transformed to BPEL.[490] HUTH & WIELAND propose to extend the EPC for usage in Service-oriented Architectures, which should also support a transformation of EPC to BPEL.[491] Therefore, a number of new symbols were introduced that specialize existing EPC elements like functions (e.g. "synchronous service" and "asynchronous service") and events (e.g. "exception event" and "timeout event"), which makes modeling unnecessarily complex, since the respective elements can be easily displayed without altering existing EPC symbols.[492] SPECHT ET AL. also resort to VAN DER AALST'S workflow patterns for a transformation from EPC to BPEL. In addition, they argue that the annotation of IT systems and documents involved in the workflow would overload the EPC model. To separate the EPC from details seemingly needed only in BPEL, they introduced so-called extended function allocation diagrams. These diagrams "describe the exact sequence of activities and operations and transmitted data"[493] assumedly needed in BPEL.

The previously mentioned authors used EPC models as the source of a direct transformation to BPEL, while other authors use BPMN as an intermediary format. Thus, ALLWEYER described a transformation chain based on EPC, BPMN and BPEL (though BPEL is tackled only implicitly, by producing BPMN models that are directly transformable to BPEL).[494] THOMAS, LEYKING & DREIFUS proposed a method for EPC-based BPEL generation as well, using BPMN as an intermediary format for the configuration of the conceptual EPC models.[495] Obviously, a direct transformation from EPC to BPEL has the advantage that no additional intermediary transformations have to be executed, avoiding the risk of transformation errors and a reduction of traceability. However, it

---

488 Compare ZIEMANN & MENDLING (2005), the bottom-up direction was described in MENDLING & ZIEMANN (2005).

489 Compare KOPP, UNGER & LEYMANN (2006). The transformation is based on a deviation of the EPC called N-EPC, the main difference between both formats being that in the N-EPC functions and events do not have to alternate.

490 The ARIS toolset is an industrial tool for modeling and verifying business processes. Compare STEIN & IVANOV (2007).

491 Compare HUTH & WIELAND (2007).

492 Corresponding EPC models were described in ZIEMANN & MENDLING (2005).

493 Compare SPECHT ET AL. (2005).

494 Compare ALLWEYER (2007).

495 Compare THOMAS, LEYKING & DREIFUS (2008).

might be argued that the EPC is not as suited to describe required technical details as, for example, BPMN would be and therefore should not be the basis of a direct transformation.

	Ziemann & Mendling	Specht et al.	Kopp et al.	Stein & Ivanov	Ziemann & Loos	Vanderhaeghen	Huth & Wieland	Thomas et al.	Allweyer	Hoyer et al.	Andres	Filograna et al.	Palmer	Ouyang et al.	Yu et al.	Bruno et. al	Hofreiter & Huemer
EPC	●	●	●	●	●	●	●	●	●	●	●						
Technical EPC	●	●		●	●		●										
BPMN						●	●	●	●	●		●	●	●			
UML											●				●	●	●
XPDL												●	●				
BPEL	●	●	●	●	●		●	●					●	●	●	●	●
Collaborative focus					●	●				●						●	●

● / Blank: Approach explicitly supports / does not support process standard or collaborations

Table 5: Process formats covered in EPC-to-BPEL transformation chains

Table 5 shows the transformation chains mentioned above as well as related transformation chains that represent a model-driven BPEL development. Ouyang et al., for example, described a detailed algorithm for transforming BPMN into BPEL.[496] Other authors rely on UML instead of BPMN. Thus, Andres described an EPC-based process automation where UML class diagrams are derived from EPC.[497] Yu et al. also proposed a transformation from UML to BPEL, using the EDOC UML profile as a basis.[498] Some authors also include XPDL in the chains for process automation.[499] Palmer sees XPDL as a means to store BPMN and accordingly proposes a chain of BPMN-XPDL-BPEL. Filograna et al. described a mapping between BPMN and XPDL as well a tool to execute a BPMN to XPDL transformation;[500] however, it remains unclear, to what length they see XPDL as an execution language or just an intermediate storage format.

Transformation Chains Focusing on Collaborations Some authors also described a vertical transformation explicitly aiming at the development of CBP. This includes Hoyer, Bucherer & Schnabel, whose approach supports private and public views, though they only tackled a transformation between EPC and BPMN.[501] Ziemann & Loos already proposed a transformation chain comprising EPC, technical EPC and BPEL taking into account different collaborative views.[502]

---

496 Compare Ouyang et al. (2006).
497 Compare Andres (2006).
498 Compare Yu et al. (2007). The “Enterprise Distributed Object Computing” (EDOC) standard was created by the OMG to support distributed computing based on MDA and SOA; the core of the EDOC is the Enterprise Component Architecture, abbreviated as ECA (compare OMG, 2004, p. 1-2). The ECA is also described in this thesis, see p. 89.
499 The XML Process Definition Language (XPDL) is a standard from the WfMC to exchange workflow definitions between different workflow engines. Compare WfMC (2008).
500 Compare Filograna et al. (2007).
501 Compare Hoyer, Bucherer & Schnabel (2008).
502 Compare Ziemann & Loos (2008).

BRUNO & LA ROSA described a transformation of “collaboration models” from UML Activity Diagrams to BPEL; however, they only tackled global models in the form of orchestrations. VANDERHAEGHEN, ZANG & SCHEER described a detailed transformation from EPC to BPMN.[503] Aiming at the development of CBP, they also proposed the usage of public and private process models, though they only shortly described the usage of these collaborative views. HOFREITER & HUEMER as well proposed an UML-based BPEL transformation. Therefore, they used the UML profile of the UN/CEFACT Modeling Methodology (UMM) to model global processes.[504] In a second step, they derived the private process comprised in the global processes and transformed these into BPEL Executable Processes.

Summary In Table 4, the three vertical levels (business, technical and execution level) are mapped to the standards visible in Table 5: EPC, BPMN, UML and BPEL.[505] This mapping shows that in the context of SOA many approaches exist, that are based on the three vertical AIOS levels. Interestingly, these levels were equally used in approaches for intra- and inter-organizational business processes. While some approaches tackled only two levels (e.g. a transformation from technical to execution level), other approaches used all three levels (e.g. business, technical and execution level).

3.3.2.3 Conclusions

The review showed that the three vertical levels proposed for the AIOS are also used in current approaches for a (business) model-driven generation of Service-based processes:

- Business/CIM level: Due to its suitability to capture business-level requirements, many approaches use the EPC as a starting point for BPEL generation.
- Technical/PIM level: To provide a basis for an automated generation, the business-level EPC models were converted into “technical” EPC models. Therefore, BPEL specific elements must be added (e.g. web services responsible for the execution of an EPC function) and the EPC models were adapted to be compliant with the semantics of BPEL (e.g. by transforming logical operators not supported by BPEL). On this level, also BPMN, UML and XPDL were used to model technical processes.
- Execution/PSM level: Based on the technical process models, BPEL models were derived to represent web service-based processes.

Thus, these three levels are chosen to represent the MDD axis of the AIOS. Figure 34 illustrates how the levels of technical granularity are used both for describing the interfaces of organizations as well as for developing CBP. The horizontal arrows illustrate the interlinkage of private, public and global models,[506] while the vertical arrows represent transformations between different levels of

[503] Compare VANDERHAEGHEN, ZANG & SCHEER (2005).

[504] UMM was created by UN/CEFACT as a successor of the UN/EDIFACT standard, especially targeting small and medium sized enterprises. The objective of UMM is to capture the commitments made by business partners. Compare HOFREITER, HUEMER & KIM (2006), pp. 223.

[505] In addition to the standards comprised in Table 4, the Business Process Automation approaches also used XPDL and technically enriched EPC, which belong to the technical level.

[506] As explained in the following section, the principle known from the process dimension to create private, public and global views can be applied to other enterprise dimensions as well.

technical granularity. On the business level, the compatibility of the business processes offered by each organization is realized. For this purpose, internal and CBP are modeled by business analysts who ensure that the interfaces of the collaborating partners contain complementary business concepts. Afterwards, the business level complementarity has to be extended to the technical level. Here, the IT concepts of the collaborating organizations are synchronized with each other. Since the execution level models contain further information (e.g. the network addresses of the web services to be invoked), in a third step the models on the execution level have to be synchronized, as well.

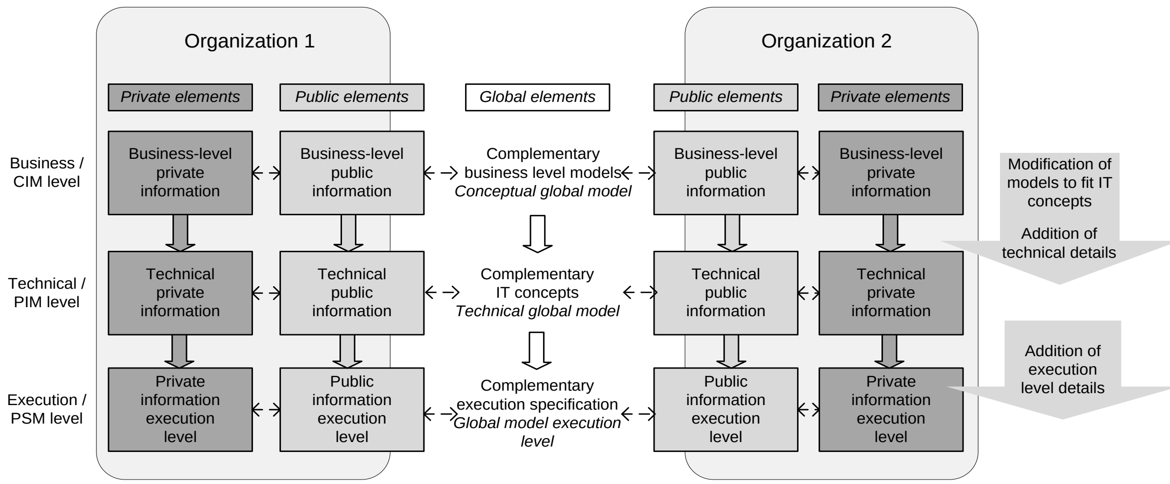


Figure 34: The three vertical levels supported by the AIOS

Note, that according to the MDA approach, models are transformed automatically only between the technical and the execution level, while the development of the technical from business-level models requires human judgment.[507] A similar relationship of the levels was described by SCHEER.[508] The reason for these differences is that the structure of the technical levels can differ substantially from the structure of the business-level models,[509] while the differences between technical and execution level models are rather syntactical. However, both model transformations (from CIM to PIM and from PIM to PSM) imply a technical refinement of the base model. And in order to generate a basis for the development of technical models, an automated transformation of CIM to PIM levels can be useful. Apart from avoiding errors in the manual re-modeling of CIM level concepts that appear on the PIM level, the automated generation and the explicit correlation of CIM with PIM level models supports monitoring and controlling functionalities, since the controlling of business processes requires that technical components are mapped to business functions. From an interoperability perspective, the description (and in consequence their explicit correlation) of models on all three vertical levels is also beneficial, since this multi-dimensional description increases

[507] Compare OMG (2003).
[508] Compare SCHEER (2000).
[509] On the technical level, it has to be decided in what form business level concepts are implemented as technical components. In the case of functions, for example, business level functions must not necessarily have a comparable counterpart in the IT concept. Thus, based on a set of business functions, a technical concept could be developed where one function is implemented as a standalone component, another function is implemented as an operation of an existing component or – if no IT support is feasible for this component– is not implemented at all.

the chance that the models can be understood and implemented correctly in inter-organizational scenarios.

3.3.3 Specifying the Collaborative View Axis

After the specification of the enterprise dimension and the MDD axis of the AIOS, in the following the collaborative view axis of the AIOS is described. In a first step, the general suitability of using private, public and global views outside the technical process dimension is discussed. Afterwards, the generalization of the collaborative views to the other dimensions is described.

3.3.3.1 On the Suitability of Collaborative Views for Different Vertical Levels

The concept of collaborative views described here stems from the workflow area, where public views on internal workflows were created. Later, this concept was used in SOA, resulting in the concepts of orchestration, choreography interface and choreography. Thus, on the execution level, all three collaborative views (private, public, and global) are used. Seeing the need for a conceptual preparation of these execution level process types, approaches were created to represent them on a business level, using for example EPC, to model private, public, and global business processes.[510] Similar to collaborative business processes, also the other enterprise dimensions need to be systematically developed and thus be described on different levels of technical granularity.

3.3.3.2 On the Suitability of Collaborative Views for the Enterprise Dimensions

Above, the need to use different enterprise dimensions not only in intra-organizational settings but also in the modeling of inter-organizational processes was described. However, the private-public-global concept until now focused on the control flow of processes, while other enterprise dimensions were neglected.

Collaborative Views in the ArKoS Project In ArKoS, for example, the public view is only supported for the process dimension.[511] Instead of differentiating inside each ARIS dimension between private, public and global knowledge, individual ARIS dimensions are completely declared as being either global or private; the only exception being the process dimension, where "process modules" serve as a connection between global and private viewpoints. There, it is argued that the *organization dimension* and the output dimension must be global, since "otherwise a purposeful collaboration is impossible"[512]. Though it is true that most collaborations require an explicit, global modeling of the involved organization units, it is unclear why internal organization units related to the globally visible ones should not also be represented in the form of local knowledge. The same argument applies to the *output dimension*: why should internal goals not be explicitly modeled and related to externally visible goals? Apart from that, it remains questionable if a CBP can only be "purposeful" if the output view is modeled explicitly.[513] The fact, that "*data interfaces*" are regarded

---

[510] Compare ZIEMANN, KAHL & WERTH (2007).

[511] The project was described above, see p. 82.

[512] HOFER ET AL. (2005), p. 5, translated from German.

[513] In many cases of cross-organizational Business Process Automation, the output view is modeled only implicitly as part of the function view, e.g. as an input/output parameter of a function. Nonetheless, though not being an imperative necessity, we agree that an explicit modeling of the output view in CBP can be beneficial. This applies especially in

as internal knowledge only, must also be seen critically: the exchange of messages represents the cornerstone of CBP automation, obviously the syntax and semantics of exchanged documents has to be described and agreed upon between the collaboration partners, e.g. on a global level. In the ArKoS solution, globally visible data structures are only modeled as attributes of the process modules,[514] which does not reflect the importance of the data view in CBP. As indicated by the fact that data specifications represent the core of established eBusiness protocols like UN/EDIFACT,[515] the data dimension should be modeled separately from the process dimension, enabling a reuse of message types in different processes. A similar critique applies to the neglect of the *function dimension* in the ArKoS global view, which is also only regarded as part of the process dimension. The description of functions on a global level is a necessity, as recently indicated by the success of service description languages like WSDL, which describe functions on a technical level and are an essential element of (cross-organizational) SOA descriptions. Thus, a global view should tackle functions and processes in separate dimensions.

Therefore, apart from the processes, also organization structures, data, documents and services should be available to partners in the form of white, gray or black boxes. For this purpose, a generalization of the private-public-global concept known from processes to other enterprise modeling dimensions is proposed.

3.3.3.3 Generalizing the Private/Public/Global Concept

Accordingly, Figure 35 displays the relations between private, public and global models aligned to the relationships among private, public and global processes and their elements. The concept of collaborative views stems from the process dimension, which differs from the other AIOS dimensions: a process is dynamic in the sense that it describes the time/logical relationship between the elements (functions) comprised in it. The other dimensions do not take into account time relationships and thus have a rather static character; thus, deviations from the collaborative views as known from the process dimension can be expected. In Chapter 4, the specifics of the collaborative views are described in detail for each enterprise dimension. However, in order to define a common denominator among the enterprise dimensions usable for the collaborative view axis in the AIOS, the following generic characteristics of the collaborative views can be defined:

Public Models as Abstractions of Private Models The private/public/global concept as described in this thesis stems from the process dimension, and it can be assumed that in other dimension deviations from this concept might be useful. However, in all four dimensions the relation between private and public models is the same: The elements of a private system are situated inside one organization. A public system can be seen as the interface of a private system: It only comprises those parts of the private system that are relevant for collaboration partners (like for example send and receive activities, exchanged documents, or organizational roles involved in the collaboration). It does not comprise any classified elements of the private system.

---

the context of SOA, where an explicit, business-level description of the elements consumed and produced by a service increases its discoverability.

514 Compare HOFER ET AL. (2005), p. 5.

515 Compare for example UNITT & JONES (1999).

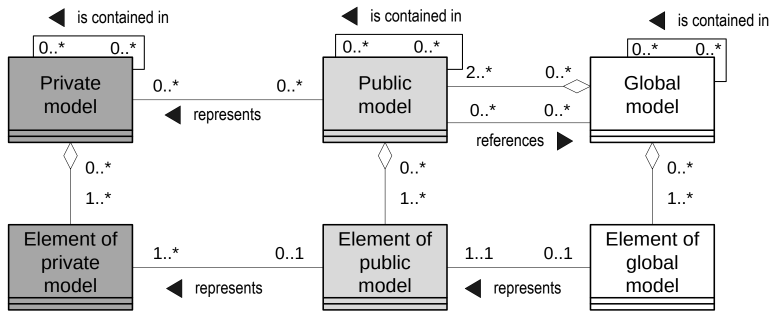


Figure 35: Generalizing the private, public, global concept

Different Interpretations of Global Models A global model always represents a globally known model, accessible and understandable by all collaboration partners. Following the distinction between public and global processes, a global system would be defined as a collection of complementary public systems. However, based on the relation between public and global elements, two different understandings of global models can be distinguished:

- The global model as aggregation of complementary public models: In the private/public/global concept as known from the process dimension, a global process consist of complementary parts – e.g. the different public processes from collaboration partners – that jointly form a valid collaborative business process. Similar to global process models, organization models on the global level can consist of *complementary* organization models. In this case, the organization elements of different partners comprised in a global view should also "fit together" as public processes comprised in a global process. For example, company A could publish the elements {"buyer", "is part of", "purchase department", "sends RfQ"} while company B publishes {"seller", "is part of", "sales department", "receives and answers RfQ"}.
- The global model as reference for a public model: While the public processes of two collaborating organizations in a global process should be complementary (e.g. send request and receive request), organization elements normally are not restricted to the execution of exactly one function and thus could be similar.[516] For example, two organizations could agree to automate a collaborative business process where the actor on each side has a role allowing him to "handle classified information". Especially in complex scenarios where many organizations participate, instead of complementary organization models, non-complementary global reference models describing rights and roles occurring in the different organizations are useful.[517] Thus, it also makes sense to use global models that serve only as a globally known and accepted reference for public models, even if the global models do not represent aggregations of different public models.

[516] If the organization model would be so fine-grained that each role is only related to one function, roles in a CBP would have to be complementary as well. E.g. organization A would offer the role "RfQ sender" while organization B would offer the role "RfQ receiver".

[517] For example, in an eGovernment scenario, the collaborating parties could agree on the role "investigating judge", specifying his access to documents or describing other permissions assigned to this role, e.g. ("investigating judge", "decide about revocation", "European Arrest Warrant"). Now each country participating in the collaboration can implement this role locally (e.g. "investigating judge Belgium", "investigating judge Italy").

Both understandings of a global model are represented in Table 6, where examples of both forms are given for different enterprise dimensions.

	Character of global model	Relation of public to global model	Construction of global model	Example of global model
Process	Set of complementary public processes	Is part of	Correlate complementary public processes	Public processes joined into a global process
Process	Globally known reference process stub	Is a	Relate public process reference public process	Reference protocol for interaction with a car manufacturer
Organization	Set of complementary public organization structures	Is part of	Correlate complementary public organization structures	Judge and attorney of the state
Organization	Globally known organizational reference model	Is a	Relate public model to globally known reference model	(Investigating Judge is a) Judge
Data	Set of complementary data schemes	Is part of	Correlate complementary data scheme	Complementary scheme for product description and service agreement
Data	Globally known data scheme	Is a	Relate document to globally known reference data scheme	(RfQ of company XY is a) Standardized RfQ
Function	Set of complementary functions	Is part of	Correlate complementary functions	Transaction, process
Function	Globally known reference function	Is a	Relate function to globally known reference function	(myRating is a) Service for CreditRating

Table 6: Different types of global models among the enterprise dimensions

In the case of *processes*, literature clearly concentrates on the first case, where a global process consists of complementary public processes. This applies to technical approaches, where complementary choreography interfaces are composed into choreographies as well as to business level-related approaches, where public processes models are combined into global process models.[518] In the case of *organization* elements, the second case, where certain reference roles are mapped to local roles seems to be more common. As described below (see p. 156), this approach is taken, for example, by distributed Role Based Access Control (distributed RBAC) and the distributed eXtensible Access Control Markup Language (distributed XACML). In the *function dimension*, a set of (complementary) functions usually is represented as a process;[519] thus, the understanding of a global model as reference for one function seems more useful in this dimension. In the case of *data* and documents, also the function of global models as (non-complementary) references is more frequently applied.[520]

Global Models vs. Ontologies The examples in Table 6 illustrate that global models can also be understood as ontologies: As described in Chapter 2 (p. 29), ontologies describe a group of concepts shared by a group of stakeholders in a formal way, like for example a modeling language. Further,

---

[518] Compare for example W3C (2006) or GREINER ET AL. (2006).

[519] It can be argued, that a set of functions only represents a process if these are time/logical ordered. However, a set of complementary functions in most cases implies a certain order among them.

[520] Compare for example BUGAJSKI, GROSSMAN & VEJCIK (2006) or GUO (2006).

the concepts described in the ontology are used and accepted by a certain user group – these characteristics apply to global models also. Moreover, like ontologies global models are not supposed to be used only in one context (e.g. one collaborative business process), but are supposed to be reusable in different collaborations. For example, the globally accepted specification of the document type "Request for Quote" could be used in various collaborative business processes.

3.3.3.4 Conclusions

As described above, private, public and global views can not only be used in the process dimension, but in a comparable way also in the other enterprise dimensions, e.g. function, data/output and organization. It was also described that these views can be used in the different MDA-levels CIM, PIM and PSM. Based on these presumptions, in Chapter 4 the usage of the three collaborative views for each enterprise dimension will be described in detail.

3.4 Consolidation of Overall Structure

Combining the results of the preceding sections, Figure 36 displays the three orthogonal axes that constitute the AIOS: Enterprise modeling dimensions (process, functions, organization, data/output), MDD-levels (business, technical and execution level), and collaborative views (private, public and global models). The result is a cube consisting of 34 complementary model types. However, as described in the application of the AIOS in Chapter 5 (pp. 235), not all model types have to be defined to enable a successful interaction: enterprise dimensions not needed in specific collaboration could be omitted; similarly, not all model types have to be described on all three vertical levels.

3.4.1 The Business Interoperability Interface as Core of the AIOS

In Chapter 2, the Business Interoperability Interface was defined as the sum of all models an organization has to provide to collaboration partners to enact a collaborative business process with them. It comprises those elements of a collaborative business process that are provided by the organization itself as well as the elements the organization expects from collaboration partners.

The model types comprised in the public view (displayed by the bold lines in Figure 36) fulfill this definition and they represent the interface between the internal information system and the adjacent information systems of collaboration partners. In other words, the public view of the AIOS represents the essential part of a Business Interoperability Interface. However, in addition to the public views, it is useful to also include a global view in the BII, e.g. to not only store public models in the BII, but also their relation to globally known models; this ensures that the collaboration partners have a shared understanding of the public models. Accordingly, the definition of the BII described in Chapter 1 (p. 37) is now instantiated as follows:

> *The Business Interoperability Interface of an organization represents the sum of all public models this organization has to provide to collaboration partners to enact a collaborative business process with them. The public models should describe all enterprise dimensions (processes, data/output elements, organization elements, and functions) and MDD-levels (business, technique and execution) relevant for the enactment of the collaborative business*

process. To support the correct interpretation of the public models, it is useful to describe in the BII also the correlation between the public models and their global counterparts.

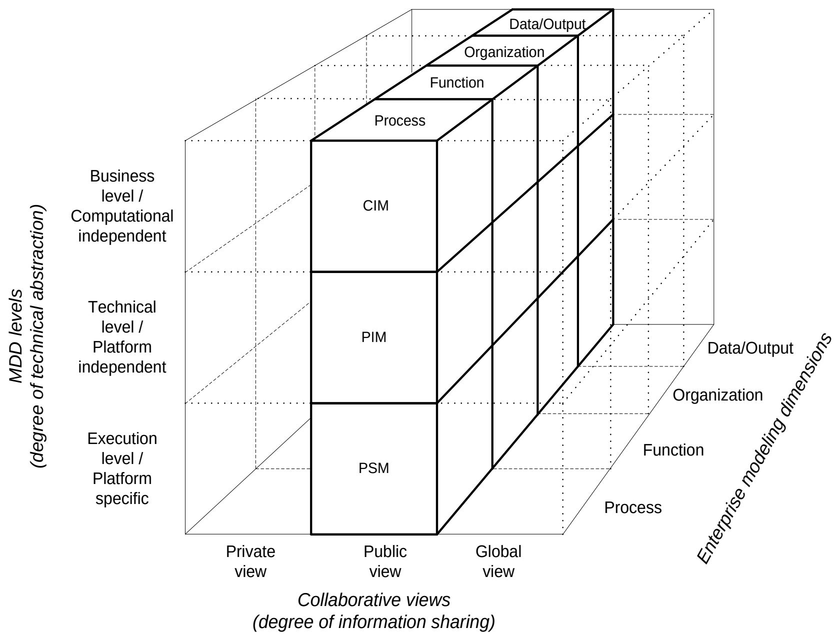


Figure 36: The Business Interoperability Interface at the center of the AIOS

Figure 37 illustrates that each organization offers one Business Interoperability Interface, accessible at least to the adjacent organizations, e.g. those organizations with which messages are exchanged. Inside each BII, different types of global models and related public models can be contained, for example, public and global views on document types, organigrams and business functions.

Note, that depending on the aims of the collaboration, different global processes can be comprised in this scenario. As described in Chapter 2,[521] in order to reduce complexity and to avoid information dispersal, a small scope of a global process is generally recommendable. In this vein, each couple of interlinked interfaces displayed in Figure 37 could be understood as one global process, describing, for example, the interactions among organizations A and C (e.g. A←→C). If, on the other hand, the organizations were interested in (monitoring) the activities of indirect collaboration partners, it would make sense to extend this notion of a global process. Thus, it would also be possible to describe the interactions A←→B and A←→C in one individual public process, which – eventually added to a public process that describes the interaction B←→C – would result in a global process that provides more knowledge to C than it requires for interacting with A.

---

[521] Compare pp. 75.

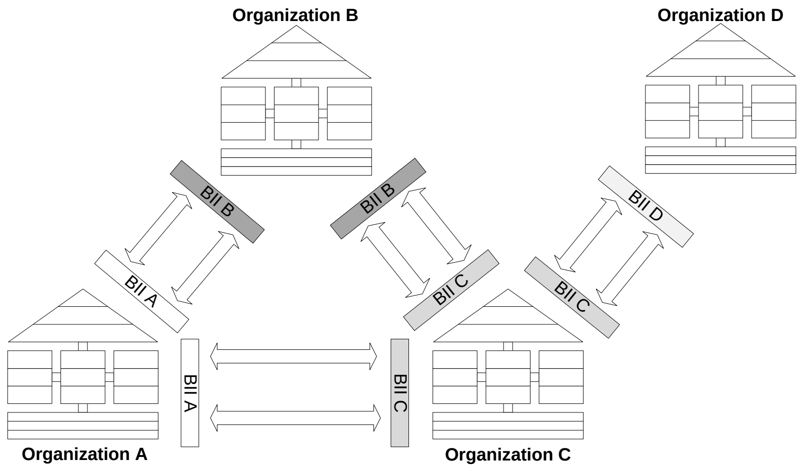


Figure 37: Each organization publishes its BII to collaboration partners

To also enable information sharing between indirect collaboration partners, collaboration spheres can be established among the collaboration partners. A collaboration sphere designates a group of organizations that collaborate, although not all of them interact directly with each other. Thus, in Figure 37 organization D does not interact with organization A. However, for strategic reasons it might be necessary for A to know that D is part of the collaboration. Going beyond elements needed in individual global processes, in the collaboration sphere, public and global models can be specified that are of interest for all collaboration partners. For example, a global organization model could be provided to be used throughout the collaboration sphere. Or, an organization could describe public views of its services, enabling the organizations in the collaboration sphere to discover potentially needed services.

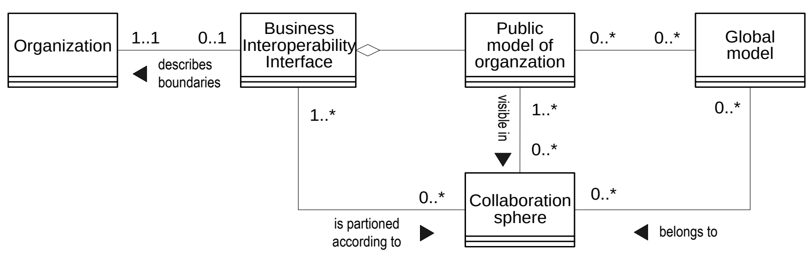


Figure 38: Relations between organizations, BII, comprised models and collaboration sphere

Figure 38 illustrates the above-described relations among organizations, BII, collaboration spheres, public and global models: One organization publishes one Business Interoperability Interface, which describes the interaction potential of the organization to collaboration partners. Since one organization can participate in different collaborations, the BII can be divided into disjoint subsections for different collaboration spheres.[522] The public models comprised in the BII are related to the collaboration spheres, and are visible only to the collaboration partners that are part of the colla-

[522] For example, a producer of metal doors could participate in one collaboration sphere with an airplane manufacturer and in another collaboration sphere with a car manufacturer.

boration sphere.[523] To put them in a context and to ensure a shared understanding, the public models can be related to global models that are agreed upon in the collaboration sphere.

3.4.2 Run Time vs. Design Time Functionalities

The AIOS focuses on design time aspects, e.g. the systematic development of interoperable information systems. It delivers the models needed for the description and the enactment of collaborative business processes, including not only business-level models, but, for example, also models of services, service-based workflows, access control mechanisms and document specifications. The deployment of these models and the technical infrastructure to execute collaborative business processes (for example, workflow engines or SOA execution environments like an Enterprise Service Bus) are not covered by the AIOS.[524]

The only component of the AIOS that can be used during run time is the BII-repository, in which each organization publishes its BII to collaboration partners. Since it comprises external views on information system elements, it can support (run time) publishing and discovery functionalities as needed in SOA: In the BII, the externally relevant processes, services, organization structures etc. are described on various levels of technical granularity, enabling other organizations to search also for business-level concepts and not only for technical artifacts. This implies that – different from the traditional SOA approach – instead of one central service directory, various partner-specific repositories are implemented. However, in the scope of this thesis, the BII-repository is understood mainly as a design time means that supports the joint development of collaborative business process models.[525]

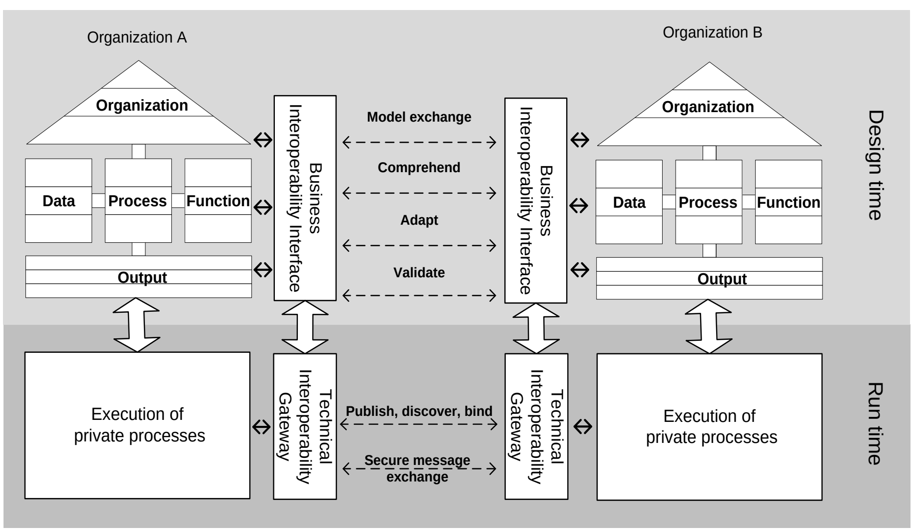


Figure 39: Overview of AIOS and the complementing execution environment

[523] The distribution of access rights on the BII of each organization is further described in Chapter 5, pp. 195.
[524] Compare also pp. 102, where it is stated, that the AIOS only tackles the logical, but not the physical level.
[525] However, it could also be accessed during run time, for example, for monitoring purposes or for the dynamic discovery of services. The BII-repository and its functionalities are described in detail in Chapter 5.

Complementing the run time functionalities of the AIOS, a corresponding *execution environment* must include the process engines implemented by each collaboration partner and mechanisms to ensure a secure transmission between these engines. Figure 39 illustrates the AIOS design time functionalities and the functionalities of a complementary execution environment: The BII is used mainly during design time, when the collaboration partners exchange, adapt and validate models needed for the enactment of collaborative business processes. The consolidated public and global models are then accessible via the BII of each organization. Complementary to the design time models, each organization implements a technical (run time) infrastructure, which realizes the secure execution of the cross-organizational processes described in the BII.

Such an environment was developed in the R4eGov project; compatible with the SOA focus of the AIOS, the execution environment is based on web services and an Enterprise Service Bus.[526] Further, the R4eGov execution environment describes an Interoperability Gateway, which acts as an intermediary between internal and external messaging systems.[527] Thus, it provides run time functionalities for sending, receiving, evaluating and routing messages as well as their secure transmission. Special attention was directed to the security of the collaboration, e.g. to mechanisms that ensure confidentiality, integrity and authenticity for the data transferred between the Interoperability Gateways of collaborating organizations. The corresponding security architecture is based on distributed RBAC, using XACML[528], which, as described in Chapter 5, is also supported by the AIOS.

---

[526] For a detailed description of the execution environment, refer to PLATE & HUYS (2007).

[527] Thus, it intercepts “functional communication” sent by external organizations and redirects messages to modules responsible for the addressed functions; compare PLATE & HUYS (2007), pp. 75.

[528] Compare SVIRSKAS (2007), pp. 22.

4 Specifying Individual AIOS Dimensions

While in the previous chapter the overall structure of the AIOS was developed, in this chapter, the dimensions of the architecture are defined in detail. For each enterprise dimension, a metamodel is specified that correlates private, public and global views. Following the research method described in Chapter 1, the development of the AIOS dimensions is based on the review of existing concepts and the goal, to realize the AIOS concept developed in the previous chapter. Thus, the overall design goals for each of the enterprise dimension specified in the following are:

- To enable collaborative views: The metamodel for each enterprise dimension should provide distinct views on private, public and global elements and enable their correlation.
- To support different levels of technical granularity: Apart from capturing business requirements, the metamodel should also represent a valid basis for a transformation to the execution layer. Thus, not only business-level concepts, but also concepts used at the execution layer are investigated to complement the metamodel with the possibilities offered by execution level technologies (the mapping of the resulting metamodels to exemplary execution level standards is described in Chapter 5).

Since the dimensions represent architectural views, each dimension should be "consistent and complete"[529]. Moreover, the dimensions should represent complementary entities and offer connection points to adjacent dimensions.

4.1 Process Dimension

In this section, the metamodel for the process dimension of the AIOS is defined. In a first step, existing approaches to model and implement private, public and global processes are analyzed. Based on this evaluation, the overall approach for modeling enacting collaborative business process views in the AIOS is specified. Following this design decisions, the process metamodel for the AIOS is specified.

4.1.1 Approaches for Collaborative Process Views

In the last decade, the usage of public views on internal processes was described by several authors. Though they all follow the mechanism known from database views, where a (public) view on a private system is created by abstracting from selected internal elements, significant differences exist between the approaches: Some tackle coarse-grained processes on a business level, others describe processes on the execution level. While in some cases, the public processes are only virtual concepts used on the design level, in other cases the public processes are implemented as autonomous

529 VAN DER AALST ET AL. (2007), p. 95.

workflows. In addition, the relationship among public and global processes differs, and correspondingly the operations to derive process views from each other. Some authors see public processes as contracts, which have to be fulfilled by an organization, other see them as communication guidelines for external parties. To prepare the exact specification of the approach to be used in the AIOS, in the following different approaches to describe and implement collaborative process views are reviewed.

4.1.1.1 Literature Review

VAN DER AALST described an approach for inter-organizational *workflow* development beginning with a global view on the overall interaction (coined public workflow) and the subsequent allocation of its elements (coined private workflow) to different partners.[530] Thus, in the terms defined in Chapter 2, he focused on global and private processes and the relation between them. In particular, he described problems that can occur if the private process differs from the corresponding parts in the global process. To prevent such inconsistencies he proposed to use workflow inheritance: If a private process is correct in the context of a global process, is must be a *specialization* of that part of the global process that specifies the behavior of the corresponding party. A similar approach was described by VAN DER AALST & WESKE, where it was called “Public-to-Private” approach.[531]

SCHULZ & ORLOWSKA described the usage of private and public views to enable cross-organizational workflows. However, instead of understanding public processes as mere abstractions of private processes, here they do not represent a sub-assembly of private processes, but can contain additional information: Their public processes represent *executable workflows* and serve as an active proxy between incoming requests and private processes. SCHULZ & ORLOWSKA also distinguished between the enactment of workflows as a choreography or an orchestration, referring to these concepts as “mediated” and “unmediated communication”. Global processes are called “coalition workflows”; congruent with the definition of global processes provided in Chapter 2, they state “a coalition workflow references workflow views, thus distributing the overall execution of the coalition workflows”[532]. To implement the private/public concept, they propose a tight coupling between private and public processes via state dependencies and a loose coupling between public processes of different organizations via control flow dependencies.[533] Thus, public and private processes would be implemented as closely interlinked state engines,[534] while the control flow among public processes is transmitted by passing “state and workflow-relevant data” [535] among the collaborating workflow engines. They also describe a corresponding architecture to implement CBP.

[530] VAN DER AALST (2002 B2B).

[531] Compare VAN DER AALST & WESKE (2001).

[532] SCHULZ & ORLOWSKA (2004), p. 121.

[533] Since both process types belong to one organization, it seems reasonable that the coupling between public and private process is tighter than the coupling between the public processes (belonging to different collaboration partners). However, whether a “control flow” represents a looser coupling than “state dependencies” depends on how both concepts are implemented – but normally, the term control flow indicates a tight coupling of process elements.

[534] They provide the example of a private process consisting of two tasks where the possible states of each task are modeled in detail. The public process consists of a similar state transition diagram, referring explicitly to the different states of the private tasks. The synchronization between the states of private and public processes is realized via the sending and receiving of events. Compare SCHULZ & ORLOWSKA (2004), p. 121.

[535] SCHULZ & ORLOWSKA (2004), p. 128.

SHEN & LIU described workflow views in a similar granularity as SCHULZ & ORLOWSKA but used slightly deviating concepts and terminology:[536] Private processes are called *Base Processes* and public processes are called *Process Views*. Further, they describe public processes as virtual processes whose "virtual states" display the status of private processes, and in this function can be accessed by partners to monitor and control the progress of an organization.[537] Thus, public processes are not only used during design time, but implemented as executable processes at run time, also. As Figure 40 illustrates, the authors do not only distinguish between private and public processes, but also between private and public functions (*Base Activity* and *Virtual Activity*), private and public data (*Base Process Relevant Data* and *Process View Relevant Data*) and between private and public control flow elements (*Base Dependency* and *Virtual Dependency*). Since public functions can aggregate various private functions, an explicit mapping of a public function to its private elements is necessary. Instead of global processes, they use the term "collaboration workflow", which is defined as the set of public processes offered by the collaborating organizations.[538] Thus, they do not use an integrated global process model for optimizing or validating the overall interaction but only a set of complementary public process. In addition to global processes, they introduce a so-called "integrated process", which represents the global process from the viewpoint of one participant and connects his private process with the public processes of the partners. Like SCHULZ & ORLOWSKA, they also describe a correlation between private and public processes based on state transitions.

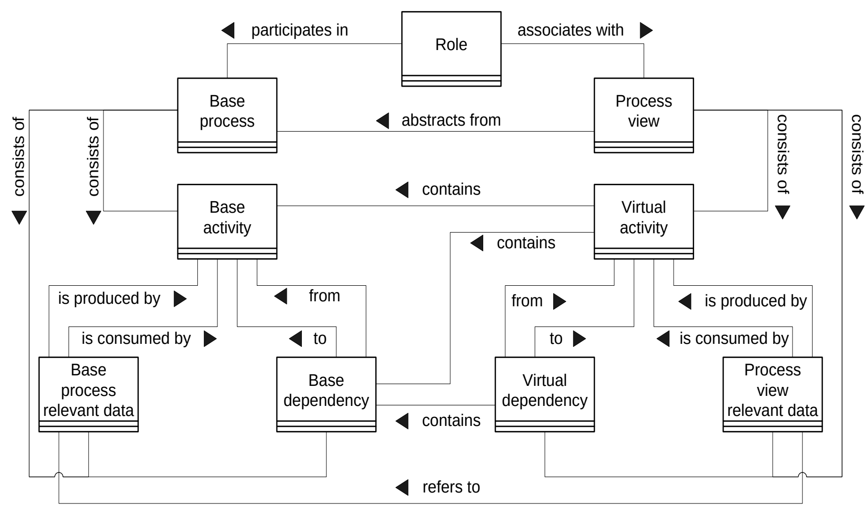


Figure 40: Workflow views according to SHEN & LIU[539]

CHEBBI, DUSTDAR & TATA picked up the concept of private and public workflows and related it to SOA.[540] Supporting the loose coupling between services in SOA, they used data flows to connect the interacting parties. Activities representing cross-organizational data exchanges are called cooperative activities; here, two types are distinguished: data producing and data consuming activities.

---

536 Compare LIU & SHEN (2003) and SHEN & LIU (2001).
537 Compare SHEN & LIU (2001), p. 275.
538 Compare SHEN & LIU (2001), p. 275.
539 SHEN & LIU (2001).
540 CHEBBI, DUSTDAR & TATA (2006).

Additional to private and public processes, they introduced the type *cooperative process*, which is defined as “the ‘minimal’ connected and compacted sub-process that contains all cooperative activities”[541]. In other words, it represents that subset of a private process that interacts with collaboration partners.[542] Thus, a cooperative process is very similar to our understanding of a public process. However, in difference to a public process, a cooperative process is not directed to a specific partner organization, but describes the interactions of the owning organization with any partner involved in the private process. Accordingly, the authors stated that a cooperative process can comprise many public processes. Interestingly, they restricted the number of partners addressed in one public process to only one (each public process is specific for one partner). Like LIU & SHEN, the authors also declared the functions inside a public process to be “virtual”. As they wrote, virtual activities do not produce or consume input/output but are only transferring data to and from other workflows; and they are not meant to be “executed by local role”.[543] They also proposed to extend the public-private concepts to other dimensions and specify cross-organizational data flows with so-called dataflow contracts.

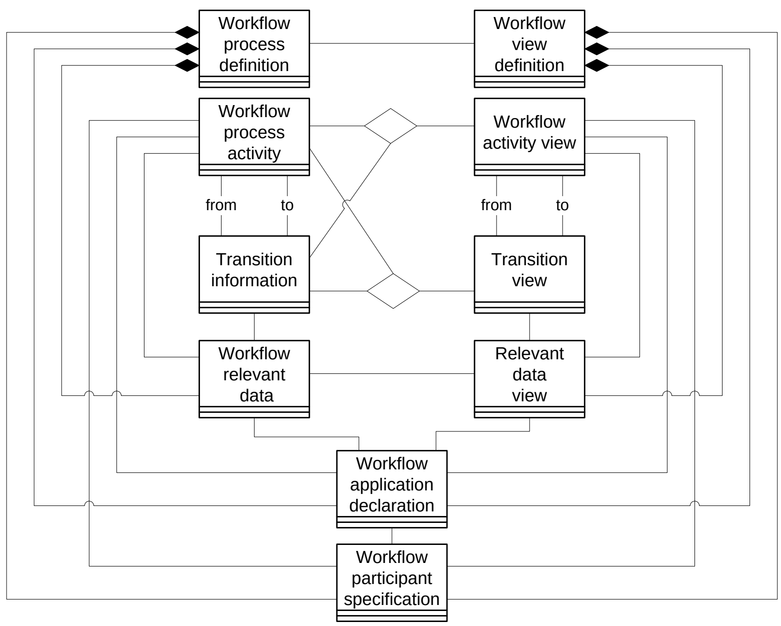


Figure 41: Workflow view metamodel from SHAN ET AL.[544]

SHAN ET. AL described an approach very similar to the one from SHEN & LIU.[545] They also distinguished between private and public versions of workflows, workflow activities, control flow ele-

---

541 CHEBBI, DUSTDAR & TATA (2006), p. 154.

542 Thus, it is different from other internal processes, which do not contain activities directly related to collaboration partners.

543 CHEBBI, DUSTDAR & TATA (2006), p. 156. However, their description of virtual activities appears vague, since the difference between producing/consuming and “transferring” data remains unclear. It is also unclear, how complex the logic for “transferring” data is; it could be a 1:1 relationship, e.g. operation A of a proxy service always redirects incoming data to operation B of an internal service. But also a 1:n relationship, where the message directed to operation A is analyzed and afterwards, the corresponding internal service is chosen out of a selection of available services.

544 SHAN ET AL. (2006).

ments (*Transition Information*) and data. Interestingly, both SHAN ET AL. and LIU & SHEN did not expand this division to the roles (*Workflow Participant Specification*); instead, only a global visible class for roles exists. In addition to the concept of LIU & SHEN, they related applications with the functions of the process. They did not describe the composition of public to global processes.

CHIU ET AL. described workflow views focusing on their use as electronic contracts, where a workflow view is defined as "a structurally correct subset of a workflow definition"[546]. As the cardinalities on the relationship between workflow and workflow view displayed in Figure 42 indicate, this also implies, that one workflow view represents at most one (private) workflow; note, that this understanding differs from other workflow view concepts, where one workflow view can also comprise functions contained in several private workflows.

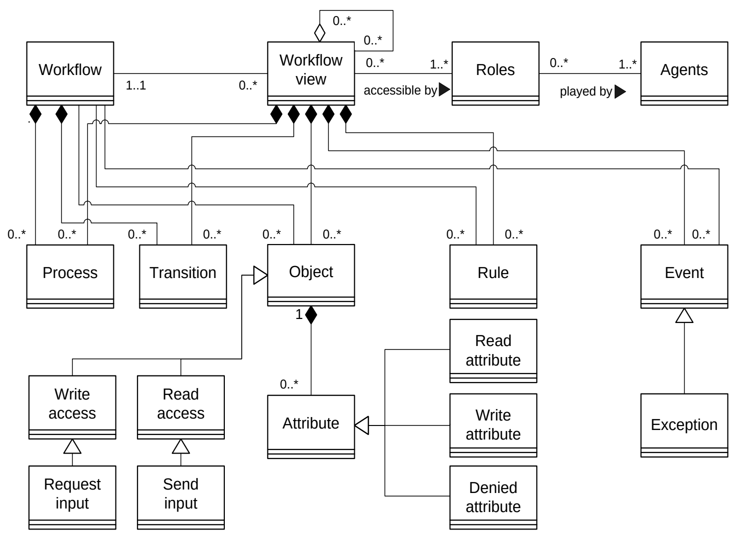


Figure 42: Metamodel for workflow views from CHIU ET AL.[547]

The only way to derive a public process (in their terms: workflow view) from a private process that they describe is through *aggregation*. They argued that most (private) processes are composed of sub-processes and that a public process can simply be derived by publishing different depths of sub-process hierarchies. For example, in a white box approach all sub-processes would be comprised in the public process, while in a black box approach only the top-level private process would be represented in the public process. Further, they advocated the possibility to rename private functions in case the original name would reveal critical information. They also argued that a public process should cover the same width of elements as a private process does. In consequence, elements like input/output parameters, objects, rules, events and exception handlers are not only related to private but also to public processes (compare Figure 42). Thus, objects of a private process also appear in the public process, though the attributes of the object might be hidden ("denied"). The access rights on objects are only coarse-grained ("read" and "write") and cannot be related to roles or time/spatial conditions. While the importance of events in the public processes is empha-

[545] Compare SHAN ET AL. (2006).
[546] CHIU ET AL. (2002), p. 199.
[547] CHIU ET AL. (2002), p. 200.

sized, it is not described how events in the public process relate to events in the private process. Neither did they distinguish between internally or externally visible roles. Further, they described the implementation of an engine to enact the public processes, serving as a run time-interface between external partners and internal workflow engines.

ATHENA In the ATHENA project the technical public-private-global concept as specified by SCHULZ for workflow views was extend for the modeling of business-level processes.[548] More specifically, three vertical modeling levels were distinguished: the business, technical and execution level.[549] On each level, a global process was defined to comprise various public processes, while a public process represents an abstraction of one or more private processes. The fact that "several tasks of a private process can be combined into one view task" and the statement that "a view process combines one or more private processes to an abstract level"[550] indicates that in the ATHENA approach, one public process does not necessarily have to be the subset of one private process. This on the other hand implies that public processes have to be implemented as executable processes, which work as run time-proxies between incoming message flows and internal processes.

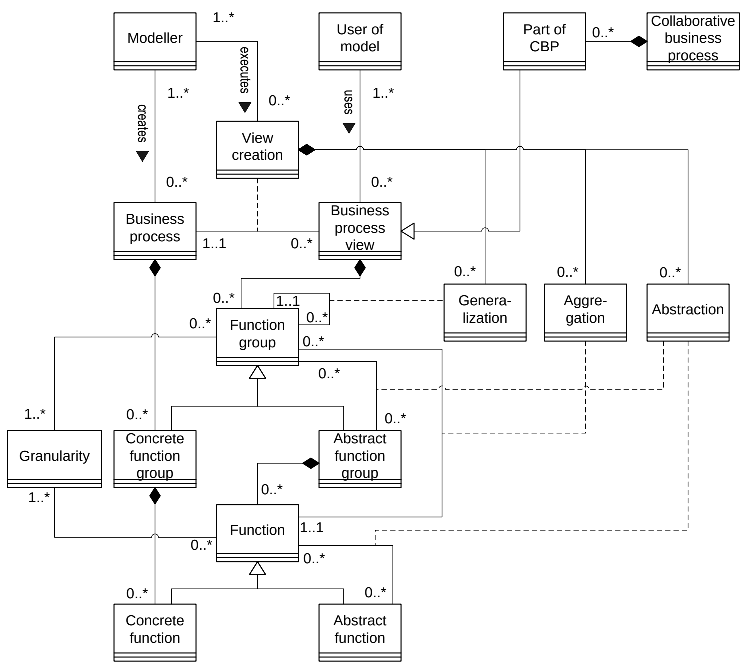


Figure 43: Global, public and private processes according to WERTH[551]

[548] Compare SCHULZ (2002) or SCHULZ & ORLOWSKA (2004).

[549] Compare ATHENA (2007), BORN ET AL. (2009), KAHL ET AL. (2007) and ZIEMANN, KAHL & WERTH (2007). On the business level, collaborative business processes were modeled with EPC and Mo^{2}Go (the latter is a process modeling notation created by the German Fraunhofer Institute, compare MERTINS & JAEKEL, 2006). On the technical level, a notation for technical collaborative workflows was used; on the execution level, processes were implemented with BPEL.

[550] BORN ET AL. (2009), p. 467. The term "view process" in their work is used equivalently to the term public process used in this work.

[551] WERTH (2006), p. 145; the text in the original graphic is in German.

WERTH described process views in a similar way as done in ATHENA.[552] However, different from the other authors, apart from abstraction and aggregation, he also counts generalization as an operation to derive public from private processes. In this context, he distinguished between the alteration of the process structure and the alteration of individual functions. Further, he stated that one public process can display only one private process, while (agreeing with most other authors) one private process might be represented by various public processes.[553] A description for the implementation of this concept is lacking, nonetheless, the aggregation relationship implies a realization of the public processes as executable proxies. As Figure 43 shows, a global process ("collaborative business process") is seen as a composition of public processes ("business process view"). In difference to the previous metamodels, his metamodel only distinguishes between processes and functions but does not comprise details of a process, as for example control flow elements, data, or roles related to functions.

BPEL is probably the most widely accepted standard for implementing Service-based workflows; since its first version, it supports two process types: Executable and Abstract Processes, which correspond to private and public processes. More specifically, a BPEL Abstract Process is a partially specified private process that is not intended to be executed. While BPEL Executable Processes are completely specified and thus can be executed, a BPEL Abstract Process can hide operational details contained in an executable process. However, in general, all constructs available to model BPEL Executable Processes can be part of their public counterparts.[554] For hiding elements of an executable process, BPEL Abstract Processes provide two concepts: Elements can be explicitly hidden by using the *Opaque Token*, and implicitly through omission.[555] The Opaque Token in an Abstract Process is a placeholder for an element in an Executable Process.[556] Besides activities, expressions and attributes can also be opaque. Complementary to the operations for disguising elements, the specification allows two operations to create an Executable Process from an Abstract Process: First, by replacing every Opaque Token with a corresponding executable token, second, by adding new BPEL elements somewhere in the process.[557] An Abstract Process is only valid, if at least one Executable Process exists that represents a completion of the Abstract Process. Furthermore, an Abstract and an Executable Process are called compatible if the Executable Process represents a completion of the Abstract Process.

Interestingly, two different usages of public processes are supported: First, for describing *contracts* for partners in collaborative scenarios, which also might be used at run time; second, to provide adaptable *reference models for internal processes* that can be completed at design time. In support of the first case, a "Profile for Observable Behavior" is defined, to provide predictable descriptions of observable service behavior like business process contracts. In support of the second case, BPEL offers an abstract process profile ("Template Profile"), which includes execution details and extension points for adding (internal) activities. This template allows process developers to complete execution details at a later stage – for example, adding conditions and defining endpoints for a

---

[552] Compare WERTH (2006), pp. 135 and ATHENA (2007).
[553] Compare WERTH (2006), p. 145.
[554] Compare OASIS (2007), pp. 7.
[555] Compare OASIS (2007), pp. 147.
[556] For example in Executable Process A, a certain opaque activity could be realized as an "invoke", while in Executable Process B this opaque activity could be implemented as an "empty" activity.
[557] Compare OASIS (2007), p. 152.

completion of the Abstract Process.[558] The specification does not describe in exactly which way BPEL Abstract Processes can be implemented, but indicates that the public processes following the Profile for Observable Behavior can be used during run time as business process contracts.[559] The public processes following the Profile for Templates on the other hand, are designated as design time artifacts.[560]

WSCI describes choreography interfaces in a way very similar to BPEL Abstract Processes. Both describe the externally visible behavior of web services, e.g. the sequence in which this service receives and sends messages, and most WSCI elements have an equivalent in BPEL. In contrast to BPEL, WSCI does not support the implementation of private processes (e.g. service orchestrations) but explicitly supports the forming of global models. A global model in WSCI consists of a set of connected choreography interfaces. To correlate the different interfaces, the interaction activities of web services participating in the choreography are related in the global model, e.g. the "consume" operation from a web service is mapped to the "produce" operation of another web service.[561]

BPEL4Chor DECKER ET AL. proposed an extension of BPEL in order to form global models by explicitly connecting BPEL abstract processes.[562] Therefore, they use the same mechanism as WSCI does: A mapping table that connects the choreography interfaces of all involved services. In this table, the sending activities of one service are related to the corresponding receiving activities of the partner service.[563]

WS-CDL With the Web Service Choreography Description Language, interaction-based global models can be described.[564] Thus, the global model is represented as an ordered set of interactions (where an interaction is defined as an exchange of messages between parties), and not as in WSCI, as a collection of connected choreography interfaces. Accordingly, WS-CDL does not support the modeling of public or private processes.

RosettaNet The consortium developed the Partner Interface Processes to describe global processes between exactly two organizations.[565] For this purpose, PIP describe both the message exchanged between the parties and the sequence of the message exchange.[566] However, neither private nor public processes are modeled explicitly. Since PIP are originally described only with UML Activity Diagrams, text tables and complementary XML documents, additional means are needed to describe them in a machine interpretable way. Accordingly, DAMODARAN proposes to describe PIP with ebXML BPSS,[567] MASUD as well as KHALAF propose to describe PIP, representing proven,

---

[558] Compare OASIS (2007), p. 159.
[559] Compare OASIS (2007), p. 155.
[560] Compare OASIS (2007), p. 159. BPEL is further discussed in the function dimension, where also the metamodel of BPEL's elements is described; compare pp. 142.
[561] W3C (2002), p. 8.
[562] Compare DECKER ET AL. (2007).
[563] This mapping list – which is called global model in WSCI – is called "Participant Topology" by them.
[564] Compare W3C (2006).
[565] RosettaNet was introduced above, compare pp. 82.
[566] Compare DAMODARAN (2004).
[567] Compare DAMODARAN (2004), p. 192.

well-established reference models for cross-organizational processes, in the form of BPEL processes.[568]

ebXML/UMM In ebXML, the Business Process Specification Schema is used to describe cross-organizational business processes.[569] The semantics underlying BPSS were derived from the metamodel of the UN/CEFACT Modeling Methodology; thus, UMM is closely related to ebXML. The UMM metamodel consists of four views: the business domain view, the business requirements view, the business transaction view and the business service view. The sequence of the views displays their level of technical granularity: While the first two focus on economical aspects and characteristics of individual business functions, the fourth tackles network components and application systems. In the third view, individual business transactions and their choreography into business collaborations are described. However, this description only covers interactions between exactly two organizations in form of global models,[570] but no public processes or private processes are described explicitly.

Further Approaches Besides the approaches listed above, further work related to collaborative process views can be found. LOHMAN ET AL., for example, developed the concept of controllability, which takes a complementary approach to public processes: instead of providing a collaboration partner with the public process he has to interact with, the partner is provided with a "Strategy" which is derived from the public process.[571] The Strategy describes a set of processes that can interact with the public process. Thus, instead of developing himself a process that is complementary to a given public process, the partner can directly implement the process described by the Strategy.[572] HOYER, BUCHERER & SCHNABEL described private, public and global processes on a business level. Following a model-driven approach similar to the one developed in ATHENA,[573] they propose to first model the different collaborative business process types in EPC and afterward refine these models into BPMN.[574]

4.1.1.2 Conclusions

Table 7 illustrates the heterogeneity of the approaches tackling collaborative process views. The majority of them stem from the workflow area (*technical/formal modeling*); only the approaches of ATHENA and WERTH include the modeling of process views on the business level. The table also shows, that, fostered by the proximity to formal models, many approaches propose architectures for implementing corresponding workflows (*description of execution mechanisms*).

Furthermore, the table shows that eBusiness protocol like ebXML and RosettaNet focus on global process models (*explicit global process*), while some approaches from the workflow area only describe private and public process models (*explicit private/public process*). Though the authors

---

568 Compare MASUD (2003) and KHALAF (2005).
569 Compare CLARK ET AL. (2001).
570 Note that these global processes can be seen as a special kind of public process, since the public process of organization A implicitly describes the complementary activities of organization B. Accordingly, these global processes also could be described with only one public process model.
571 Compare LOHMAN ET AL. (2006).
572 Compare also ZIEMANN, MATHEIS & FREIHEIT (2007).
573 Compare BORN ET AL. (2009).
574 Compare HOYER, BUCHERER & SCHNABEL (2008).

concentrating on the (public) process view approach often did not explicitly describe global processes, in most cases it was implied that global processes represent compositions of public processes. However, different relationships between public and global processes can be found, for example, regarding the number of partners a public processes can access. Thus, in the approach of CHEBBI, DUSTDAR & TATA a public process addresses exactly one partner, while in BPEL a public process can address various external organizations.[575]

	VAN DER AALST ET AL.	SCHULZ & ORLOWSKA	LIU & SHEN	SHAN ET AL.	CHIU ET AL.	CHEBBI ET AL.	ATHENA	WERTH	BPEL	WSCI	BPEL4Chor	WS-CDL	RosettaNet PIP	ebXML/ UMM
Business-level modeling[576]							●	●						○
Technical/formal modeling[577]	●	●	●	●		●	○			○		●	●	●
Description of execution mechanisms[578]		●	●	●	●		●		●	○	●		○	●
Explicit private process	●	●	●	●	●	●	●	●	●		●			
Explicit public process		●	●	●	●	●	●	●	●	●	●			
Public process is subset of private process					●	●			●	●	●			
Run time enactment of public processes[579]		●	●	●	●	○	●		○	○	○			
Explicit global process	●	●	●				●	●		●	●	●	●	●
Global process restricted to two partners													●	●
Interface-based global model	●	●	●	●	●	●	●	●	●	●	●		●	●
Interaction-based global model												●		

● / ○ / Blank : Criterion is completely / partly / not fulfilled

Table 7: Different approaches to creating collaborative process views and their characteristics

The row *public process is subset of private process* indicates whether the public process is an exact subset of the private process, e.g. if each activity in the public process is comprised exactly once in the private process. This is, for example, the case in BPEL, where the knowledge comprised in the public process model allows a partner to interact directly with the private process. It is not the case if the public process acts as an executable proxy and contains functions that are not comprised

575 Compare CHEBBI, DUSTDAR & TATA (2005) and OASIS (2007).

576 Though approaches aiming at business level modeling for ebXML exist, these are based on technical modeling languages, e.g. UML.

577 Though the public views in ATHENA were based on the formally described concepts of SCHULZ (2002), in the project itself, these process types were not described formally. Also in WSCI, the conception of public processes is only described in natural language.

578 The usage of WSCI models during run time is not described explicitly; however, like BPEL Abstract Processes WSCI models could be used as contracts or protocols being executed in a "conversation controller" (compare also ALONSO ET AL., 2004, pp. 209). RosettaNet PIP represent design time models; however, concepts to map models to execution standards were described for example by MASUD (2003) and DAMODARAN (2004).

579 In the concept of CHEBBI ET AL., a public process consists of virtual activities that transfer data between external and private processes. In the case of BPEL, BPEL4chor and WSCI, public processes can be used at run time as conversation protocol; however, the usage of these processes at run time is not described in detail.

in the private process,[580] as for example in the workflow view approach of SCHULZ & ORLOWSKA.[581]

In summary, it can be said that many approaches for collaborative process views focus on technical details of public processes (e.g. in the form of workflow views), neglecting the relationship of technical solutions to business level requirements. On the other side, some of the business-level concepts for collaborative views remain too abstract and do not address the feasibility of the approach. For example, it sometimes remains unclear, if public processes are only design time artifacts, or if (and how exactly) they could be used during run time. An exception is BPEL, where the usage of public process is explicitly described for both design time – as templates for internal processes – and for run time, where public processes act as contracts controlling the correctness of a conversation. The row *run time enactment of public processes* indicates whether the approaches imply an implementation of public processes as executable proxies.[582]

Also in the modeling of global processes, differences exist. Since the existence of public process models imply their further usage by combining them into global processes, the majority of the approaches follow the *interface-based* style. Only the global models in WS-CDL do not comprise explicit public processes but follow the *interaction-based* style. While in the newer approaches the number of participants in global processes is not restricted, in eBusiness protocols global models displaying the interactions between two parties are used.

4.1.2 Design Choices for the Process Dimension

The literature review revealed the diversity of the approaches for realizing collaborative views on processes, implying the need to decide which approach should be chosen for the AIOS. More specifically, it became clear that two major design choices have to be made: whether interface- or interaction-oriented global models should be pursued, and, whether public processes should be realized as choreography interfaces or workflow views.

4.1.2.1 Public Processes as Choreography Interfaces

In the evaluation of the literature review, two characteristics of approaches for implementing public processes were distinguished: whether the public process is an abstraction of the private process, and whether the public process is executable in form of a proxy. Figure 44 illustrates that based on these two characteristics, two understandings of public processes and corresponding implementation forms can be distinguished.

Two Types of Public Processes In the first case, public processes are executed during run time by workflow engines. Instead of the private process, a physical instance of the public process is addressed by collaboration partners. Thus, the public process acts as a *proxy* that transfers messages between internal and external parties. To enable this transmission, additional information is neces-

580 For example, the public process can contain the public function F1 which is an aggregation of the two private functions F2 and F3; however, the function F1 itself is not part of the private process.

581 Compare SCHULZ & ORLOWSKA (2004).

582 Details about public processes that are pure abstractions from private processes and public processes that can serve as executable proxies for private processes are described in the next section.

sary to describe how the public process is mapped to one or more private processes. In the second case, only the private process is executed by a workflow engine while the public process can be omitted at run time. The public process serves only as a *protocol*, describing in which sequence the private process consumes and produces messages. Here, the private process is addressed directly by the collaboration partners, while the public process model only describes how the private process should be addressed. Therefore, the functions described in a public process model must be comprised in the private process also.

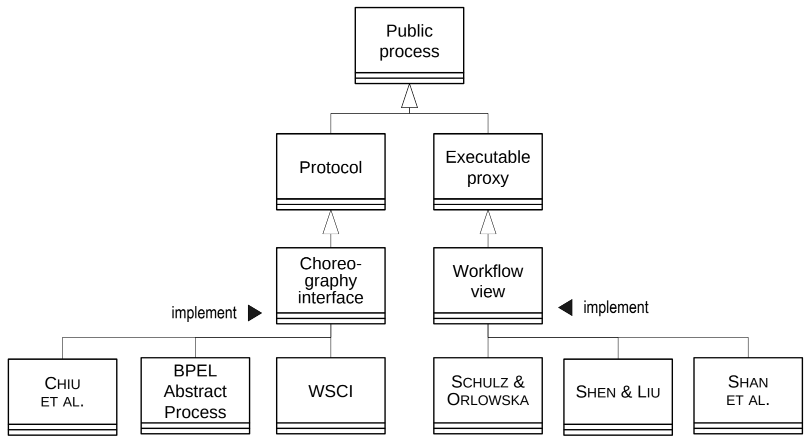


Figure 44: Different types of public processes and their implementation

Public Processes as Executable Proxies Most process view approaches described above suggest an implementation of public processes as executable proxies. This applies, for example, to the workflow views proposed by SHEN & LIU, SHAN ET AL. or SCHULZ & ORLOWSKA.[583] Here, the elements of public processes do not have to be identical to the elements of private processes. Thus, they can represent aggregations of private process elements or, in the case of a refinement, can represent only one part of an element of a private process. On the right-hand side of Figure 45, an executable proxy is illustrated. The public function C1 represents an aggregation of the private functions A1 and A2. If a message is send to C1, the different parts of the message have to be dissected and related to the corresponding internal functions. Thus, additional to the information comprised in the public process C1 and the private process A, information is needed on how the functionalities of C1 as well as its input and output elements are mapped to private processes. The private function B1 on the other hand is an aggregation of C2 and C3. In this case, the mapping mechanism has to combine the messages directed to C2 and C3 and relate them to B1.

Public Processes as Choreography Interfaces Workflow views can be endowed with elements going beyond the functionalities of one private workflow. Instead of displaying exactly the capabilities of one workflow, they rather represent an independent description of an externally required workflow, assembled from the functions of various internal workflows. In the context of SOA on the other hand, public processes are used to display exactly the externally visible behavior of a service. Thus, the public process is not allowed to describe details that are not part of the private process implementation. Accordingly, the choreography interfaces realized with WSCI or BPEL

[583] Compare SHAN ET AL. (2006), SHEN & LIU (2001) and SCHULZ & ORLOWSKA (2004).

Abstract Processes display the public elements of a private process,[584] which are implemented by one service; therefore, elements of the private process can be hidden, but no additional elements can be added to the public process. This implies that one public process represents a subset of one private process and that each function represented in the public process is represented in the private process. Since the aggregation of private process elements would require an executable public process, between private and public process elements only the abstraction relationship is applicable.

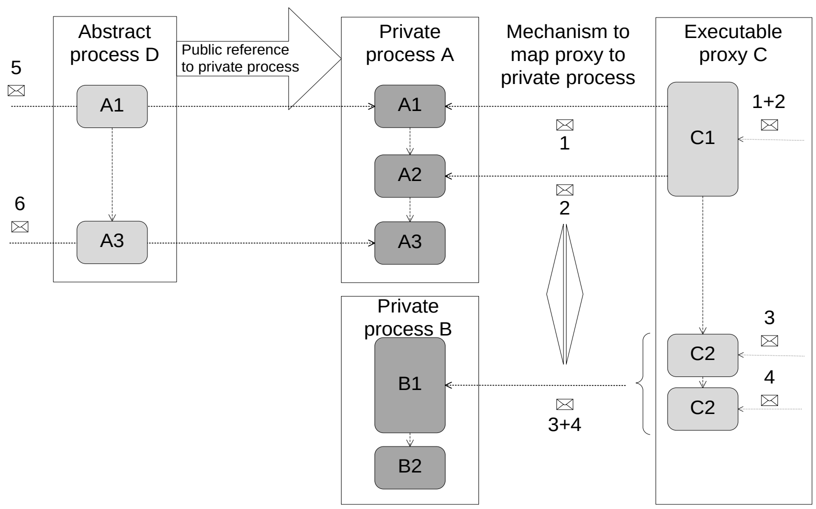


Figure 45: Abstract process and executable proxy as implementations of public processes

Public processes in the form of choreography interfaces are mainly design time instruments; at run time they can only be used to verify the correctness of a conversation or to discover a service, but not to process and transmit information as executable proxies do. Figure 45 illustrates that an abstract process represents a subset of the private process. The messages 5 and 6 from a service requestor are not sent to an instance of a public process, but directly to the private process. This is possible, because the public process describes exactly the elements to be addressed in the private process; it also comprises a reference to the service that implements the private process.

Conclusions The advantage of implementing public process as executable proxies is that more complex relationships between the private and public processes can be realized: Apart from pure abstractions, also aggregations are possible as well as the possibility to realize the functions of one public process by more than one private process. However, to gain this advantage an additional layer of executable processes must be implemented as well as a mechanism to connect private and public processes. In the case of the choreography interface on the other hand, apart from the private process, no additional public process must be implemented at run time to enable a communication with external stakeholders. Further, the realization of public processes as externally visible descriptions of a service is compliant with the SOA principle of interface-orientation. To avoid the costs for an additional run time layer and to support the principles of SOA, for the AIOS the implementation of public processes as choreography interfaces is chosen.

[584] Compare W3C (2002) and OASIS (2007).

4.1.2.2 Interface-Based Global Models

As described in Chapter 2, global processes can either be described with interface-based or with interaction-based models. The existence of competing choreography standards (WS-CDL vs. WSCI/BPEL4chor), which follow either an interaction-based or an interface-based approach, indicate that different opinions exist on which approach is "better".[585] For both representation styles, generic arguments can be found:

DECKER & HAFNER argue in favor of the interaction-based global models, their main argument being that "incompatibility between different participants cannot occur in choreography models"[586]. However, this statement is a tautology, since in this model type participants are defined by the sum of their activities in the choreography. In this sense, the participants in the model are always "compatible" with each other. Naturally, this does not mean that the characteristics of the services described in the global model could also be implemented by the participating partners. Moreover, ZAHA ET AL. state, "some global models may not be translatable into a collection of local models such that the sum of the local models equals the original global model", e.g. that some global model "can not be enforced locally".[587]

To summarize, both the overall compatibility of the interacting services and the ability of each collaboration partner to fulfill its part of the choreography are important. From a *business* perspective though, it seems more likely that an enterprise will resort first to the service it delivers and the services it can expect from collaboration partners, and only afterwards to the detailed interactions that connect the different services. From a *technical* perspective, interface-orientation is one of the main paradigms of SOA and it makes sense to support this principle by maintaining the interfaces of the participating services not only implicitly, but also explicitly in the choreography model. In this vein, the following arguments in favor of interface-based global models can be found:

- Easier to understand for collaborating parties: For the individual organization, it is easier to analyze its role in the choreography and the activities it has to provide, as well to understand the role of partners.[588] The better comprehensibility of the interface-oriented approach is also indicated by modeling tradition, since historically interaction models are mostly clustered according to the involved organizations: Taking the UML models found in literature, interactions have been displayed predominantly in sequence diagrams, or, more recently, in activity diagrams divided into swim lanes – but hardly ever in activity diagrams without swimlanes. DECKER & PUHLMANN argue that interaction-based models have fewer elements than interface-based,[589] which is questionable; actually, Figure 10 (p. 44) illustrates that the opposite can be the case. At least the number of interaction activities should be the same in both model types; and, as shown in the same figure, the interaction-based model requires that for each interaction, the affiliated

---

585 Compare also DECKER & PUHLMANN (2007), p. 37. Also in the R4eGov project, the question was discussed intensely, whether to rely on an interface-based style (using BPEL), or if the interaction-based representation of global models should be supported, relying on WS-CDL.

586 DECKER & PUHLMANN (2007), p. 38.

587 Compare ZAHA ET AL. (2006).

588 This effect is also visible in Figure 10, p. 44.

589 Compare DECKER & PUHLMANN (2007).

organization has to be explicitly modeled – which is not the case in the interface-based style, where all activities are affiliated to the owning organization.

- Easier to implement: By definition, choreographies are to be implemented by various services. And, prior to its implementation, for each service an interface description must exist. As interface-oriented models already contain the choreography interface of each partner, a potentially error-prone process of converting an interaction-oriented into an interface-oriented model can be evaded.
- Easier to change or alter services in the global model: Since interface-oriented models explicitly display the characteristics of individual services, the exchange or alteration of individual services in the choreography is easier. In an interaction-oriented global model, the activities belonging to one service and the interdependencies between them are harder to comprehend.[590]

Following this arguments, the interface-based representation of global models is chosen for the AIOS.

4.1.3 Metamodel for the Process Dimension

4.1.3.1 Relations Between Collaborative Views

Based on these design decisions, the relations between private, public and global processes for the AIOS process dimension can be defined as follows:

Processes, Sub-Processes and Process Elements Since a process describes the sequence of functions, it must comprise at least two functions. Thus, as displayed in Figure 46, a private process consists of 2 to many elements, including functions, events and control flow elements. One private process element can be contained in 0 to many private processes (e.g. the function "credit checking" can be used in different processes). A private process can have 0 to many sub-processes; on the other hand, a private process can be contained in 0 to many upper-level private processes. The same relations apply for public and for global processes: They consist of 1 to many elements and each can be contained in 0 to many processes; each process can have 0 to many sub-processes, a process can be contained in 0 to many upper-level processes.

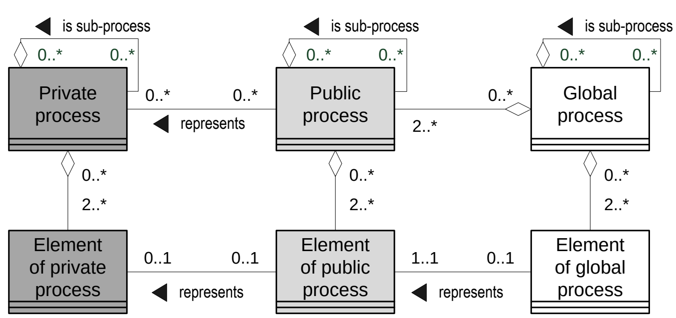


Figure 46: Generic metamodel for collaborative business processes

590 The interdependencies between activities of the altered service and the other services will have to be changed and validated. However, this applies for both types of global models.

Relations Between Private, Public and Global Processes One *private* process can have 0 to many public processes, for example, one public process illustrating interactions with partner A and one separate public process illustrating interaction with partner B. On the other hand, one public process can represent 0 to many private processes, e.g. the interaction sequence "receive proposal for meeting and reply either with accept or decline" could be part of various private processes. Following the interface-based interpretation of global processes, a global process represents a set of at least two interconnected public processes. In case a public process is contained in 0 global processes, it just represents the interface of an internal process.[591] The fact that one service can be used in different collaborations implies that one public process can be part of different global processes.

Relations Between Private, Public and Global Process Elements Based on the assumption that a public process abstracts information from a private process, one element of a private process has 0 to 1 counterparts in its public interface.[592] Since public processes are to be implemented as choreography interfaces, only the abstraction relationship among public and private processes is allowed. That implies that one element of a public process can represent at most one element of a private process. Since the interaction elements (e.g. data producing or consuming activities) of the private processes must be described in the public process, each interaction activity in the public process must have an identical counterpart in the private process. However, the public process may contain control flow elements, which are not part of the private process.[593]

If a public process is part of a global process, the complete public process should be part of the global process. However, since not every public process is part of a global process, one element of a public process can be matched to 0 or 1 elements of global process. On the other hand, each global process can be decomposed into disjoint different public processes; thus, each element of a global system can be mapped to exactly one element of a public process.

4.1.3.2 Connecting Conceptual and Execution Level

After the relationships between private, public and global process models have been defined, the process metamodel can now be refined in order to represent a basis for connecting conceptual and executable business processes. As motivated above,[594] the EPC is chosen as foundation for the metamodel of the AIOS process dimension, since it is suitable for a comprehensive modeling of busi-

---

[591] Note, that for any given public process the public processes complementary to it can automatically be derived, resulting in a global interaction model. Thus, a public process model could also be seen as a specialization of a global model. However, in this work only global models are regarded, which explicitly contain the public processes comprised in the collaboration.

[592] In the context of the ATHENA project we also pondered the possibility that one element of a private process can be displayed by an aggregation of various elements from a public process (a corresponding modeling prototype was implemented also). This is only reasonable if a public process is seen as a protocol, which is finer grained than the private processes. However, in this work we assume that related private, public and global models describe contents on the same level of technical granularity.

[593] For example, the private process may consist of the sequence of functions [F1, F2, F3] while the public process consists of the sequence [F1, F3]. Thus, in the public process a control flow edge exists among the functions F1 and F3 that does not exist in the private process.

[594] Compare p. 68 and p. 107.

ness level concepts, and at the same time represents a good basis for a generation of collaborative BPEL processes.

In Figure 47 the elements of the EPC and their relationships are displayed. The main process elements are functions and events; the sequence of both is determined by AND, OR, and XOR operators and control flow edges (displayed by the class *Successor/predecessor*). Resources can be attached to the functions and a specific class describes the relationship between function and resource.[595] Resources are represented in ARIS' data, output and organization dimension. Note that in ARIS the application system that implements a function can also be attached to a function. However, fostering a model-driven approach, business functions and the IT services that implement them, in the AIOS are understood as the same function only on different levels of technical granularity. Therefore, in the following, the correlation of functions to application systems is not regarded as part of an enterprise dimension, but as part of the vertical mapping from business to execution level.

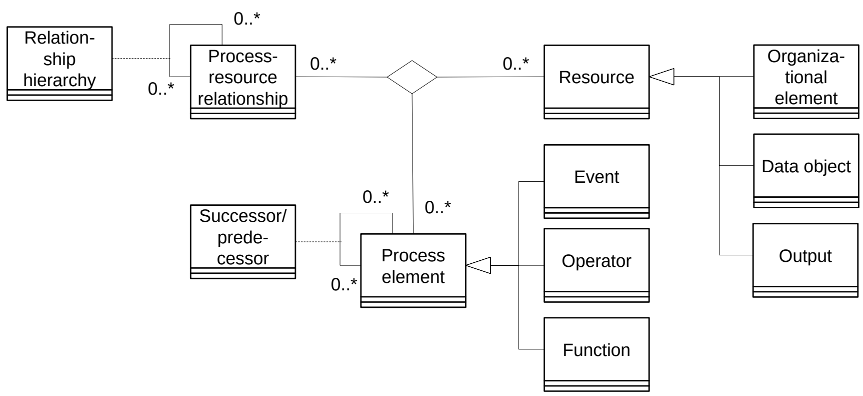


Figure 47: Metamodel of the Event-driven Process Chain[596]

Figure 48 displays the metamodel for the process dimension of the AIOS. It is based on the core EPC elements; only functions, events and their sequence are described; further business process elements, for example, organization elements and documents, are described in the following sections. The relationship between the private and the public view on the process follows the specifications of the generic metamodel for collaborative business processes.[597] In Figure 48, the differences between relations among process elements and relations among control flow elements are also visible: While each event, function or operator of the public process needs a counterpart in the private process, not all public control flow edges are required to have private counterparts.

To support a loose coupling among the public processes, they are only connected via events. For example, the event "RfQ is sent" produced by the public process of organization A could represent

595 For further descriptions of EPC refer for example to KELLER, NÜTTGENS & SCHEER (1992) or MENDLING (2007).

596 Adapted from BECKER ET AL. (2002), p. 87. In the original metamodel, the resource types are deviating from the ones displayed above. The resource types in the illustration are closer aligned to ARIS (compare SCHEER, 2000).

597 Compare Figure 46.

the connection point to a corresponding consuming activity of organization B.[598] To enable the correct correlation of public processes, an event emitted from one public process should be mapped to exactly one event of an adjacent public process.

Supporting the interface-based representation (and implementation) of global processes, in the metamodel, the global model is not displayed as one class, but as the sum of adjacent public processes. For illustrative reasons, only two collaborating organizations are depicted in the graphic. Nevertheless, Figure 48 illustrates that one public process can be connected to 1 to many adjacent public processes; these can be positioned either within one organization or in several organizations. Thus, while the graphic indicates that one collaboration partner offers one public process, likewise many collaboration partners could offer many public processes, creating one global process made of many public processes.

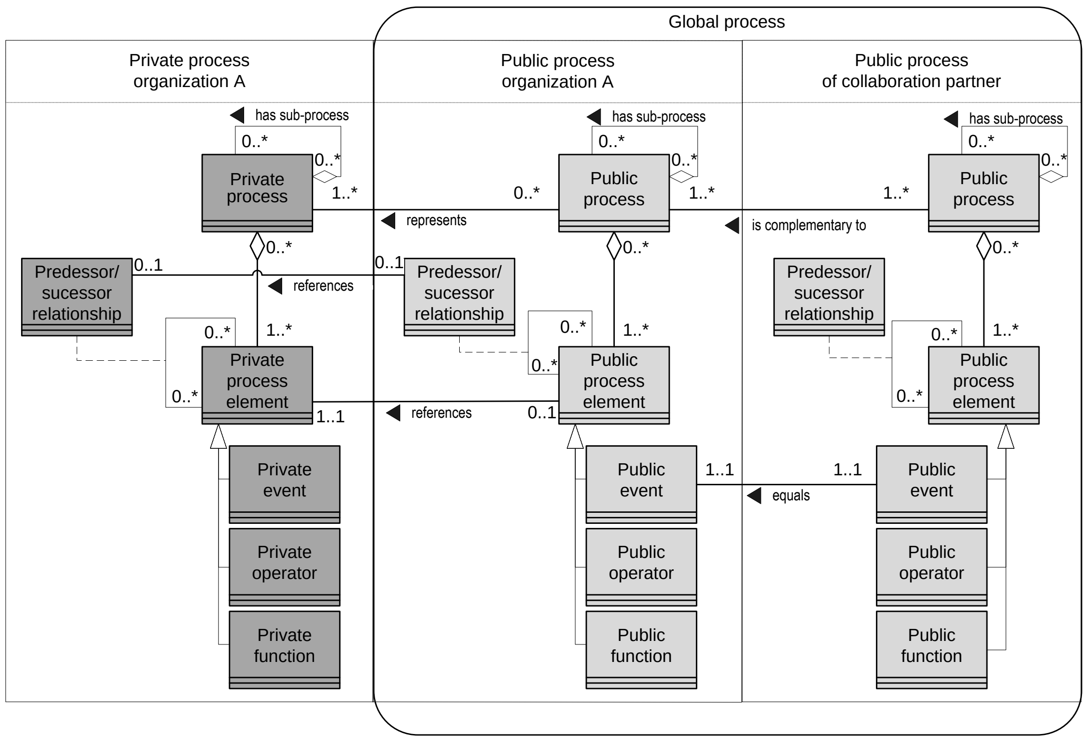


Figure 48: Metamodel for the process dimension of the AIOS

4.2 Function Dimension

The objective of this section is to define a metamodel for the function dimension of the AIOS. Similar to the other dimensions of the BII, the function dimension should represent a self-sustained layer

598 Note that in company B, the event could have a different name; instead of “collaboration partner A has *sent* RfQ” the event can be described from the perspective of company B, e.g. stating, “RfQ was *received* from collaboration partner B”.

complementary to the other dimension, connect public with private and global views and support model-driven development.

4.2.1 Approaches for Collaborative Views on Functions

As mentioned before, functions and processes are closely related:[599] A process represents a sequence of functions, and, on a lower level of granularity, a process also can be seen as a function. Business functions and their technical counterparts (e.g. components, modules, services, classes) are traditional means of achieving information hiding in information systems. In difference to process interfaces, interfaces of functions only expose static information by describing the set of operations offered by a function and their input/output parameters, but not the sequence in which operations are used.

Due to the essential role of functions in the implementation of IT systems, many approaches exist to describe functions on different levels of technical granularity. Since functions usually are described as black boxes – and thus allow for controlled information hiding – existing means for modeling functions are implicitly well suited for collaborative business. However, as described in the following, several modeling approaches exist which offer additional means for functional information hiding, going beyond the principles usually associated with modules. To assess the characteristics of functions on different levels of technical granularity, approaches that are not specific for collaborations are also reviewed in the following.

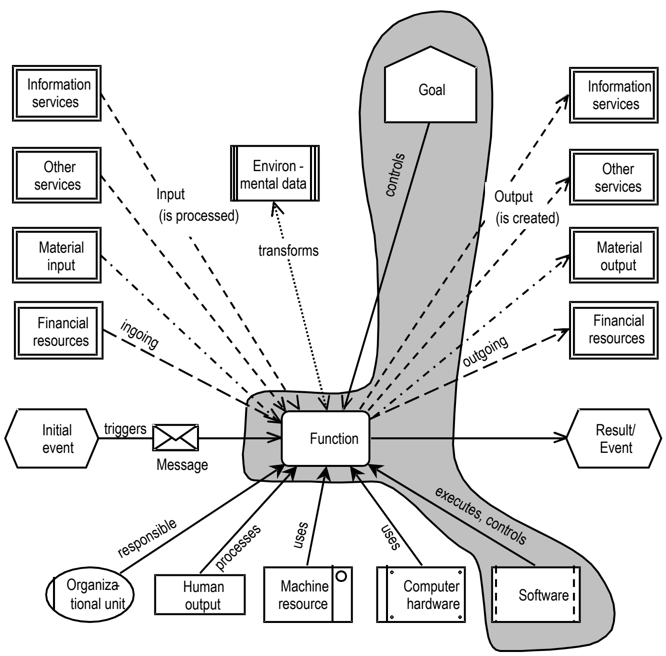


Figure 49: Function view in ARIS[600]

[599] Compare Chapter 2, p. 16.
[600] SCHEER (1999), p. 14.

4.2.1.1 Literature Review

On the business level, SCHEER defined functions in four levels of granularity: Function bundle, function, sub-function and elementary function. Regarding the execution of function, he distinguishes between system functions that are executed by application systems and manual functions, executed by humans. The approach explicitly supports process automation;[601] to this purpose, functions on the business level are correlated with modules on the execution level.[602] Apart from the functions themselves, ARIS' function dimension comprises the goals that drive the functions and the software applications that implement the functions.[603] Figure 49 illustrates that functions in ARIS are pivotal elements among the other dimensions, since the elements of the other ARIS dimensions are attached to functions. The graphic also illustrates the close relationship between functions and events. Events indicate and trigger status modifications,[604] since such modifications are realized via functions, in ARIS functions and events are always alternating. On the execution level, events are represented by data updates.[605] Also in the function view of ARIS, three vertical levels are distinguished: on the requirements level, functions, related applications systems and goals are modeled. On the technical level, the software modules that realize the functions are sketched out, using, for example, mini specifications to describe the content of modules. On the execution level, the specification of the software modules is realized with programming languages, e.g. executable source code is generated.[606]

BECKER, ALGERNISSEN & FALK recently described a modeling method for coarse-grained functions based on so-called process building blocks.[607] The principles for the development of these building blocks are well known from traditional software components: modularity, standardized interfaces, abstraction, discoverability, modifiability and completeness.[608] The description of each building block comprises its overall functionality ("*what* does it do?") as well as the method to realize this functionality ("*how* is this functionality realized?").[609] However, the description of each building block stays on a rather generic level, comprising its name, a number of attributes and a list of the relationships between the building block and other model elements.

WERTH described process building blocks for the support of collaborative business processes. In accordance with the definition of functions described in Chapter 2,[610] these building blocks are defined as producing an output using certain resources. However, they are coarser grained than functions, since one process block represents an aggregation of functions. Three phases of functions are

---

[601] Compare SCHEER (2000), p. 40.
[602] Compare SCHEER (2000), pp. 21.
[603] Similar to functions, goals in ARIS can also be displayed in hierarchies, thus goals can be split into sub-goals. Compare SCHEER (2000), p. 27.
[604] Thus, the event "RfQ was sent" indicates a status, at the same time it can trigger a reaction, for example the evaluation of the RfQ at the receiving organization.
[605] Compare SCHEER (2000), pp. 125.
[606] Compare SCHEER (2000), pp. 50.
[607] Compare BECKER ET AL. (2007), p. 277.
[608] Compare BECKER, ALGERNISSEN & FALK (2007), pp. 100. The last characteristic signifies that the functionalities contained in a set of building blocks completely cover the demand of the environment they are used in.
[609] Note, that this contradicts the principle of modularity, which requires that a module shows only *what* it does, while the process of *how* this is achieved is subject to information hiding. Compare SCHEER (2000), p. 42.
[610] Compare p. 16.

distinguished: pre, main, and post phase.[611] Special attention is given to the interfaces of functions; here WERTH distinguishes between external and internal interfaces for each of the phases described before.[612] In other words, he suggests private and public views on a collaborative function.

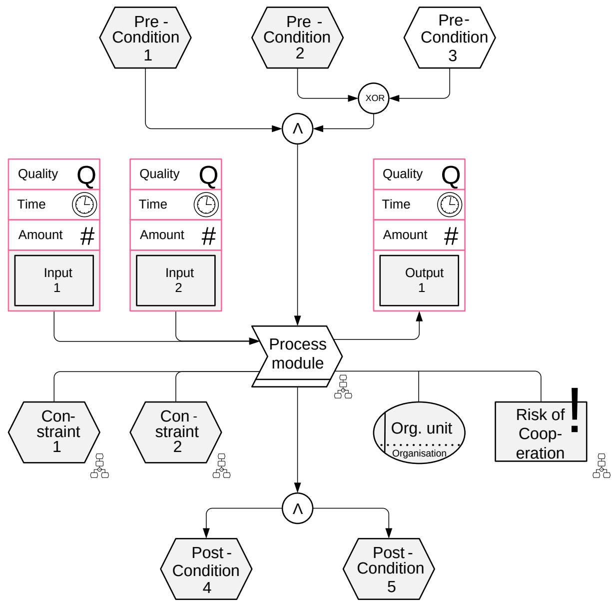


Figure 50: Process modules according to KLEIN, KUSCH & SCHEER[613]

KLEIN, KUPSCH & SCHEER described the modeling of cross-organizational processes in the context of ARIS. As a means to abstract and communicate internal process knowledge to collaboration partners, they propose the use of process modules. A process module can subsume functions as well as sub-processes; it is used to encapsulate a clearly defined, logical unit that represents a part of a business process.[614] Since it can comprise either individual functions or sub-processes, it represents a process aggregation; however, different from public processes, a process module does not display the sequence of externally relevant functions comprised in it. Figure 50 illustrates the different attributes a process module can be annotated with: input and output elements, different types of events representing pre-conditions, post-conditions and constraints of a module, organization units and risks associated with the execution of a module. The pre- and post-conditions as well as the constraints are useful to describe the module's characteristics to collaboration partners.[615] Nevertheless, in the case, that sub-processes are encapsulated by a process module, in addition to the description of the module, a collaboration partner also needs information on the sequence and logic in which input and output elements are required or produced. Thus, in case different organization units are involved in the sub-process, a correlation of the organization units with the input or output ele-

611 His proposal of distinguishing these three phases was already discussed in Chapter 3, pp. 50.

612 Compare WERTH (2006), pp. 217. He used the German term "Prozessbaustein", which can be translated into "process building block".

613 Adapted from KLEIN, KUPSCH & SCHEER (2004), p. 13.

614 Compare KLEIN, KUPSCH & SCHEER (2004), p. 11.

615 These elements are especially interesting in the context of SOA, since they can be used to support an automatic discovery and binding of services.

ments is also necessary. To provide this additional information, a process module should be complemented with a public process model.

MASAK described the interfaces of a service on a technical level. He lists various elements that service interfaces should contain, including: operations offered by the interface (comprising their syntax, their semantics and restrictions), locally defined data types, exception handling, quality of service, pre-conditions of using the overall interface as well as individual elements of the interface, the purpose and the goal of the interface and a user guide.[616] The latter category describes the sequence of interactions between the interface and its partners. Thus, his approach goes beyond a static description of operations, and includes dynamic aspects covered in the process dimension of the AIOS.

CHEBBI, DUSTDAR & TATA described how collaborative workflows can be designed and implemented based on private and public views. In this context, they described so-called *Partner Access Contracts*, which define "the set of external activities" [617] a partner can execute inside a collaboration. More specifically: one access contract comprises for each participating organization the set of operations it offers to the collaboration. Thus, it represents the static elements of a global process; or in other words, the set of (global) functions available in a global process.

LIU & SHEN described private and public processes on a workflow level. In this vein, they also distinguished between *Base activities* as being part of a private process, and *Virtual Activities* as being part of a public process. A Virtual Activity represents an "abstraction of a set of base activities"[618], and for further abstraction may comprise other, previously defined Virtual Activities. Being different from an activity in a private process, "it is not performed, but rather is used to express the progress information of a set of activities"[619].

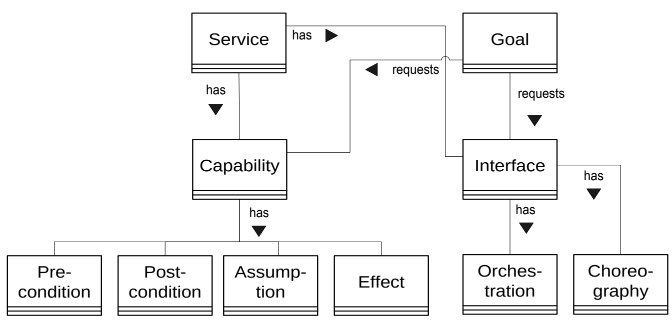


Figure 51: WSMO elements for the description of services[620]

WSMO The Web Service Modeling Ontology (WSMO) is described as a meta-ontology for semantic web services,[621] providing a conceptual framework for semantically describing all relevant aspects of web services in order to facilitate the automation of their discovery, combination and

[616] MASAK (2007), pp. 276.
[617] CHEBBI, DUSTDAR & TATA (2005), pp. 19.
[618] LIU & SHEN (2003), p. 512.
[619] LIU & SHEN (2003), p. 511.
[620] Adapted from W3C (2005).
[621] FEIER ET AL. (2006).

invocation.[622] The core elements for describing web services in WSMO are *Capability*, which describes the functionality of a web service, and one or more technical *Interfaces*.[623] An Interface comprises two parts: the Choreography part, which describes how the service interacts with the service requestor, and the Orchestration part, which describes how the service interacts with other services in order to fulfill the requested functionality. The Capability class is designed to support a (automated) discovery of a service by potential requestors and is composed out of the elements *Pre-condition*, *Post-condition*, *Assumption* and *Effect*. While Pre-condition and Post-condition describe concretely the state of the information received and emitted by the service, the Assumption and Effect elements describe more vaguely the "state of the world"[624] before and after service execution. Interestingly, WSMO distinguishes between goals as seen from the service perspective (described in the Capability element) and goals as seen from a requestor's perspective. The latter is described in the *Goal* class, which references to the searched Capabilities (compare Figure 51).

WSDL The description of web service interfaces with WSDL is similar to the interface description of modules: the operations comprised in the service and the related input and output messages are described. In WSDL, the implementation of the service is explicitly separated from its interface; thus, the interface defined in WSDL can be bound to different endpoints.[625] This separation of implementation and interface of a service also offers the possibility of developing different interfaces for the same service. For example, a private and publicly visible WSDL description for a service could be generated.

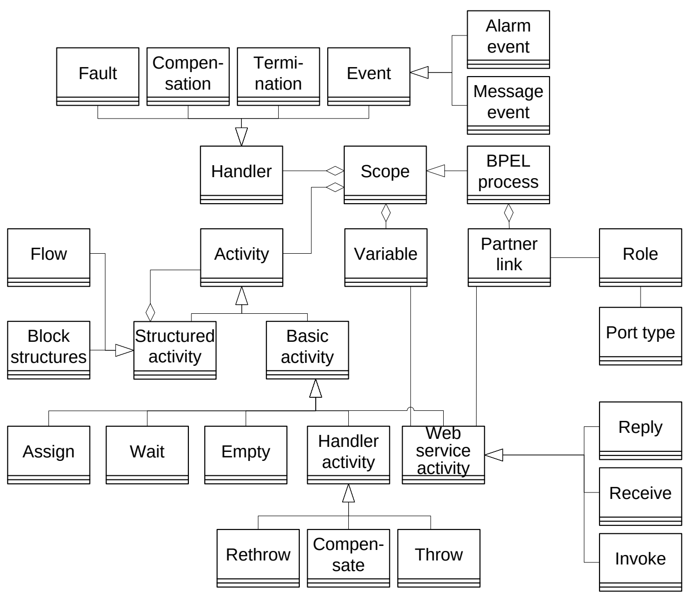


Figure 52: BPEL activities and related elements

---

[622] Compare W3C (2005).
[623] Compare ROMAN ET AL. (2005), p. 88.
[624] ROMAN ET AL. (2005), p. 89.
[625] Compare W3C (2001).

BPEL Business functions appear in three forms in BPEL: First, a BPEL process is always published as a web service; thus, the overall BPEL process can be seen as a function with various operations. Second, the main purpose of BPEL is to connect the functionalities of different web services in one process; correspondingly, each web service being part of the BPEL process can represent a business function – this type of function is implemented via BPEL's web service activities. Third, apart from web service-based activities, BPEL's *Basic Activities* comprise different other activities, including the Wait, Empty and Assign activity. This third type of function rarely realizes a goal on the business level but rather technical functions.[626] Figure 52 also shows that apart from Basic Activities, *Structured Activities* are supported as well. However, since these describe the sequence of Basic Activities, they do not represent business functions but a part of the control flow. Interestingly, BPEL does not only support message-based events,[627] but also time-based events in form of the *Alarm Event*. This event can represent either a point of time (e.g. the event "January the first 2011 has arrived") or duration (e.g. "Three hours have passed").[628] To hide functions in BPEL Abstract Processes, BPEL Activities that are part of a private process can be omitted completely in the public process; they can also be referenced as *Opaque Activities*. An Opaque Activity is an explicit placeholder for exactly one activity of a private process.[629] Apart from hiding a complete function, the information hiding can be restricted to individual attributes of the function as well. Thus, three degrees of functional abstraction exist in BPEL: omit the existence of a private function, indicate the existence of a private function but disguise its nature and its attributes, or restrict the information hiding to the attributes of a private function.

Programming Languages For the description of business functions on the programming level, different forms of modules are used; for example, procedures and functions in procedural programming languages or classes in Object-oriented languages. These concepts resemble the characteristics of technical modules: they hide the implementation of its functionality while exposing available operations as well as input and output parameters. In case of classes, also attributes can be exposed. Refining these measures for information hiding, in Object-oriented languages the visibility of individual attributes and operations can also be specified: private elements are accessible only inside a class, protected elements only inside the class and classes derived from it and public elements are visible for all classes.[630]

4.2.1.2 Conclusions

Table 8 illustrates the different levels of technical granularity of the approaches for the modeling of functions; it also illustrates, that several approaches explicitly support the modeling of functions in collaborations. Furthermore, it is shown that not all approaches offering collaborative views on functions are aiming at cross-organizational scenarios.

---

[626] The Wait and the Empty Activity for example do not produce an output. The Assign Activity on the other hand, which modifies data stemming from incoming documents to produce outgoing documents, produces an output and thus could be seen as a business function.

[627] An example for a message-based event would be "document 'RfQ' from company A was received".

[628] Compare OASIS (2007).

[629] Compare OASIS (2007), p. 149.

[630] Compare BALZERT (2000), p. 200.

Vertical Levels in the Function Dimension The literature review showed, that approaches for describing functions can be found on all three vertical levels of the AIOS:

- Business level: On the first level, functions are described on a business level using, for example, EPC function trees, and goal descriptions. Functions on this level were described by Scheer, Becker et al. and Werth.
- Technical level: On this level, the business models are enriched with details needed on the execution level. This level is described for example by the ARIS design view, Chebbis, Dustar & Tata and Klein, Kupsch & Scheer.
- Execution level: Here, functions are implemented as executable components; therefore, the technical concepts of the previous level are refined into a machine-readable format. BPEL, WSDL and programming languages are examples for concepts applicable on this level.

The literature review also showed that the different approaches proposed different elements to be contained in the function dimension (for example goals, capabilities, events). The adoption of these elements in the AIOS is discussed in the following section.

	ARIS	Becker et al.	Masak	Werth	Chebbi et al.	Liu & Shen	Klein et al.	WSDL	BPEL	WSMO	Object-oriented Languages
Business level	●	●		●						○	
Technical level	●		●	○	●	●				○	
Execution level	●				●	●	●	●	●	○	●
Explicit collaboration support				●	●	●	●				
Support of collaborative views				●	●	●	●	○	○		○

● / ○ / Blank : Criterion is completely / partly / not fulfilled

Table 8: Overview of approaches for describing functions

Collaborative Views in the Function Dimension The usefulness of hiding private system elements behind a static interface is hardly arguable, especially in the context of collaborative business processes.[631] In addition to the black box principle, the following mechanisms were used in the approaches described above:

- Private and public views on functions: Werth described “process building blocks” of collaborative business processes with an internal and an external interface.[632] A corresponding mechanism is supported by WSDL, which allows relating different interface descriptions to one service implementation. Liu & Shen describe public visibly activities that represent abstractions of private functions. In addition, BPEL and Object-

[631] Compare also pp. 34.
[632] Compare Werth (2006), pp. 217.

oriented programming languages allow defining the visibility of selected operations or attributes of functions.

- Global functions: CHEBBI, DUSTDAR & TATA described functions that serve as agreement between collaboration partners, describing which partners have to fulfill which functions.[633]
- Functions as views on sub-processes: KLEIN, KUPSCH & SCHEER propose to use process modules to encapsulate private sub-process, only exposing the static interfaces of these process parts to collaboration partners.[634]

While the first two concepts will be supported by the AIOS, the usage of process modules is disregarded: their usage bears the risk of losing necessary information about the sequence in which functions comprised in the module are to be invoked. As the literature review indicates, it can be useful to distinguish between public and private functions, similar to the distinction between private and public processes. The development of globally known and accepted reference models that can be mapped to local service models also appears useful. Thus, all three collaborative views will be applied in the function dimension of the AIOS.

4.2.2 Metamodel for the Function Dimension

Based on the reviewed concepts for describing functions on different levels of technical granularity and for the development of private, public and global views on functions, now the metamodel for the function dimension of the AIOS can be defined.

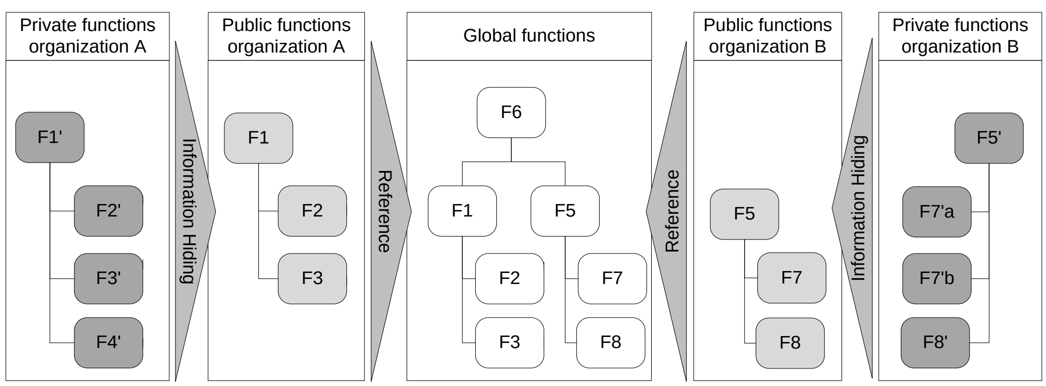


Figure 53: The private/public/global concept applied to functions

4.2.2.1 Relationship Between Collaborative Views

Figure 53 exemplifies the principle of private, public and global views applied to functions. The function bundle in the global view represents a reference function tree, comprising sub-functions that are referenced by organization A and B.[635] The graphic also illustrates, that the internal descrip-

[633] Compare CHEBBI, DUSTDAR & TATA (2005).
[634] Compare KLEIN, KUPSCH & SCHEER (2004).
[635] For example, F6 could be a globally known function for "car production", where F1 represents "delivery of car parts" and F5 the function "assembly of parts".

tion of the function can differ from the one communicated to the partners. As in the case of process abstraction, at least two reasons can motivate the hiding of private functions: the external partner is not interested in this knowledge or the owning organization wants to disclose the knowledge for internal reasons. For example, the undisclosed function F4 could represent a function that investigates the financial standing of the collaboration partner. Thus, organization A is able to offer various variations of internal functions to its partner organizations. The relation of the private and public functions of organization B illustrates, that one public function (F7) can also be mapped to various internal functions (F7'a and F7'b). On a generic level, the relationships between private, public and global functions can be described as follows (compare also Figure 54):

Relationships Between Functions and Operations Following the definition of SCHEER,[636] a function is seen as a set of operations that are applied to objects for the purpose of supporting one or more goals. Thus, one function can comprise 1 to many operations. An operation should contribute to at least one goal (and thus to a function), but could also be comprised in various functions; thus it can be related to 1 or many functions. One function can contain 0 to many sub-functions or be comprised in 0 to many higher-level functions.

Relationships Between Private and Public Functions One private function can be published in 0 to many public functions. A public function, on the other hand, can be realized by 1 or many private functions. Since operations represent the lowest level of granularity in a function hierarchy, they cannot be split further; thus, in case a relationship between the associated private and public functions exists, one private operation must be related to one public function (and vice versa).

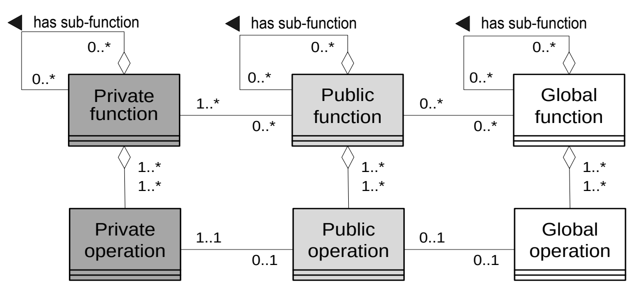


Figure 54: Generic metamodel for private, public and global functions

Relationships Between Public and Global Functions As described in Chapter 3,[637] in general, two relationships between public and global models can be distinguished: In the first case, a public function is related to one globally known reference function (*is-a* relationship). In the second case, the global function represents a set of sub-functions and each public function is related to a sub-function of the global function (*part-of* relationship). In both cases, one public function can be related from 0 to many global functions. Also in both cases, a global function can be related to 0 or many public functions (though in case of 0 related public functions, the global function is not referenced in the collaboration). Similar to the relationship of private and public functions, in the relationship of public and global functions each associated operation should only be mapped to one other operation at most.

[636] Compare SCHEER (2000), p. 22.
[637] Compare pp. 111.

4.2.2.2 Connecting Conceptual and Execution Level

In order to capture the characteristics of functions both from a business and a technical perspective, the generic metamodel displayed in Figure 54 now is enriched with elements of the various approaches described in the literature review.

Pre- and Post-Conditions Apart from functions and their sub-functions, also their input and output, their goals as well as related application systems are modeled in the function dimension of ARIS. In order to support the automatic discovery of functions, in WSMO, the modeling of goals is operationalized by describing the envisioned effects of a function in the form of pre- and post-conditions.[638] The explicit modeling of pre- and post-conditions of functions in the context of collaborations was also proposed by KLEIN, KUPSCH & SCHEER.[639] The modeling of pre- and post-conditions of functions via events is also included in ARIS: In the EPC events and functions have to alternate, thus each function has a pre- and a post-condition. Usually just one event is modeled; nevertheless, SCHEER also describes the possibility of describing hierarchies of events, which enables the modeling of complex pre- and post-conditions.[640]

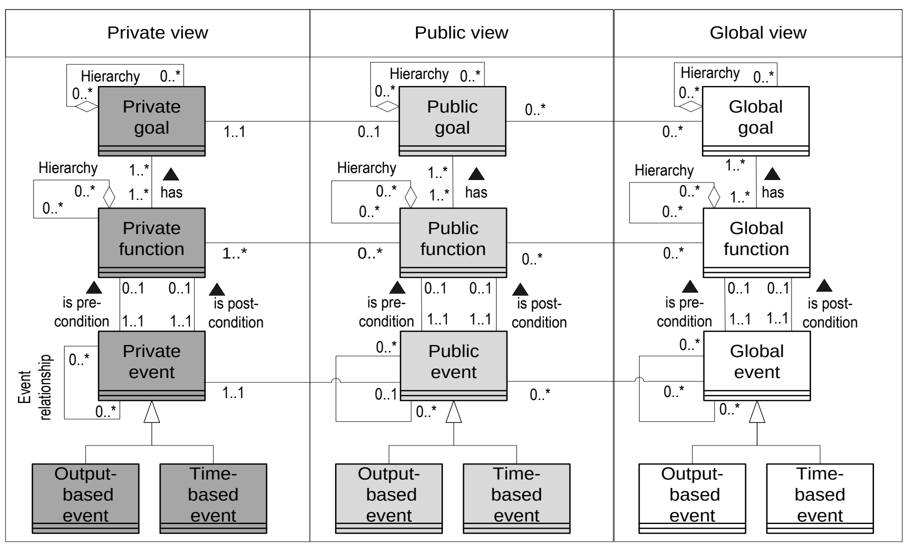


Figure 55: Metamodel of the function dimension of the AIOS

Further Types of Events To enable a mapping to execution level standards like BPEL that distinguish between *message-based* and *time-based* events,[641] the AIOS function dimension supports this distinction. However, going beyond message-based events where a function is triggered by a message (i.e. the output of a previously finished function), the broader concept of *output-based* event is used, where output refers to any result of a function. In the context of automated collaborative business processes, this will usually be digital messages, but the concept of output-based events

638 Compare W3C (2005).
639 Compare KLEIN, KUPSCH & SCHEER (2004).
640 Compare SCHEER (2000), pp. 128.
641 Compare OASIS (2007).

would also comprise material pre-conditions or post-conditions of functions.[642] The resulting metamodel is illustrated in Figure 55. Note, that since operations can also be represented as sub-functions, they are not explicitly comprised in the metamodel.

Two Ways of Describing Goals The definition of a function already implies a close relationship between goals and functions.[643] Correspondingly, in the business-level approaches as well as in the SOA-related approaches (where services are to be discovered based on their functionality), goals and capabilities were correlated to functions. In WSMO for example, a service is correlated with goals and capabilities, and in ARIS goal hierarchies are an integral part of the function dimension. Here, two levels of goal descriptions can be distinguished: First, goals can be described explicitly in an informal way, detached from individual operations or input/output related to a function. This approach is realized, for example, by the modeling goal hierarchies in ARIS or by the rather imprecise descriptions of Assumption and Effect of a service comprised in WSMO. Second, goals of a function can be described implicitly in a formal way, based on the sum of tangible results of a function. This approach is supported by describing pre- and post-conditions of services in WSMO; it is also supported in ARIS by modeling input and output of functions as well as events that represent pre- and post-conditions. Both types of goal modeling are supported by the AIOS: An explicit class for goal hierarchies enables the collaboration partners to describe the objectives of functions on the business level, while pre- and post-conditions of functions describe goal implicitly on an operational level.

4.3 Organization Dimension

In this section, the metamodel for the AIOS organization dimension is developed. Since the usage of organization models in the context of collaborations is not as common as process models, in a first step, the need for such models in the context of SOA and collaborative business is described. Secondly, to prepare the following literature review, essential elements of organization models are described. Based on the results of the literature review, thirdly, an organization metamodel is described that supports private, public and global views and serves as a connection between business level and execution level models.

4.3.1 Preliminaries

In the context of business process management and enterprise modeling, the modeling of organization structures is an established technique.[644] Thus, in the organization dimension of ARIS, organization units, positions and roles comprised in an enterprise as well as their relationships are displayed. This business-oriented organization view describes the hierarchical organization, e.g., the organization units and the communication and reporting relationships among them. Stressing the need for an explicit organization modeling, SCHEER points out that in enterprise applications, the hierarchical organization model often is not as clearly defined as data, functions and processes. Instead, in business process models it is hidden behind a “flat” vocabulary (e.g. “sales group”),

[642] For example, the event “transported goods were damaged” could represent a pre-condition for a “repair” function.
[643] Compare also the description of business functions in Chapter 2, p. 16.
[644] Compare for example SCHEER (2000), WESKE (2007) or ROSEMANN & ZUR MÜHLEN (1999).

while a more detailed description of organization units and their relationships are anchored only in the programs. When implementing a business application, however, it would be crucial to reconcile organization structures.[645]

Need for Organization Modeling in the Context of CBP and SOA This argument is even more true for cross-organizational scenarios, where stakeholders do not share the same vocabulary and thus an increased need exists, to describe explicitly the characteristics of actors involved in a collaborative business process. In addition to the lack of shared vocabularies, in CBP often the trust level is lower than in intra-organizational processes, which is another motive for modeling the roles and rights of involved actors. Nonetheless, research on organization models in the context of business process automation has been relatively limited. Thus, despite the importance of the description of roles and rights in cross-organizational scenarios, little work exists on organization modeling in the context of CBP.[646]

Also in the context of SOA, the importance of organization modeling has been discovered: In the development of SOA, both organization structures and processes should be modeled, but methods for organization modeling in the context of SOA are lacking.[647] Thus, ORTNER stated that an organizational theory for computer systems would be needed. Indicating the need for organization modeling on a technical level, he also states that existing modeling languages like UML would allow for a more precise description of organization structures than concepts known from business administration.[648] However, WOHED ET AL. criticize the lacking support for organization elements also in technically-oriented process modeling languages like UML and BPMN.[649]

In the context of CBP and SOA, two functions supported by organization models are of special interest:

- Discovery of roles and their characteristics: In SOA, rather than actors, functionalities of services are described and queried. Nevertheless, in some environments the type of actor executing a service is as important as the functionality he provides, since the actor implies essential characteristics of the function. This applies, for example, to eGovernment scenarios, where roles as judge, attorney of the state or police officer are well defined and correlated with an expected service. Here, the relevant criterion for selecting a service provider is first the type of actor, then the function itself. An example for a corresponding query could be "show me an *attorney of the state* that can *issue* a *European Arrest Warrant*".

---

645 SCHEER (2000), p. 53.

646 Compare also VAN DER AALST, KUMAR & VERBEEK (2003), p.5. An exception are LEGNER & VOGEL (2008), who describe various levels of agreement necessary for achieving interoperability, one of them being "organizational and role model".

647 Accordingly, KONTOGIANNIS ET AL. (2007), p. 4, name "models for organizational structures in SOA environments" as a research challenge.

648 Compare ORTNER (2008).

649 Compare WOHED ET AL. (2006), pp. 174. More specifically, they state that the resource perspective is not well covered in either UML Activity Diagrams or in BPMN, and that those BPMN elements which support the organization dimension are not fine-grained enough to support the execution of processes, where resources have to be allocated to tasks at run time.

- Assurance of role characteristics: Different from internal processes, in collaboration less informal knowledge about roles and rights of partner organizations exist. Moreover, the trust level is lower than in intra-enterprise processes. In consequence, the participating partners require reliable, explicit information about the rights and roles of collaboration partners. For example, before a Public Administration transmits a document, it might consult an organization model regarding the question "do members of the *partner organizations IT department* posses the right to *transfer documents* classified as *confidential* to *third parties*?"

Since mechanisms for dynamic discovery are not in the scope of this work, in the following, the focus lies on the second function. However, since the metamodel for the organization dimension of the AIOS is not restricted to access rights but also covers the positions and roles related to business functions, the first function will implicitly be supported as well.

Essential Elements of Organization Models The content and the design of organization structures are discussed in various contexts, including business administration,[650] IT security and in the implementation of workflows. Depending on their background, the corresponding organization models have different but overlapping objectives: On the business level, the modeling of organization units, actors and their responsibilities aims at the analysis and modification of organization structures. In the context of workflow management, pending tasks have to be allocated to the resources responsible for their execution. For IT security, fine-grained models have to describe the rights of actors and roles for accessing specific resources. Corresponding to the different vertical levels of the AIOS, both coarse-grained, business-level models and fine-grained, technical models are relevant. Representing a foundation for the following review of organization metamodels, these essential organization elements should be known:

- Function, actor, resource and object: In comparison to process models, instead of on the *how*, organization models rather focus on the *who* and (to a lesser degree) on the *what*. To specify the "what", organization models also have to refer to the business function as the atomic work entity. Nonetheless, other than models from the IT security context, organization models from enterprise modeling usually refer only implicitly to functions, but do not comprise an individual class to represent function.[651] An actor is understood as a person or a software application, which executes a function; a resource, on the other hand, is a means needed by the actor to execute the function (e.g. a hammer needed to pound a nail into a wall). In the context of computer systems security, it is often necessary to specify the object that is modified by a function, e.g. the document that an actor is permitted to modify.
- Role: Actors are rarely directly allocated to particular functions, because the organization model would have to be updated if these persons were transferred or if they left the

[650] Compare for example WÖHE (1996), pp. 183.
[651] This is due to the separation of enterprise views, which separate functions and organization elements in different views. In ARIS, these views are joined in the control view, where functions and organization units are correlated (compare SCHEER, 2000, p. 105).

company; instead, roles are used. A role describes a set of functions, which can be assigned to an actor.[652]

- Position and organization unit: A position is the smallest unit within the organization structure; it normally comprises a set of functions, which fall in the responsibility of one person compassing the (full) working capacity of this person.[653] An organization unit is the coarsest grained element in the organization structure and embodies a set of positions or roles. It represents departments or divisions of an enterprise, like for example "purchasing" or "human resources" department. A project can be seen as a temporarily limited variant of an organization unit.[654] Various types of relationships can exist between positions as well as between organization units or roles. This comprises relations regarding the right of command, substitutions (e.g. role A can substitute B) or functional relationships (e.g. the staff unit gives counsel to the management unit). In the context of roles and rights, the delegate relationship is comprised in various metamodels for organization models.
- Permission, rule and policy: These terms are relevant in the context of access control and authorization. There, a permission is defined as an approval to perform an operation on one or more protected objects.[655] A policy is defined as "a set of rules indicating which subjects are permitted to access which resources using which actions under which conditions", where a rule is defined as the combination of a condition, a target and an effect.[656]

4.3.2 Approaches for Collaborative Organizational Views

In the following, existing organization metamodels are evaluated regarding their suitability to form part of the AIOS organization dimension metamodel. To support the three vertical levels of the AIOS, organization metamodels from different levels of technical granularity are reviewed.

4.3.2.1 Literature Review

ARIS As Figure 56 illustrates, ARIS' organization metamodel comprises organization units, positions and roles, which correspond to the basic organization elements introduced in the previous section. Thus, *Organization Unit* is an umbrella for different types of organization elements, whereas a *Position* represents the finest grained unit within the organization structure, defined by the function amount that an individual employee can handle. Structural relations between organization units are represented in an explicit class (*Organization Structure*). The metamodel illustrates, that *Qualifications* are seen from two sides: To describe the *Requirements,* e.g. the skills required for a certain position, and, in form of the class *Profile*, to describe the skills attributed to a certain user group. Further, organization units are divided into human output producers and technical output

---

[652] Different understandings of roles are discussed below, compare p. 159.
[653] Compare SCHEER (2000), p. 57.
[654] Compare ROSEMANN & ZUR MÜHLEN (1998), p. 82.
[655] ANSI (2004), p. 4.
[656] OASIS (2005), p. 4. There is also mentioned, that in the field of access control and authorization several closely related terms are in use, e.g. the terms permission, privilege, authorization would by used synonymously to the term rule.

producers, the latter is again divided into material processing and information processing producers. The metamodel does not contain a correlation of organization units or roles to functions; however, this correlation is described in the ARIS control view.[657]

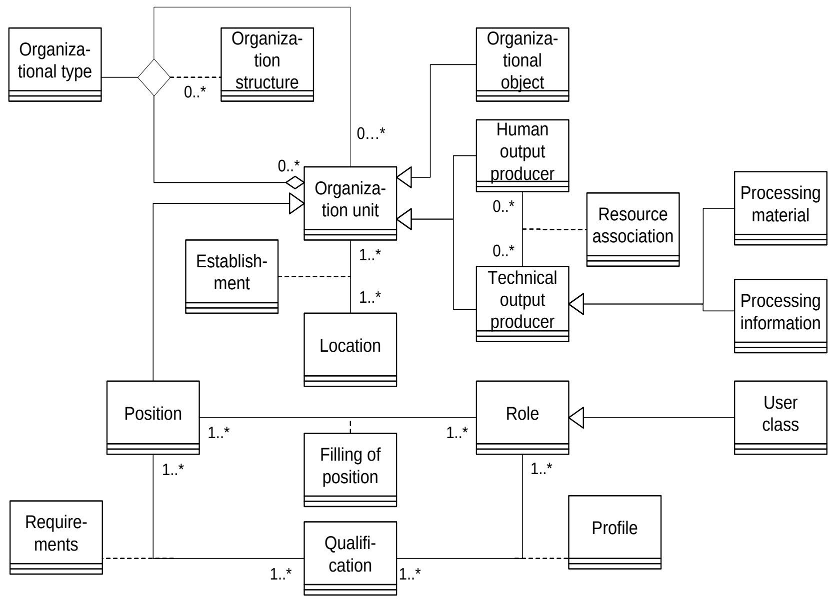


Figure 56: Metamodel of ARIS' organization dimension[658]

WESKE recently described a coarse-grained organization metamodel in the context of business process management, which contains the basic elements of the ARIS metamodel: *Position*, *Role* and *Organization Unit*. Similar to the "Human output producer" of ARIS, here the class *Person* is used; further, *Organization Unit* is refined by the class *Organization team*. In difference to SCHEER'S metamodel, it does not specialize these elements into finer grained elements. Neither does it offer to display hierarchical relations between organization units or positions.

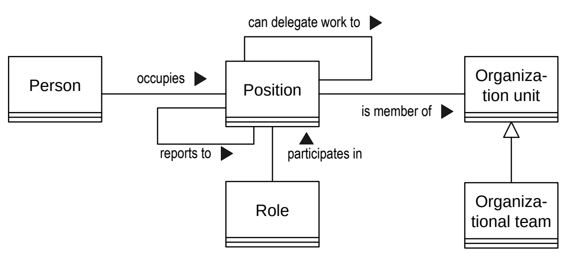


Figure 57: Organization metamodel from WESKE[659]

ROSEMANN & ZUR MÜHLEN pointed out a lack of research on organization modeling in the context of workflow management and responded with the creation of a reference metamodel for organization structures. To create a basis for this, they compared the organization models comprised in

657 Compare SCHEER (2000).
658 SCHEER (2000), p. 56.
659 Adapted from WESKE (2007), p. 103.

several Workflow-Management-Systems.[660] Their metamodel resembles the main elements of the ARIS metamodel, e.g. *Organization Units* comprise *Positions*, which are filled by *Persons* (represented in ARIS by a “user class”). In difference to ARIS, they also use a *Position Type* class, which can be seen as a “meta-position”, comprising various positions with similar characteristics.[661] Further, they distinguish between two kinds of roles: *Qualifications*, which express the potential of a person, and *Competences*, which comprise functions that a person may execute inside an organization. The distinction between different types of roles is useful, since a competence – in the sense of a permission of a role to execute a certain function – is relevant for workflow execution and IT security, while a qualification – in the sense of abilities a role – is relevant for organizational engineering. The class *Position Type* though appears to be redundant; first, the relation between position and position type is unclear. The example they provided implies rather an “is a” relationship or an instantiation. If the class Position Type is only used to group similar positions, this could also be displayed by a recursive relationship on the class position. Different from ARIS, only human actors are comprised in the metamodel.

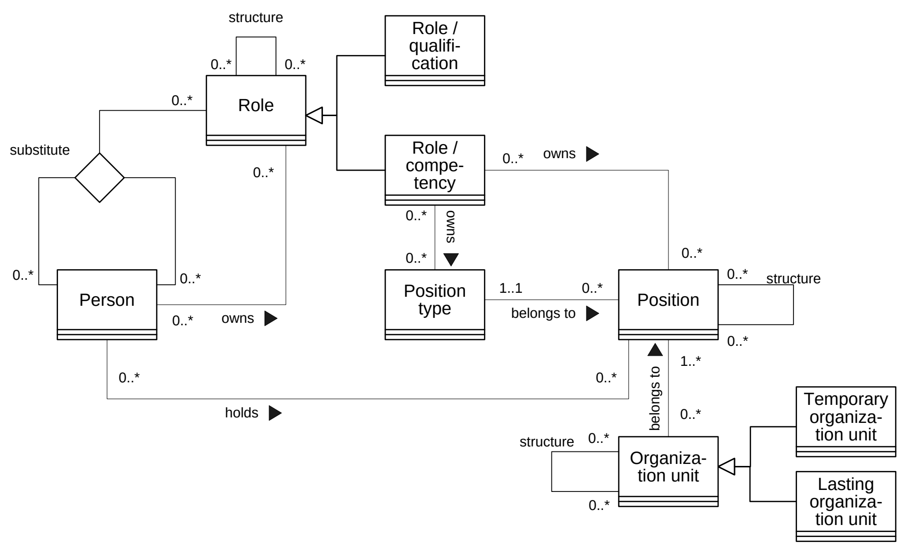


Figure 58: Organization metamodel from ROSEMANN & ZUR MÜHLEN[662]

THELING ET AL. created an XML-based organization metamodel as a complement to an XML representation of BPMN. Their metamodel is very coarse-grained: The only organization element it comprises is *Organization Unit*; in addition to this, only three different types of relationships between organization units are described (hierarchy, contract and delivery).[663] Building on this work, SARSHAR & THELING proposed a method for modeling organizations in the context of collaborative business processes. Similar to the approach followed in the AIOS, they distinguished between globally visible organization units and internally visible organization units. However, they focused on the modeling of global organization units, and not on collaborative views on organization units.

660 Compare ROSEMANN & ZUR MÜHLEN (1999).
661 They provide the example of the position type “secretary” and the position “secretary of Professor Smith”.
662 ROSEMANN & ZUR MÜHLEN (1999), p. 81.
663 Compare THELING ET AL. (2005).

Thus, the only relationship between collaborative views indicated in their work is aggregation: a globally visible organization unit can be refined into various (finer-grained) internal organization units.[664]

CHIU ET AL. suggested applying public views to different flow types needed in system integration, one of them being "Security Flow". Here, they distinguished between an (internal) Security Flow and a corresponding *Security Flow View*.[665] However, the description of both remains abstract. They did not go beyond proposing a combination of BPEL, WS-Security and the Security Assertion Markup Language; further, no explicit distinction between internal and external (nor global) security related information is made in the Security View.

POP* was developed in the ATHENA project as an intermediate format to be used in the transformation between different enterprise modeling languages. For example, an ARIS model could be transformed to POP* and the resulting POP* model could be transformed to a third language required by a collaboration partner.[666] Thus, it was designed to be a common denominator between various enterprise models used in industry and can be understood as a meta-metamodel of enterprise modeling languages.[667]

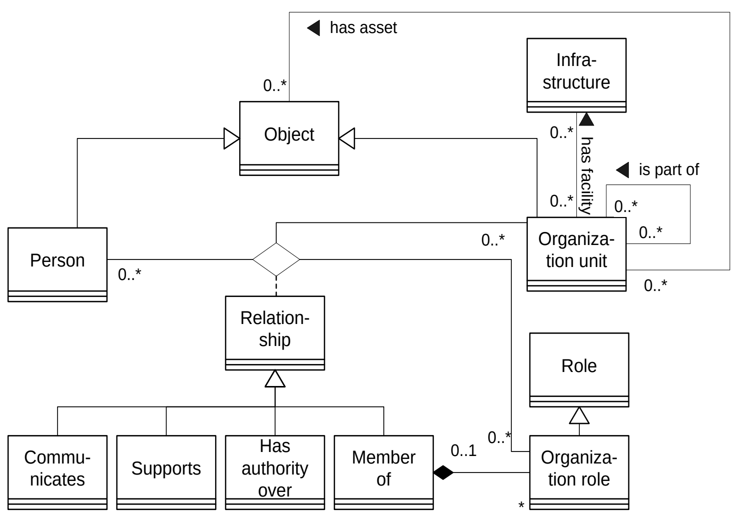


Figure 59: The POP* organization metamodel[668]

POP* comprises different enterprise dimensions, including a process, a decision and an organization dimension. Figure 59 illustrates the metamodel of POP*'s organization dimension. Similar to ARIS, in POP* the element *Organization Unit* is used to represent organizations and parts of organizations. *Person* designates an individual human being. To express the relationship between persons and organization units, the *Member Of*, *Has Authority Over*, *Supports* or *Communicates* relationship can be chosen. The relationship *Has Asset* can be used to correlate certain assets to an organization

664 Compare SARSHAR & THELING (2007).
665 Compare CHIU ET AL. (2005).
666 Compare also ZIEMANN ET AL. (2007)
667 Compare ATHENA (2006).
668 Adapted from ATHENA (2006), p. 23.

unit, like for example computer hardware or rooms. A peculiarity of this metamodel is the prominent role of the relationships which are displayed in distinct classes (communicates, supports, has authority over, member of). However, due to the great variety of possible relationships among organization elements,[669] the usefulness of explicitly modeling each possible relationship seems questionable. Different from other organization models, the POP* organization dimension contains various connections to other enterprise modeling views: *Role* and *Object* are both generic elements also being used in the other dimensions.

Organization Structure Metamodel Complementary to process standards like BPMN, the OMG is specifying a so-called Organization Structure Metamodel.[670] Though this specification is still in progress, it shows, that also in the technical branch of business process management, the need for explicit organization models complementary to process models is acknowledged.

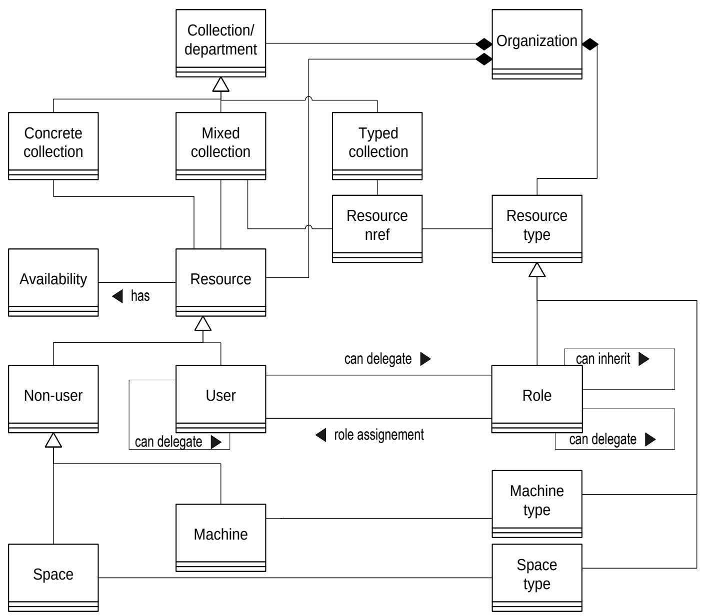


Figure 60: Organization metamodel for workflows from VAN DER AALST ET AL.[671]

VAN DER AALST, KUMAR & VERBEK described an organization metamodel for workflows and implemented this in XML.[672] Their objective was to complement an XML-based process description language with an organization dimension. In comparison to the organization dimension of ARIS, the described metamodel contains more elements relevant for workflow execution, e.g. a class describing when resources are available to execute a class and the possibility of delegation between users and roles. As described by the metamodel displayed in Figure 60, they understand an *Organization* as an aggregation of three entities: *Resources*, *Resource type*, and *Collection/Departments*. According to their metamodel, the class Organization, which is displayed as a

669 Apart from those comprised in the POP* metamodel, other relationships could be for example "delegates" or "is qualified for".
670 Compare OMG (2006). The OSM is also described in CUMMINS (2005).
671 Adapted from VAN DER AALST, KUMAR & VERBECK (2003), p. 604.
672 Compare VAN DER AALST, KUMAR & VERBECK (2003).

composition of departments, represents the overall organization. The fact that hierarchies of organization units cannot be modeled, displays the background of workflow execution, while enterprise modeling is neglected. Unusual are the different types of Collection classes contained in the metamodel, which are motivated with the argument that most Workflow-Management-Systems only allow allocating one resource to one task: To allow the allocation of a group of resources to one task, the collection class is used. Three types of collections are distinguished: *Concrete Collection*, representing a group of resource instances, *Typed Collection*, representing a group of resource types, and *Mixed Collection*, comprising both instances and types of resources. However, the usefulness of distinguishing these collections types in three separate classes seems questionable, since these could also be represented implicitly, e.g. via relations between a resource group to roles and persons. If the metamodel contained a *task* class, no "collections" at all would be needed, since arbitrary numbers of users could be correlated to a task. The *Availability* class describes when resources are available or reserved.

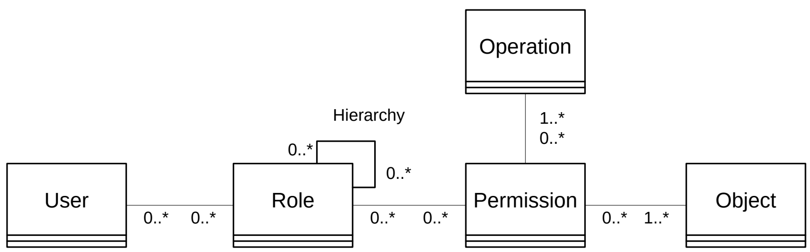


Figure 61: An RBAC metamodel

RBAC and XACML The concept of Role Based Access Control originates from the area of IT security. It restricts systems access to authorized users by specifying which users have the role required to access a certain resource.[673] Core RBAC, as defined by ANSI[674], comprises the following five elements: users, roles, objects, operations and permissions.[675] Figure 61 illustrates that each role is associated to permissions that specify which operations a role can execute on certain objects. As the recursive relationship in the metamodel indicates, RBAC also supports hierarchies of roles. The basic RBAC model can be mapped easily to elements of organization models, e.g. actors, role hierarchies, functions and input/output objects related to business functions.

XACML is a language for specifying security policies supporting the RBAC concept.[676] The main elements of the XACML model are rule, policy and policy set, where a *Rule* represents the most elementary unit of a policy. A rule is evaluated based on its *Condition*; depending if the target meets this condition, the *Effect* of the rules is either "permit" or "deny". The target of a rule is composed of four elements: A *Subject* that is supposed to execute a certain *Action* on a *Resource* in the boundaries of a certain *Environment* (spatial or temporal).[677] In order to exchange rules between

[673] Compare FERRAIOLO & KUHN (1992).
[674] ANSI (2004).
[675] Compare OASIS (2005), p. 16.
[676] In OASIS (2005), p. 23, the following example of a XACML policy is provided: "Any user with an e-mail name in the 'med.example.com' namespace is allowed to perform any action on any resource".
[677] For example, the subject "Mr. Smith" is allowed to execute the action "read" on the resource "document 'European Arrest Warrant'" in the environment "only after July 2010 in a member state of the European Union".

actors, various rules can be encapsulated in a policy, which itself can be part of a policy set. The *Policy Set* serves as a container of various XACML elements and can be arranged in a hierarchical order. If a nested policy set is evaluated, the *Policy Combining Algorithm* describes logical operations to compute one overall result based on the various sub-results of the nested policy sets. OASIS also specified an RBAC profile for XACML, which enables expressing RBAC policies in XACML. Therefore, five core RBAC elements are mapped to XACML elements: *Users* are implemented using XACML subjects and *Roles* are expressed using one or more XACML subject attributes. *Objects* are represented as XACML resources, *Operations* are expressed using XACML actions and *Permissions* are expressed via the XACML policySet.[678]

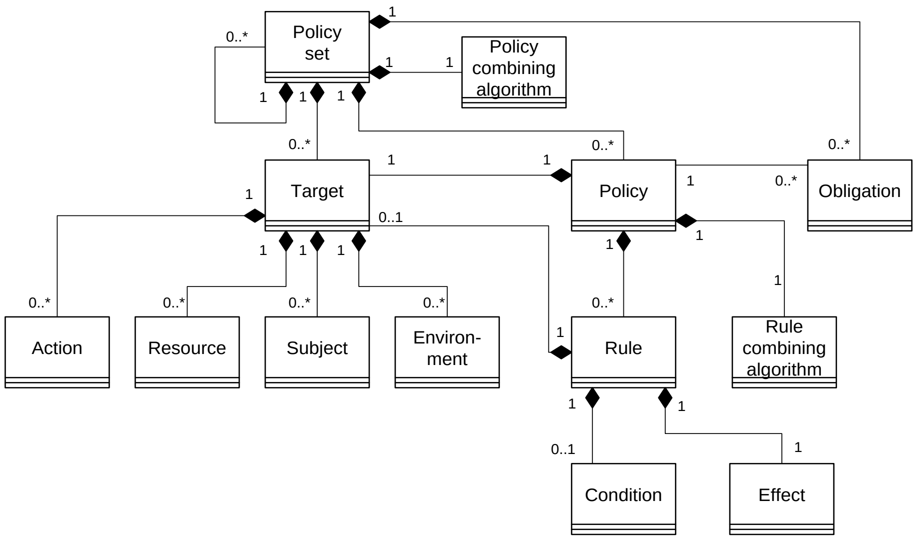


Figure 62: XACML metamodel[679]

Distributed RBAC and Distributed XACML FREUDENTHAL ET AL. questioned the applicability of the traditional RBAC concept to collaborative scenarios and proposed "distributed RBAC" as an alternative.[680] The traditional RBAC system relies on a central, trusted computing base, administered by a single authority, which contains the entire organizations security policy. Therefore, it would not scale to large numbers of anonymous users existing in collaborative scenarios. To overcome this obstacle, FREUDENTHAL ET AL. proposed a decentralized access control mechanism where permissions can be delegated in a transitive fashion. For example, an actor A who has been granted the permissions associated with a role R, can be enabled to delegate the role R to actors B and C. The resulting sequence of delegations across various actors is called a delegation chain. The set of distributed roles (d-Roles), which constitutes the chain, represents a layer of abstraction between the roles of different organizational domains. In other words, "the d-Role concept elegantly encapsulates the set of knowledge that needs to be shared among collaborative partners"[681]. This

[678] Compare OASIS (2005 RBAC), p. 16.
[679] OASIS (2005), p. 19.
[680] Compare FREUDENTHAL ET AL. (2002).
[681] LEE (2007).

statement illustrates, that a d-Role has a similar function in the organization dimension as that of a global process in the process dimension.

Building on the concept of d-Roles and the RBAC profile of XACML, LEE & LUEDEMANN proposed a collaborative extension of XACML[682]. As illustrated in Figure 63, they followed the understanding of the RBAC profile that XACML subjects correspond to roles. [683] In summary, their concept supports global roles in the form of d-Roles – illustrated in the solid box of Figure 63 – and private roles. While d-Roles are divided into requesting and providing roles, private roles are divided into roles played by internal stakeholders (*Internal Role*) and roles played by partners (*External Role*). Similarly, they divide the XACML resources into internal and external resources. Though not visible in the metamodel of Figure 63, not only the roles of external stakeholders, but also the Internal Role elements have to be related to the d-Roles. Accordingly, the procedure to develop the roles needed in collaboration comprises the following steps: Mapping the relevant elements of the requestor's domain to d-Roles, mapping the elements of the provider's domain to d-Roles and ensuring that the d-Roles of requestor and provider match each other.[684]

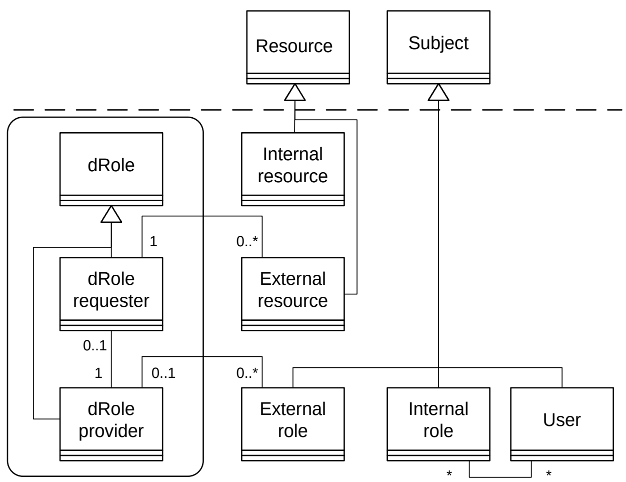


Figure 63: Collaborative extension of XACML[685]

BPEL4People BPEL 2.0 only provides a coarse-grained role concept in form of *partnerLinkTypes*. A partnerLinkType describes the conversational relationship between two web services. Therefore, each partnerLinkType comprises two roles: the role that the BPEL process plays and the role the partner web service plays, for example, "Buyer" and "Seller".[686] Thus, the role concept is based on the interacting services (more precisely, on the interfaces of the interacting processes), and does not distinguish between individual actors in services. A role concept that only assigns roles to process interfaces but not to individual functions is too coarse-grained both for role based access

682 Compare LEE & LUEDEMANN (2007).
683 Note, that the classes *resource* and *subject* displayed above the dotted line in the metamodel are part of the XACML metamodel displayed in Figure 60.
684 Compare LEE & LUEDEMANN (2007), p. 87.
685 Adapted from LEE & LUEDEMANN (2007), p. 86.
686 A role also specifies the WSDL port type that is referred to in the BPEL process. Thus, a 1:1 relationship exists between a WSDL port type and the assigned role. Compare OASIS (2007), p. 36.

mechanisms and for describing competencies of stakeholders involved in the process. This weakness of BPEL regarding the support of the resource dimension is acknowledged; in response, additional standards like *BPEL4people* and *WS-HumanTask*[687] were created.[688] However, both specifications focus on integrating human tasks – as opposed to batch tasks executed by web services – in BPEL, but not on providing a comprehensive organization metamodel complementary to BPEL.[689] Accordingly, these standards do not comprise an authorization framework that would specify the rights of resources, e.g. what actions a resource can undertake at run time.[690] In consequence, it was suggested to define a native extension of BPEL4People/WS-HumanTask, which would provide explicit, implementation-independent concepts "that enable the direct and explicit integration of access control relevant information"[691].

	Sarshar & Theling	ARIS	Weske	Roseman & zur Mühlen	van der Aalst et al.	POP*	RBAC	XACML	Collaobaritve XACML
Hierachical org. units	●	●	○	●	○	●			
Positions		●	●	●					
Hierarchical roles (responsibilites)		●	○	●	●	○			
Hierarchical roles (permissions)[692]		○		○			●	●	●
Objects		○				●	●	●	●
Actors (human and IT) [693]		●	○	○	●	○	●	●	●
Delegation of roles					●				●
Correlation to functions		●					●	●	●
Collaborative views	○								●

● / ○ / Blank: Requirements are completely / partly / not fulfilled

Table 9: Comparison of metamodels for internal organization structures

687 Compare OASIS (2008 B4P) and OASIS (2008 HT). Note, that both standards are complementary and closely interlinked.

688 Compare for example Wohed et al. (2006), p. 175.

689 This was confirmed by Russel & van der Aalst, who analyzed BPEL4people and WS-HumanTask regarding their support of work distribution and resource management. They found that these standards provide a relatively poor organization model, which shows lacks in establishing relationships between organization units, functions and resources.

690 Compare Russel & van der Aalst (2008).

691 Mendling, Ploesser & Strembeck (2008), p. 10.

692 For a full support of access control roles, the metamodel should relate roles with objects and functions; the requirement was judged as partly fulfilled, if the approach distinguished between roles in the sense of permissions or responsibilities.

693 This requirement was judged as completely fulfilled, if the approach supports both human and automated actors.

4.3.2.2 Conclusions

Table 9 provides an overview of the organization metamodels and the concepts they support. While the organization models on the left-hand side stem from an *enterprise modeling* background and display organization structures best, organization models from the *security area* (on the right-hand side of the table) provide a detailed role concept referring to actors and tackled objects. Between both types, the *workflow*-related organization models are depicted, which focus on the allocation of tasks to resources. Apart from these, further differences among the approaches for modeling organization elements can be found:

Different Levels of Technical Granularity Corresponding to their origin (enterprise modeling, workflow modeling and IT security) the reviewed organization models exhibit different levels of technical granularity. While on the business level, the models focus on the skills of actors, models from the area of IT security focus on the access rights of actors. Though it is acknowledged that in systematic software development also the organization dimension has to be modeled on different levels of granularity,[694] the vast majority of model-driven software development approaches disregard the organization dimension.[695] Nevertheless, the concepts for organization modeling described above can be categorized in the three AIOS levels of business, technical and execution level as follows:

- Business level: Here, organigrams known from enterprise modeling are used to describe main organization units and their relationships from a business perspective. The focus lies on correlating organizational roles with skills and responsibilities for business functions. This level was tackled, for example, in the approaches of SCHEER, WESKE, POP*, SARSHAR & THELING and ROSEMANN & ZUR MÜHLEN.
- Technical level: Here the business level model is enriched with information necessary for the execution of business processes. Thus, finer grained models are required that specify actors on the instance level as well as the functions executed by them.[696] This can include information on the detailed availability of resources or the possibility to delegate tasks as found in the workflow-related approaches of VAN DER AALST, KUMAR & VERBECK. Apart from roles and their competencies, on this level the detailed rights of roles are also described, as done, for example, by the RBAC models.[697]
- Execution level: On this level, elements of organization models are described in a machine-readable format to be interpreted at run time, for example, in XACML.

Different Role Understandings The table illustrates that roles represent the connection between organization models from enterprise modeling and IT security, where the understanding of the term role in both contexts differs slightly. Following SCHEER, in the context of *enterprise modeling*, a

[694] Compare SCHEER (1999).
[695] One exception is HOLMES ET AL. (2008), who propose a model-driven generation of organization structures targeted at BPEL4people (OASIS, 2008 B4P). However, as described above, the possibilities of BPEL4people to capture organization structures are limited.
[696] In the context of the Europol-Eurojust collaboration for example, BOUJRAF ET AL. (2007), p. 20, described a corresponding table showing different instantiations of the roles.
[697] An organigram on this level of granularity for Eurojust is described for example in BOUJRAF ET AL. (2007).

role describes a "certain type of employee with clearly defined qualifications and skills".[698] Similarly, BECKER ET AL. define a role as a set of skills and competences of an actor. While they understand skills as a set of qualifications and psychological and physical skills, they define a competence as the right of an actor to execute certain functions.[699] In the context of *IT security*, a role is defined as "a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role"[700]. These definitions illustrate, that, depending on the relation between actor and the function set allocated to him, different understandings of roles can be distinguished:

- Skill: A role can express the *skill* of an actor to execute certain functions. For example, the role "Chauffeur" could be associated with the skill "is able to drive a car".
- Permission: A role can also express the *permission*, or the right of a person to execute a function. Obviously, a permission to execute a function should imply the skill to execute a function. However, skills and permission can be disjoint: A person may have the skills to drive a car but no permission; but on the other hand, a driver's license could theoretically also be acquired without having the skills to drive a car.
- Responsibility: While the previous two understandings of a role are based on the *potential* of an actor, a role can also express the *duty* or the *responsibility* of an actor to execute certain functions. For example, the role travel agent could be assigned to the function "reserve flight". This role understanding is usually exposed to collaboration partners: Though he presumes that the agent holds the necessary right to fulfill the advertised function, the customer of a travel agency is not interested in which permissions to access IT systems the travel agents holds but in the functions he can fulfill for him. Permissions and responsibilities can also be disjoint; for example, a principal can have the right to teach but not the obligation.

Scarce Support for Collaborative Business Processes Similar to process views, corresponding views on private, public and global organization models can be created. A comparable idea was proposed by CHIU ET AL. for "security flows".[701] Coming from the area of IT security, LEE & LUEDEMANN proposed an extension of XACML similar to the private, public, global concept.[702] Though the organization dimension of POP* aims explicitly at the support of interoperability and CBP, it does not distinguish between private, public and global views but pursues an approach comparable to ontologies, that allows mapping of heterogeneous enterprise models to shared concepts. Thus, an organization model in POP* could be interpreted as a global, organization model; however, the mapping of a private (or public) organization model to a global model is not described. SARSHAR & THELING focused on globally visible organization units, which for private use can be refined into the constituting internal organization units. In summary, it can be said that existing ap-

---

698 SCHEER (2000), p. 58.

699 Compare BECKER ET AL. (2002), pp. 82. ROSEMANN & ZUR MÜHLEN (1998) have a closely related understanding of this concept and define roles as the qualifications and competences of an actor.

700 ANSI (2004), p. 4. Another example for the understanding of roles as permissions can be found in FREUDENTHAL ET AL. (2002), p. 413, where "roles represent classes of permissions".

701 Compare CHIU ET AL. (2005).

702 LEE & LUEDEMANN (2007).

proaches for collaborative views on organization models indicate the feasibility of such views, but are described in such low detail that they can only be seen as a starting point for further work.

4.3.3 Metamodel for the Organization Dimension

4.3.3.1 Relations Between Collaborative Views

Similar to concepts for describing private, public, and global process, concepts have to be created to communicate organizational roles to collaboration partners. This description should tell partners (e.g. organization B and C), which rights certain roles in organization A have, enabling them to judge, for example, if they can send classified documents to this role. It can be expected that such descriptions are foremost useful in collaborations, in which sensitive objects are exchanged that justify elaborate mechanisms to protect their security, and corresponding mechanisms are already in place.[703] The need for such collaborative views was also confirmed by the modeling methods applied by practice partners in the R4eGov project.[704] The modelers intuitively used two different types of organization models to prepare the collaboration: internal models, describing organization elements for internal stakeholders, and externally visible models, where collaboration partners describe for each other which organization elements are involved in the collaboration. In the latter model types, instead of all organization units being available in one organization, only those units directly relevant for the other collaboration partner are exposed. Thus, they represent global models: the partners create one shared, agreed upon model that contains all elements relevant for both collaboration partners.

In the case of only two collaborating organizations, these two model types can suffice. However, in case one organization collaborates with various, changing partners, it would be more efficient to have a description of the externally relevant organization units that could be offered to the different partners in order to form corresponding global processes. Therefore, a public view for organization elements from each organization should be supported explicitly, not only implicitly in form of a global model. Thus, in the following, a metamodel is described that offers private, public and global views on organization models. Before the details of the organization models are defined, first a generic metamodel for the three collaborative views is developed (illustrated in Figure 64). The relationships between the elements of the generic metamodel can be defined as follows:

Relations Between Organization Model and Model Elements An organization model consists of 1 to many elements where one organization element can be comprised in 0 to many organization models (e.g. the role “secretary” can be used in various organization models).[705] Various relationships can exist among the elements themselves, in the case of roles for example the “is subordinate”, or in the case of organization units the “is part of” relationship. These relationships apply equally to private, public and global organization models.

---

[703] Such mechanisms could include for example the classification of documents according to their secrecy level, and a role concept specifying what kind of actor can access which document type.

[704] More specifically, the stakeholders from two large European Public Administrations, who in the R4eGov project described the planned interaction between them; the corresponding scenario is described in Chapter 5.

[705] It is assumed, that more than one organization model can exist inside on organization. First, due to vertical decomposition, which allows to display sub-models (e.g. hierarchies); second, due to horizontal decomposition.

Relations Between Private and Public Organization Models One private organization model (OM) can be represented by 0 to many public OM; for example, by one public OM describing roles relevant for partner A, and a separate public OM that comprises the roles relevant for partner B. On the other hand, one public OM could represent 0 to many private OM, e.g. the organization elements "secretary", "deals with correspondence of", "professor" could occur in various departments of a university.[706]

Relations Between Elements of Private and Public Organization Models Following the assumption that a public OM abstracts information from a private OM, one element of a private process can represent 0 to 1 elements of a public processes. Since it is a public counterpart of a private element, a public element represents at least 1 element of a private OM, but it can also subsume many elements of a private OM.

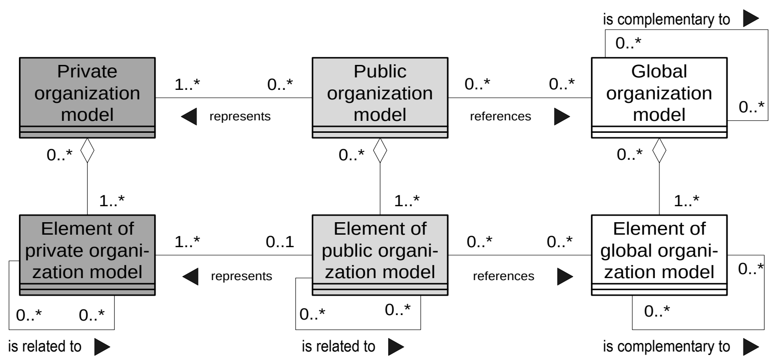


Figure 64: Generic metamodel for organization elements in collaborations

Relations Between Public and Global Organization Models Also in the case of organization structures, global models can either represent complementary public models or individual reference models. In cases when they represent a composition of complementary public models, the functionalities of the described roles would be complementary. For example, the roles "Buyer" and "Seller" could be defined with complementary functions; thus, each organization participating in the corresponding collaboration could offer these public roles and relate them to private roles. However, in the following, a global OM is seen as an individual reference OM, which collaboration parties can refer to ensure a common understanding of organization elements.[707] A public OM can reference to 0 or many globally known reference models and a global OM can be referenced by 0 or many public OM.

Relations Between Elements of Public and Global Organization Models If a public OM references to a global OM, an element comprised in the public model can reference to 0 or many global elements. In cases with 0 global counterparts, the public element is not part of the reference; in cases with many global counterparts, the public element represents an aggregation of global ele-

[706] Note though that – following the principle determined in the process dimension – a public view represents an interface of exactly one private service; correspondingly, at run time, one public OM should be correlated to exactly one private OM.

[707] For example, the role "judge" could be described in a global model, specifying his rights and responsibilities to execute certain functions as well as roles that are related to him, e.g. "secretary of judge".

ments.[708] One element of a global OM that is referenced by a public OM on the other hand can be referenced by 0 or many elements of the public OM.

4.3.3.2 Connecting Conceptual and Execution Level

The metamodel displayed in Figure 64 describes the relationships of private, public and global organization models. Based on these relationships, now the metamodel of the AIOS can be defined, which should also represent a connection between business and execution level. In consequence, the functions to be supported by the organization models comprise: Business-level representation of organization units and positions involved in the collaboration, support of workflow execution by correlating functions, roles and actors, and support of access control mechanisms by correlating actors, roles, functions, and objects. Figure 65 displays the corresponding metamodel for the organization dimension of the AIOS.

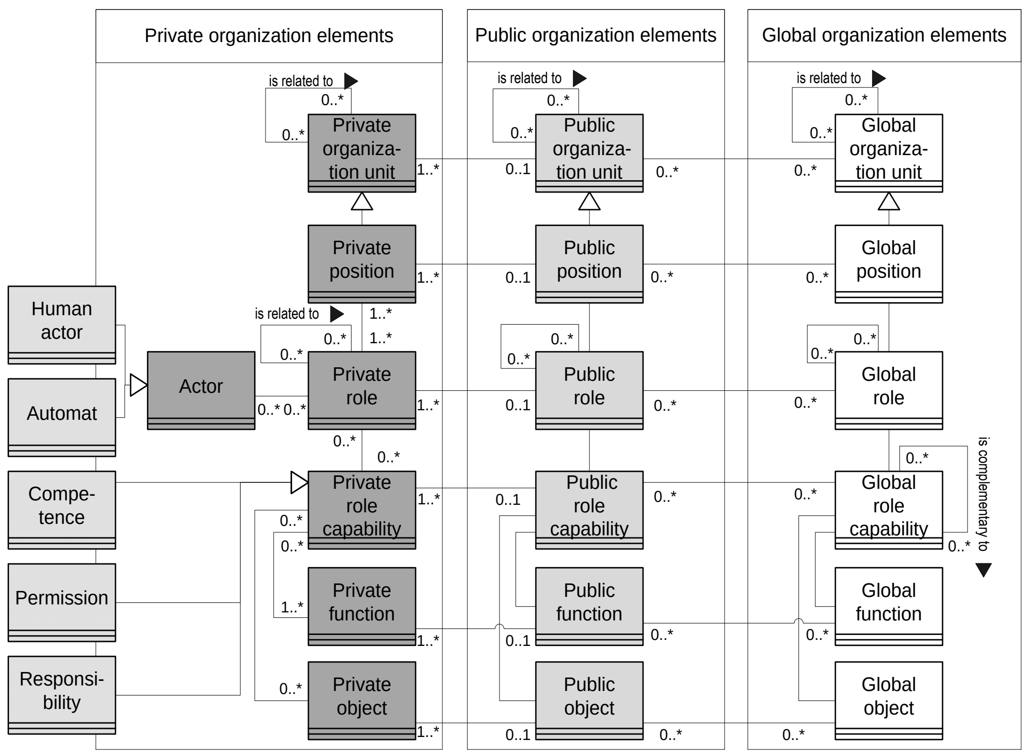


Figure 65: Metamodel for the organization dimension of the AIOS

Ensuring a connection to business-level models, the main organization elements are aligned with the metamodel from SCHEER, e.g. organization units, positions and roles correspond to the ARIS elements. In accordance with SCHEER as well as VAN DER AALST ET AL.[709], actors are divided into *Human Actors* and *Automats*. In difference to the organization models described above, the AIOS metamodel distinguishes three ways to correlate roles with functions and objects: *Competences*,

708 Thus, the public role “technical Support” could refer to the global roles “network administrator” and “database administrator”.

709 VAN DER AALST, KUMAR & VERBEEK (2003).

describing the skills of an actor, *Permissions*, describing the access rights of an actor and *Responsibilities*, describing the tasks an actor has to execute. All three types represent specializations of the generic class *Role Capability*. This distinction supports the three application areas of organizational models described above (business-level modeling, workflow enactment and IT security):

Competences and Responsibilities are mainly used in business-level organigrams, though responsibilities can also be used in the execution of workflows to allocate resources to functions. For the allocation of tasks to actors, VAN DER AALST ET AL. also included in their metamodel its availability,[710] while SCHEER included the location of actors. Though such time and local restrictions of actors are not illustrated in the AIOS metamodel, they can be described as attributes of the class Role capability. VAN DER AALST ET AL. also modeled delegation relationships, enabling the delegation of tasks between users and between roles. Such relationships are be captured by the recursive "is related to" association attached to the *Role* class; this generic association can also capture other relations, for example organizational relationships like "is supervisor of". To support access control mechanism needed in the context of IT security and a mapping to XACML, the core elements of RBAC are represented in the metamodel: Permissions that correlate actors, roles, functions and objects.

The relationships between the private, public and global views correspond to those defined in the generic metamodel (compare Figure 64). The recursive relationship on the class *Global Role Capability* can be used to indicate whether global organization units are complementary to each other, e.g. if the roles in adjacent global organization models have matching rights and responsibilities. Note, that for illustrative reasons, the cardinalities of the vertical relationships were only depicted in the private view; the same cardinalities apply to their counterparts in the public and global views. Similarly, the specializations of the private role capabilities also apply to public and global role capabilities. Concrete actors on the other hand are usually displayed only in the private view, while in the public and global view only the role of an actor is displayed.

4.4 Data Dimension

In this section, the metamodel for the AIOS data dimension is specified, that enables private, public and global views on data models. Since collaborating organizations interact mainly through the exchange of messages, the data dimension is essential in the development of collaborative business processes; sometimes, a business relationship is even defined as a *sequence of document exchanges*.[711] The importance of the data dimension in collaborative business is also indicated by the fact that traditional eBusiness protocols like UN/EDIFACT focus on document specifications, while the automation of business processes as workflows does not receive a similar attention.[712] In the context of business processes, data can be divided into items that constitute the control flow and items that are consumed or produced by the functions of the process. While control flow-related data items are described in the process dimension, this section focuses on data that is produced or

[710] VAN DER AALST, KUMAR & VERBEEK (2003).
[711] Compare GLUSHKO & MCGRATH (2002), p. 42.
[712] Compare for example BERNAUER, KAPPEL & KRAMLER (2003).

consumed by functions of a (collaborative) business process, and thus implicitly also tackles the AIOS output dimension.[713]

4.4.1 Preliminaries

Refining the distinction of data that is part of the control flow and other data, SCHEER distinguishes between the four types of information objects illustrated in Figure 66: *Events* and *Messages* belong to the control flow directing the process. The third type, *Environment Status*, represents the status of the process environment. The data type *Information Service* comprises documents, which serve as an input or an output of information services.[714] Resorting to this distinction, the AIOS data dimension focuses on the latter type, i.e. on documents that are produced and exchanged between collaborating organizations.

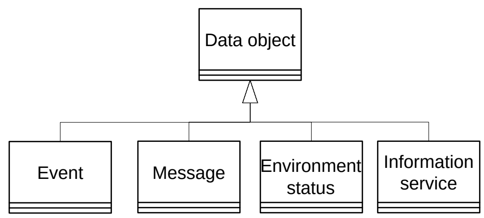


Figure 66: Different roles of data objects[715]

Macro vs. Micro Data Further, two granularities of data objects can be distinguished: Macro data objects represent coarse-grained entities like document types, voice recordings or files. Micro data objects represent fine-grained entities comprised in the macro data objects, describing detailed data elements and their relationships using notations like the Entity Relationship Diagram (ERM).[716] GLUSHKO & MCGRATH draw a comparable distinction between "document-centric analysis" and "data-centric analysis",[717] where the first corresponds to the macro data view and the latter to the micro data view. The document-centric approach originally focused on information exchange between human beings. Thus, in addition to the data, its formatting was also described. The data-centric approach on the other hand has its roots in information systems design and focuses on computer data files and databases.[718]

Model-Driven Development of Documents The distinction between micro and macro objects translates into a separation of documents and their elements, which does not necessarily imply the usage of different technologies to represent both types, but rather the existence of different levels of abstraction on the same technological level. However, also in the data dimension various complementary standards on different levels of technical granularity exist. Similar to the web service stack,

---

[713] The relationship between the data and the output dimension is further described in the next section, pp. 175.
[714] Compare SCHEER (2000), pp. 67.
[715] Adapted from SCHEER (2000), p. 68.
[716] Compare SCHEER (2000), pp. 67.
[717] Compare GLUSHKO & MCGRATH (2002).
[718] Compare GLUSHKO & MCGRATH (2002), p. 43.

standards to describe documents can be classified in various layers; thus, in Figure 67, a document stack is illustrated that comprises five layers.[719]

On the *bottom layer*, basic data types are described. On the *second* layer, a syntax is provided to structure basic information types into document schemas, for example by XML. On the *data descripton layer*, data types are described. While XML represents a means of describing the syntax of a document, standards like DFDL[720] and UN/CEFACT's Core Components (Data Types) describe the semantics of data types that can be used to assemble business documents. On the *fourth layer*, these data types are aggregated into core components, which represent semantic building blocks being used as a basis to construct electronic business messages.[721] Thus, they can be combined in different ways to create libraries of interoperable business documents.[722] Currently, Core Components represent the dominant approach for defining business semantics and underlie the design of most open document interchange standards (*fifth layer*), like for example UBL[723] or OAGIS[724]. RosettaNet and UN/EDIFACT are examples of standards that cover different layers of the stack (but different from ebXML, RosettaNet does not employ a component-based approach).[725]

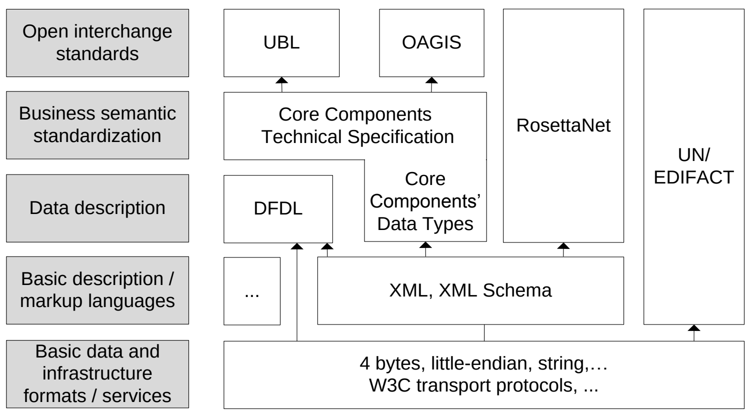


Figure 67: A document standards stack[726]

Two Level Approaches As the existence of the document standards stack suggests, also in the data dimension approaches for model-driven development exist. Motivating a model-driven development of documents in the context of SOA, LIEGL states that most eBusiness standards describe documents only on the implementation level and points out that the lack of a more abstract, conceptual description of business documents makes it difficult to communicate and modify document standards.[727] Thus, HUEMER & LIEGL describe an approach based on two vertical levels, where a UML profile for UN/CEFACT's core component standard is used to generate XML schemas for

---

719 Compare JANIESCH & THOMAS (2006), pp. 61.
720 Data Format Description Language, compare DFDL (2008).
721 UN/CEFACT (2003), p. 11.
722 Compare HUEMER & LIEGL (2007), p. 1.
723 Universal Business Language, OASIS (2004).
724 Open Applications Group Integration Specification, OAGIS (2007).
725 Compare JANIESCH & THOMAS (2006), p. 62.
726 Adapted from JANIESCH & THOMAS (2006), p. 62.
727 Compare LIEGL (2008).

business documents.[728] Another example for XSD messages that are derived from a specific UML profile is provided by KRAMLER ET AL.[729]

Three Level Approaches GLUSHKO & MCGRATH also propose a model-driven development of eBusiness documents. In difference to the previously described approaches, they propose the use of three vertical levels, comprising logical models (conceptual view), physical models (technology view) and completely specified documents (implementation level). JANIESCH ET AL. propose a comparable approach that describes processes and documents on the business, technical and execution level. On the business level, the document is specified from a business perspective, forming the general structure of the business document. On the technical level, data types are assigned to the contents and further attributes, e.g. relations to reference ontologies, are added. On the execution level, the document is described "in a certain schema in a certain syntax wrapped in business messages"[730].

Interestingly, in order to stay independent of the interchange standards of the highest layer, all approaches described above only derive documents contents up to the fourth layer of the document stack ("Business Semantic Standardization", compare Figure 67). The approaches based on two vertical levels disregard the business level and transfer technical models to the execution level. Since the AIOS should also support the business level, a three level approach is followed, where the first level represents a business view of the documents. More specifically, the three vertical levels of the AIOS in the data dimension are defined as follows:

- Business level: On this level, the type of document (e.g. "Request for Quote") and its elements are defined on a coarse-grained, business-oriented level. Referring to the distinction of micro and macro data objects,[731] this level focuses on macro data objects.
- Technical level: Here the micro data objects are specified in detail, including their data types and possible references to ontologies. While the previous level concentrated on the types of documents and major elements, here the structure of the overall document is defined, including the format of the document elements.
- Execution level: On this level, a computer interpretable format of the message is described that can be used to store and exchange documents. While the previous two levels are used at design time, on this level documents are specified for run time purposes.

Thus, on the execution level, documents could be specified in document interchange format like UBL or OAGIS; however, they could also be specified in proprietary formats defined in XSD. In this work no specific interchange format are proposed, only generic document specifications based on XSD. This ensures that the AIOS stays domain independent, since the use of XSD is not restricted to eBusiness but can also be used for interactions among public administrations. Further, the usage of XSD provides flexibility and the possibility to create individual document types going beyond existing document standards.

---

728 Compare HUEMER & LIEGL (2007).
729 Compare KRAMLER ET AL. (2006).
730 JANIESCH ET AL. (2006), p. 518.
731 Compare SCHEER (2001), pp. 67.

4.4.2 Approaches for Collaborative Views on Data

The public/private/global concept originates from the data dimension, since the concept of workflow views was inspired by views used in databases.[732] In database theory, a view is a query designated for a specific user group and their demands or rights. For example, a SQL-view[733] that grants collaboration partners the right to query public documents would be implemented as follows:

```
create view publicDocuments as select * from internalDocuments
where visibility = "external Partners";
grant select on publicDocuments to collaborationPartners;
```

Three levels of a database are distinguished: The *internal* scheme describes how data is physically stored in a database. The *conceptual* scheme abstracts from technical specifics and describes the database on a logical level. The *external* scheme provides user specific views on the internal scheme.[734] Thus, different from ordinary database tables, a view is not directly related to physical schema but represents a virtual table assembled dynamically from a database.[735] The following objectives of view creation can be distinguished:

- Information abstraction: The view contains only those elements of a database that a user group is interested in; other elements are omitted.
- Information hiding: Internal knowledge is protected against the interest of external users by exposing only unclassified information in the view.
- Information aggregation: Information relevant for a user can be extracted from different tables to be aggregated into one view. Therefore numeric operation might be applied, e.g. to summarize database entries.

Additionally, views can be used to implement generalizations or specializations among data elements, where the sub-element automatically belongs to the super-element and inherits its attributes. Therefore, either the sub-element or the super-element can be defined as a view.[736] Note, that the different objectives of database view creation are not competing but rather complementary. Thus in the example view listed above, a supplier company can provide database access to collaborating car manufacturers while both protecting internal knowledge and presenting the data in an aggregated form suitable for the car manufacturer.

4.4.2.1 Literature Review

The concept of publishing selected data to specific user groups stems from the database area (e.g. micro data) and surely is more common there, than it is in the document (e.g. macro data) area. However, with the transition from paper-based to digital forms, the complexity of data stored in documents is approaching the complexity found in databases. Thus, HUEMER & LIEGL criticize the fact that XML-based document standards try to "include every possible element that may be re-

---

[732] Compare for example LIU & SHEN (2003), p. 506 or PANKRATIUS & STUCKY (2005), p. 84.
[733] SQL stands for Structured Query Language; compare for example MOLINARO (2006).
[734] Compare PERNUL & UNLAND (2003), pp. 181.
[735] Compare for example KEMPER & EICKLER (2004) or PERNUL & UNLAND (2003).
[736] Compare KEMPER & EICKLER (2004), pp. 130.

quired in any partnership, even knowing that it is not used in most of the partnerships". As result, only 3% of the document elements specified in the standards are used in a specific collaboration. The consequence is that in order to reduce the standard to their actual needs, partners have to agree on adapted document structures for each collaboration.[737] Accordingly, also in the area of document modeling, mechanisms are applied that – like database views – allow abstracting from irrelevant document elements.

Information Abstraction KRAMLER ET AL. for example, described a framework for modeling interactions between web services, which consists of various layers, including layers for information items, documents, interactions and services. Both the document and the information layer serve to define the data structures of documents being exchanged between collaboration partners. For this purpose, a "Message Content Model" is used which is annotated in a specific UML profile; this profile serves as a basis to generate XML messages. Further, two types of message models are supported: An *abstract* message model specifies the message element minimally needed in a collaboration, while the complete message model contains additional information needed in specific collaboration instances.[738]

Information Hiding While the approach described above focuses on omitting unnecessary information (e.g. information abstraction), collaborative views as known from the process dimension aim foremost on privacy maintenance (e.g. information hiding). In the context of business process automation, concepts to control data visibility focus on individual, intra-organizational processes. Thus – comparable to programming languages, which distinguish between local and global variables – in workflows, *task, block and workflow data* is distinguished: Task data is only visible for one specific activity, block data is visible in a certain sub-process and workflow data is visible in the complete workflow.[739] However, going beyond intra-organizational processes, a couple of approaches exist that describe views on documents in the context of collaborative business processes:

BUGAJSKI, GROSSMAN & VEJCIK distinguished between global data, to be used in CBP, and data used only internally, referred to as local data.[740] According to them, data and services generally are local, and only when they have to be used in CBP are they required to be global. Global data should be designed in a way that "different applications storing, accessing, querying or updating global data using global services can easily interoperate"[741]. To support this requirement, they described various properties that data must fulfill to be global; for example, that data elements are defined via XML schemata and accompanied by XML-based metadata. Moreover, this metadata should be accessible through a distinct repository and should describe how the data can be accessed. However, they did not describe in detail the relationship between local and global data; neither did they distinguish between global data belonging to one party and centrally global data being owned by all parties.

---

737 Compare HUEMER & LIEGL (2007).
738 Compare KRAMLER ET AL. (2006).
739 Compare WESKE (2007), p. 101. A similar mechanism is implemented in BPEL, where variables can be declared to be visible for a complete process or for individual process parts (scopes), compare OASIS (2007).
740 Compare BUGAJSKI, GROSSMAN & VEJCIK (2006), p. 593.
741 BUGAJSKI, GROSSMAN & VEJCIK (2006), p. 593.

GOU proposed a "Collaborative Document Exchange" framework to support different forms of cross-organizational processes, including peer-to-peer-based processes. In this framework, two concepts resembling private and global data structures are provided: The "business document structure" describes a business document from the viewpoint of one organization, while the "collaboration document structure" serves as a mediator between different, possibly heterogeneous "business document structures".[742]

DAMIANI ET AL. described a fine-grained access control system for XML documents. More specifically, they developed a role-based access mechanism that allows for specifying which users can execute which actions on selected parts of a document described in a Document Type Definition (DTD). Since these actions are not restricted to *reading* but also comprise *writing* activities, their approach goes beyond the development of views and allows partners to alter internal documents. Similar to database views, their approach does not distinguish between public and global views.[743]

CHIU, SHAN & HUNG suggested applying the public/private principle to other workflow dimensions, including the data dimension. In consequence, they proposed a distinction between "data flow" and "data flow views".[744] However, they did not detail their proposal, thus the nature of their data flow views remains unclear.

LIU & SHEN likewise described data in the context of workflow views, distinguishing between data relevant for private processes and data relevant for public processes. According to them, public data is produced or consumed by the (public) functions of a public processes. Since they understand public functions as abstractions of private functions, public data sets represent abstractions of the data sets which are consumed or produced by private functions.

CHEBBI, DUSTDAR & TATA similarly described workflow views and tackled the data dimension using so-called dataflow contracts. A dataflow contract specifies all messages that are needed in one global process, e.g. all messages that each organization participating in the collaboration sends.[745] Like CHIU, SHAN & HUNG, they did not describe explicitly the relation of this global data to public or private data elements. Nevertheless, their description implies that the global messages represent the sum of the public messages (e.g. messages associated to individual organizations) used in a global processes.

JANIESCH ET AL. criticized that the interdependencies in developing collaborative business processes and the documents needed in these processes is not tackled by research. To close this gap, they proposed an integrated approach, describing the joint development of CBP and the related documents. For the development of collaborative business processes, they also used private, public and global views on processes. Similar to the AIOS, they transfer this concept to documents by categorizing them into private, public and global documents.[746] However, their work focuses on the configuration of the individual information elements not on the collaborative views. Thus, the collaborative views on documents are only sketched out, indicating a focus on global document specifica-

742 Compare GUO (2006).
743 Compare DAMIANI ET AL. (2002).
744 CHIU ET AL. (2005), p. 108.
745 Compare CHEBBI, DUSTAR & TATA (2006), pp. 209.
746 Compare JANIESCH ET AL. (2006).

tions on both the business and the technical level, whereas on the execution level, public document specifications are used. The relation between public and private documents is not described in detail.

BPEL The data flow among and between BPEL processes is based on XML data types and more specifically, WSDL message types. To maintain the state within a business process, these message types can be accessed via variables. Interestingly, the BPEL specification names the different treatment of data as a major difference between private and public process implementations: “The rich data manipulation that occurs in executable processes need not be described in public process contracts”[747]. However, no distinct concept for private and public data elements exists in BPEL, only public and private processes (e.g. BPEL Abstract and Executable Processes) are distinguished explicitly. Nevertheless, since abstract processes can omit all private elements of the executable processes, all private documents simply can be omitted in the abstract process. In consequence, in the abstract process only public documents are visible, while in the private process complementary private documents can be contained. In addition to the simple omission of private documents, the possibility exists to explicitly declare attributes of public documents (e.g. those documents comprised in an abstract process) as private.[748] Note that the attributes of a BPEL variable can refer only to the specification of the overall variable and not to single parts of it. Thus, it is not possible to declare individual parts of the document specification as private; only the overall specification can be annotated as being “opaque”.[749]

	Bugajski et al.	Guo	Chiu et al.	Liu & Shen	Damiani et al.	Chebbi et al.	Janiesch et al.	BPEL
Private documents	●	●	●	●	●		●	●
Public documents			●	●	●	○	●	●
Global documents	●	●				●	●	

● / ○ / Blank: Views are completely / partly / not supported

Table 10: Different approaches for collaborative views on documents

4.4.2.2 Conclusions

The literature review showed, that in the area of document modeling two approaches for the creation of views exist: In the first approach an abstracted view of a document is generated in order to focus on the document elements relevant in a specific collaboration; however, both the abstract and the complete document specification would be available to all collaboration partners. The second approach resembles collaborative views as known from the process dimension and has a stronger focus on information hiding.

[747] OASIS (2007), p. 155.

[748] More specifically, referring to abstract processes the BPEL specification states, that “all WS-BPEL attributes are allowed to be opaque in the common base“ (OASIS, 2007, p. 148).

[749] The BPEL specification provides the example of “<variable name="commonRequestVar" element="##opaque" />“ (OASIS, 2007, p. 161), where the content of the attribute “element” refers to the XML definition of the variable.

Table 10 provides an overview of the approaches where collaborative views on documents are proposed. Of the six approaches, only the one from JANIESCH ET AL. incorporates all three (private, public and global) collaborative views. The other approaches only support two views: an internal and an external view on documents. The relationships between the views in most cases are described only vaguely; or, in the case of BPEL, only implicitly. Aligned to the collaborative views from the process dimension and the document views existing in literature, private, public and global document views can be defined as follows:[750]

Private and Public Documents Private documents in the context of collaboration are designed for usage inside of an organization but comprise elements relevant for collaboration partners. Public documents are specified for partner organizations to describe which input and output documents an organization can process. Note, that most private process involved in a collaboration implicitly comprise private and public documents. However, the explicit distinction between private and public documents helps to avoid leakage of private information, since only public documents are allowed to be transmitted over organizational boundaries. Thus, the clear definition of the relation between private and public document elements simplifies the transition of information across organizational boundaries, since it is already specified which elements can or cannot be copied for external usage. In addition to *data protection*, the explicit modeling of collaborative document views has the advantage of increasing *reusability*: A predefined mapping of internal and external documents can be used in different processes, superseding the need for investigating the relationship of sensitive private data and public documents each time such a transfer occurs.

Global Documents While a public document only represents the viewpoint of one organization, a global document specification represents a reference model known and agreed upon among the collaboration partners. For this aim, standard messages from well-known eBusiness protocols can be used, like for example the specification of an invoice message defined by UN/EDIFACT. However, a global document does not have to be specified by a standardization body, it is sufficient if the partners of a specific collaboration agree on the document specification. A global document specification used in a smaller scope is for example the European Arrest Warrant, which is used among European agencies.[751] As discussed above,[752] global models can have either the form of complementary public models or the form of individual reference models. In the case of documents, both approaches are feasible,[753] though the usage of global models as non-complementary reference models seems to be more common. Following this assumption, the only difference between a public and global document is its global acceptance among the collaboration partners. Nevertheless, even in this case, the distinction between public and global models is useful: If an organization engages in different collaborations, it will be beneficial for the organization to define the boundaries of its data dimension in the form of public documents. Then, the organization does not have to define the relationship between internal and external data for each collaboration, but can resort to the previously defined relationship between private and public documents. Thus, the public documents represent a starting basis for the negotiation of the global document; afterwards, the global docu-

[750] In the following, the terms private/public/global *view* on a document are used synonymously with the terms private/public/global document.
[751] Compare EAW (2009).
[752] See pp. 111.
[753] An example of complementary documents could be a pair of documents like “RfQ” and “quote”.

ment elements can be mapped to the public document elements and, in a second step, enable a correlation with private document elements.

On the Support of Cross-Organizational Database Access As described above, a document and a database-centric approach for information exchange can be distinguished. The AIOS pursues the first approach, and direct access to databases of collaboration partners is granted only in the case of the BII-repository, which focuses on design time functionalities.[754] Thus, following the paradigms of Service-orientation (not data-orientation) and loose coupling, during the execution of business processes, data exchanges between organizations are executed only via services with clearly defined input and output documents, i.e. here no direct access to the databases of partner organizations is granted.

4.4.3 Metamodel for the Data Dimension

In this section, the relationships between private, public and global elements in the data dimension are specified. To leave room for different document interchange languages as well as for individual document definitions, and in order to support a mapping both to the business and the execution level, the metamodel stays on a generic level; it distinguishes only between documents, their elements and the three collaborative views (compare Figure 68).

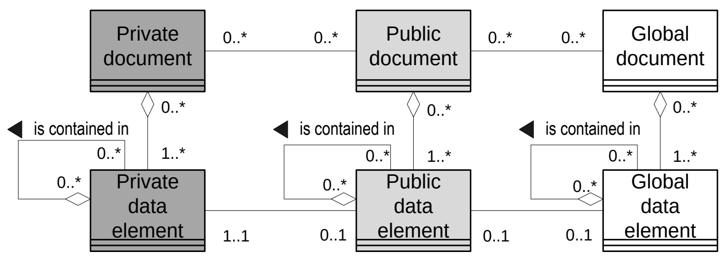


Figure 68: Metamodel for the data dimension of the AIOS

Documents and Their Elements Three different types of document contents can be distinguished: First, the content components, representing the individual pieces of information contained in a document; these are also referred to as document *elements.* Second, the structure of a document, which describes the arrangement of the elements. And third, presentation components, which describe the formatting or rendering of both the structure and content components.[755] In the following, we will abstract from the presentation and structuring components and refer only to documents and their elements (e.g. the content components). Since the latter might have a hierarchical structure, also sub-elements are taken into account. Thus, a document consists of 1 to many data elements, and one data element can be contained in 0 to many documents (e.g. the data set “Address” can be used in different document types). Further, an element can have 0 to many sub-elements (e.g.

[754] The BII-repository is described in Chapter 5, pp. 195. This repository comprises the BII of one organization and resembles a database accessible to collaboration partners. However, this database focuses on functionalities needed during the design and implementation of the collaboration; it does not comprise run time information, e.g. instances of processes or documents.

[755] Compare GLUSHKO & MCGRATH (2002), p. 44.

the element “Application” may comprise various subsections like “Curriculum Vitae” and “Credentials”); on the other hand, a sub-element can be contained in 0 to many upper-level elements.

Figure 69 exemplifies the relationships between private, public and global document specifications: On the left-hand side, organization A uses the document A1 for internal purposes. For collaborative purposes, only the elements A1.1 and A1.2 are relevant and public to partner organizations. To ensure that no private elements are published the documents A1.1’ and B1.2’ are created as public counterparts to A1.1 and A1.2. Since A1.1’ is sent to organization B and B1.2’ is received from organization B, the specifications of the public documents in both organizations must be consistent with each other. Thus, they also play the role of global documents. To simplify the process of agreeing on common document formats, the organizations can resort to globally accepted document standards, e.g. to the documents 1.1 and 1.2.

Relations Between Private and Public Documents One public document can represent 0 to many private documents, and one private document can have 0 to many public counterparts. In order to allow for copy-operations, document elements that are mapped to each other should have the same format and have a 1:1 relationship. However, in case that a private document has no public counterpart, the elements of this private document would have no public counterpart, either. On the other hand, each element of a public document should be mapped to one element of a private document, since otherwise the public element would not be used internally.

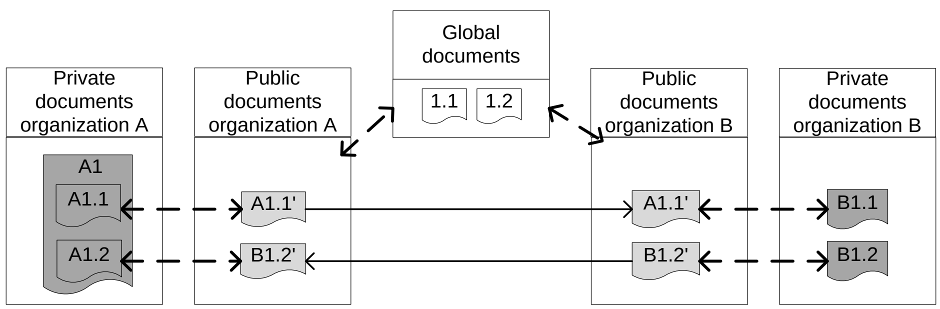


Figure 69: Relations between private, public and global document specifications exemplified

Relations Between Public and Global Documents In cases where public documents reference global documents, normally one public document would be related to one global document. However, it is also possible to relate the contents of one public document to many global documents (e.g. specifying that half of the public document consists of the global document “Curriculum Vitae”, while the other half consists of the global document “Certificates”). On the other hand, it is equally possible that one global document references many public documents. Obviously, both public and global document specifications can also exist independently from each other. Accordingly, one element of a public document can be related to 0 or 1 element of a global document, while one element of a global document can be related to 0 or 1 element of a public document.

4.5 Output Dimension

As mentioned in Chapter 3, in the output dimension all physical and non-physical input and output of functions are described.[756] Obviously, the specification of such output elements is important in the development of collaborations,[757] since knowledge about the expected results of the overall collaboration as well as the output expected from individual collaboration partners is a prerequisite for any joint work. The output dimension is also useful in the context of SOA, where a genuine description of the service output (and input) supports the dynamic discovery of services. However, the position of the output dimension in the collaboration lifecycle differs from the positions of the other dimensions: it focuses on the requirements level,[758] and thus is used mainly in the initiation of the collaboration that precedes the design phase. In this vein, the output dimension was used in the ArKoS project to specify the products that should be the outcome of a collaboration.[759] Nevertheless, on the operational level, the output dimension is usually displayed only implicitly, e.g. in the data dimension. Accordingly, Table 3 (p. 100) illustrated that in the context of cross-organizational business process automation normally no explicit output dimension is used. However, as already described in Chapter 3, in order to stay consistent with ARIS and to provide a bridge to non-operational phases, the output dimension is seen as part of the AIOS, though in this work it will be tackled only implicitly, in the data dimension.[760]

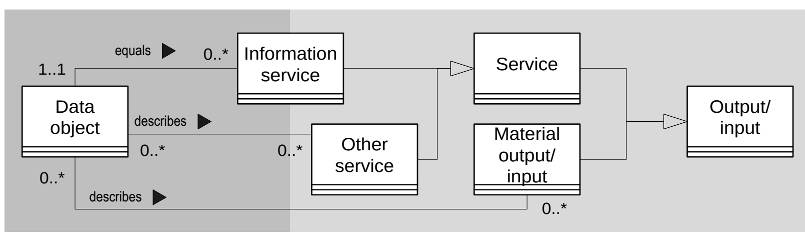


Figure 70: Relationship between elements of ARIS' data and output dimension[761]

Output Dimension vs. Data Dimension In ARIS' output dimension, the output produced by a function as well as the input flowing into a function is described.[762] As illustrated in Figure 70, the items comprised in this dimension can be split into the following disjoint classes: *material input/output* and immaterial input/output in form of *services*, where services are split again into *information services* and *other services*.[763] Obviously, in information systems all these output/input types are represented as data objects, resulting in a strong overlap of the data and output dimension. As illustrated in Figure 70, in cases of so-called information services, the output of such a service

---

[756] Compare pp. 99.
[757] Compare also HOFER ET AL. (2005), p. 5.
[758] Compare SCHEER (2000), p. 93.
[759] Compare HOFER (2006), pp. 80.
[760] Compare pp. 99.
[761] Based on metamodels from SCHEER (2000), p. 97 and p. 168. *Data objects* belong to the data dimension, while the elements with the lighter background belong to the output dimension. *Information services* are allocated in both the data and the output dimension.
[762] Compare SCHEER (2000), pp. 93.
[763] The report about the creditworthiness of a certain person represents an example for an information service, while a haircut is an example for an "other service".

equals a data object; for example, the production of a digital document represents an information service and at the same time, a data object. Thus, the output type "information service" is part both of the output and the data dimension.[764] The types "other service" and "material output/input" on the other hand, do not represent data objects; however, in order to be representable in an information system, they are described in data objects.

Collaborative Views on Output Elements In the context of collaborative business processes, the ARIS output dimension was tackled in the ArKoS project.[765] Elements of the output view are often modeled in hierarchical structures, resulting in product trees that describe which sub-products are comprised in a product.[766] Thus, HOFER proposed the usage of global product trees, which represent a hierarchy of all products referenced in a collaboration.[767] This proposal already indicates the applicability of private, public and global views on the output dimension.

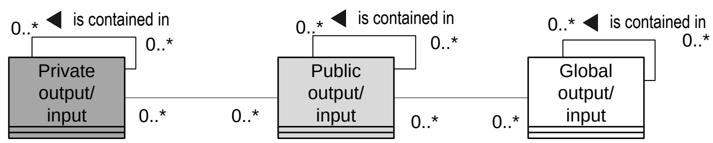


Figure 71: Metamodel for the output dimension of the AIOS

Obviously, the public view on output objects is necessary to describe to collaboration partners – or in the context of SOA, to service requestors – the output they can expect from a function offered by a specific organization. The mapping of the public output to a global output reference model (defined independently from specific organizations) is also useful, since it enables collaboration partners to better understand and judge the output produced by specific organizations.[768] The mapping of public to private output objects is useful as well, since the public description of an output might differ from the internal output description. For example, the same product could be offered under different names to different target groups. Another example is that the public product comprises different variations of internal products; thus, the public product "Limited library access" might comprise the private product "Access to library A", while the public product "Unlimited library access" could comprise the private products "Access to library A" as well as "Access to library B".

The corresponding relationships among private, public and global output elements are illustrated in Figure 71.[769] This generic metamodel covers all specializations of output/input elements illustrated in Figure 70, including "information services". Thus, the metamodel also can be seen as a generalization of the metamodel for private, public and global documents displayed in Figure 68. For the reasons mentioned above, in the following, the output dimension of the AIOS will be cov-

---

764 Compare SCHEER (2000), p. 68. This is also illustrated in Figure 66 (p. 165), where information services are depicted as part of the data dimension.
765 Compare HOFER (2006), pp. 80.
766 Compare SCHEER (2000), pp. 95.
767 Compare HOFER (2006), pp. 80.
768 For example, a producer of car parts could qualify a generic product by stating that it fulfils all requirements specified in the product definition of a globally known product catalogue.
769 The recursive "is contained in" relationship on each element indicates that also hierarchies, e.g. product trees, can be represented in the collaborative views.

ered only as part of the data dimension, e.g. by describing collaborative views on documents that are produced or consumed inside a collaborative business process.

4.6 Integration of Dimensions

At the beginning of this chapter it was stated, that the different dimensions of the AIOS should not only be internally consistent, but also be complementary to each other and offer corresponding connection points. This requirement applies for all three AIOS axes, i.e. the dimensions of the enterprise model axis but also to the dimensions of the collaborative view axis and the dimension on the MDD axis. The integration along the three AIOS axis can be described as follows:

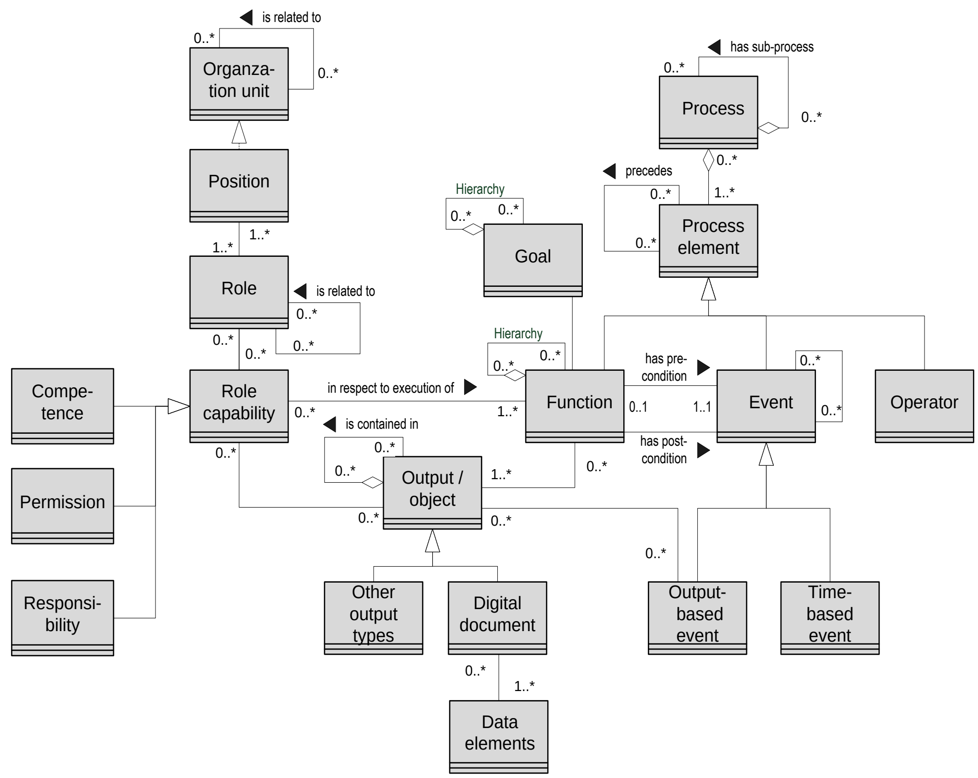


Figure 72: Relations among the AIOS enterprise dimension elements in the public view

Integration Along the Collaborative View Axis The definition and integration of private, public and global views was the main objective of this chapter and the relationships between the three collaborative views in each enterprise dimension were specified in several metamodels (compare Figure 48, Figure 55, Figure 65, and Figure 68). Hence, the units on the collaborative view axis are considered as being integrated.

Integration Along the Enterprise Dimensions Axis In this chapter, the individual enterprise dimensions were described separately. However, Figure 72 illustrates that the elements of the AIOS enterprise dimensions can be integrated in one metamodel. On the left-hand side, the elements of the organization dimensions are described; on the right-hand side, the elements of the process dimension. As shown in the middle of the metamodel, these dimensions are connected by the elements of the function dimension. Below the function dimension, the elements of the output and the data dimension are depicted. Note, that the metamodel only comprises the elements of the public view, which – connecting private and global process elements – represents the central view of the AIOS. Nevertheless, since all public elements have a counterpart in the private as well as in the global view, the relationships among the enterprise dimensions in the private and the global view can be inferred from this metamodel.

Integration Along the MDD Axis In the previous sections of this chapter, in each enterprise dimension concepts for displaying collaborative views on the business, technical and execution level were reviewed. Based on this review, for each enterprise dimension a metamodel was defined, that integrates elements from both the business and the technical level, and can easily be mapped to the execution level. The specification of standards to be used on the different vertical levels is not in scope of this work; thus, a detailed mapping between the standards used on the different vertical levels is not provided, either. However, standards like EPC, BPEL or XACML were used as representations of well-known, established knowledge that influenced the shaping of the AIOS metamodels. In Chapter 5, these standards are used to demonstrate the applicability of the AIOS metamodels for supporting a model-driven development comprising the business, technical and execution level. Again,[770] these standards should not be seen as the only possible solution; they do not imply a determination of which standards must be used in the architecture. For example, here EPC and a technically enriched version of EPC are proposed for process modeling, but in cases where more collaboration partners use BPMN, this standard might be used alternatively.

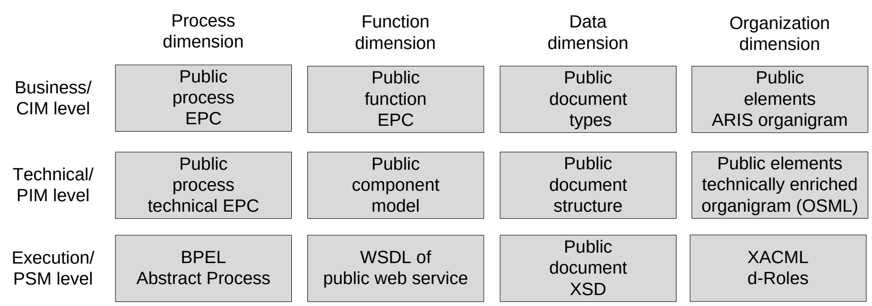


Figure 73: Example standards for vertical levels in the public view of the AIOS

The standards used in the procedure models described in Chapter 5 are illustrated in Figure 73: On the business level, ARIS-based notations are used. On the technical level, these are refined in order to represent a basis for the automated generation of execution level models. On the execution level, BPEL, WSDL, XSD and XACML are used.

770 Compare also Chapter 3, p. 75.

5 Procedure Model and Application

The goal of this chapter is twofold: First, to describe the dynamic aspects of the architecture in the form of a procedure model for the stepwise development of the model types comprised in the AIOS. Second, to provide a proof of concept for the AIOS, by applying it to a use case and by describing a toolset that is based on the metamodels described in Chapter 4.

In a first step, a generic procedure model for the AIOS is described (generic in the sense that it omits the AIOS enterprise dimensions). Further, the toolset is described, and – preparing the subsequent application of the AIOS – a use case from the R4eGov project is introduced. In a second step, the generic procedure model is refined and adapted to the specifics of each AIOS enterprise dimension. Thereby, it is also shown, how the collaboration in the use case can be developed with the procedure model and how the prototype can be used to support the steps comprised in the procedure model.

5.1 Generic Procedure Model

As described in Chapter 2, an architecture can comprise a *description view*, describing the various elements contained in architecture models and their relationships, and a *construction view*, representing a method for the systematic development of the architecture.[771] As described in Chapter 3, interoperability frameworks focus on the descriptive aspects of interoperability, for example by providing standards or technical recommendations; but they neglect the methodical aspects needed when developing collaborative business processes.[772] Thus, a general need to extend existing interoperability frameworks with methodical aspects can be observed.[773] Answering to this need, in the following, a procedure model is described that explains how the different model types of the AIOS can be systematically developed.

On the Supported Interoperability Types In Chapter 3, it was described that the AIOS supports the development of interoperable information systems in two cases: First, an organization prepares for a hypothetical collaboration and develops a BII based on its expectations but without existing collaboration partners; second, the BII development has to be synchronized with collaboration partners.[774] In the first case, the definition of the BII seems easier, since here no coordination with partner organizations is necessary and a potential collaboration partner would have to accept the conditions he finds in the BII. However, presuming organizations that elaborate collaborative business processes jointly, in the following a mutual adjustment of the BII among the collaboration partners is assumed.

[771] Compare pp. 31.
[772] Compare pp. 80.
[773] Compare also MATHEIS, ZIEMANN & LOOS (2006).
[774] Compare pp. 75.

5.1.1 Lifecycle Phases Covered by the AIOS

Methods to systematically implement IT systems are usually displayed by *lifecycle models*. Closely related to lifecycle models are *procedure models* and *phase models*; often, these three terms are used synonymously.[775] Depending on the granularity of the architecture elements, "big picture" procedure models and finer grained methods can be distinguished. Big picture methods represent coarse-grained procedure models, comprising for example a number of lifecycle phases to develop an architecture. Finer grained methods are then used inside these phases, to describe the development of individual architecture elements.

Most lifecycle models roughly follow a sequence of analysis of the as-is state, design of the to-be state, implementation and controlling. VAN DER AALST, TER HOFSTEDE & WESKE for example, described a lifecycle comprising the four phases *diagnosis, process design, system configuration* and *process enactment*.[776] JOST & SCHEER described a coarse-grained business process lifecycle with the phases *design, implementation and controlling*.[777] Focusing more strongly on the implementation of business processes, SMITH & FINGAR proposed a fine-grained lifecycle model, comprising eight different phases including *deployment, execution, monitoring, control* and *analysis*.[778] BREMER described and compared various procedure models for the development of information systems, including the Waterfall model, the U-model and the Spiral-model.[779] HOFER compared six different procedure models for the implementation of business processes, which follow comparable "generalized phases": *preparation and strategy, analysis of current state, development of the to-be concept, implementation* and *analysis/controlling*. She further compared six different procedure models in the context of collaborations and identified four generalized phases: Initiation and search for partners, specification, execution and termination.[780] Similar phases for collaboration lifecycles are described by WERTH and THELING.[781] In the R4eGov project, on an operational level, a so-called interoperability lifecycle was described, consisting of the seven phases: *requirements engineering, interaction design, model transformation, simulation/validation, deployment/execution, controlling/monitoring,* and *interaction governance*.[782]

Aligned with the above-mentioned procedure models, Figure 74 illustrates a lifecycle that focuses on the development of interoperable information systems. In a first step, *strategic* decisions are made to establish and automate collaborative business processes. Here, the collaboration partners are elected and strategic goals of the collaboration are defined. Moreover, in this phase the partners in the collaboration sphere agree on the major parameters of the collaboration; for example, to offer Business Interoperability Interfaces to each other, to use certain standards (e.g. EPC or BPEL), and to agree to certain legal constraints of the collaboration. A pre-condition for further development steps is, that the collaborating organizations are aware of the current state of their systems; to ensure this, in this phase an explicit *as-is analysis* should also be executed.

---

[775] Compare HOFER (2007), pp. 110.
[776] Compare VAN DER AALST, TER HOFSTEDE & WESKE (2003), p. 5.
[777] Compare JOST & SCHEER (2002), p. 44.
[778] Compare SMITH & FINGAR (2002), p. 90.
[779] Compare BREMER (1998).
[780] Compare HOFER (2007), pp. 119.
[781] Compare WERTH (2006), p. 61 and THELING (2008), p. 47.
[782] Compare MATHEIS ET AL. (2008).

After the context parameters are clarified, the design of the interoperable information systems begins, divided into the phases of *business level design* and *technical design*, where the creation of executable models is subsumed under the latter phase. The AIOS focuses on these design phases. Here, possibly in various iterations, private, public and global processes are described and refined, covering the different enterprise dimensions as well as the three MDD-levels. As result, the internal and external views on the information system of each organization are described on the business, technical and execution level; in other words: the model types comprised in the AIOS cube (illustrated in Figure 36, p. 115) are described for each collaboration partner.

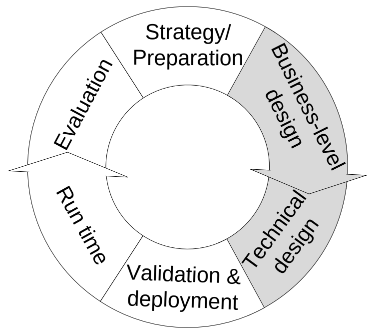


Figure 74: Development phases of interoperable information systems covered by the AIOS

The models developed in the previous step are then *validated* and *deployed* to the process engines and the related technical infrastructure of each partner.[783] In the *run time* phase, the deployed services and processes are executed. Other than functions related to the process execution (e.g. functions for a secure message exchange), here mechanisms for monitoring and controlling collaborative business processes can be used also.[784] After the execution of the CBP, the overall interoperability solution should be *evaluated* regarding the fulfillment of strategic goals. As illustrated in Figure 74, the evaluation can lead to major readjustments on the strategic level. However, if only minor modifications are necessary, in the next iteration of the lifecycle, the strategy phase can be omitted.

5.1.2 Procedure Models Supporting Private, Public and Global Views

As mentioned above, two generic approaches for developing CBP models based on collaborative views can be followed: an outside-in or an inside-out procedure.[785] The *inside-out* procedure starts from existing private process, derives from them corresponding public processes and composes these into global processes. The *outside-in* approach starts from a given global model, derives from

---

[783] In the validation of collaborative business processes it is of special interest, whether the public processes defined by each party are compatible; compare for example WESKE (2007), pp. 238. On the execution level, the compatibility of interacting BPEL processes is discussed by various authors, for example by FISTEUS, FERNÁNDEZ & KLOOS (2005) or FREIHEIT & MOHNDORF (2007).

[784] A concept for the controlling of collaborative business processes related to the AIOS is described by MATHEIS & LOOS (2008).

[785] Compare for example GREINER ET AL. (2006) or BECKER, JANIESCH & PÖPPELBUß (2008). Other authors – e.g. WERTH, WALTHER & LOOS (2007) and FU, BULTAN & JIANWEN (2003) – use the terms bottom-up and top-down. Nevertheless, in the following the afore mentioned terms will be used, since the transformations between different privacy levels (private, public and global) represent *horizontal* transformations, while the terms bottom-up/top-down imply a *vertical* transformation.

it the public models and forces the collaboration parties to adapt their internal processes to the given global model.[786] In the model-driven development of CBP, usually a *mixture* of both approaches can be observed, which involves various loops of adaptations between global, public and private processes. For example, a coarse-grained global process could be used as a starting point (outside-in approach). Coming from there, the involved partners create a detailed process model of the private processes that are comprised in the global process. In these private processes, they identify and detail those elements that should be visible to the partner, i.e. the public processes. In case a private process cannot be adapted the requirements of a global process, the global process must be adapted to the public processes (inside-out approach).

Outside-In vs. Inside-Out Approach An advantage of the outside-in approach is that the effort to coordinate the involved parties is lower: It is possible to use a given, possibly validated reference model of a global process. Each party then implements its part and the collaboration can be executed. In the same vein, FU, BULTAN & JIANWEN argue that the existence of one integrated global process model supports model-checking techniques, while a composition of public process models would be harder to validate.[787] WEBER, HALLER & MÜLLER developed choreographies in the context of virtual organizations and argue as well in favor of an outside-in approach, stating that in contrast to the inside-out approach, the outside-in approach would fulfill the "intrinsic necessities" of virtual organizations.[788] WERTH, WALTHER & LOOS on the other hand support a decentral process development and correspondingly advocate the inside-out approach, stating that the outside-in approach would force organizations to constrain their autonomy in an unacceptable manner.[789]

The different opinions indicate that the choice for one of the approaches is context dependent: If collaboration partners require a global process that is tailored to their individual private processes and they can bear the potentially high costs of developing and validating an individual global process, they will opt for an inside-out approach. If they prefer to engage in a standardized collaboration situation, which is described via a validated reference model, or if they are willing to adapt their internal processes to the demands of a global process owner, they will opt for an outside-in approach.

In the context of collaborative business process development, the term *harmonization* – indicating that the collaborating parties have to agree on shared standards – is sometimes used with a negative connotation.[790] However, if parties want to collaborate, a certain degree of harmonization is necessary, since they always have to agree on a common global process. But following the inside-out approach they agree on it cooperatively, with the aim to preserve their internal processes and regulations as they are, keeping adaptations as limited as possible. The outside-in approach on the

786 Such an outside-in approach was for example described by VAN DER AALST & WESKE (2001). In their proposal, the collaboration partners first agree on a global workflow, afterwards each partner implements that part of the global workflow, which was assigned to him. More technical outside-in CBP development procedures were described for example by MENDLING & HAFNER (2005) or KIM & HUEMER (2004).

787 Compare FU, BULTAN & JIANWEN (2003).

788 Compare WEBER, HALLER & MÜLLER (2006), p. 325.

789 Compare WERTH, WALTHER & LOOS (2007).

790 For example, in the R4eGov project partners were skeptical about an interoperability approach that would require the "harmonization" of collaborating organizations, which is explained by the traditional independence of large public administrations and their reluctance to succumb to changes imposed on them from the outside.

other hand bears the risk of ignoring the requirements of individual partners. Thus, addressing autonomous, independent collaboration partners, the inside-out approach seems more suitable for collaborative business process development.

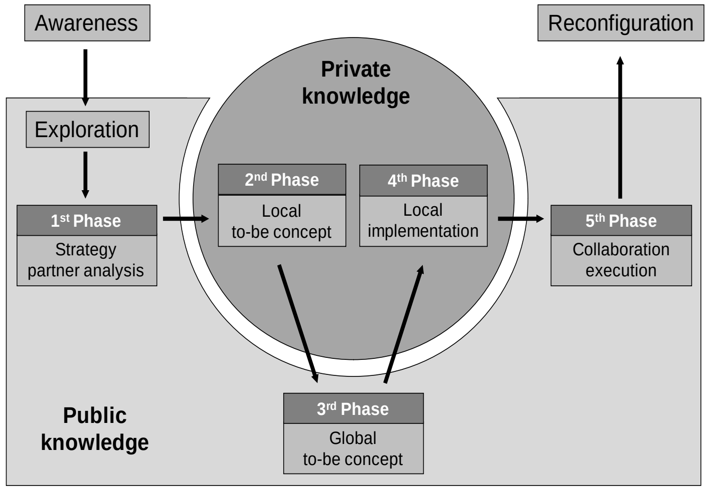


Figure 75: ArKoS lifecycle for collaborative business process management[791]

Since the view concept for business processes is relatively new, few detailed procedure models exist for developing private, public and global processes – and these procedure models focus on the process dimension, neglecting the organization, function, data and output dimension.[792] Most procedure models represent simple, coarse-grained sequences, for example by proposing to derive public processes from a global process and afterwards derive the corresponding private processes;[793] only rarely are alternatives to the proposed procedures discussed. WERTH, however, explicitly described a procedure model for an outside-in development of collaborative business processes, which comprises phases to create private processes, derive public processes, compose these into a global process, and to validate the global process.[794] On a more technical level, WESKE described an outside-in procedure model, where, after the development of a choreography (e.g. a global process model), choreography interfaces and later orchestrations are created.[795] In the ArKoS project, the lifecycle model depicted in Figure 75 was created, which contains elements of an outside-in approach though the focus lies on the inside-out approach: In the first phase, a global agreement on joint goals is developed. In the second phase, private processes are created that comply with the

791 Adapted from ADAM ET AL. (2005), p. 86.

792 One exception is the procedure model based on the ArKoS lifecycle (compare Figure 75), which also distinguishes between private and public elements of the output dimension, compare for example HOFER (2007). Other enterprise dimensions in which the public/private/global concept was applied comprise the data and the organization view (compare Chapter 4). However, these approaches were only shortly sketched out without an explicit procedure model.

793 Such coarse-grained procedure models can be found for example in WERTH, WALTHER & LOOS (2007), ZIEMANN, KAHL & MATHEIS (2007) or MENDLING & HAFNER (2005).

794 Compare WERTH (2006), pp. 195.

795 Compare WESKE (2007), pp. 231.

goals of the previous phase. In the third phase, a global process is composed based on the private processes. In the fourth and fifth phases, the concepts are implemented and executed.[796]

While the procedure models described above focus on the public/private aspect of collaborative business process models (e.g. on *horizontal* model transformation), a comprehensive procedure model for CBP development should also support *vertical* model transformations, to enable process automation and interoperability on different levels of technical granularity. In summary, it can be stated that existing procedure models for CBP development show gaps in all three axes of the AIOS: Collaborative views are supported only scarcely (e.g. the ArKoS procedure model neglects the public view), the enterprise axis is mostly reduced to the process dimension, and the MDD axis is disregarded in most approaches for developing collaborative views on business processes.

5.1.3 Procedure Model for the AIOS

5.1.3.1 Dimensions and Transformation Types

The development of the AIOS has to take into account the three axes introduced in Chapter 3: enterprise dimensions, MDD-levels and collaborative views. Thus, the procedure model should describe the development paths between the different model types comprised in the AIOS cube.

On the Sequence of Enterprise Dimension Development Though SCHEER describes a coarse-grained procedure model in which enterprise models are technically refined,[797] he does not describe explicitly the sequence in which the different enterprise dimensions have to be developed. However, the task of the control view in ARIS is “to reconnect the views (function, organization, data, and output view, respectively)”[798]. This could be seen as an indicator for the fact that the static dimensions are to be developed before the process dimension. A corresponding procedure model would resemble a *bottom-up* process, where first the elements appearing in the process are defined, and the connecting structure is established afterwards. For example, the document types needed in the collaboration could be specified first, then the functions in which documents are processed, afterwards the roles (responsible for certain functions) and at last, the process structure that integrates the different elements. In this vein, it could be argued that first the functions in a collaboration should be defined, and afterwards the processes that connect them. Accordingly, CHEBBI, DUSTDAR & TATA propose that – in order to define “the visibility levels of its workflow” – each organization participating in a collaboration first specifies its global functions, and only afterwards the public process that describes the sequence in which these functions are invoked.[799] However, the development of CBP can also start with the process dimension. In this case, the related static dimensions can be inferred from the process model, like for example, function trees implied by the process. This would represent a *top-down* procedure, where the process as the integrative part is developed first, and elements related to the process are developed in detail later on.

---

[796] Compare ADAM ET AL. (2005).
[797] Compare SCHEER (1999), p. 7. The ARIS procedure model includes the phases requirements definition, design specification and implementation description.
[798] SCHEER (2000), p. 102.
[799] Compare CHEBBI, DUSTDAR & TATA (2006), pp. 19.

In summary, it can be said that – different from the MDD-levels, which correspond to concept and implementation phases – the nature of the individual enterprise dimensions does not imply a sequence in which they should be developed. Instead, the development sequence depends on the collaboration context. For example, in an environment that traditionally uses specific document forms, the data dimension is modeled probably first and only afterwards the processes that describe in which sequence the documents are exchanged. On the other hand, in an environment where responsibilities and rights are strongly regulated and certain functions are implied by roles, the CBP development would rather start with modeling the organizational roles involved in the collaboration.[800] Therefore, in the following no generic sequence for the development of enterprise dimension is prescribed.

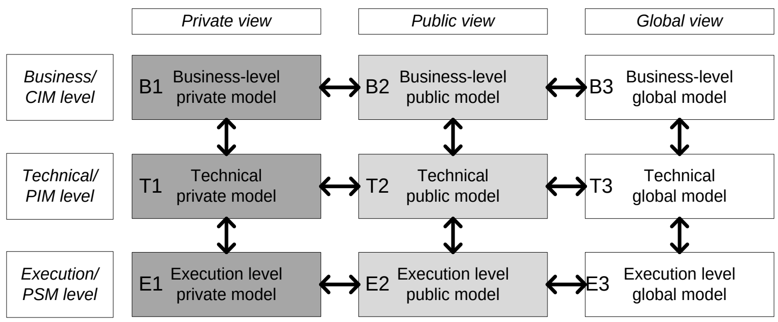


Figure 76: Model transformations involved in the model-driven development of CBP

Sequences for Developing Collaborative Views on Different Vertical Levels Accordingly, this section tackles only the sequence in which the collaborative views and the MDD-levels are developed. Figure 76 displays the corresponding AIOS elements, only covering the collaborative views and the MDD-levels. In order to implement a CBP, at least the three model types on the execution level of the matrix illustrated in Figure 76 should be specified. Based on the nine elements of the matrix, six horizontal and six vertical operations can be part of the CBP development:[801]

- B1←→B2: Deriving a conceptual public model from a conceptual private model or creating a private model fitting to the public one. The same operations apply on the technical and the execution level, e.g. T1←→T2 and E1←→E2.
- B2←→B3: Compose public models to a global model or decompose a global model into public models. The same operations apply on the technical and the execution level, e.g. T2←→T3 and E2←→E3.
- B1←→T1: Transforming a conceptual private model to a technical private model or transforming a technical private model to a conceptual private model. These relationships also apply for the corresponding public and global models, e.g. B2←→T2 and B3←→T3.

[800] This was the case, for example in the EP-EJ use case described in R4eGov, where the first model displayed the actors and organization units involved in the collaboration between EP and EJ.

[801] As illustrated in Figure 76, B1, B2, B3, T1, T2, T3, E1, E2, E3 stand for private, public and global views on business, technical and execution level models.

- T1←→E1: Transforming a technical private model to a private model on the execution level or transforming an execution-level private model to a technical private model. These relationships also apply for the public and global models, e.g. T2←→E2 and T3←→E3.

Thus, 24 different transformations could be part of the software-development process.[802] Since this work is not aiming at the re-engineering or monitoring of CBP but at the development of interoperable software systems, the vertical direction of the transformation generally is *top-down*: Collaborations are modeled on the design level first and afterwards are refined and implemented. What is less clear is the sequence of the horizontal transformations.

5.1.3.2 Different Development Paths

Illustrating the different sequences that are eligible in the model-based development of collaborative business processes, in the following, five different development paths are discussed:

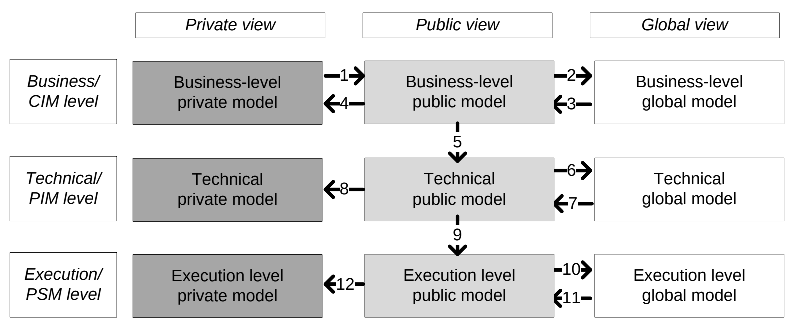


Figure 77: Inside-out approach with vertical transformation via public models

Figure 77 illustrates the transformation steps in an inside-out approach that starts with the business-level private process of each party taking part in the collaboration.[803] In Step 1, public models are derived from business-level private models. In Step 2, the public models are composed into a global model. The consolidated global model is then decomposed again into the comprised public models, which are returned to the owning organizations. If the consolidated public models deviate from those created in Step 1, the private model now must be adapted to the modified public model (Step 4). In Step 5, the conceptual public model is technically enriched. Based on the resulting technical public model, a technical global model (Step 6) is created by the collaboration partners. The consolidated technical global model is then returned to the organization (Step 7) and a corresponding technical private model is created (Step 8). After the technical details have been specified, now the resulting public model can be transformed into executable code (Step 9). Then, the execution level global model can be composed from the public models (Step 10). If this composition re-

[802] Note that on a generic level these transformations were described in Chapters 2 and 3; more specifically: The horizontal transformations (between private, public and global models) were described on pp. 44 and vertical transformations were described on pp. 55 and pp. 108.

[803] As described above, we omit here the preparation phase preceding the model development, like for example, the agreement on collaboration goals.

quires changes to the involved public models, these are returned to the owning organization in Step 11 and 12.

Thus, starting from the business-level private process, global processes were implemented and adjusted to the executable private processes. Note that this procedure model does not illustrate the synchronization of the private and global execution level models with their counterparts (private and global models) on the business level. However, since all collaborative views (e.g. global model and public model, and thus implicitly also the private model) on the higher levels were synchronized among the partners before the vertical transformations, it can be expected that no major differences between business-level private models and execution level private models exist.

Technically, the transformation between the business and technical level is similar to the transformation between the technical and execution level, since both transformations equally represent primarily a technical refinement of the upper level.[804] Thus, assuming that the transformation sequence between the business and technical level corresponds to the sequence between the technical and execution level, in the following procedure models, the business level is omitted.

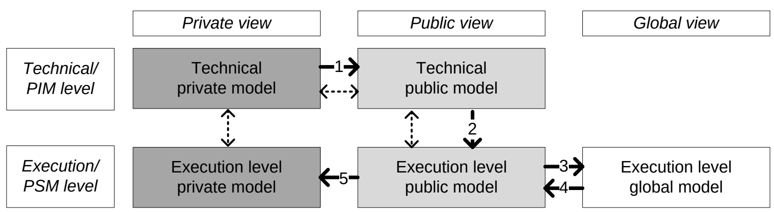


Figure 78: Inside-out approach with global harmonization on execution level

Figure 78 shows that an inside-out approach does not necessarily have to include all model types in the matrix. In this case, the public models are assembled into a global model only on the execution level, omitting the global process on higher levels. However, since no global model on the technical level is created, this procedure model bears the risk that during the composition of the execution-level global process (Step 5), substantial adaptations have to be made here, which might not be congruent with the upper level models. Thus, in this procedure model a synchronization loop (indicated by the dotted arrows) is required to ensure that the resulting execution level public model is synchronized with the technical public model. Therefore, the models of the public views have to be synchronized, and in case the upper level does not allow for the modification proposed on the technical level, a new iteration of the procedure model has to be initiated.

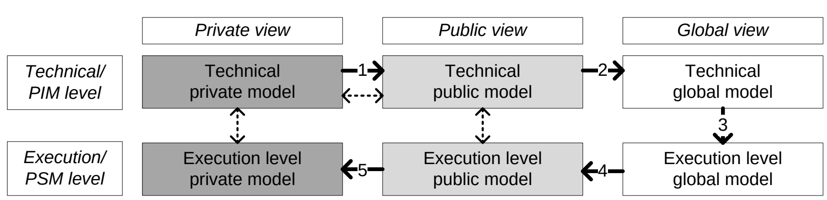


Figure 79: Inside-out approach with vertical transformation via global models

[804] Apart from the refinement, of course a syntactical transformation has to be applied, e.g. from EPC to BPEL syntax.

In the previous examples, the vertical transformation was based on the public view. This supports a decentral, interface-based approach since it implies that the public models of one company are synchronized with each other, e.g. public models on the conceptual level correspond to those on the execution level. However, it would also be possible to shift the vertical transformation to the global view. Figure 79 illustrates a corresponding procedure model, where the collaborating parties first agree on a technical global model (Steps 1 and 2) and, based on this, derive a global model on the execution level (Step 3). In Step 4, the global model on the execution level is decomposed into the comprised public models, which in turn represent the basis for the derivation of executable private processes (Step 5). As it was the case in the previous procedure models, here various iterations might also be needed to synchronize the upper level with the lower level models. For example, it could turn out, that a private model on the execution level deviates too strongly from the private model on the design level, which would initiate another synchronization iteration.

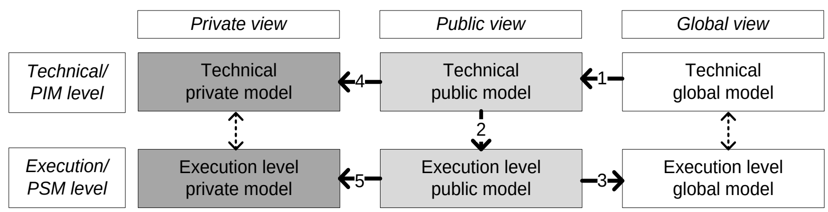


Figure 80 : Outside-in approach with vertical transformation via public models

Figure 80 illustrates an outside-in procedure model executed with a vertical transformation via the public view. In contrast to the inside-out approach, here the design starts with the global model. The focus of the outside-in approach shown in Figure 80 lies in the synchronization of the public models on different vertical levels, thus the risk of transformation errors, which might occur through imprecise transformations following the vertical transformation on the global view, is minimized. However, since the global model is more complex than the public model, it seems recommendable to start the synchronization chain in the global view and rather follow the procedure model displayed in Figure 81.

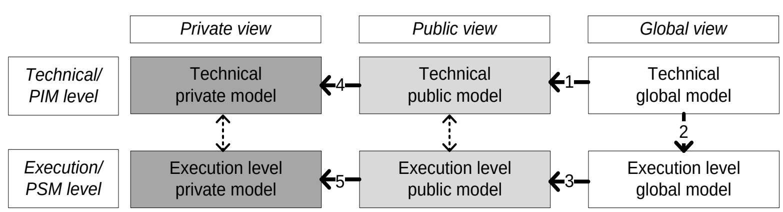


Figure 81: Outside-in approach with vertical transformation via global models

Figure 81 represents an outside-in approach based on a vertical transformation via the global view. The subsequent transformation to the execution level (Step 2) ensures that the design and implementation of the global model are synchronized, and validates the feasibility of the global model.[805] A general advantage of the outside-in approach is that the global model implies the public

[805] Naturally, this implies that the transformation from design to implementation leads to a correct, machine-interpretable model. If the implementation level turns out to be incorrect, either because it cannot be interpreted on execution level

models comprised in the collaboration. Thus, Steps 1 and 3 can usually be executed automatically,[806] without the need for further human coordination efforts (unlike the composition of public to global models, which usually require more effort). If these transformations are correct, the public models on the different vertical levels should be synchronized. Afterwards, the public models can be handed over to the collaboration partners responsible for implementing the corresponding private processes, which are derived in Steps 4 and 5.

Conclusions Five different ways for the development of collaborative business processes were described. Each of the transformation paths has advantages and disadvantages, and a decision for one of them depends on the context of the cross-organizational process. However, since it addresses the needs of autonomous collaboration partners, for the AIOS an inside-out approach is chosen. Thus, the transformation path displayed in Figure 77 will be used as a reference for the development of each AIOS enterprise dimension. Apart from realizing an inside-out procedure, this approach is further suitable for the AIOS since it supports the interface-orientation of the AIOS and ensures the consistency of the public views comprised in the Business Interoperability Interface of each collaboration partner. And, unlike for example the path displayed in Figure 78, it includes all nine model types in the transformation path and thus decreases the need for additional synchronization iterations.

5.2 Introduction to the EP-EJ Use Case

In the context of the R4eGov project, various scenarios of interactions between large European Public Administrations were described and evaluated to identify interoperability barriers and prepare corresponding solutions.[807] One of these scenarios is the Europol-Eurojust scenario, which will be used in this thesis to exemplify and validate the described concepts.

The Eurojust-Europol collaboration represents a European cross-border collaboration in the area of law enforcement. Europol is the European Police Office and Eurojust the European Judicial Co-operation Unit, enabling, for example, the collaboration of public prosecutors from the 27 member states of the EU. A closer collaboration between Europol and Eurojust is a political objective, as well as higher interaction efficiency through the replacement of manual with digital interactions. Since the main objective of the agencies consists in supporting cross-organizational processes between different European authorities, interoperability is an essential aspect of their overall strate-

or the interpretation does not comply with the specification, the source model (e.g. the technical global model) or the transformation rules have to be modified.

806 This is for example the case in interface-based global process models, which consist of several connected public process models. Thus, each party participating in the collaboration can simply be assigned to implement one public process.

807 Five scenarios were investigated in detail comprising 1. the German Supreme Court (Bundesgerichtshof – BGH) scenario, 2. a scenario related to the issuing of visas for international students coming to the EU, 3. the Austrian Federal Chancellery (Bundeskanzleramt – BKA) and exchange of documents with related administration, 4. an eProcurement scenario involving the Trade Register of the Paris Business Court (GTCP) and 5. the collaboration between Europol, Eurojust and related administrations. Compare BOUJRAF & NOBLE (2007), MATHEIS ET AL. (2007) and DIEDRICH ET AL. (2007).

gy.[808] Their collaboration is constrained and determined by a legal framework, which explicitly defines the boundaries of the interactions between those agencies. The collaborative scenario involves 27 actors from five countries and ten different document types being exchanged through seven different IT systems.[809]

Note that due to legal restrictions, some of the original terms in the process are modified, though the functionalities described in the following are congruent with the original scenario. *Indicating that the case described here is based on the Europol-Eurojust scenario but not identical with it, in the following these organizations will be referred to as EP and EJ.*

Figure 82 provides a high-level view of the elements involved in the use case. On the upper left-hand side, EP with comprised IT systems and related organizational roles (EP Liaison Officer, abbreviated as ELO) are depicted. Below, authorities related to EP but distributed over the various EU member states (MS) are illustrated: EP National Units and the Law Enforcement Authorities of the different member states.

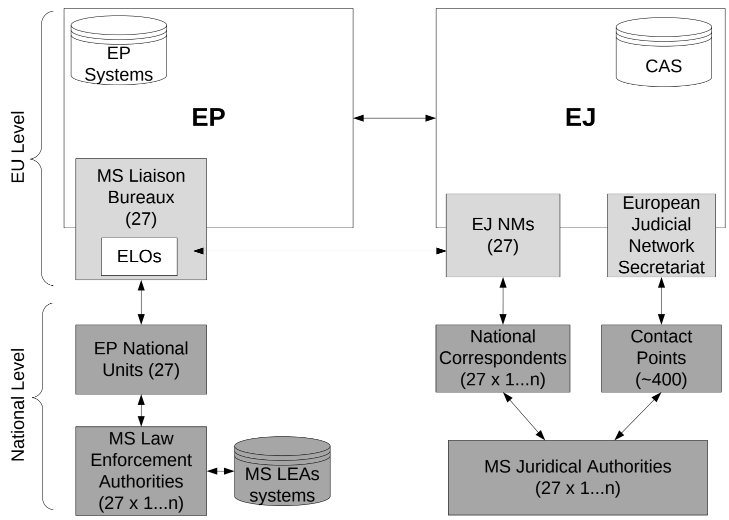


Figure 82: Overview of actors and IT systems involved in the EP-EJ collaboration[810]

On the upper right-hand side, EJ is depicted. It comprises the Case Analysis System (CAS), which stores the cases worked on in EJ, 27 EJ National Members (EJ NM) and the European Judicial Network Secretariat. As the graphic illustrates, these are related to several roles on the national level, e.g. 27 Correspondents, Contact Points and Member States Juridical Authorities. In this setting, different collaborative business processes can be instantiated. The following process will be used to illustrate the AIOS:

[808] This is also indicated by a recent report in which Eurojust described six strategic objectives for 2008 and 2009, where the first objective comprises the ability to process and manage terrorism information transmitted to Eurojust. The second objective also comprises two IT-related sub-goals: the creation of a "strong and secure ICT support environment for casework" and the creation of secure transmission links to the member states. Compare EUROJUST (2008), p. 63.

[809] Compare BOUJRAF & NOBLE (2007) and BOUJRAF (2007).

[810] Adapted from LEE & LUEDEMANN (2007), p. 84.

1. A member of a juridical authority (e.g. a French national prosecutor) discovers a case of arms smuggling and suspects that it involves organized crime from various EU countries. To gather more information about this from other countries, the French prosecutor writes a rogatory letter to his EJ National Member (EJ NM France). In reaction to this, the EJ NM creates a new work file in the CAS and asks his colleagues from EJ to have a meeting regarding the case.
2. In the meeting, the EJ NM from countries possibly related to the crime participate. It is decided, that the EJ NM from country A (EJ NMa) should contact EP to gather information about related incidents in his country.
3. To this aim, EJ NMa contacts the EP Liaison Officer of his country (ELOa) and sends him the case description.
4. Since the ELO cannot directly access all relevant systems, he asks the EP National Units of his country (ENUa) to retrieve information from the national law enforcement authorities.
5. ELOa also searches through the internal EP databases he can directly access.
6. After retrieving all requested information, ELOa forwards it to EJ.
7. EJ NMa uploads the information in the CAS and gives corresponding access rights to the EJ NM involved in the case, including the French EJ NM.
8. In a subsequent meeting, the involved EJ NM decide about activities that result from the information gathered.

In this process, currently different means of communication can be used, for example telephone, personal meetings, email and postal mail. To guarantee the privacy of the sensitive information, all communications are highly secured. Therefore, all information items transferred are restricted for use by certain roles. Though some interactions between organization units inside EP and EJ could also be tackled by an interoperability solution, in the following, the focus lies on the interaction between EP and EJ.

On the Representative Nature of the Use Case The objectives of using the EP-EJ scenario in this thesis are twofold: First, to illustrate individual concepts in a comprehensible way and thus to increase the traceability of the research process. Second, to illustrate that the deductively developed AIOS can be used in a real-life scenario. The characteristics and requirements of the EP-EJ use case can be summarized as follows:

- Joint objectives, complementary organizations: The organizations share a common objective they want to reach collaboratively. Although they posses heterogeneous internal structures, adjacent concepts are comparable and can be mapped to each other.
- Autonomy: Though they work together, the collaboration partners are autonomous; no partner is dominating another or could force a partner to modify his internal systems.
- Interoperability instead of integration: The interacting information systems should be only loosely coupled, only the interfaces of systems are connected, while internal structures remain independent from each other. The degree, to which internal knowledge is exposed, is precisely controlled.
- Elaborated software development: The parties represent large, complex organizations willing to invest in the modeling of business processes, organization elements and other elements necessary for process automation.

- Focus on information services: Instead of physical goods, the collaboration partners exchange information, usually in the form of documents.
- Need for process compliance: The shape of the processes is strongly determined by laws and regulations.
- Collaboration rectifies major investment: Inter-organizational processes are a core part of the business model of each party. Thus, they are willing to make a substantial investment in the automation of inter-organizational processes.
- Model reuse probable: The supported collaboration is not a one-time occurrence but, in a comparable form, will be executed over a long period. Thus, it can be expected that organizational roles, documents, processes and services appearing in a specific collaborative scenario can be reused in related scenarios.

Since these characteristics are compliant with the collaboration definition provided in Chapter 2, the EP-EJ use case is suitable for demonstrating the applicability of the AIOS. Obviously, different from eCommerce scenarios, all actors involved in the EP-EJ scenario are public administrations. However, requirements similar to those of the EP-EJ collaboration can also be found in large eCommerce scenarios.[811] Nevertheless, it could be argued that the security requirements in the EP-EJ are higher than in most eCommerce scenarios, since in the EP-EJ scenario data from citizens and criminal investigations is transmitted. Moreover, in contrast to enterprises, EP and EJ have monopoly-like positions inside the EU. Thus, it might be objected that they depend more strongly on each other (since no alternative organization can offer the complementary service needed) than collaborating enterprises usually do, and thus are more strongly inclined to invest in a sustainable collaboration infrastructure.

However, even if in eCommerce scenarios the dependence between two specific actors were lower, it can usually be expected that a large enterprise will stick to similar types of collaboration partners and thus could reuse the models defined in the AIOS.[812] Additionally, it should be noted that the focus of the EP-EJ scenario on information services concords with the focus of this thesis,[813] since only documents have to be modeled as input and output of services and processes, while the modeling of further output types (e.g. material products) is not necessary.

5.3 Tool Suite and Modeling Prototype

To illustrate them and to validate their feasibility, the concepts described above were implemented as prototypes. The prototype central for this thesis is the *VPD/GPD tool*,[814] which supports the modeling of different dimensions of the AIOS from private, public and global viewpoints. However,

---

811 For example, comparable conditions can be expected in a setting where a large supplier and a large producer of consumer goods exchange products in a large volume over a longer period of time.

812 For example, in case a producer of consumer goods replaces his old supplier with a new one, he can use his previously specified Business Interoperability Interface as a basis for the collaboration with the new supplier.

813 As described above, the output dimension is covered only to the extent in which it overlaps with the data dimension; compare also pp. 99 and pp. 175.

814 The acronym stands for “View Process Demonstrator/Global Process Demonstrator”, though the functionalities of the tool today go beyond the process dimension.

this tool is embedded in an *integrated tool suite* that provides a comprehensive coverage of the CBP development lifecycle, including tools for the design, verification, execution and analysis of CBP. These complementary tools were also described in the context of the R4eGov project.[815]

5.3.1 Overall Tool Suite

Figure 83 provides an overview of the tool suite, which represents a tool chain for the stepwise development and execution of the AIOS models. The tool chain starts with the creation of private business processes on the conceptual level in the form of EPC:

Modeling of Internal Processes with ARIS or Kindler Tools Various tools exist to create, validate and transform processes stored in the EPC Markup Language (EPML),[816] including an open source tool from CUNTZ & KINDLER,[817] which can model the private processes to be used inside the VPD tool. Another option is the ARIS toolset, a widespread industrial tool for enterprise modeling. The ARIS toolset exports EPC in the form of ARIS Markup Language (AML), which can automatically be transformed into EPML via an XSLT.[818]

Deriving and Connecting Collaborative Views with the VPD/GPD On the upper left, the AIOS modeling tools are shown that represent the core of the design solution: The VPD is a tool to model processes and organization elements, both from a private and a public perspective. Taking the public models from the VPD as input, the GPD can connect complementary public process models into a global process model. Both VPD and GPD use an extended version of EPML as input and output format,[819] and private EPML processes exported by the tools described above can be imported in the VPD.

Generation of BPEL To generate the BPEL files corresponding to the public and private processes from the VPD, various commercial BPEL designers are available. Such tools also allow the import of conceptual models, for example in BPMN, and generate BPEL processes from them.[820] Since in the VPD, both EPC and BPMN models can be created, the BPMN models developed in the VPD in a BPEL designer can be used as a basis to generate BPEL.[821]

Verification of CBP To ensure the complementarity of public processes that are supposed to form a global process, the abstract BPEL processes created in the BPEL designer can be validated

---

815 Compare ZIEMANN ET AL. (2008).

816 EPML was defined by MENDLING & NÜTTGENS (2005), and, besides the ARIS Markup Language, represents one of the two major formats for storing EPC. Due to its lightweight specification and its popularity in the scientific community, instead of the ARIS Markup Language, in the following EPML is used as a storage format for EPC.

817 Compare CUNTZ & KINDLER (2006).

818 Compare MENDLING & NÜTTGENS (2004). XSLT is a language for the transformation of XML documents; the acronym stands for "Extensible Stylesheet Language Transformation".

819 These extensions are described below, compare pp. 204.

820 In the R4eGov project, for example, the Intalio BPEL designer was chosen that fulfils these functionalities, including an BPMN-based BPEL generation. Compare also MATHEIS ET AL. (2008).

821 Since the scientific benefit of an automated BPMN (from the VPD) to BPMN (of the Intalio tool) transformation is questionable, we abstained from implementing an automated transformation from the VPD process export format to the Intalio import format. Therefore, the BPMN process from the VPD currently has to be re-modeled with the Intalio designer.

using a Petri Net-based verification mechanism.[822] Therefore, in a first step the BPEL processes are transformed into Open Workflow Nets (oWFN)[823] by using the BPEL2oWFN tool.[824] The resulting oWFN can then be evaluated in the in the Fiona tool[825]. As a result, the complementarity of the public processes is confirmed or declined.[826]

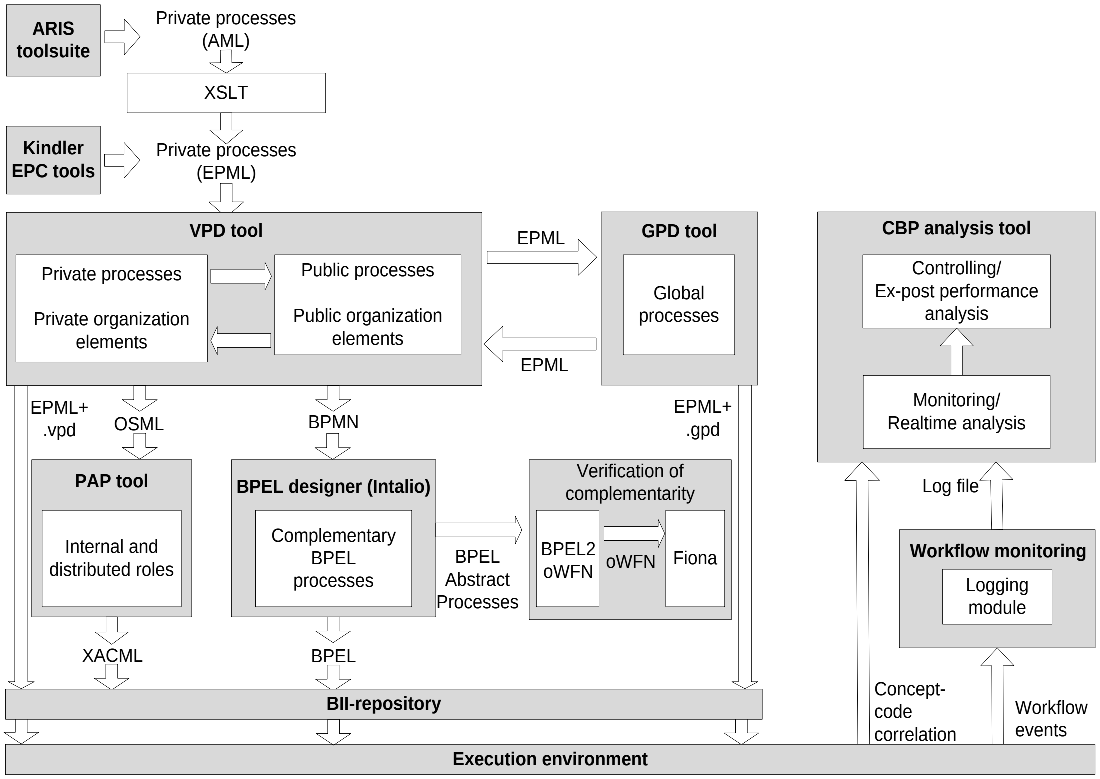


Figure 83: Tool chain supporting the modeling and enactment of AIOS elements

Describing Organizational Roles and Rights with VPD and PAP In the organization view of the VPD, organization structures are exported in form of the Organization Structure Markup Language (OSML), describing private, public and global views on organization elements as well as their correlation. These models constitute the basis for finer grained models, which describe the rights and roles inside collaborations, and as such can be used as input for the PAP tool. The PAP ("Policy Administration Point") prototype was developed to support the concept of d-Roles and distributed XACML. More specifically, it can be used to model XACML policy instances that link local roles to distributed role sets or to link distributed role sets to access control lists.[827] While the VPD tool tackles public, private and global roles on a

---

822 Compare FREIHEIT & MOHNDORF (2007).
823 oWFN are an extension of VAN DER AALST's workflow nets, compare MASSUTHE, REISIG & SCHMIDT (2005).
824 Compare LOHMANN, GIERDS & ZNAMIROWSKI (2007).
825 Compare MASSUTHE & WEINBERG (2007).
826 A detailed example for such a verification in the context of the Europol-Eurojust scenario is provided in FREIHEIT & MOHNDORF (2007), pp. 60.
827 Compare LEE & LUEDEMANN (2007).

conceptual level, the PAP tool refines private and global models to make them usable on the execution level.

Repository The BII of each organization is stored and published via the BII-repository, which can be accessed both at design time and at run time of collaborative business processes. A detailed description of the repository follows in the next section.

Execution Environment The execution environment complementary to the design time elements of the AIOS was already described in Chapter 3.[828] It is responsible for the execution of collaborative business processes, and comprises engines that execute internal processes as well as technical interoperability gateways that ensure the secure exchange of messages between the internal engines.

Monitoring and Controlling of CBP To enable run time monitoring of processes, a workflow-monitoring component receives events from the execution environment. For further analysis of collaborative business processes, the CBP analysis tool imports the log files generated by the monitoring component; it also accesses the BII-repositories of the collaborating organizations in order to correlate the technical events with conceptual business processes.[829]

5.3.2 BII-Repository

The BII-repository implements the Business Interoperability Interface, and thus has a central role in the AIOS implementation. The BII of an organization contains all models to be published to collaboration partners. To communicate the BII, each organization publishes its BII in a repository, which is accessible at least to its partners in the collaboration sphere.[830]

Content Figure 84 illustrates an example of BII-repositories from two collaborating organizations, where each organization offers public and global views of all four AIOS dimensions to its partner. Thus, public and global views of functions, documents, organization elements and processes are published. Though this in not illustrated in Figure 84, the different vertical levels of the AIOS are also represented in the repository; thus, besides the business-level models, also their counterparts on the technical and execution level are stored in the repository. To connect the different model types, the repository contains correlation files; these describe the correlation between public and global models (*horizontal correlation*) as well as the *vertical correlation*, where the relationship between conceptual models and their technical counterparts (e.g. between EPC and BPEL processes) is specified. The horizontal correlation is illustrated in the form of vertical black arrows in Figure 84. Note, that since the correlation between private and public processes should not be exposed to collaboration partners, this correlation information and the private models themselves are stored outside the repository. Additionally, the vertical arrows in the graphic illustrate the connection of the different enterprise dimensions with each other. Here, no separate correlation files are used, since the correlation information is comprised in the individual models. For example, a

[828] Compare pp. 117.

[829] These tools are currently being developed in the R4eGov project; a more detailed description of them can be found in MATHEIS & LOOS (2008).

[830] Compare also Chapter 3, pp. 114.

process model can comprise references to elements of an organization model, or an organization model can comprise references to a function model.

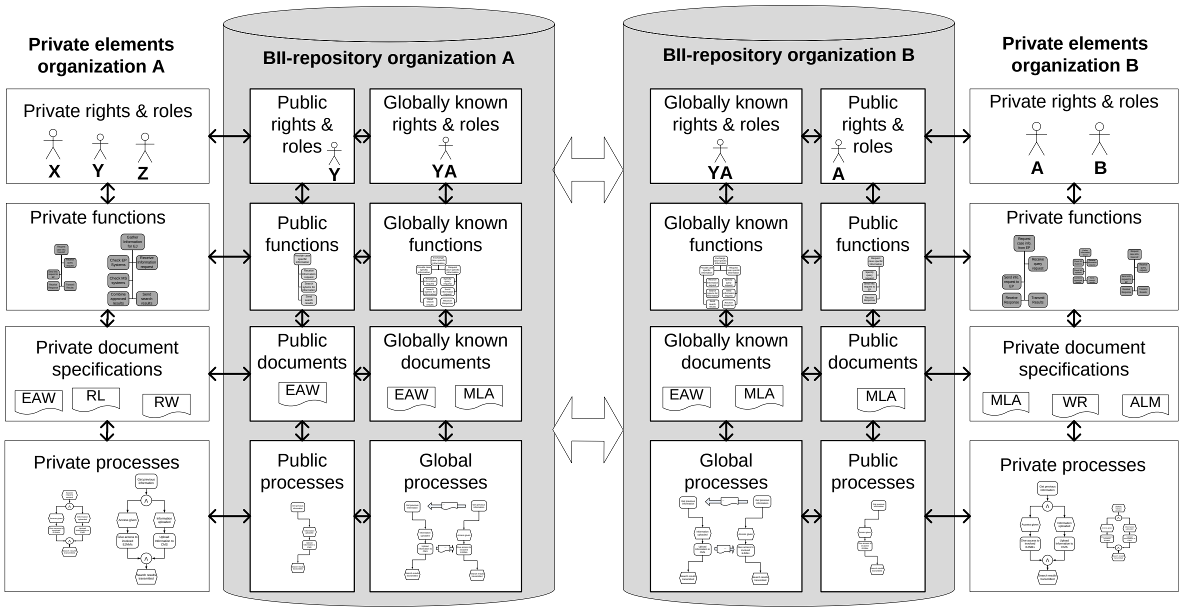


Figure 84: Each organization offers public and global models in its own BII-repository

Functionalities Thus, the repository is a major communication medium for collaborating organizations. During design time, parties can mutually access their models to understand and adapt their processes and to agree on global processes. During run time, the repository can be queried by partners for information relevant at run time, for example in the monitoring of a collaborative business process to check how a technical event corresponds to a business event. Accordingly**,** Figure 83 illustrated that the models stored in the repository are used by the execution environment as well as by the monitoring and controlling components. Hence, the repository has the following functionalities:

- Design time model repository: During design time, collaboration partners can communicate via the repository and publish public and global models on a conceptual, technical and execution level for each other.
- Directory: During the preparation of a collaboration, the BII-repository can be accessed by partners that are interested in the services of the organization owning the BII. Thus, comparable to an SOA registry, the BII can support the discovery of services.
- Contract: After the parties agreed on the specifics of the collaboration, the repository preserves the descriptions of all elements an organization offers or can expect in the context of a specific collaboration. Hence, the contents of the repository can also be understood as a contract, which bindingly describes the services of each collaboration partner.
- Support of monitoring and controlling: The information on the horizontal and vertical correlation of models stored in the repository can be used for the monitoring and controlling of collaborative business processes. For example, the completion of a technical BPEL activity can be mapped to the corresponding EPC function on the business level.

Or, representing an example of a horizontal mapping, the completion of an internal function can be mapped to the corresponding global function visible to collaboration partners.

On the Decentral Implementation of the Repository As mentioned before, each organization publishes its own BII-repository; thus, a decentral approach is followed, where the data needed inside a collaboration sphere is distributed among the various partners. Alternatively, a central approach could be followed, where all collaborating parties store shared data in a central repository.[831] The advantage of such a central approach is that inconsistencies among the interfaces of the collaborations partners would be easier to identify. However, in the AIOS no individual collaboration partner should have the responsibilities and rights related to the management of a central repository. Another advantage of a decentral solution is that a decentral repository supports the loose coupling of organizations: instead of a physical, central integration of the collaboration data, each organization remains the owner of its interface descriptions; the integration of the interfaces remains logical and can easily be modified or aborted. To implement the functionalities described above, the repository should offer the following mechanisms:[832]

- Querying the repository: On the one hand, the BII-repository can be queried by modules of the run time environment, for example, by monitoring modules that need information on the correlation between technical and business-level models. On the other hand, the repository can also be queried during design time. For the latter case, query mechanisms for the different BII elements should be supported, for example, to discover roles associated with specific functions.
- Differentiate between external user groups: Since one organization can be active in different collaborations and might want to describe its services also to potential partners, different types of external organizations with different access rights can be distinguished; for example, unknown organizations, former collaboration partners, current collaboration partners and partners in collaboration spheres that directly interact with the organization owning the repository.
- Ensure the consistency of global models: It has to be ensured that the partners in the collaboration sphere refer to the same global models and that, in case a global model consists of different public models, the public models are complementary with each other. Since global models also act as contracts among the organizations, it is important that changes to global models are synchronized with affected partners. Thus, mechanisms to coordinate writing access to models used in the collaboration must be created.

5.3.3 Modeling Prototype

The VPD/GPD prototype was created to support the development of the models needed in the AIOS.[833] Apart from demonstrating the concepts related to the AIOS, the prototype also illustrates

---

[831] As proposed for example by THELING (2007), pp. 115.

[832] For a more detailed concept for the technical realization of the BII-repository refer to ZIEMANN ET AL. (2008).

[833] A first version of the tool was developed at the end of the ATHENA project; it was substantially extended in the R4eGov project. The conceptualization and implementation was guided by the author. The largest part of the pro-

the connection of the concepts to existing business process management tools, like for example the ARIS toolset. The prototype is a Java-based stand-alone tool that supports both the ARIS typical Event-driven Process Chains and BPMN. However, the focus lies on the EPC, which was extended with elements to support the modeling of collaborations and the modeling of web service-based workflows. Correspondingly, the functionality of the tools covers the two upper levels the AIOS matrix (compare Figure 85) and thus builds a basis for their transformation into executable code. To support interoperability, the import and export formats are based on XML: The GPD and the process module of the VPD use an extension of EPML format to store and import process models, and organization models are stored in OSML.[834]

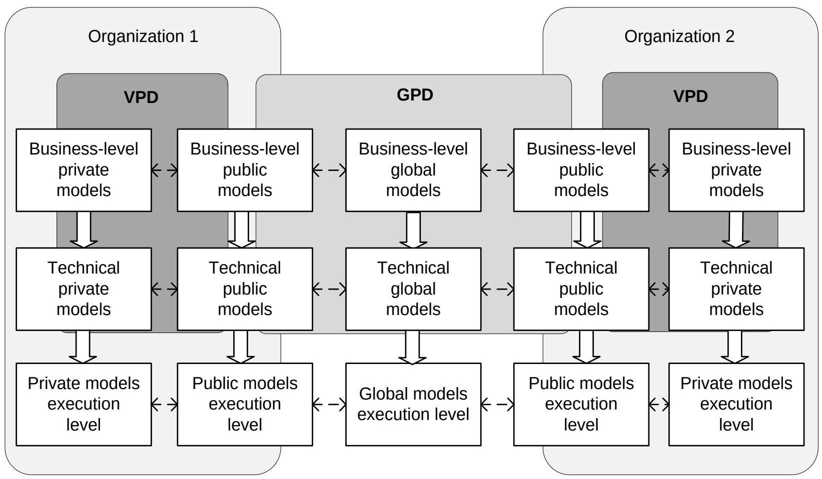


Figure 85: Model types and transformations supported by the VPD/GPD prototypes[835]

As Figure 85 indicates, the tool is split in two parts: The VPD part focuses on the relationship of private and public processes, while the GPD part focuses on the relationship of public and global processes. Further, the tools concentrate on the process and organization dimension, while the data, functional and output dimension are supported only implicitly.[836] Thus, apart from a module for separating and correlating private and public processes, the VPD contains a distinct module for modeling organization elements that enables private and public views on organization elements.

In summary, the VPD/GPD prototype offers three core functionalities: Deriving public from private processes and correlating both process types (VPD process module), describing private organization elements and their public counterparts (VPD organizational module), and assembling public processes to global processes (GPD). The implementation of the GPD and the VPD process module

gramming was done by Tobias Dumont, aided by Dima Panfilenko, Bogdan Woldert-Jokisz, Olivier Engelkes, Mirko Schackmann and Fabian Pittke.

834 OSML is described below, compare p. 206.

835 The horizontal arrows in the dark-gray VPD boxes illustrate the functionality to automatically derive public from private processes. The horizontal arrows in the light-gray GPD box illustrate the functionality of the GPD to connect public processes into one global process. The vertical arrows in the VPD and the GPD box illustrate the possibility to enrich business-level models with technical elements.

836 Note that this does not imply that the data, function and output dimension should not also be supported by corresponding modules. However, due to resource capacities, a choice for selected dimensions had to be made; and, due to their complexity, from a research standpoint the latter dimensions seemed to be more interesting.

follows the metamodel for the process dimension of the AIOS as specified in Chapter 4, while the implementation of the organizational module of the VPD is based on the metamodel for the organization dimension of the AIOS.[837]

Related Work The novelty of the AIOS compared to state-of-the-art concepts was described in Chapter 3 and 4. Since the VPD/GPD implements main concepts of the AIOS (e.g. private, public and global views on different enterprise dimensions on different levels of technical granularity), the novelty of the tools described here is implied. However, in conjunction with the state-of-the-art concepts described in Chapter 4, tools were developed that cover parts of the VPD/GPD functionality. In summary, it can be said that existing tools, which support the modeling of collaborative views focus on the execution level, support only one enterprise dimension, and usually concentrate on private and public views while neglecting the global view.[838]

5.3.3.1 VPD – Process Module

The main goal of the VPD process module is to support the derivation of public process from private process models and the technical enrichment of the resulting models. To this purpose, the graphical user interface of the VPD process module (illustrated in Figure 86) is divided in two frames: In the left frame, a private process can be edited. In the right frame, the public process resulting from the annotations in the private process is shown. The major functionalities of the VPD process module are:

- Creation of private processes on business level: In the left frame of the graphical user interface, the private process is imported and modified. To prepare the derivation of the public processes, the elements of the private process can be annotated, for example, as being “abstracted” or “aggregated”.[839]
- Deriving and correlating public processes: Based on the annotation in the private process, the public process model can be automatically derived. As a result of this derivation, elements contained in the aggregation sphere of the private process appear in the form of one new element in the public process. Elements annotated as being abstracted – as well as elements that are logically dependent from abstracted elements – are omitted in the private processes.[840] Afterwards, the private process model, the public process model and their correlation can be stored in three distinct files.
- Technical refinement of models: To prepare the transformation to BPEL, the private and public process models can be enriched with web services representations, XML-documents and scopes.

[837] See Figure 48 (p. 136) and Figure 65 (p. 163).

[838] Compare the state-of-the-art review in Chapter 4, pp. 120. Examples for such tools comprise the PAP tool from LEE & LUEDEMANN (2007), the “Nehemia” Workflow-Management-System for cross-organizational workflows from SCHULZ & ORLOWSKA (2004), the “Maestro” tool for technical process modeling created in the context of the ATHENA project (compare BORN ET AL., 2009), and a modeling environment for BPEL4Chor described by DECKER ET AL. (2008 Tool).

[839] As illustrated in Figure 86, elements abstracted in the private process are annotated with padlocks.

[840] An example for the treatment of such a logical dependency is the deletion of one branch of an AND-join. In cases where only one branch is left, the tool recognizes that the AND-split and the AND-join are now redundant and abstracts these from the public process model.

Moreover, the VPD offers a functionality to switch between process notations: process models can be illustrated either as technically enriched EPC or as BPMN. As explained below, the process module also allows to import private and public views on organization models and to connect their elements with corresponding private and public process elements.

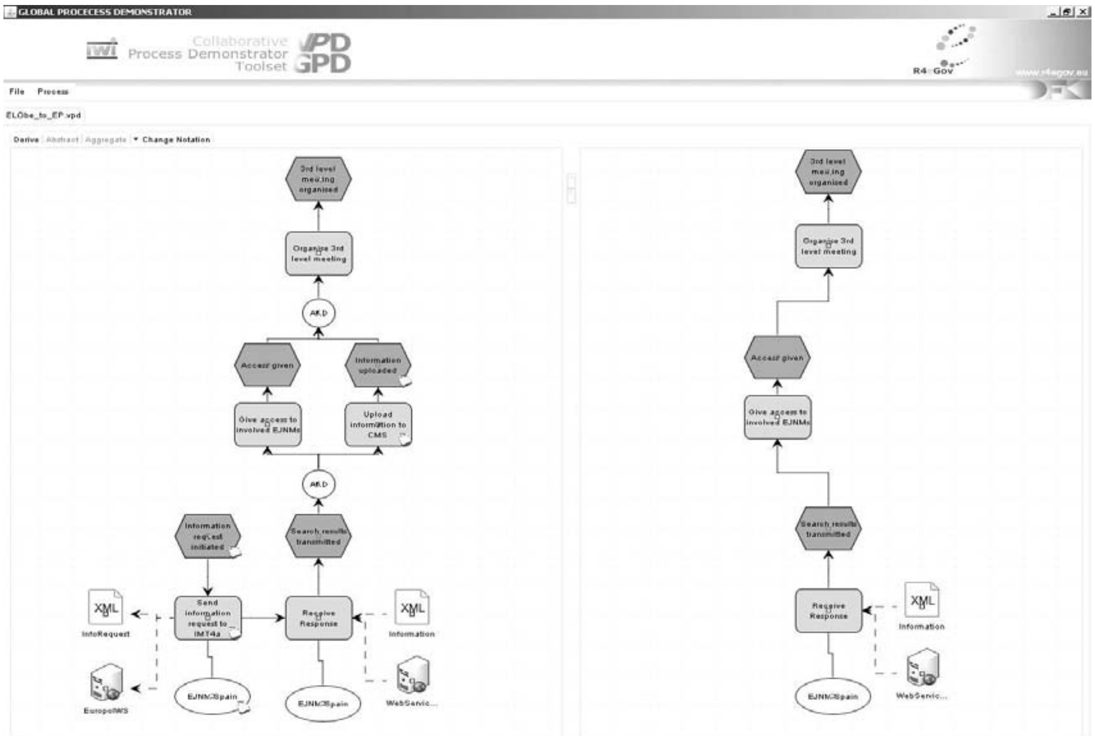


Figure 86: The graphical user interface of the VPD process module

5.3.3.2 VPD – Organizational Module

Similar to the process module in the VPD, the organizational module of the VPD allows creating private and public views on organization structures. Figure 87 shows the graphical user interface of the organizational module of the VPD. Following the public/private concept, on the left-hand side the private, and on the right-hand side the public counterpart of the organization structures are shown.

Major elements of the organization metamodel defined in Chapter 4 (compare p. 163) are supported: Positions, displayed in the upper part of the screenshot, and roles, shown in the lower part. Each element has an ID and a name; additionally, the field "reference" indicates for each organization element whether it is correlated with a function in the corresponding process model.[841] Both private and public positions can be related to superior positions, to their (public or private) counterpart and to related roles. The relationship between the elements is also illustrated by the highlighting in the graphical user interface that indicates which elements are related to the currently selected

[841] The relation between organization model and process model in the VPD is described below.

element. Thus, the user interface in Figure 87 indicates that the private role “EP_Communication” is related to its public counterpart “EJ Info Requestor” as well as to the private position “EJ NM”.

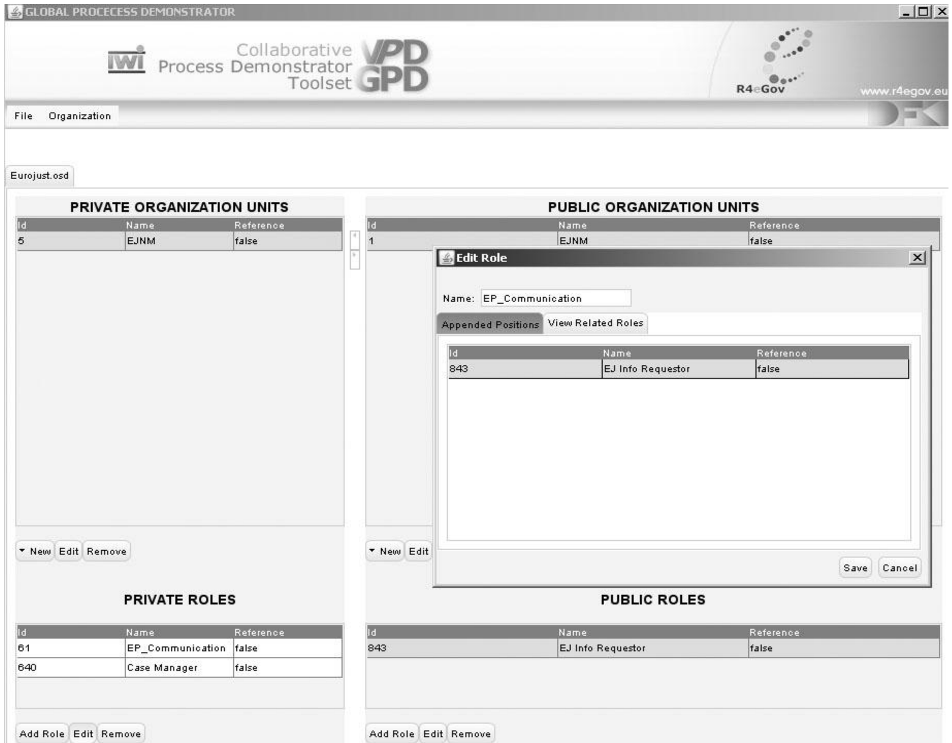


Figure 87: The graphical user interface of the VPD organizational module

5.3.3.3 Integrating Organization and Process Views in the VPD

The VPD tool also illustrates how private and public views of organization models can be integrated in process models: In the process module of the VPD, private organization units can be attached to a private process model. If a public process is derived from a private process, organization units in the public process are automatically populated with the public versions of the organization units.

Figure 88 illustrates that, in order to relate an organization model and a process model in the VPD, both models are first opened in the tool. Then, in the process view, an organization model is selected and related to the process model. Afterwards, individual functions in the private process frame can be related to elements of the organization model.[842] Subsequently, the “derive authorization” functionality of the VPD can be used to automatically attach the public counterparts of private organization elements to the corresponding functions in the public process.[843]

[842] An organization unit can be attached to a function by selecting the function and selecting “set authorization” in the context menu. In the upcoming window, all available private organization elements are listed.

[843] To this aim, the context menu of each function in the private process offers a corresponding functionality.

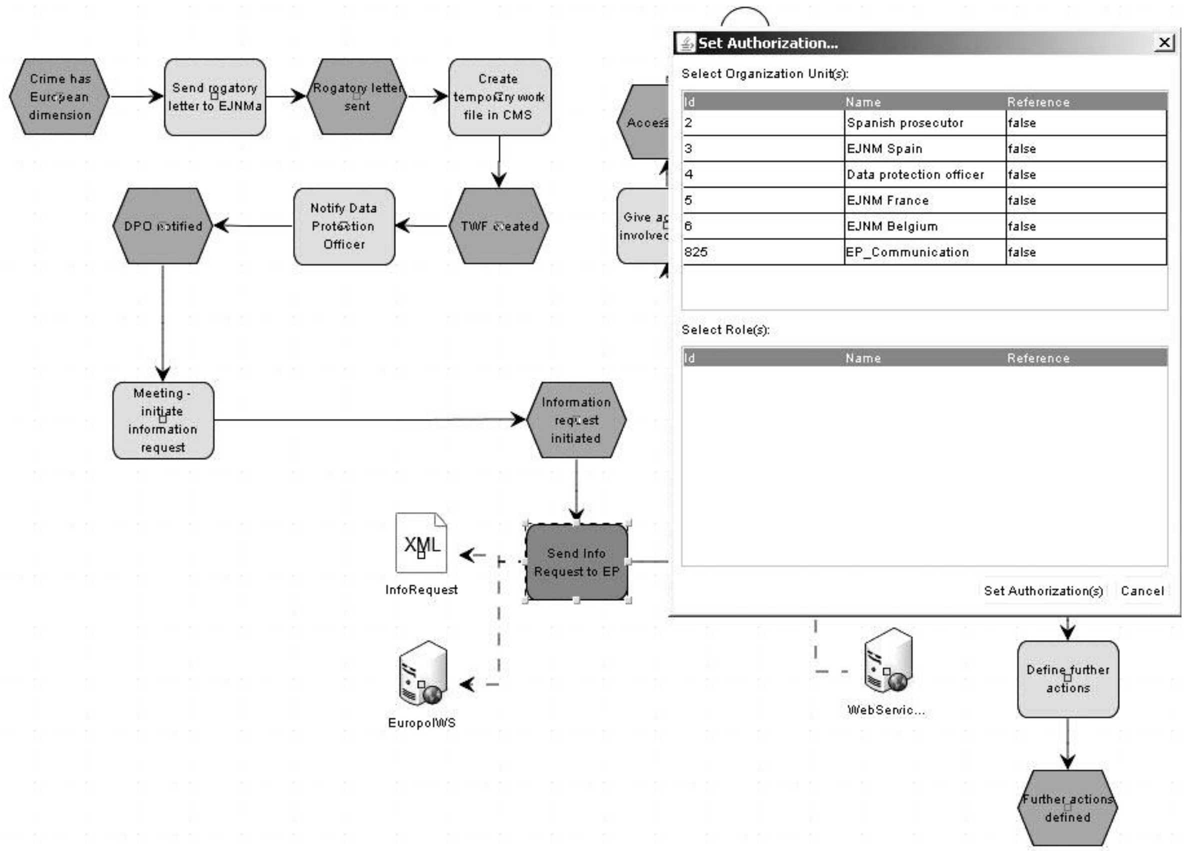


Figure 88: A private organization element is attached to a function in a private process

Figure 89 shows a corresponding screenshot: Here, the private position “EP_Communication” was attached to a function in the private process. Since this position is only used inside of EJ, the public counterpart of this position (“EJ Info Requester”), which is visible to EP, was attached automatically to the public process, as shown in the left frame of the screenshot.

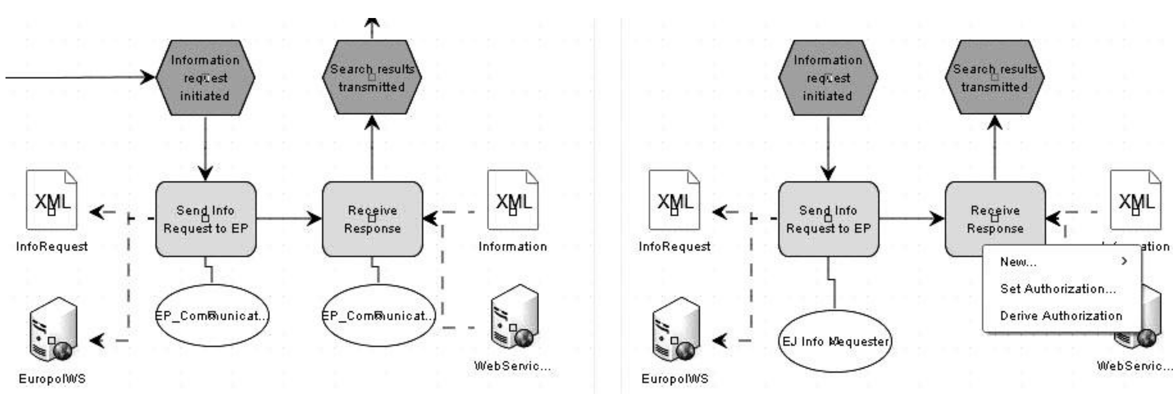


Figure 89: The public counterpart of the private organization unit is derived automatically

5.3.3.4 Connecting Public Processes with the GPD

Offering a function complementary to the VPD, the GPD imports the public processes produced by the VPD to connect them into global processes. Following the metamodel for the AIOS process dimension (compare Figure 48, p. 136), the public processes are integrated only via events. More specifically, an event outgoing from a function belonging to a public process A is connected to a corresponding event caught by a function of the adjacent public process B. Currently, in the GPD only message-based events – annotated as documents produced or consumed by functions – can be correlated with each other.[844] If public processes are to be combined into global processes, the different enterprise dimensions appearing in them must be complementary. Regarding the organization dimension, this means, for example, that the organization units related to interacting functions must match each other. Thus, in a security sensitive environment it must be ensured that interacting roles have a similar clearance level. Currently, the GPD offers the following functions to support the connection of complementary public models:

- Function dimension – complementary functions: A *sending* function must be mapped to a *receiving* function. In the GPD this is checked by the direction of the mapping between functions and the attached document; only if the flow is direct from the sending function to the document A and from the document A' in the adjacent process to the receiving function, the connection between A and A' is permitted (otherwise, an error message occurs). Going beyond the semantics of the interaction function, in future also the complementarity of the technical realization should be verified, e.g. by checking if the web services implementing the functions match each other.[845]
- Process dimension – complementary interaction sequences: If the logic expressed in the sequence of adjacent public processes is not complementary, deadlocks can occur; for example, both parties could wait for the other party to send the initial message. The GPD currently only supports a manual checking of process complementarity by illustrating the adjacent public processes and their connections. However, as described above, further model checking techniques are provided by the adjacent tools of the tool suite described above (pp. 193).
- Data dimension – congruent messages: The format of the sent document A must correspond to the document A' expected at the receiver site. The sent document must comprise at least those fields accessed by the receiving party. Currently a basic checking mechanism is implemented in the GPD, which allows a correlation between a sent and a received document only if the names of both document types are identical.

Figure 90 illustrates the checking mechanism currently implemented in the GPD. In the example, the public process on the top is to be connected with the public process on the bottom. However, as the error message on the left-hand side indicates, the messages cannot be connected since the out-

---

[844] Thus, the global process in Figure 97 comprises the information that the document “approval” coming from the function “send approval or denial of request” corresponds to the document “approval” destined to the function “receive approved search results”.

[845] In the VPD, functions inside public processes can be annotated with the source and the target web service. Building on this annotation, in the GPD it could automatically be verified whether the web services of interacting functions match each other.

going message has another format than the one expected by the receiving function. The error message on the right indicates that the semantics of the public processes does not match the message flow foreseen in the global process: In the global model, the modeling of a message flowing from the top to the bottom process was attempted (this flow is indicated by the bold, vertical arrow on the right). But since the corresponding functions inside the public processes indicate a flow in the opposite direction, an error message is produced.

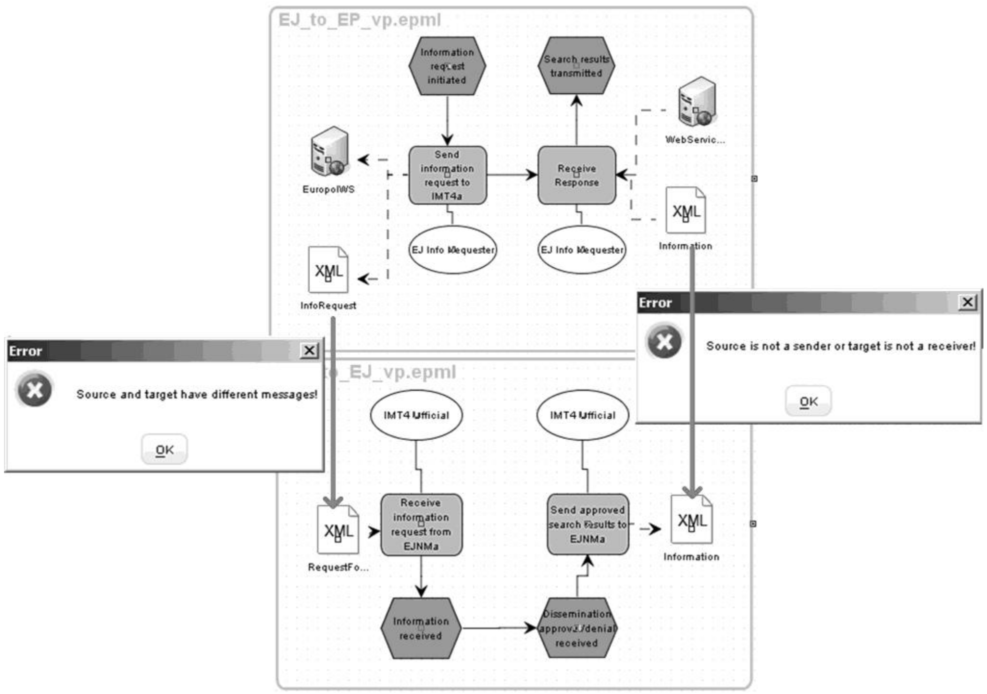


Figure 90: Error messages in the GPD indicating non-matching public process elements[846]

5.3.3.5 Mechanisms for Storing and Correlating the Collaborative Models

The principle of the AIOS to provide distinct but correlated views on interoperable information systems is displayed in the storage mechanism of the VPD/GPD tools: private, public and global models are stored separated from each other; to describe the relationship among them, correlation files are used. This storage enables for example that a public processes model can be transmitted to a collaboration partner, while the private process and the correlation of the public process to the private process can remain inside the organization.

As a storage format for the processes, EPML is used. To provide a better basis for a transformation to an executable process format like BPEL but also to more technical process notations like BPMN, the original EPML syntax was extended: XML-documents and web service representations can be attached to functions and process elements can be grouped into scopes. Furthermore, the extension supports the coupling of private, public and global processes.

[846] In the graphic, two screenshots of the GPD tool were combined: a screenshot where the error message on the left-hand side was produced and a screenshot where the error message on the right-hand side was produced.

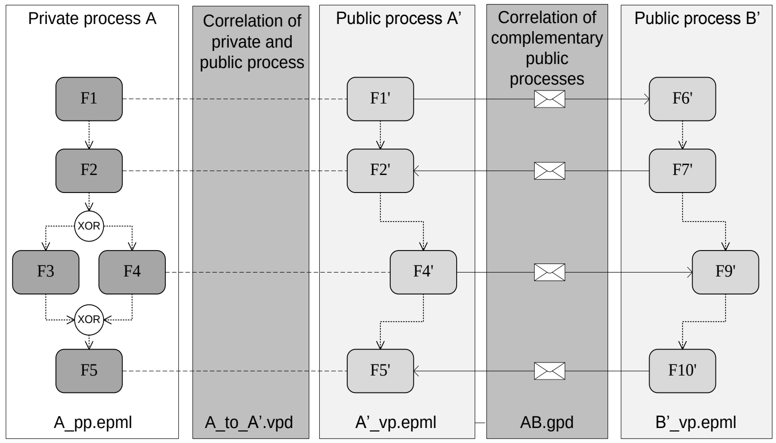


Figure 91: File types used to represent and correlate process views

As illustrated in Figure 91, four different file types are used to store processes:[847]

- Private process: The private process is stored in extended EPML, displaying only internal processes to be kept inside the boundaries of the owning organization.
- Public process: The public process is also stored in extended EPML format, and contains those parts of a private process, which are public to collaboration partners.
- Correlation of private and public process: Since it could reveal private information, the mapping between public and private processes should remain inside the organization. And, since one private process can be correlated with different public processes, this mapping should not be stored in the private process. Thus, it is stored in a separate file, which correlates the process models as well as the elements of the models.[848]
- Correlation of public processes: In the AIOS, a global process represents a set of complementary public processes that are connected via message exchanges. To store the mapping among the public processes that make up a global process, a distinct correlation file is used. Apart from the names of the public processes comprised in one global process, the file describes how the elements of adjacent public processes are correlated to each other (e.g., "function F1' of the public process A' emits the message RfQ to the function F6' in the public process B'").

Storage Format for Collaborative Views on Organization Models The data in the organization view is specified in OSML, which is defined based on the metamodel specified in Chapter 4.[849] Similar to the process dimension, the private and public views on organization structures are stored in separate files, which are also connected via a correlation file. Thus, three file types are specified:

---

[847] Example instances of these files produced by the VPD/GPD are illustrated on p. 209 and p. 213.

[848] Thus in the example, the correlation file on the left connects the public process model "A'_vp.empl" with its private counterpart "A_pp.epml"; further, it correlates individual elements of both processes, e.g. the public function F2' and the private function F2.

[849] See p. 163.

- Private organization models: The private view of the organization structures of one collaboration partner is stored in OSML. It contains all organization elements relevant for the collaboration as well as related internal organization elements not public to partners.
- Public organization models: The public view on the organization structures of one partner is also stored in OSML. Note that since the public organization model follows the syntax of the private organization model, it is also possible to derive a "public model of a public model"; e.g. to develop another organization model that represents an abstraction of the first public view. This model could then serve as a global organization model, being correlated to one or more public organization models (of different collaboration partners).
- Correlation of adjacent organization models: The mapping between two adjacent organization models – may it be a mapping of private with public or a mapping of public with global models – is stored in a separate file with the ending ".osd". As Figure 92 illustrates, all elements of an organization model can be correlated to their counterparts in the adjacent model. The example also illustrates the case where an element of the private model ("role C") should be invisible in the collaboration, and thus does not have a counterpart in the public model.

Code excerpts from a private and a public organization model and the corresponding correlation file are provided in Figure 111 (see p. 224).

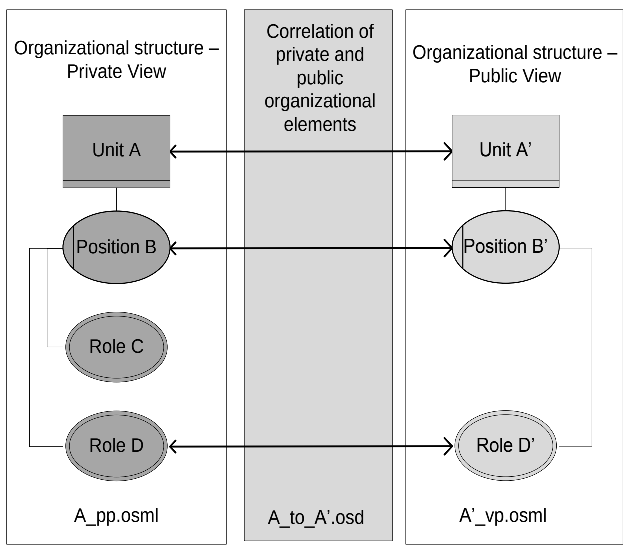


Figure 92: File types used to represent and correlate views on organization elements

5.4 Refinement and Application of Procedure Model

In the previous sections of this chapter, a procedure model for the AIOS was developed that abstracted from the different enterprise dimensions by describing an inside-out approach incorporating the three collaborative views and the three vertical levels of the AIOS. In the following, this procedure model will be verified and adapted to the specifics of each enterprise dimension. The resulting procedure model will be applied to the EP-EJ use case; here, it will also be shown how the VPD/GPD tools can be used in the development process.

5.4.1 Preliminary Agreements

As described above,[850] the AIOS focuses on the operational design of interoperable systems, while the preceding strategic phases are not tackled by this thesis.[851] However, as a pre-condition for the design activities described in the next sections, certain agreements among the collaboration partners should exist,[852] including the goals of the collaboration, the organizations forming the collaboration sphere and their roles, the modeling standards as well as other surrounding conditions.

Thus, in the example of EP and EJ, the organizations would have agreed that these organizations want to engage in a long-term collaboration aiming at the automated information exchange on European crime. They would also decide on the exact bodies forming part of the collaboration sphere, e.g. not only EP and EJ but also national organizations directly related to EP and EJ. They would agree on the usage of the BII and the corresponding technical infrastructure – including the fact that both organizations offer BII-repositories to each other that comprise the models constituting the CBP – and on the usage of certain standards.[853] In addition, they would agree to bilaterally define the interaction among themselves, roughly following the inside-out approach illustrated in Figure 77.

5.4.2 Process Dimension

In this section, the generic procedure model is adapted to the process dimension and applied to the EP-EJ use case. Thus, it is shown how EP and EP develop complementary process models to realize the collaborative business process described above (pp. 189). Apart from public processes on the business, technical and execution level, this also includes the corresponding private, public and global process models.

5.4.2.1 Procedure Model Adapted to Process Dimension

Figure 93 illustrates the generic inside-out approach (that is displayed in Figure 77) adapted to the specifics of the process dimension. On the first level, EPC are used to capture business-level requirements. On the second level, the structure of the processes is adapted to the possibilities of BPEL and is enriched with technical details. As a storage format, EPML is used for business-level processes and the extension of EPML to describe technical processes.[854] On the execution level, only BPEL is used, e.g. the global models on the execution level are represented as complementary BPEL Abstract Processes. While the transformation between business-level and technical process

---

[850] See p. 180.

[851] For a detailed description of the preparation phase in collaborations, refer for example to HOFER (2007), pp. 134.

[852] As described above, a collaboration consisting of real organizations is presumed. Nevertheless, in case an organization wants to develop a BII for still unknown, potential collaboration partners, it must determine similar goals on a strategic level.

[853] In our example, the following standards are used: EPC, BPMN and BPEL for the process dimension, OSML and XACML for the organization dimension, EPC and Web services for the function dimension and XSD for the data dimension.

[854] The technical extension of EPML for the AIOS was introduced above; compare pp. 204.

models is manual, the transformation of process models from the technical to the execution level is automated. Note, that for practical reasons described below, the transformation from the technical to the execution level (Step 9) is not executed via public models, but via private process models.

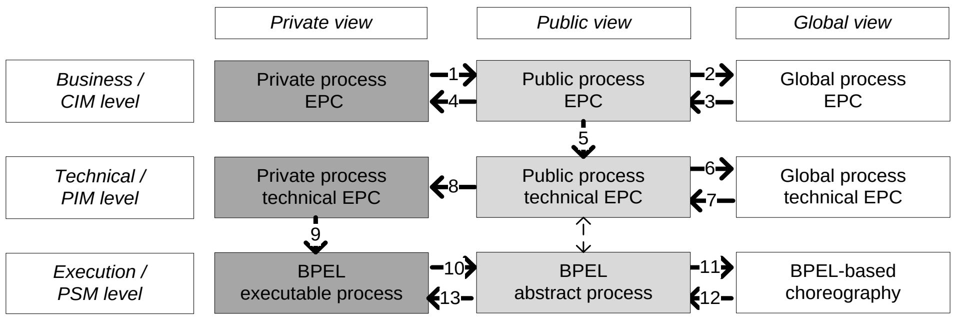


Figure 93: Inside-out procedure model specific for process dimension

5.4.2.2 Developing Processes on the Business Level (Step 1-4)

During the above-described preparation of the design phase, EP and EJ agreed on the goals and the general course of the collaboration. Though these preliminary agreements do not comprise explicit process models, they must enable the parties to develop a first proposal of business-level private processes for the collaboration. For example, in a first step, EP and EJ could agree that they will implement an asynchronous information exchange in which EP delivers search results asked for by EJ.

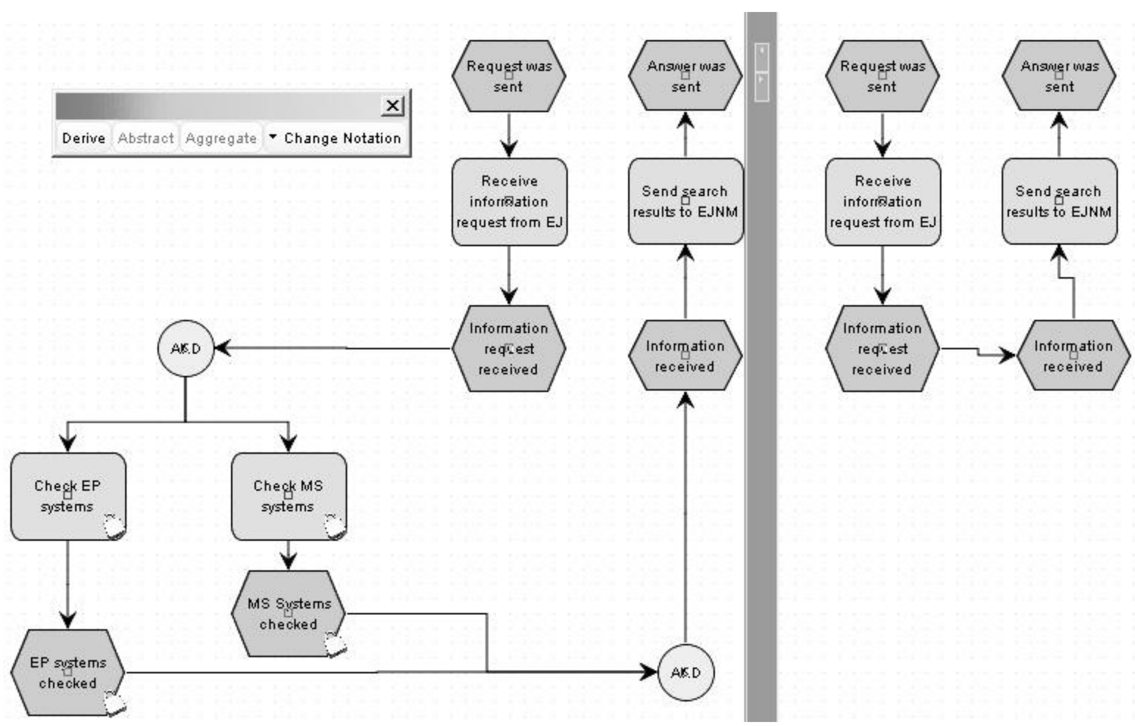


Figure 94: Using the VPD tool to derive EP's business-level public process

Based on these agreements, EP and EJ each create a process model of the private processes involved in the interaction. Thus, the left-hand side of Figure 94 displays the business-level private process of EP, comprising elements that have to be communicated to the partner organization as well as elements not directly relevant to the collaboration partners. In the example, the latter part comprises the internal activities to query different databases, while the public elements comprise the Receive and the Send activities. In order to define the public process, each collaboration partner identifies and annotates those elements in his private process that should be visible to partners. Note, that public process elements comprised in a private process might only be public for specific partners (e.g. organization B) while other organizations (e.g. organization C) should not have access to this part of the private process. In this case, the public elements of the business-level private process have to be annotated accordingly, specifying to which organizations the elements are public. For example, the private process could contain one public area visible to organization B, and one public area visible to organization C. In consequence, various partner specific public processes result from one private process.

Figure 94 also illustrates the VPD tool used to annotate the public process elements of the EP private process: The elements on the left-hand side, illustrated with padlocks, are annotated as being “abstracted” in the private process. Thus, in the public view of the tool (visible on the right-hand side) only the public elements of the process appear. While the elements of the private process were annotated manually, the VPD tool derived the control flow resulting in the public process. Thus, the private elements were not only hidden, but also the control flow edge between the events “Information received” and “Dissemination approval/denial received” was added automatically.

EP_to_EJ_pp.epml

```
….
  <event id="2">
    <name>Information request
          received</name>
    <graphics>
      <position x="30.0" y="210.0"…
    </graphics>
  </event>
  <function id="0">
    <name>Receive information request from
          EJ</name>
    <graphics>
      <position x="30.0" y="110.0" …
    </graphics>
  </function>
….
```

EP_to_EJ.vpd

```
<?xml version="1.0" encoding="UTF-8"?>
<vpd>
<privateProcess file="EP_to_EJ_pp.epml" />
<viewProcess file="EP_to_EJ_vp.epml" />
<mapping ppId="1" vpId="1" />
<mapping ppId="615" vpId="615" />
<mapping ppId="0" vpId="0" />
<mapping ppId="2" vpId="2" />
<mapping ppId="925" vpId="925" />
<mapping ppId="85" vpId="85" />
</vpd>
```

EP_to_EJ_vp.epml

```
…
<event id="2">
<name>Information request received
</name>
<graphics>
<position x="360.0" y="210.0" …
</graphics>
</event>
<function id="0">
<name>Receive information request
from EJ</name>
<graphics>
<position x="360.0" y="110.0" …
</graphics>
</function>
….
```

Figure 95: Code excerpts from the files used to represent and correlate process views

The three listings displayed in Figure 95 correspond to the three files for storing the private and public process models of Figure 94. Illustrated on the left-hand side, the file “EJ_to_EP_pp.epml” holds the information of the private process while “EJ_to_EP_vp.epml” (one the right hand side) stores the public process model. In their middle, the file “EJ_to_EP.vpd” is illustrated, which connects the private and the public process. More specifically, functions and events of the private process are mapped to their counterparts in the public processes. For example, it is indicated that the private function with the ID "0" corresponds to the public function with the ID "0".[855] The public process model of EJ is derived in the same way as described above for EP; the result is illustrated in the upper half of Figure 96.

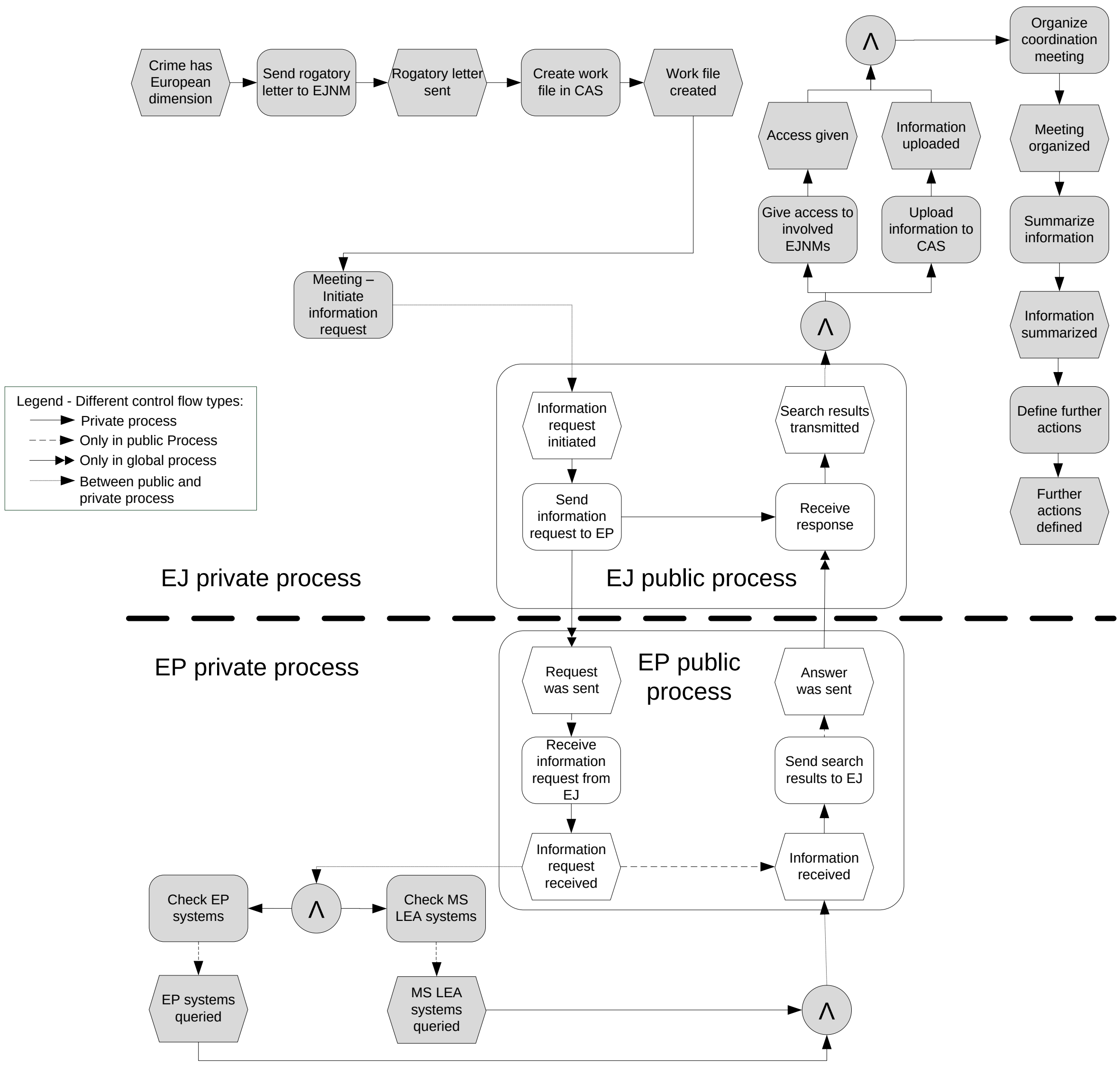


Figure 96: Business-level global process with adjacent private processes

After both parties have modeled their public processes, it has to be verified that the public process models match each other. Before an automated verification, a manual adjustment takes place in which the involved organizations exchange their public processes, adapt them if necessary and compose them into a global process (Step 2). After possible adaptations of the public processes, these are returned to the owners of the private process models, which are modified to comply with

[855] Naturally, this implies that (different from the example) the name and the IDs of the correlated elements can be different from each other.

the public processes (Step 3 and 4). Figure 96 displays the result of Steps 1 to Step 4: The complementary public processes of EP and EJ as well as the private processes related to them. The example also illustrates the different control flow types in a collaborative business process:

- Control flow inside private processes: This type comprises all control flow edges of the private process, including those that are also part of the public and global process. Note, that Figure 96 illustrates also, that not all control flow edges comprised in a public process form part of the global model.[856]
- Control flow only valid in public and global processes: If elements of a private process are abstracted, a predecessor-successor relationship exists between the element foregoing and following the abstracted area not comprised in the original private process model.[857]
- Control flow between public processes: This control flow type should be distinguished, because it represents the interaction between organizations and thus is accompanied by a message exchange. It is comprised only in global process models, not in private or public process models.
- Control flow between public and private process elements: Depending on the implementation of both process types, it can be useful to distinguish the transition between the public and the private process. For example, in case the public process is implemented as an individual executable process, this transition also represents an interaction between two processes.

5.4.2.3 Developing Processes on the Technical Level (Step 5-8)

After the agreement on the business-level global process (including on the comprised public processes), the business-level models are technically enriched (Step 5). Following the procedure model, each partner first refines its public process model; for example, by adding the specification of XML documents being sent or received by the partner or by specifying the web services that implement the business functions.

In Step 6 of the procedure model, technically enriched global processes are created. As the screenshot in Figure 97 illustrates, here the GPD tool can be used to join the technical public processes of EP and EJ into one global process: First, the separately developed public process of EP and EJ are imported in the tool. In a second step, the documents emitted by sending functions (e.g. the document “search result” related to the function “send search results to EJ”) are correlated with the documents connected with the corresponding receiving function (e.g. the document “search results” related to the function “receive response”). Here, the tool checks if the correlated document types are identical and that the interaction consists of one sending and one receiving activity. If this

---

[856] Thus, the control flow edge between the functions “send information request to EP” and “receive response” is part of EJ’s public and private process, but not part of the global process, since the global process model also comprises the activities executed by the collaboration partner, which are placed in between these two functions.

[857] An example is shown in Figure 96, where a control flow edge between the event “information request received” and the event “information received” of EP’s public process exist. Obviously, this edge is not valid in the private process, since other process elements are placed between the two events.

is the case, the XML documents of the sending and the receiving functions are accepted as connections points of the public processes.

Figure 98 shows code excerpts from the three files necessary to store the resulting global process: The two public process models comprised in the global process and a file that correlates them. The public processes are stored in extended EPML. The correlation file relates the public process models with each other ("EJ_to_EP_vp.epml" to "EP_to_EJ_vp.epml") as well as the XML documents that constitute the interaction among the public processes. Thus, the example file contains an edge that connects the document with the ID 24 (emitted by the function "send information request to EP" of the EJ process) with the document having the ID 578 (received by the function "receive information request from EJ NM" in the EJ process).[858]

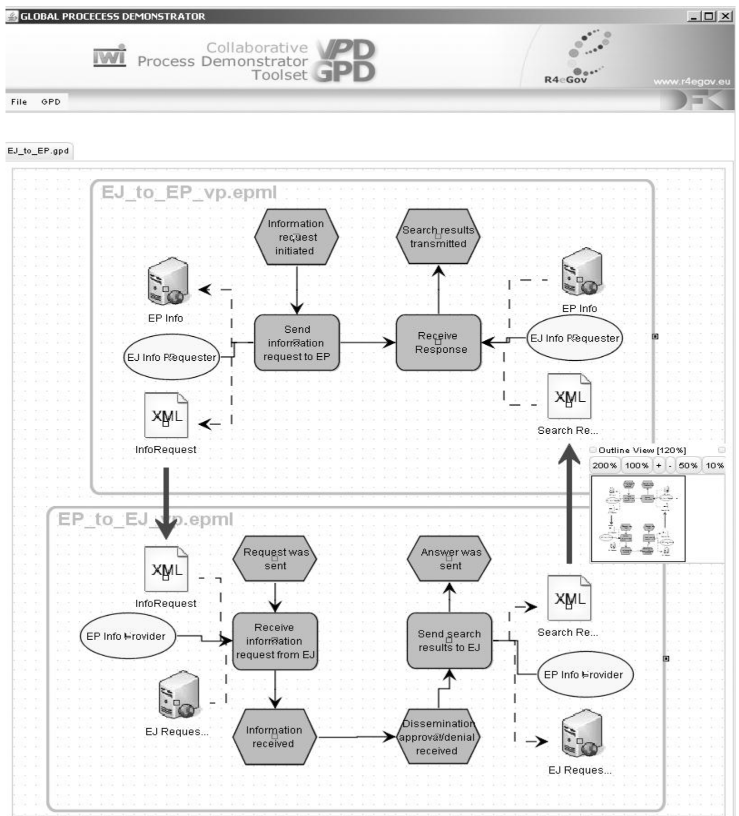


Figure 97: Using the GPD tool to compose the global process between EJ and EP

Ensuring Compliancy with BPEL The technical process model should represent the basis for an automated transformation to the execution level, e.g. to BPEL. Thus, Step 6, the refinement of the EPC, is not limited to the addition of elements like web services or XML documents. In this step, it also has to be ensured that elements existing in the technical EPC have a counterpart in

858 The IDs refer to graphical elements in the process model; thus, the correlation is not based on the document types (which could lead to confusion if the same document type is used more than once), but to documents in a specific position in the process model.

BPEL. Since BPEL can represent both graph-based and block-oriented control flows, most elements of the EPC control flow can be transformed to BPEL.[859] Thus, both languages show a comparable support of workflow patterns; in particular, they support the same basic control flow patterns.[860] If an EPC function represents an interaction (e.g. “receive message” or “send message”), it can be transformed to the similar BPEL activity. If an EPC function represents a business function that is outside the BPEL specification (e.g. “check creditworthiness”), a web service has to be created that will be invoked by the BPEL process.[861]

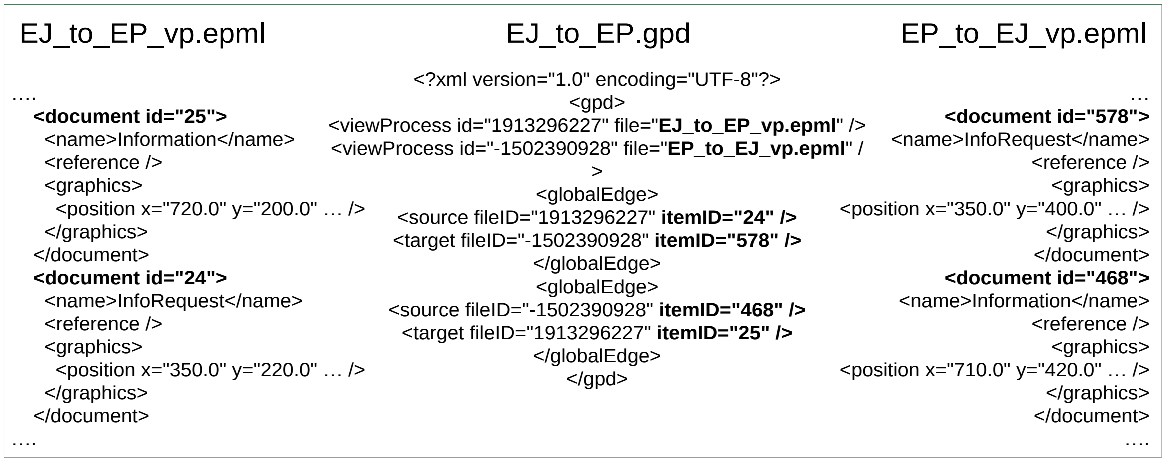


Figure 98: The GPD tool stores global processes as correlated public processes

Note that the transformation from the technical process level to the execution level does not have to be harmonized among the collaboration partners: it can differ for each organization, depending on the transformation tool they choose. KOPP, UNGER & LEYMANN for example, describe a BPEL generation based on a deviation of the EPC in which functions and events do not have to alternate.[862] The ARIS toolset, on the other hand, presumes an input format that is compliant with BPEL’s block-oriented control flow structures (e.g. loops),[863] and the transformation described in ZIEMANN & MENDLING uses the graph-oriented control flow elements of BPEL (e.g. edges between activities). Accordingly, the profile of the technical processes necessary for the BPEL transformation might also differ among the partners. However, this does not lead to interoperability problems, neither on the technical nor on the execution level: Since the profiles for technical processes used by the different collaboration partners should represent subsets of a commonly used profile for technical processes, the technical processes can be exchanged and understood among the partners. Syntactical interoperability on the execution level is also ensured, since the target models of the transformation must be compliant with the BPEL specification.

---

[859] An exception is for example the OR-operator of the EPC; for specifics of the transformation between EPC and BPEL refer to ZIEMANN & MENDLING (2005).

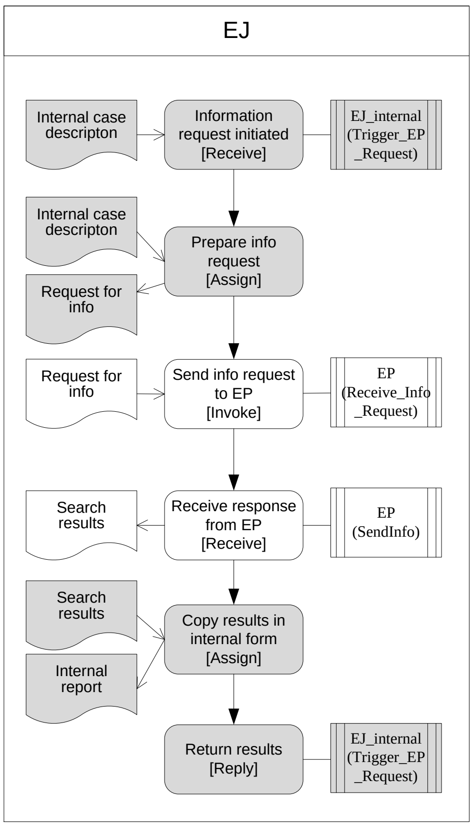
[860] Compare MENDLING, NEUMANN & NÜTTGENS (2005) and WOHED ET AL. (2002).

[861] For a more detailed description of the compliancy of EPC models with BPEL refer to ZIEMANN & MENDLING (2005).

[862] Compare KOPP, UNGER & LEYMANN (2006).

[863] Compare STEIN & IVANOV (2007).

In Step 7, the global process is decomposed into public processes and distributed to the partners, who, in Step 8, complement their public processes in order to form a private process. Presuming that possible modifications of the public processes during the forming of the global process are accepted by the individual collaboration partners, both steps are straightforward. In Step 7, only the EPML files stored by the GPD tool are returned to the partners. In Step 8, private process elements are added to the process stub, possibly using the VPD tool. The left-hand side of Figure 99 illustrates the result of these steps: The private process of EJ as a technical process model, representing a valid basis for a transformation to BPEL. The elements illustrated transparently represent the public process, while the gray elements represent the private elements complementing the public process.

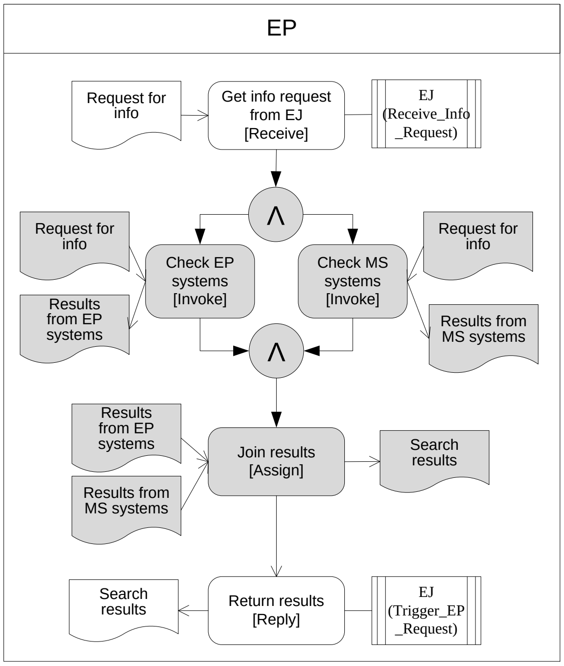



```
<sequence>
  <receive
    name="ReceiveRequest"
    partnerLink="EJ_Internal"
    portType="wsdlNS:Initiate_InfoExchange"
    operation="Trigger_EP_Request"
    variable="inputVar"
    createInstance="yes">
  </receive>
  <assign name="PrepareRequest">
    <copy>
      <from variable="inputVar" part="inputType"/>
      <to variable="partnerInputVar" part="inputType"/>
    </copy>
  </assign>
  <invoke
    name="Send_Info_Request"
    partnerLink="EP"
    portType="partnerNS:Requests_to_EP"
    operation="Receive_Info_Request"
    inputVariable="partnerInputVar">
  </invoke>
  <receive
    name="Receive_Response"
    partnerLink="EP"
    portType="partnerNS:Replies_from_EP"
    operation="Send_Info"
    variable="partnerOutputVar"
    createInstance="no">
  </receive>
  <assign name="EditResponse">
    <copy>
      <from variable="partnerOutputVar" part="resultType"/>
      <to variable="outputVar" part="resultType"/>
    </copy>
  </assign>
  <reply
    name="ForwardResults"
    partnerLink="EJ_Internal"
    portType="wsdlNS:Initiate_InfoExchange"
    operation="Trigger_EP_Request"
    variable="outputVar"/>
</sequence>
```

Figure 99: Private and public parts of EJ's BPEL process

5.4.2.4 Developing Processes on the Execution Level (Step 9-12)

From Technical Private Processes to Executable BPEL (Step 9) Focusing on the coherency of the process interfaces on different vertical levels, in the generic inside-out procedure model, the transformation between technical and execution level is executed via the public models (compare Figure 77). However, since to our knowledge all existing approaches for a transformation of conceptual processes to BPEL only refer to BPEL Executable Processes and not to BPEL Abstract Processes, in the process dimension the transformation to the execution level is executed via the private process models (compare Figure 93). Thus, first the technical private processes are transformed into BPEL executable processes; afterwards, abstract BPEL processes are derived from the executable BPEL processes. To transform the technical process models into BPEL, various mechanisms and corresponding tool support exist.[864] In Figure 99 an excerpt of EP's BPEL Executable Process is displayed that was derived from the corresponding technical private process illustrated on the left; in Figure 100, the complementary BPEL process from EJ is shown.[865]




```
<sequence>
  <receive
    name="GetInfoRequest"
    partnerLink="EJ"
    portType="wsdlNS:Requests_to_EP"
    operation="Receive_Info_Request"
    variable="SearchRequest"
    createInstance="yes">
  </receive>
  <flow name="Flow1">
      <invoke name="CheckEPS" partnerLink="EP_Systems_PL_Type"
              operation="Query_EP_Systems"
              portType="wsdlNS:EP_Systems_PortType"
              inputVariable="SearchRequest"
              outputVariable="Results_EPS_var"/>
      <invoke name="CheckMSS" partnerLink="MSS_System_PL"
             operation="Query_MS_Systems"
             portType="wsdlNS:MS_Systems_PortType"
             inputVariable="SearchRequest" outputVariable="Results_MS_var"/>
  </flow>
  <assign name="JoinResults">
    <copy>
      <from>concat($Results_MS_var.part1, $Results_EPS_var.part1)
         </from>
      <to variable="SeachResultsVar" part="resultType"/>
    </copy>
  </assign>
  <invoke
    name="ReturnResults" partnerLink="EJ"
    portType="wsdlNS:Replies_from_EP" operation="Send_Info"
    inputVariable="SearchResultsVar">
  </invoke>
</sequence>
```

Figure 100: Private and public parts of EP's BPEL process

Deriving Public Process on the Execution level (Step 10) In general, two ways of implementing BPEL-based public processes are feasible: First, implementing the public process separate from the related private process in the form of an BPEL Executable Process. Second, implementing the public process as part of a BPEL Executable Process and using BPEL Abstract Processes to describe the public process elements to collaboration partners. The first approach has the disadvantage that the executable process representing the public process could comprise internal activities that should be disclosed, e.g. how the executable process is invoked and replies to internal systems. Thus, a collaboration partner would not only see interactions directed to him, but also interaction activities directed to internal systems. In addition, this approach is only possible if the public process represents a continuous sequence, which interacts with internal systems only at the beginning and

---

864 A detailed description of the EPC-to-BPEL transformation is not in scope of this work. However, an overview of existing approaches to transform EPC into BPEL was given in Chapter 3 (p. 106), including for example the approaches of ZIEMANN & MENDLING (2005), who described a mapping from EPC to BPEL, KOPP, UNGER & LEYMANN (2006), who provided a detailed algorithm for the automatic transformation of EPC models into BPEL models and STEIN & IVANOV (2007), who described the implementation of an EPC-to-BPEL transformation in the ARIS toolset.

865 For illustrative reasons, the excerpts show only the core of the BPEL processes, e.g. the activities comprised in a process and their sequence. In addition to the sequence, the example BPEL process also comprises the definition of "partner links" and variables. Further, the example BPEL process is related to WSDL and XSD files, which describe the static interfaces of the process and define the messages used by the processes.

the end of the process.[866] However, the second approach has the disadvantage that (at least) two overlapping models of the same public process exist, since the public process elements are stored in both the private and the public process model, which might lead to synchronization problems. Nonetheless, since the form of the public process should not be restricted by the implementation form, the second approach – based on BPEL Abstract Processes – is used here to implement public processes on the execution level.

The main difference between BPEL Executable and BPEL Abstract Processes is that the abstract process implicitly or explicitly (by using the “opaque” element) omits elements of the executable process. Thus, a BPEL Abstract Process can be derived from a BPEL Executable Process by leaving out elements that should not be visible to collaboration partners. As mentioned above, BPEL offers two profiles for abstract processes: One focuses on internal processes and abstracts mainly from “arbitrary execution details”,[867] the other focuses on cross-organizational processes, where partners have to describe their public processes to each other. The latter profile for observable behavior explicitly aims at “hiding internal processing”[868]. Applying this profile, the public process of EJ would comprise the *invoke* and the *receive* activity illustrated in the middle of Figure 99, while the private activities illustrated in gray would be omitted in the abstract BPEL process. Correspondingly, the abstract process of EP (illustrated in Figure 100) would comprise only the first *receive* and the last *reply* activity, while the process elements in the gray area are left out.

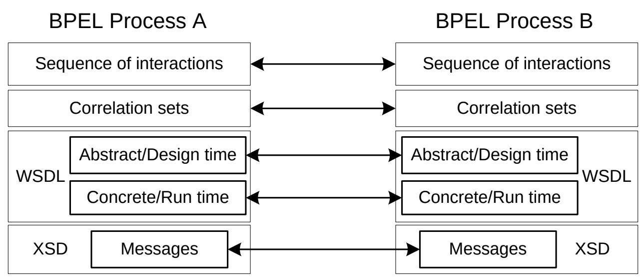


Figure 101: Overview of BPEL elements to be harmonized in a choreography

Composing BPEL Abstract Processes into Choreographies (Step 11) Due to the model-driven procedure, the core of the abstract processes comprised in the choreography (e.g. sequence of interactions, involved web services, and message types) should be complementary. Thus, only elements specific for the execution level have to be synchronized among the public processes. Figure 101 provides an overview of the elements necessary to synchronize two BPEL processes with each other in order to form a valid choreography. Apart from the *sequence* of the interaction activities, also the *message* formats, specified via XSD, have to be harmonized. To ensure the correlation of messages with process instances, it has to be ensured that the messages comprise the information required in the BPEL *correlation* sets. The *WSDL files* also have to be synchronized: port types and operations have to be specified and referenced, at least on the abstract level, and in order to execute the

[866] The public part visible in Figure 99 for example, could be implemented as one separate BPEL Executable Process, whereas the public process illustrated in Figure 100 could not be implemented as a BPEL Executable Process because it is interrupted by private process elements.
[867] OASIS (2007), p. 159.
[868] OASIS (2007), p. 156.

processes, the WSDL binding information needed at run time must be specified as well. Nevertheless, the validation of most adjustments mentioned above is covered by current BPEL design tools, including the assessment of the WSDL specifications of collaborations partners via network connections. Using the tools described above (see p. 192), the sequence of the choreography can be validated automatically as well.

Similar to Step 7 and 8 of the procedure model, in Step 12 and 13, the BPEL abstract processes comprised in the choreography are returned to the owning organizations. In case the public processes have been modified, in Step 13 it has to be ensured that the BPEL Abstract Processes are still complementary with their executable counterparts.

5.4.2.5 Summary

The generic inside-out procedure model was adapted to the process dimension and applied to the EP-EJ use case. The procedure model started with each party describing their business-level private processes with EPC models. Using the VPD tool, public processes were derived from the private processes, which later were composed into global processes. The loose coupling of the resulting EPC process models was realized via separate files, which correlate private with public models and public models with each other in order to form global models. On the technical level, the process models were enriched with elements needed for a transformation to BPEL, e.g. services and XML documents. Moreover, rules to ensure the compliancy of the process models with BPEL were applied. On the execution level, the BPEL code corresponding to the technical private and public process models was derived. In the last step, it was described how the public processes from EP and EJ are correlated into a complete choreography. As a result of this procedure, both EP and EJ possess private, public and global process models on all three vertical levels. In other words, the process dimension of the AIOS, including the corresponding parts of the Business Interoperability Interfaces, was completely specified.

5.4.3 Function Dimension

In the following, the inside-out procedure model specific for the function dimension of the AIOS is specified and applied to the EP-EJ use case. Thus, it is demonstrated how the collaboration partners develop private, public and global functions on different levels of technical detail. As mentioned above, the process and the function dimension are closely related. This implies that if the process dimension has already been defined, the function models could be derived from the process dimension. However, in order to support different sequences in which the enterprise dimensions can be developed, the development of the models in the function dimension is described independently from the models created in the previous section.

5.4.3.1 Procedure Model Adapted to Function Dimension

Figure 102 illustrates the generic inside-out approach from Figure 77 adapted to the function dimension. To ensure that the three vertical levels in the function dimension are congruent with the corresponding vertical levels in the process dimension, modeling notations congruent to EPC, technical EPC and BPEL were chosen. Thus, on the business level, function trees are described. On the technical level, these business functions are enriched with technical details needed for the forming of executable components. Enabling their implementation as web services, on the execution level these components are described with WSDL.

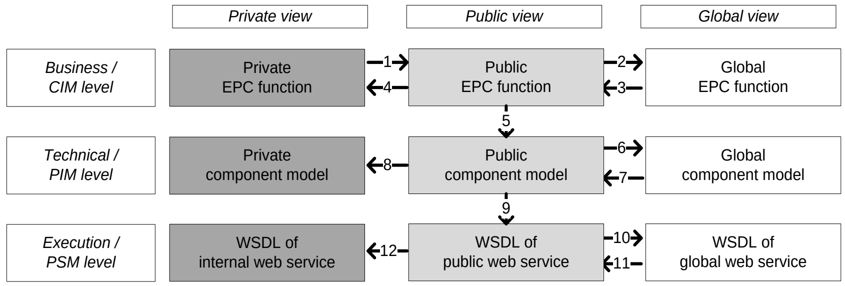


Figure 102: Inside-out procedure model adapted to the function dimension

5.4.3.2 Developing Functions on the Business Level (Step 1-4)

In Steps 1 to 4 of the procedure model, the functions offered by each partner organization are specified on the business level. Before Step 1 starts, the partners have already agreed on the overall goals of the collaboration. Based on these goals, each organization identifies the internal functions that it will contribute to the collaboration. In Step 1, a public stub of the private function is created specifically for this collaboration. This step could comprise measures to protect internal knowledge, which should not be visible in the collaboration, or it could comprise the creation of a variant of the internal function aligned to the needs of the collaboration. In Step 2, the public functions are discussed among the collaboration partners; if they each agree on the functionalities offered the others, these functions are seen as global functions. In case standardized reference functions suited for the collaboration exist, the global functions could also be based on these. In Steps 3 and 4, the public and private functions are adapted to the global functions.

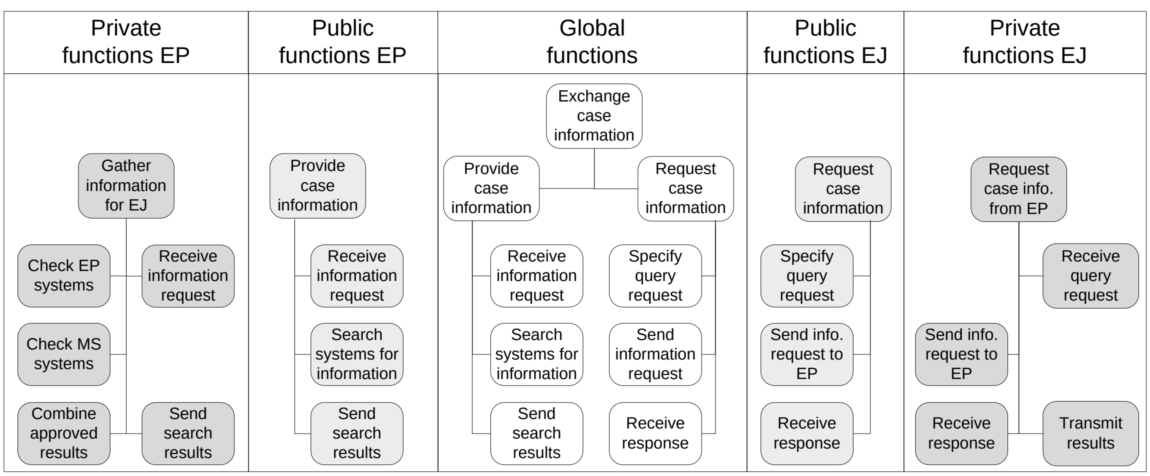


Figure 103: Private, public and global functions in the EP-EJ scenario

Figure 103 shows the result of these steps on the EP-EJ scenario. In the middle, the global function tree is illustrated. The overall collaboration fulfils the function “exchange case specific information” between EP and EJ. Below this function, the function trees to be fulfilled by each party are depicted (“request” and “provide case specific information”). Since the public function trees were synchronized with the global requirements, the function trees in the public views correspond to the sub-functions in the global function tree. The private views on the other hand, differ from the public views: First, the internal naming of the functions is different; second, the private functions comprise additional operations for the communication with internal systems, which are hidden from the collaboration partners. Thus, the public function of EP “search systems for information” is refined in two functions for querying the databases of EP and MS and a function to combine the results.

5.4.3.3 Developing Functions on the Technical Level (Step 5-8)

In Step 5, the business-level models of the public function trees are transformed into technical specifications, which describe how individual functions can be implemented. For example, it is specified whether a function is implemented as an individual component, or only as an operation of a component. Moreover, it is described which input and output parameters the operations of each function have. In Step 6, the descriptions of the public technical functions are synchronized among the collaboration partners in order to prepare the implementation of globally accepted models of the software components. In Steps 7 and 8, the public and private descriptions of the technical models are adapted to the global specifications.

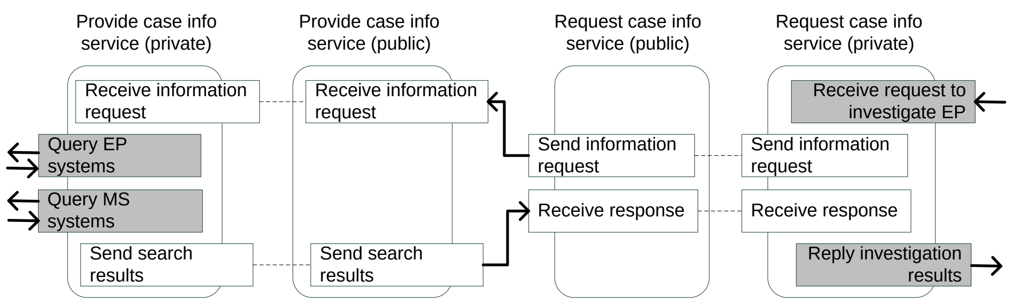


Figure 104: Model of private and public components in the EP-EJ interaction

Figure 104 illustrates the corresponding model of the EP-EJ example. The public functions of EP were implemented in the “Provide case info service” which has two public operations to interact with EJ. Note, that the function “Search systems for information” contained in the business-level model in Figure 103 is not modeled as a public operation, since the implementation of this function is an internal matter. Thus, the operations to realize this function are only shown in the private model of the service in the form of the operations “Query EP systems” and “Query MS systems”. Another function of the business-level model of EP’s service is not modeled on the technical level either: the function “Combine approved results”. The reason for this omission is the decision that – instead of using a separate software component – the function should be realized within the component “Provide case info service”. Similarly, the private component of EJ contains additional operations, which serve to communicate with internal stakeholders, and thus are not published either.

5.4.3.4 Describing Functions on the Execution Level (Step 9-12)

Based on the technical model, in Step 9 of the procedure model, each collaboration partner derives an execution level interface in the form of WSDL. In this interface description, the operations of each function as well as the messages received by each operation are described, optionally also the physical binding of the service. Similar to Step 6 on the technical level, in Step 10 the WSDL descriptions of the public services are synchronized among the partners. If they agree on the WSDL specifications, the public services can be used in the collaboration as globally accepted services. In case modifications are necessary, in Steps 11 and 12 the public and private WSDL interfaces are adapted to the specifications of the global service model.

As a result of these steps, both EP and EJ have specified their private and the public views on their collaborative services in WSDL. Since before they agreed on the public views and ensured that these are complementary, the public models at the same time represent globally valid service descriptions. Figure 105 shows the WSDL specification of the private view on the “EP information service”. Only in this view, both the private operations (displayed in gray) and the public operations are visible; in the public WSDL description of the service, the private operations are omitted. Since various WSDL files can be related to one service, two different WSDL descriptions are generated: While the WSDL file public to EJ contains only public operations, the private WSDL file additionally comprises the operations needed only internally.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions…
        <wsdl:message name="SearchRequest"> ...
        <wsdl:message name="SearchResults"> ….
        <wsdl:message name="Results_EPS"> ….
        <wsdl:message name="Results_MSS"> ….
        <wsdl:portType name="Requests_to_EP">
                <wsdl:operation name="Receive_Info_Request">
                        <wsdl:input message="tns:SearchRequest"/>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:portType name="Replies_from_EP">
                <wsdl:operation name="Send_Info">
                        <wsdl:input message="tns:SearchResults"/>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:portType name="EP_Systems_PortType">
                <wsdl:operation name="Query_EP_Systems">
                        <wsdl:input message="tns:SearchRequest"/>
                        <wsdl:output message="tns:Results_EPS"/>
                </wsdl:operation>
        </wsdl:portType>
        <wsdl:portType name="MS_Systems_PortType">
                <wsdl:operation name="Query_MS_Systems">
                        <wsdl:input  message="tns:SearchRequest"/>
                        <wsdl:output message="tns:Results_MSS"/>
                </wsdl:operation>
        </wsdl:portType> …
</wsdl:definitions>
```

Figure 105: Private and public parts in the WSDL of EP’s information service

5.4.3.5 Summary

The generic inside-out procedure model for developing elements of CBP was adapted and applied to the function dimension. On the business level, private, public and global functions were displayed as function trees, where the elements of the global function tree were mirrored in the public functions of EJ and EP. It was demonstrated that the public function trees represent subsets of the private function trees, which contained additional functions for internal purposes. On the technical level, components were defined for implementing the private and public functions trees. In this step, it was also decided which functions should be implemented as components or as an operation of a component. On the execution level, the implementation of the different component types with WSDL was shown. As a result of this step, each organization possesses one private WSDL file and one public WSDL file for the collaboration partner. In order to be compliant with existing standards like WSDL, the development focused on functions and their operations.[869] Thus, after applying the procedure model, the function dimension of the AIOS including the corresponding parts of the Business Interoperability Interfaces is completely specified for both EJ and EP.

5.4.4 Organization Dimension

In this section, the inside-out procedure model is adapted to the organization dimension and demonstrated on the EP-EJ use case. Thus, it is illustrated how both collaboration partners can systematically develop the complementary organization elements that are necessary for the description and the enactment of the scenario.

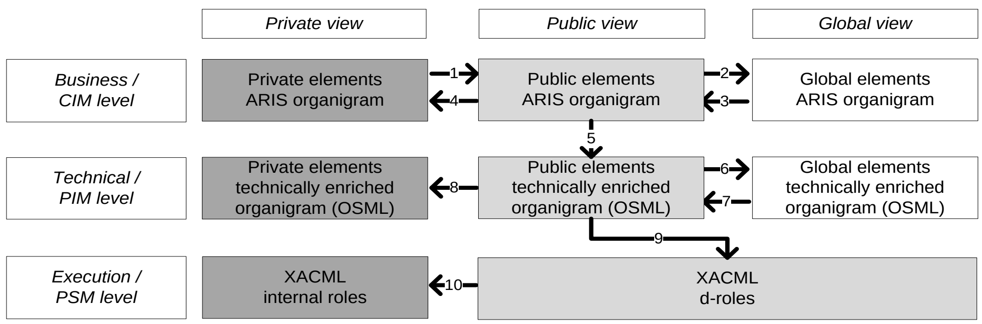


Figure 106: Inside-out procedure for developing organization elements in CBP

5.4.4.1 Procedure Model Adapted to Organization Dimension

Figure 106 shows the adaptation of the generic inside-out approach to the organization dimension. The procedure model focuses on the technical level, where OSML is used to specify private, public and global organization structures. On the business level, various formats can be used, including those of ARIS' organization dimension. Since the OSML format is aligned to ARIS, ARIS models can – realizing a model-driven development – easily be mapped to OSML models. Nonetheless, different, less formalized organization modeling forms can also be used on the business level. As described in Chapter 4, organization models in collaborative scenarios should support three different types of roles: competencies, rights and responsibilities.[870] However, the allocation of responsibilities to actors is mainly an internal, not a collaborative endeavor. The description of competencies, on the other hand, is of most interest in the context of dynamic service discovery, which is outside the scope of this work. Thus, the technical level of the organization dimension focuses on the description of rights and roles in collaborations based on XACML. Since in the collaborative extension of XACML only internal and distributed roles are regarded,[871] on the execution level of

[869] Due to the fine-grained form of the services, their naming combined with the models of their input and output documents already described their functionality sufficiently, thus an explicit goal modeling was not necessary. Events were not modeled explicitly, either. This was implied by the example where no time-based events were needed. However, events are implicitly part of the models, since the operations are connected with incoming and outgoing messages, which can be interpreted as output-based events.

[870] See Chapter 4, p. 163.

[871] Compare also the description of distributed XACML in Chapter 4, p. 156.

the procedure model only private and global models are represented.[872] The distinction between public and global roles is most important on the business and technical level, where – during the design phase – differences among proposed public roles and the public roles needed in the global process exist. On the execution level, the roles of the public view should be identical with those in the global view.

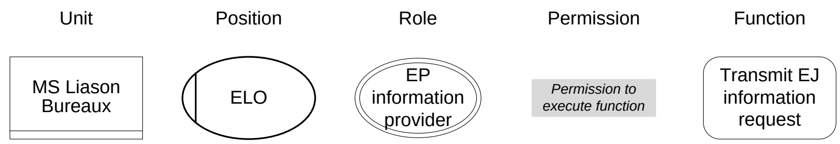


Figure 107: Symbols used to model organization elements

The notation to model organization elements in the following is based on the symbols used in ARIS; the main elements being organization units, positions, roles, permissions, functions and resources attached to functions (compare Figure 107). Objects that are accessed by roles can be displayed by different symbols, like for example document or database symbols.

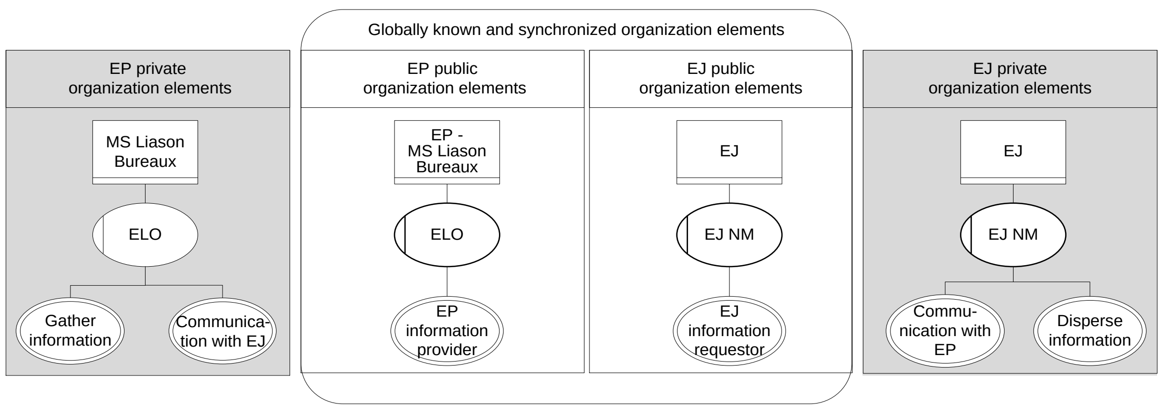


Figure 108: Private and public organization elements from EP and EJ on the business level

5.4.4.2 Developing Organization Elements on the Business Level (Step 1-4)

In the first phase of the procedure model, the partners agree on the main organization units and actors involved in the collaboration. The inside-out procedure starts with coarse-grained, strategic organization models of each organization.[873] From these models, those organization elements are extracted which are of relevance to the collaboration partner (Step 1). The partner organization likewise develops a public organization model. In Step 2, the collaborating organizations compare their public models and – if necessary – modify them to be complementary. One possibility for depicting a business-level global organization model is illustrated in Figure 82, where the collaboration of EP and EJ is prepared. Another possibility aligned to the ARIS annotation is illustrated in

[872] Note that this does not contradict the private/public/global concept, since a global process in the organization dimension consists of complementary public roles.

[873] Examples of strategic organigrams that display only the main organization elements and their relationships of Europol and Eurojust can be found in EUROPOL (2008), p. 31, and EUROJUST (2008).

Figure 108. In this model, apart from the global elements, also the private counterparts of EP and EJ are displayed. In Step 3, the possibly modified public organization models are returned to the individual organizations, and in Step 4, any changes that result from the collaboration agreement are transmitted to the private organization model.

5.4.4.3 Developing Organization Elements on the Technical Level (Step 5-8)

Step 5 In this step, the business-level model is transformed in a detailed model that serves as a basis for a subsequent generation of access control models. In case ARIS models were used on the business level, these models are now enriched with finer-grained elements needed for access control, like for example roles, functions and resources. If other notations were used, now the relevant organization units, positions and roles are extracted from these business-level models and are likewise enriched with fine-grained elements. Thus, from the model shown in Figure 82, a fine-grained interaction can be modeled based on an organization model that describes in detail the involved organization units (EP MS Liason Bureaux vs. EJ), positions (ELO vs. EJ NM) and the public roles (EJ Information Requester vs. EP Information Provider). A corresponding model of the public organization elements is illustrated the middle of Figure 109.

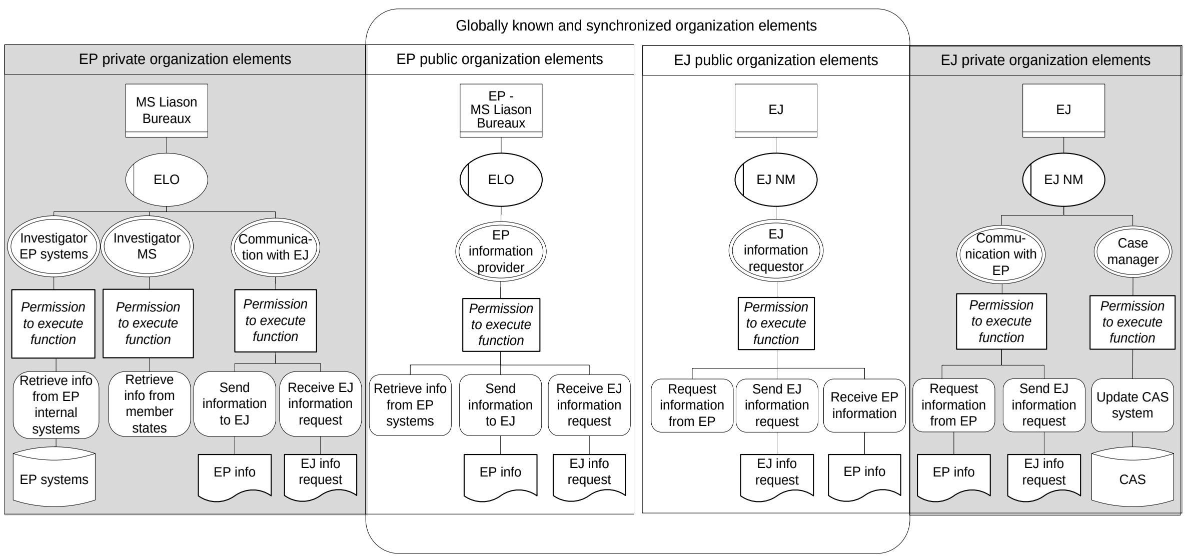


Figure 109: Private and public organization elements from EP and EJ on technical level

In order to disclose as little internal information as possible and to present collaboration partners with an expressive organizational role, EP creates a public visible proxy of this organization unit called “EP information provider”. This role has the right to execute two functions: “receive EJ information request” and “send EP information”, both functions being attached to corresponding documents. EJ creates a public visibly proxy of its internal role “EJ NM”, which has the right to send information requests to EP and to receive information coming from Europol. Thus, all elements inside the area of the globally known elements represent public elements, including functions and documents. Accordingly, the global functions comprised in it should be part of a public process and the document specification should be public to the relevant collaboration partners as well.

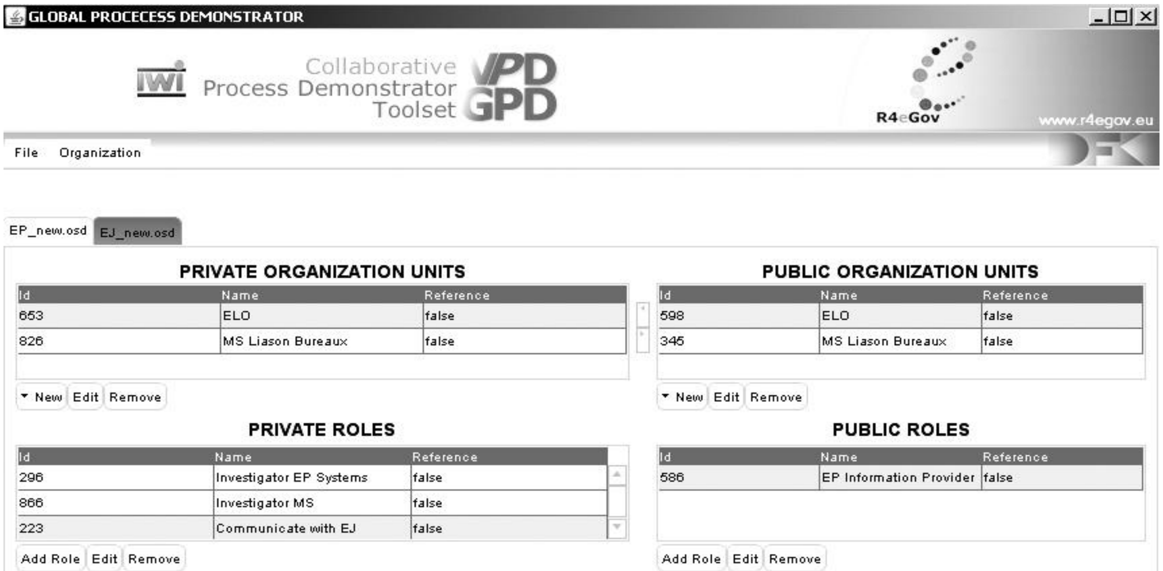


Figure 110: Development of EP's private and public organization elements with the VPD tool

EP_io.osml

```
... <position id="653">
    <name>ELO</name>
    <roleReference id="223" />
    <roleReference id="866" />
</position>
<orgUnit id="826">
    <name>MS Liason Bureaux</name>
</orgUnit>
<role id="296">
    <name>Investigator EP Systems</name>
</role>
<role id="866">
    <name>Investigator MS</name>
</role>
<role id="223">
    <name>Communicate with EJ</name>
</role> ...
```

EP.osd

```
<?xml version="1.0" encoding="UTF-8"?>
<osd>
<internalOrg file="EP_new_io.osml" />
<externalOrg file="EP_new_eo.osml" />
<mapping ioId="653" eoId="598" />
<mapping ioId="223" eoId="586" />
</osd>
```

EP_eo.osml

```
<?xml version="1.0" encoding="UTF-8"?>
<osml:osml xmlns:osml="http://www.dfki.de/
2007/osml">
<osml>
<position id="598">
<name>ELO</name>
</position>
<orgUnit id="345">
<name>MS Liason Bureaux</name>
</orgUnit>
<role id="586">
<name>EP Information Provider</
name>
</role>
</osml>
</osml:osml>
```

Figure 111: Storage format for private and public views on EP's organization elements

Steps 6 and 7 Since Figure 109 already represents complementary public organization elements, the results of Step 6 – possible modifications of public models to form complementary organization elements – is also visible in this model. The elements in the global organization model have to fulfill two characteristics: First, they must be globally understood and accepted. For example, the role "EP information provider" as well as its rights and possibilities are not only known to the owning organizations (EP), but also to the partners (EJ). One way of ensuring this, would be the usage of d-Roles, transmitted among the partners via delegation chains.[874] Second, the global organization elements must be complementary, which means that they should correlate the actors to the rights and competencies necessary to interact with each other as required in the global process. Thus, in

[874] Compare also pp. 156 of this thesis.

the example, the position ELO has the right to send the document “EP info” while his counterpart, the role “EJ NM”, has the right to receive the same document. In Step 7, the globally agreed public models are returned to the individual organizations.

Step 8 In this step, the public organization model elements forming part of the collaboration are related to internal organization elements. On the left-hand side of Figure 109, the private organization elements of EP are displayed, adjacent to the public elements of EP. While the interacting unit “ELO” is equally represented internally and externally, the private and public roles associated with it differ: The public role “EP information provider” is internally referred to as “communication with EP”. The “send” and “receive” rights associated with the public role are mirrored in the private role. The function “EP information provider” associated with the role serves to inform partner organizations about the services offered by this role; since it has no internal operational value, this function is not comprised in the private model. The roles “investigator EP systems” and “investigator MS” on the other hand are only part of the private model, since the partner does not need to know these roles, which are active only in private processes.

In this step, also the VPD tool can be used to interlink public and private organization models. Thus, the screenshot in Figure 110 illustrates how the tool is used to model and correlate the private and public organization models of EP. On the left-hand side, the public units and roles are modeled, on the right-hand side, the private elements. The highlighted area indicates which elements are correlated, e.g. the pubic role “EP information provider” corresponds to the internal role “communication with EJ” which belongs to the position “ELO”. Figure 111 shows the three different files in which the resulting models are stored. On the left-hand side, the file representing the private view is displayed, while the file on the right-hand side displays the public organizational roles. The file in the middle (“EP.osd”) correlates both files. Obviously, not only the overall files, e.g. the private with the public view, but also the individual elements are correlated. For example, the private role “communicate with EJ” is related to the public role “EP information provider” and the private position “ELO” is related to the public position “ELO”.

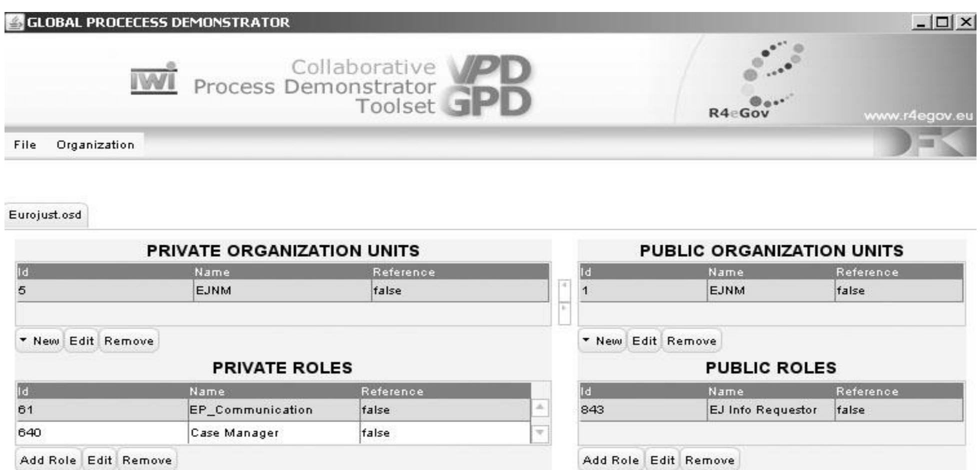


Figure 112: Development of EJ’s private and public organization elements with the VPD

The private and public organization elements of EJ are developed in the same way; the resulting models created with the VPD tool are illustrated in Figure 112. On the left-hand side, the private organization elements of EJ are shown: The position “EJ NM” having the two roles “Case Manager" and "Communication with EP". Only the latter role is public for collaboration partners, thus on the right-hand side of the screenshot the role "EJ Info Requester" is highlighted as being the public counterpart of the private role "Communication with EP". The model also indicates that the role "EJ NM" as well as its correlation with the role "EJ Info Requester" is public to collaboration partners.

5.4.4.4 Developing Organization Elements on the Execution Level (Step 9-10)

Step 9 In the previous steps, the basis was laid for the derivation of a fine-grained access control model. Each party identified and described the roles that participate in the collaboration, both from a private and a global perspective. The description of the roles comprised not only organization units and the role itself, but also the specification, which functions the role may permit on which objects. Thus, three of the four XACML *Target* elements are specified: *Action*, *Resource* and *Subject*. The fourth element, *Environment*, can be added later, but it can also be omitted since a XACML Target does not have to comprise all four elements: if an element is missing, any value is permitted for the omitted object.[875] As described above, the procedure to implement d-Roles in a collaborative scenario comprises two major steps: mapping of private roles to global roles (e.g. d-Roles) and ensuring that the d-Role roles are globally understood, for example, by establishing a corresponding delegation chain.[876] Both the specification and the correlation of private and global roles based on OSML were described above. This shows that the models of EJ's and EP's private and global roles illustrated in Figure 109 represent a good basis for a generation of d-Roles based on XACML. Consequently, the roles created in the VPD tool can be reused in the PAP tool, which creates XACML and connects local with d-Roles.

Step 10 Figure 113 shows how the private and public roles of EP developed in the VPD tool (compare Figure 110) are reused in the PAP tool to generate XACML policies of local and distributed roles. More specifically: On the right-hand side the private roles of EP are displayed, while in the middle the global roles available in the collaboration are shown. The highlighting indicates that the global role "EP info provider" is related to the private role "communicate with EJ". Thus, using the PAP tool, the global roles of EP developed on the technical level are now correlated with the private roles of EP.

Two file types can be generated with the toolset: Local XACML policies, which describe the internal role and their correlation to the global roles and distributed role policies in XACML, which describe the globally visible roles.[877] Figure 114 shows excerpts from XACML generated by the PAP tool based on the roles of EP developed in the VPD (compare Figure 113). On the left-hand side, the globally visible d-Roles are specified, i.e. those policies agreed upon between EP and EJ. On the right-hand side, the local policies of EP are shown, which describe how the global roles map to EP's private roles. Thus it is described that the private role "communicate with EJ" corresponds to the global role "EP info provider".

[875] Compare LEE & LUEDEMANN (2007), p. 86.
[876] Refer to p. 156 for the description of d-Roles and XACML.
[877] Compare LEE & LUEDEMANN (2007), p.88.

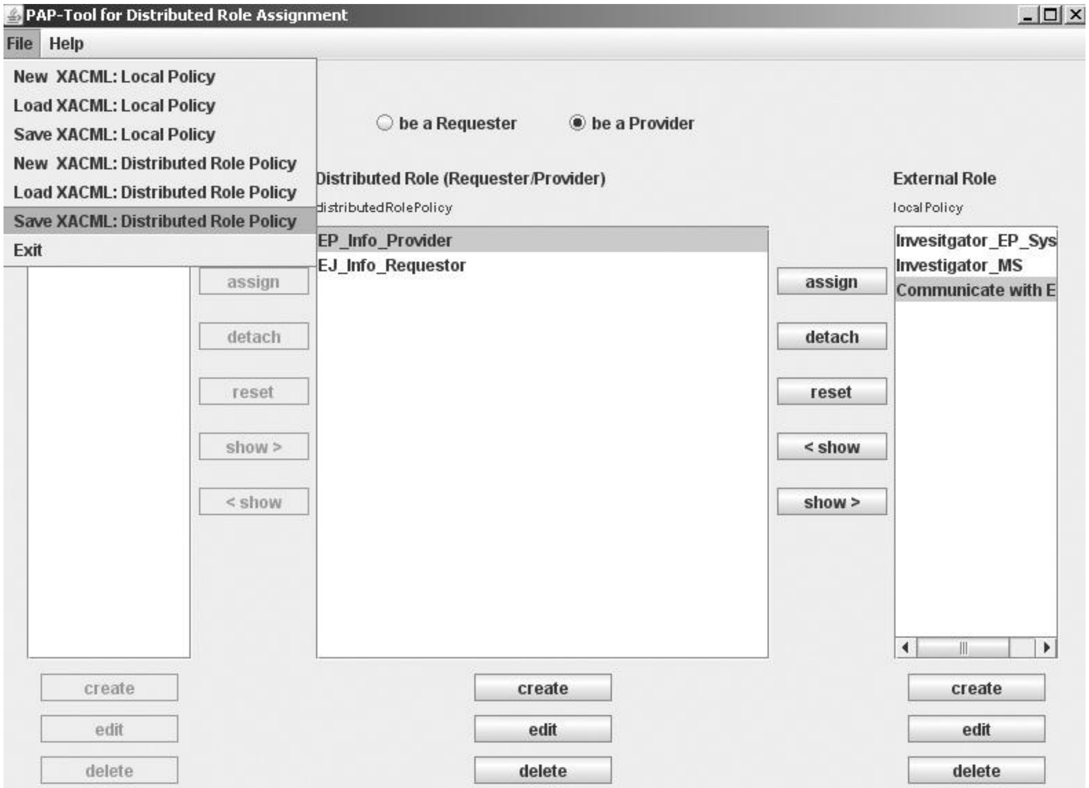


Figure 113: Using the PAP tool to correlate d-Roles with EP's internal roles

EP-EJ_dRoles.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PolicySet …  PolicySetId="dRolePolicy"> …
  <Policy … >
    <Description>Provides Information from EP to EJ</Description>
    <Target>  <Subjects>  <Subject>
          <SubjectMatch … >
            <AttributeValue ...
                >&amp;externalResource;EP_Info_Provider</AttributeValue>
          </SubjectMatch>
     </Subject> </Subjects> ...  </Target>  </Policy>
  <Policy … >
    <Description>Role from EJ entitled to request information from EP</Description>
    <Target>  <Subjects>  <Subject>
          <SubjectMatch … >
            <AttributeValue ...
                >&amp;externalResource;EJ_Info_Requestor</AttributeValue>
          </SubjectMatch>
        </Subject>  </Subjects> …  </Target>
  </Policy>
</PolicySet>
```

EP-local_roles.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PolicySet … PolicySetId="localPolicy">
…    <Policy … PolicyId="CEJ">
<Description>Interaction point for EJ</Description>
<Target> <Subjects> <Subject>
<SubjectMatch MatchId=
"urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue … >&amp;
externalRole;Communicate with EJ
</AttributeValue>
</SubjectMatch>
<SubjectMatch MatchId=
"urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeValue … >&amp;
dRole;EP_Info_Provider
</AttributeValue>
</SubjectMatch>
</Subject> </Subjects> </Target>
</Policy>
</PolicySet>
```

Figure 114: XACML representation of EP's internal roles and their correlation with d-Roles

5.4.4.5 Summary

The generic inside-out procedure model for developing elements of collaborative business processes was adapted and applied to the organization dimension. The development focused on the technical level, where the organization metamodel described in Chapter 4 and its implementation in OSML were used to correlate private, public and global organization elements. Here, also the usage of the VPD prototype was demonstrated, including the de-coupled storage of private and public or public and global organization elements. On the business level, private, public and global elements of both parties were modeled aligned to ARIS; moreover, an alternative annotation for business-level global models was shown. On the execution level, private and global XACML roles were used to transmit the results to the execution environment. In that vein, it was also demonstrated, how roles developed in the VPD tool can be used inside the PAP tool, which generates internal and distributed XACML roles. Thus, on all three levels (business, technique, and execution) private, public and global organization models were developed. In other words, all organization elements of the Business Interoperability Interfaces from EJ and EP were developed and correlated with internal organization elements.

5.4.5 Data Dimension

In this section, an inside-out procedure model for the data dimension is described and applied to the EP-EJ use case. Thus, it is shown how the partners develop the various document types necessary to implement a collaborative business process, including the correlation of private document types with externally used document types.

5.4.5.1 Procedure Model Adapted to Data Dimension

Figure 115 shows the adaptation of the generic inside-out approach (displayed in Figure 77) to the data dimension. Since current document modeling approaches concentrate on the execution level,[878] few business-level modeling approaches explicitly support documents. However, business process modeling standards like EPC, BPMN and UML Activity Diagrams allow the modeling of document types related to the functions of a process. On the technical level, some authors propose the use of specific UML profiles to model documents, which might be useful if the UML profile is tailored to specific business semantics.[879] Since no specific business semantics are prescribed in the AIOS, no such UML profile is used. Thus, on the business and technical level informal annotations are used to model the contents of business documents, while on the execution level XSD is used to specify documents.

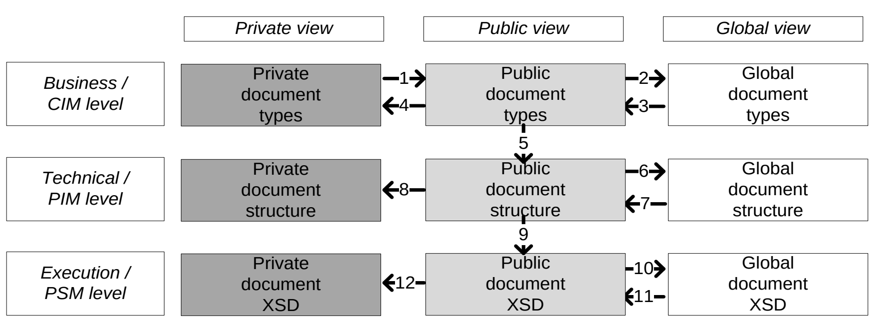


Figure 115: Procedure model adapted to data dimension

[878] Compare LIEGL (2008).
[879] Compare HUEMER & LIEGL (2007) and KRAMLER ET AL. (2006).

5.4.5.2 Defining Documents on the Business Level (Step 1-4)

On the business level, document types and their main elements are defined on a coarse, business-oriented granularity. Following the inside-out procedure displayed in Figure 115, the public document contents are derived from an internal document (Step 1). For example, the internal document "search results", which are the products of a query of internal databases, might be used as a basis to define the contents of the public document "EP info" that contains only the public fields of the private document. In order to derive a globally acceptable document specification, in Step 2 the public document is synchronized with the requirements of the partner organization. For example, EJ could clarify that apart from the fields "results" and "confidentiality level" the document "EP info" should contain the contact person in EP that is responsible for the document contents. In Step 3, the agreement on the global document specification is communicated to internal stakeholders, which might have to adapt the specification of the private document (Step 4).

Figure 116 provides an example of document types used in the EP-EJ scenario where the private, public and global document types of each party are displayed. The documents on top ("Request for info" and "EP info") are exchanged in the process illustrated in Figure 96. As a result of the previous steps, the public documents of EJ and EP are harmonized; in consequence, the global corresponds to the public document specifications. The private document specifications on the other hand differ from the public ones. For example, the "search results" are the origin of the contents of the "EP info" document and thus can be seen as the private view on the "EP info" document. Differing from the "EP info" document, it may contain private information that is not supposed to be published in the collaboration.

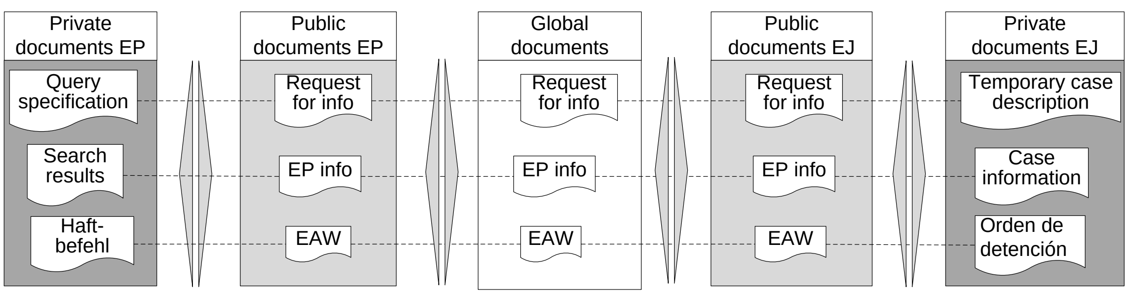


Figure 116: Correlation of private, public and global documents in the EP-EJ scenario

The EAW as an Example of a Global Document Specification On the bottom of the graphic, the example of the European Arrest Warrant (EAW) is provided. The EAW is used inside the European Union for the detention of criminal suspects and is valid throughout the states of the European Union;[880] currently it is implemented in all 27 member states.[881] To increase the usability of the document, it is available in the different languages of the member states. In the example scenario, the English version of the EAW is used as the global document specification, agreed upon between all members of the collaboration. Thus, also the public documents offered by the different public administrations correspond to the English EAW specification. These public specifications are then

880 Compare EAW (2009).
881 Compare PÉRIGNON & DAUCÉ (2007).

mapped to the national counterparts of the EAW, for example, to the Spanish version of the EAW ("orden de detención") or the German Version ("Haftbefehl"), where they can be enriched with further information needed in national prosecution.

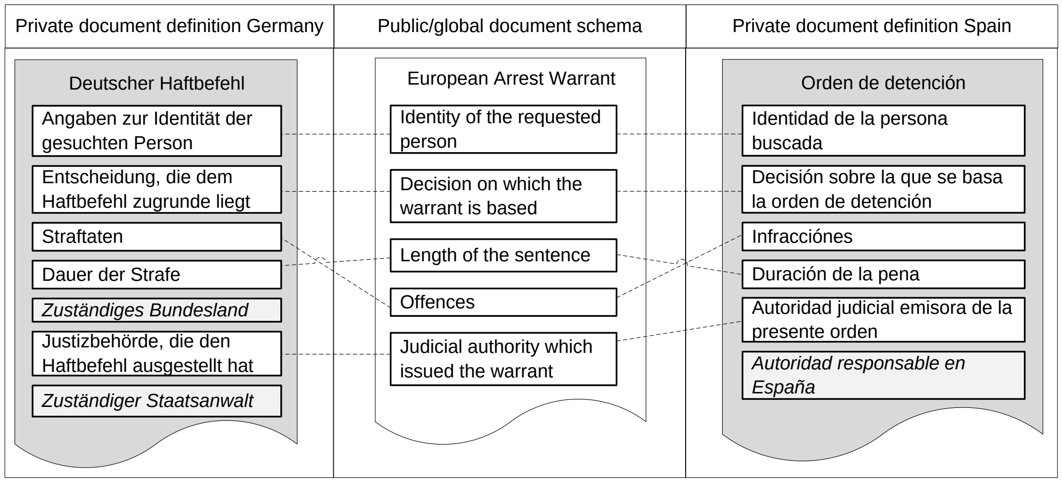


Figure 117: The EAW as an example for correlating private and global document elements

5.4.5.3 Defining Documents on the Technical Level (Step 5-8)

After the agreement on the main elements of documents used in the collaboration, the coarse-grained document specification is enriched by detailing the structure and adding technical details. Following the procedure model, first the public document is refined (Step 5). Thus, in the example the document "EP Info" in which EP communicates its internal search results would be enriched by specifying all contained fields and their format (e.g. numeric or text). In Step 6, EP and EJ would compare their technical specifications of the document, modify it if necessary and reach an agreement on a consolidated, global document specification of the "EP Info" document.

Note, that here a peculiarity of the data dimension becomes evident: In the other dimensions, for example the process dimension, the ownership of a public model in most cases is obvious. Thus, in case there are autonomous organizations, in a first step each organization will derive its own public process based on its private processes. However, the ownership of a public document inside a collaboration is harder to assess: for example, the public document "request for info" could fall under the responsibility of the service requester (EJ), who sends the document and knows best what information he requires. It could also fall under the responsibility of the service provider (EP), who has to process the document and knows best what information is needed to realize the requested functionality.

In Steps 7 and 8, the public and private models of the document contents are adapted to the global model. Figure 117 illustrates the outcomes of Steps 5 to 8, referring to the example of the European Arrest Warrant.[882] In the middle, the global document is specified that is used to exchange

[882] Note that in the example additional fields comprised in the original EAW (compare EAW, 2009) were omitted. The private, country-specific documents also differ from the original EAW, since fields were added ("zuständiges Bundesland", "zuständiger Staatsanwalt", "autoridad responsable en España") to indicate the usage of internal fields.

documents across countries, on the left and right-hand side the country specific documents are shown, which are mapped to the global document.[883]

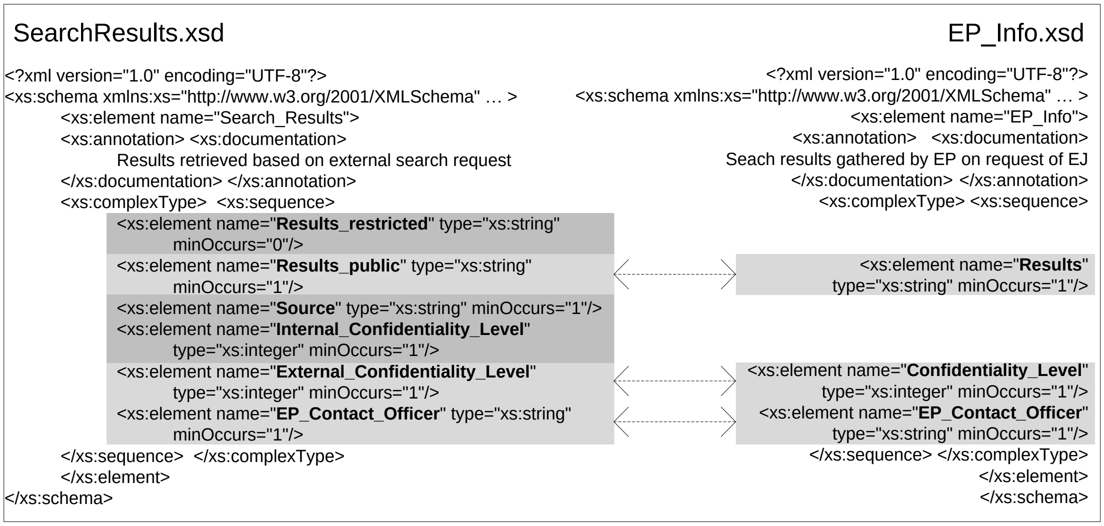


Figure 118: Private and public XML Schema Definitions of the "EP_info" document

5.4.5.4 Defining Documents on the Execution Level (Step 9-12)

Based on the technical specification of the public document, in Step 9 of the procedure model an XML schema definition of the public document is derived. Referring to the example from above, the right-hand side of Figure 118 displays the public "EP info" document. The code describes the three elements of the document, their format, their cardinality (e.g. "minOccurs="1") in the form of XSD. In Step 10, is it ensured that XML Schema Definition from EP corresponds to the definition of EJ; the result is a consolidated XSD specification of the "EP info" document. In Step 11, the public document specification is adapted to the global XSD specification and in Step 12, the public document is correlated to the private XSD specification. Figure 118 illustrates the correlation between the elements of the private and the public document: while the element displayed in dark gray on the right-hand side remain private, the elements displayed in light gray have a public counterpart. Similar to the correlation mechanisms in the process or the organization dimension, an additional file has to be generated to correlate the private with the public XSD specification of the documents. The specification of such a correlation file is not in the scope of this work; however, following the metamodel in Chapter 4,[884] this file should enable a mapping of the overall documents (e.g. stating that "EP_Info.xsd" is the public version of "SearchResults.xsd") as well as their elements (e.g. stating that "results public" corresponds to the public element "results").[885]

[883] Though they should be specified, for illustrative reasons further technical specifications of the individual fields are not shown in the figure.

[884] Compare Figure 68, p. 173.

[885] This mapping could be implemented for example by using XPath ("XML Path Language") to address the individual elements and XSLT to describe the relation between the elements; compare W3C (2007) and W3C (1999).

5.4.5.5 Summary

In this section, it was demonstrated how private, public and global documents can be specified on different levels of technical granularity. Thus, for the business level, private, public and global views on document types were developed, where the private document types contained additional information not part of the public documents and the global document types corresponded to the public document types of EJ and EP. On the technical level, the structure of the documents was modeled. Here, also individual document elements of the private and public documents were correlated to each other. On the execution level, the corresponding document structures were described as XSD. In order to enable a systematic transfer between public and private document contents, it was illustrated how elements of the private and public XSD specifications can be mapped to each other. Different from the process and the organization dimension, the VPD/GPD prototypes only implicitly support the development of the AIOS elements in the data dimension.[886] Nevertheless, the application of the procedure model led to the specification of private, public and global documents on all three vertical levels (business, technique, and execution); in other words, the AIOS data dimension for the EJ-EP collaboration was completely specified.

5.5 Integrated Results

5.5.1 Resulting Business Interoperability Interfaces

As result of the procedure model described in the previous section, EJ and EP each have specified complete Business Interoperability Interfaces covering the business, technical and execution level in four enterprise dimensions (process, function, data and organization). Figure 119 illustrates the contents developed in the example described above. On the left-hand side, the BII-repository of EP is shown; it contains all models that should be visible to the collaboration partner, i.e. to EJ. To the left of the repository, the private counterparts of the public elements as well as their correlations are illustrated. On the right-hand side, the same elements are shown for EJ. Thus, apart from a business-level description of the CBP, the repositories contain all the information necessary to enact a collaborative business process between both parties.

In this overview, also the differences between the different implementations of the collaborative views in each enterprise dimensions are visible. For example, the different types of correlation files used among the views: In the business level of the *process dimension*, distinct files types are used to correlate public and private processes, as well as to correlate public processes into a global process. On the execution level on the other hand, no separate correlation file is needed, since the BPEL Abstract Processes reference each other and correlation tokens ensure the correct mapping of messages during run time.

[886] More specifically, in the tools, document types (like for example “RfQ”) can be attached to functions of processes; these documents can then be opened to attach an XSD to the document type. Figure 97 for example displays a screenshot of the GPD, where a global process is modeled including XML documents. Thus, private, public and global documents can be displayed; however, the tools currently do not support the correlation of individual model elements, like the mapping of public document fields to private document fields or the mapping of public to global documents.

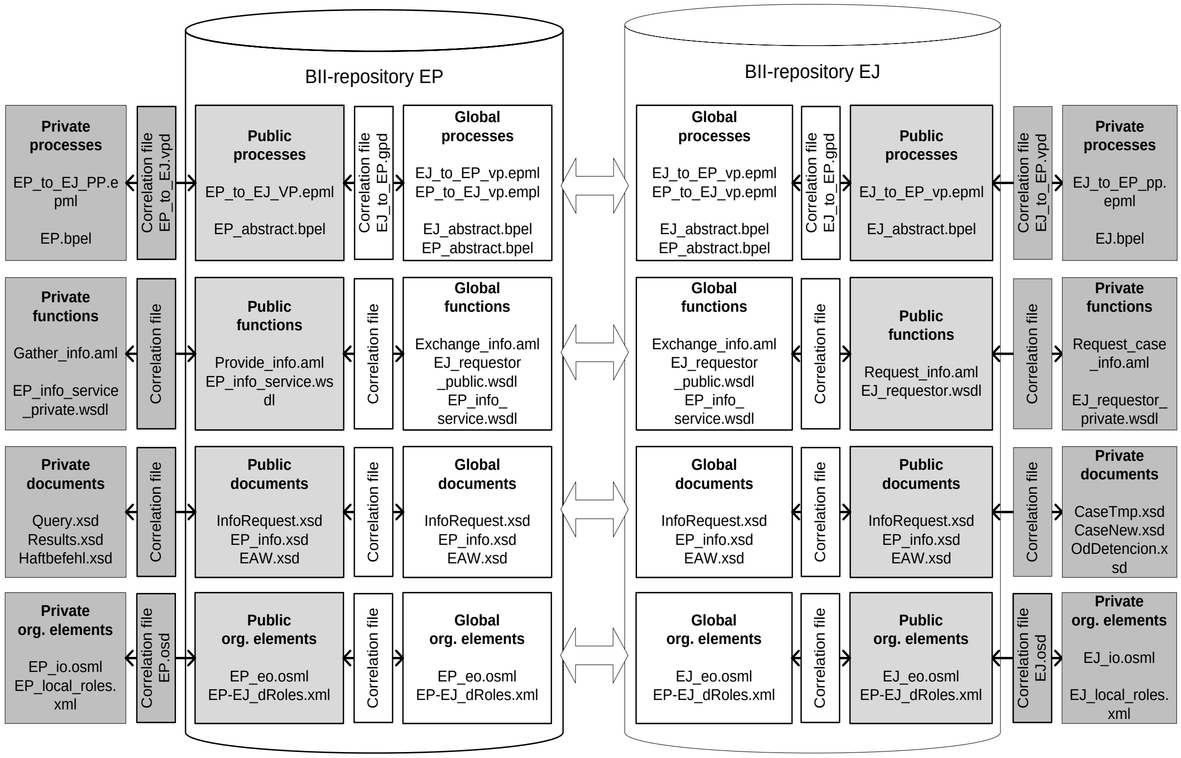


Figure 119: BII-repository contents resulting from the application of the procedure model

On the business level of the *function dimension*, the correlation files map elements of private and public functions. On the execution level, no explicit mapping is used, since the public operations of a service are part of the private service and thus can be accessed directly.[887] In the example, the public functions were identical to parts of the global function model, thus here no correlation among private and public models was used.[888] However, to avoid misunderstandings about service characteristics implemented in the collaboration, in general it seems recommendable to map globally known services explicitly to the public implementations offered by specific partners. On the execution level, the global model again was constituted by the sum of the comprised interfaces, e.g. the complementary WSDL interfaces of EJ and EP. Thus, here no correlation file was needed either.

While in the function dimension, global models comprised information beyond the public models (e.g. the function "exchange_info.aml"), in the *data dimension* global documents directly mirror the public documents. Thus, the three document types used in the collaboration are specified in the global view, as well as in the public views of EP and EJ. Since the structure of the internal documents differs from the public documents related to them, a correlation file between private and public document specifications has to map the private and public elements to each other.

[887] This corresponds to our usage of public processes, which only abstract from private processes but do not display modifications of private elements. Thus, both in the process and function view the collaboration partner directly addresses the public elements of the private models, and in consequence, no mapping between the public and the private view is necessary. Compare also the corresponding illustration in Figure 104.

[888] While the services are described in WSDL, the corresponding function trees can be described with the ARIS toolset, which stores them in the ARIS Markup Language (AML).

Similar to the data dimension, the global models in the *organization dimension* are identical to the public models. Hence, on the business level in the example no distinct correlation file was used to map public and global models to each other. On the business level, private models were specified in OSML and mapped (via the correlation files with the ending ".osd") to their public counterparts. On the execution level, public and global roles are described as d-Roles. Here, no distinct correlation file was used, since the mapping of private to public roles is comprised in the file that defines the private roles (e.g. "EP_local_roles.xml").

5.5.2 On the Integration of the Enterprise Dimensions

Though they represent self-sustained architecture views, the different AIOS enterprise dimensions are closely related and offer connection points to each other. To reduce the complexity of the overall AIOS development, in the previous sections, the development of each dimension was described separately.

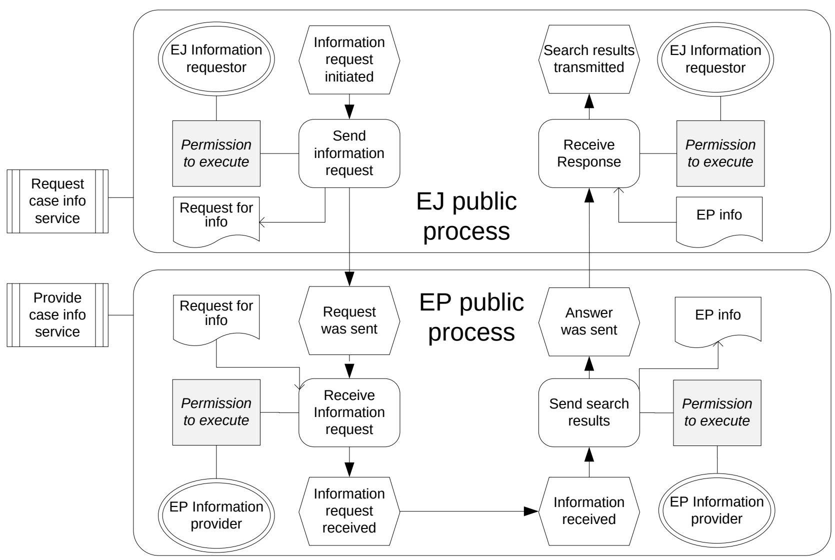


Figure 120: A global process as assembly of global data, function and organization models

While the specification and integration of execution level standards for the four enterprise dimensions is not part of this thesis,[889] for the business and technical level, the relationships between the AIOS enterprise dimensions as well as the relationships between their elements were described in Chapter 4.[890] The application of the procedure models confirmed the intertwined character of the dimensions: In the organization dimension, for example, the capabilities of organizational roles are explained by relating them to functions. In the function dimension, the input and output of a func-

[889] However, as shown in the application of the procedure model, execution level standards for the process dimension and the functional dimension (BPEL and WSDL) are compatible with each other; the XSD used in the data dimension can be used in BPEL and WSDL as well. Nevertheless, the realization of the complete AIOS metamodels on the execution level – covering all elements of the individual enterprise dimensions and supporting private, public and global views on them, as well as the integration of the resulting standards – remains a future research topic. Compare also the point regarding the "implementation of AIOS mechanisms in SOA standards" in Chapter 6, p. 245.

[890] Compare also Figure 72, p. 177.

tion is specified resorting to elements of the data dimension. The process dimension connects functions into sequences. Note that this implies that not only models from the process dimension, but also models from the organization and function dimension could be used to integrate elements of adjacent enterprise dimensions.

However, since the overall aim of defining the various dimensions is the enactment of (cross-organizational) business processes, the process dimension plays a distinguished role in the integration of enterprise dimensions. Figure 120 displays a global process as an assembly of the global elements defined in the other enterprise dimensions: Globally known organizational roles responsible for the execution of related functions, globally agreed document specifications, as well as the sequence of the globally understood business functions. The model also indicates that the global process represents a combination of two public processes, each one being implemented as a web service. As described above, the integration of different enterprise dimensions and their collaborative views is also supported by the VPD prototype.[891]

5.5.3 On the Effort to Develop the Models

The application of the procedure models to the EP-EJ use case illustrated that the development of all models comprised in the AIOS can result in a significant effort, since many different model types have be to created and correlated. However, depending on the context, individual elements of the AIOS can be omitted, and not all steps of the procedure model have to be executed. For example, collaboration partners could omit the specification of the data dimension, and describe exchanged documents only implicitly in the process dimension. Or, in case the collaborating organizations have few explicitly described roles, the partners might even omit the explicit definition of the organization dimension, and only refer to the interacting services. Thus, instead of a binding description that must be fulfilled completely, the AIOS can be also be used as a reference system that describes the elements (and their relationships) potentially needed in specific collaborations.

Nevertheless, to foster model reuse and to be prepared for a wide range of collaborations, it is recommendable to specify the different CBP dimensions explicitly, completely and separately from each other. The AIOS is based on the assumption of large collaborating organizations, which are able to make a long-term investment in sustainable, interoperable information systems. In this vein, the effort for developing and correlating the individual models comprised in the AIOS should be seen as the cost for fulfilling (ambitious) requirements, including:

- Costs for interoperability: Organizations want to maintain their legacy systems and the privacy of internal processes, yet they want to interact with collaboration partners that use different types of information systems. If this objective should be realized, organizations have to carry the costs for implementing interoperability mechanisms, e.g. interfaces that connect their internal systems with adjacent systems.
- Costs for compliancy and sustainability: Especially if collaborations are restricted by laws and regulations, stakeholders require mechanisms to ensure that run time solutions are compliant with business-level models, like for example model-driven software de-

[891] Compare Figure 89, where the usage of private and public organization models for the derivation of public processes is described.

velopment.[892] A model-driven development also increases the sustainability of information systems, since existing models can more easily be analyzed, modified or re-built.[893] To achieve this, organizations have to carry the costs for developing and correlating models on different levels of technical granularity.

- Costs for secure but automated collaborations: In the EP-EJ scenario for example, high security requirements were present as well as the desire to automate cross-organizational processes. To enable this, rights and roles have to be usable across organizational boundaries; in other words, if organizations want to support process-automation in security sensitive collaborations, they have to carry the costs for correlating internal roles and rights to globally known rights and roles.
- Costs for automatic service discovery: Though the run time discovery of services is not tackled explicitly in this thesis, the Business Interoperability Interface can also be seen as a multi-dimensional service description, supporting the dynamic discovery of services. If a service description should enable an automated discovery, a one-dimensional description of a service (e.g. covering only the enterprise dimension) is insufficient. Obviously, the chances of finding the right service are increased, if the service requestor as well as the provider can describe a holistic picture of the service.[894] An automated service discovery is also enhanced if services are described on different levels of technical granularity, e.g. both from a business and a technical perspective. Thus, if an automated discovery of services is wanted, organizations have to carry the costs for describing services comprehensively.

5.5.4 Revisiting the Goals of the AIOS

In Chapter 3, the goals of the AIOS were described. After the detailed description of static and dynamic parts of the AIOS, its application to a use case and its implementation in a prototype, the fulfillment of these goals can be judged as follows:

- Construction of interoperable information systems: The different levels of technical granularity in the AIOS already indicate its suitability for a systematic software development. The suitability of the AIOS to enable stepwise software-development involving different organizations was further shown in various procedure models. The ability of these procedure models to construct interoperable information systems was illustrated by its application to the EJ-EP use case, as well as by modeling prototypes that implement parts of the AIOS. It was further described that the AIOS enables the construction

---

[892] Apart from ensuring compliancy in the development process, the correlation of design models with execution models is also a prerequisite for the monitoring and controlling of processes, since here execution level events are translated into business events.

[893] Note, that due to the complementary character of the different enterprise dimensions, it is also possible to reuse and extend models from the different enterprise dimensions independently from each other; for example, roles or documents defined in the context of a business process can be reused in other collaborative business processes as well.

[894] For example, not only the function view, but also the process view should be covered in a service description. Correspondingly, the service description would be improved if the organization view – e.g. by describing organizational roles involved in the services – and the data view were part of the service description, for example by referring to globally known document types that are consumed or produced by the service. Compare also the corresponding explanations on p. 75.

of interoperable information systems in case there is a predefined-group of collaboration partners, as well as in case there is one organization, which describes a collaborative business process with potential partners.

- Description of interoperable information systems: The Business Interoperability Interface comprised in the AIOS describes the boundaries of an information system in all dimensions usually needed for the enactment of collaborative business processes. It also describes, how this interface can be related to the interfaces of partner organizations. The description of the Business Interoperability Interfaces of EJ and EP and the different file types created in the VPD/GPD prototype confirmed this judgment: The elements comprised in each BII enabled the organizations to interoperate with each other and at the same time to establish a connection of global process elements to internal elements.
- Comprehensive approach: The assumption that the enterprise dimensions comprised in the AIOS cover the elements usually comprised in collaborative business processes was confirmed by the literature review and in the application of the use case. The application of the procedure model also showed that the MDD-levels provide the levels needed in the construction of CBP. Likewise, it was demonstrated that the private, public and global views provide the collaborative perspectives needed in the development of interoperable systems.
- Comprehensible approach: Since the AIOS is based on three well-known, orthogonal axes (enterprise dimensions, MDD-levels, and collaborative views), a good comprehensibility of the overall approach can be assumed.
- Relate to modern implementation technologies: The AIOS incorporates essential SOA principles, most importantly, interface-orientation. It was shown, how the business-level models comprised in the AIOS can be enacted with SOA-related standards. Moreover, the principle of publishing and discovering services based on a (multi-dimensional) description of their interfaces is supported by the BII-repository.
- Preserve internal systems: It was demonstrated that the collaborative views are suitable for describing a system without the need to disclose internal information. Moreover, different public views can be created for different partners, while the corresponding private models can remain unchanged. The prototype implemented mechanisms for the separation and correlation of public and private models, including the storage of distinct files to correlate public/external and private/internal views on systems.
- Interface-orientation: The Business Interoperability Interface – describing the behavioral possibilities and constraints of an organization to collaboration partners – represents the core of the AIOS. The model types comprised in it were described and demonstrated in the use case; it was also described how the Business Interoperability Interface can be implemented in the form of the BII-repository.
- Decentral enactment of processes: In general, the separation of private, public and global models supports decentral process enactment, since public models describe what each organization has to enact in a collaborative business process. Due to the usage of BII-repositories, each party can manage the public models it contributes to the collaboration. Supporting the choreography concept, each party executes their part of a collaboration, thus no central execution engine is needed. The development of the corres-

ponding models, for example public processes in EPC and in BPEL, was demonstrated in the use case and supported by the VPD/GPD prototype.

- Loose coupling: Due to the interface-orientation and the explicit separation of private and public models, the internal systems of collaboration partners are not intertwined and the interoperation is clearly restricted to the public elements described in the Business Interoperability Interface of each organization. Moreover, since processes are executed decentrally, each partner can control his involvement in a CBP and, if wanted, terminate their involvement easily.

6 Conclusions

This chapter concludes the thesis. Therefore, in the first section, the main research results are reviewed based on the criteria of novelty, validity and usefulness. The second section provides an outlook on future research.

6.1 Review of Results

To assess the quality of design science research, a variety of criteria can be found in literature. HEVNER ET AL., for example, described seven guidelines that design-science research should fulfill: Design as an artifact, problem relevance, design evaluation, research contributions, research rigor, design as a search process and communication of research.[895] In a similar vein, FETTKE stated, that the quality of an information systems research endeavor can be judged on the criteria abstraction, innovation, non-triviality, systematic approach and evaluation;[896] comparable criteria are described by FRANK.[897] In the following, these criteria will be concentrated into the three categories of *novelty, validity and usefulness* of research results, where validity and novelty are pre-conditions for research usefulness. Table 11 illustrates, how the criteria described by HEVNER ET AL., FRANK and FETTKE can be mapped to these three categories.[898]

Novelty	Research contribution (HEVNER ET AL.), innovation (FETTKE)
Validity	Research rigor, design as a search process, creation of design artifact, communication of research (HEVNER ET AL.), design evaluation (HEVNER ET AL., FETTKE), systematic approach (FETTKE), transparency, skepticism, justification (FRANK)
Usefulness	Problem relevance (HEVNER ET AL.), originality/superiority (FRANK), abstraction/genericity (FRANK, FETTKE), non-triviality (FETTKE)

Table 11: Overview of criteria to judge design science research results

Thus, the research results obtained in this thesis in the following will be reviewed based on the categories novelty, validity and usefulness.

6.1.1 Novelty of Results

As described in Chapter 1,[899] the five main research objectives of this thesis are:

- The development of the overall interoperability architecture,
- a Business Interoperability Interface as core of the architecture,

[895] Compare HEVNER ET AL. (2004), pp. 82.
[896] Compare FETTKE (2008), p. 34.
[897] Compare FRANK (2006), pp. 33.
[898] Compare HEVNER ET AL. (2004), FRANK (2006), and FETTKE (2008).
[899] Compare pp. 5.

- the detailed specification of the architectural dimensions,
- a procedure model for the stepwise development of the static elements of the AIOS, and
- the development of tools that illustrate and validate the modeling of collaborative business processes in the AIOS.

The novelty of the results corresponding to these objectives was verified by several state-of-the-art reviews; more specifically: To ensure the novelty of the overall solution, in Chapter 3, approaches comparable to the AIOS were described and related to it.[900] To ensure the novelty of each AIOS dimension, in Chapter 4, for each enterprise dimension a state-of-the-art review was executed. Since they are based on the AIOS, the novelty of the procedure model and the developed prototypes is implied. However, in Chapter 5 the relationship of existing procedure models and tools to the solutions developed in this thesis was briefly described.[901] Thus, the research contributions of this thesis that extend the state-of-the-art can be summarized as follows:

Architecture of Interoperable Information Systems Previously, comprehensive systems for describing information systems and enacting business processes focused on intra-organizational scenarios. For inter-organizational scenarios, only partial solutions existed, which did not support all of the needed enterprise dimensions, lacked support for SOA or lacked a connection with the business-level. Thus, seen from an *enterprise modeling perspective*, the novelty of the AIOS lies in its extension of current enterprise modeling approaches with SOA concepts (more specifically: private, public and global views on enterprise models), improving the usability of enterprise models in collaborative business. From a *SOA perspective*, the AIOS complements existing concepts for service-based processes by tackling all dimensions of a business process and by providing a connection between execution-level and business-level concepts. And, different from existing interoperability frameworks, the AIOS explicitly covers both the *description* and the *development* of interoperable information systems.

Business Interoperability Interface Though with SOA new concepts for describing the boundaries of systems were created, existing concepts for information system interfaces generally focus on technical aspects, support only view enterprise dimensions or do not support the connection of internal and external processes. For example, frameworks from the area of collaborative business and (eGovernment) interoperability lack support for process-oriented interfaces, and the interface concepts created in the context of SOA lack support of business-level elements. Following the recommendation of the European Commission to develop interfaces that also display the interaction capability of an organization on the business level,[902] the BII developed in this thesis covers not only technical and execution-level, but also business-level concepts. Moreover, it connects internal with external business processes and provides genuine views on all enterprise dimensions needed in the enactment of collaborative business processes.

900 Figure 30 on p. 91 provides an overview of comparable approaches and their relationship to the AIOS.

901 On pp. 181, related procedure models were described, while on p. 199 tools comparable to the VPD/GPD tools were described.

902 Compare European Commission (2004), p. 18.

Collaborative Views on all Enterprise Dimensions Previously, in the context of collaborative business, private, public and global views were used only for the process dimension. In this thesis, a concept was developed to use such views for all enterprise dimensions. More specifically, resorting to work from related research areas like IT security, databases and Object-oriented programming, a concept was developed to create and correlate collaborative views on all dimensions relevant for collaborative business processes: organization, data, process and function. For each enterprise dimension, detailed metamodels were specified that describe the elements of the private, public and global views, as well as the integration of the enterprise dimensions. The metamodels also served as a basis for the development of corresponding storage formats, enabling the separated yet correlated storage of private, public and global views on collaborative business processes. Thus, for the process dimension, EPML was extended to enable the description and correlation of private, public and global process models; for the organization dimension, the Organization Structure Markup Language was specified that enables one to store and correlate private, public and global views on organization elements.

Model-Driven Development of Interoperable Systems Previously, few procedure models existed, that supported a model-driven development starting on a business requirements level and leading to the definition of service choreographies. Existing approaches focused on the process dimension and usually lacked detail as well as a proof of concept. Going beyond these approaches, different alternative development paths were described and a procedure model was developed that covered the business, technique and execution level, private, public and global views on information systems as well as the four enterprise dimensions mentioned above.

Business-Level Modeling Tools that Support Collaborative Views Existing tools in the context of collaborative views focus on the derivation of public processes from private processes on the workflow level, tackling only the control flow. The different dimensions of business-level processes are neglected, as well as the forming of global processes from public processes. In this thesis, a tool was described that can derive public from private processes and provides the above-mentioned storage formats. In a second step, the tool enables the correlation of public processes to global process models. Apart from the process dimension, the tool also supports the organization dimension by providing private and public views on organization structures. Moreover, it allows connecting the different enterprise dimensions, for example, by integrating private and public views on organization elements in private and public process models. Going beyond the modeling of collaborative business processes, a concept for a tool suite was provided that describes the run time functionalities of the AIOS, including the concept for the BII-repository, where organizations can publish the contents of the Business Interoperability Interface.

6.1.2 Validity of Results

As illustrated in Table 11, the validity of results from design science is usually assessed by criteria like research rigor, design as a search process, design evaluation, communication of research, systematic approach, transparency and skepticism. Characteristics of this thesis that imply the fulfillment of such criteria are:

Explicit Description of Research Method The research method was described explicitly in Chapter 1 and – in case refinements were necessary – in other chapters as well (compare for example the illustration on p. 71). As described in Chapter 1, the research method follows the design-science approach and deductive reasoning. The individual steps of the research method were based on a method for system development described by NUNAMAKER, CHEN & PURDIN.[903] Thus, new artifacts were created based on a requirements definition and existing theories. Subsequently, the validity of the created artifacts was demonstrated with a use case and a prototype.

Design as a Search Process/Review of Alternatives The AIOS was developed in various iterations: In Chapter 1, a coarse-grained state-of-the-art overview and a corresponding requirements definition was presented. In Chapter 3, the requirements described in Chapter 1 were refined and validated in a detailed state-of-the-art review. First, frameworks and architectures comparable to the AIOS were reviewed. After another literature review for each AIOS axis, the units of each axis and the overall structure of the architecture were consolidated. In Chapter 4, the details of the AIOS were specified by defining a metamodel for each enterprise dimension of the architecture. Here again, for each axis a comprehensive state-of-the-art review was executed, and from the existing solutions those parts most suitable for the AIOS were selected and extended. Also in Chapter 5, different alternative development paths possible in the model-driven enactment of collaborative business processes were discussed before a reference development path was chosen.

Illustrative Writing A potential drawback of a deductive research approach is that the chain of arguments can get too complex to be understood (with reasonable effort) and the resulting lack of transparency and traceability inhibits a judgment of the research. To enhance the comprehensibility and to validate the textual description of complex concepts, many semi-formal models were used to illustrate the relationships between the AIOS elements as well as the overall research process.

Communication of Research As described in Chapter 1, this thesis was developed in the context of different European research projects that tackle (business) interoperability and SOA. Thus, the first outlines and foundational concepts of the AIOS were discussed at various conferences.[904] Apart from the research community, feedback on elements of the AIOS could also be gathered from industrial stakeholders in research projects as well as from project supervisors of the European Commission.[905]

Coherent Design Artifact As illustrated in Figure 4,[906] the constituting elements of the AIOS (overall structure, individual dimensions and procedure model) are closely related and mutually validate each other, thereby ensuring the coherency of the overall design artifact. Thus, the AIOS structure was validated in the fine-grained specification of the architectural dimen-

---

903 NUNAMAKER, CHEN & PURDIN (1999).

904 For example during the presentation of the work from ZIEMANN, KAHL & MATHEIS (2007), ZIEMANN & MENDLING (2005), ZIEMANN, MATHEIS & FREIHEIT (2007) and ZIEMANN, MATHEIS & WERTH (2008).

905 Such feedback was gathered for example on the second year review of the R4eGov project, where the author – in the role of the lead of the “Model-driven Interoperability” work package – presented a first version of the AIOS.

906 Compare p. 12.

sions in Chapter 4, where the feasibility of private, public and global views on each enterprise dimensions was confirmed. The applicability of the AIOS metamodels for a model-driven development and description of collaborative business processes, on the other hand, was validated by the procedure model in Chapter 5. Providing a basis for a coherent and unambiguous usage of terms during the development of the AIOS, in Chapter 2 essential concepts were defined. Moreover, fostering the coherency inside individual AIOS elements, at the end of the development Chapters (3, 4 and 5), the integration of the elements created in the chapters was described.

Evaluation of Design Artifact The fulfillment of the research objectives provided in Chapter 1 was described in the previous section, while the fulfillment of the more specific requirements on the AIOS (provided in Chapter 3) was discussed at the end of Chapter 5.[907] The overall objectives of the AIOS – description and enactment of collaborative business processes – were evaluated by the creation of a prototype and its application to a use case. Thus, proving the suitability of the AIOS metamodels for the modeling of collaborative business processes, the VPD/GPD prototype was developed. Additionally, a tool suite was described that shows how the processes developed with the AIOS can be executed. To prove its applicability to real-life scenarios, the procedure model and the prototype were applied to a scenario from the R4eGov project. Moreover, the incorporation of technical concepts in the development process of the AIOS, most importantly in Chapter 4, can be seen as a validation as well, since here it was already ensured that the business-level concepts (e.g. for private and public views on processes and organization elements) can be mapped to execution level concepts.

6.1.3 Usefulness of Results

The relevance of the research question tackled by the AIOS was laid out in Chapters 1 and 3. Thus, in Chapter 1 the need for a comprehensive, business-driven development of interoperable information systems was stated. It was described that this need was confirmed in recent research projects as well as from the practice side, where the amount of governmental investments in interoperability as well as statements from industrial stakeholders indicated the need for a comprehensive method for enacting collaborative business processes.[908] A refined state-of-the art analysis in Chapter 3 confirmed that currently no integrated solution for such a development exists.[909] In other words, the relevance of the AIOS is indicated by the fact that it meets the demand for such an integrated solution and closes this gap by combining the strengths of SOA, Business Interoperability and enterprise modeling approaches into one system to comprehensively describe and systematically enact collaborative business processes.[910]

[907] Compare pp. 236.
[908] See also pp. 1 and pp. 72.
[909] Compare pp. 77.
[910] To make this gap more tangible: If a large organization asked, “How can we describe our information systems comprehensively, covering conceptual and technical levels, business processes, organization units as well as business data” – for intra-organizational scenarios, the answer would be, “Use an enterprise architecture, for example ARIS”. However, for inter-organizational scenarios, given the deficiencies described above, such a comprehensive solution was missing.

Applicability/Genericity Due to the deductive research method pursued in its development and its foundation on widespread and generic concepts, it can be assumed that the AIOS is applicable in a broad area of organizations. The three widespread and well-known axes the AIOS is based on increase the probability that the AIOS can be used in organizations by extending their existing solutions towards the AIOS. The enterprise axis, for example, covers the enterprise dimensions of ARIS, which can be seen as one of the process modeling frameworks with the highest industrial acceptance. Moreover, the process, function, data and organization view are confirmed as essential dimensions by most business process modeling approaches. The MDD-axis, representing levels of technical granularity, is also well known since its units (CIM, PIM, PSM) are supported by the OMG, as well as by enterprise modeling approaches like ARIS. The units of the collaborative view axis on the other hand, were made popular by SOA concepts, where orchestration, choreography interfaces and choreographies are used to implement private, public and global processes.

For similar reasons it can be assumed that a broad scientific audience can relate to the AIOS: From an enterprise modeling perspective, it can be used to extend enterprise modeling frameworks for collaborative business and SOA. The prototypes developed in this thesis already indicate how the private processes modeled in existing toolsets– like for example ARIS – could be correlated to public and global models. From a Business Interoperability perspective, the AIOS can serve as a reference for interoperability frameworks and eCommerce suites, e.g. regarding the coverage of public, private and global views on the different enterprise dimensions. From a SOA perspective, the AIOS can serve as a reference for existing SOA solutions, like for example standards for automating business processes or attempts to create "SOA metamodels"; in this vein, the Business Interoperability Interface can also be understood as a means for a comprehensive services description that supports the (automated) discovery of services.

6.2 Future Research

The AIOS developed in this thesis focuses on the design of interoperable information systems, more specifically, on the description of collaborative business processes on both conceptual and technical levels. A future research topic that falls in the design phase of the AIOS is the modeling and enactment of *distributed transactions*, leading to the description of transaction and compensation spheres in the public processes of collaboration partners.

As illustrated in Figure 74 (p. 181), these design activities are surrounded by a number of complementary lifecycle phases, which should be tackled by future research. For example, mechanisms should be developed, which enable an *automated verification* making sure that public models comprised in global models are complementary to each other – for the process dimension as well as for other dimensions, e.g. the organization dimension. Further, the *run time functionalities* of the AIOS laid out in the thesis need to be detailed; this comprises the development of the BII-repository as well as other parts of the execution environment. In addition, mechanisms to analyze collaborative business processes, e.g. *monitoring and controlling* functionalities complementary to the AIOS, should be developed. Apart from these, the following AIOS-related topics are left for future research:

Extension of the Modeling Prototypes Currently, the VPD/GPD prototypes focus on the process and the organization dimension while the data and function dimension are only supported implicitly. Similar to the organization dimension, a genuine view for collaborative views on document structures and functions should be created. This would increase the possibility to include cross-links in the models (e.g. by not only referencing organizational but also data and function models in the process dimension) and increase the compatibility of the prototypes with commercial tools like ARIS.

Model Transformations and Mappings This thesis focused on the horizontal transformation and mapping of models, e.g. the correlation of private, public and global models. Vertical model transformations, like EPC-to-BPEL transformations, were briefly described during the application of the AIOS to the use case. However, to enable an automated top-down development, the vertical transformations in the different AIOS dimensions should be described in more detail. Apart from ensuring the compliancy of execution-level with business-level models, the resulting correlation should also support the controlling of collaborative business processes.

Model Management The description and automation of collaborative business processes requires many different model types as well as model instances. Future research should tackle the management of AIOS models. On the one hand, this comprises *intra-organizational model management*. Here, mechanisms to manage the relationships between private and public model have to be created. For example, synchronization rules have to be implemented, ensuring that if a public model is changed, the private model is changed accordingly, or at least the owner of the private model is notified. On the other hand, an *inter-organizational model management* should support the synchronization of public and global models among the collaborating organizations.

Implementation of AIOS Mechanisms in SOA Standards In this thesis, the integration of the enterprise dimensions focused on the conceptual levels, since on the execution level, standards like BPEL, WSDL, XSD and XACML only allow a partial integration with each other and an extension of these standards was out of scope. Future research should investigate in how far existing SOA standards should be modified in order display business processes in a comprehensive and integrated manner and be compatible with business-level modeling requirements. Especially in the organization dimension, a gap between business and execution level standards can be observed; here, the AIOS metamodel could be used for the creation of an XML-based standard that closes this gap, possibly building on the Organization Structure Markup Language (OSML) described above.[911] A related research topic is the integration of organization elements in the BPEL standard.

Refinement of Output Dimension Though a generic metamodel for the output dimension was described,[912] this thesis concentrated on that part of the output dimension that coincides with the data dimension: documents, which are produced in business functions and are exchanged between the collaboration partners. The main reason for the limited coverage of the output dimension is that the output dimension seems to be most useful in the requirements definition phase, while the other dimensions (process, function, organization and data) are essential for all levels of collaborative business process automation.[913] However, especially in eCommerce

[911] Compare pp. 204.
[912] Compare Figure 70, p. 175.
[913] Compare also pp. 99 and pp. 175.

scenarios, the creation of private, public and global product models, for example, in the form of product trees as proposed in the ArKoS project,[914] certainly is useful. Thus, future research should demonstrate to what extend elements of the output dimension not tackled in this thesis, i.e. output not produced by information services, can be supported in the AIOS.

Standardized References for Global Models This thesis followed a decentral approach and no centrally defined global models (as described for example by standards like UN/EDIFACT or RosettaNet) were used. Instead, the document types, services, processes and organizational roles used in the collaboration were defined by the collaboration partners. Since the resulting global models/ontologies can be reused in later collaborations, this procedure is equivalent to a bottom-up development of ontologies. Future research should assess the feasibility of a top-down approach, where a set of standardized global models is managed by a central body. Enterprises that did not collaborate before could use such an ontology as a starting basis and as a complementation of collaboration-specific global models. Thus, such an ontology would ease the development of global models and lower the barriers for using the AIOS.

Format for Heavyweight AIOS Ontology The global models of the AIOS are used by collaboration partners to define a shared understanding of concepts; furthermore, they are supposed to be reused in different collaborative business process. In other words, global models represent ontologies.[915] However, the aim of this thesis was rather to describe the concepts needed in collaborative business processes – different types of organization elements, business functions, documents, etc. – instead of developing the format necessary to represent a heavyweight ontology. Thus, the standards used in the different AIOS dimensions had no binding character, and no specific modeling standard was defined or bindingly recommended. Future research should clarify how beneficial the usage of standards that explicitly aim at the representation of ontologies would be for the AIOS. Potential benefits include an easier mapping of "local ontologies" (e.g. public models used only inside one organization) to the mapping of "global ontologies" (e.g. global models used in a specific collaboration), as well as a better identification of concepts available in a collaboration sphere; or in SOA terms, better means to describe and discover services. Thus, similar to ontologies from the areas of enterprise architecting or eCommerce,[916] the AIOS metamodels specified in Chapter 4 could be implemented in semantic web standards like OWL-S. Due to the proximity of the AIOS to ARIS, this ontology could build on existing concepts that use ARIS as a basis for web service ontologies.[917]

Enabling (Dynamic) Discovery of Services with the BII The Business Interoperability Interface describes the elements that an organization offers and expects from partners in collaborative business processes. Thus, the BII can also be used to describe the boundaries of a web service, describing not only the function, but also the process, data and organization dimension of it. It can be expected that such a comprehensive service description is necessary to enable the discovery of services from a usually very large number of available services: Only this

[914] Compare HOFER (2006), pp. 80.
[915] Compare also p. 113.
[916] Compare JARRAR, VERLINDEN & MEERSMAN (2003) and IDEAS (2009).
[917] Compare ANG ET AL. (2005).

multi-dimensional description of requirements for a service combined with a corresponding description of its capabilities can enable an automated run time service discovery. Thus, the usage of the BII in approaches for the automated discovery and composition of services should be investigated.

Bibliography

Abran, A., Moore, J. (2004): SWEBOK - Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, Los Alamitos, California.

Ackermann, J., Brinkop, F., Conrad, S., Fettke, P., Frick, A., Glistau, E., Jaekel, H., Kotlar, O., Loos, P., Mrech, H., Raape, U., Ortner, E., Overhage, S., Sahm, S., Schmietendorf, A., Teschke, T. and Turowski, K. (2002): Standardized Specification of Business Components. In: Turowski, K. (ed.): Memorandum of the working group 5.10.3 Component Oriented Business Application Systems.

Adam, O., Chikova, P., Hofer, A., Vanderhaeghen, D. (2005): Process-oriented Framework for Collaborative Business in Dynamic Networks. In: Mosca, R., Tonelli, F., Melioli, R. (eds.): Proceedings of the 7th International Conference on the Modern Information Technology in the Innovation Processes of the Industrial Enterprises (MITIP). Genova, Italy, pp. 84-89.

Aier, S., Riege, C., Winter, R. (2008): Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. Wirtschaftsinformatik, vol. 50:4, pp. 292-301.

Aier, S., Schönherr, M. (2006): Evaluating Integration Architectures –A Scenario-Based Evaluation of Integration Technologies. In.: Draheim, D., Weber, G. (eds.): TEAA 2005. LNCS 3888, Springer, Berlin et al., pp. 2–14.

Allweyer, T. (2007): Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen. In: Nüttgens, M., Rump, F., Gadatsch, A. (eds.): EPK 2007. Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. St. Augustin, Germany, pp. 23-38.

Alonso, G., Casati, F., Kuno, H., Machiraju, V. (2004): Web Services – Concepts, Architectures and Applications. Springer, Berlin et al.

Andres, T. (2006): Vom Geschäftsprozess zur Anwendung - Modellgetriebene Entwicklung betriebswirtschaftlicher Software. In: Scheer, A.-W., Kruppke, H., Jost, W., Kindermann, H. (eds.): Agilität durch ARIS Geschäfprozessmanagement. Springer, Berlin et al., pp. 231-242.

Andresen, A. (2003): Komponentenbasierte Softwareentwicklung - mit MDA, UML und XML. Carl Hanser, München.

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I.,, Weerawarana, S. (2003): Business Process Execution Language for Web Services, Version 1.1. Http://www.ibm.com/developerworks/library /specification/ws-bpel/, last accessed: September 2008.

Ang, C., Gu, Y., Sourina, O., Leng Gay, R. (2005): An ARIS-based Transformation Approach to Semantic Web Service Development. Proceedings of the 2005 international Conference on Cyberworlds. IEEE Computer Society, Washington, DC, pp. 297-306.

ANSI (2004): Role Based Access Control - ANSI INCITS 359-2004. American National Standard for Information Technology, Information Technology Industry Council.

Anzböck, R., Dustdar, S. (2005): Semi-automatic Generation of Web Services and BPEL Processes - A Model-Driven Approach. Business Process Management. LNCS 3649, Springer, Berlin et al., pp. 64-79.

ATHENA (2004): First Version of State of the Art in Enterprise Modelling Techniques and Technologies to Support Enterprise Interoperability. Deliverable D.A1.1, ATHENA IP, project number 507849.

ATHENA (2005 Operators): Computational Operators for Business Process Model Manipulation. Deliverable D.A2.2 – Annex 1. ATHENA IP, project number 507849.

ATHENA (2005): Cross-Organisational Business Process requirements and the State of the Art in Research, Technology and Standards. Deliverable D.A2.1, ATHENA IP, project number 507849.

ATHENA (2006 Procotols): Analysis of Industry Best Practice in Business Documents and Protocols. Deliverable D.A7.1, ATHENA IP, project number 507849.

ATHENA (2006): POP* Revised Framework. Deliverable D.A1.8, ATHENA IP, project number 507849.

ATHENA (2007 IOP): Specification of Interoperability Framework and Profiles, Guidelines and Best Practices, Deliverable D.A4.2, ATHENA IP, project number 507849.

ATHENA (2007): Business Interoperability Framework. Deliverable D.B31. ATHENA IP, project number 507849.

Austin, D., Barbir, A., Peters, E., Ross-Talbot, S. (2004): Web Services Choreography Requirements - W3C Working Draft 11 March 2004. Http://www.w3.org/TR/2004/WD-ws-chor-reqs-20040311/, last accessed: April 2009.

Australian Government (2007). The Australian Government Business Process Interoperability Framework - Enabling Seamless Service Delivery. Department of Finance and Administration. Australian Government Information Management Office. Http://www.agimo.gov.au/__data/assets/pdf_file/0007/58849/Business_Process_Interoeprabiltiy_Framework.pdf, last accessed: June 2008.

Axenath, B., Kindler, E., Rubin, V. (2005): An open and formalism independent meta-model for business processes. In: Kindler, E., Nüttgens (eds.): Workshop on Business Process Reference Models 2005 (BPRM 2005), Satellite event of the 3rd International Conference on Business Process Management, pp. 45–59.

Balzert, H. (2000): Lehrbuch der Software-Technik - Software-Entwicklung. 2nd Edition, Spektrum, Heidelberg.

Barros, A., Decker, G., Dumas, M., Weber, F. (2007): Correlation Patterns in Service-Oriented Architectures. In: Dwyer, M., Lopes, A. (eds.): FASE 2007. LNCS 4422, pp. 245–259. Springer, Berlin et al.

Barros, A., Dumas, M., Oaks, P. (2005): A Critical Overview of the Web Services Choreography Description Language (WS-CDL). BPTrends Newsletter, vol. 3:3.

Barros, A., Dumas, M., Oaks, P. (2006): Standards for Web Service Choreography and Orchestration: Status and Perspectives. In: Bussler, C. et al. (eds.). BPM 2005 Workshops. LNCS 3812, Springer, Berlin et al., pp. 61-74.

Bass, L., Clements, P., Kazmann, R. (2006): Software Architecture in Practice. 2nd Edition, Addison-Wesley.

Basten, T., van der Aalst, W. (2001): Inheritance of behavior. Journal of logic and algebraic programming, vol. 47:2, pp. 47–145.

Bastida, L., Berre, A., Elvesæter, B., Hahn, C., Johnsen, S., Kämper, S., Kerrigan, M., Larrucea, X., Limyr, A., Muth, M., Nilsen, G., Roman, D., Rubina, J., Stollberg, M. (2009): Model-driven methodology and architecture specification. Deliverable D2.1, SHAPE, project number 216408.

Bauer, B., Müller, J. (2004): MDA Applied - From Sequence Diagrams to Web Service Choreography. In: Koch, N., Fraternali, P., Wirsing, M. (eds.): ICWE 2004. LNCS 3140, Springer, Berlin et al., pp. 132–136.

Baumöl, U. (2006): Methodenkonstruktion für das Business/IT Alignment. Wirtschaftsinformatik, vol. 48:5, pp. 314-322.

Becker, J. (1996): Eine Architektur für Handelsinformationssysteme. In: Becker, J., Grob, H., Müller-Funk, U., Vossen, G. (eds.): Arbeitsberichte des Instituts für Wirtschaftsinformatik, vol. 46, July 1996, Münster.

Becker, J. Algermissen, L. Falk, T. (2007): Prozessorientierte Verwaltungsmodernisierung - Prozessmanagement im Zeitalter von E-Government und New Public Management. Springer, Berlin et al.

Becker, J., Algermissen, L., Pfeiffer, D., Räckers, M. (2007): Bausteinbasierte Modellierung von Prozesslandschaften mit der PICTURE-Methode am Beispiel der Universitätsverwaltung Münster. Wirtschaftsinformatik, vol. 49:4, pp. 267-279.

Becker, J., Delfmann, P., Knackstedt, R., Kuropka, D. (2002): Konfigurative Referenzmodellierung. In: Becker, J., Knackstedt, R. (eds.): Wissenensmanagement mit Referenzmodellen. Physica, Heidelberg, pp. 25-144.

Becker, J., Janiesch, C., Pöppelbuß, J. (2008): Konfiguration kollaborativer Informationsmodelle. Proceedings of Multikonferenz Wirtschaftsinformatik (MKWI 2008), ISIH-Workshop - ERP und Collaborative Business. München, pp. 813-824.

Becker, J., Kahn, D. (2005): Der Prozess im Fokus. In: Becker, J., Kugeler, M., Rosemann, M. (eds.): Prozessmanagement - Ein Leitfaden zur prozessorientierten Organisationsgestaltung. 5th Edition, Springer, Berlin.

Becker, J., Pfeiffer, D. (2005). Beziehungen zwischen behavioristischer und konstruktionsorientierter Forschung in der Wirtschaftsinformatik. Proceedings of the Fachtagung Fortschrittskonzepte und Fortschrittsmessung in Betriebswirtschaftslehre und Wirtschaftsinformatik. Essen, pp. 39-57.

Becker, J., Schütte, R. (2004): Handelsinformationssysteme – Domänenorientierte Einführung in die Wirtschaftsinformatik. 2nd Edition, Redline Wirtschaft, Frankfurt.

Beeckman, D. (1989): CIMOSA - Computer Integrated Manufacturing Open Systems Architecture. International Journal for Computer Integrated Manufacturing, vol. 2:2, March - April 1989, pp. 94 – 105.

Beeri, C., Bernstein, P., Goodman, N. (1989): A model for concurrency in nested transaction systems. Journal of the ACM, vol. 36:2, pp. 230–269.

Bellmann, K. (2002): Produktion und Beschaffung - Management einer innerbetrieblichen Schnittstelle. In: Hahn, D., Kaufmann, L. (eds.): Handbuch Industrielles Beschaffungsmanagement - Internationale Konzepte, innovative Instrumente, aktuelle Praxisbeispiele. 2nd Edition, Gabler, Wiesbaden, pp.361-379.

Berardi, D., Giacomo, G., Mecella, M., Calvanese, D. (2005): Automatic Composition of Process-based Web Services - a Challenge. Fourteenth International World Wide Web Conference (WWW2005). Chiba, Japan.

Bernauer, M., Kappel, G., Kramler, G. (2003): Comparing WSDL-based and ebXML-based Approaches for B2B Protocol Specification. ICSOC, LNCS 2910, Springer, Berlin et al.

Bernus, P., Schmidt, G. (2006): Architectures of Information Systems. In: Bernus, P., Mertins, K. Schmidt, G. (eds.): Handbook on Architectures of Information Systems. 2nd Edition, Springer, Berlin et al.

Blanc, S. (2005): Interoperability problems - Management of Evolution of Collaborative Enterprises. Interop-ESA ’05, Doctoral Symposium, Geneva.

Born, M., Greiner, U., Lippe, S., Ruggaber, R., Kahl, T., Ziemann, J., Jäkel, F. (2009): The ATHENA Framework for Cross-Organizational Business Processes. In: di Nitto, E., Traverso, P., Sassen, A., Zwegers, A.: At your Service – Service-oriented Computing from an EU Perspective. MIT Press Series on Information Systems, pp. 457-488.

Boujraf, A., Aydogmus, O., Vermer, R. (2006): Integration, Analysis and First Evaluation of Case Studies. Deliverable WP3-D5, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Boujraf, A., Noble, M. (2007): Delivery of Final Master Case Study. Deliverable WP3-D7, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Boujraf, A. (2007): Eurojust/Europol Interaction Flows. Deliverable WP11-D1, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Bremer, G. (1998): Genealogie von Entwicklungsschemata. In: Kneuper, R., Müller-Luschnat, G., Oberweis, A. (eds.): Vorgehensmodelle für die betriebliche Anwendungsentwicklung. Teubner, Stuttgart et al.

Brown, A., Conallen, J., Tropeano, D. (2005): Introduction - Models, Modeling, and Model-Driven Architecture (MDA). In: Beyeda, S., Book, M., Gruhn, V. (eds.): Model-Driven Software Development. Springer, Berlin et al., pp. 1-16.

Bruno, G., La Rosa, M. (2006): From Collaboration Models to BPEL Processes Through Service Models. In: Bussler, C. et al. (eds.): BPM 2005 Workshop, LNCS 3812, Springer, Berlin et al., pp. 75-88.

Bugajski, J., Grossman, R., Vejcik, S. (2006): A Service Oriented Architecture Supporting Data Interoperability for Payments Card Processing Systems. In: Dan, A., Lamersdorf, W. (eds.): ICSOC 2006, LNCS 4294, Springer, Berlin et al., pp. 591 – 600.

Bussler, C. (2002): Public Process Inheritance for Business-to-Business Integration, TES 2002, LNCS 2444, Springer, Berlin et al., pp. 19-28.

Camarinha-Matos, L. Afsarmanesh, H. (2007): A comprehensive modeling framework for collaborative networked organizations. Journal of Intelligent Manufacturing, vol. 18:5, pp. 527-615.

Cambridge (2005): Cambridge Advanced Learner's Dictionary. 2nd Edition, Cambridge University Press.

Campbell, S., Mohun, V. (2006): Mastering Enterprise SOA with SAP NetWeaver and mySAP ERP. Wiley Publishing, Indiana.

Carney, D., Fisher, D., Morris, E., Place, P. (2005): Some Current Approaches to Interoperability. Integration of Software-Intensive Systems Initiative, technical Note CMU/SEI-2005-TN-033.

Chebbi, I., Dustdar, S., Tata, S. (2006): The view-based approach to dynamic inter-organizational workflow cooperation. Data & Knowledge Engineering, vol. 56:2, February 2006, pp. 139-173.

Chiu, D., Karlapalem, K., Li, Q., Kafeza, E. (2002): Workflow view based e-contracts in a cross-organizational e-services environment. Distributed and Parallel Databases, vol. 12:2-3, pp. 193–216.

Chiu, D., Shan, Z., Hung, P., Li, Q. (2005): Designing Workflow Views with Flows for Large-Scale Business-to-Business Information Systems. In: Shan, M. et al. (eds.): TES 2004, LNCS 3324, Springer, Berlin et al., pp. 107 – 121.

Clark, J., Casanave, C., Kanaskie, K., Harvey, B., Clark, J., Smith, N., Yunker, J., Riemer, K. (2001): ebXML Business Process Specification Schema Version 1.01. UN/CEFACT and OASIS.

Cummins, F. (2005): Organization Structure Metamodel (OSM). Draft Submission Overview. Http://www.omg.org/ docs/bmi/05-12-05.pdf, last accessed: June 2008.

Cuntz, N., Kindler, E. (2006): EPC Tools. Http://wwwcs.upb.de/cs/kindler/Forschung/EPCTools/, last accessed: March 2009.

Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P. (2002): A fine-grained access control system for XML documents. ACM Transactions for Information System Security, vol. 5:2, May 2002, pp. 169-202.

Damodaran, S. (2004): B2B integration over the Internet with XML - RosettaNet successes and challenges. Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters. New York, USA, pp. 188-195.

Daniel, F., Pernici, B. (2006): Insights into Web Service Orchestration and Choreography. International Journal of E-Business Research, vol. 2:1, pp. 58-77.

Davenport, T. (1993): Process Innovation - Reengineering work through information technology. Harvard Business School Press, Boston.

Decker, G. (2008): Choreografiemodellierung: Eine Übersicht. Informatik-Spektrum, vol. 31:2, April 2008, Springer, Berlin et al., pp. 161-166.

Decker, G., Kopp, O., Leymann, F., Pfitzner, K., Weske, M. (2008 Tool): Modeling Service Choreographies using BPMN and BPEL4Chor. In: Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE '08). LNCS 5074, Springer, Berlin et al., pp. 79-93.

Decker, G., Kopp, O., Leymann, F., Weske, M. (2007): BPEL4chor - Extending BPEL for modeling choreographies. Proceedings of the 2007 IEEE international conference on Web Services (ICWS 2007). IEEE Computer Society, New York, pp. 296–303.

Decker, G., Puhlmann, F. (2007): Extending BPMN for Modeling Complex Choreographies. On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, LNCS 4803, Springer, Berlin et al.

Department of the Treasury (2001): Treasury Enterprise Architecture Framework - Version 1, July 2001. Department of the Treasury, Chief Information Officer Council.

DFDL (2008): Data Format Description Language WG (DFDL-WG). Http://forge.gridforum.org/projects/dfdl-wg/, last accessed: October 2008.

Diedrich, E., Schmidt, D., Wimmer, M. (2006): A Three Dimensional Framework to Realize Interoperability in Public Administrations. In: Abecker, A., Mentzas, G., Stojanovic, L. (eds.): Proceedings of the Workshop on Semantic Web for eGovernment at the European Semantic Web Conference 2006. Pp. 54-59.

Diedrich, E., Wimmer, M., Schmidt, D., Mondorf, A., Matheis, T., Ziemann, J., Freiheit, J. (2007): Analysis of R4eGov Scenarios for Interoperability. Deliverable WP4-D3, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Dijkstra, E. (1982): On the role of scientific thought. In: Dijkstra, E.: Selected writings on Computing - A Personal Perspective. Springer, New York, pp. 60-66.

Ding, Y., Fensel, D., Klein, M. Omelayenko, B., Schulte, E. (2004): The Role of Ontologies in eCommerce. eStaab, S., Studer, R. (eds.): Handbook on Ontologies. Springer, Berlin et al., pp 593-616.

Duden (2006): Das Herkunftswörterbuch – Etymologie der deutschen Sprache. 4th ed., Dudenverlag, Mannheim, Germany.

Duden (2007): Das Fremdwörterbuch. 9th Edition, Dudenverlag, Mannheim, Germany.

EAW (2009): The European Arrest Warrant Project. Http://www.eurowarrant.net, last accessed: January 2009.

eEurope (2005): An Information Society for all: E-Government. Http://europa.eu.int/information_society/eeurope/2005/all_about/action_plan/index_en.htm, last accessed: August 2006.

Elvesæter, B., Hahn, A., Berre, A., Neple, T. (2005): Towards an Interoperability Framework for Model-Driven Development of Software Systems. In: Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N. (eds.): Interoperability of Enterprise Software and Applications. Springer, London, pp. 409-420.

Engels, G. (2008): Modellierungssprache. Enzyklopädie der Wirtschaftsinformatik, Online Lexikon. Oldenbourg Wissenschaftsverlag, Germany.

Eurojust (2008): Annual Report 2007. Http://eurojust.europa.eu/press_releases/annual_reports/2007/Annual_Report_2007_EN.pdf, last accessed: June 2008.

European Commission (2003): Linking-up Europe: The importance of interoperability for e-government services. Commission of the European Communities, Staff Working Document.

European Commission (2004): European Interoperability Framework for pan-European eGovernment Services - Version 1.0. Http://europa.eu.int/idabc, last accessed: August 2006.

Europol (2007): Annual Report 2006. Http://www.europol.europa.eu/publications/Annual_Reports/EuropolAnnualReport2006.pdf, last accessed: June 2008.

FEA (2007): FEA Consolidated Reference Model Document – Version 2.2. Executive Office of the President of the United States, Office of Management and Budget. Http://www.va.gov/oit/ea/4_3/extDocs/ea/FEA/FEA%20Consolidated%20Reference%20Model.pdf, last accessed: February 2009.

Feier, C., Roman, D., Polleres, A., Domingue, J., Stollberg, M., Fensel, D. (2005): Towards Intelligent Web Services: Web Service Modeling Ontology (WSMO). International Conference on Intelligent Computing (ICIC) 2005, Hefei, China.

Ferguson, D., Storey, T., Lovering, B., Shewchuk, J. (2003): Secure, Reliable, Transacted Web Services - Architecture and Composition. IBM Developer works. Http://www.ibm.com/developerworks/webservices/library/ws-securtrans/, last accessed: June 2008.

Ferraiolo, D, Kuhn, D. (1992): Role Based Access Control. 15th National Computer Security Conference, pp. 554-563.

Ferstl, O., Sinz, E. (2008): Grundlagen der Wirtschaftsinformatik. 6th Edition, Oldenbourg Wissenschaftsverlag, Germany.

Ferstl, O., Sinz, E., Amberg, M. (1996): Stichwörter zum Fachgebiet Wirtschaftsinformatik. In: Broy, M., Spaniol, O. (eds.): Lexikon Informatik und Kommunikationstechnik, 2nd Edition, VDI, Düsseldorf.

Fettke, P. (2006): Referenzmodellevaluation – Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Güterkriterien. In: Loos, P. (ed.): Wirtschaftsinformatik – Theorie und Anwendung. Logos, Berlin.

Fettke, P. (2008): Empirisches Business Engineering. Grundlegung und ausgewählte Ergebnisse. Professorial dissertation, University of Saarland, Saarbrücken.

Fettke, P., Loos, P. (2007): Framework and Meta Model for Specifying Business Components. In: Business Process Management Journal, vol. 13:5, pp. 628-643.

Fettke, P., vom Brocke, J. (2008): Referenzmodell. Enzyklopädie der Wirtschaftsinformatik, Online Lexikon. Oldenbourg Wissenschaftsverlag, Germany.

Filograna, A., Giunta, G., Ingraffia, N., Loiacono, L., Corallo, A., Za, D. (2007): An integrated approach for modeling business processes using BPMN and XPDL standards. Knowledge Management in Organisations (KMO'2007), Leece, Italy.

Fink, A., Schneidereit, G., Voß, S. (2005): Grundlagen der Wirtschaftsinformatik. 2nd Edition, Physica, Heidelberg.

Fisteus, J., Fernández, L., Kloos, C. (2005): Applying model checking to BPEL4WS business collaborations. Proceedings of the 2005 ACM Symposium on Applied Computing, New York. Pp. 826-830.

Ford, T., Colombi, J., Graham, S., Jacques, D. (2007): A Survey on Interoperability Measurement. 12th International Command and Control Research and Technology Symposium (ICCRTS), June 19-21, Newport, USA.

Foster, H., Uchitel, S., Magee, J., Kramer, J., (2007): WS-Engineer: A Model-Based Approach to Engineering Web Service Compositions and Choreography. In: Baresi, L,. Di Nitto, E. (eds.): Test and Analysis of Web Services. Springer, Berlin et al, pp. 87-119.

Frank, U. (1994): Multiperspektivische Unternehmensmodellierung – Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg, München.

Frank, U. (2006): Towards a Pluralistic Conception of Research Methods in Information Systems. ICB Research Report Number 7, University Duisburg-Essen, Duisburg.

Frank, U. (2008): Multiperspektivische Unternehmensmodellierung. Enzyklopädie der Wirtschaftsinformatik, Online Lexikon. Oldenbourg Wissenschaftsverlag, Germany.

Frank, U., van Laak, B. (2003): Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsbericht des Instituts für Wirtschafts- und Verwaltungsinformatik der Universität Koblenz Number 34, Januar 2003.

Frankel, D. (2003): Model Driven Architecture – Appying MDA to Enterprise Computing. Wiley Publishing, Indiana.

Freiheit, J., Mohndorf, A. (2007): Formal Analysis of web service standards. Deliverable WP4-D4, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Freudenthal, E., Pesin, T., Port, L., Keenan, E., Karamcheti, V. (2002): dRBAC - Distributed Role-Based Access Control for Dynamic Coalition Environments. Proceedings of 22nd IEEE International Conference on Distributed Computing Systems. Pp. 411-420.

Fu, X., Bultan, T., Jianwen, S. (2003): A top-down approach to modeling global behaviors of web service. Workshop on Requirements Engineering and Open Systems (REOS), Monterey, California.

Gabriel, R. (2008): Informationssystem. Enzyklopädie der Wirtschaftsinformatik, Online Lexikon. Oldenbourg Wissenschaftsverlag, Germany.

Gadatsch, A. (2008): Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 5th Edition, Vieweg, Wiesbaden.

Gerth, E. (1971): Zwischenbetriebliche Kooperation. Poeschel, Stuttgart.

Glushko, R., McGrath, T. (2002): Document engineering for e-business. Proceedings of the 2002 ACM Symposium on Document Engineering. ACM, New York, pp. 42-48.

Goeken, M. (2003): Die Wirtschaftsinformatik als anwendungsorientierte Wissenschaft. Arbeitsbericht 01/2003, Institut für Wirtschaftsinformatik, Universität Marburg.

Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004): Ontological Engineering – with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, Berlin et al.

Greefhorst, D., Koning, H., van Vliet, H. (2006): The many faces of architectural descriptions. Information Systems Frontiers, vol. 8:2, pp. 103-113.

Greiner, U., Legner, C., Lippe, S., Wende, K. (2007): Business Interoperability Profiles - Relating Business Interoperability Issues To Technical Interoperability Solutions. Enterprise Interoperability II. Springer, Berlin et al.

Greiner, U., Lippe, S., Kahl, T., Ziemann, J., Jaekel, F. (2006): Designing and Implementing Cross-Organizational Business Processes – Description and Application of a Modeling Framework. I-ESA Conference, Bordeaux, France.

Grief, J., Seidlmeier, H. (2005): Modellierung von Flexibilität mit Ereignisgesteuerten Prozessketten (EPK). In: Nüttgens, M., Rump, F. (eds.).: Proceedings of the 4th GI Workshop on Event-Driven Process Chains (EPK 2005). Hamburg, Germany, CEUR workshop proceedings, vol. 167, pp. 137-155.

Gruber, T. (1993): A translation approach to portable ontologies. Knowledge Acquisition, vol. 5:2, pp. 199-220.

Gruhn, V., Pieper, D., Röttgers, C. (2006): MDA – Effektives Software-Engineering mit UML 2 und Eclipse. Springer, Berlin et al.

Gulati, R., Nohria, N., Zaheer, A. (2006): Strategic Networks. In: Hahn, D., Taylor, B. (eds.): Strategische Unternehmungsplanung — Strategische Unternehmungsführung. Springer, Berlin et al., pp. 293-309.

Guo, H., Lin, X. (2006): A Study on an Application Integrated Model Supporting Inter-enterprise Collaboration. Computer Supported Cooperative Work in Design II. LNCS 3865, Springer, Berlin et al., pp. 463 – 472.

Guo, J. (2006): Inter-enterprise business document exchange. Proceedings of the 8th international Conference on Electronic Commerce (ICEC '06). ACM, New York, pp. 427-437.

Hack, S, Lindemann, M. (2007): Enterprise SOA einführen. Galileo Press, Bonn.

Haesen, R., Snoeck, M., Lemahieu, W., Poelmans, S. (2008): On the Definition of Service Granularity and Its Architectural Impact. Advanced Information Systems Engineering, LNCS 5074, Springer, Berlin et al, pp. 375-389.

Hailpern, B., Tarr, P. (2006): Model-driven development - the good, the bad, and the ugly. IBM Systems Journal, vol. 45:3, pp. 451-461.

Hammer, M., Champy, J. (1993): Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business.

Han, D., Shim, J., Kwak, M. (2000): Workflow System Architectures and Their Performance Model. The 4th Pacific Asia Conference on Information Systems (PACIS). Hong Kong, China, p. 1088-1102.

Hansen, H., Neumann, G. (2001): Wirtschaftsinformatik I. 8th Edition, Lucius & Lucius, Stuttgart.

Hanson, J., Nandi, P., Kumaran, S. (2002): Conversation Support for Business Process Integration. Proceedings of the Sixth international Enterprise Distributed Object Computing Conference (Edoc'02). IEEE Computer Society, Washington, DC, pp. 65.

Hartono, E., Holsapple, C. (2004): Theoretical foundations for collaborative commerce research and practice. Information Systems and E-Business Management, vol. 2:2., pp. 1-30.

Heutschi, R., Österle, H., Winter, R., Brenner, W. (2007): Serviceorientierte Architektur - Architekturprinzipien und Umsetzung in der Praxis. Springer, Berlin et al.

Hevner, A., March, S., Park, J., Ram, S. (2004): Design science in information systems research. MIS Quarterly, vol. 28:1, pp. 75–105.

Hevner, A. (2009): Interview with Alan R. Hevner on “Design Science”. Interview lead by Winter, R. Business and Information Systems Engineering, vol. 1, pp. 126-129.

Hinterhuber, A. (2002): Value Chain Orchestration in Action and the Case of the Global Agrochemical Industry. Long Range Planning, vol. 35:6, pp. 615-635.

Hofer, A. (2007): Prozessorientiertes Kooperationsmanagement – Methoden, Vorgehensmodell und Anwendungsszenario. Logos, Berlin.

Hofer, A., Adam, O., Zang, S., Scheer, A.-W. (2005): Architektur zur Prozessinnovation in Wertschöpfungsnetzwerken. Veröffentlichungen des Instituts für Wirtschaftsinformatik, vol. 181, Saarbrücken, Germany.

Hofreiter, B., Huemer, C. (2008): A model-driven top-down approach to inter-organizational systems: From global choreography models to executable BPEL. Joint Conference on E-Commerce Technology (CEC'08) and Enterprise Computing, E-Commerce, and E-Services (EEE'008).

Hofreiter, B., Huemer, C., Kim, J. (2006): Choreography of ebXML business collaborations. Information Systems and E-Business Management, vol. 4:3, Juli 2006, pp. 221-243.

Hohpe, G. (2007): Konversationen zwischen lose gekoppelten Services. In: Starke, G., Tilkow, S. (eds.): SOA-Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt, Heidelberg, pp. 439-446.

Holmes, T., Tran, H., Zdun, U., Dustdar, S. (2008): Modeling Human Aspects of Business Processes - A View-Based, Model-Driven Approach. 4th European Conference on Model Driven Architecture Foundations and Applications, pp. 247-262.

Hoyer, V., Bucherer, E., Schnabel, F. (2008): Collaborative e-Business Process Modelling: Transforming Private EPC to Public BPMN Business Process Models. Business Process Management Workshops, LNCS 4928, pp. 185- 196. Springer, Berlin et al.

Hubert, R. (2001): Convergent Architecture – Building Model-Driven J2EE Systems with UML. John Wiley & Sons, New York et al.

Huemer, C., Liegl, P. (2007): A UML Profile for Core Components and their Transformation to XSD. IEEE 23rd International Conference on Data Engineering Workshop, IEEE Computer Society.

Huth, S., Wieland, T. (2007): Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK. In: Nissen, V., Petsch, M., Schorscht, H. (eds.): Service-orientierte Architekturen – Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Gabler, Wiesbaden, Germany, pp. 61-76.

IDABC (2005): The IDABC program (2005-2009). Http://europa.eu/scadplus/leg/en/lvb/l24147b.htm, last accessed: June 2008.

IDEAS (2009): International Defence Enterprise Architecture Specification Group. http://www.ideasgroup.org/, last accessed: March 2009.

IEEE (2007): IEEE 1471 Website, IEEE Std. 1471 Frequently Asked Questions (FAQ) - Version 5.0, 19 July 2007. Http://www.iso-architecture.org/ieee-1471/ieee-1471-faq.html, last accessed: May 2009.

IFIP-IFAC (2000): Generalised Enterprise Reference Architecture and Methodology Version 1.6.3. IFIP–IFAC Task Force on Architectures for Enterprise Integration. Http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html, last accessed: August 2008.

INTEROP (2007): Report on Model Interoperability. Deliverable DTG2.2, Interop NOE, project number 508011.

ISO (1994): OSI Basic Reference Model – The Basic Model. ISO/IEC 7498-1, ISO - International Organization for Standardization.

Jablonski, S., Böhm, M., Schulze, W. (1997): Workflow Management - Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie. dpunkt, Heidelberg.

Janiesch, C., Dreiling, A., Greiner, U., Lippe, S. (2006): Configuring processes and business documents – an integrated approach to enterprise systems collaboration. Proceedings of IEEE International Conference on e-Business Engineering (ICEBE). Shanghai, pp. 516-521.

Janiesch, C., Thomas, S. (2006): Business Document Taxonomy – Comparison of the State-of-the-art and Recommendations for Future Applications. International Journal of Interoperability in Business Information Systems (IBIS), vol. 2:2, pp. 59-78.

Janssen, M., Scholl, H. (2007): Interoperability for electronic governance. Proceedings of the 1st international Conference on theory and Practice of Electronic Governance (ICEGOV '07). ACM, New York, pp. 45-48.

Jarrar, M., Verlinden, R., Meersman, R. (2003): Ontology-Based Customer Complaint Management. Proceedings of On the Move to Meaningful Internet Systems 2003 - OTM 2003 Workshops. Springer, Berlin et al., pp. 594-606.

Jones, S. (2005): Toward an Acceptable Definition of Service. IEEE Software, vol. 22:3, pp. 87-93.

Jørgensen, H. (2001): Interaction as a Framework for Flexible Workflow Modelling. Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work.

Jost, W., Scheer, A.-W. (2002): Geschäftsprozessmanagement – Kernaufgabe einer jeden Unternehmensorganisation. In: Scheer, A.-W., Jost, W. (eds.): ARIS in der Praxis - Gestaltung, Optimierung und Implementierung von Geschäftsprozessen. Springer, Berlin et al., pp. 33-44.

Juric, M., Mathew, B., Sarang, P. (2006): Business Process Execution Language for Web Services. 2nd Edition, Packt Publishing.

Kahl, T., Vanderhaeghen, D., Werth, D. (2005): Interoperabilität in Unternehmensnetzwerken. Industrie Management, vol. 21:4, pp. 45-48.

Kahl, T., Zinnikus, I., Roser, S., Hahn, C., Ziemann, J., Müller, J., Fischer, K. (2007): Architecture for the Design and Agent-based Implementation of Cross-organizational Business Processes. In: Gonçalves, R., Müller, J., Mertins, K., Zelm, M. (eds.). Enterprise Interoperability II - New Challenges and Approaches, Springer, Berlin et al., pp. 207-218.

Kalat, J. (1999): Introduction to Psychology. 5th Edition, Wadsworth Publishing Company, Belmont, USA.

Karakostas, B., Zorgios, Y. (2008): Engineering Service Oriented Systems - A Model Driven Approach. IGI Global.

Karlapalem, K., Dani, A., Krishna, P. (2001): A Frame Work for Modeling Electronic Contracts. In: Kunii, H. Jajodia, S., Sølvberg, A. (eds.): Conceptual Modeling - ER 2001. LNCS 2224, Springer, Berlin et al., pp. 193-207.

Keller, G., Nüttgens, M., Scheer, A.-W. (1992): Semantische Prozessmodellierung. Veröffentlichungen des Instituts für Wirtschaftsinformatik, vol. 89, Saarbrücken, Germany.

Keller, M., Scherer, R., Menzel, K., Theling, T., Vanderhaeghen, D., Loos, P. (2006): Support of Collaborative Business Process Networks in AEC. ITcon - Electronic Journal of Information Technology in Construction, Special Issue Process Modelling, Process Management and Collaboration , pp. 449-465.

Keller, W. (2007): Regeln und Heuristiken für gutes Schnittstellendesign. In: Starke, G., Tilkov, S. (eds.): SOA-Expertenwissen - Methoden, Konzepte und Praxis serviceorientierter Architekturen. Dpunkt, pp. 357-380.

Kemper, A., Eickler, A. (2004): Datenbanksysteme – Eine Einführung. 5th Edition, Oldenbourg, München.

Khalaf, R. (2005): From RosettaNet PIPs to BPEL Processes - A Three Level Approach for Business Protocols. In: van der Aalst, W. et al. (eds.): BPM 2005, LNCS 3649, Springer, Berlin et al., pp. 364–373.

Kim, J., Huemer, C. (2004): From an ebXML BPSS choreography to a BPEL-based implementation. ACM SIGecom Exchange, vol. 5:2, November 2004, pp. 1-11.

Kindler, E. (2003): On the semantics of EPCs - A framework for resolving the vicious circle (Extended Abstract). In: Nüttgens, M., Rump, F. (eds.): EPK 2003 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens, Bamberg, pp. 7-18.

Kindler, E., Axenath, B., Rubin, V. (2006): AMFIBIA - A Meta-Model for the Integration of Business Process Modelling Aspects. In: Leymann, F., Reisig, W., Satish, R., van der Aalst, W. (eds.): The Role of Business Processes in Service Oriented Architectures. Dagstuhl Seminar Proceedings, vol. 06291.

Klaus, G. (1963). Kybernetik in philosophischer Sicht. Berlin, Dietz.

Klein, R., Kupsch, F., Scheer, A.-W. (2004): Modellierung inter-organisationaler Prozesse mit Ereignisgesteuerten Prozessketten. Veröffentlichungen des Instituts für Wirtschaftsinformatik, vol. 178, Saarbrücken.

Klischewski, R., Scholl, H. (2006): Information quality as a common ground for key players in e-government integration and interoperability. 39th Hawaii International International Conference on Systems Science (HICSS-39), Hawaii.

Kloppmann, M., König, D., Pfau, G., Scheible, M. (2007): Workflow von IBM - Eine Reise aus den Weiten des Weltalls ins Zentrum des On-Demand-Universums. IT – Technologien, Lösungen, Innovationen. Springer, Berlin et al.

Kontogiannis, K. , Lewis, G. , Smith, D., Litoiu, M., Muller, H., Schuster, S., Stroulia, E. (2007): The Landscape of Service-Oriented Systems - A Research Perspective. Proceedings of the International Workshop on Systems Development in SOA Environments (SDSOA '07). IEEE Computer Society.

Kopp, O., Unger, T., Leymann, F. (2006): Nautilus Event-driven Process Chains - Syntax, Semantics, and their mapping to BPEL. Proceedings of the 5th GI Workshop on Event-Driven Process Chains (EPK 2006). Vienna, Austria, pp. 85-104.

Krallmann, H., Schönherr, M., Dietrich, J., Offermann, P. (2007): B2B-Modellierungssprachen und –methodologien im Kontext der Konzeption und Implementierung Service Orientierter Architekturen. In: Loos, P., Krcmar, H.: Architekturen und Prozesse - Strukturen und Dynamik in Forschung und Unternehmen, Springer, Berlin et al.

Kramler, G., Kapsammer, E., Retschitzegger, W., Kappel, G. (2006): Towards Using UML 2 for Modelling Web Service Collaboration Protocols. In: Konsantas, D., Bourrieres, J., Leonard, M., Boudijilida, N. (eds.): Interoperability of Enterprise Software and Applications, Springer, London, pp. 227-238.

Kramler, G., Retschitzegger, W., Bernauer, M., Kappel, G. (2003): Specification of Interorganizational Workflows - A Comparison of Approaches. In: Callaos, N. Lesso, W., Schewe, K., Atlam, E. (eds.): Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003). International Institute of Informatics and Systemics, pp. 30 - 36.

Kronz, A. (2000): Implementing Workflow Systems Using ARIS Models. In: Scheer, A.-W.: ARIS – Business Process Modeling. 3rd Edition, Springer, Berlin et al., pp. 184-188.

Kruczynski, K. (2008): Business Process Modelling in the context of SOA - An empiric study of the acceptance between EPC and BPMN. Proceedings of the 5th International Conference on Information Technology and Applications (ICITA 2008). Cairns, Queensland, Australia.

Kupsch, F. (2006): Framework zur dezentralen Integration systemübergreifender Geschäftsprozesse. Logos, Berlin.

Kurose, J., Ross, K. (2002): Computer Networking – A Top-Down Approach Featuring the Internet. Addison-Wesley.

Lankhorst, M. (2005): Enterprise Architecture At Work – Modelling, Communication, and Analysis. Springer, Berlin et al.

Lee, H. (2007): Unraveling decentralized authorization for multi-domain collaborations. Proceedings of International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), pp. 33-40.

Lee, H., Luedemann, H. (2007): A Lightweight Decentralized Authorization Model for Interdomain Collaborations. ACM Workshop on Secure Web-Services, Fairfax, Virginia, USA, SWS '07. ACM, New York, pp. 83-89.

Legner, C., Vogel, T. (2008): Leveraging Web Services for Implementing Vertical Industry Standards - A Model for Service-Based Interoperability. Electronic Markets, vol. 18:1, pp. 39-52.

Legner, C., Vogel, T., Loehe, J., Mayerl, C. (2007): Transforming Inter-Organizational Business Processes into Service-Oriented Architectures - Method and Application in the Automotive Industry. 15th ITG/GI - Fachtagung Kommunikation in Verteilten Systemen (KivS), Bern.

Legner, C., Lebreton, B. (2007): Business Interoperability Research - Present Achievements and Upcoming Challenges. Electronic Markets, vol. 17:3, pp. 176 – 186.

Legner, C., Wende, K. (2006): Towards an Excellence Framework for Business Interoperability. Proceedings of the 19th Bled eConference eValues. Bled, Slovenia.

Legner, C., Wende, K. (2007): The Challenges of Inter-organizational Business Process Design - A Research Agenda. European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland.

Leymann, F., Roller, D. (2000): Production Workflow: Concepts and Techniques. Prentice Hall, New Jersey.

Leymann, F., Roller, D. (2006): Modeling Business Processes with BPEL4WS. Information Systems and E-Business Management, vol. 4:3, July 2006, pp. 265-284.

Leymann, F., Roller, D., Schmidt, M. (2002): Web Services and Business Process management. IBM Systems Journal, vol. 41:2, pp. 198-211.

Liegl, P. (2008): Business documents in a service oriented context. 10th International Conference on Electronic Commerce (ICEC08), Innsbruck.

Linthicum, D. (2001): B2B Application Integration. Addison-Wesly Longman, Amsterdam.

Lippe, S., Greiner, U., Barros, A. (2005): A Survey on State-of-the-Art to Facililitate Modelling of Cross-Organisational Business Processes. Proceedings of XML4BPM 2005. Karlsruhe.

Liu, D., Shen, M. (2003): Workflow modeling for virtual processes - an order-preserving process-view approach. Information Systems, vol. 28:6, pp. 505-532.

Lohmann, N., Gierds, C., Znamirowski, M. (2007): BPEL2oWFN - Translating BPEL Process to Open Workflow Nets. Version 2.0.3. Http://www.gnu.org/software/bpel2owfn/index.html, last accessed: November 2008.

Lohmann, N., Massuthe, P., Stahl, C, Weinberg, D. (2006): Analyzing Interacting BPEL Processes. International Conference on Business Process Modelling (BPM 2006), Vienna, Austria.

Loos, P. (1998): Integriertes Prozeßmanagement direkter und indirekter Bereiche durch Workflow-Management. Industrie Management, vol. 14:2, pp. 13-18.

Loos, P., Vanderhaeghen, D. (2007, eds.): Kollaboratives Prozessmanagement - Unterstützung kooperations- und koordinationsintensiver Geschäftsprozesse am Beispiel des Bauwesens. Logos, Berlin.

Masak, D. (2006): IT-Alignment – IT Architektur und Organisation. Springer, Berlin et al.

Masak, D. (2007): SOA? – Serviceorientierung in Business und Software. Springer, Berlin et al.

Massuthe, P., Reisig, W., Schmidt, K. (2005): An Operating Guideline Approach to the SOA. Annals of Mathematics, Computing & Teleinformatics, vol. 1:3, pp. 35-43.

Massuthe, P., Weinberg, D. (2007): Functional Interaction Analysis for open Workflow Nets. Version 2.0. Http://www2.informatik.hu-berlin.de/top/tools4bpel/fiona/, last accessed: November 2008.

Masud, S. (2003): Use RosettaNet-based Web Services, Part 1 - BPEL4WS and RosettaNet. IBM Developerworks. Http://www-128.ibm.com/developerworks/webservices/library/ws-rose1/, last accessed: April 2009.

Mathas, C. (2008): SOA intern – Praxiswissen zu Service-orientierten IT-Systemen. Carl Hanser, München, Wien.

Matheis, T., Loos, P. (2008): Monitoring cross-organizational business processes. International Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Government, (EEE 2008), Las Vegas, USA.

Matheis, T., Ziemann, J., Diedrich, E., Schmidt, D., Wimmer, M. (2007): Methodical interoperability requirements for eGovernment. Deliverable WP8-D1, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Matheis, T., Ziemann, J., Loos, P. (2006): A Methodical Interoperability Framework for Collaborative Business Process Management in the Public Sector. Proceedings of the Mediterranean Conference on Information Systems. Venice, Italy.

Matheis, T., Ziemann, J., Schmidt, D., Freiheit, J. (2008): R4eGov IOP tool suite for modelling cross-organizational processes and data. Deliverable WP4-D10, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

McGovern, J., Tyagi, S., Stevens, M. Mathew, S. (2003): Java Web Services Architecture. Elsevier, San Francisco.

Melzer-Ridinger, R. (2007): Supply Chain Management - Prozess- und unternehmensübergreifendes Management von Qualität, Kosten und Liefertreue. Oldenbourg Wissenschaftsverlag, Germany.

Mendling, J. (2007): Detection and Prediction of Errors in EPC Business Process Models. Doctoral Thesis, Vienna University of Economics and Business Administration. Vienna, Austria.

Mendling, J., Hafner, M. (2005): From Inter-Organizational Workflows to Process Execution - Generating BPEL from WS-CDL. In: R. Meersman, Z. Tari, P. Herrero et al. (eds.): Proceedings of OTM 2005 Workshops. LNCS 3762, Agia Napa, Cyprus, pp. 506-515.

Mendling, J., Neumann, G., Nüttgens, M. (2005): Towards Workflow Pattern Support of Event-Driven Process Chains (EPC). In Nüttgens, M., Mendling, J., (eds.): Proceedings of the 2nd Workshop XML4BPM. Karlsruhe, Germany, pp. 23–38.

Mendling, J., Nüttgens, M. (2004): Transformation of ARIS Markup Language to EPML. In: Nüttgens, M., Rump, F. (eds.): Proceedings of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004). Luxembourg, Luxembourg.

Mendling, J., Nüttgens, M. (2005): EPC Markup Language (EPML) - An XML-Based Interchange Format for Event-Driven Process Chains (EPC). Technical Report JM-2005-03-10. Vienna University of Economics and Business Administration.

Mendling, J., Ploesser, K., Strembeck, M. (2008): Specifying Separation of Duty Constraints in BPEL4People Processes. Business Information Systems 11th International Conference, BIS 2008, Innsbruck, Austria, pp. 273-284.

Mendling, J., Ziemann, J. (2005): Transformation of BPEL Processes to EPCs. In: Nüttgens, M., Rump, F. (eds.): Proceedings of the 4th GI Workshop on Event-Driven Process Chains (EPK 2005). Hamburg, Germany, CEUR Workshop Proceedings vol. 167, pp. 41-53.

Mertens, P., Bodendorf, F., König, W., Picot, A., Schumann, M., Hess, T. (2005): Grundzüge der Wirtschaftsinformatik. 9th Edition. Springer, Berlin et al.

Mertins, K., Jaekel, F. (2006): MO²GO - User Oriented Enterprise Models for Organizational and IT Solutions. In: Bernus, P., Mertins, K. Schmidt, G. (eds.): Handbook on Architectures of Information Systems. 2nd Edition, Springer, Berlin et al., pp. 649-663.

Miles, R., Snow, C. (1986): Organizations: New Concepts for New Forms. California Management Review, vol. 28:3, pp. 62-73.

MODINIS (2005): MODINIS program: Follow-up of eEurope 2005 action plan. Http://europa.eu/scadplus/leg/en/lvb/l24226c.htm, last accessed: August 2006.

Molinaro, A. (2006): SQL Cookbook. O'Reilly, Sebastopol, California.

Mondorf, A., Schmidt, D., Ziemann, J., Freiheit, J. (2008): Specification of Functional and Non-functional Data required by the IOP Gateway. Deliverable WP4-D6, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Nastanky, L., Hilpert, W. (1994): The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automatisation. In: Wolfinger, B. (ed.). Innovationen bei Rechen- und Kommunikationssystemen. Springer, Berlin et al.

Newcomer, E., Lomow, G. (2005): Understanding SOA with Web Services. Addison-Wesley.

Nitu (2009): Configurability in SaaS (software as a service) applications. Proceedings of the 2nd Annual Conference on India Software Engineering Conference, ISEC '09. ACM, New York, NY, pp. 19-26.

Noran, O. (2004): A Meta-Methodology for Collaborative Networked Organisations. Doctoral dissertation, Faculty of Engineering and Information Technology, Griffith University Brisbane, Australia.

Nunamaker, J., Chen, M., Purdin, T. (1999): System Developments in Information Systems Research. Journal of Management Information Systems, vol. 7:3, pp. 89-106.

O'Sullivan, J., Edmond, D., ter Hofstede, A. (2002): What's in a Service? Towards Accurate Description of Non-Functional Service Properties. Distributed and Parallel Databases, vol. 12. Kluwer Academic Publishers, Netherlands, pp. 117–133.

OAGIS (2007): Open Applications Group Integration Specification (OAGIS) - Release 9.1. Http://openapplications.org/oagis/9.1/index.html, last accessed: October 2008.

OASIS (2004): Universal Business Language 1.0. Http://docs.oasis-open.org/ubl/cd-UBL-1.0/, last accessed: October 2008.

OASIS (2005 RBAC): Core and Hierarchical Role-based Access Control (RBAC) Profile of XACML v2.0. Http://docs.oasis-open.org/xacml/2.0/access controlxacml-2.0-rbac-profile1-spec-os.pdf, Feb. 2005, last accessed: July 2008.

OASIS (2005): Extensible Access Control Markup Language (XACML) Version 2.0, OASIS Standard, 1. Feb. 2005. Http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, last accessed: July 2008.

OASIS (2007): Web Services Business Process Execution Language Version 2.0 - OASIS Standard, 11 April 2007. Http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf, last accessed: July 2008.

OASIS (2008 B4P): WS-BPEL Extension for People (BPEL4People) Specification Version 1.1 - Working Draft 02. Http://docs.oasis-open.org/bpel4people/bpel4people-1.1-spec-wd-02.html, last accessed: May 2008.

OASIS (2008 HT): Web Services – Human Task (WS-HumanTask) Specification Version 1.0 - Working Draft 02. Http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-wd-02.html, last accessed: May 2008.

OASIS (2008): Reference Architecture for Service Oriented Architecture Version 1.0 - Public Review Draft 1. Http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf, last accessed: July 2008.

OMG (2000): OMG Unified Modeling Language Specification - Version 1.3. Http://www.omg.org/docs /formal/00-03-01.pdf, last accessed: March 2009.

OMG (2003): MDA Guide Version 1.0.1. Http://www.omg.org/docs/omg/03-06-01.pdf, last accessed: February 2009.

OMG (2004): Enterprise Collaboration Architecture (ECA) Specification - Version 1.0. Http://www. omg.org/docs/formal/04-02-01.pdf, last accessed: February 2009.

OMG (2006 BPMN): Business Process Modeling Notation (BPMN) Specification - Final Adopted Specification, February 2006. Http://www.bpmn.org/Documents/OMG%20Final%20 Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf, last accessed: March 2009.

OMG (2006): Business Motivation Model (BMM) Specification - Adopted Specification. Http://www. omg.org/docs/dtc/06-08-03.pdf, last accessed: June 2008.

OMG (2008): OMG Model Driven Architecture. Http://www.omg.org/mda/, last accessed: May 2008.

O'Riordan, D. (2002): Business Process Standards for Web Services. In: Clark, M., Fletcher, P., Hanson, J.J., Irani, R., Waterhouse, M., Thelin, J. (eds): Web Services Business Strategies and Architectures. Wrox Press, pp 157-174.

Ortner, E. (2008): Für die Hochzeit zwischen Organisationen und IT-Architektur fehlt noch die Reife. Computer Zeitung, vol. 27, 30th of June, p. 6.

Österle, H. (2000): Geschäftsmodell des Informationszeitalters. In: Österle, H., Winter, R. (eds.): Business Engineering – Unternehmen der Zukunft gestalten. Springer, Berlin et al.

Österle, H., Fleisch, E., Rainer, A. (2001): Business Networking – Shaping Collaboration Between Enterprises. Springer, Berlin et al.

Ouyang, C., van der Aalst, W., Dumas, M., ter Hofstede, A. (2006): Translating BPMN to BPEL. BPM Center Report BPM-06-02, BPMcenter.org.

Overhage, S., Turowski, K. (2007): Serviceorientierte Architekturen – Konzept und methodische Herausforderungen. In: Nissen, V., Petsch, M., Schorscht, H. (eds.): Service-orientierte Architekturen. Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Deutscher Universitätsverlag.

Oxford (2005): Oxford Advanced Learner's Dictionary. 7th Edition, Oxford University Press.

Palmer, N. (2006): Understanding the BPMN-XPDL-BPEL Value Chain. Business Integration Journal, November/December 2006, pp. 54-55.

Panetto, H., Scannapieco, M., Zelm, M. (2004): INTEROP NoE - Interoperability Research for Networked Enterprises Applications and Software. Proceedings of On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops. Springer, Berlin et al., pp. 866-882.

Pankratius, V. , Stucky, W. (2005): A formal foundation for workflow composition, workflow view definition, and workflow normalization based on petri nets. Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling. Vol. 43, Newcastle, Australia, pp. 79-88.

Papazoglou, M. (2003): Web Services and Business Transactions. World Wide Web: Internet and Web Information Systems, vol. 6:1, pp. 49-91.

Papazoglou, M., van den Heuvel, W. (2007): Service oriented architectures - approaches, technologies and research issues. The VLDB Journal, vol. 16, Springer, Berlin et al., pp. 389–415.

Peltz, C. (2003): Web Service Orchestration and Choreography - A look at WSCI and BPEL4WS. Web Services Journal, vol. 3:7.

Pérignon, I., Daucé, C. (2007): The European Arrest Warrant: a growing success story. ERA-Forum, vol. 8:2, Springer, Berlin et al., pp. 203-215.

Peristeras, V., Tarabanis, K. (2006): The Connection, Communication, Consolidation, Collaboration Interoperability Framework (C4IF) For Information Systems Interoperability. In IBIS – Interoperability in Business Information Systems, vol. 1:1.

Pernul, G., Unland, R. (2003): Datenbanken im Unternehmen – Analyse, Modellbildung und Einsatz. 2nd Edition, Oldenbourg, München.

Plate, H., Huys, D. (2007): Specification and design of an extensible execution environment for collaborative eGovernment processes. Deliverable WP5-D8, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Porter, M. (1985): Competitive Advantage - Creating and Sustaining Superior Performance. New York, The Free Press.

Prahalad, C., Hamel, G. (2006): The Core Competence of the Corporation. In: Hahn, D., Taylor, B. (eds.): Strategische Unternehmungsplanung — Strategische Unternehmungsführung. Springer, Berlin et al., pp. 275-292.

Public Safetey & Homeland Security Bureau (2008): Tech Top 1 – Interoperability. Http://www.fcc .gov/pshs/techtopics/tech-interop.html, last accessed: September 2008.

Queensland Government Chief Information Office (2007): Queensland Government Enterprise Architecture – Framework. Queensland, May 2007.

Rausch-Schott, S. (1997): TriGSflow – Workflow Management Based on Active Object-Oriented Database Systems and Extended Transaction Mechanisms. PhD thesis, University Linz.

Ray, D., Gulla, U., Gupta, M., Dash, S. (2008): Interoperability and constituents of Interoperable Systems. In: Weerakkody, V., Janssen, M., Dwivedi, Y. (eds.): Handbook of Research on ICT-Enabled Transformational Government: A Global Perspective, 2008.

Reich, J. (2008): Zu Choreographiemodellierung: Eine Übersicht. Informatik Spektrum, vol. 31:2, p. 360.

Reijers, H., van der Toorn, R. (2002): Integrating Business Process Reengineering with Application Development under Architecture. The 6th Pacific Asia Conference on Information Systems (PACIS). Tokyo, Japan 2002, pp. 659-673.

Rodon, J., Busquets, X., Christiaanse, E. (2005): Orchestration in ICT-enabled Business Networks: A Case in the Repairs Industry. 18th Bled eConference, eIntegration in Action. Bled, Slovenia.

Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel. D. (2005): Web Service Modeling Ontology. Applied Ontology, vol. 1:1, pp. 77 - 106.

Rosemann, M. (1996): Komplexitätsmanagement in Prozessmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden, Gabler.

Rosemann, M., zur Mühlen, M. (1998): Modellierung der Aufbauorganisation in Workflow-Management-Systemen: Kritische Bestandsaufnahme und Gestaltungsvorschläge. EMISA Forum, Mitteilungen der GI-Fachgruppe 2.5.2 "Entwicklungsmethoden für Informationssysteme und deren Anwendung", Bonn, vol. 7:1, pp. 78 - 86.

Ruggaber, R. (2006): ATHENA - Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications. In: Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N. (eds.): Interoperability of Enterprise Software and Applications. Springer, London, pp. 459-460.

Russel, N., van der Aalst, W. (2008): Work Distribution and Resource Management in BPEL4People - Capabilities and Opportunities. In: Bellahsène, Z., Léonard, M. (eds.): CAiSE 2008. LNCS 5074, Springer, Berlin et al., pp. 94–108.

Rüter, A., Schröder, J., Göldner, A. (2006): IT-Governance in der Praxis – Erfolgreiche Positionierung der IT im Unternehmen. Anleitung zur erfolgreichen Umsetzung regulatorischer und wettbewerbsbedingter Anforderungen. Springer, Berlin et al.

Sadiq, W., Orlowska, M. (2000): On Business Process Model Transformations. In: Laender, a., Liddle, S., Storey, V. (eds.): ER2000 Conference. LNCS 1920, Springer, Berlin et al., pp. 267-280.

Sandhu, R., Coyne, E., Feinstein, H., Youman, C. (1996): Role-Based Access Control Models. IEEE Computer 29 (2), pp. 38–47.

Sarshar, K., Theling, T. (2007): Makromodellierung kollaborativer Geschäftsprozesse. In: Loos, P., Vanderhaeghen, D. (eds.): Kollaboratives Prozessmanagement - Unterstützung kooperations- und koordinationsintensiver Geschäftsprozesse am Beispiel des Bauwesens. Logos, Berlin, pp. 83-104.

Scheer, A.-W. (1989): Enterprise-Wide Data Modelling. Springer, Berlin et al.

Scheer, A.-W. (1990): EDV-orientierte Betriebswirtschaftslehre. Grundlagen für ein effizientes Informationsmanagement. 4th Edition, Berlin et al.

Scheer, A.-W. (1992): Architecture of Integrated Information Systems - Foundations of Enterprise Modelling. Springer, Berlin.

Scheer, A.-W. (1994 CIM): CIM Computer Integrated Manufacturing – Towards the factory of the future. 3rd Edition, Springer, Berlin et al.

Scheer, A.-W. (1994): Business Process Engineering – Reference Models for Industrial Enterprises. 2nd Edition. Springer, Berlin et al.

Scheer, A.-W. (1996): Industrialisierung der Dienstleistung. Veröffentlichungen des Instituts für Wirtschaftsinformatik, vol. 122, Saarbrücken, Germany.

Scheer, A.-W. (1999): ARIS – Business Process Frameworks. 3rd Edition, Springer, Berlin et al.

Scheer, A.-W. (2000): ARIS – Business Process Modeling. 3rd Edition, Springer, Berlin et al.

Scheer, A.-W., Abolhassan, F., Jost, W., Kirchmer (2004): Business Process Automation – ARIS in practice. Springer, Berlin et al.

Scheer, A.-W., Jost, W. (2002): ARIS in der Praxis – Gestaltung, Implementierung und Optimierung von Geschäftsprozessen. Springer, Berlin et al.

Scheer, A.-W., Kruppke, K., Heib, R. (2003): E-Government - Prozessoptimierung in der öffentlichen Verwaltung. Springer, Berlin et al.

Schekkerman, J. (2004): How to survive in the jungle of Enterprise Architecture Frameworks. Trafford, Victoria.

Schmidt, D. (2006): Guest Editor's Introduction: Model-Driven Engineering. IEEE Computer, vol. 39:2, pp. 25-31.

Schmidt, D., Wimmer, M., Mondorf, A., Herborn, T., Dietrich, L., Matheis, T., Ziemann, J. (2007): Critical State of the Art Report on Interoperability in the Public Sector. Deliverable WP4-D2, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Schmidt, G. (1999): Informationsmanagement - Modelle, Methoden, Techniken. 2nd Edition, Springer, Berlin et al.

Schmidt, G. (2002): Prozessmanagement - Modelle und Methoden. 2nd Edition, Springer, Berlin et al.

Schmoll, G. (2001): Kooperationen, Joint Ventures und Allianzen. Deutscher Wirtschaftsdienst, Köln.

Schneckenbach, R., Zaier, A. (2003): Collaborative SCM in Branchen. Galileo Press, Bonn.

Schneider, C. (2007): eGovernment- Integration. Konzeption einer service-orientierten Integrationsarchitektur zur Digitalisierung von Verwaltungsprozessen. In: Thomas, R., Böhnlein, C. (eds.): Aspekte der Wirtschaftsinformatik und Logistik. Oxgen, Würzburg, Germany.

Schönherr, M. (2004): Enterprise Architecture Frameworks. In: Aier, S., Schönherr, M. (eds.): Enterprise Application Integration - Serviceorientierung und nachhaltige Architekturen. Gito, Berlin 2004, pp. 3-48.

Schroth, C. (2008): A Service-oriented Reference Architecture for Organizing Cross-Company Collaboration. In: Mertins, K., Ruggaber, R., Popplewell, K., Xu, X.: Enterprise Interoperability III. Springer, Berlin et al., pp. 71-83.

Schubert, P. (2008): Integrationsszenarien für Business Collaboration. In: Hofmann, J., Meier, A. (eds.): Webbasierte Geschäftsmodelle. HMD – Praxis der Wirtschaftsinformatik, vol. 261, pp. 32-41.

Schulz, K. (1998): Implementation of the distributed workflow service. Technical report, ESPRIT Vega, Project no. EP 20408.

Schulz, K. (2002): Modelling and Architecting of Cross-Organizational Workflows. PhD Thesis, University of Queensland.

Schulz, K., Orlowska, M. (2004): Facilitating Cross-Organizational Workflows with a Workflow View Approach. Data & Knowledge Engineering, Vol. 51, pp. 109-147.

Shan, Z., Yang, Y., Li, Q., Luo, Y., Peng, Z. (2006): A Light-Weighted Approach to Workflow View Implementation. In: Zhou, X., et al. (eds.): APWeb 2006. LNCS 3841, Springer, Berlin et al., pp. 1059 – 1070.

Shen, M., Liu, D. (2001): Coordinating Interorganizational Workflows Based on Process-Views. Proceedings of the 12th International Conference on Database and Expert Systems Applications. LNCS 2113, Springer, London, pp. 274-283.

Silberberger, H. (2003): Collaborative Business und Web Services - Ein Managementleitfaden in Zeiten technologischen Wandels. Springer, Berlin et al.

Simon, H. (1996): The Sciences of the Artificial. 3rd Edition, MIT Press, Cambridge, USA.

Sinz, E. (2001 Model): Modell. In: Mertens, P. (ed.): Lexikon der Wirtschaftsinformatik. 4th Edition, Springer, Berlin et al., pp. 311-312.

Sinz, E. (2001): Modellierung. In: Mertens, P. (ed.): Lexikon der Wirtschaftsinformatik. 4th Edition, Springer, Berlin et al., pp. 312-313.

Sinz, E. (2002): Architektur von Informationssystemen. In: Rechenberg, P., Pomberger, G. (eds.): Informatik-Handbuch. 3rd Edition, Hanser, München, pp. 1055-1068.

Smith, H., Fingar, P. (2002): Business Process Management - The Third Wave. Meghan-Kiffer, Florida.

Spadone, P. (2006): Collaborative workflow requirements for eGovernment. Deliverable WP5-D1, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Spath, D., Renner, T., Weisbecker, A. (2005): Unternehmensübergreifende Geschäftsprozesse und E-Collaboration. Real-Time Enterprise in der Praxis. Springer, Berlin et al., pp. 13-30.

Specht, T, Drawehn, J., Thränert, M., Kühne, S. (2005): Modeling Cooperative Business Processes and Transformation to a Service Oriented Architecture. Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), München, Germany.

Stachowiak, H. (1973): Allgemeine Modelltheorie. Springer, Berlin et al.

Stein, G. (1922): Sacred Emily. Geography and Plays. Four Seas Co., Boston, pp. 178-188.

Stein, S., Ivanov, K. (2007): EPK nach BPEL Transformation als Voraussetzung für praktische Umsetzung einer SOA. In: Bleek, W. Raasch, J., Züllighoven, H. (eds.): Software Engineering 2007. Lecture Notes in Informatics (LNI), vol. 105, Hamburg, Germany, pp. 75-80.

Störrle, H., Glock, W. (2007): Geschäftsprozessmodellierung für Service-Orientierte Architekturen. In: Nissen, V., Petsch, M., Schorscht, H. (eds.): Service-orientierte Architekturen - Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Gabler, Wiesbaden.

Strahringer, S. (1996). Metamodellierung als Instrument des Methodenvergleichs - Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker, Aachen.

Stuke, G. (1974): Zwischenbetriebliche Kooperation im Absatzbereich von Industrieunternehmen. Dissertation, University Mannheim.

Svirskas, A. (2007): Architecture of basic security and privacy mechanism. Deliverable WP6.D4, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Taylor, D. (1995): Business Engineering with Object Technology. John Wiley & Sons.

The Open Group (2003): TOGAF 8.1.1 Online. Http://www.opengroup.org/architecture/togaf8-doc/arch/, last accessed: March 2009.

Theling, T. (2008): Integrierte Informationssystemarchitekturen in dynamischen Kooperationen – Entwicklung und Implementierung einer Architektur zur Integration und Prozessunterstützung in dynamischen Unternehmensnetzwerken. Doctoral Thesis, University of Mainz, May 2008.

Theling, T., Loos, P. (2004): Determinanten und Formen von Unternehmenskooperationen. In: Loos, P. (ed.): Working Papers of the Research Group Information Systems & Management, Paper 18, Mainz.

Theling, T., Sarshar, K., Jerrentrup, M., Loos, P. (2005): PNML-notierte Objektpetrinetze zur Integration von Prozess- und Organisationsmodellen. In: R. Eckstein; R. Tolksdorf (eds.): Berliner XML Tage. Berlin, pp. 191-202.

Thomas, O. (2005): Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation. IWi - Veröffentlichungen des Instituts für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz, vol. 184, Saarbrücken, Germany.

Thomas, O. (2006): Das Referenzmodellverständnis in der Wirtschaftsinformatik - Historie, Literaturanalyse und Begriffsexplikation. IWi - Veröffentlichungen des Instituts für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz, vol. 187, Saarbrücken, Germany.

Thomas, O., Leyking, K., Dreifus, F. (2008): Using Process Models for the Design of Service-Oriented Architectures - Methodology and E-Commerce Case Study. 41st Hawaii International Conference on System Sciences (HICCS-41 2008), Hawaii.

UDDI Consortium (2001): UDDI Executive White Paper. Http://www.uddi.org/pubs/UDDI_Executive _White_Paper.pdf, last accessed: August 2008.

UN/CEFACT (2003): Core Components Technical Specification – Part 8 of the ebXML Framework. Version 2.01. Http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf, last accessed: October 2008.

Unitt, M., Jones, I. (1999): EDI — The Grand Daddy of Electronic Commerce. BT Technology Journal, vol. 17:3.

University Consortium for Geographic Information Science (1996): Research Priorities for Geographic Information Science. Cartography and Geographic Information Science, vol. 23:3, July 1996, pp. 115-127.

Van der Aalst, W. (1998). The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers, vol. 8:1, pp. 21-66.

Van der Aalst, W. (1999): Process oriented architectures for electronic commerce and interorganizational workflow. Information Systems, vol. 24:8, pp. 639-671.

Van der Aalst, W. (2002 B2B): Inheritance of Interorganizational Workflows to Enable Business-to-Business E-commerce. Electronic Commerce Research, vol. 2, pp. 195-231.

Van der Aalst, W. (2002): Making work flow: On the application of Petri Nets to Business Process Management. In: Esparza, J., Lakos, C. (eds.): Proceedings of the 23rd International Conference on Applications and Theory of Petri Nets. LNCS 2360, Springer, Berlin, pp. 1-22.

Van der Aalst, W. (2003): Don't go with the flow: Web services composition standards exposed. IEEE Intelligent Systems, vol. 18:1, pp. 72- 76.

Van der Aalst, W., Beisiegel, M., Van Hee, K., König, D., Stahl, C. (2007): An SOA-based architecture framework. International Journal of Business Process Integration and Management, vol. 2:2, pp. 91 – 101.

Van der Aalst, W., Dumas, M., ter Hofstede, A., Russell, N. (2005): Life After BPEL? Formal Techniques for Computer Systems and Business Processes. LNCS 3670, Springer, Berlin et al., pp. 35-50.

Van der Aalst, W., Kumar, A., Verbeek, H. (2003): Organizational modeling in UML and XML in the context of workflow systems. In: Caroll, J. (ed.): Symposium on Applied Computing (SAC) 2003. Melbourne, Florida, pp. 603-608.

Van der Aalst, W., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K. (2008): From Public Views to Private Views – Correctness-by-Design for Services. Web Services and Formal Methods. LNCS 4937, Springer, Berlin et al., pp. 139-153.

Van der Aalst, W., ter Hofstede, A., Weske, M. (2003): Business Process Management - A survey. Van der Aalst, W., ter Hofstede, A., Weske, M. (eds.): Proceedings of the Business Process Management International Conference - BPM 2003. LNCS 2678, Springer, Heidelberg, pp. 1-12.

Van der Aalst, W., van Hee, K. (2002): Workflow Management – Models, Methods and Systems. MIT Press, Cambridge.

Van der Aalst, W., Weske, M. (2001): The P2P Approach to Interorganizational Workflows. Proceedings of the 13th International Conference on Advanced Information Systems Engineering. LNCS 2068, Springer, Berlin et al., pp. 140-156.

Vanderhaeghen, V. (2009): Konstruktion einer Support-Plattform für das computergestützte Management dynamischer kollaborativer Geschäftsprozesse. Logos, Berlin.

Vernadat, F. (1997): Enterprise Modelling Languages. International Conference on Enterprise Integration and Modeling Technology 1997 (ICEIMT'97), Gaithersburg USA.

Vernadat, F., Kosanke, K. (1992): CIMOSA - A reference architecture for CIM. Proceedings of the IFIP TC5/WG5.3 eight international PROLAMAT conference on human aspects in computer integrated manufacturing. FIP Transactions, vol. B-3, North-Holland.

Voigtmann, P., Zeller, T. (2003): Beiträge zur Integrationsproblematik im Kontext von Electronic Business und Elektronischen Marktplätzen. In: Uhr, W., Esswein, W., Schoop, E. (eds.): Wirtschaftsinformatik - vol. 1. Physica, Heidelberg, pp. 215–237.

Völter, M. (2007): Services, Komponenten, Modelle. In: Starke, G., Tilkow, S. (eds.): SOA-Expertenwissen. Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt, Heidelberg, pp. 423-438.

Vom Brocke, J. (2008): Konstruktionstechniken zur Referenzmodellierung. Enzyklopädie der Wirtschaftsinformatik, Online Lexikon. Oldenbourg Wissenschaftsverlag, Germany.

Von Henning, H. (2007): Granularität von Services – Kriterien zur Strukturierung einer Servicelandschaft. In: Starke, G., Tilkov, S. (eds.): SOA-Expertenwissen – Methoden, Konzepte und Praxis serviceorientierter Architekturen. dpunkt, Heidelberg, pp. 343-356.

W3C (1999): XSL Transformations (XSLT) Version 1.0 - W3C Recommendation 16 November 1999. Http://www.w3.org/TR/1999/REC-xslt-19991116, last accessed: January 2009.

W3C (2001): Web Services Description Language (WSDL) 1.1 - W3C Note 15 March 2001. Http://www.w3.org/TR/2001/NOTE-wsdl-20010315, last accessed: January 2009.

W3C (2002): Web Service Choreography Interface 1.0 - W3C Note 8 August 2002. Http://www.w3.org/TR/2002/NOTE-wsci-20020808, last accessed: May 2008.

W3C (2004): Web Service Architecture – W3C Working Group Note 11 February 2004. Http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, last accessed: June 2008.

W3C (2005): Web Service Modeling Ontology (WSMO) - W3C Member Submission 3 June 2005. Http://www.w3.org/Submission/2005/SUBM-WSMO-20050603/, last accessed: January 2009.

W3C (2006): Web Services Choreography Description Language Primer - W3C Working Draft 19 June 2006. Http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619/, last accessed: July 2008.

W3C (2007 SOAP): SOAP Version 1.2 Part 0, Primer (Second Edition) - W3C Recommendation 27 April 2007. Http://www.w3.org/TR/2007/REC-soap12-part0-20070427/, last accessed: May 2009.

W3C (2007): XML Path Language (XPath) 2.0 - W3C Recommendation 23 January 2007. Http://www.w3.org/TR/2007/REC-xpath20-20070123/, last accessed: January 2009.

W3C (2008): Extensible Markup Language (XML) 1.0 (Fifth Edition) - W3C Recommendation 26 November 2008. Http://www.w3.org/TR/2008/REC-xml-20081126/, last accessed: May 2009.

W3C (2009): XML Schema Definition Language (XSD) 1.1 Part 1: Structures - W3C Working Draft 30 January 2009. Http://www.w3.org/TR/xmlschema11-1/, last accessed: May 2009.

Weber, I., Haller, J., Müller, J. A. (2006): Automated Derivation of Executable Business Processes from Choreographies in Virtual Organizations. In: F. Lehner, H. Nösekabel, P. Kleinschmidt (eds.): Multikonferenz Wirtschaftsinformatik 2006 (MKWI 2006), Band 2, XML4BPM Track, GITO, Berlin, pp. 313-328.

Weerakkody, V., Janssen M., Hjort-Madsen, K. (2007): Integration and Enterprise Architecture Challenges in E-Government - A European Practice. International Journal of Cases in Electronic Commerce, vol. 3:2, pp. 13 – 35.

Weigert, T., Weil, F., Marth, K., Baker, P., Jervis, C., Dietz, P., Gui, Y., van den Berg, A., Fleer, K., Nelson, D., Wells, M., Mastenbrook, B. (2007): Experiences in Deploying Model-Driven Engineering. In: Gaudin, E., Najm, E., Reed, R. (eds.): SDL 2007 - Design for Dependable Systems. LNCS 4745, Springer, Berlin et al., pp. 35–53.

Werth, D. (2005): E-Government Interoperability. In: Khosrow-Pour, M. (ed.): Encyclopedia of Information Science and Technology. Idea Group Inc., Hershey et al., pp. 985-989.

Werth, D. (2006): Kollaborative Geschäftsprozesse. Logos, Berlin.

Werth, D., Walter, P., Loos, P. (2007): Conceiving an Environment for Managing the Lifecycle of Collaborative Business Processes. In: Oberweis, A. et al. (eds.): eOrganisation: Service-, Prozess-, Market-Engineering. Proceedings of 8. Internationale Tagung WI. Vol. 1, Universitätsverlag Karlsruhe, pp. 805-822.

Weske, M. (2007): Business Process Management – Concepts, Languages, Architectures. Springer, Berlin et al.

WfMC (1999): Workflow Management Coalition Terminology & Glossary. Http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf, last accessed: June 2008.

WfMC (2008): Process Definition Interface - XML Process Definition Language - Version 2.1a. Workflow Management Coalition Workflow Standard.

Wilde, T., Hess, T. (2007): Forschungsmethoden in der Wirtschaftsinformatik – Eine empirische Untersuchung. Wirtschaftsinformatik, vol. 49:4, pp. 280-287.

Wimmer, M., Liehmann, M., Martin, B. (2006): Offene Standards und abgestimmte Spezifikationen - das österreichische Interoperabilitätskonzept. Multikonferenz Wirtschaftsinformatik (MKWI), Passau.

Winter, R., Fischer, R. (2007): Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. Journal of Enterprise Architecture, vol. 3:2, pp. 7-18.

Wöhe, G. (1996): Einführung in die allgemeine Betriebswirtschaftslehre. 19th Edition, Vahlen, München.

Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede, A. (2002): Pattern-Based Analysis of BPEL4WS. Report FIT-TR-2002-04, Queensland University of Technology, Brisbane, Australia.

Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede, A., Russell, N. (2006): On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J., Sheth, A. (eds.): BPM 2006. LNCS 4102, Springer, Berlin et al., pp. 161–176.

Woods, D., Mattern, T. (2006): Enterprise SOA – Designing IT for Business Innovation. O'Reilly, Tokyo.

Wutke, D., Martin, D., Leymann, F. (2008): Model and Infrastructure for Decentralized Workflow Enactment. 23rd ACM Symposium on Applied Computing (SAC2008), Fortaleza, Brasil.

Yu, X., Zhang, Y., Zhang, T., Wang, L., Zhao, J., Zheng, G., Li, X. (2007): Towards a Model Driven Approach to Automatic BPEL Generation. In: Akehurst, D., Vogel, R., Paige, R.(eds.): Model Driven Architecture - Foundations and Applications, LNCS 4530, Springer, Berlin et al., pp. 204-218.

Zachman, J. (1987): A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26:3, pp. 276-292.

Zaha, J., Dumas, M., ter Hofstede, A., Barros, A., Decker, G. (2006): Service Interaction Modeling - Bridging Global and Local Views. Enterprise Distributed Object Computing Conference (EDOC '06), 10th IEEE International, pp. 45-55.

Zdravkovic, J., Kabilan, V. (2005): Enabling Business Process Interoperability Using Contract Workflow Models. On the Move to Meaningful Internet Systems 2005, LNCS 3670, Springer, Berlin et al., pp. 77-93.

Zelewski, S. (1987): Schnittstellen bei betrieblichen Informationssystemen. Betriebswirtschaftliche Forschung und Praxis, vol. 39, pp. 550-561.

Zentes, J., Swoboda, B., Morschett, D. (2005): Kooperation, Allianzen, Netzwerke – Entwicklung der Forschung und Kurzabriss. In: Zentes, J., Swoboda, B., Morschett, D. (eds.): Kooperationen, Allianzen, Netzwerke – Grundlagen, Ansätze, Perspektiven. 2nd Edition, Gabler, Wiesbaden, pp. 3-32.

Ziemann, J., Kahl, T., Matheis, T. (2007): Cross-organizational processes in Public Administrations - Conceptual modeling and implementation with Web Service Protocols. In: Oberweis, A., Weinhardt, C., Gimpel, H., Koschmider, A., Pankratius, V, Schnizler, B. (eds.): eOrganisation: Service-, Prozess-, Market-Engineering. 8. Internationale Tagung Wirtschaftsinformatik, Universitätsverlag, Karlsruhe, pp. 181-98.

Ziemann, J., Kahl, T., Werth, D. (2007): Using View Process Models in Collaborative Business Processes. In: Putnik, G., Cunha, M. (eds.): Encyclopedia of Networked and Virtual Organizations, Idea Group Inc., Hershey et al.

Ziemann, J., Leyking, K., Kahl, T., Werth, D. (2006): Enterprise Model driven Migration from Legacy to SOA. In Gimnich, R., Winter, A. (eds.): Workshop Software-Reengineering und Services, Multikonferenz Wirtschaftsinformatik, Passau.

Ziemann, J., Loos, P. (2008): Transforming Cross-organizational Processes between European Administrations – Towards a Comprehensive Business Interoperability Interface. In: Weerakkody, V., Janssen, M., Dwivedi, Y. (eds.): Handbook of Research on ICT-Enabled Transformational Government: A Global Perspective, 2008.

Ziemann, J., Matheis, T., Freiheit, J. (2007): Modelling of Cross-Organizational Business Processes - Current Methods and Standards. Enterprise Modelling and Information Systems Architectures, vol. 2:2, p. 23-31. Also published in: Manfred Reichert, Stefan Strecker, Klaus Turowski (2007, eds.): Enterprise Modelling and Information Systems Architectures - Concepts and Applications. Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07), St. Goar, Germany.

Ziemann, J., Matheis, T., Schmidt, D., Mohndorf, A. (2008): R4eGov IOP Architecture and IOP Lifecycle (Version 1). Deliverable WP4-D7, R4eGov - Towards e-Administration in the large, project number IST-2004-026650.

Ziemann, J., Matheis, T., Werth, D. (2008): Conceiving interoperability between public authorities - a methodical framework. 41st Hawaii International Conference on System Sciences (HICCS-41 2008), Hawaii.

Ziemann, J., Mendling, J. (2005): Transformation of EPCs to BPEL – A pragmatic approach. 7th International Conference on the Modern Information Technology in the Innovation Processes of the industrial enterprises, Genoa.

Ziemann, J., Ohren, O., Jaekel, F., Kahl, T., Knothe, T. (2007): Achieving Enterprise Model Interoperability Applying a Common Enterprise Metamodel. In: Doumeingts, G., Müller, J., Morel, G., Vallespir, B. (eds.): Enterprise Interoperability – New Challenges and Approaches. Springer, London, pp. 199-208.

Zimmermann, O., Krogdahl, P., Gee, C. (2004): Elements of Service-Oriented Analysis and Design - An interdisciplinary modeling approach for SOA projects. IBM Developer Works, http://www-128.ibm.com/developerworks/webservices/library/ws-soad1/, last accessed: February 2009.

Zouggar, N, Chen, D., Vallespir, B. (2008): Enterprise modelling and Ontology. Proceedings of the 17th IFAC World Congress. Seoul, Korea.