A Model Driven Approach for Open Distributed Systems using an Enterprise Architecture Framework

Melanie Langermeier[1,2,3], Arne-Jørgen Berre[4] and Bernhard Bauer[1]
[1]*Elite Graduate Program Software Engineering, University of Augsburg, Augsburg, Germany*
[2]*Ludwigs-Maximilian-Universität, München, Germany*
[3]*Technische Universität München, München, Germany*
[4]*SINTEF ICT, Oslo, Norway*

Keywords: Open Distributed Systems, UML, BPMN, Enterprise Architecture Management, Cyber Physical Systems.

Abstract: Open distributed systems are complex systems, which contain a lot of components provided by different vendors and built with different technologies. The use of well-established and standardized modeling techniques is one way to deal with the problems that occur in the specification and development process of these systems. Enterprise Architecture Frameworks provide a good foundation to structure the various required modeling techniques. Existing modeling solutions in the context of Enterprise Architecture Frameworks do not provide optimal support for the specification of open distributed systems. In this paper a coherent MODeling approach for open distributed systems using an Enterprise Architecture framework (MODEA) is developed. The approach uses the latest OMG modeling standards. As an enterprise architecture framework the Reference Model for Open Distributed Systems (RM-ODP) has been chosen. Finally MODEA is illustrated through the specification of two case studies.

1 INTRODUCTION

"An Open Distributed System is made up of components that may be obtained from a number of different sources, which together work as a single distributed system" (Crowcroft, 1996). Thereby the components are provided from several vendors and the system compasses heterogeneous IT resources and multiple domains. Such distributed, open systems are significantly more complex than closed, centralized systems. An increasing scope leads to an increasing number of involved people and components and often concludes in a complex definition of the system and communication problems (Leist and Zellner, 2006). Open distributed systems become important because of an increasing demand on information exchange between cooperating organizations and a growing need of interconnect information processing services to provide the required functionally (ISO/IEC, 2010b). Examples for open distributed systems are Information Systems in the business domain but also Cyber Physical Systems in the embedded context.

In open distributed systems a lot of different parties have to work together. The issue that "each domain has its own description technique" (Lankhorst, 2009) and following "different fields speak their own languages, draw their own models, and use their own techniques and tools" (Lankhorst, 2009) affects the current practice in architecture specification. To avoid a Babylonian confusion it is important that the various vendors of the open distributed system agree on one language used for the specification and documentation of the system (ISO/IEC, 1998a).

The problems in open distributed systems are similar with those in Enterprise Architecture where software systems of different departments have to work together to achieve to enterprises' vision. Therefore we apply the ideas of EA to open distributed systems.

The purpose of this paper is to create a coherent model driven approach based on existing and well-established modeling techniques to cope with the growing complexity and scope of information system (Leist and Zellner, 2006). To support the development and specification process of open distributed systems metamodel-based modeling techniques and a well-defined framework is used as basis.

First we describe the requirements and existing modeling approaches for architecture frameworks. Then the viewpoints in MODEA are described as well as the relationships between them. At least

Langermeier M., Berre A. and Bauer B..
A Model Driven Approach for Open Distributed Systems using an Enterprise Architecture Framework.
DOI: 10.5220/0004438502840291
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 284-291
ISBN: 978-989-8565-61-7

the defined approach is applied to two environmental systems in the cyber physical context for evaluation purposes. Based on these experiences we compare MODEA with the existing approaches. This paper is a short form of the technical report (Langermeier, 2013). More details about the presented approach can be found there.

2 REQUIREMENTS

In (Langermeier, 2013) several problems in open distributed systems were identified based on literature and the experiences within the case studies. Afterwards requirements for a model driven approach were specified which support in coping with these problems. These requirements are presented briefly in the following.

A common framework applied with a shared set of modeling techniques, which is used by all participants, provides support for the problems occurring during the development of open distributed systems. Such a common framework should be divided into several viewpoints (Romero et al., 2012) and make use of metamodel-based modeling techniques, which are widely accepted and used. The established diagrams should be easy to understand but also powerful in their expressions. The use of standards as well as a sound tool support makes the approach complete. When using several viewpoints it is important that relationships between them are clearly defined to keep the overall model consistent. In the best case the tool also enables model-to-model transformations and model-to-code transformation to support the development process.

Use Cases are often used for determining the requirements of a software system (Anda et al., 2001) and should be integrated in a coherent modeling approach. To ensure that the system provides the right functionality the specification of the motivation and of the requirement is important (Engelsman et al., 2011).

The single components in an open distributed system have to be integrated and composed together to a whole system. A modeling approach for open distributed systems should support both the decomposition and composition concept. For an adequate specification of them techniques have to be defined to make the interface and behavior of the components as well as the interaction between them explicit (Crowcroft, 1996). To capture the variety the modeling approach has to support different architectural styles and make the differences between them visible. Moreover the approach should provide techniques to assign responsibilities to certain architectural elements (ISO/IEC, 1998a).

To support the interoperability of the components and to gain a flexible architecture a service oriented architectural style has to be supported by the modeling approach (Khoshnevis et al., 2009). The service-concept enables a common language between the collaborators and an adequate separation of concerns. The approach should allow the specification, identification and classification of service as well as the reuse of them. Furthermore the approach should support distribution transparencies. That means that the approach has to provide mechanisms to explicitly specify the way the distribution is provided. But also enable the abstraction of those for example in the case when the logical functionality is described (ISO/IEC, 1998a; Coulouris et al., 2005).

3 ARCHITECTURE-RELATED MODELING APPROACHES

An architecture description facilitates a good communication in the development projects by providing a common understanding. Thereby the focus lies on the main part of the system and its structure; irrelevant details are hidden (Lankhorst, 2009). The systematic approach for the establishment of the architecture is called an architecture framework. Usually a framework contains different viewpoints on a system to enable the description of different perspectives (Tang et al., 2004). Frameworks exist on the software level as well as on the enterprise level. Through a coherent description of the different parts of the enterprise architecture, it becomes an instrument in controlling the complexity of enterprises and its processes and systems (Lankhorst, 2009).

Established architecture frameworks are for example Zachman (Zachman, 1987), TOGAF (The Open Group, 2011), DoDAF in the military domain (U.S. Department of Defense, 2010), RM-ODP for open distributed systems (ISO/IEC, 1998a), the FEAF for US federal agencies and other governmental agencies (U.S. Executive Office, 2012) or the 4+1 View model of Architecture (Kruchten, 1995; Tang et al., 2004). In the following three modeling approaches related to enterprise architectures are presented:

The **Unified profile for DoDAF/MODAF (UPDM)** is a standardized UML profile that supports both EAF of the military domain. It includes a service-oriented component modeling with SoaML and a SysML system modeling to represent the DoDAF and MODAF architecture views (OMG, 2012b).

UML4ODP is the short name for ISO/IEC 19793, which defines a UML profile to model the five viewpoints of RM ODP (ISO/IEC, 2009). UML4ODP is a very well specified modeling approach and built completely upon the formal RM-ODP specification. But this results in over-weighted diagrams at some points.

ArchiMate is a modeling approach for TOGAF. Its goal is to "provide a graphical language for the representation of enterprise architectures over time" (The Open Group, 2012). ArchiMate is a well-connected and well-understandable modeling approach. But main purpose of it is visualization and it is not intended to be used as a language for a model driven development with model transformation and code generation.

All these approaches define their own modeling language or UML profile for the architecture description. Although they all enable the specification of architectures with a service-oriented style, some required concepts like identification of services, support for reuse or classification are not well supported. Additionally especially UML4ODP and UPDM are very extensive approaches, which are often too complex and over-weighted. Furthermore UPDM contains specific aspects for the military domain. The main problem with ArchiMate is, that it is only created for visualizing architectures, and not for model driven development. None of the three modeling approaches fulfills all of the requirements in the previous chapter (Langermeier, 2013).

Furthermore in the CEN/TR 15449-4 standard, which is under approval at the moment, recommendations are provided how to use techniques and tools to specify a service-centric specification in the context of spatial data infrastructures. Thereby it uses RM ODP for structuring (CEN/TR, 2013). The methods described in CEN/TR 15449-4 are also adapted in the working draft for an update of ISO 19119 (ISO 19119, 2013).

4 MODEA

MODEA is an acronym for **MOD**el driven **E**nterprise **A**rchitecture. The RM-ODP framework, an ISO/IEC standard in cooperation with ITU-T, is used as underlying Enterprise Architecture Framework for it. It is a framework for specifying and building large or complex systems. The systems dealt within RM ODP are Open Distributed Processing systems and they can be amongst others classical IT systems, information systems, embedded systems or business systems (ISO/IEC, 1998a; ISO/IEC, 2010a; ISO/IEC, 2010b; ISO/IEC, 1998b).

RM ODP was chosen for several reasons: The single viewpoints are well-defined using a set of formal concepts as foundation and a specific language for each viewpoint on top of them. The connections between the viewpoints are defined with little overlap. These characteristics make RM-ODP a good choice for our modeling approach (ISO/IEC, 1998a). Another reason is the domain independency of the framework and additionally its support for high heterogeneous systems. The framework is already used in the current deliverables of the two pilot cases introduced in chapter 1. The particular use of RM-ODP in the two projects in illustrated in (Langermeier, 2013).

4.1 Overview

The concepts described in the RM ODP standard will be modeled using UML, SoaML, BMM and BPMN. This follows the recommendation that is given in CEN/TR 15449-4 (CEN/TR, 2013).

UML is chosen, since it "is the most frequently used language for visualizing static and dynamic aspects of software–intensive systems" (Brown, 2008). But UML lacks an overarching concept of how to link the various single diagrams. In MODEA such a linking between the different diagrams with respect to the specification of the RM ODP framework is made. With the integration of UML into the context of architectural modeling, its strong support in tools and industry, should be earned in the specification of open distributed systems too.

A service-oriented style addresses several of the identified problems in open distributed systems. The Service oriented architecture Modeling Language (SoaML) is an OMG standardized metamodel and UML profile that enables the identification and specification of services as well as defining service consumers and producers. SoaML enables a linking of services to model elements of the OMG Business Motivation Model, UML Use Cases and also to process notations (OMG, 2012a).

The Business Motivation Model BMM is a business modeling specification, published by the OMG group, to define the motivation e.g. "to be able to say why" (OMG, 2010) for certain business activity. One advantage of BMM is the very simple definition, since the few concepts only have basic attributes and most of the associations are unconstrained (OMG, 2010). A lot of modeling tools provide support for requirements modeling using the BMM standard and enable the definition of links between the elements of BMM and thus of UML and BPMN.

At least Collaboration and Process Diagrams from the Business Process Modeling Notation (BPMN) are chosen as technique for defining business processes.

This notation is supported in nearly all modeling-tools, which provide support for UML 2. Furthermore BPMN has two more advantages: The notation is well known in all domains, from the business users to the technical developers and it closes the gap between business process design and its implementation (OMG, 2011a).

The meta models of each viewpoint in MODEA as well as an overarching one are shown in the following. The concepts of RM ODP serve as orientation for what to model in which viewpoint. This paper focuses on how to link the different modeling concepts of UML, BMM, BPMN and SoaML in the context of an architecture framework. Further details about the derivation of these from the RM ODP concepts can be found in (Langermeier, 2013).

4.2 Enterprise Viewpoint

The Enterprise Viewpoint (EntV) in RM-ODP answers questions about the purpose, the business requirements and the key stakeholders with their interactions (Linington et al., 2011). The purpose and motivation for the system is shown in a Business Motivation Model. The requirements of the system are captured in Use Case templates and their corresponding Business Processes are described using BPMN. The stakeholders and their interactions with the system are shown in an UML Use Case Diagram.

Figure 1 represents a meta model of the elements and diagrams used in the EntV and how they are connected to each other. From the BMM the higher level concepts like Vision, Goals, Mission, Strategy and Business Policy are not shown, because only the low-level concepts, which implement the higher level ones have links to elements from the other diagrams.

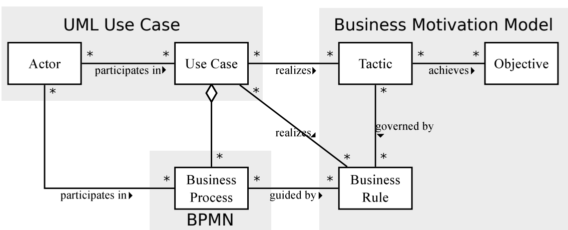

Figure 1: Concepts used in the Enterprise Viewpoint.

In MODEA the tactics defined in the BMM are realized by Use Cases. A business rule can also be realized directly in a Use Case or only indirectly. In the last case no explicit relationships to other model elements are possible. If a business rule is realized in a Use Case it also has effects on the business process describing this Use Case. Use Cases, as representations of interactions between the stakeholders, are summarized in a UML Use Case Diagram. The role a stakeholder plays in a specific use case is represented as Actor. The behavior of a use case is defined through a Business Process using BPMN. The pools in a Business Process represent the Actors that participate in the described Use Case. Furthermore each use case is linked to use case descriptions containing detailed requirements, either informally and unstructured or in a formal style or with pseudo code.

4.3 Information Viewpoint

The Information Viewpoint (IV) in RM-ODP answers questions about the handled data types as well as their states and relationships. Furthermore it deals with the actions in the system and their dependencies to the data (Linington et al., 2011). In MODEA the data types with their relationships to each other are described using an UML Class diagram, instances of the data types are described using an UML Object Diagram. At least the state of the data and allowable actions are described using UML state machines and SoaML Message Types. Figure 2 shows the used diagrams in the EntV and how its elements are connected to each other.

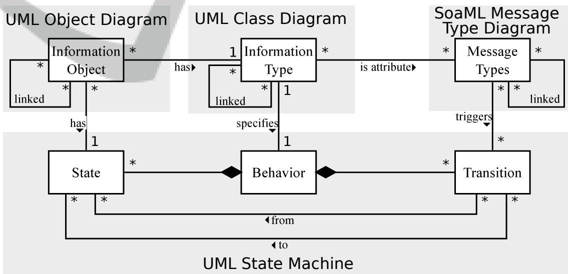

Figure 2: Concepts used in the Information Viewpoint.

Each information object has a specific type and a specific state. The type is represented through instantiation of the object with the corresponding class. The state is represented through an attribute. The information type specifies the possible state changes of an information object through a linked UML State Machine. The outgoing transitions in one state represent the exits that could be used to change the state. Special message types can trigger one or more transitions.

4.4 Computational Viewpoint

The Computational Viewpoint (CV) in RM-ODP answers questions about the basic functionality and its provisioning via service as well as the internal realization of service with components and connectors (Linington et al., 2011). In MODEA the system at a high level is specified using a Service Architecture. Through participants requesting and providing services the functionality is described. A stepwise refinement of service contracts and participants as well as Sequence and Process Diagrams describe the internal behavior. Therewith the SoaML service-contract based approach to specify services is used (Elvesæter et al., 2011). Figure 3 shows the meta-model for MODEA and is a simplified meta model of the used part of SoaML.

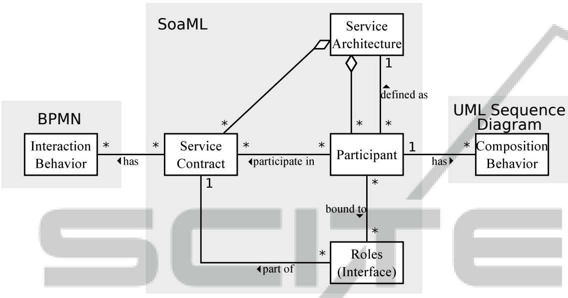

Figure 3: Concepts used in the Computational Viewpoint.

The overall system is described as Service Architectures, which contains Participants and the Service Contracts between them. The Service Contracts specify the roles that can be played within the service. A role is represented as interface and can be either a provider or a consumer and a participant will be bound to this role when interacting with the service. An UML Sequence Diagram defines the interaction behavior of the participating parties in a service. Thereby the lifelines represent the roles.

A Service Contract can also be composed of other Service Contracts. These nested Service Contracts provide a more fine-granular description of the service. Such a compound service does not represent an implementation through calling other services. This can be specified in a Participant Architecture or Process Diagram. The use of the contract-based approach for SoaML enables the definition and usage of patterns (OMG, 2012a).

The participants can be further described through a service architecture defining how the services are provided through internal parts or use of other services. Another possibility to describe the internal behavior of a participant is through the use of a process diagram like BPMN. Here the composition of required services and required internal actions are defined to describe how the functionality is provided. The mapping of the participants to system components and the specification of the provider and consumer interfaces are done in the Engineering Viewpoint.

4.5 Engineering Viewpoint

The Engineering Viewpoint (EngV) in RM-ODP answers questions about how distribution is realized and how nodes and linking channels are structured. It describes how the interaction between the objects defined in the CV is achieved and what resources are required for this, for example discovery services or request brokers (ISO/IEC, 1998a; Linington et al., 2011). In MODEA the essential element is the component diagram, which describes the system distribution using components. The communication channels between the various components are modeled through the use of ports and assembly connectors, linking required and provided interfaces. Figure 4 shows the meta model of the EngV in MODEA.

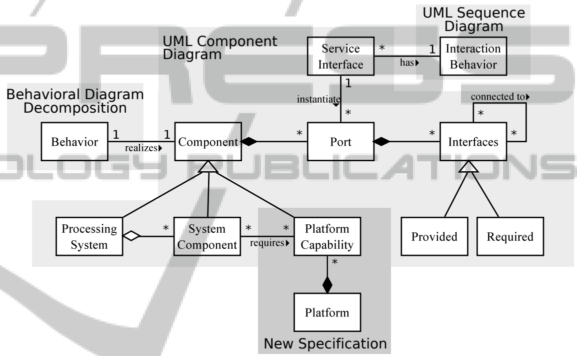

Figure 4: Concepts used in the Engineering Viewpoint.

Components can represent Processing Systems, System Components and Platform Capabilities from external Platforms. System components provide application functionality but also necessary functionality to support the distribution or communication in the ODP system. For example the replication transparency can be modeled with distributing the replicated system component on two processing systems. The hierarchical structure of the components is represented through a composition of them. Components are connected through ports and interfaces as well as realization and usage dependencies. The component diagram enables the specification of logical components but also physical components. Following a link to the logical description of the system as well as the deployment can be made (OMG, 2011b). The internal behavior of a component can be modeled in two ways. Either it is described using a behavioral diagram like State Machines or a Process Diagrams or the structure is further refined through a decomposition of the component. The service contracts of the CV are defined more technically in the EngV using SoaML Service Interfaces or simple UML Interfaces in case of a one-way service.

4.6 Technology Viewpoint

The Technology Viewpoint (TV) provides the link between the other four viewpoints and the real implementation. It describes the hard- and software components as well as possible implemented standards of the technology components (ISO/IEC, 1998a). In RM ODP this is mainly done by structuring and linking 'Technology Objects'. In MODEA we use the UML Deployment Diagram as shown in Figure 5.

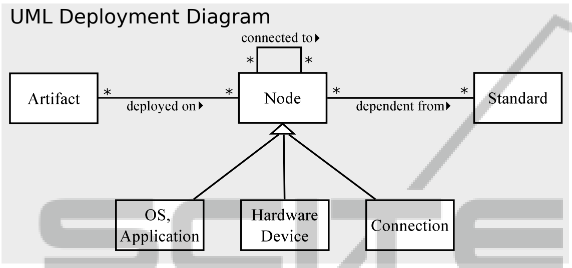

Figure 5: Concepts used in the Technology Viewpoint.

Each system component, defined in the EngV, is manifested through an Artifact. Afterwards Artifacts are deployed to Nodes, e.g. Applications, Hardware Devices or Communication connections. Standards are represented as artifacts, since they are "concrete elements in the physical world that are the result of a development process" (OMG, 2011b). A dependency-connector links the standards to the appropriate nodes.

Despite the Technology Objects and implementable standards RM ODP also provides concepts for describing the implementation process and extra information for conformance testing (IXIT). The concepts related to conformance testing are not examined in this paper. The implementation process can be described using BPMN.

4.7 Relationships between the Viewpoints

Figure 6 shows the five different viewpoints with their relationships between each other. For a better readability the labels between the elements of one viewpoint are invisible.

Each use case of the EntV, realizing one or more tactics, will have a corresponding service in the CV. The participants providing or requesting service are in the first step derived from the actors and are then further refined through decomposition. Such a participant, a logical element, can be then realized by a component, a physical element, in the EngV. A service, defined by a service contract, fulfills a service interface of a specific port in the EngV. This relationship between the EntV, CV and EngV is illustrated by an example in figure 7.

The components of the EngV are at least mapped to processing nodes in the TV using manifestation and UML Artifacts.

The IV provides a common set of information types and actions as well as constraints on those. All the other viewpoints have to be consistent with this definition and use these actions to specify interfaces or information types for information flows.

5 EVALUATION

We applied MODEA in two pilot cases. The first case study is the Personal Environmental Information System (PEIS) from the ENVIROFI Project[1]. The second one is the Oil Spill pilot from the ENVISION Project[2]. Both are large, distributed systems where sensor data, processing services and user interaction services have to be integrated with as much reuse as possible. Figure 7 shows excerpts of the Oil Spill pilot from the EntV to the EngV illustrating the context of the Cod Effects Prediction. For space reasons we are not able to present the full case studies in the paper, they can be found in (Langermeier, 2013).

For modeling we used Sparx System Enterprise Architect. At least each tool that provides support for UML 2.0 and BPMN can be used for MODEA. SoaML is an UML profile and if a modeling tool does not provide a direct support, it can be easily integrated through the use of the appropriate stereotypes. Also BMM, although there is no UML profile, can be easily used through stereotyping, since there are only a few simple concepts and associations.

Additionally to the practical application we compared MODEA with the modeling approaches UPDM, UML4ODP and ArchiMate based on the defined requirements. Especially in the fields where the three other frameworks are weak MODEA contains concepts to deal with them. These are the use of existing modeling language standard, the integration of use cases as well as the possibility of specifying distribution transparencies. In the case of support for model transformation and code generation, MODEA is on the same stage as UPDM and UML4ODP. All approaches except ArchiMate use metamodel based modeling techniques.

[1]Environmental Observation Web and its Service Applications within the Future Internet: www.envirofi.eu

[2]Environmental Services Infrastructure with Ontologies: www.envision-project.eu

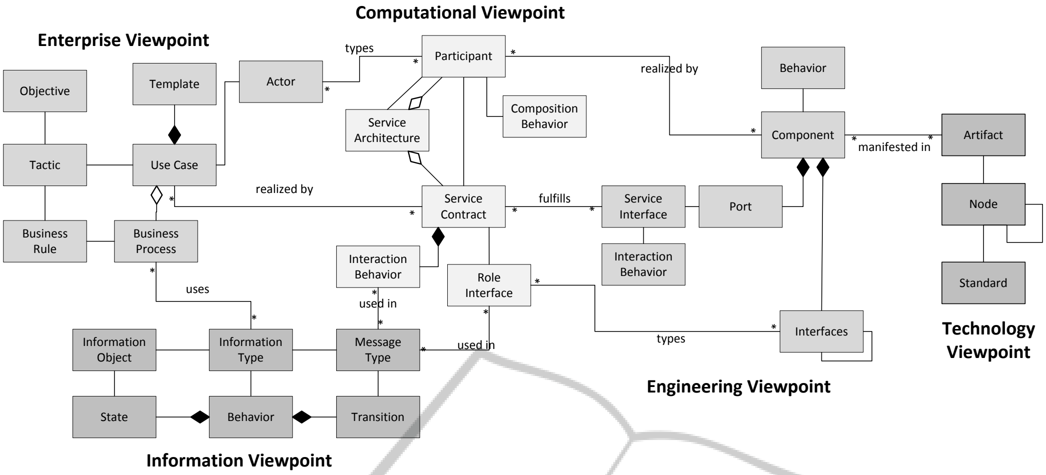

Figure 6: Meta-model of the relationships between the five Viewpoints.

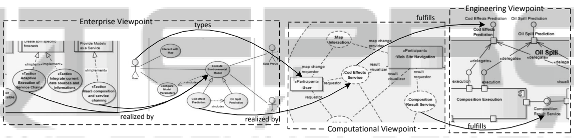

Figure 7: Excerpts of the Oil Spill Pilot from the Enterprise to the Engineering Viewpoint.

MODEA has a strong support for service-oriented architectures since SoaML provides support for identification, reuse and the specification of services. This is one point where existing approaches often do not provide appropriate techniques. With BPMN processes and collaborations the composition behavior in terms of orchestration and choreography can be described. Especially ArchiMate and UML4ODP lack techniques for choreography and orchestration.

For a full support of different architectural styles in MODEA UML lacks methods for defining communication details between components. For example it is not possible to differ between a rest-oriented communication, a stream-oriented communication, an event-oriented communication or a simple request and reply.

With UML, BPMN, SoaML and BMM widely accepted, standardized, meta-model based languages are chosen for MODEA. The tool variety for these languages supports the creation of readable diagrams, while they keep powerful in their expression. This point distinguishes MODEA from the three other approaches since we do not use some profile or a domain specific language, which are typically specialized and only little in use. A detailed listing of the requirements and their degree of fulfillment in the four modeling approaches can be seen in (Langermeier, 2013).

6 CONCLUSIONS

MODEA is a coherent model driven approach for specifying and designing enterprise architectures, especially those of open distributed systems. It is built upon the enterprise architecture framework RM ODP, which provides a sound basis of how to structure the overall specification. The concepts of the framework are modeled using the OMG standard modeling languages UML, BPMN, SoaML and BMM and with the integration of a service-oriented architectural style. These techniques provide a common foundation to cope with the heterogeneity and complexity and enable an effective collaboration between the various vendors and a flexible, optimized architecture. Although there is a quite good tool support for modeling, further work is required to enhance the support for model transformation and code generation. In the presented approach a first mapping of concepts from the different viewpoints to each other and possibilities of how to derive one model out of another are introduced. Further work has to be done to include all specification elements of the used techniques in the MODEA meta model and to define formal rules for model transformation. Finally, concepts have to be specified of how to generate code out of these models.

ACKNOWLEDGEMENTS

This work was partially sponsered by FuE-Programm Informations- und Kommunikationstechnik Bayern.

REFERENCES

Anda, B., Sjøberg, D., and Jørgensen, M. (2001). Quality and understandability of use case models. In Knudsen, J. L., editor, *ECOOP 2001 Object-Oriented Programming*, number 2072 in Lecture Notes in Computer Science, pages 402–428. Springer Berlin Heidelberg.

Brown, A. W. (2008). MDA redux: Practical realization of model driven architecture. In *7th International Conference on Composition-Based Software Systems*, pages 174–183.

CEN/TR (2013). CEN/TR 15449-4: Geographic information – Spatial Data Infrastructure (SDI) – Service centric view. Standard under approval, expected date available 2013-01.

Coulouris, G., Dollimore, J., and Kindberg, J. (2005). *Distributed Systems, Concepts and Design*. Addison Wesley, 4. rev. edition edition.

Crowcroft, J. (1996). *Open Distributed Systems*. Artech House.

Elvesæter, B., Berre, A.-J., and Sadovykh, A. (2011). Specifying Services using the Service oriented architecture Modeling Language (SoaML): A baseline for Specification of Cloud-based Services. In *CLOSER 2011*, pages 276–285. SciTePress.

Engelsman, W., Quartel, D., Jonkers, H., and Sinderen, M. v. (2011). Extending enterprise architecture modelling with business goals and requirements. *Enterprise Information Systems*, 4(3):9–36.

ISO 19119 (2013). ISO 19119: Geographic information Service. Working draft for international standard.

ISO/IEC (1998a). ITU-T x.901 | ISO/IEC 10746-1: Information technology - open distributed processing - reference model: Overview. International Standard V1.

ISO/IEC (1998b). ITU-T x.901 | ISO/IEC 10746-4: Information technology - open distributed processing - reference model: Architectural semantics. International Standard V1.

ISO/IEC (2009). ITU-T recommendation x.906 | ISO/IEC 19793: Information technology open distributed processing use of UML for ODP system specifications. International standard.

ISO/IEC (2010a). ITU-T x.901 | ISO/IEC 10746-2: Information technology - open distributed processing - reference model: Foundations. Final Draft International Standard V01.04.

ISO/IEC (2010b). ITU-T x.901 | ISO/IEC 10746-3: Information technology - open distributed processing - reference model: Architecture. Final Draft International Standard V01.04.

Khoshnevis, S., Shams Aliee, F., and Jamshidi, P. (2009). Model driven approach to service oriented enterprise architecture. In *IEEE Asia-Pacific Services Computing Conference*, pages 279–286.

Kruchten, P. (1995). The 4+1 view model of architecture. *IEEE Software*, 12(6):42–50.

Langermeier, M. (2013). A model-driven approach for open distributed systems. Technical Report 2013-03, University of Augsburg.

Lankhorst, M. (2009). *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, 2 edition.

Leist, S. and Zellner, G. (2006). Evaluation of current architecture frameworks. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1546–1553. ACM Press.

Linington, P. F., Milosevic, Z., Tanaka, A., and Vallecillo, A. (2011). *Building enterprise systems with ODP an introduction to Open Distributed Processing*. CRC Press, Boca Raton, FL.

OMG (2010). Business motivation model. OMG Specification V 1.1.

OMG (2011a). Business process modeling notation. OMG Specification V 2.0.

OMG (2011b). Unified modeling language superstructure. OMG Specification V 2.4.1.

OMG (2012a). Service oriented architecture modeling language. OMG Specification V 1.0.

OMG (2012b). Unified profile for DoDAF and MODAF (UPDM). OMG Specification V 2.0.

Romero, J. R., Jan, J. I., and Vallecillo, A. (2012). A tool for the model-based specification of open distributed systems. *The Computer Journal*.

Tang, A., Han, J., and Chen, P. (2004). A comparative analysis of architecture frameworks. In *11th Asia-Pacific Software Engineering Conference*, pages 640–647. IEEE.

The Open Group (2011). *TOGAF Version 9.1*. Van Haren Publishing.

The Open Group (2012). *ArchiMate 2.0 specification : Open Group Standard*. Van Haren Publishing.

U.S. Department of Defense (2010). The dodaf architecture framework version 2.02. in: http://dodcio.defense.gov/dodaf20.aspx, accessed 09/04/2013.

U.S. Executive Office (2012). Federal enterprise architecture. in: http://www.whitehouse.gov/omb/e-gov/fea/, accessed 09/04/2013.

Zachman, J. A. (1987). A framework for information architecture. *IBM System Journal*, 26(3):276–295.