Report from Dagstuhl Seminar 12131

Open Models as a Foundation of Future Enterprise Systems

Edited by

Robert B. France[1], Ulrich Frank[2], Andreas Oberweis[3], Matti Rossi[4], and Stefan Strecker[5]

1 **Colorado State University, US,** france@cs.colostate.edu
2 **Universität Duisburg-Essen, DE,** ulrich.frank@uni-due.de
3 **KIT – Karlsruhe Institute of Technology, DE,** oberweis@kit.edu
4 **Aalto University, FI,** matti.rossi@hse.fi
5 **FernUniversität in Hagen, DE,** stefan.strecker@fernuni-hagen.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12131 "Open Models as a Foundation of Future Enterprise Systems". Research on open models introduces a new model of collaboration among researchers, developers, and prospective users of reference enterprise models—leading to the prospect of shaping future enterprise systems. This seminar brought together researchers and practitioners with expertise in a broad range of fields including conceptual modelling, model-driven engineering, enterprise systems, software architectures, and modelling tool development. The seminar mixed short presentations on the attendees' perspectives on open models with keynote presentations and working groups on selected research issues. Topics discussed include the shape of future enterprise systems amalgamated with open reference enterprise models, business domains to be addressed in first open models, requirements towards a technical infrastructure as well as organisational issues of open model initiatives. The seminar's discussions benefitted from the different perspectives of attendees on the common topic, raised important new questions on open models, and brought to light overlooked aspects important to future research activities.

Seminar 25.–30. March, 2012 – www.dagstuhl.de/12131

1998 ACM Subject Classification D.2 Software Engineering, D.2.2 Design Tools and Techniques, D.2.9 Management, D.2.10 Design, D.2.11 Software Architectures, D.2.13 Reusable Software, H.1 Models and Principles, H.4 Information Systems Applications

Keywords and phrases Enterprise Modelling, Enterprise Systems, Reference Model, Meta Modeling, Method Engineering, Information Systems Architectures

Digital Object Identifier 10.4230/DagRep.2.3.67

Open Models as a Foundation of Future Enterprise Systems, *Dagstuhl Reports*, Vol. 2, Issue 3, pp. 67–85
Editors: Robert B. France, Ulrich Frank, Andreas Oberweis, Matti Rossi, and Stefan Strecker
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Ulrich Frank
Andreas Oberweis
Matti Rossi
Robert B. France
Stefan Strecker

To effectively support business operations and managerial decision-making, future enterprise systems require an elaborate conceptual foundation that promotes a tight mutual alignment of information systems and the business. Enterprise models provide such a foundation. They integrate conceptual models of an information system (e. g. an object model) with models of the surrounding action system (e. g. business process models or strategy models). Thereby, they relax the notorious cultural chasm between business and IT experts and provide a versatile instrument for the conjoint development of large-scale, mission-critical enterprise systems and for analyzing and (re-) designing the corporation.

However, the development of comprehensive enterprise models requires efforts, expertise, and resources beyond the capabilities of even large corporations. Therefore, the development and dissemination of reference enterprise models that can be adapted to a wide range of companies is a pivotal success factor. Enterprise models are usually specified by domain-specific modelling languages (DSML). The development and evaluation of reference enterprise models and corresponding DSML is an attractive scientific challenge. It corresponds to the development of theories: Reference models and DSML are linguistic constructions (on different levels of abstraction) that come with the claim for general validity or suitability respectively—not just for one particular occurrence but for an entire class of organizations.

They integrate and consolidate contributions from several scientific disciplines such as Computer Science, Information Systems, and Management Science. Both, reference models and DSML provide a reification of an attractive vision: Higher quality of software systems at lower cost. It is the complexity of modern organizations and the diversity of involved perspectives that renders the development of reference enterprise models and corresponding DSML a particular research challenge. Inspired by the remarkable results of the free/open source movement, recent work on reference enterprise models has resulted in the notion of open reference enterprise models (open models for short). Research into open models does not only address the feasibility issue. Furthermore, it introduces a new model of collaboration among researchers, developers, and prospective users of reference enterprise models—leading to the prospect of shaping future enterprise systems. Recent initiatives on joint, collaborative modeling of open licensed conceptual models, thus, provide a new, innovative model for research on reference enterprise models that served as the starting point to this Dagstuhl seminar. It links to research on collaborative modeling, modeling tool development, model management, models@run.time, enterprise systems, and model-driven engineering.

This Dagstuhl seminar was aimed at bringing together a multi-disciplinary group of academic and industry researchers from the disciplines of Wirtschaftsinformatik, Computer Science, Information Systems, and Software Engineering, specifically those working in Requirements Analysis, Conceptual Modelling, and Enterprise Modelling to foster our understanding of how to develop, evaluate, disseminate, and promote the use of open reference enterprise models. The primary emphasis of the seminar was to determine the present state-of-the-art in this multi-disciplinary research field, and to establish a research agenda for future work towards solving theoretical and practical challenges related to the development of open reference enterprise models. The following overview describes more particular questions/objectives and related achievements:

1. **What are key characteristics of future ES?** The analysis of this question started with assumptions about relevant changes to be expected for the use of future ES. On the one hand, it was commonly expected that in many industries there will be a growing need for adapting the ES quickly to changing demands, e. g. to benefit from sudden opportunities or to build effective protection against threats. On the other hand, it was assumed that a growing number of managers will have received professional training in sophisticated uses of information systems. As a consequence, it was concluded that future ES should not only be based on an elaborate conceptual foundation but should also make this foundation, e. g. an enterprise model, accessible to prospective users—on various levels of abstraction and detail. This would not only empower users to perform more advanced analyses, but also to modify the ES to a certain extent by applying changes to certain parts of the underlying conceptual model. From a software engineering perspective such a conception of future ES creates the challenge to allow for using models at run time—and to synchronize models and code. It was concluded that programming languages which allow for an arbitrary number of abstraction layers provide a promising approach to address this challenge.
2. **What is a promising strategy for the development of a common modeling platform?** A platform for enterprise modeling needs to integrate an extensible set of DSML editors. Also, it should support the specification of DSML and the development of corresponding model editors. Furthermore, it should enable model analysis and support the use of models at run time. The participants agreed that there is no environment available that would satisfy all these demands. At the same time, developing such an environment would require a substantial amount of resources and would take years. During that time, the intended modeling activities would be compromised, since they lacked the required tool platform. Therefore, it was concluded that only an evolutionary approach to developing a common modeling platform is a realistic option. It should start with existing modeling tools that are gradually extended or replaced with more advanced systems.
3. **What are key features to be offered by a repository to integrate contributions from a wide range of participants?** Since a common modeling environment cannot be expected at the beginning of an open model initiative, there is need to integrate contributions (models, meta models etc.) from various sources. That puts emphasis on a versatile repository that allows handling a wide range of representations on a level of semantics that enables model integration and various forms of retrieval and analysis. A working group focused on a corresponding architecture and presented an elaborate proposal.
4. **What are appropriate guidelines to establishing and sustaining initiatives and corresponding processes of collaborative modeling of open models?** Apart from incentives, discussions centered on organisational issues involving considerations of the economics of open models and success factors related to community aspects, procedural aspects, stakeholder aspects and infrastructure aspects. A life-cycle and a maturity model were proposed together with an initial process model aimed at guiding the steps to establish and sustain open model initiatives. The concluding plenary discussions corroborated the need for a guided and concerted division of labor.

A joint publication by the organizers is currently in preparation to reflect the seminar's key results. It is to appear in 2013.

2 Table of Contents

3 Overview of Contributions

3.1 Abstract

Jörg Becker (Universität Münster, DE)

So far, reference models (valid for a specific group of companies, built to be reused) have not found the attention in practice as we – as researchers – had wanted them to be used. There may be some reasons for that: 1 There are few 2 The ones that are in place are poor 3 The ones who want to use them do not know that there are some in place 4 Reference models are not useful. My own experience with many companies shows: 4 does not hold true (Working with companies and using reference models has very helpful). 1 and 2 hold true partly (we have to work on better reference models!) 3 holds true → Here, the open model initiative can help! So it's worth working on opening reference models to companies. The work in Dagstuhl was fruitful, inspiring, and bringing the idea of open models forward. We worked on modeling languages, content of reference models, abstraction, meta-modeling, scientific foundation, and how bringing the idea of open models to life.

3.2 The Model Driven Enterprise

Tony Clark (Middlesex University, GB)

Organisations increasingly rely on a distributed collection of heterogeneous systems, find themselves required to comply with a range of dynamically changing regulations, all within a business context that produces events, opportunities and demands using a variety of digital formats and modes. Furthermore, the choice of IT systems that can be used by organisations to replace manual systems and to implement business processes increases and also changes on a regular basis making it difficult and risky to commit to one particular technology choice. Modelling technology has advanced in the last 20 years or so to the point where it is possible to describe complex data, transformations and processes in a technology independent way. Modelling techniques such as transformations, models@RunTime, version control, team working and code generation make it possible to envision a situation where an organisation can encode its business as a collection of technology independent models and to run entirely from the models. This situation is attractive for a collection of reasons. Firstly, it reduces the risk of committing to technology platforms that either change regularly or may not be the optimal choice, since the same models can be made to target different technologies. Secondly, domain-specific modelling techniques can be used to being the representation of an organisation within the grasp of people whose expertise is not technology. In particular domain-specific techniques can provide different views of an organisation for different roles within the company, for example allowing the CEO to view progress, successes and failures at the IT level in terms of the goals of the organisation. Finally, modelling is based on abstraction and thereby allows otherwise highly complex technology to be expressed at an appropriate level of detail. In order to realise the Model Driven Enterprise, it is necessary to address a number of research challenges: when viewed as an engine, what are the key features of an organisation, for example goals, directives, processes, information, roles etc?; what languages should be provided for modelling the enterprise?; what techniques can be used to manage the models within an enterprise? how can the context of an organisation be modelled?; how can organisational models be compared and migrated?

3.3 Open Models @ Runtime

Patrick Delfmann (Universität Münster, DE)

One challenge in establishing a comprehensive support for open reference models and open reference modeling is to provide corresponding methodical and tool support for an overall reference model lifecycle. Such a lifecycle comprises the construction of a reference model, its adoption and adaptation by enterprises, and its use and refinement by enterprises. Furthermore, experiences made in using (potentially adapted) reference models should be integrated into the original reference model in order to consider special requirements coming from particular business players. A new lifecycle of reference modeling can start as soon as the special requirements are integrated into the original model. To establish such a support, an according methodology or platform has to provide mechanisms supporting preferably every step of the modeling lifecycle. One great challenge for the Open Model Initiative will be to establish a corresponding open model platform. Since full support will be a future goal, our working group proposes to set up a tool stepwise, beginning with the possibility to understandand share reference models. The next levels could incorporate manipulation of models, followed by collaboration and transformation tools, modeling language definition and manipulation tools, and model processing tools including variant management, transformation, monitoring, refinement, and re-integration.

3.4 From Model-driven to Model-Integrating Software Development

Gregor Engels (Universität Paderborn, DE)

During the last two decades, the usage of models as a relevant step within software development has been advocated. Unfortunately, this had not lead to an industrial success, as the additional burden of erecting und maintaining models and at the same time the increased market and budget pressure hindered software development teams to invest in such a model-driven development. Therefore, we started a novel research initiative to integrate models and code into a coherent unit, called MoCo. This implies that any information is only represented once, i. e., in case of a flexible notation as a model and in case of an efficient notation as code. During runtime of a software system consisting of MoCos, it may change its state. This means that pieces of code which need an update are re-transformed into a model representation, while models which appear to be stable are compiled into efficient code. This approach of using MoCos is nowadays already present in process-driven service-oriented architectures, when processes are expressed as business process models and business logic as application services. What is missing here, is an on-the-fly transition between model and code and back again. The research described here in conducted in close cooperation with J. Ebert, University of Koblenz.

3.5 The Open Models Initiative as a Platform for the Implementation of Modelling Methods: The Case of the SeMFIS Project

Hans-Georg Fill (Universität Wien, AT)

One vision of an Open Models Initiative is to support the sharing of know-how on the implementation of modelling methods. For this purpose several requirements have to be met in regard to the description of a modelling method, its design in terms of a particular meta modelling approach and its technical realization using a concrete implementation platform. In the paper at hand we will discuss these requirements and show how they were realized in the context of the Semantic-based Modelling Framework for Information Systems (SeMFIS) project. SeMFIS provides a set of model types, algorithms and services for managing semantic aspects of conceptual models about information systems and has been realized using the ADOxx meta modelling platform and the Protégé ontology management toolkit. Subsequently we derive a set of general guidelines for other Open Models projects based on these insights.

3.6 Multi-Level Modelling

Ulrich Frank (Universität Duisburg-Essen, DE)

In recent years, the idea of domain-specific modeling languages has raised remarkable attention. This is for good rashes. DSML provide reconstructions of domain-specific technical languages. They promise to promote modeling productivity and the quality of models. However, a close look at DSML shows that there are frequent differences in the use of domain-specific terms. This poses a serious challenge to reusing DSML. The approach I presented addresses this challenge. On a higher level of abstraction a generic DSML serves to represent textbook knowledge that is applicable to a wide range of domains. The level below serves to represent organization specific instantiations of concepts defined with a corresponding generic DSML. The differentiation of multiple levels of models – and modeling languages respectively – promises to overcome the conflict between a high range of reuse (which recommends a low level of semantics, but promotes economies of scale) and a high benefit of reuse in a particular case (which recommends concepts that fit the specific requirements of a particular domain). In addition to that, it is also suited to foster integration: If two organizations do not succeed in specifying a common schema on the type level, because the conceptual diversity is too big, they can still go for common concepts on a higher (meta) level, thereby allowing for integration at least on this higher level. To give an example: Two companies that deal with clearly different types of products (e.g. software and industrial components) could still define common meta types of products which then could be instantiated into specific types. The prospects of multi-level modeling are contrasted by substantial challenges which are mainly related to restrictions of prevalent programming languages: To build corresponding model editors one would need a language that is not restricted to two levels of abstraction (such as “class” and “instance”). In recent years a number of (meta) programming languages have emerged that allow for overcoming this restriction be providing an arbitrary number of meta levels. They form a promising foundation for future research on multi-level modeling and corresponding tools.

3.7 Hub Services as a Use Case for Open Enterprise Models

Andreas Hess (Capgemini München, DE)

Technology trends like mobility and the availability of public cloud services support new business activities in market-facing units, close to clients or in cooperation with collaborators from other enterprises that can co-exist together based on a loose-coupled, stateless consumption of "services" on demand. As a result, future Enterprise Systems might have the characteristics of an Enterprise Integration Hub that supports dynamic interactions between collaborators inside and outside of the enterprise using application services that are provided by the involved enterprises, are acquired in the cloud or are created using services of the hub. To enable this interaction these hubs will make use of models that describe information and its exchange, offered and consumed services including choreography as well as orchestration of services and associated business rules. Because of the dynamics of the interaction and the affiliation of the collaborators to different enterprises these models necessarily need to be open. To effectively support the collaboration the services of the integration hubs will cover social network like functionality, the creation and provisioning of data and meta data including model management plus the acquisition, creation and usage of services on demand. Because of their characteristics the integration hubs can serve as catalyzers for the development of open enterprise models: They request the existence of models for their operation and offer the environment needed for the development of such models as open content at the same time.

3.8 The ADOxx® Metamodelling Platform: Functional Requirements

Dimitris Karagiannis (Universität Wien, AT)

Enterprise models have the potential to act as a conceptual foundation for enabling mutual alignment between information systems and business. Hence the use, development, and evaluation of modelling methods is not only (a) an attractive scientific challenge, but also (b) a business goal to achieve efficient model-based development for future enterprise systems. Modelling method tool support requires and relies on available IT-infrastructure and a conceptual backbone, like a $meta^2$ concept. This concept evolved to a mature approach for developing, aligning, using and evaluating hybrid modelling methods for enterprise applications. The functional capability of the underlying metamodelling platform is a critical success factor for both(a) working on scientific issues and (b) realising future enterprise solutions. The first part of this paper focuses on: (a) the core elements of a metamodelling platform and (b) the nature and origin of its functional requirements. The second part is concerned with three basic observations. First, technological trends such as—but not limited to: (a) web-applications, (b)collaboration and social software, (c) adaptability and personalisation of software, (d) mobile devices and third party interaction, (e) semantics and (f) cloud computing as well as very large data sets that need to be taken into consideration. Second, concrete user scenarios from industrial and research projects in the domain of business and IT modelling. Third, the maturity of existing metamodelling platforms as commercial as well as open source/use software. The third part describes how ADOxx®—a metamodelling development and configuration platform for the implementation of modelling methods can be applied. The ADOxx® platform supports: (1) modelling languages by inheriting modelling concepts from a metamodel to define syntax, semantics and notation,(2) modelling mechanisms and algorithms by providing generic platform functionality that can be used or adapted, scripting possibilities, integration and interaction with third party add-ons, as well as (3) modelling procedures by combining model types as part of the modelling language, and scripts as part of the mechanisms and algorithms to support the sequence of modelling. The paper concludes with an evaluation of ADOxx® applications, which are realized on the Open Model Initiative (www.openmodels.at), and the outlook on future functionality.

3.9 Feedback on seminar topic

Mogens Kuehn-Pedersen (Copenhagen Business School, DK)

The seminar revealed a general expectation that modelling Future Enterprise Systems entailed new open modelling processes that would heed cross-company developments and mobility supported by multiplicity of platforms. Common select, domain specific semantics would be supported by numerous technologies including intelligent agents design, standards and tools. Practice would increasingly benefit from application of open models as shared data become a precondition for operational effectiveness and innovative improvements.

3.10 Structured design of a modeling language

Marc Lankhorst (Novay – Enschede, NL)

Joint work of Lankhorst, Marc; Proper, Erik; Jonkers, Henk
Main reference M.M. Lankhorst, H.A. Proper, and H. Jonkers, "The Anatomy of the ArchiMate Language," IJISMD 1.1 (2010): 1–32. Web. 29 Mar. 2012.
URL http://www.novay.nl/publicaties/the-anatomy-of-the-archimate-language/64893

In current business practice, an integrated approach to business and IT is indispensable. In many enterprises, however, such an integrated view of the entire enterprise is still far from reality. To deal with these challenges, an integrated view of the enterprise is needed, enabling impact/change analysis covering all relevant aspects. This need sparked the development of the ArchiMate language, which was developed with the explicit intention of becoming an open standard, and as such has been designed such that it is extendable while still maintaining a clear and orthogonal structure. This article is concerned with documenting some of the key structures and design principles underlying the ArchiMate language. ArchiMate is designed as an architecture description language (ADL) for enterprise architectures. Developing such a language comes with many challenges. The design principles of the ArchiMate language aim to tackle these challenges. The modelling concepts of ArchiMate were derived in a stepwise process, applying these principles and successively refining high-level, abstract concepts to obtain concepts relevant for enterprise architects. In this, we make a distinction between concepts needed to model domains in general, the modelling of dynamic systems, and the modelling of enterprise architecture specific elements. This approach helps to, on the one hand, base our language on established conceptual modeling approaches, and on the other hand realize a concrete and usable language. Moreover, this backbone structure allows for extensions of the language by refining the higher-level structure for specific domains and/or users, which then become an integral part of the language, not just something that is grafted on as an afterthought.

3.11 Abstract

Sina Lehrmann (TU Dresden, DE)

The brain storming session and the group work revealed a categorization of reference models according to differing purposes. Corresponding to the viewpoint categorization within the Archimate Specification reference models could be differentiated in reference models for the purposes designing, deciding and informing.

- Designing: Reference models serve as a blueprint for a good solution. Enterprises could adopt the offered design solution for similar problems.
- Deciding: Reference models provide decision support by pointing out different alternatives, parameters, experiences etc. In general these models areconstructed inductively. The Open Model Initiative could support the construction and evolution of this kind of reference models by gathering and generalizing experiences from different sources.
- Informing: Reference models could promote the reverse direction of communication by announcing publicly that certain enterprise systems are aligned to it. E.g.reference models could act as a means for certification, which could be used asa marketing instrument in turn. To provide the standard or regulation as are ference model improves transparency and facilitate the negotiation of reasonable regulations. The Open Model Initiative could be the independent and reliable third party.

Particularly the last category for reference models contains innovative research ideas.

3.12 Challenges for Open Reference Models

Peter Loos (Universität Saarbrücken, DE)

Conceptual models play an important role in all phases of information systems life cycle, e.g. business engineering, IS development and ERP customizing. To reduce the effort and improve the fault-prone process of modeling user requirements, reference models as blue prints for enterprise-specific models are regarded as an appropriated means. Hence, a reference model is generic for a certain type of companies or organisations according to their typological characteristics, e.g. industry domain and company size. Since reference models represent a common body of knowledge it is suitable that they are available as open models. Openness refers to public availability as well as open development, e.g. in a crowd-sourced manner. First collections of reference models in the form of reference model catalogs exist (e.g. rmk.iwi.uni-sb.de). More challenging is the development of open models. Organizing collaborative development processes and finding appropriated tools and platforms for the design of open models can be based on the experiences with and can adopt techniques from the field of open source software. However, there are some specific challenges concerning open models: (1) Contrary to open source software the appropriate level of model abstraction is not clear. If the model it to detailed, it might be too specific to use it as a blue print, while a coarse-grained model provides only marginal support. (2) Furthermore, there are only limited means for quality measurements and quality assurance of conceptual models. However, for conceptual models high quality is crucial since flaws in the requirements specification lead to expensive delay in software development or ERP customizing. (3) The development process of reference models can differ from the development process of software code. While conceptual models are usually constructed in a deductive way (collecting requirements, formalize them and describe them by means of a model language) like software code, reference models can also be derived in an inductive way. Comparable to process mining approaches, where process models are derived from event logs of process instances, reference model can be mined by analyzing various enterprise-specific conceptual models (reference model mining). A combined deductive and inductive approach for developing is assumed to foster the quality of open reference models.

3.13 Open Models for Business Information Systems Development

Andreas Oberweis (KIT – Karlsruhe Institute of Technology, DE)

Main reference F. Schönthaler, G. Vossen, A. Oberweis, T. Karle, "Business Processes for Business Communities. Modeling Languages, Methods, Tools," Springer-Verlag, 2012
URL http://www.horus.biz/public-space.html

The open models concept provides promising opportunities to improve the development of future business information systems. Open models allow a better alignment of information systems to user needs due to more intensive user participation in the design phase. Furthermore open models support a more open discussion of functional and non-functional requirements for business information systems than traditional requirements engineering concepts.

A key research objective in the field of open modeling is developing constructive and analytical methods to guarantee the quality of open models. Languages for open models should provide mechanisms for consistently refining and coarsening models since different user groups might require different levels of model granularity. Another important challenge in the field of open models is efficient maintenance of large sets of open models in repositories. A query language is required for effectively finding models in possibly distributed repositories. Another open issue is the question whether integrated models, including e.g. descriptions of activities, objects and roles, are preferable to more separate models for different aspects of information systems. A practical question of constructing open models is about who should build open models, and how modelers, especially experienced practitioners, can be motivated to participate in open modeling efforts. Collaborative modeling activities must be effectively supported.

Horus is a set of languages, methods and software tools for information systems modeling. Horus especially supports modeling processes within business communities. It integrates concepts of typical social software systems in order to collaboratively develop different types of models in an open process. Horus includes simulation and analysis tools for community based evaluation and improvement of models. Reference models are provided in public spaces to improve productivity and quality of modeling processes. Horus can be downloaded from http://www.horus.biz/public-space.html.

3.14 Coherent Modelling Landscape

Erik Proper (Radboud University Nijmegen, NL)

Currently, models produced during one stage of the transformation process (such as an ArchiMate model) quite often have to be re-drawn, or even re-modelled, in some other language in a later stage of the process (such as a BPMN or a UML model). This leads to unnecessary delays and costs during a transformation process, and basicially constitutes a major disinvestment. The coherence (and automatic transformations) between different models is hampered due to the inherent disconnectedness of the modelling languages used, such as BPMN, UML, ArchiMate, et cetera. With "inherent disconnectedness" I refer to the fact that the meta-models underlying these languages have (from their designs) no formal connections. At the same, time an actor used in e.g. an ArchiMate model will re-appear as an actor in a BPMN model, while this latter model may also provide more details of the business process used in the original ArchiMate model. Of course it is possible to provide a mapping from (relavant parts of) an ArchiMate model to a BPMN model. However: A better integration of the meta-models would make such transformations more easy. A BPMN model provides a detailed view of the actual process and the roles of the actors involved, than what an ArchiMate model would. Therefore, one would expect the BPMN meta-model to be a specialisation of (part of) the ArchiMate meta-model as well. Regretfully, this is not the case at present, but might be strived for by the standardisation bodies. Even more, the needed transformations between e. g. (a relevant part of) an ArchiMate model towards/backwards a BPMN model could we standardised and become part of the body of standards (e. g. supporting boundaryless information flow at the level of models). This would ensure the portability of these transformations between different modelling tools in use by organisations. Both of these require an active role of the standardisation organisations such as the OMG and The Open Group, as well as their core members to take their responsibility in this. One might argue that the problem of coherence between models can be solved easily by creating one integrated modelling language. Essentially UML already provides such a language focusing at the level of software applications and their direct usage environment, while ArchiMate provides such a language focussed at the representation of enterprise architectures over different levels of abstraction (from technology via applications to the business level). The operative word here is "focussed". When designing a modelling language, one selects different modelling constructs to express the models. As argued in two earlier papers (1, 2), the modelling concepts included in a modelling language should really provide a real utility in relation to the purpose/focus of the language. Depending on the stage of an enterprise transformation, the aspects of the enterprise one focusses on, etc, different sets of modelling concepts are necessary. Therefore, a single unified modelling language will be hard to create, and even harder to use. In that sense we are likely to end up with several more focussed languages, with their own added value. At the same time, this does not have to mean that we cannot have coherence between the different models. For example, within a single enterprise transformation, one may use:

- e3Value to model the position of the enterprise in a value web
- DEMO to elaborate the essential transactions between the enterprise and its environment as well as the essential internal workings of the enterprise
- ArchiMate to elaborate the enterprise architecture towards IT support for the enterprise's activities, and
- BPMN and UML to refine things even further to the level of specific applications and business processes.

These are all valid reasons for using the distinctive modelling languages. At the same time, it is only fair to expect to be able to trace the relations between:

- value exchanges between the enterprise and other actors in a value web (e3Value),
- the transactions between these actors operationalising these value exchanges and the essential processed needed to realise them (DEMO),
- the implementation of these essentual transactions and processes in terms of tangible actors, applications and IT, in terms of an enterprise architecture (ArchiMate),
- the actual realisation of these artefacts in applications and business processes (BPMN and UML).

In other words, a coherent modelling landscape is called for. To really be able to do so, requires these models to be interrelated, and eventually, the meta-models of the underlying modelling languages. The most basic way of realising this is to at least use persistant naming of actors, processes, etc, accross the different models. However, to explicitly express the fact that a specific value exchange (e3Value) is implemented using a number of transactions (DEMO), requires additional relations matching the two meta-models. The most practical way to proceed at the moment would be to apply a disciplined naming convention for the concepts used. A practical way of doing this would be the use of a domain model of the different domain concepts used accross the specific e3Value, DEMO, ArchiMate, etc, models, and a consequent use of the (names of these) concepts accross the models. Actually, creating such a domain model may also help modellers in the creation of more specific models such as value models and process models, since they can then start from a thorough understanding of the domain. A more ambitious approach would also require more advanced modelling tools, in which meta-models of different modelling languages are positioned in a hierarchy in such a way that models can also be mutually related and essentially be re-interpreted in terms of more specific meta-models. In the past, dome some initial work has been done in this regard.

3.15 Future Enterprise Systems in Business Ecosystems

Mirja Pulkkinen (University of Jyväskylä, FI)

As maybe the most challenging features, adaptability and flexibility are expected of future enterprise systems for modifiability in quick responses to changes in the business environment and thus changing needs of the business these systems support. Among the facets of an envisioned future of enterprise systems are ecosystems, where several enterprises, in conjoined efforts, participate in provisioning business services. Besides interlinked, interoperable information systems, another trend is the provisioning of services is to an extent migrating into the computing cloud. Cloud services allow for evolving ecosystem participation by business partners and customers. The flexible, enhanced ICT capabilities are in future even more a business enabler, with the potential to meet the market needs quickly and precisely. Reference models are an expedient for the design and maintenance of these capabilities. Within an ecosystem, a common understanding of the systems, the business services and the processes to provision the services must exist for the information system supported co-operation. Open reference models (ORM) contribute essentially to the collaboration, both in intra-organizational settings and in inter-enterprise constellations, when design models for interoperable systems are created and interfaces designed. Reference models or model elements have different origins and audiences. There are differences in the modeling languages and disciplines in communicating them. A research avenue is opened here to explore the support for the collaborative construction, maintenance and use of open reference models. This is an effort across different communities of practice among the stakeholders either in a single enterprise, their reference groups like professional communities, or further, in the business ecosystems at a broader scale. Different contexts and goals of the diverse communities present both a challenge, and a potential driver for open reference models: there are common, reusable but also community specific features with existing models and modeling methods. However, the modeling and different aspects of it (languages and the overarching communication between the communities of practice with their specific linguistic practices) presents a challenge. The combination of IS design and methodology knowledge, and the knowledge on the linguistic behavior and communication in communities of practice is a possibility to meet the challenge.

3.16 Faithful Models of Discrete Dynamic Systems

Wolfgang Reisig (HU Berlin, DE)

This talk focuses on models of discrete, dynamic systems rather than datamodels. I start with the fundamental observation that the choice of the level of abstraction is fundamental for a modeler (whereas a programming lan-guage fixesthe level of detail for a programmer). We identify four requirements that a good modeling technique should fulfill: 1. Free choice of the level of abstraction: A good modeling language allows the modeler maximal freedom to chose the level of abstraction. 2. Faithful models: A model is faithful if—on the chosen level of abstraction—the elementary system items and operations correspond bijectively to the elementary model items and elementary model operations; the composed system items and operations correspond bijectively to model compositions; and the system states and steps correspond bijectively to model states and steps. In a faithful model, every property expressible on the chosen level of abstraction corresponds to a property of the model. Systematic refinement, i.e. steps to more detailed levels of abstraction, should ideally yield faithful models again. A modeler may "open up" his model until a distinguished detail level of abstraction has been reached. 3. Minimal infrastructure: Each model of dynamic systems assumes some kind of infrastructure ("Operating system") that guarantees runs to continue, if possible. Assumptions about the effect of the environment should be made explicit and kept to a minimum. Implicit assumptions about the infrastructure are the source of most mismatches of models. 4. All this in one formalism: Is it possible to squeeze the above assumptions into one formalism? In fact, this can be achieved on the basis of Tarski structures, sig-algebras and Gurevich's Abstract State Machines.

3.17 The Business of Open Models

Dirk Riehle (Universität Erlangen-Nürnberg, DE)

Main reference The Business of Open Models (Blog entry)
URL http://dirkriehle.com/2012/03/27/the-business-of-open-models/

For open models to be sustainable, they'll need a business model. My suggestion is to create developer foundations like Apache or Eclipse for this.

3.18 Abstract

Matti Rossi (Aalto University, FI)

I see OMI as on important possibility for changing how enterprise systems are developed and deployed. For this kind of idea to become widespread, several obstacles need to be overcome. First there needs to be a demand for the models and a critical mass of models to start with, when these are available, there needs to be a community working with the models in the repository. Repository itself and tools for using it are needed also. Finally there needs to be use cases and tools to support those use cases in the OMI site. I believe that industry specific ES reference models could be a good starting point. This could provide a platform for an ecosystem of new ES and individual services to be build and for companies within the industry to use.

3.19 Open Models: Community-driven Collaboration to Promote Development and Dissemination of Reference Models

Stefan Strecker (FernUniversität in Hagen, DE)

Main reference U. Frank, S. Strecker, "Open Reference Models – Community-driven Collaboration to Promote Development and Dissemination of Reference Models," in: Enterprise Modelling and Information Systems Architectures: An International Journal, Vol. 2, No. 2, November 2007, pp. 32–41.

Reference models constitute a reification of a promising vision: Higher quality of information systems at less cost through reuse of confirmed domain knowledge and systems design. Paradoxically, however, development and, in particular, reuse of reference models has been ratherlimited both in practice and academia. The Open Model movement draws on analogies to free and open source software development to overcome the present barriers to the development and adoption of reference models. It has been reasoned that ancommunity effort involving participants from academia and industry promises to leverage complementary know-how and resources to create a win-win situation for those who contribute domain knowledge as well as those who contribute modelling know-how. It has, however, become clear over the past few years – and first attempts to establish open model initiatives – that the Open Model conception requires a convincing (i.e. elaborate) kernel of models, modelling languages and tools in order to provide incentives for third parties to join in and to reach a critical mass. The Dagstuhl seminar on Open Models as a Foundation of Future Enterprise Systems not only underlined the necessity of such a kernel but also pointed at very attractive applications of open reference models in the context of next-generation enterprise systems.

3.20 Science and art of conceptual modelling / Pragmatism for Open Models: Codesign + Pattern + Storyboarding

Bernhard Thalheim (Universität Kiel, DE)

Science and art of conceptual modelling

Conceptual models are one of the main instruments for information systems development. A large body of knowledge has been developed in the past and resulted in sophisticated modelling techniques and languages. It needs however a combination, compilation, systematisation, and a general art (in the sense of the book series “The art of programming” by D.E. Knuth). At the same time, most of the notions of conceptual modelling must be clarified: what is a model, what is a concept; what is the use and value of a model; which community of practice acts; . . . The talks survey our results on the definition of the notion of a “model” as an artifact with specific characteristics and qualifying propoerties, of “to model” as primitive or composite acts or activities, and of “modelling” as a systematic art or science, of concepts, of intention, of purpose as the main driving force, of (added) value of a model, of roles and plays of members from a community of practice, etc. It continues the theory of conceptual modelling in the Handbook of Conceptual Modelling.

Pragmatism for Open Models: Codesign + Pattern + Storyboarding

The codesign approach to conceptual modelling covers structuring, functionality, distribution and interactivity specification for large information systems specification and realisation. This approach has been certified to be on SPICE level 3. The codesign methodology might thus serve as a starting point for an integration of models which are concentrating on covering complete enterprise models. It uses the experience we have gained by our industrial schema library. Abstraction is an essential feature for the development of an open model library. One kind of abstraction—beside the meta-(meta-(meta-))-level abstraction—is generalisation abstraction. It can be based on pattern, i. e. generic solutions to basic and composite modelling problems. At the same time, models are for use and deployment. Therefore, they are bound to deployment and development stories. These stories can be modelled as storyboards.

3.21 Stakeholder-specific Modeling

Michael zur Mühlen (Stevens Institute of Technology, US)

Models serve multiple purposes: They provide a shareable conceptualization of some subject matter that can be shared among stakeholders. They can replicate or explain a phenomenon, or they can predict, guide and constrain future phenomena. In the first two cases, the process of modeling may prove to be as significant as the resulting model. Collaborative modeling creates a shared understanding of same problem (or solution) space. Consensus in this area requires understandable methods, tools and design processes. But: Stakeholders should be able to retain their specific interests that guide, constrain, and predict need to be understood and interpreted in a uniform fashion. In this area, models may be more important than the process of their creation. It is important for the conceptual modeling community to recognize these use case differences and to focus differently, depending on the area of application.

4 Working Groups

Five working groups discussed pertinent research issues in the vicinity of the seminar's scope. The four working groups were:

1. **Future Enterprise Systems**: Gregor Engels, Andreas Oberweis, Eric Proper, Mirja Pulkinnen, Stefan Strecker, Bernhard Thalheim.
2. **Modelling domains and purposes**: Jörg Becker, Marc Lankhorst, Sina Lehrmann, Peter Loos, Erik Proper, Mirja Pulkkinen.
3. **Technical Infrastructure and tools**: Dimitris Karagiannis, Andreas Oberweis, Florian Matthes, Wolfgang Reisig, Dirk Riehle, Matti Rossi.
4. **Organisation**: Hans-Georg Fill, Dirk Riehle, Mogens Kühn Pedersen, Michael zur Mühlen.
5. **Open Models @ Runtime**: Tony Clark, Patrick Delfmann, Jörg Desel, Werner Esswein, Robert France, Ulrich Frank, Andreas Hess.

5 Open Problems

Open research issues and practical problems will be discussed in a joint publication by the organizers to appear in 2013.

Participants

- Jörg Becker
Universität Münster, DE
- Tony Clark
Middlesex University, GB
- Patrick Delfmann
Universität Münster, DE
- Jörg Desel
FernUniversität in Hagen, DE
- Gregor Engels
Universität Paderborn, DE
- Werner Esswein
TU Dresden, DE
- Hans-Georg Fill
Universität Wien, AT
- Robert B. France
Colorado State University, US
- Ulrich Frank
Universität Duisburg-Essen, DE
- Andreas Hess
Capgemini – München, DE
- Dimitris Karagiannis
Universität Wien, AT
- Mogens Kühn-Pedersen
Copenhagen Business School, DK
- Marc Lankhorst
Novay – Enschede, NL
- Sina Lehrmann
TU Dresden, DE
- Peter Loos
Universität des Saarlandes, DE
- Florian Matthes
TU München, DE
- Andreas Oberweis
KIT – Karlsruhe Institute of Technology, DE
- Erik Proper
Radboud Univ. Nijmegen, NL
- Mirja Pulkkinen
University of Jyväskylä, FI
- Wolfgang Reisig
HU Berlin, DE
- Dirk Riehle
Univ. Erlangen-Nürnberg, DE
- Matti Rossi
Aalto University, FI
- Stefan Strecker
FernUniversität in Hagen, DE
- Bernhard Thalheim
Universität Kiel, DE
- Michael zur Muehlen
Stevens Inst. of Technology, US