Chapter VI

Evaluation of Reference Models

Ulrich Frank, University of Duisburg-Essen, Germany

Abstract

Evaluating a reference model is a demanding task. Not only do reference models inherit the problems well known from the evaluation of conceptual models in general, but furthermore, their claim for general (re-) usability implies the ability to take into account the possible variety of requirements and specific constraints within the set of potential applications. This Chapter presents a method that is aimed at fostering a differentiated and balanced judgement of reference models. For this purpose, it takes into account various perspectives—among others, economic, engineering and epistemological. It also includes a process model that demonstrates how to organize a specific evaluation project.

Evaluation of Reference Models as a Multi-Faceted Challenge

Reference models are a reification of a very attractive vision: They promise higher quality of information systems at less cost. This vision goes along with two pivotal claims. On the one hand, reference models are intended to provide appropriate *descriptions* of an application domain. On the other hand, reference models are aimed at delivering *blueprints* for a distinctively good design of information systems and related organizational settings. Thus, they are descriptive and prescriptive at the same time. While many conceptual models include descriptive and prescriptive elements, reference models should fit the specific needs of a whole range of organizations. Since the idea of reference modeling is emphasizing the improvement of quality, evaluating them is a core issue: From the perspective of prospective users, it can hardly be taken for granted that a particular reference model is of superior quality. However, evaluating reference models is a major challenge. This is for various reasons. Not only that reference models inherit the problems well known from the evaluation of conceptual models in general, but furthermore, that their claim for general (re-) usability implies a takeing into account the possible variety of requirements and specific constraints within the set of potential applications. Another source of complexity is the variety of objectives related to the use of reference models. They include economic goals, such as increase of productivity, or goals related to specific analysis, design or implementation tasks. In addition to that, testing the claim for excellence faces deep and subtle epistemological problems.

Against this background, the paper will propose a *method* for evaluating reference models. It consists of a conceptual framework that serves to structure the overall evaluation problem, which is supplemented by a prototypical process model that demonstrates how to organize a specific evaluation project.

Related Work

While reference models are arguably of pivotal importance for the IS discipline, the idea of reference models has not been around for too long. This is even more the case for actual reference models. In a recent survey focussed on German speaking countries, Fettke and Loos identified only 33 reference models of various kinds, 22 of which were accessible (Fettke & Loos, 2004). Therefore, it does not come as a surprise that there have been only a few approaches that focus explicitly on the evaluation of reference models. However, there is other work which is directly related to this topic: approaches to the evaluation of conceptual models and approaches to the evaluation of modeling languages.

Evalutation of Conceptual Models

Reference models are conceptual models. A conceptual model is an abstraction that stresses the core terms or concepts which characterize an application domain, while neglecting technical aspects that are related to the implementation of corresponding software systems. Hence, they should contribute to a better understanding of a domain and foster the communication between the various stakeholders involved in a particular project. In a definition that has been quoted frequently, conceptual models are regarded as "... descriptions of a world enterprise/slice of reality which correspond directly and naturally to our own conceptualizations of the object of these descriptions" (Mylopoulus & Levesque, 1984, p. 11). However, describing reality is only one facet of a conceptual model. Usually, a conceptual model is a (re-) constructing description of a domain that also includes prescriptive elements. This is for two reasons. First, it will often make no sense to leave a domain the way it is if one wants to foster the introduction of efficient software systems. Instead, it will usually be required to re-organize patterns of action, such as business processes. Second, the development of software has to take into account the limitations of implementation level languages. In order to support a seamless transformation of conceptual models into implementation level documents, it is not advisable to completely neglect the concepts used on the implementation level.

There is a widespread consensus that conceptual modeling is pivotal for the professional development of information systems. But only if conceptual models are of high quality themselves, will they foster the implementation of high quality software. Therefore, the evaluation of a conceptual model's quality is an important topic in IS. With respect to designing information systems, there have been several attempts to guide the evaluation of a model's quality. They all stress the necessity to use a multi-criteria approach for conceptualizing the notion of quality. Moody and Shanks suggest six criteria to evaluate Entity Relationship models: *simplicity*, *understandability*, *flexibility*, *completeness*, *integration and implementability* (Moody & Shanks, 1994). In a textbook on data modeling, Batini et al. suggest similar criteria (Batini, Ceri, & Navathe, 1992). Lindland et al. emphasize, among other things, the need for using a modeling language that is appropriate for the problem domain and for the expected audience (Lindland, Sindre, & Sølvberg, 1994).

To validate a particular model, evaluators differentiate syntactic, semantic and pragmatic quality. While syntax and semantics are considered on a formal level, among other things, "inspection" and "explanation" are suggested as instruments to foster pragmatic quality. In addition to syntax, semantics and pragmatics, Krogstie et al. (1995) propose "knowledge quality" explicitly as an evaluation criterion. It refers to the knowledge of people who participate in a modeling project. Therefore, they suggest "perceived semantic quality" as a further criterion. They do not, however, discuss how this aspect could be used for the overall evaluation of a conceptual model. A later refinement of this approach (Krogstie, 1998) does not answer that question either. Becker et al. (Becker, Rosemann, & Schütte, 1995) suggest six principles for appropriate conceptual modeling: *correctness*, *relevance*, *economics*, *clarity*, *comparability* and *systematic construction*. Later, this approach was refined by Schütte—mainly based on epistemological considerations. Among other things, he replaced "correctness" with *constructive fitness*. In a recent review of quality frameworks, Moody demands a unification of existing frameworks and stresses the need for disseminating them into practice (Moody, 2005). Weber suggests a focus on the question of how well a model represents a user's conception of the real world (Weber, 1997, p. 72). While this important question is difficult to answer—unless you favour a naive realism—it is not sufficient for many conceptual models either. Since conceptual models often describe future domains, they cannot be evaluated against a user's perception of reality only. In a recent article, Shanks et al. suggest to use ontologies for validating conceptual models (Shanks, Tansley, & Weber, 2003). However, they mainly discuss how a philosophical ontology could contribute to the selection of an appropriate modeling language.

To summarize, research on evaluating the quality of conceptual models has resulted in various frameworks that suggest criteria which cover—and sometimes mix—a wide range of quality aspects from language concepts and syntactical features to user perception. Since reference models are special kinds of conceptual models, all of these criteria could be applied to them, as well.

Evaluation of Modeling Languages

The quality of a conceptual model depends on the suitability of the modeling language being used. Hence, it implies taking into account the quality of modeling languages as well. There are not many publications on evaluating the quality of mode\ling languages. They can be differentiated into three categories: approaches that focus on formal requirements, approaches that focus on pragmatic aspects concerning the use of a modeling language and approaches that make use of ontologies.

In software engineering, the purpose a modeling language should serve is mainly restricted to formal aspects: It should provide a suitable basis for the implementation of correct and reliable software. Hence, formal properties like *completeness*, *simplicity* and *correctness* (for instance, Süttenbach & Ebert, 1997) are of outstanding importance for the evaluation of a language. In addition to that, the analysis of languages in computer science is sometimes related to their expressive power, for instance, by referring to a particular layer of the Chomsky hierarchy. While both aspects, formal rigour and expressive power, are relevant for a number of purposes models may have to serve, they neglect entirely the users' perspectives and those purposes that are not directly related to the implementation of software. Notice also that such approaches to evaluate modeling languages do not allow for discriminating between a set of modeling languages that are complete and correct and share the same expressive power.

Partially, approaches to a more pragmatic evaluation of modeling languages were motivated by the need to compare modeling languages. Some of the approaches are not focussed on modeling languages alone, but at entire modeling methods, e.g., De Champeaux and Faure (1992), Hong and Goor (1993), and Monarchi and Puhr (1992). As part of a comprehensive analysis of Petri nets, Zelewski (1995) has developed a framework for evaluating modeling languages. While his focus is primarily on Petri nets, the criteria he suggests can be applied to other modeling concepts/languages as well. He differentiates between general language features (like expressive power) and features that are helpful for specific applications of a language. In order to support the evaluation of the latter, he introduces a number of criteria. Among other things, they include concepts to express causality, temporal semantics, to coordinate tasks, sequential and parallel processes, etc. In addition to approaches that are aimed on the conceptualisation of quality, there have been attempts to detect quality aspects through empirical studies. They are focused mainly on the perception of language users. Two studies, Goldstein and Storey (1990) and Hitchman (1995) found evidence that many prospective users have difficulties understanding and hence using Entity Relationship models. However, the studies suffer from two shortcomings. First, they were not representative—both with respect to prospective users and the selected modeling languages. Second, they did not take into account how the level of training for applying a particular language would influence a user's judgement.

Referring to the philosopher Bunge, Weber recommends to regard the level of "ontological completeness" (Weber, 1997, p. 94) as essential for the quality of a modeling language (he speaks of a "grammar"). A language is ontologically complete if it provides concepts to represent each class of phenomena in the real world. Despite the formal definition he introduces for ontological completeness, Weber admits that there is hardly a complete list of phenomena everybody could agree on. To be more concrete, he suggests a number of features a modeling language should provide in order to be ontologically complete. They include concepts to express "things," "properties of things," "types," "states," "laws" (comparable to constraints), "lawful states" (comparable to invariants) and events. By applying his criteria to the Entity Relationship Model, Weber establishes that the ERM is not ontologically complete. Opdahl and Henderson-Sellers use Bunge's ontology for the evaluation of an object-oriented modeling language (Opdahl & Henderson-Sellers, 2005). Fettke and Loos apply the same ontology to a language for enterprise modeling (Fettke & Loos, 2003). Ontologies can foster a better understanding of modeling languages. However, they are not a sufficient reference for evaluating modeling languages or models on the object level: Both modeling languages and models are abstractions that should serve a certain purpose. This does not imply that they have to be complete in an ontological sense. There is only one exception where an ontological evaluation makes sense: in the case of (meta) modeling languages that come with the claim of covering all possible modeling purposes related to information systems, such as the UML. It is common practice in IS research to use Bunge's ontology as a reference without questioning it. Such an attitude is hardly acceptable. Bunge's ontology—like any other—is an artifact in itself. While it is elaborated and Bunge has a remarkable reputation as a philosopher, it is certainly not appropriate to take it for granted.

Evaluation of Reference Models

There are hardly any specific approaches to the explicit evaluation of reference models. In order to foster a systematic description, Fettke and Loos propose a method to guide the classification of reference models (Fettke & Loos, 2003d). With respect to the evaluation of reference models, the same authors suggest a multi-perspective approach (Fettke & Loos, 2003c). For this purpose, they outline various research approaches to study the quality of reference models—e.g., "feature-based evaluation," "ontology-based evaluation" and "cognitive psychology-based evaluation." In a further publication, they demonstrate how to use the "Bunge-Wand-Weber Model," which is based on Bunge's ontology, for the evaluation of reference models (Fettke & Loos, 2003b). However, they mainly apply the ontology to conceptual models and modeling languages in general. Mišic and Zhao present a "linguistic-based comparison framework" for evaluating reference models and apply it to a few selected models. Their notion of reference model is slightly different from the one outlined above, since they put more emphasis on system architecture: "…a conceptual framework for describing system architecture" (Mišic & Zhao, 2000, p. 484). They extend the framework proposed in Lindland, Sindre, and Sølvberg (1994) by a few criteria, e.g., "level of stratification" (does a model offer different levels of abstraction) or "orientation" (technology or business). However, they do not discuss any feature that would be specific to reference models.

A Multi-Perspective Framework for Evaluation

Our brief overview of the state of the art in evaluating reference models reveals a number of peculiarities. First, there has been only little work on the explicit evaluation of reference models. The majority of related work is concentrated on evaluating conceptual models or—to a lower extent—on evaluating modeling languages. Most authors suggest a *multi-perspective* approach. Perspectives are often inspired by linguistic categories (syntax, semantics and pragmatics), sometimes extended by a more differentiated consideration of users' perception or a model's relationship to reality. While most frameworks include the judgment of language features, some lack an explicit differentiation of meta- and object levels. The use of ontologies is a valuable contribution to a more comprehensive and obliging evaluation. However, usually the ontology that serves as a reference is taken for granted, thereby terminating the course of reasoning in a somewhat ideological way.

In part, the framework presented in the following section makes eclectic use of the work discussed so far. Therefore, it stresses a multi-perspective approach. It also takes the burden of these approaches: that an objective evaluation is hard to accomplish. Hence, the idea is to get closer to objectivity by fostering a more differentiated and balanced judgment. In this sense, the structure that is suggested here is an attempt, not the solution—or, following Wittgenstein—*a* structure, not *the* structure. The conceptual framework includes four main perspectives, which are structured in a number of specific aspects. The perspectives are not necessarily independent. Their differentiation is mainly motivated by analytical reasons. The *economic perspective* is aimed at discussing criteria that are relevant for judging costs and benefits related to the use of reference models. Among other things, it takes into account protection of investment, possible effects on information quality and competitiveness. The *deployment perspective* is focused on criteria that are relevant for those who work with the models. It stresses criteria such as comprehensibility, compatibility with other representations being used in an organization, availability of tools, etc. Reference models are artifacts that have been designed for a certain purpose. Also, they will usually be related to the analysis and design of information systems. The *engineering perspective* is aimed at evaluating a reference model as a design artifact that has to satisfy a specification—including the support for analysis and transformation. With respect to their claim for general validity, reference models resemble scientific theories. The *epistemological perspective* is aimed at evaluating reference models as the results of scientific research. For this purpose, it focuses on criteria for evaluating scientific theories as they are discussed in Philosophy of Science. In order to differentiate between conceptual models in general and reference models, features that are more specific to reference models will be marked as such. Note, however, that the borderline between a conceptual model and a reference model cannot be drawn precisely. The evaluation of a reference model depends also on its type—e.g., an object model, a data model, a business process model, etc. However, due to the limited space of this chapter, specific features of particular model types will be widely abstracted in form. The suggested criteria are intended to provide guidance for evaluating reference models. They do not imply a specific scale level. Most of them will allow for classification, some for applying an ordinal scale only, e.g., a Likert scale. If there is need for calculating aggregated evaluation measures, one could define corresponding higher order scales. However, this would cause a distortion of the evaluation result.

These suggestions are based on previous work on the evaluation of modeling languages. In Frank and Prasse (1997), a framework for the evaluation of object-oriented modeling languages is presented. It includes 33 criteria which are applied to a comparison of the UML and the OML (Firesmith, Henderson-Sellers, Graham,& Page-Jones, 1996). Frank (1998) suggests a multi-perspective framework for the discursive evaluation of modeling languages. Frank and Lange (2005) are aimed at languages for modeling business processes. It presents a comprehensive analysis of requirements for these kinds of languages.

The Economic Perspective

Both, the construction and the (re-) use of reference models chiefly depend on economic aspects. We will mainly take the viewpoint of a potential model user rather than that of a model developer. The type of user depends on the purpose a reference model is deployed for. Some will take a reference model as a foundation for developing software (*pre-development use*—referred to as type 1 in the table that illustrates the framework). For other users, a reference model serves mainly as a documentation of existing software (*post-development use*—referred to as type 2). Both pre- and post-development use can be applied to object models (or data models respectively) or business process models. In the case of post-development use, component models or application models—which would mainly focus on interfaces—are an option as well. In order to illustrate their deployment, they should be integrated with business process models. A third group of potential users is primarily interested in organisational or strategic issues (*business (re-) design*-referred to as type 3). Reference models that represent corporate strategies or organisations (e.g., business processes and organisational charts) are suitable for this category of use. In a particular case, different approaches to using a reference model may be combined, for instance pre-development use and business (re-) design. During the following discussion of economic aspects, we will at first abstract from these different types of uses. Only later, they will be taken into account again. For the evaluation of primarily economic issues, three main categories are suggested: *costs*, *benefits* and *protection of investment.*

Focus on Costs

While reference models are aimed at reducing costs, their use will cause costs, too. The following criteria serve to guide the estimation and evaluation of costs to be expected with the use of a reference models. Sometimes, they depend on features of a model that are the subject of other perspectives. The aspects are differentiated into three main categories: *introduction*, *transformation and analysis and maintenance*. Costs for introducing a reference model include acquisition or license costs as well as costs for training, adaptation, strategic re-design, organisational re-design and integration. Transformation costs are caused by transforming a model into other representations, such as implementation level documents.

Table 1. Introduction

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Acquisition	1, 2, 3	Cost of purchasing, licensing model Cost of inhouse development Economies of scale	Cost of inhouse development is often hard to determine; the more prospective users, the higher the economies of scale	Yes	
Training	1,2, 3	Familiarity of own staff with modelling language, terminology Inhouse modelling expertise Availability of training offers Overall complexity of model	Training costs depend heavily on the complexity of a model and the expertise of prospective users	Yes	Deployment -> understandability, attitude
Adaptation	1, 2, 3	Concepts that support adaptation in a safe and convenient way Availability of tools Cost of tools Cost of integrating with existing tools/ systems	Adaptation costs are often hard to estimate in advance; if available, one should look at costs caused in similar projects	Yes	Engineering -> Technical model features
Strategic Re-Design	1, 2, 3	Model recommends/ requires strategic adaptation Degree of change required	Strategic adaptation can be a chance, but also a threat. In any case, it will usually require major investments.	Yes	
Organisational Re-Design	1, 2, 3	Model recommends/ requires organisational adaptation Degree of change required	Depending on the degree of change, estimating related costs can require an extensive analysis.	Yes	
Integration	1, 2	Integration with existing models Integration with business partners Amount of integration required Compatibility of modelling concepts	Here, the focus is on modelling languages and on concepts (semantics) being used in already existing models.	No	Benefits -> Openness

Table 2. Transformation and analysis

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Suitability	1, 2	Modelling concepts allow for automatic transformation into implementation level documents Modelling concepts support required types of analysis If necessary: cost for adapting model for transformation/analysis	If the suitability of a model is not satisfactory, there is no chance to deploy tools; hence, high costs can be expected.	No	Engineering -> Technical model features
Tools	1	Availability of tools that feature transformation/analysis functions Cost of tools Cost of integrating tool with existing software development environment	For handling complex models, tools are almost mandatory.	No	Benefits -> Openness
Training/Support	1, 2	Skills required for performing transformation/analysis tasks available Cost of training Cost of external support	This is especially relevant for models that serve as vehicles for analysis or transformation.	No	Deployment -> understandability, attitude

Table 3. Maintenance

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Conceptual support	1, 2, 3	Concepts that support adaptation in a safe and convenient way	In case a model lacks these concepts, adaptation becomes risky and expensive.	Yes	Engineering -> Technical model features, Language features
Tools	1, 2, 3	Availability of tools that support model management (versions, users) Cost of tools	This includes multi-user access.	No	Engineering -> Technical model features Benefits -> Openness
Skills	1, 2, 3	Cost of internal skills Cost of external skills	These costs depend on the complexity of maintenance tasks and the spreading of a model.	Yes (external skills)	Benefits -> Openness

Analysis costs result from analysing a model with respect to a specific purpose, e.g., using a business process model for detecting bottlenecks or for running simulations. Maintenance refers to costs that are required for keeping a model up to date in the long run, which includes small and major changes.

Focus on Benefits

Using a reference model promises a number of benefits. While the ex ante quantification of these benefits has to face a number of severe obstacles, differentiating the overall potential benefit in a number of aspects can contribute to an evaluation that supports a comprehensive comparison with alternatives—such as developing a corresponding model on one's own or doing without conceptual models. Three categories are proposed for this purpose: *efficiency*, *flexibility* and *coordination/communication*.

Figure 4. Efficiency/effectiveness

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Software Development and Maintenance	1	Improvement of productivity Improvement of software quality Functionality and maturity of available tools Compatibility with existing abstractions Skills of software developers Willingness to use reference model	These are core promises of reference models. Evaluating them requires taking into account relevant requirements, model features and user competence.	Yes	Deployment -> understandability, attitude Engineering -> technical model features
Business/Management	1, 2, 3	Increased efficiency of affected business processes Cost reduction within business processes Support for specific decision scenarios Familiarity with model based decision making Willingness to use model within decision scenarios Improved customer-orientation	These are crucial criteria for the benefits to be expected from a model. They require a thorough analysis.	No	Benefits -> coordination Deployment -> understandability, attitude

Figure 5. Flexibility/integration

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Dependence from IT-vendors	1, 2	Number of relevant IT-vendors that support model Number of users Degree of customization Standardisation Level of industry commitment	Dependence does not have to be avoided, if there is a satisfactory level of trust.	Yes	Openness Protection of Investment -> spreading/commitment Deployment -> understandability
Openness	1, 2	Compatibility to relevant standards Integration with further reference models Coverage of possible future business models	This includes standards both for modelling languages and models.	Yes	Protection of Investment -> Spreading/Commitment
Expressive Power	1, 2, 3	Degree of (ontological) completeness of modelling language	This requires analysing the actual need for expressive power.	No	Engineering -> language features
Relationship to other IT Artifacts	1, 2	Concepts that foster integration/transformation into other relevant representations	For instance: ER to Relational Model, business process models to workflow schema	No	Cost -> tools

This aspect serves to analyse whether a reference model contributes to an organisation's ability to respond to change.

Coordination/Knowledge Management

A conceptual model should serve as a medium to foster communication between stakeholders with different professional backgrounds, such as software users, managers, software developers, consultants, etc. At the same time, it can be regarded as object and reification of corporate knowledge management: A conceptual model represents knowledge about a firm and supports people who want to learn how a company works. A reference model can provide additional support for coordination and knowledge management, since it may increase the number of people/institutions to communicate with, and it may incorporate knowledge from external sources that adds to the corporate knowledge base.

Table 6. Coordination/knowledge management

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Coordination	1, 2, 3	Helps to overcome communication barriers within company Fosters communication with external partners Improves coordination of business processes Fosters the establishment of inter-organisational coordination	While this is an aspect that applies to conceptual models in general, reference models promise to amplify the corresponding effects.	No	Deployment -> understandability Openness Protection of Investment -> Spreading/ Commitment
Knowledge Management	1, 2	Contributes to internal dissemination of relevant knowledge Supports development of relevant skills of employees Contributes to cross-organisational exchange of knowledge Incorporates relevant, external knowledge Decreases time to bring new employees, business partners up to date Contributes to a unified, enterprise-wide terminology	Makes knowledge available to people who formerly had no access	Yes	Deployment -> understandability, attitude

In case a more detailed analysis of benefits is required, the deployment of a reference model can be analysed in association with related business objectives. If, for instance, the reference model is an enterprise-wide object model, models of relevant business processes could be used to study the potential effects on important goals associated with these processes. If the reference model itself is a business process model, it could be evaluated using a model of the corporate strategy: The strategic goals can then be used to analyse the contribution of a certain process type. For an example of how to relate features of business processes to strategic plans, see Frank and Lange (2005).

Figure 7. Protection of investments

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Spreading/ Commitment	1, 2, 3	Number of organisations that use the model Number of vendors and service providers that support the model Standardisation of modelling language Standardisation of model	Corresponding statements of vendors should be tested thoroughly.	Yes	Benefits -> openness
Technological Change	1, 2	Independent from a particular technology Supports technologies that can be expected in near future	This requires identifying the core concepts of a technology.	No	Engineering -> technical model features

Focus on Protection of Investments

Taking into account that using a reference model can cause substantial investments, the question of how these investments are protected is a core issue.

The Deployment Perspective

The success of a reference model depends heavily on its users. This includes their ability as well as their willingness to deal with the model. Within this perspective, the framework includes the following aspects: *understandability*, *appropriateness* and *attitude*. In order to foster communication between the involved stakeholders, a model should be understandable. In other words, it should correspond to concepts, the prospective model users are familiar with. A reference model should stress an appropriate level of abstraction in detail—with respect to the purpose, a model is supposed to fulfil. If prospective users are not willing to make use of the model or if there are any objections against the model's usability, this lack of attitude can become a critical success factor. Therefore, it should be taken into account, even if it does not necessarily correspond directly to certain model features.

Figure 8. Deployment perspective

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Understand-ability	1, 2, 3	Elaborate structure for documentation (e.g., with design patterns) Comprehensive documentation Scenarios and examples Familiarity with modelling language Familiarity with terminology Intuitive access to graphical representation Views for different groups of stakeholders	A modelling concept is the more understandable, the more it corresponds to concepts/terms, an observer is familiar with.	No	Engineering -> explanation
Appropri-ateness	1, 2	Amount of support for purposes relevant for users Supports technologies that can be expected in near future	Implies requirements analysis	Yes	Benefits -> suitability
Attitude	1, 2, 3	"Not invented here"-syndrome Reputation of model developers Resistance to organisational change Cultural barriers	If resistance is to be expected, it can help to get developers involved in time.	Yes	

The Engineering Perspective

A reference model is a design artifact that can be regarded as a specification of possible solutions to a range of problems. From an engineering viewpoint, two questions are pivotal: Does the model fulfil the requirements to be taken into account? Is the specification suited for supporting the intended purposes of the model? To analyse these questions, four aspects are differentiated: *definition*, *explanation*, *language features, model features*.

Testing a model against requirements implies the requirements are to be made explicit in a comprehensive and precise way. Requirements include a definition of the intended application domains as well as a definition of the purposes to be satisfied. In the ideal case, these definitions should allow for deciding whether the model fits a particular application area or whether it supports a certain purpose. Note, however, that this does not only depend on the quality of the requirements documentation. Furthermore, every prospective user should

Figure 9. Engineering perspective

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Definition	1, 2, 3	Comprehensive description of intended application domains Comprehensive description of intended purposes	In both cases definitions should allow for deciding whether a reference model fits specific needs	No	Deployment -> understandability Epistemological -> general principle
Explanation	1, 2, 3	Assigning model elements to requirements Justification/substantiation of design decisions Discussing design compromises and resulting drawbacks Discussion of alternative approaches	An elaborated explanation of this kind is a tremendous support for model evaluation.	No	Epistemological -> general Principles
Language Features	1, 2, 3	Level of formalization, extensibility, supported conceptual views, integration of views, tool support, concepts to support the adaptation of models, concept to foster model integrity	The modelling language is essential for the engineering use of a model.	No	Benefits -> expressive power Epistemological -> critical distance
Technical Model Features	1, 2, 3	Formal correctness/consistency Model architecture Use of classes Use of generalisation/specialisation Use of modularisation/encapsulation	Here, it has to be analysed, whether the concepts provided by the modelling language have been used appropriately to achieve integrity and flexibility.	No	Language Features

know the requirements and purposes of the application he has in mind. In addition to merely defining the requirements, the model should also be explained in the sense that a potential user is supported in understanding and judging it. This includes an assignment of model elements to requirements as well as a substantiation of major design decisions that the model is based on. Often, design decisions require a compromise. This should be discussed including the resulting drawbacks. With respect to a modeling language, the following criteria are relevant: *level of formalization*, *extensibility*, *supported conceptual views*, *integration of views*, *tool support* and concepts to support the *adaptation of models*. Technical features of a model include *formal correctness*, *model architecture* and *adaptability*.

From an engineering point of view, adapting a reference model to individual requirements in a safe and convenient way is a core challenge. A key idea to accomplish adaptability of this kind is *abstraction*. It recommends differentiating between invariant parts of the model and those parts that are subject to change and adaptation. This differentiation can be reflected in the architecture of a model. In the ideal case, changing variant parts should not cause any side effects on other parts of the model. Abstraction requires corresponding concepts within the modeling language as well as their adequate use in the reference model. Important concepts to foster abstraction are classes, generalisation/specialisation and modularisation/encapsulation. The concept of a class allows for abstracting from single instances. As a consequence, changes can be applied to all instances of a class at one place. Generalisation allows for abstracting from special features of subclasses. Changes that are applied to a superclass are transparently effective in all subclasses as well. On the other hand, adding a further subclass does not affect the semantics of existing classes. Encapsulation is an abstraction of internal structures of a class (in case of modularisation: of a module). It allows for adapting the implementation of a class to individual needs without changing its interface.

The Epistemological Perspective

This perspective serves to enrich the evaluation of reference models with epistemological considerations. They are differentiated into four interrelated aspects: the *evaluation of theories*, *general principles of scientific research*, *critical reflection of human judgement* and *reconstruction of scientific progress*.

Reference models reveal similarities to scientific theories. Like theories, they are supposed to provide representations not just of a single instance (an enterprise, an application, etc.), but of an entire class. Also, they can be regarded as contributions to the body of knowledge within a certain domain of interest. Therefore, it makes sense to apply criteria that are used for the evaluation of theories to the evaluation of reference models. There is, however, one major difference between theories and reference models. A theory is aimed at describing the world as it is. Hence, a key criterion for assessing theories is truth, or rather: a certain concept of truth, such as the correspondence, coherence or consensus concept. A concept of truth is only of limited use for evaluating reference models, since they are usually aimed at intended systems or future worlds: They are not only descriptive, but also prescriptive. Nevertheless, also with reference models, the claim for truth cannot be entirely neglected-- we could speak of "relaxed truth": A reference model does not have to fit reality entirely; however, it should not contradict evidence. Hence, the descriptive parts of the model and the assumptions underlying the prescriptive parts can be evaluated according to the judgement of theories.

The correspondence concept of truth recommends testing a hypothesis against reality. This requires a precise description of the model and its intended applications as well as testing procedures that allow for comparing a statement with perceptions of reality. The coherence concept of truth recommends that a new hypothesis should be in line with an established body of knowledge, e.g., with research results and opinions found in acknowledged publications.

Applied to reference models, this implies that assumptions underlying the design of a model should not contradict accepted knowledge, e.g., established accounting principles (notice that this is just one notion of truth). From the viewpoint of truth as result of a consensus, emphasis is on discursive judgement by experts. This would recommend getting acknowledged people involved who should discuss and eventually confirm the assumptions a reference model is based on.

Despite the ongoing discussion on concepts of truth and corresponding research methods, there are three generic principles that allow for differentiating scientific research from other sources of knowledge: *abstraction*, *originality* and *judgment*. While not necessarily with the same rigour, they should apply to reference models, too. A high quality reference model should abstract from peculiarities of single instances and from changes that may occur over time. Abstraction, however, does not simply mean to arbitrarily fade out parts of the domain. Instead, abstraction should be made explicit and should include hints of how to turn it into a concrete description that applies to a particular case. With reference models, originality is hard to judge. Nevertheless it is certainly important. This is especially the case for reference models that result from scientific research (see "progress"). Judgement in science means that there has to be given comprehensive reason/justification for any hypothesis. For this purpose, one will usually refer to the preferred concept of truth and the related testing procedures. This can be applied to the descriptive parts of a reference model, too. With respect to decisions that motivate prescriptive elements of a reference model, this is different, because truth is not the issue. In order to provide reasons for design decisions, reference could be made to the accepted state of the art (following the coherence concept) or to discursive judgement by experts (following the consensus concept). In any case, judgement implies that every non-trivial assumption that design decisions are based on should be made explicit and reasons given for the choice.

Epistemology deals with the study of scientific judgements or, in other words, with the limits of human knowledge. Despite the ongoing debate, a critical or even sceptical evaluation of our perception and ability to judge prevails. There are many kinds of deception. With respect to the social sciences (or the humanities), perception and judgement are often biased by social/cultural constructions one is not entirely aware of. With respect to reference models, there is even more reason for epistemological scepticism. Reference models are linguistic artifacts: They are described using a language and—on another level of abstraction—they represent a language themselves. Although we are able to reflect upon language, for instance by distinguishing between object and meta-level language, our ability to speak and understand a language is commonly regarded as a competence that we cannot entirely comprehend. Therefore, any research that aims at inventing new "language games" (i.e., artificial languages and actions built upon them), has to face a subtle challenge: Every researcher is trapped in a network of language, patterns of thought and action he or she cannot completely transcend, leading to a paradox that can hardly be resolved.

Understanding a language is not possible without using a language. At the same time, any language we use for this purpose will bias our perception and judgement or, as the early Wittgenstein put it, "The limit of my language means the limit of my world" (Wittgenstein, 1981, §5.6). If one has to judge a reference model specified in UML and happens to dislike UML, an objective evaluation of the model will be hardly possible. Also, if a reference model of an accounting system makes use of terms that are different from those we use for accounting, it is very likely that we do not find it comprehensive—although it might be superior with respect to consistency or adaptability.

Figure 9. Epistemological perspective

Aspect	Relevant for type of use	Criteria	Comment	Specific to reference model?	Related to
Evaluation of theories	1, 2, 3	Precise description of core concepts with respect to corresponding real world concepts; precise description of underlying assumptions	Precision in this case does not require formalization. Instead, the description should allow for testing against reality.	Yes	Engineering -> definition Deployment -> understandability
Generic Principles	1, 2, 3	Abstraction Originality Judgement	Different from theories, judgement does not have to relate to truth.	Yes	Engineering -> explanation
Critical Distance	1, 2, 3	Subjective nature of underlying decisions Bias through familiarity with modelling language High degree of spreading may be mistaken for high quality	The main purpose of this aspect is to motivate a critical reflection on the constraints an evaluation has to face.	Yes	
Scientific Progress	1, 2, 3	Discussion of long-term goals of research Elaborate documentation of model with respect to generic principles and long-term research goals Comparison with alternatives	While this is mainly an aspect that is of concern for scientific research, developing a notion of progress in reference modelling can also help with evaluation in practice.	Yes	

While it seems impossible to entirely overcome these obstacles, they can be met with certain precautions. Everyone involved in the evaluation of a reference model should name the modeling languages he or she is familiar with as well as preferences for modeling languages and technical languages. Then, everyone should reflect upon the question how his or her language background could influence their judgement. This could contribute to a more critical distance and a more objective judgement.

Taking into account their similarity to theories, reference models are an ideal subject of design-oriented research. If reference models are regarded as results of scientific research, there is need for identifying or reconstructing progress in the field. This requires reference models to be compared with respect to their contribution to the discipline's body of knowledge. Usually, that does not happen. Hevner et al. suggest that an artifact "may extend the knowledge base or apply existing knowledge in new and innovative ways" (Hevner, March, Park, Ram, 2004, p. 87). They do not, however, discuss how this could be accomplished. While there is no objective measure of progress with regard to reference models, there is only one approach to foster the identification of progress: documentation and competition. Reference models need to be documented in a way that makes them comparable. This requires a common structure—comparable to structures being used for the documentation of design patterns. In addition to documentation, it is necessary that everyone who presents a reference model compares it thoroughly with existing similar models—which in turn demands the definition of design goals. Such elaborated comparative documentation could then be used for presenting the body of knowledge of the field and for reconstructing progress.

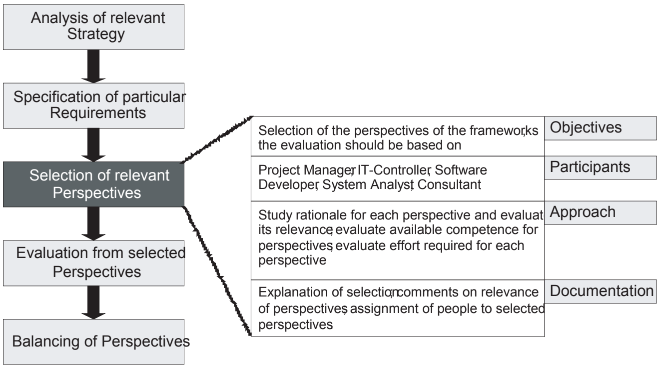
Applying the Framework: Outline of a Process

The rationale behind the framework presented here is to emphasize the need for considering a complex object such as a reference model from different perspectives in order to contribute to a more balanced judgement. The complexity of this task recommends the definition of a project. The generic process model that accompanies the framework is intended to guide the management of evaluation projects. The process consists of five major stages. Each stage is described using a common structure (see Figure 1).

The introduction of a reference model can be a major investment with implications that last for a long time. If this is the case, the process should start with a strategic analysis. It is aimed at studying the effect of a reference model on a company's competitiveness, which includes its ability to cope with change, to reduce cost, to improve its customer orientation, etc. Only if this stage results in a potential benefit to be expected from a reference model, does it makes sense to continue. In addition to the generic criteria presented in the framework, it is necessary to define concrete requirements that are related to the specific purpose of the model. The level of detail requirements analysis should cover depends on the assumed degree of peculiarities to be dealt with. Although the framework includes four perspectives, it might not be appropriate to use all of them in every project. The epistemological perspective especially requires a certain mindset and competence that is not always available. Also, one may want to do without the deployment perspective. Therefore, the perspectives that are regarded as relevant—and affordable, have to be selected. Furthermore, it is possible at this stage to modify the criteria assigned to a perspective. The following step is focussed on the evaluation of the reference model using the (customized) framework. Finally, the perspective-specific evaluations have to be integrated in order to accomplish an overall, balanced judgement.

The method, i.e., the framework and the process model, should not be mistaken as clear directions. Instead, they are only a guideline to structure the overall problem. The identification and interpretation of specific model features requires a competent and thorough analysis. For this reason, staffing is a key success factor. This is the case for those who are taking certain perspectives and even more so for those who moderate the process of balancing the perspectives.

Figure 1. continued



Conclusion

For a number of reasons, the evaluation of reference models is a challenging, yet important task. For a business firm, the deployment of a reference model is not only a substantial investment but it changes the process of software development and may affect the motivation and productivity of software developers. This recommends a thorough and elaborated evaluation as it is fostered by the method presented in this chapter. However, such an evaluation does not come for free. While one can expect the overall judgement to improve with the effort put into the evaluation, there will be a point when the cost caused by an evaluation overcompensates for its benefit. This implies that there is need for evaluating the evaluation from a controlling perspective. Especially in cases where no experiences with similar projects are available, estimating costs and benefits of an evaluation project in advance is hardly possible. Nevertheless, it is a good idea to reflect upon these aspects before defining a particular project.

While the motivation for evaluating reference models in academia is different, the challenges are similar. From a scientific point of view, it would make sense to study the effects produced by a reference model in the long run. Often, this is no option, since it would require resources that are not available to research institutions. If the evaluation is based on a method only, a scientific approach recommends a critical analysis of the method itself—again an evaluation of the evaluation. If this is done by deploying a further (meta) framework, one would finally produce an endless regress. Therefore, similar to controlling, a pragmatic solution is required. It could include the evaluation of the framework by peers and prospective users as well as empirical studies that focus on the use of the framework—or the method respectively.

References

Batini, C., Ceri, S., & Navathe, S.B. (1992). *Conceptual database design*. Redwood City.

Becker, J., Rosemann, B., & Schütte, R. (1995): Guidelines for modeling. In *Wirtschaftsinformatik, 37*(5), 435-445. [in German]

De Champeaux, D., & Faure, P. (1992). A comparative study of object-oriented analysis methods. In *Journal of Object-Oriented Programming*, *1*(5), 21-33.

Fettke, P., & Loos, P (2004). *Systematic survey of reference model: Preliminary results*. Working Papers of the Research Group Information Systems & Management. Paper 19, Mainz, Germany. [in German]

Fettke, P., & Loos, P. (2003a). Ontological evaluation of the semantic object model. In E.J. Sinz, M. Plaha, & P. Neckel (Eds.), *Modellierung betrieblicher Informationssysteme. Proceedings of the MobIS Workshop* (pp. 109-129). Bamberg, Germany. [in German]

Fettke, P., & Loos, P. (2003b). Ontological evaluation of reference models using the Bunge-Wand-Weber-model. In *Proceedings of the 9th Americas Conference on Information Systems* (pp. 2944-2955). Tampa, FL.

Fettke, P., & Loos, P. (2003c, October 13). Multiperspective evaluation of reference models: Towards a framework. In M. A. Jeusfeld & Ó. Pastor (Eds.), *Conceptual modeling for novel application domains: ER 2003 Workshops ECOMO, IWCMQ, AOIS, and XSDM*, Chicago. *Lecture Notes in Computer Science (LNCS)* (Vol. 2814, pp. 80-91). Berlin, Germany: Springer.

Fettke, P., & Loos, P. (2003d). Classification of reference models: A methodology and its application. In *Information Systems and e-Business Management*, *1*(1), 35-53.

Firesmith, D., Henderson-Sellers, B., Graham, I., & Page-Jones, M. (1996). *OPEN modeling language (OML). Reference manual.* Version 1.0, December 8, 1996, from http://www.csse.swin.edu.au/OPEN/comn.html

Frank, U. (1998). *Evaluating modelling languages: Relevant issues, epistemological challenges and a preliminary research framework.* Arbeitsberichte des Instituts für Wirtschaftsinformatik der Universität Koblenz-Landau/Germany.

Frank, U., & Lange, C. (2005). E-MEMO: A method to support the development of customized electronic commerce systems. *Information Systems and e-Business Management*.

Frank, U., & Prasse, M. (1997). *A framework for evaluating object oriented modeling languages: Exemplified by the examples of OML and UML*. Arbeitsberichte des Instituts für Wirtschaftsinformatik der Universität Koblenz-Landau, No. 7. [in German]

Frank, U., & Van Laak, B. (2003). *Requirements for business process modeling languages.* Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 34, University of Koblenz-Landau, Germany. [in German]

Goldstein, R. C., & Storey, V. (1990). Some findings on the intuitiveness of entity-relationship constructs. In F.H.Lochovsky (Ed.), *Entity-relationship approach to database design and querying* (pp. 9-23). Amsterdam, NL: Elsevier Science.

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Hitchman, S. (1995). Practitioner perceptions on the use of some semantic concepts in the entity-relationship model. *European Journal of Information Systems, 4*, 31-40.

Hong, S., & Goor, G. (1993). A formal approach to the comparison of object-oriented analysis and design methodologies. In J.F. Nunamaker & R. H. Sprague (Eds.), Information systems: Collaboration technology, organizational systems, and technology. *Proceedings of the 26th International Hawaii International Conference on System Sciences* (pp. 689-698). Los Alamitos.

Krogstie, J. (1998). Integrating the understanding of quality in requirements specification and conceptual modeling. *Software Engineering Notes, 23*(1), 86-91.

Lindland, O. I., Sindre, G., & Sølvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software, 11*(2), 42-49.

Mišic, V. B., & Zhao, J. L. (2000, October 9-12). Evaluating the quality of reference models. In A. H. F. Laender, S. W. Liddle, & V. C. Storey (Eds.), *Conceptual modelling. ER 2000 - 19th International Conference on Conceptual Modeling*, Salt Lake City, Utah (pp. 4484-498). Berlin.

Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. *Data and Knowledge Engineering, 55*(3), 243-276.

Monarchi, D. E., & Puhr, G. (1992). A research typology for object-oriented analysis and design. *Communications of the ACM, 35*(9), 35-47.

Moody, D. L., & Shanks, G. G. (1994). What makes a good data model? Evaluating the quality of entity relationship models. In *Proceedings of the 13th International Conference on the Entity-Relationship Approach (ER'94)* (pp. 94-111). Manchester, UK.

Mylopoulus, J., & Levesque, H. J. (1985). An overview of knowledge representation. In M.L. Brodie, J. Mylopoulos, & J. Schmidt (Eds.), *On Conceptual Modelling. Perspectives from Artificial Intelligence, Databases and Programming* (pp. 3-17). Berlin/Heidelberg, Germany: Springer.

Opdahl, A .L., & Henderson-Sellers, B. (1999). Evaluating and improving OO modelling languages using the BWW-model. In *Proceedings of the Information Systems Foundations Workshop* (Ontology, Semiotics and Practice). [digital publication]

Shanks, G., Tansley, E., & Weber, R. (2003). Using ontology to validate conceptual models. *Communications of the ACM, 46*(10), 85-89.

Süttenbach, R., & Ebert, J. (1997). *A booch metamodel*. Fachberichte Informatik 5/97, Universität Koblenz-Landau.

Weber, R. (1997). *Ontological foundations of information systems*. Melbourne: Coopers & Lybrand.

Wittgenstein, L. (1981). *Tractatus logico-philosophicus*. Englewood Cliffs, NJ: Routledge Kegan Paul.

Zelewski, S. (1995). Petrinet-based modeling of complex production systems. Vol. 9: Evaluating the concept of Petrinets. *Arbeitsbericht des Instituts für Produktionswirtschaft und industrielle Informationswirtschaft, Nr. 14*, University of Leipzig/Germany. [in German]