UNIVERSITÄT KOBLENZ · LANDAU

Institut für Wirtschaftsinformatik
Fachbereich Informatik
Universität Koblenz-Landau

ULRICH FRANK

EVALUATING MODELLING LANGUAGES: RELEVANT ISSUES, EPISTEMOLOGICAL CHALLENGES AND A PRELIMINARY RESEARCH FRAMEWORK

Dezember 1998

Arbeitsberichte des Instituts für Wirtschaftsinformatik Nr. 15

UNIVERSITÄT KOBLENZ · LANDAU

Institut für Wirtschaftsinformatik
Fachbereich Informatik
Universität Koblenz-Landau

ULRICH FRANK

EVALUATING MODELLING LANGUAGES: RELEVANT ISSUES, EPISTEMOLOGICAL CHALLENGES AND A PRELIMINARY RESEARCH FRAMEWORK

Dezember 1998

Arbeitsberichte des Instituts für Wirtschaftsinformatik Nr. 15

Die Arbeitsberichte des Instituts für Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i.d.R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

The "Arbeitsberichte des Instituts für Wirtschaftsinformatik" comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

Anschrift des Verfassers/
Address of the author:

Prof. Dr. Ulrich Frank
Institut für Wirtschaftsinformatik
Universität Koblenz-Landau
Rheinau 1
D-56075 Koblenz

Arbeitsberichte des Instituts für Wirtschaftsinformatik
Herausgegeben von / Edited by:

Prof. Dr. Ulrich Frank
Prof. Dr. J. Felix Hampe

Bezugsquelle / Source of Supply:

Institut für Wirtschaftsinformatik
Universität Koblenz-Landau
Rheinau 1
56075 Koblenz

Tel.: 0261-287-2520
Fax: 0261-287-2521
Email: iwi@uni-koblenz.de
WWW: http://www.uni-koblenz.de/~iwi

Institut für Wirtschaftsinformatik
Fachbereich Informatik
Universität Koblenz-Landau

Abstract

Evaluating and comparing modelling languages is a prerequisite for progress in the field of conceptual modelling. However, only little research has been dedicated to the investigation of appropriate evaluation methods. It is common practice that mainly those who design modelling language decide about the relevance of particular features. In this paper, we argue that such an approach does not satisfy common academic standards. The quality of a modelling language depends on a variety of tasks and potential users, some of which are beyond the scope of language designers. Therefore, the evaluation of modelling languages requires a cross-disciplinary approach. Furthermore, it has to be taken into account that the analysis of a language imposes severe epistemological problems. Against this background, we introduce a meta-framework for the evaluation of modelling languages. It includes a conceptual model to guide and structure multi-perspective evaluations.

1. Introduction: The Need for Evaluating Modelling Languages

It is widely accepted that conceptual models are a prerequisite for successfully planning and designing complex systems: They are a medium to foster communication with prospective users and they (should) provide a sound basis for system implementation. However, conceptual modelling often does not result in satisfactory models. It is a matter of experience that there is something like the quality of conceptual models. While the quality of models is of essential importance for the quality of the system to be developed, it is hard to judge. Quality depends on a number of aspects, some of which can hardly be evaluated using objective measures. To give a few examples: It is a delicate task to judge a model's relationship to the real world domain it is to render. Not only that it is impossible to identify the objective features of reality - unless you favour a naive realism, the quality of the relationship, hence the abstractions used within a model, also depends on the purpose of the model. Often, the purpose of a model will not be defined in a sufficiently precise way. A model should be *intuitive*. That means that its elements should directly correspond to the perceptions and conceptualisation of those who use the model. Patterns of perceptions and conceptualisation, however, are not only difficult to identify, they also vary from person to person and over time. In recent years there has been growing awareness of those problems. There are a few publications that suggest criteria/measures for evaluating the quality of conceptual models ([MoSh94], [Lin94], [KrLi95]) – like *Simplicity, Understandibility, Flexibility, Completeness, Integration, Implementability* ([MoSh94]). Usually, those publications either focus on the entity relationship model.

Taking into account the popularity of entity relationship models, it may seem reasonable not to bother with other modelling languages. However, a comprehensive analysis of quality in conceptual modelling requires to take into account other modelling languages as well. This is for various reasons. First, their is not doubt that the quality of a conceptual model is not independent from the language used to design it – this is the case both for the semantics and the (graphical) notation a modelling language provides. Furthermore, the dominance of the entity relationship approach is currently being challenged – mainly by object-oriented modelling methods. However, not only that those methods do not share a common terminology, they also come with a variety of modelling languages to express different aspects of system analysis and design (for instance: UML offers 10 different options for building diagrams [Rat97], OML even 12 [FiHe96]). At the same time, there is a growing demand in industry for standardised object-oriented modelling languages to allow for a better protection of investment. The OMG is currently in the process of establishing such a standard, apparently based on one proposal – UML – backed by a number of vendors (for more details see [Fra97a]). Standardisation, however, recommends to consider and evaluate options. An additional reason for looking at modelling languages can be seen in the growing number of approaches that suggest models to support organisational analysis and design – for instance models of business processes. Often the term ”enterprise modelling” is used whenever those models are to be integrated with information models. Specialised approaches to support the development of enterprise models offer a variety of – more or less precisely defined – modelling languages. Selecting a particular approach recommends to evaluate the modelling languages it offers against relevant requirements.

However, evaluating modelling languages is not only a matter of protecting investments in training and tools. From an academic point of view, it is a necessity for any discipline that deals with the design and use of modelling languages. The notion of scientific research is based on the idea of progress – in terms of growing knowledge and improving technologies. Progress, however, implies the existence of criteria that allow to discriminate between competing options – be it explanations of reality or artefacts that help to cope with it. A research discipline that does not seriously care about such criteria risks to sacrifice its identity. It may well be considered as a playground for inventing artefacts that serve mainly one purpose: to satisfy their inventors.

The work presented in this paper was motivated by the development of a set of modelling languages for enterprise modelling ([Fra97b]). We will first discuss the various aspects that have to be taken into account when evaluating modelling languages – together with related research topics and essential epistemological challenges to be faced. Against this background we will outline a method for evaluating modelling languages that is based both on a pragmatic framework and a few epistemological presumptions.

2. Evaluating and Designing Modelling Languages: A Cross-Disciplinary Endeavour

If we consider a modelling language not to be an end in itself, its evaluation requires to look at the purposes it is to serve and the people who are supposed to use it. The plethora of object-oriented modelling methods that have evolved during the last decade has caused a growing demand for comparing them. There have been several attempts to evaluate/compare some of those modelling methods ([Big97], [DeFa92], [HoGo93], [Hsi92], [MoPu92]). From our point of view, however, none of them is satisfactory. Firstly, they usually do not separate the evaluation of modelling languages from other parts of a method (such as process model, general heuristics etc.). Secondly, those investigations do not explicitly include the purpose/task and the potential users. Instead, it seems that they focus implicitly on one group of users – usually the group the authors belong to, namely system analysts or software developers. They also do not bother with a detailed analysis of the purpose the modelling languages are to be used for. This is different with evaluations – or requirements – that are common in computer science, especially in the area of software engineering. There, the purpose a modelling 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: [SüEb97]) 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 modelling languages do not allow to discriminate between a set of modelling languages that are complete and correct and share the same expressive power – leaving the judgement of simplicity to individual taste.

Often, promoters of object-oriented software development consider explicitly how certain modelling concepts relate to users' perception and cognitive style. While some point to the experience they have gathered with applying their approach in practice (for instance [FiHe96], p. 9, or [CeIb97], p. 1), others present their beliefs as if they had to be taken for granted: "People regard their environment in terms of objects. Therefore it is simple to think in the same way when it comes to designing a model." ([JaCh92], p. 42) Meyer does not even allow the objection that objects are sometimes hard to identify and to specify: ”... the objects are just there for the picking.” ([Mey89], p. 51) One may or may not agree with those statements. This is not relevant for our point of view. Instead, we want to point to the fact that some hypotheses which are of pivotal importance for designing and evaluating modelling languages are based on layman’s judgement. An evaluation, however, that reflects (biased) individual experience or repeats existing fairy tales and ideologies, can hardly be accepted as part of serious research. Without substantial knowledge about the way how people perceive and apply modelling concepts, it is hard to tell whether those concepts contribute to software quality - or to "disaster" ([Ber97]).

The analysis of language is the constitutive subject of linguistics. In particular, linguistic research deals with the structure and semantics of languages, the learning of languages and their impact on human thought and social interaction. This variety of aspects has fostered the evolution of a number of specialised linguistic disciplines as well as multi-disciplinary research efforts (for a comprehensive overview see [GrHa87], pp. 26). *Semiotics* is focusing on the symbols of a language: How are they perceived and learned, which are the associations they cause, etc. Apparently, the evaluation of a modelling language’s notation could be regarded as a part of this general research subject. *Psycho-Linguistics* analyses the influence of language on human thought and action. It is a common hypothesis, backed by intensive research that goes back to inter-cultural studies of language conducted by Whorf in the thirties, that language can be regarded as a general framework that guides, if not determines the thinking of those who use it. Wittgenstein has introduced the notion of ”language game” in order to emphasise that the analysis of language has to take into account the modes of its use as well as the associated actions in a particular context. Another important aspect of linguistics is the search for a *universal grammar*, a notion introduced by Chomsky. It aims at discovering the generic concepts of all human languages - concepts that every human being is given by birth. Again, it is obvious that relevant aspects of modelling languages can be regarded as genuine subjects of linguistic research: the impact a modelling language has on the intellectual activity of describing a certain domain, or the perception and use of its core concepts.

Other research areas of interest for the evaluation of modelling languages include cognitive psychology and communication theory as part of sociology, psychology and linguistics. Considering the various purposes a modelling language may serve, there may be even other disciplines to be taken into account. For domain specific modelling languages (for instance: languages to design communication networks, to design business processes or enterprise models) it may be helpful to incorporate specialised terminologies and abstractions commonly used in the domain of interest. Typically, the development of terminologies is an essential task of domain specific disciplines, as, for example, management science or organisation studies. Considering the various aspects of modelling languages, it is evident that a comprehensive evaluation cannot be accomplished by any of those disciplines alone. Instead, the analysis and evaluation of modelling languages requires a cross-disciplinary approach. While such a request is easy to make – and fairly popular these days - it remains nothing more than a *desideratum* as long as appropriate multi-disciplinary research projects are not established. Nevertheless, considering common practice in the design of modelling languages, two lessons can be learnt from the fact that we deal with a cross-disciplinary subject:

- Beware of layman’s judgements whenever there is the option to consult experts.
- Whenever you cannot avoid layman’s judgements make sure that you identify them as such.

3. Epistemological Challenges

So far there have been only few studies on the use of languages for conceptual modelling ([Hit95], [GoSt90]). While they are rather narrow focused and do not provide striking results, they indicate that further empirical research will be helpful to gain a better understanding of how people use modelling languages. Nevertheless, even with intensive multi-disciplinary research, we cannot expect to gain knowledge substantial enough to cover every aspect to be taken into account for the evaluation of modelling languages. This is mainly for three reasons: the dual nature of language as an expression and prerequisite of human thought, the attitudes of those who create and use modelling languages, and the various trade-offs that have to be dealt with during an evaluation process.

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 ([Lor96], p. 49). Therefore any research that either aims at analysing a language and its use or 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 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." ([Wit81], §5.6).

3.1 Complexity and Arbitrariness

Such considerations may seem to be of philosophical nature only. However, they characterise precisely one dilemma that evolves from the evaluation of modelling languages. Evaluation implies the knowledge of a modelling language. The more you know about the requirements of modelling, the more likely will you be able to understand existing concepts or to point at missing features. Modelling languages are usually designed by people who have gained an outstanding experience with the use of such languages. Usually they will design a language to fit their preferences. While both, the design and the evaluation of modelling languages require modelling experience, it is exactly this experience that will have a tremendous influence on the outcome of those efforts. Not only that this sort of bias hinders the invention of new modelling paradigms, at the same time – and this is an additional problem – we can assume that many people who are affected by modelling languages have perceptions and cognitive styles different from the language experts. Even for experts, the evaluation of a modelling language can be a frustrating task. This is due to the complexity of language specifications and design decisions that will always reflect personal preferences to a certain extent. Fig. 1, 2 and 3 show small parts of metamodels that are used to define modelling languages.

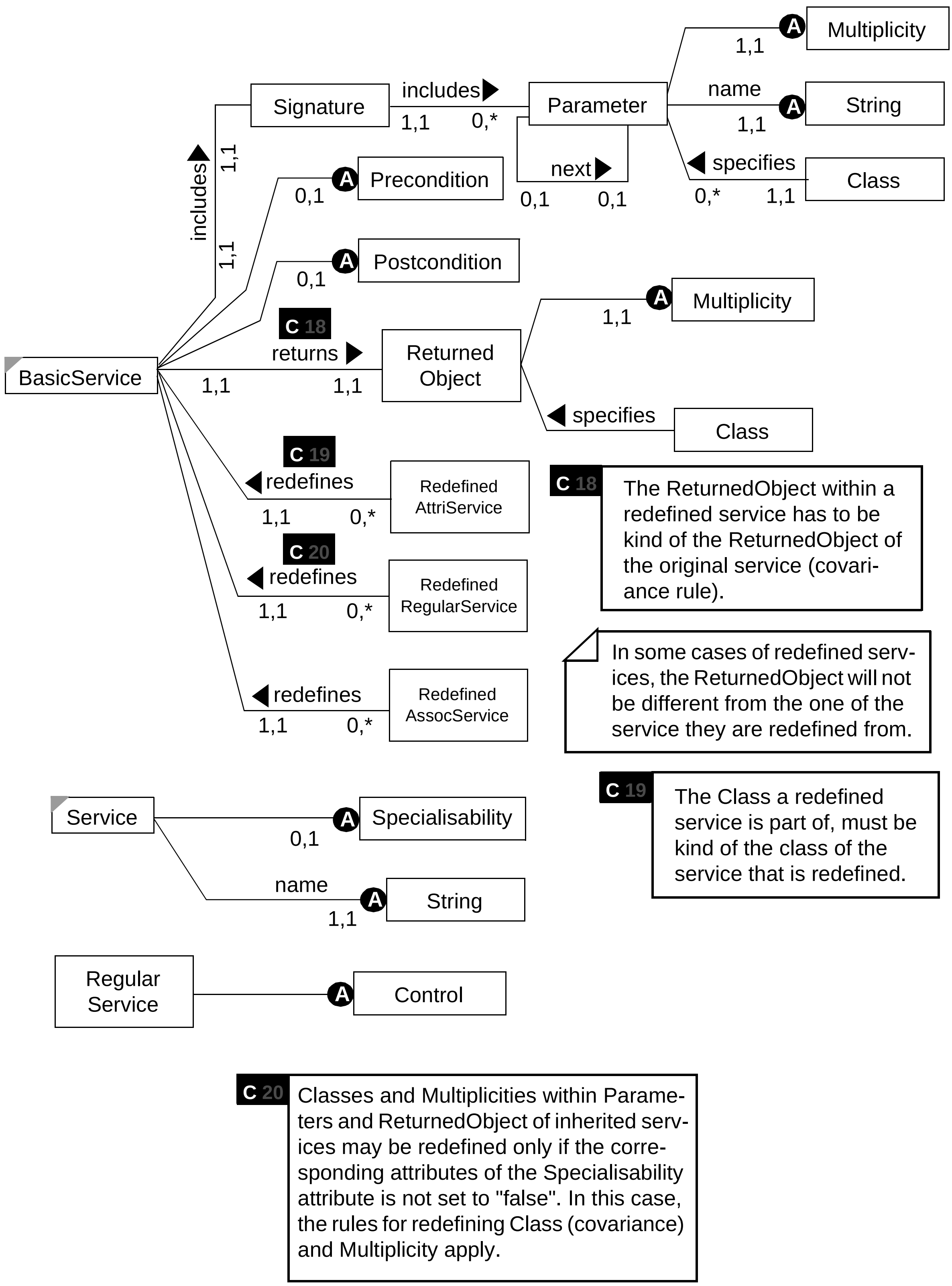

Fig. 1: Part of Metamodel of MEMO-OML ([Fra98b], p. 41)

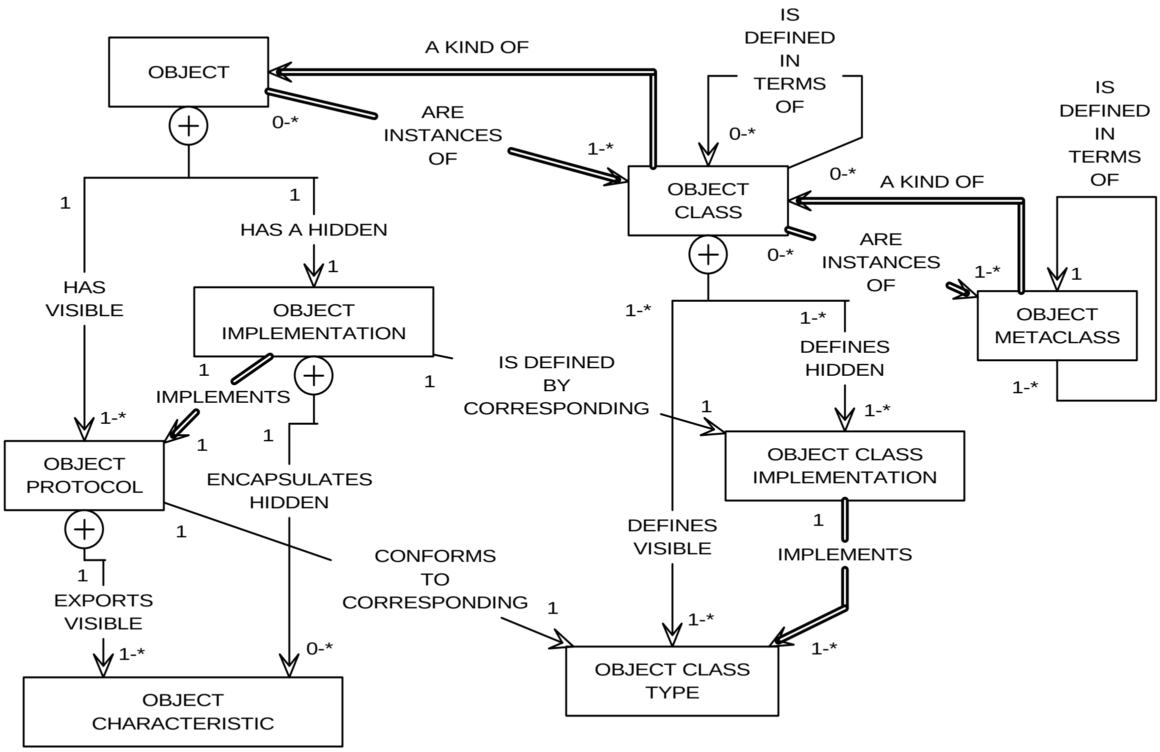

Fig. 2: Part of Metamodel of OML ([FiHe96], p. 21)

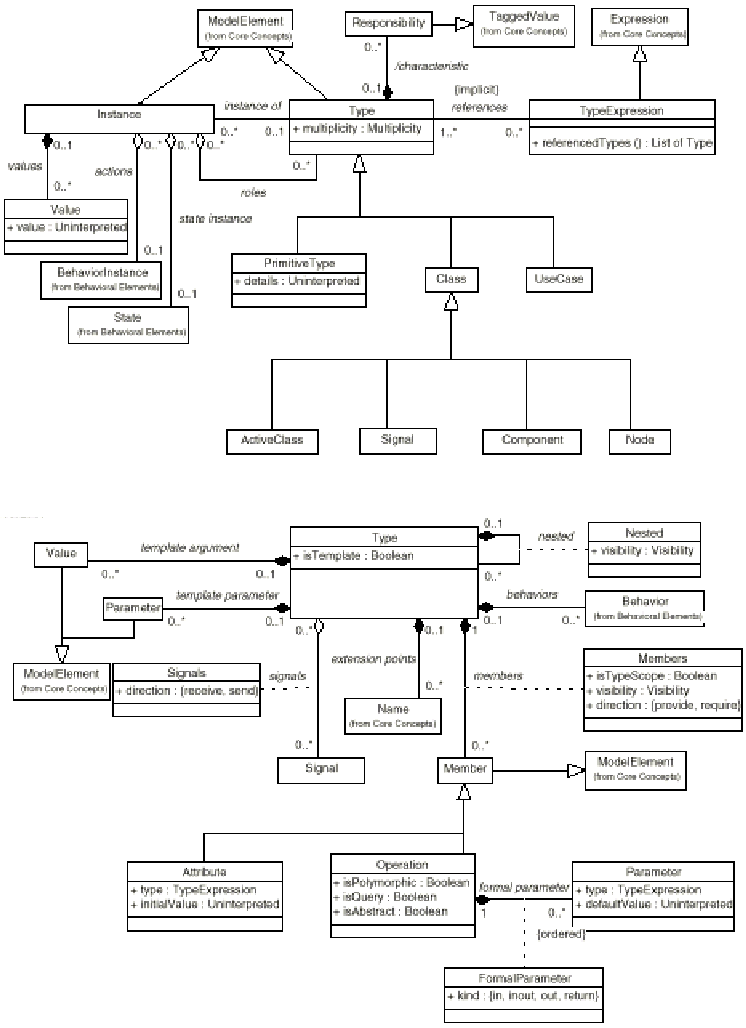

Fig. 3: Part of Metamodel of OML ([Rat97a], p. 24, p. 42)

The preferences and attitudes of language designers create yet another problem. The design of a comprehensive modelling language can be a remarkable effort. Language designers tend to regard ”their” languages as personal achievements which have to be – this is a matter of justifying such an effort – somehow better than existing languages. For this reason, language designers are – this is at least our (self critical) experience - extremely reluctant to ”sacrifice” their own creations for alternative artefacts created by somebody else. This resistance can be expected to be even higher than the one Kuhn ([Kuh70]) has discovered for theories in science: Different from scientific theories, modelling languages cannot be ”falsified” by comparing them against reality.

Finally, the fact that requirements are often not coherent implies the need for trade-offs which in turn impose another inhibitor of objective evaluation. Everybody who has designed a modelling language will know of decisions that require a compromise. For instance, the design of a metamodel used to specify a language often requires to choose from a set of alternative abstractions – each offering specific advantages and shortcomings. In addition to that, there are fundamental conflicts imposed by the variety of tasks and users. For instance: A modelling language should be easy to use. Its notation should be intuitive, which implies that it corresponds to the conceptualisations its users prefer. At the same time it should support the design and implementation of software. At some point, that requires to introduce formal concepts which are suited to be mapped to implementation languages – and which may be regarded as hard to understand by some of those who have to deal with a modelling language. Handling such trade-offs implies a certain amount of arbitrariness.

3.2 State of the Art

Compared to the relevance of conceptual modelling, there are hardly any publications that focus on the evaluation of modelling languages. The approaches to compare object-oriented modelling methods (i.e. [Big97], [DeFa92], [HoGo93], [Hsi92], [MoPu92]) usually do not provide a satisfactory evaluation of languages. Often they evaluate a particular language against a superset of concepts known from other modelling languages. It hardly happens that designers of object-oriented modelling languages discuss requirements a language should fulfil. This is especially the case for the semantic concepts a language should provide. Rumbaugh ([Rum96], p. 11) suggest a list of requirements the graphical notation of a modelling language should fulfil:

- Clear mapping of concepts to symbols
- No overloading of symbols
- Uniform mapping of concepts to symbols
- Easy to draw by hand
- Looks good when printed
- Must fax and copy well using monochrome images.
- Consistent with past practice
- Self consistent
- Distinctions are not too subtle
- Users can remember it
- Common cases appear simple
- Suppressible details

Despite their plausibility these principles are of little help for the specification of a concrete notation, because they remain on a rather abstract level.

As part of a comprehensive analysis of Petri nets, Zelewski ([Zel95]) has developed a framework to evaluate modelling languages. While his focus is primarily on Petri-Nets, the criteria he suggests can be applied to other modelling 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, concepts to coordinate tasks, sequential and parallel processes etc. Such a framework is certainly helpful for the description of a given language. However, it takes into account only part of the story. Firstly, there may be different approaches to specify concepts that serve the same or a similar purpose. Secondly, judging semantics only does not help with the evaluation of the notation - which can be of crucial importance for the value of a modelling language.

Referring to the philosopher Bunge, Weber recommends to regard the level of "ontological completeness" ([Web97], p. 94) as essential for the quality of a modelling language (he speaks of a "grammar"). A language is ontological 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 modelling language should provide in order to be ontological complete. They include concepts to express "things", "properties of things", "types", "states", "laws" (comparable to constraints), "lawful states" (comparable to invariants), events. Similar to Zelewski's framework, these criteria help with the description of a particular language. They are, however, only of limited use for the evaluation of a language. By applying his criteria to the Entity Relationship Model, Weber establishes that the ERM is not ontological complete. However, that does not come as a surprise to anybody who knows about the poor semantics that is provided by the ERM.

4. A Multi-Perspective Framework for Discursive Evaluations

The problems that have to be taken into account when evaluating a modelling language suggest that an objective judgement is not possible. That does not imply, however, that there cannot be any kind of useful evaluation. Against the background of our previous considerations, a reasonable evaluation process should satisfy the following requirements:

- Since there is no mature theory that would cover all aspects of modelling languages, it should be possible to *adapt* evaluation processes to changing knowledge.
- Due to the variety of tasks and users, an evaluation process should take into account *multiple perspectives* which includes a *cross-disciplinary* approach.
- It should also take into account the subtle *epistemological challenges* imposed by any analysis of language.

The generic conceptual model (fig. 4) reflects some of these aspects. It contains existing knowledge about relevant tasks and features of a number of languages. New tasks and features may be added over time. The description of existing tasks and features may be refined. However, this part of the framework is assumed to be relatively stable over time. The idea of multi-perspective evaluations is supported by introducing perspectives each of which corresponds to exactly one instance of a role. A person that participates in an evaluation may hold one or more roles. A perspective is focused on a set of relationships each of which links one of the tasks provided with the framework to a set of features that seem to be relevant for this task. The framework is to inspire the person that performs the evaluation to reflect upon the role which is associated with a particular perspective and to differentiate between facts and subjective judgement.

From a pragmatic point of view, there is not doubt that it will usually be too much an effort to instantiate and initialize the entire framework. Nevertheless, the framework serves an important function by allowing to judge the evaluation itself: The more perspectives, roles, tasks etc. have been neglected, the less relevant is the result of an evaluation.

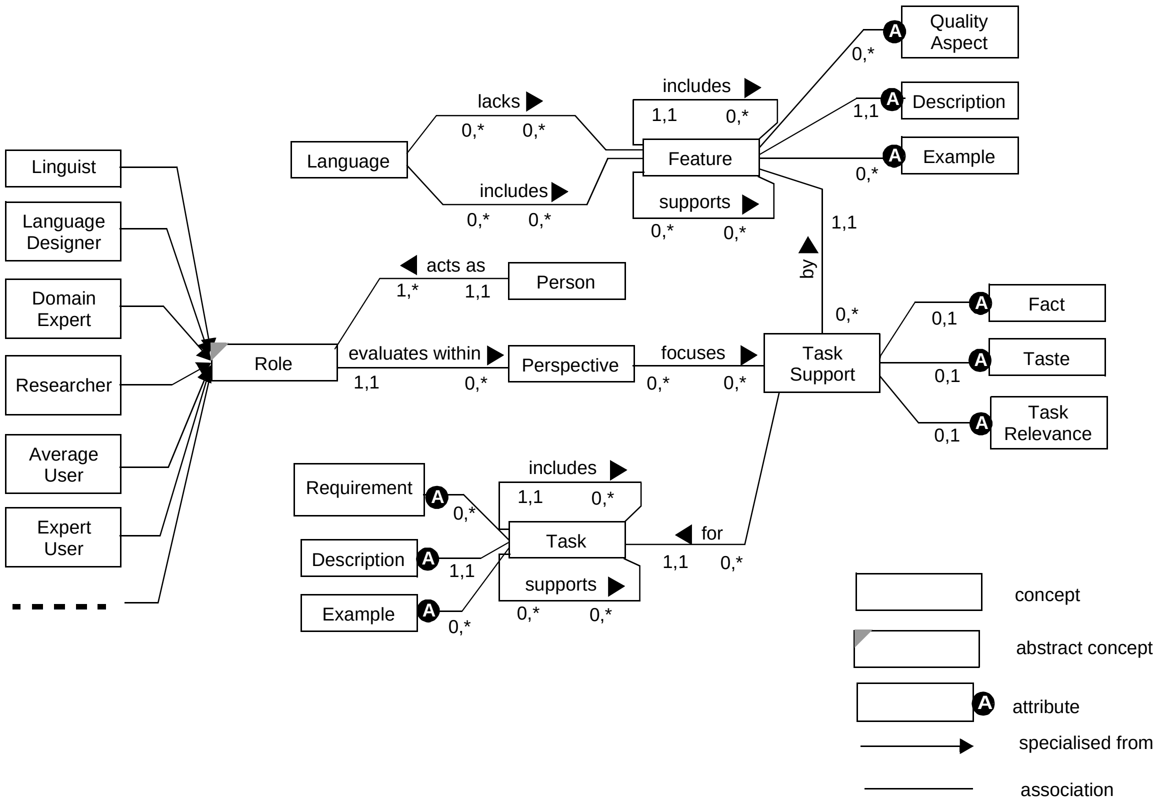

Fig. 4: Conceptual Model of the Research Framework

The language used to define the conceptual model is similar to the Entity Relationship Model. It is specified within the MEMO meta-metamodel [Fra98a].

The instantiation of the framework we have used to compare UML and OML ([FrPr97]) includes perspectives only that are related to the roles ”Language Designer” and ”Expert User” within the task of designing and implementing software systems. It provides an extensive list of tasks and requirements to be taken into account as well as detailed judgements of related language features. A small excerpt of the predefined part of the framework is shown in fig. 5. Notice that despite the complexity of this particular evaluation, it is not comprehensive in the end because it does not take into account relevant perspectives (like "domain expert" or "linguist") to a satisfactory extent.

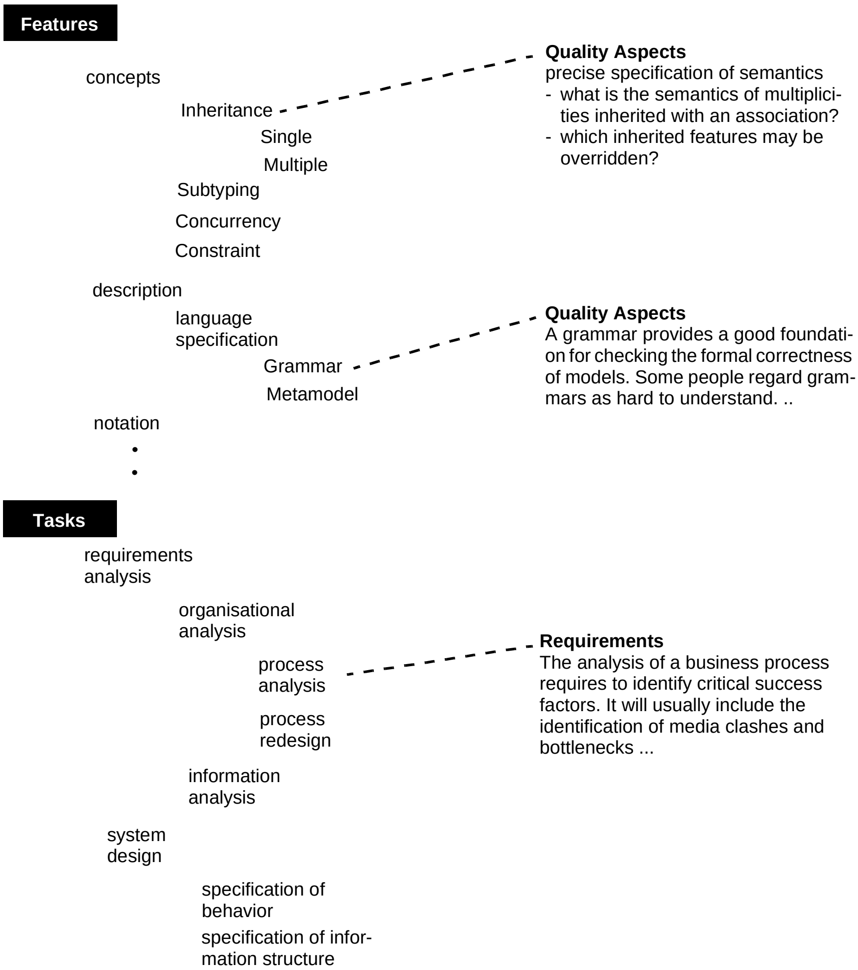

Fig. 5: Illustration of Criteria to evaluate selected Features

The framework is not intended to be used as a simple questionnaire. Instead, we would like to suggest its use as a guideline for structured *discourses*. This is motivated by the problems that accompany any analysis of a language. The idea of a rational discourse as it is used by some authors within the theory of science is based on the assumption that the exchange of thoughts is the only chance to overcome subjective perspectives. In order for a discourse to allow for that, it should fulfil a number of requirements, for instance ([Hab82], ([Ape81]):

- The participants should be seriously interested in achieving a commonly accepted result.
- They should never use arguments they do not believe in.
- They should try to transcend their subjective preferences and attitudes.
- They are interested in understanding the perspectives of other participants.
- They should have sufficient knowledge about the subject of the discourse.

There is not doubt that such an approach is not completely satisfactory after all. Although the requests seem to make sense, they are hard to check. Therefore the main problem that accompanies those recommendations is related to the selection of those who participate in a discourse. In other words: Who is going to decide which group of people is best suited to fulfil the requirements? In our case, one possible answer would be: Everybody who is directly affected by a modelling language - be it as a user or as subject of his professional field -should be entitled to participate in those discourses. While this is still a rather idealistic request, we think that there is hardly any other option – unless you are satisfied with ”oracles” provided by single experts. Considering the fact that there are subtle differences in how people perceive (artificial) languages and that we are all (to whatever) degree captives of the ”language games” we used to play, the reflection upon rules for rational discourses seems to be the only chance to get closer to ”objective” judgements (although we might never achieve them).

Although there is an obvious need for discursive evaluations, the principles recommended by Habermas or other philosophers remain rather abstract. *Action research* is a research paradigm that allows for a more pragmatic approach. The basic idea of action research is to overcome the traditional separation of researchers and those who are subject of research in the social sciences: The researcher is part of a team that works on an actual problem. Social research is regarded as a cyclic feedback process where different participants (including researchers) try to find common interpretations and solutions. The solutions will then be implemented, reviewed and eventually revised ("survey guided feedback"). Hence, action research does not only emphasize the cooperation between scientists and practitioners, it also aims at the implementation of acceptable solutions. Taking into account the fact that the informations systems discipline is commonly regarded as an applied science that aims at knowledge which is helpful for successful action, it is not surprising that Baskerville and Wood-Harper state: "The discipline of IS seems to be a very appropriate field for the use of actions research methods." ([BaWo98], p. 90) Action research stresses a hermeneutic approach. That implies a number of severe epistemological problems: What is the adequate language to describe the reflections upon an action research project? What are appropriate methods to evaluate corresponding statements? Nevertheless action research includes a few features that make it a good candidate for the analysis of conceptual models. The perception and understanding of a conceptual model includes complex psychological processes which depend on individual backgrounds and the context of a particular situation. A researcher who interacts with a user over a longer period of time should have a better chance to get a substantial impression of users’ perception than somebody who applies behaviouristic methods only. In addition to that, the evaluation of models and languages will usually require to learn concepts. Within action research projects, learning is subject and result of research. Since modelling languages are a medium to communicate, it is desirable to find evaluations most participants can agree upon. That in turn requires to overcome individual preferences and appreciate the perspectives of other participants. The interactive nature of action research helps to accomplish this goal. In order to involve people who use modelling languages, it is probably not a good idea to define an action research project that aims at the evaluation of a language at first place. Instead, it seems more appropriate to regard the language as an instrument for designing models. Therefore the project team should start with the common development of a conceptual model for a particular domain. Only after a preliminary model has been created, the researchers would have to shift the focus to the modelling language. Fig. 6 illustrates how a process to evaluate a modelling language through action research could be organized. [Fra+98] provides a detailed discussion of such an approach.

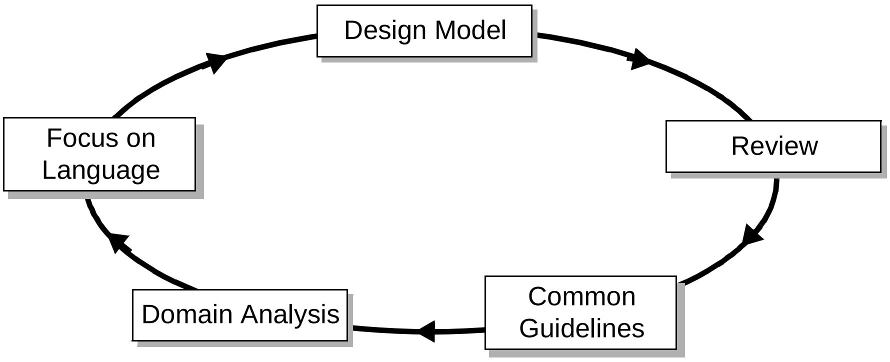

Fig. 6: Process Model for the Evaluation of Modelling Languages through Action Research

An extendable framework – together with rational discourses and action research – may provide a foundation for gradually developing balanced evaluations. But the framework does not only serve as a guideline for further empirical research. In addition to that, it can be used to document languages – similar to structures used for the documentation of design patterns. Finally, by gathering additional criteria, attitudes and judgements, it may eventually lead to a comprehensive theory of modelling languages.

5. Concluding Remarks

Evaluating modelling languages is a prerequisite for research disciplines that design and use those languages: Otherwise there is no chance for substantial comparisons and hence no idea of progress in the field. Leaving the evolution of modelling languages to the market is an option we do no want to accept. It sacrifices academic standards and prevents competition that is focusing on language quality only. The standardisation of an object-oriented modelling language that is currently promoted by the OMG provides a good example of the relevance of aspects other than those directly related to a language.[1]

Therefore, developing and maintaining common standards for the evaluation of modelling languages is an issue of pivotal importance. We have, however, to take into account that the evaluation of modelling languages cannot be accomplished only by those who design them or use them in an academic setting. Instead, it is necessary to engage discourses with others who are affected by those languages and who may provide additional insight from other research perspectives. This is a challenging task – not only because it depends on the collaboration of many people who are involved. But in the end there may be insights that could only evolve from a multi-perspective approach: "Der Gegenstand des Denkens wird fortschreitend deutlicher durch die Vielfalt der Perspektiven, die sich auf ihn richten."[2] ([BeLu80], p. 11)

1. No matter, how good or bad the UML will be in the end: Academic institutions were not permitted to submit proposals ([OMG96]).
2. To be translated as: “The subject of thinking becomes gradually clearer with the number of perspectives that are used to look at it.”

References

[Ape81] Apel, K.-O.: Transformation der Philosophie. Bd. 2: Das Apriori der Kommunikationsgemeinschaft. 2nd Ed., Frankfurt/M.: Suhrkamp 1981

[BaCe92] Batini, C.; Ceri, S.; Navathe, S.B.: Conceptual Database Design. Redwood City: Benjamin Cummings 1992

[BaWo98] Baskerville, R.; Wood-Harper, A.T.: Diversity in Information Systems Action Research Methods. In: European Journal of Information Systems. Vol. 7, 1998, S. 90-107

[Ber97] Berard, E.V.: Be Careful With "Use Cases". 1997 (http://www.toa.com/pub/html/use_case.html)

[BeLu80] Berger, P.L.; Luckmann, T. : Die gesellschaftliche Konstruktion der Wirklichkeit. Eine Theorie der Wissenssoziologie. Frankfurt/M.: Fischer 1980

[Big97] Biggs, P.: A Survey of Object-Oriented Methods. (http://www.dur.ac.uk/~dcs3pjb/survey.html)

[DeFa92] De Champeaux, D.; Faure, P.: A comparative study of object-oriented analysis methods. In: JOOP, No. 1, Vol. 5, 1992, pp. 21-33

[FiHe96] Firesmith, D.; Henderson-Sellers, B.; Graham, I.; Page-Jones, M.: OPEN Modeling Language (OML). Reference Manual. Version 1.0. 8 December 1996 (http://www.csse.swin.edu.au/OPEN/comn.html)

[Fra97a] Frank, U.: Towards a Standardisation of Object-Oriented Modelling Languages? Arbeitsberichte des Instituts für Wirtschaftsinformatik. Nr. 3, Koblenz 1997

[Fra97b] Frank, U.: Enriching Object-Oriented Methods with Domain Specific Knowledge: Outline of a Method for Enterprise Modelling. Arbeitsberichte des Instituts für Wirtschaftsinformatik. Nr. 4, Koblenz 1997

[FrPr97] Frank, U.; Prasse, M.: Ein Bezugsrahmen zur Beurteilung objektorientierter Modellierungssprachen - veranschaulicht am Beispiel von OML und UML. Arbeitsberichte des Instituts für Wirtschaftsinformatk, Nr. 6, Koblenz 1997

[Fra98a] Frank, U.: The MEMO Meta-Metamodel. Arbeitsberichte des Instituts für Wirtschaftsinformatk, Nr. 9, Koblenz 1997

[Fra98b] Frank, U.: The MEMO Object Modelling Language (MEMO-OML). Arbeitsberichte des Instituts für Wirtschaftsinformatik. Nr. 10, Koblenz 1998

[Fra+98] Frank, U.; Klein, S.; Krcmar, H.; Teubner, A.: Aktionsforschung in der WI - Einsatzpotentiale und Einsatzprobleme. In: Schütte, R. Siedentopf, J. Zelewski, S. (Eds.): Wirtschaftsinformatik und Wissenschafts-theorie. Grundpositionen und Theoriekerne. Arbeitsberichte des Instituts für Produktion und Industrielles Informationsmanagement. Nr. 4, University of Essen 1998, pp. 71-90

[GoSt90] Goldstein, R.C.; Storey, V.C.: Some Findings on the Intuitiveness of Entity Relationship Constructs. In: Lochovsky, F.H. (Ed.): Entity Relationship Approach to Database Design and Query. Amsterdam: Elsevier 1990

[GrHa87] Grewendorf, G.; Hamm, F.; Sternefeld, W.: Sprachliches Wissen. Frankfurt/M.:

Suhrkamp 1987

[Hab82] Habermas, J.: Theorie des kommunikativen Handelns. Vol. 1: Handlungsrationalität und gesellschaftliche Rationalisierung. Frankfurt/M.: Suhrkamp 1981

[Hew91] Hewlett Packard: An Evaluation of Five Object-Oriented Development Methods. Software Engineering Department, HP Laboritories. Bristol 1991

[Hit95] Hitchman, S.: Practitioner Perceptions on the Use of some Semantic Concepts in the Entity Relationship Model In: European Journal of Information Systems, Vol. 4, 1995, pp. 31-40

[HoGo93] Hong, S.; Goor, G.: A Formal Approach to the Comparison of Object-Oriented Analysis and Design Methodologies. In: Nunamaker, J.F.; Sprague, R.H. (Ed.): Information Systems: Collaboration Technology, Organizational Systems, and Technology. Proceedings of the 26th International Hawaii International Conference on System Sciences. Los Alamitos 1993, pp. 689-698

[Hsi92] Hsieh, D.: Survey of object-oriented analysis/design methodologies and future CASE frameworks. Menlo Park, Ca. 1992

[JaCh92] Jacobson, I.; Christerson, M; Jonsson, P; Overgaard, G.: Object-Oriented Engineering. A Use Case Driven Approach. Reading, Mass.: Addison-Wesley 1992

[Jon91] Jonsson, S.: Action Research. In: Nissen, H.E.; Klein, H.K.; Hirschheim, R. (Hg.): Information Systems Research: Contemporary Approaches and Emergent Traditions. Amsterdam: North-Holland 1991, S. 371-396

[KrLi95] Krogstie, J.; Lindland, O.I.; Sindre, G.: Towards a Deeper Understanding of Quality in Requirements Engineering. In: Iivari, K.; Lyytinen, K.; Rossi, M. (Eds.): Proceedings of the 7^{th} Conference on Advanced Information Systems Engineering (CaiSE ’95). Berlin et al.: Springer 1995, pp. 82-95

[Kuh70] Kuhn, T.S.: The structure of scientific revolutions. 2nd Ed., Chicago, Ill.: University of Chicago Press 1970

[Lin94] Lindland, O.I.; Sindre, G.; Sølvberg, A.: Understanding the Quality in Conceptual Modeling. In: IEEE Software, vol. 11, no. 2, 1995, pp. 82-95

[Lor96] Lorenz, K: Sprache. In: Enzyklopädie Philosophie und Wissenschaftstheorie. Hg. von J. Mittelstraß. Bd. 4, Stuttgart, Weimar: Metzler 1996, pp. 49-53

[Mey97] Meyer, B.: Object-Oriented Software Construction. Englewood Cliffs: Prentice Hall 1989

[MoPu92] Monarchi, D.E.; Puhr, G.: A Research Typology for Object-Oriented Analysis and Design. In: Communications of the ACM, Vol. 35, No. 9, 1992, pp. 35-47

[MoSh94] Moody, D.L.; Shanks, S.: What Makes a Good Data Model? Evaluating the Quality of Entity Relationship Models. In: Loucopoulos, P. (Eds.): Entity-Relationship Approach - ER'94. Business Modelling and Re-Engineering. 13th International Conference on the Entity-Relationship Approach. Berlin, Heidelberg etc.: Springer 1994, pp. 94-111

[MyLe84] Mylopoulos, J.; Levesque, H.J.: An Overview of Knowledge Representation. In: Brodie,M.L.; Mylopoulos,J.; Schmidt,J. (Eds.): On Conceptual Modelling. Perspectives from Artificial Intelligence, Databases and Programming. Berlin/Heidelberg: Springer 1984, pp. 3-17

[OMG96] OMG: Object Analysis & Design RFP-1, 6/6/1996 (http://www.omg.com)

[Rat97a] Rational: UML Semantics. Version 1.0, 01-13-97, 1997. (http://www.rational.com)

[Rat97b] Rational: UML Semantics. Version 1.1 alpha R6, 07-21-97, 1997. (http://www.rational.com)

[Rum96] Notation Notes: Principles for Choosing Notation. In: Journal of Object-Oriented Programming. May 1966, pp. 11-14

[SüEb97] Süttenbach, R.; Ebert, J.: A Booch Metamodel. Fachberichte Informatik, 5/97, Universität Koblenz-Landau 1997

[Web97] Weber, R.: Ontological Foundations of Information Systems. Melbourne: Coopers&Lybrand 1997

[Wit81] Wittgenstein, L.: Tractatus logico-philosophicus. London: Routledge & Kegan Paul 1981

[Zel95] Zelewski, S.: Petrinetzbasierte Modellierung komplexer Produktionssysteme. Bd. 9: Beurteilung des Petrinetz-Konzepts. Arbeitsbericht des Instituts für Produktionswirtschaft und industrielle Informationswirtschaft, Nr. 14, Universität Leipzig 1995