MODELLING THE RATIONALE OF METHODS

PÄR J. ÅGERFALK , KENNETH ÅHLGREN

E-mail: {pak | kan}@esa.oru.se
Phone: +46 19 303000

Örebro University, Dept. of Informatics (ESA)
SE-701 82 ÖREBRO, SWEDEN

Abstract — This paper presents a meta-modelling approach that allows the argumentative dimension of a method to be modelled by systematic treatment of modelling primitives concerning activities and goals, and their relationships. It is argued that the capture of a methods rationale is imperative if the purpose of modelling is to understand or learn a method, or to combine existing methods. Applications of the approach are presented and elaborated. Suggested applications are: the construction of context-sensitive help systems in CASE-tools; method reconstruction; verification of proposed method achievements; consistency checking of aggregated activities; and evaluation, adoption, implementation, and combination of existing methods.

Keywords*:* action, goal, intention, meta-modelling, method, methodology engineering, model, rationale, software engineering, systems development.

Formal reference to this paper is: Ågerfalk P J, Åhlgren K (1999). Modelling the Rationale of Methods. In *Managing Information Technology Resources in Organizations in the Next Millennium*, proceedings of the 1999 Information Resources Management Association International Conference. Hershey, PA, USA.

1. INTRODUCTION

Traditional approaches to meta-methods and methodology engineering are, in general, restricted to the conceptual aspects of the modelled methods [for example, Hey93, Kum92]. Methods can be viewed as normative conceptualizations in the sense that they direct the method user's attention by means of chosen modelling primitives, that is, by means of the concepts that are focused, or searched for, and that constitute elements in the method's various kinds of models. The primitives used in traditional, conceptually oriented meta-methods are thus restricted to concepts (entities), attributes, and relationships.

We believe that *what to focus* must be in parity with the *purpose* of the modelling (or model). An implication is that for some purposes, a conceptual meta-model is all that is needed. Such meta-model usage includes, but is not limited to:

- Implementation of methods in computerized tools (for example, CASE) during which a formal description of the method is needed.
- Conceptual overviews in, for example, method handbooks.
- Creation of languages to enable inter-subjectivity (defined below) regarding concepts and notation, for example, the unified modelling language (UML) [Boo99].

What a conceptual meta-model can never capture is the method's prescribed activities and their rationale. We understand a *method's rationale* to be the *argumentative dimension* of the method. The capturing of that dimension is imperative if the modelling purpose is, for example, to understand or learn a method. The rationale is also important when communicating methods, when constructing, reconstructing. or combining methods. and when evaluating methods.

What, then, constitutes a method? Basically, a method is a way of doing things, a kind of process, that is, an internalized successful pattern of activities, with the purpose of achieving some effect or goal. A method can also, in its externalized form, be viewed as a normative prescription for actions. Such a process can often be characterized as social interaction [Ber89] between developers, users and other systems stakeholders. In addition, a method, and the communication of its results, always implies the use of some concepts which can be expressed in some notation (textual or graphical, and more or less formal), which corresponds to the conceptual dimension mentioned above. This notion of method is quite common in the literature. For example, Rumbaugh [Rum95], as well as Blaha and Premerlani [Bla98], describes the concept of method in terms of concepts, notation and process. How, then, do methods exist? Methods can exist as actions or as results of actions (situational existence). Other "method existences" are methods as prescriptive guidelines (generic existence), and methods more or less implicit in various kinds of tools such as CASE-tools (implicit existence), cf. [Gol94a].

This paper deals with methodology-based analysis and design of methods, or methodology engineering. The focus is on methods for such engineering, that is meta-methods, and their possibilities for directing attention towards primitives concerning a method's rationale. Since the rationale can be viewed as the reasons and argumentation for prescribed actions, this paper's primary focus is on analysis of method activities (which constitute the method's process) and method goals (in the sense of, for example, intention) and, in particular, the relations between these two concepts. Note that various results of method actions might also be associated with goals. We do not present this approach as an alternative to conceptual method modelling but rather as a complement—they are both needed and serve different purposes.

A method's activities, together with some sequence constraints among them, form what we call the method's process. Such a process can be partially understood in terms of the questions asked to find the elements that constitute modelling primitives (which questions and how they are asked). If a method has been externalized and stated explicitly, for example, in a handbook, that statement can consist of several different components. It might consist of prescriptions regarding modelling concepts and their notation. It might consist of prescribed activities, questions and rules. Finally, it might consist of effects and qualities regarding the expected results. Such results can be formulated as either direct results or more indirect ones. Statements about expected results can, to some extent, be seen as arguments and thereby rationales for the method. Unfortunately, these arguments are not always made explicit within handbooks and other forms of description. On the other hand, sometimes the expected results are the only way in which the method is represented.

Methods represent knowledge and experiences [Gol94a] and, therefore, methods also represent rationale. A method user thus inherits both the knowledge and the rationale of the method creator. A method user is forced to rely on the method creator's judgements and normative statements. To be successful the method user, therefore, has to both understand and ultimately internalize the creator's arguments and perspective on the problem domain. As mentioned above, methods exist in several different forms. Two forms of method existence that deserve special attention are situational existence (or methods in use) and generic existence (or methods at a type level). Situational existence is the result of an interpretation, and an accomplishment, of the generic method by a method user. Such interpretation is a function not only of the generically existing method but also of the user's pre-knowledge, experience, values, rationale, tacit knowledge, and so on, cf. the infological equation as proposed by Langefors [Lan66]. The generically existing method is, in the same manner, an externalization of the method creator's expressible conceptualization of the subject area of the method.

We argue that a wider perspective on methods can be applied with the use of primitives additional to the traditional ones: concepts, attributes, and relationships. The added primitives suggested in this paper are action, actor, role, action object, goal, and intention, which all aim at understanding the rationale of methods.

2. THEORETICAL FOUNDATIONS

This chapter describes the main theoretical work that has influenced our work on rationale analysis. It is not an attempt to be an exhaustive description but rather to point out some important foundations and influences in order to serve as arguments for, and definitions of, the concepts we use.

Conceptual modelling aims at describing the meaning of concepts, and their relationships in order to get an accurate "image" of the studied piece of reality. Such an approach seems to be what Austin [Aus62] refers to as a "descriptive fallacy", assuming that language is merely used for descriptive purposes. Our approach aims at understanding the intentions of, and arguments for, activities. This approach emanates from the so-called "speech act theory" [Sea69]. Speech act theory emphases what people *do* while communicating, that is, actors using language to perform actions. In order to communicate, actors must share enough pre-knowledge to be able to conceptualize some part of the world in the same manner. Such sharing, or commonality is known as inter-subjectivity. Thus, communication is the process of one actor communicating something (acting), with some purpose, to another actor. That other actor then interprets the communicated "something" and reacts in response, so that a social interaction is taking place. We use the terms "internalize" and "externalize" when reasoning about methods in the context of social interaction. Internalizing means “making something to your own” and externalizing is the process of making something internalized communicable to others. However, sometimes the knowledge of an actor is difficult, or even impossible, to verbalize. One can do things but cannot explain how they were done, or even why. These are expressions of “tacit knowledge” [Mol96] and “hidden rationale” [Sto91], which are parts of what Polanyi refers to as “the tacit dimension”, or “we can know more than we can tell” [according to Mol96].

To know something and to be able to argumentatively discuss it in a dialogue with others constitute what we call rationality. Our notion of rationale is thus based on Goldkuhl’s [Gol94a] “argumentative rationality” which in turn is influenced by Habermas’s social-critical concept of rationality [Hab84]. In this view, well-founded knowledge is characterized as being [Gol94a] discursive, re-constructive, critical, and congruent. The discursive dimension is a part of all the others and discursive rationality means that knowledge is socially transferable by the use of arguments in inter-subjective dialogues.

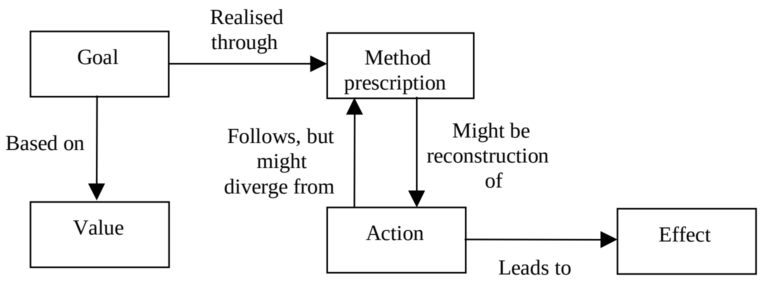

Fig. 1. Method as action

To focus means to direct attention to certain kinds of phenomena (or kinds of terms, to be precise). Some of these are recognized as primitives, that is, kinds of modelling elements. This means that the focus is determined by the questions asked to analytically abstract the phenomena. Methods include prescriptions that direct attention towards such primitives. The prescriptions are, in some sense, realizations of the method’s goals, which are always based on some values. The action prescriptions are then instantiated in actions during method use, which lead to an effect. On the other hand, if actions are performed, and observed to be “good” in some sense, they might eventually become institutionalized [Ber89] into a method (situational existence). Such institutions might then be reconstructed into methods at a generic existence level. Figure 1 illustrates how these concepts are related.

3. ACTIONS, GOALS AND RATIONALE

The proposed analysis focuses on three activities that can be logically separated, even though they are typically performed in parallel (see Fig. 2). The first activity, *activity analysis*, is concerned with the methods process: activities, actors, prerequisites, results, and possible sequence restrictions. The second activity, *goal analysis,* is concerned with the method’s goals. Here, the concept of “goal” is used in a broad sense and includes notions such as intention, reason, value, and effect. The third activity, *rationale analysis*, is concerned with the relations between the primitives used in the other two activities. The real essence of the modelling is reached, motivated, and illustrated by the applications in which it may be used, that is, when combining the analysis results.

Method actions, and thereby intentions (which in turn are underpinned by rationale), exist on different levels of abstraction, just as do methods *per se*. Method user actions, based on prescribed activities, are intentional at the situational level of method existence. The method creator, on the other hand, had some intention in mind when stipulating the method actions in the first place. Therefore, the method actions are also intentional at the generic level of method existence. In this paper, we will pay attention mainly to the generic level of method existence. It is, however, important to distinguish between the two levels, and in some cases the relation between them is the main object of study. The latter is the case, for example, when reconstruction of a method is taking place. Reconstruction is the process of externalizing a method and generalizing it from situational to generic existence. This is often done with the purpose of making implicit methods communicable in the form of handbooks or tools etc., or as part of method integration efforts.

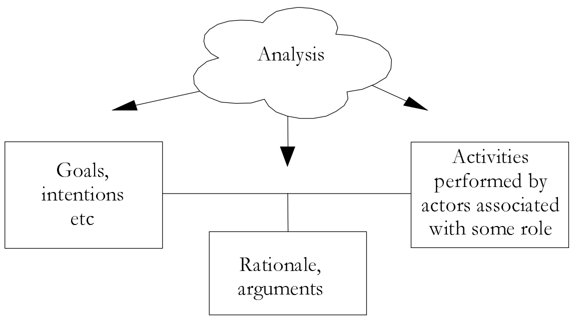

Fig. 2. An application as analysis context

To summarize, an actor associated with a role performs some action that has some outcome. Both the action and the outcome are associated with some intention that in turn is related to one or more goals.

One can argue that the three analysis activities mentioned are not really analysis but rather data collection. The collected data is then analysed in what we call different *applications* or *contexts*. We do, however, retain the term analysis to be consistent with the names of the activities in MA/SIMM from which the first two are chosen.

Throughout this chapter we will use a brief example of our approach. The method we have chosen to analyse is the Object Modelling Technique (OMT) as described by Blaha and Premerlani [Bla98]. The specific part of OMT we will consider is the creation of the object model, that is, the first phase of the method. The activity analysis is performed on “Listing tentative classes”, one activity within the object-modelling phase [Bla98, pp. 127-129]. Note that this example is not given with any special purpose, or application, other than to show how the analysis might be performed and documented.

3.1. Activity Analysis

Activity analysis is performed according to the corresponding method component in the SIMM-family of methods. Goldkuhl and Ågerfalk [Gol98] present activity analysis in the context of requirements engineering and Goldkuhl and Fristedt [Gol94b] discuss its use within methodology engineering.

During activity analysis, different method actions and their relations to each other are described. The relations are constituted by sequence restrictions between activities and the possibility that one activity requires the result of another activity as input. In conjunction with activity analysis, the different actors, the performers of method actions, and other stakeholders are identified and allocated to the actions they perform or to the action objects that they may represent. Usually, the role name is used but sometimes the name of the actor is more appropriate, for example, when there is a need to distinguish between different actors sharing the same role.

To summarize, activity analysis means analysis of:

- Actions and action structure.
- Performers of actions, that is, actors associated with some role.
- Results and prerequisites of actions (or action objects), which may be either in the form of information or material (which includes, for example, documents and human resources).

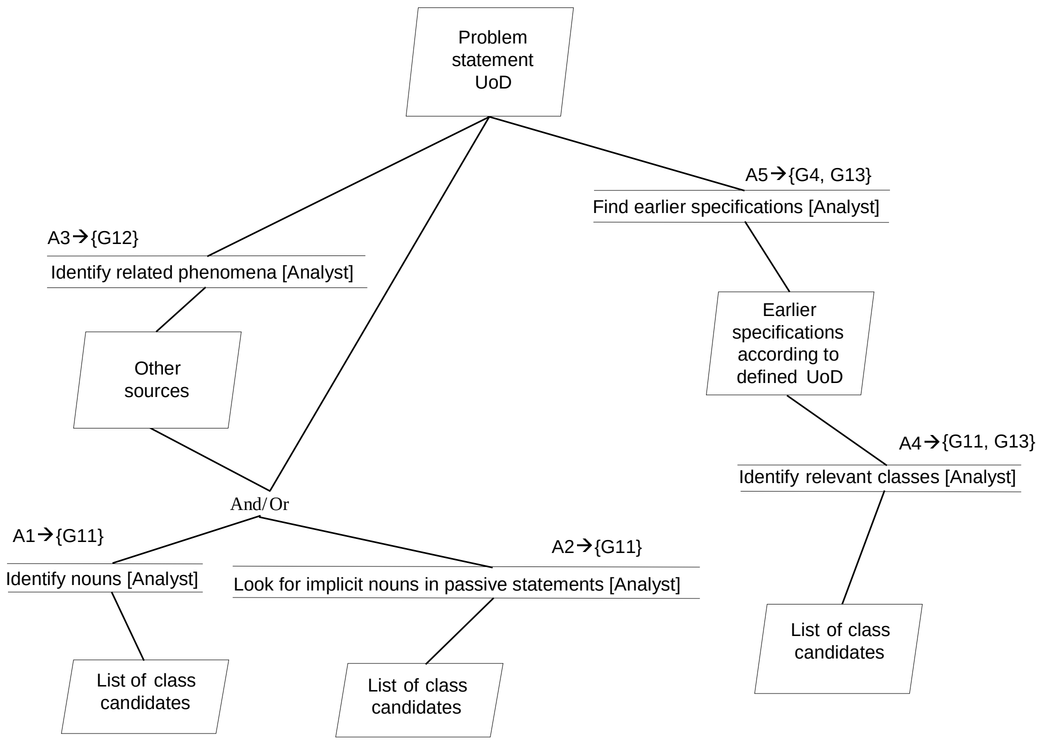

Fig. 3. Example notation of activity analysis

Figure 3 shows an example notation of activity analysis extended with mappings from activities to sets of goals. In the model the *G*s identify goals and the *A*s identify activities.

One important aspect of activity analysis is that it is performed contextually rather than compositionally [Gol92]. One strength of the contextual approach, among others, is that it allows analysts to model phenomena in the contexts in which they are used, without having to worry about compositional restrictions. The activity analysis is preferably documented with Action diagrams as shown in Figure 3 (see also, for example, [Gol92, Gol98]).

3.2. Goal Analysis

Goal analysis, like activity analysis, is performed according to the corresponding method component in the SIMM-family of methods. Goldkuhl and Röstlinger [Gol93] introduced goal analysis as part of Change analysis and Goldkuhl and Fristedt [Gol94b] discuss its use within the context of methodology engineering. It is important to notice that our use of the term "goal" includes other notions associated with intentional action such as intention, motivation, reason, effect, and purpose.

During goal analysis, the goals of the method are identified and elaborated. One important issue is the analysis of relations between goals in goal/sub-goal hierarchies (or networks). A sub-goal is, by definition, a goal but it is also a means of achieving the higher goal to which it is related. In a goal hierarchy there can also be goal conflicts, with one goal contradicting another.

To summarize, goal analysis means analysis of:

- Goals.
- Relationships between goals in goal-hierarchies (means and contradictions).

Figure 4 shows an example notation of goal analysis (a goal graph) and Figure 5 shows the corresponding goal statements in a goal definition table.

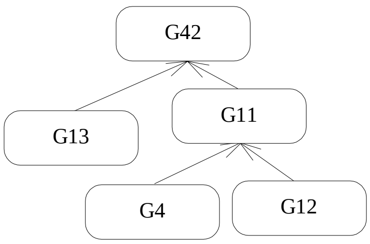

Fig. 4. Example notation of goal analysis

G4	Definitions reused from previous efforts.
G11	Class candidates found.
G12	All available sources of information considered.
G13	Integration with earlier applications enabled.
G42	All possible class candidates found.

Fig. 5. A goal definition table

The goal analysis is preferably documented in a Goal graph in conjunction with a goal definition table, as shown above in Figures 4 and 5 [c.f. Gol93].

3.3. Rationale Analysis

During rationale analysis the actors, roles, prerequisites, results and activities identified during activity analysis are explicitly related to goals identified during goal analysis. For the purpose of the analysis it is not necessary to distinguish between all modelling primitives from activity analysis. Therefore, we group all such instances together to form a set called the *activity set*. The rationale analysis thus aims at identifying a relation from the activity set to the set of goals. This analysis can be performed in two different ways. Either the elements of the activity set are iterated and related to goals, or the goals are iterated and related to activities. Normally these two approaches are both used alternately during the analysis. Sometimes the activities are well described and understood and it is straightforward to start with them, and perhaps identify new goals. In other cases, all that is known are perhaps vague goals with no intuitive relations to activities. Often, the purpose of the analysis implies the most convenient way to work (see the applications below).

The known rationale of the example can thus be modelled as a relation $R \subseteq A \times G$ where A={A1, A2, A3, A4, A5} is the set of activities and G={G4, G11, G12, G13, G42} is the set of goals, such that R={(A1, G11), (A2, G11), (A3, G12), (A4, G11), (A4, G13), (A5, G4), (A5, G13)}.

3.4. Applications

The information gathered during analysis of activities, goals, and rationale can be used in several different contexts; we describe some important ones below. Note that this is not meant as an exhaustive list but rather as examples of applications of our approach.

3.4.1. Context-sensitive help systems

One possible application is the creation of more usable context-sensitive help systems in, for example, CASE-tools. The system can keep track of the current activity and keep relations to goal descriptions as meta-data. This way the system can suggest different action patterns and motivate the different proposals argumentatively. Today, most context-sensitive help systems are focused on what to do in a particular activity but frequently fail to motivate the proposed actions.

3.4.2. Method reconstruction

Reconstruction of a method, as described above, is the process of generalizing a successful pattern of activities to an externalized representation. In order to communicate such a method, it is quite common to write a method handbook. The goals that are related to the activities can then be used to motivate those activities and thereby serve as a rationale for the method. It has become increasingly popular to augment method handbooks with a meta-model, for example, UML [Boo99] and OOA [Coa91]. Such models could be more argumentative with the use of our approach.

3.4.3. Verification of proposed achievements

Statements about the (positive) effects of using a particular method often accompany method descriptions. Such “political” statements can typically be regarded as goals. By verifying that every such goal is related to at least one activity it is possible to verify that they are made operational in the method.

3.4.4. Consistency checking of aggregated activities

Activities are often made up of other activities and thereby form activity aggregates. Examples of such aggregates are method phases like “object modelling” [Bla98]. Such phases can be compositionally divided into smaller activities, which in turn can be recursively divided into even smaller activities. Each activity aggregate can be related to one or more goals. The same holds for all activities that are part of the aggregate. It is possible to verify that all activities within the same activity aggregate are related to goals that are either related to the activity aggregate as a whole or to a sub-goal of such an activity. Such analysis can be used to verify that method activities are consistent with regard to goals. It can also be used during method construction to decide whether a proposed activity should be a part of a particular phase or step. If the proposed activity's goals do not appear in the phase's goal hierarchy it is doubtful that it should be a part of that phase at all.

Consistency checking can easily be formalized, in a mathematical sense, and automated by a computerized tool.

3.4.5. Evaluation, adoption, implementation and combination

Evaluation of methods is always done with some purpose. One common purpose should be a need to choose a method for adoption in a project. Such adoption always includes implementation of the method in a particular organizational context. By understanding the goals of the organization and comparing them to the proposed method, or methods, a better match can be reached. The goals could then be compared based on the arguments used.

At times. one single method does not satisfy the needs of a particular development situation, for example, when one method is used in one development phase and that method does not cover the following phases. At other times. more than one method is needed even in the same phase. In both cases, it is important to understand both what to do and why that should be done, in order to combine methods.

4. RELATED RESEARCH

Nilsson [Nil99] presents a meta-modelling approach used to verify that it is possible to integrate two or more methods. The modelling primitives used are *intentions*, *concepts*, and *ways of working*, which result in three different meta-models. The meta-models within the same categories are compared and integration possibilities judged by meta-model matching. If, for example, some goals are contradictory then integration is not recommended. If, on the other hand, all goals are supplementary or corresponding, it is possible to integrate the methods. The approach assumes that all method goals are made operational or that the ways of working (the action structure) really lead to stipulated goals. Our approach could be used to verify such assumptions, which would make Nilsson's approach more convincing.

Another approach to method assembly is proposed by Brinkkemper et al. [Bri98]. Their approach uses a three-dimensional classification scheme to categorize method fragments. Even though it is comprehensive, their categorization does not include the intentional dimension of methods.

Our theoretical influences are much the same as those that constitute what is known as the language action perspective (LAP) on communication modelling. Flores and Ludlow [Flo80] as well as Goldkuhl and Lyytinen [Gol82] made early contributions to LAP and Winograd and Flores [Win86] carried out seminal work. LAP emphasizes what people do while communicating and it has been used and elaborated in various research efforts, most notably in business modelling. One meta-method that builds explicitly on LAP is MA/SIMM [Gol94b]. Hence, we are strongly influenced by MA/SIMM and its contributions to understanding the action dimension of methods. We do, however, believe that the analysis, as proposed by Goldkuhl and Fristedt [Gol94b], should be carried one step further. Such an extension is not only implied as a consequence of the perspective of systems development as a social phenomenon, it is also needed in order to understand a method's rationale.

Tolvanen et al. [Tol93] discuss a methodological approach based on three levels of information architecture named GOPRR. The levels are named the ISD level, the ISD meta-level, and the ISD meta-meta-level. Their approach aims at capturing the dynamic aspects of methods as well as the conceptual. It does not, however, capture the argumentative dimension.

Design patterns [Gam95], and more recently analysis patterns [Fow97], are attempts to externalize tacit knowledge within a particular domain. Their approach aims at reuse and inheritance of actions developed by more experienced analysts and designers. We argue that we should go one step further and analyse the rationale behind such patterns in order to achieve better understanding of them. With such understanding we can be more successful, by reaching higher precision in interpretation when internalizing the patterns.

Nimal Jayaratna [Jay94] holds as a high ideal that the "whys" of a method are important when discussing and evaluating methods. We do, most certainly, identify ourselves with that ideal. What we propose is a systematic, and theoretically justified, approach to modelling and analysing those "whys", that is, what we believe are the main constituents of a method's rationale.

5. CONCLUSIONS

This paper has discussed an approach to meta-modelling that covers the argumentative dimension of methods. It is important to notice that the approach presented is meant as a complement, and not as an alternative, to conceptual approaches. The approach directs attention to activities and goals, and their relationships.

However, it is not only actions that are related to intentions and goals. Concepts used in a method are used for some reason. The same is also true for notation, documentation, and so on. The existence of such relationships is implied by the fact that all method aspects are actually results of action at the meta-level, actions performed by the method creator. It is, however, beyond the scope of this paper to discuss the use of rationale in conjunction with such non-activity-related aspects of methods.

The applications presented suggest that the proposed kind of modelling depends on the actual application context. We cannot really tell how to perform the analyses without some specific purpose. The suggested applications are: construction of context-sensitive help systems in CASE-tools; method reconstruction; verification of proposed method achievements; consistency checking of aggregated activities; and finally (and perhaps most importantly), evaluation, adoption, implementation, and combination of existing methods.

The proposed approach should make at least parts of the tacit dimension visible.

REFERENCES

[Aus62] J. Austin. *How to do things with words*. Oxford University Press (1962).

[Ber89] P. Berger and T. Luckmann. *The social construction of reality*. Anchor Books, New York (1989).

[Bla98] M. Blaha and W. Premerlani. *Object-oriented modeling and design for database applications*. Prentice Hall (1998).

[Boo99] G. Booch, J. Rumbaugh and I. Jacobsson. *The Unified Modeling Language User Guide*. Addison-Wesley Longman, Inc. (1999).

[Bri98] S. Brinkkemper, M. Saeki and F. Harmsen. *Assembly techniques for method engineering*. In proceedings, 10th Intl Conference on Advanced Information Systems Engineering (CAISE'98). Pisa, Italy, June 8-12 (1998).

[Coa91] P. Coad and E. Yourdon. *Object-oriented analysis*. Yourdon Press, Object International, Inc. (1991).

[Flo80] F. Flores and J. Ludlow. *Doing and speaking in the office*. In Fick and Sprague (Eds, 1980) Decision support systems: Issues and challenges. Pergamon Press (1980).

[Fow97] M. Fowler. *Analysis patterns: Reusable object models*. Addison-Wesley Longman, Inc. (1995).

[Gam95] E. Gamma, R. Helm, R. Johnson and J. Vlissides. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Longman, Inc. (1995).

[Gol92] G. Goldkuhl. *Contextual activity modelling of information systems*. In proceedings, 3rd Intl Working Conference on Dynamic Modelling of Information Systems, Noordwijkerhout (1992).

[Gol94a] G. Goldkuhl. *Välgrundad metodutveckling*. ("Well-founded method development", in Swedish). Research report, IDA, Linköping University (1994).

[Gol94b] G. Goldkuhl and D. Fristedt. *Metodanalys: En beskrivning av metametoden SIMM*. ("Method analysis: a description of the meta-method SIMM", in Swedish). Research report, IDA, Linköping University (1994).

[Gol82] G. Goldkuhl and K. Lyytinen. *A language action view of information systems*. In proceedings, 3rd Intl conference on information systems, Ann Arbor (1982).

[Gol98] G. Goldkuhl and P. J. Ågerfalk. *Action within Information Systems: Outline of a requirements engineering method.* In Dubois et al., proceedings of the 4th Intl workshop on requirements engineering: foundations for software quality (REFSQ'98). Pisa, Italy, June 8-9 (1998).

[Gol93] G. Goldkuhl and A. Röstlinger. *The legitimacy of information systems development: A need for change analysis*. In Proceedings, IFIP Conference on Human-Computer Interaction, London (1993).

[Hab84] Habermas J. *The theory of communicative action 1. Reason and the rationalization of society*. Beacon Press (1984).

[Hey93] M. Heym and H. Österle. *Computer-aided methodology engineering*. Information and software technology. Vol. 35, No. 6/7 (1993).

[Jay94] N. Jayaratna. *Understanding and evaluating methodologies: NIMSAD, a systemic framework*. McGraw-Hill International (1994).

[Kum92] K. Kumar and R. J. Welke. *Methodology engineering: A proposal for situation-specific methodology construction*. In Cotterman, Senn (Eds. 1992), Challenges and strategies for research in systems development. John Wiley & Sons Ltd. (1992).

[Lan66] B. Langefors. *Theoretical analysis of information systems*. Studenlitteratur, Lund, Sweden (1966).

[Mol96] B. Molander. *Kunskap i handling*. ("Knowledge in Action", in Swedish). Daidalos, Gothenburg (1996).

[Nil99] A. G. Nilsson. *The Method Developer's Toolbox: Chains and Alliances between Established Methods*. In Nilsson, Tollis and Nellborn (Eds. 1999). Perspectives on Business Modelling. Springer Verlag, Heidelberg, Germany (1999).

[Rum95] J. Rumbaugh. *What is a method?* Technical Paper, Rational Software, Inc. (1995).

[Sea69] J. R. Searl. *Speech acts. An essay in the philosophy of language*. Cambridge University Press, London (1969).

[Sto91] E. Stolterman. *Designarbetets dolda rationalitet: En studie av metodik och praktik inom systemutveckling*. ("The hidden rationale of design work: A study in the methodology and practice of systems development", in Swedish). Doctoral dissertation, Umeå University, Sweden (1991).

[Tol93] J-P. Tolvanen, K, Lyytinen. *Flexible method adaption in CASE environments – the meta-modelling approach*. In Scandinavian Journal of Information Systems, vol. 1, no. 5, pp 51-77 (1993).

[Win86] T. Winograd and F. Flores. *Understanding computers and cognition: A new foundation for design*. Ablex, Norwood (1986).