On the Concept of Method in Information Systems Development

Stefan Cronholm[a] & Pär J. Ågerfalk[b]

stecr@ida.liu.se[a] and pak@esa.oru.se[b]

Dept. of computer and information science[a]
Linköping University, SE-581 83 Linköping, Sweden

Dept. of Informatics (ESA)[b]
Örebro University, SE-701 82 Örebro, Sweden

Abstract

System development methods are often used as aids during the development of information systems. This paper examines the concept of method and other related notions in order to achieve a better understanding of their meaning and structure. The motivation for this paper is the emerging flora of different and related concepts and thus a need to sort things out. The main contribution of the paper is a conceptual model describing how the method concept and other related notions relate.

Keywords: method, methodology, information systems, systems development.

BRT Keywords: FA, FC

Introduction

The concept of method has been discussed for several decades (Bubenko & Källhammar, 1969; Checkland, 1981; Goldkuhl & Lyytinen, 1982; Jayaratna, 1994). Information systems development (ISD) methods are often used during development of information systems in order to guide and support the ISD process, i.e. ISD methods can be thought of as normative conceptualizations that direct method users' attention to certain kinds of phenomena. Hence, methods are created and used to support ISD actors as they perform different tasks, aiming to reach some goals (Ågerfalk & Åhlgren, 1999).

However, the meaning of the concept of method is not always clear. When studying different theories about methods or looking into different practical manuals it is obvious that several definitions of the concept of method exist, and there are also new emerging concepts. Furthermore, there exist several related concepts—often with "method" as a prefix and a following noun (method alliances, method components, method fragments).

When looking more closely at different concepts, one can identify that there are different concepts (and terms) used for the same phenomenon and also the same concept (and term) for different phenomena. The aim of this paper is to identify different method concepts and explain how these concepts are related to each other in order to achieve a

better understanding of the concept of method. The definitions of "method" chosen in this paper originate from several researchers who have all contributed to the understanding, and to the development, of the concept of method.

As far as we know, there has not been any work reported that tries to explicitly and adequately relate different methodological research in order to create a consensus. It rather seems that different method creators and researchers either 1) take the concept of method for granted, or 2) "run their own races" without paying necessary and sufficient attention to what others are doing, or manage to combine the two approaches. The motivations for this paper are thus to elaborate on 1) the complexity of the method concept, and 2) the fact that the method concept and method-related terminology are used in several different ways with different meanings in different contexts. The paper can hence be viewed as a suggestion to bring order to the flora of method-related concepts; an order that we think both method constructors and method users (i.e. systems developers) should be able to benefit from. The scientific development of cumulative knowledge requires some inter-subjective understanding of the studied phenomena and this paper is an initial attempt to create a "common base" for method research and method researchers.

Our work has been inspired by the Grounded Theory approach for qualitative analysis (Strauss & Corbin, 1998). By studying the literature, different central concepts have been identified. Related concepts have then been grouped into categories (which we prefer to still refer to as concepts) and relations between concepts (between categories), which finally yield the conceptual framework described.

The organization of the paper is as follows. The paper starts with some well-known definitions of the concept of method, followed by definitions of related concepts. Next, the concepts identified are related to each other in order to present a structure and to contribute to a deeper understanding of the concept of method. The paper ends with some conclusions.

Analysing definitions of the concept of method

In this section we present and elaborate on different definitions of "method" and other related concepts within the field of information systems.

Method versus methodology

Jayaratna (1994) defines "method" as *"an explicit way of structuring one's thinking and actions. Methodologies contain model(s) and reflect particular perspectives of 'reality', based on a set of philosophical paradigms. A methodology should tell you 'what' steps to take and 'how' to perform those steps but most importantly the reasons 'why' those steps should be taken, in a particular order."* As we can see, Jayaratna uses the term methodology. Methodology is a Greek term meaning the study of methods. The Oxford Dictionary defines methodology as *"the study of systematic methods of scientific research."* Jayaratna justifies the use of methodology claiming: *"however, the term methodology is pragmatically well established within the field of information systems to mean the same as method."*

Another example of the use of "methodology" synonymously with "method" is that of Stamper (1988), when stating that *"I use the term 'methodology' under protest bowing only to customary usage. It would be better, as in the philosophy of science, to*

speak of 'methods' when referring to specific ways of approaching and solving problems, and to reserve 'methodology' for comparative and critical study of methods in general; otherwise this vital field of study is nameless."

A third example of the misuse of the term methodology is by Brinkkemper (1996), who states that *"the misuse of the term methodology standing for method is a sign of the immaturity of our field, and should consequently be abandoned."* Brinkkemper (*ibid*.) further defines method as an *"approach to performing systems development projects, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products."*

A third definition of the concept of method is that by Röstlinger & Goldkuhl (1994). Their definition reads: *"methods are prescriptions for human actions and methods are normative and guide the ISD process"* (our translation to English).

Another definition from Checkland (1981) runs*: "a methodology will lack the precision of a technique but will be a firmer guide to action than a philosophy. Where a technique tells you 'how' and a philosophy tells you 'what', a methodology will contain elements of both 'what' and 'how'."* Checkland also uses the term methodology when he actually means method. Checkland also uses the concept of technique. Exactly what he means by technique is not defined, but we believe that he is referring to a diagramming technique, such as data flow diagramming or entity-relationship diagramming.

When we examine the method definitions above, it is clear so far that the term "methodology" is often used when what is actually referred to is "method". Method is descended from the Greek language, meaning "way of investigation". The meaning of "method" seems to answer the question of how ISD shall be performed.

Other related notions

Method types

Nilsson (1991) presents the concept of "method type" (in Swedish: metodik, there is no corresponding established English term). Nilsson distinguishes between a method type and a method. He defines a method type as a general concept (a type of method) and a method as a specific concept (an instance). In other words a method is a concretion of a method type. Following Nilsson's definition, Object-orientation is an example of a method type and the Object Modelling Technique (OMT), (Rumbaugh *et al.*, 1991) is an example of a method of that type.

Method chains and alliances

Fåhraeus (1986) talks about "method chains" as consisting of several methods linked to each other. Further, the result from a method used in an earlier step shall be used in a following step.

Nilsson (1998) has further developed the concept of method chain. Nilsson's definition runs: *"Integration of methods between different levels of development work. This approach to combining methods is a kind of vertical integration."* Nilsson points out that there are several (abstraction) levels of development work. For example there could be a higher level dealing with conceptual modelling, followed by object modelling performed in a lower level. The object model can in turn be used when defining the database schema. Nilsson's division into a vertical integration makes sense when he also introduces the concept of method alliances.

A method alliance is an integration of methods within the same level of abstraction. This is a horizontal integration of methods. Nilsson states alliances are motivated by the need *"to tackle several problems or perspectives in concrete situations."* That is, method alliances cover several aspects of a problem domain at a specific level. In our opinion a method should cover several phases and aspects within ISD. We thus think of "method" as addressing the ISD as a congruent whole. We interpret what Fåhraeus and Nilsson mean by "method" rather as method fragments or method components (defined below).

Perspective

Another method-related concept is "perspective." A perspective is a theory of how ISD shall be performed (Nurminen, 1988). This theory shall be normative, explanative and classifying. Mathiasen (1982) defines perspective as a conceptual abstraction of a view or a specific phenomenon. Jayaratna (1994) says, *"methodologies ... reflect particular perspectives of 'reality' based on a set of philosophical paradigms."*

In other words, the method constructor's perspective is based on how he or she perceives the world. The method constructor's values and beliefs thus influence the method user when performing ISD. A perspective implies, for example, what primitives to use and these primitives in turn influence method users (cf. Ågerfalk & Åhlgren, 1999). The character of the influence can be either governed or supported. The perspective is not necessarily made explicit in the method. The method constructor's perspective is often implicit and taken for granted. One can say that a method is always based on a perspective from which follow:

- principles
- values
- conceptions
- experiences
- categories
- definitions

We can distinguish between internal and externalized perspectives of a method creator (or any human being). The internal perspective is constituted by the parts of the conception of the world that are hard (or even impossible) to externalize. The externalized perspective, on the other hand, is constituted by inter-subjective beliefs etc., to which the method creator adheres. Examples of existing externalized perspectives in ISD are business-orientation, object-orientation and user-centred development. To sum up, the perspective, explicit or implicit, influences the method user in one way or another.

Framework/model

Another related and sometimes confusing term is "model." What do we actually mean when talking about models? According to Yourdon (1989), a model is used to *"highlight, or emphasize, certain critical features of a system, while simultaneously de-emphasizing other aspects of the system."* Examples of classical tools to express models are data flow diagrams and entity-relationship diagrams (*ibid*.). Rumbaugh *et al.* (1991) define a model as *"an abstraction of something for the purpose of understanding it before building it."*

Jayaratna (1994) defines "framework" as a static model, which provides a

structure to help connect a set of models or concepts. Goldkuhl (1991) defines "model" as a structure for the ISD process. Further, a model defines and delimits specific areas within ISD that form related phases. A model answers the question of *what* is to be done but not *how* it should be done. Examples of such models are the classical Swedish SIS/RAS model and the LOGIC model.

What makes the definitions above confusing is that they are referring to different domains. When we examine Yourdon's definition, it is obvious that he is referring to a model of an information system. The same goes for Rumbaugh *et al*. However, when Goldkuhl talks about models he is referring to a model of the ISD process. In other words, they are using the same term but referring to different concepts. Jayaratna (1994), similarly to Goldkuhl, refers to the ISD process whilst the others refer to the product of such a process.

To avoid confusion we think that it is better to, as Jayaratna does, use the concept of "framework" when referring to the ISD process. In Röstlinger & Goldkuhl (1994), the framework concept is also used as a synonym to model. The concept "framework" is well defined in the software engineering community but not fully applicable in the information system community.

One definition from the software engineering community runs (Öberg, 1998): *"A framework is a generic design solution to a certain problem or a certain domain. The framework describes the different design elements involved in the solution, as well as their relations."*

If one changes the term "design solution" to "ways of performing ISD" and the term "design elements" to "phases" the definition becomes similar to Röstlinger & Goldkuhl's (1994) definition of framework/model.

Method components and method fragments

A method can be perceived as a "whole" consisting of different "parts". Therefore we also need a concept for the parts of a method. During the last few years, concepts such as "method components" (Röstlinger & Goldkuhl, 1994) and "method fragments" (Harmsen, 1997) have been proposed to talk about method parts. A reason for this is a move from viewing methods as monoliths to a generic flexibility (Röstlinger & Goldkuhl, *ibid*.) suited for situational method engineering (Harmsen, 1998; Brinkkemper *et al.*, 1998).

The concept "method fragment" is defined by Harmsen (1997) as *"… a description of an IS engineering method, or any coherent part thereof."* From this definition, a complete method, for example OMT, is a method fragment and so is any single concept used within OMT, for example "object." To sort this out, a method fragment is said to reside on a certain *layer of granularity*, of which five are possible: method, stage, model, diagram or concept. Thus, "object" resides on the concept layer and "OMT" on the method layer. Furthermore, a method fragment is either a *process fragment* or a *product fragment*. Process fragments represent the activities, stages, etc., that are to be carried out and product fragments represent deliverables, diagrams, etc., that are to be produced, or that are required during development.

Röstlinger & Goldkuhl (1994) view methods as constituted by exchangeable and reusable components. Each component consists of descriptions for ways of working (a process), notations and concepts. A process describes rules and recommendations for the ISD and informs the method (component) user what actions to perform and in what order. Notation means semantic, syntactic and symbolic rules for documentation. Concepts are categories included in the process and the notation. A method component

can be part of a method chain or a method alliance. A method component or fragment can also be used separately and independently from other components. Each method component addresses a certain aspect of the problem at hand. Examples of method components are “use case analysis” (Jacobson *et al.*, 1992) and “object modeling” (Rumbaugh *et al*, 1991), which are both parts in a whole (a method).

Thus, a method component can be thought of as the smallest *meaningful* assembly of method fragments to address a certain aspect of a problem (cf. Brinkkemper *et al.*, 1998) and consists of product fragments (notation), process fragments (process) and concept fragments (concepts) used in the other two types of fragments. Note that a method component *per se* is a method fragment at some intermediate layer of granularity.

Co-operation forms

The Scandinavian tradition of performing ISD often means that several actors are involved in the ISD process. Hägerfors (1994) describes the ISD process as a group process with actors who interact, discuss, learn, agree, disagree and argue. Several research reports argue for strong user (business actor) participation. This means that methods also should support cooperation forms. According to Goldkuhl *et al.* (1997), cooperation forms describe *“how different persons interact and cooperate when performing method-guided work.”* Cooperation also has to do with roles and division of work. One can say that co-operation forms deal with the meta-question of who is to ask the questions during ISD (*ibid.*). Examples of cooperation forms are brainstorming sessions, interviews and modelling sessions. Harmsen (1997) distinguishes between two different domains that are in focus during ISD. Some ISD activities belong to the “target domain” and some to the “project domain.” The target domain consists of activities directly addressing ISD, and the project domain consists of activities addressing management thereof. Cooperation forms thus belong to the project domain.

Relating the concepts

In this section we relate the concepts identified above to each other (see figure 1). The aim of relating related concepts is to explain the concepts in order to achieve a deeper understanding of the concept of method. The meaning of the arrows in figure 1 is to indicate the intended direction of reading.

The analysis shows that there is a need for a distinction between “method” and “methodology.” As mentioned above, the relation between them is that methodology *studies* methods. A method can be thought of as a method fragment at the “method” layer of granularity. Two distinguishable focal areas of methods have been identified as the *project domain* and the *target domain*. The recommended cooperation forms of a method belong to the project domain. Note that the project domain itself contains methods for project management, process improvement, techniques for gathering data, etc.; this yields a recursive relation not explicitly shown in figure 1. Such project domain activities are, however, out of the scope of this paper. It is important, however, that the project domain focal area shall *harmonize* with the target domain focal area.

A method implies a perspective. As discussed previously, a perspective is either explicit or implicit in the method (or methodology for that matter). Furthermore, a framework *dictates what is to be done* during ISD and thus relates to both the target domain and the project domain. As we see it, a many-to-many relation exists between frameworks and the methods target domain as well as between the framework and the

methods project domain. Thus, one method chain or method alliance can be used in different frameworks. There is also a many-to-many relation between method chains and method components as well as between method alliances and method components. Method chains are a *vertical composition* of one or more method components and one method component can be used in different method chains. Similarly, a method alliance is a *horizontal composition* of one or more method components, in which each method component might be used in several method alliances. Finally, a method component is a meaningful assembly of process fragments and product fragments and the concepts used within those fragments. By meaningful, we refer to appropriateness for addressing a certain aspect of a problem at a potentially re-usable layer of granularity.

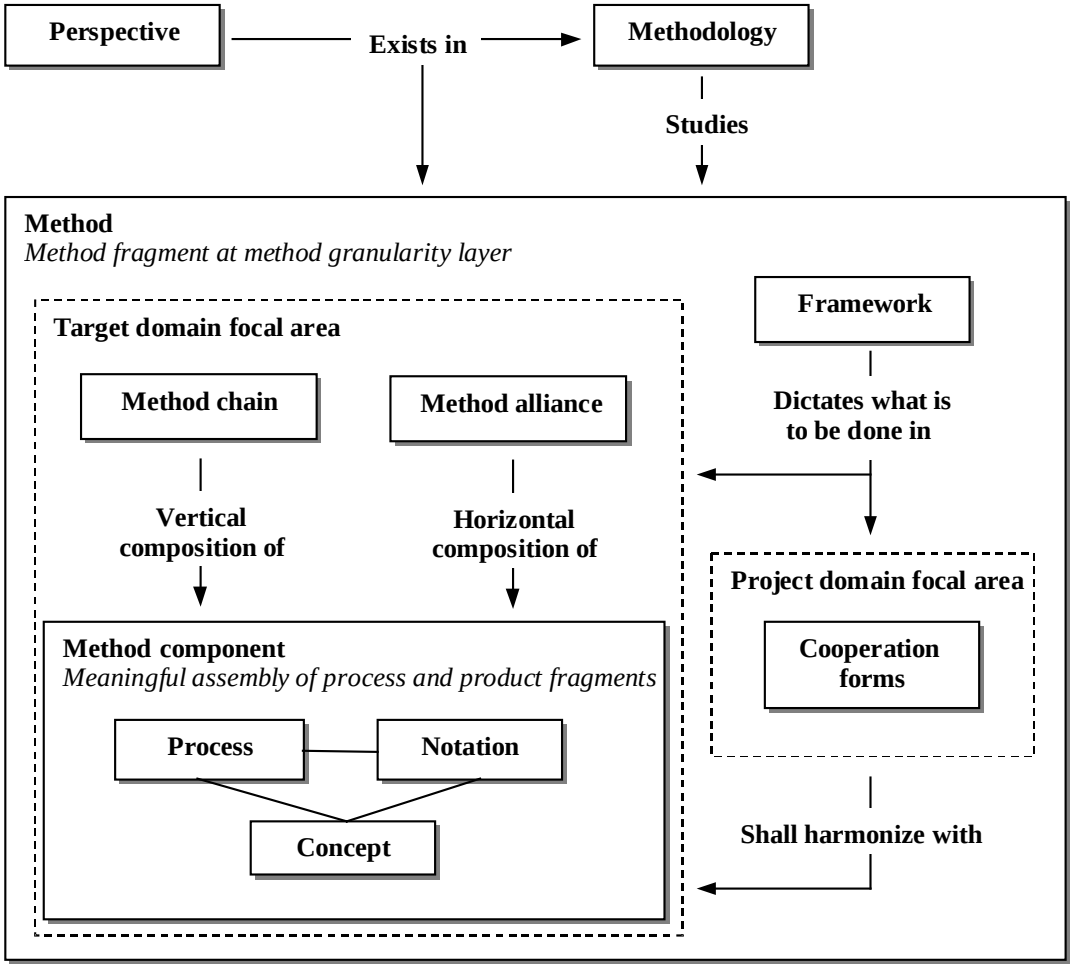

Figure 1: A conceptual model of the method concept and its relations.

Conclusions

The method concept has been widely discussed in the IS community. The main contribution of this paper is the conceptual model illustrated in Figure 1. Examples of other approaches to understanding the concept of method can be found in, for example, (Jayaratna, 1994) and (Röstlinger & Goldkuhl, 1994). What distinguishes our approach from Jayaratna's is that our approach presents a coherent conceptual model of the concept of method, including relations to related concepts. Jayaratna is more focused on

understanding and evaluating methodologies (methods). Röstlinger & Goldkuhl (1994) also present a conceptual model. Their model has inspired us to further enhancements. What is missing in their approach so far, is that they do not relate their findings to other contributions to the development of the concept of method.

One obvious conclusion is that there exist many different concepts of method. At first sight, some of these concepts seem to be similar and may be redundant. When analysing the definitions of the concepts, it is clear that the identified concepts refer to different phenomena. Our analysis has also shown that there are different terms referencing the same concept (method *vs.* methodology). In order to achieve better understanding of the various concepts, we think that a good approach is to relate them to each other—not only to understand the relations, but also to understand the concepts *per se*.

We believe that the conceptual model presented does not cover all aspects of the concept of method. In fact, it is doubtful that any model could ever claim to do that. There may be other important aspects worth mentioning, but we think that we have captured the most central ones. Nevertheless, we believe that there is a need to further analyse the concept of method in order to understand the nature, or essence, of the concept and hence of method-supported work. In particular, we have identified two future research activities in continuation of the work reported in this paper. The first activity would be to try to apply the proposed conceptual model to different ISD approaches to reveal possible inconsistencies and to verify its possible explanatory power. This would be an empirical hypothesis-testing investigation. The second activity would be to continue the theoretical work and, if possible, try to simplify the conceptual model, which has become somewhat complicated.

Another possible future research topic would be the elaboration of the differences between ideal-typical (generic) method descriptions, which we have focused on so far, and situational method usage. Using our conceptual framework in a situational setting might be a practical way to gain better understanding of method using practice and situational method engineering, and their relations to methods at the ideal-typical level (cf. Ågerfalk & Åhlgren (1999) for a discussion about generic versus situational existence of methods). In particular, the combination of the notion “method fragment” with that of “method component”, as suggested in this paper, seems to be an interesting source for further research on this topic.

References

Brinkkemper S. (1996). Method Engineering: Engineering of Information Systems Development Methods and Tools. In *Proceedings of the Fifth International Conference on Information Systems Development* (ISD’96), Wrycza-Zupancic (ed.). Gdansk, Poland.

Brinkkemper S, Motoshi S & Harmsen F. (1998). Assembly Techniques for Method Engineering. In *Proceedings of the Tenth International Conference on Advanced Information Systems Engineering* (CAiSE*98), Pernici B & Thanos C (ed.). Pisa, Italy.

Bubenko J & Källhammar O (1969). Computer Aided Design of Information Systems. CADIS Working Paper - 1, sept 1969. Objectives for Research in Information System Design. In Bubenko J jr, Källhammar O, Langefors B, Lundeberg M, Sölveberg A (Eds.): *Systemering 70*, pp. 395-403. Lund: Studentlitteratur, 1970.

Checkland P. (1981). *Systems Thinking, Systems Practice*. Chichester, UK, Wiley.

Coad P & Yourdon E. (1991). *Object-Oriented Analysis.* Englewood Cliffs, New Jersey: Prentice Hall.

Fåhraeus E. (1986). In Swedish: *Metodkedjornas och verktygens roll vid systemutveckling.*

Rapport nr 24 Riksdataförbundet, Stockholm.
Goldkuhl G & Lyytinen K. (1982). *A language action view of information systems*. SYSLAB report no 14, SYSLAB, University of Stockholm, Stockholm, Sweden
Goldkuhl G. (1991). In Swedish: *Stöd och struktur i systemutvecklingsprocessen.* Institutionen för datavetenskap, Linköpings universitet.
Goldkuhl G, Lind M & Seigerroth U. (1997). Method Integration as a Learning Process. In *Proceedings of the 5th BCS Conference on Training and Education of Methodology Practitioners and Researchers.* Jayaratna N, Wood-Harper T & Fitzgerald B (eds.). Preston, England.
Harmsen A F. (1997). *Situational Method Engineering.* PhD dissertation. Moret Ernst & Young Management Consultants, Utrecht, The Nederlands.
Hägerfors A (1994). *Co-learning in participative systems design, Enhancements of genuine participation by consideration of communication and group dynamics.* Lunds University.
Jacobson I, Christersson M, Jonsson P & Overgaard G. (1992). *Object-Oriented Software Engineering*. Menlo Park, CA: Addison-Wesley.
Jayaratna N. (1994). *Understanding and Evaluating Methodologies.* McGraw-Hill Book Company, London.
Mathiassen L. (1982). In Danish: *Systemudvikling og systemudviklingsmetode, doktorsavhandling*. Datalogisk afdelning, matematisk institut, Aarhus universitet.
Nilsson A. (1991). In Swedish: *Anskaffning av standardsystem för att utveckla verksamheter, doktorsavhandling*. Ekonomiska forskningsinstitutet (EFI), Handelshögskolan i Stockholm
Nilsson A G. (1998). The Business Developer's Toolbox: Chains and Alliances between Established Methods. In *Perspectives on Business Modelling - Understanding and Changing Organisations*, Nilsson A G, Tolis C & Nellbron C (eds.). Springer Verlag, Heidelberg.
Nurminen M. (1988). *People or Computers: Three Ways of looking at Information Systems.* Lund, Studentlitteratur.
Rumbaugh J, Blaha M, Premerlani W, Eddy F & Lorensen W. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs, New Jersey: Prentice Hall.
Röstlinger A & Goldkuhl G. (1994). In Swedish: *Generisk flexibilitet – På väg mot en komponentbaserad metodsyn*. Presenterat på VITS Höstseminarium 1994. Institutionen för datavetenskap, Linköpings universitet.
Stamper R. (1988). Analysing the Cultural Impact of a System. *International Journal of Information Management*, vol. 8, no. 3.
Strauss A, Corbin J (1998). *Basics of Qualitative Research. Techniques and Procedures for Developing Grounded Theory*. 2nd Edition. Sage Publications.
Yourdon E. (1989). *Modern structured analysis.* Englewood Cliffs, New Jersey: Prentice Hall.
Ågerfalk P J, Åhlgren K (1999). Modelling the Rationale of Methods. In *Managing Information Technology Resources in the Next Millennium*, proceedings of the 1999 Information Resources Management Association International Conference. Hershey, PA, USA, May 16-19.
Öberg F. (1998). *Object-Oriented Frameworks – A New Strategy for CASE Tool Development. Licentiate Thesis.* Department of Computer and Information Science. Linköping University.