Drilling Unresolved Issues for Method Development and Upcoming Functional Method Engineering

S.B. Goyal, Aarti Goyal, Pratima Sharma

MRCE, Faridabad, INDIA

Abstract We tried to workout the unresolved issues for method development methodologies like text-book approach, contingency approach, method engineering approach & situational method engineering approach. We found that the problem of selecting a good candidate method is a complex issue till situational method engineering. On the basis of literature survey, we believe that the solution to this can be found not in the structure of a method itself but in the work to be done, the task to be performed by the method. We are proposing upcoming solution (i.e. Functional Method Engineering) of unresolved issues of situational method engineering.

Keywords: Component, Fragment, Meta-model, Method, Method Engineering

1 Introduction of Method

Many definitions of a method have been proposed [1], [Pra94], [2], [3]. According to [1], a method is an approach based on a certain way of thinking, to carry out an Information Systems (IS) development process consisting of directions and rules structured according to a systematic ordering of development activities and corresponding development methods. [4] has defined a method as a set of specific formalisms and is the working procedure structure to build well-formed instances of specifications. According to [Pra97a], [Pra99a] a method is the decision making capability and the mechanism that supports this. Also, a method has its underlying paradigm [7]. A method is a procedure for attaining something in the field of Information System Development (ISD).

A *method* can be considered as a predefined and organized collection of techniques and a set of rules which state by whom, in what order, and in what way the techniques are used [8] to achieve or maintain some objectives.

An Information System Development Method, ISDM consists of a set of models that describe the Universe of Discourse, a set of guidelines or principles that guide the development process and a set of tools to support both the product and the process itself. Researcher has categorized methods as **data-oriented, process-oriented, behaviour oriented**.

A method produces a product. Depending on the nature of the method, it may produce a requirement engineering product, a design product, a construction design product etc. Methods in **business processes** produce business objects, e.g., a passport issuing method produces a passport. In order to produce a product, a method is defined in terms of a number of activities that comprise it. Detail of method evolution can found in table 1.

Data oriented methods emphasize the complete and thorough analysis of data and its relationships. Examples of data driven methods are ER and NIAM. **Process-oriented** methods place emphasis on activities of an application domain, their interrelationships and decompositions. Examples of process-oriented methods are SASD, SSADM, SADT, JSD and ISAC. **Behavior-oriented** methods focus on the dynamic nature of the data by analyzing and understanding the events in the real world which impact data recorded in the IS. Examples of methods based on the behavioral approach are REMORA, TAXIS and OBCM. Object-oriented analysis and design methods, and and user interface design methods also exist. These tend to integrate in them concepts found in the three categories of **data-oriented, process-oriented, behaviour oriented**.

ISDMs offer a rich set of quality criteria like **guidelines, heuristics** and **method constraints** that influence the properties of the product and the process of development. **Guidelines** identify what can/should be done while developing a product. In this sense, guidelines improve the quality of the development process. Heuristics identify the bounds/criteria for products. **Heuristics** can be defined for a variety of purposes like better product structure and product clarity. An example of a method heuristic in OMT is that the number of specialization levels should be less than 3. **Method constraints** deal with the restrictions, which make the product well defined and well formed. For example in ER model, a relationship is complete provided it has at least two entity classes associated with it.

Table 1: Method Evolution

Development Duration	**Method Name**
1960's	Petri-nets
Beginning of 1970's	Structured programming (Boehm, Dijkstra)
Mid 1970's	Structured design (Yourdon-Constaine), JSP (Jackson), ER-model (Chen)
End of 1970's	Structured analysis (DeMarco, Gane-Sarson)
Beginning of 1980's	Metrics
Mid of 1980's	Information Engineering (Martin)
End of 1980's	Object-oriented design (Booch)
Beginning of 1990's	Object-Oriented Analysis (Coad/ Yourdon, Schlaer-Mellor, Rumbaugh)
Mid 1990's	Business modeling, BPR modeling, workflow modeling
End of 1990's	OO-method standardization (UML), reference modeling, modeling of packaged software.
Beginning of 2000's	Competent modeling, application-specific modeling languages, UML profiles, various meta-models
Mid 2000's	RUP

2 Method Development Approaches

We can find in the literature some case studies of method development including extensions of current methods [Aal93], [Nis96]; the selection of methods, their development and introduction seems to be done in an ad-hoc manner by choosing tools and methods on a trial-and-error procedure [9]. These principles include how to construct and adapt methods for particular needs situation specific, how to check the applicability of the method, and how to organize method development efforts.

We have analyzed the literature on methods ([9], [10], [11], [12], [13], [14]), their selection and development. We found three basic strategies for local method development are 1) **a text-book approach**, 2) **a contingency approach** and 3) **a method engineering approach**, either at the organizational level or at the project level. These are differing in the extent of the changes that are made to methods to meet the situation specific needs.

2.1 A text-book approach: This approach offers a simple strategy for method development: the method construction and tool adaptation steps do not take place. It is applicable for organization level and believes that development situations are generally alike, and thus can be solved with standard solutions, for example, SSADM [15], IDEF [16], UML [17], they also aspire to other objectives such as communication between different ISD tools. Hence, In this approach a whole method is chosen.

2.2 A Contingency approach: This approach is based on contingency theory for method selection and assumption that there is no universally acceptable method which is applicable in all circumstances and based on the observation that situations of practice can be classified, but are more situation bound than the first approach expects [18]. This approach tried to identify prominent characteristics (i.e. situation dependencies) ([9], [19], [10], [20]) which control outcomes of the use of methods and predict their suitability. This approach mostly used to analyze situational features of methods and applied for method selection and development also. It is largely made by choosing individual techniques from a large set. It focuses on the selection of an available, appropriate method rather than on more detailed method construction [11]. It ignores the impact of organizational learning during method selection.

2.3 A method engineering approach: Method Engineering, ME approach proposed by Kumar [11] that methods should be constructed to meet a particular IS development's needs. It aims to construct or "engineer" an ISD method according to stakeholders' requirements. i.e. ME aims to develop and maintain systems for ISD. ME approaches should be adapted to construct ISD method to local situations also even if it requires detailed modification of methods. Here, the fundamental assumptions are uniqueness and difference in ISD situations which cannot be solved by general and universally valid methods or general contingency-based selection principles. It emphasized on method construction and tool adaptation. ME selection is made by choosing components of techniques (or methods) and by constructing unique components.

3 Method Engineering Approaches

ME approaches can be further distinguished by whether they aspire towards an **organization-specific** or **a project-specific** method.

Organization-based ME, is based on an assumption that development situations and thus also supported methods are alike in an organization and the method can be developed to meet these requirements. In the organization this method is then believed to be appropriate for all projects. Baskerville [Base96] calls these methods contingency methods, as they are situation specific for certain types of bounded organizational settings. Another ME approach is ***project-based ME***, which assumes that methods should be "engineered" on a project basis. Because this approach copes with the uniqueness of each ISD setting [21], it focuses on advancing method knowledge in the context of a single ISD project. Thus, it is believed that development situations differ between various projects. Although there can be an in-house method in the organization, according to project-based ME there is also a possibility to adapt it, or even to develop various project variants.

Organizations have at least one strategies follow for method development and method selection shown in Fig.1.

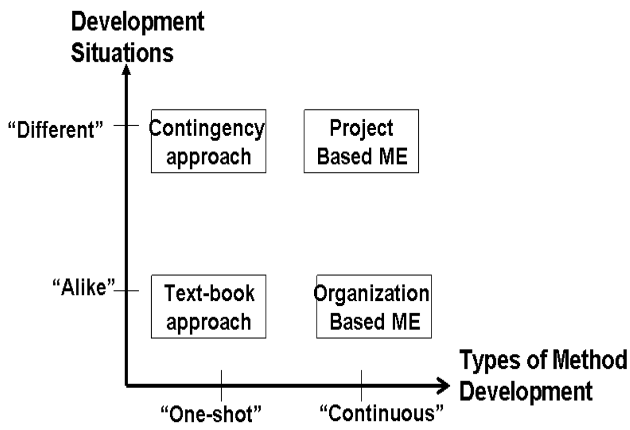

Fig.1: Strategies of organizations method selection and development.

To summarize, situation-independent, organization-independent, project-independent and universal methods are not possible because every ISD situations are so different and complex also. Applicability of a method in one situation does not mean that it provides successful results in other situations. Hence, in ME method development is viewed as a knowledge creation process which cannot be done in a "one -shot" manner. Many parts of the methods are modified, some parts are excluded, and new ones are included.

4 Method Engineering Domains

ME is a relatively new research field, there is a lack of experience in applying meta-modeling and modular method construction principles. Kumar [11] suggested four domains that have to be addressed in ME:

4.1 Modular method construction based*:* This domain [11], suggested that ME can be carried out by using pre-defined and tested method modules i.e. called a *component base* [11], or *method fragments*. They **either i)** based on method's static its conceptual structure [1], or meta-data models [1] - these models contain knowledge about the stages and tasks of a method. Many researchers have developed meta-modeling languages [23] for these models. **or ii)** the dynamic features of a method, i.e. its procedural part or by meta-activity models [1] or by process models, meta-activity models.

We found that major differences among all these approaches can be found in their modeling power and capability, degree of formality, and ways to represent method knowledge. In this category, method modeling based on method comparisons and analysis and focused on individual aspects (i.e. how a single developer understands and uses a method) rather than on the use of methods in the large and by many.

4.2 Stakeholder requirement based method composition: One important objective of ME is that constructed methods meet users' requirements. So, ME requires methods and guidelines to identify stakeholders: such as programmers, designers, IS users and managers and their requirements [11]. This is one important and essential factor in accepting constructed methods and more expected that method users will more easily learn the methods, accept them, and use them if the methods are based on their requirements, in contrast to the situation where introduced methods are purely based on requirements outside the organization. e.g. UML [17]. In this domain few empirical studies have been carried out.

Goldkuhl identified different roles and needs for the tool adaptation and focused was on technical issues dealing with customizable tools rather than on local method development. Tolvanen focused on a limited number of stakeholders and a few contingency factors.

4.3 Computer aided supporting tool based: In this domain focused on developing tools for capturing method knowledge as well as building meta-modeling-based tools that can be customized [22]. As ISD methods are supported by CASE tools, similarly meta-modeling languages are increasingly supported by meta-CASE tools and captured this symmetry and introduced a more general terminology, CAME, Computer Aided Methodology Engineering [11] to highlight the role of computer based tools in ME. We had found that either principles and requirements for such environments or represent how one particular system has been implemented and how it works. We found also that reveal that CAME tool developers have concentrated so far on techniques that allow tool adaptation rather than on developing techniques and principles for utilizing tool based knowledge about methods for example in method selection, method composition & construction, and reuse. Yet, without proven ME principles, the development of advanced tool support for ME will be very less concentrated.

4.4 Organization support based: In this domain, involves a supporting organizational structure and mechanisms that ensure method selection, development, training, use, and maintenance. Researchers shown that organizations develop their own versions of methods, these tasks are already being managed somehow. Researchers studied the roles of method engineer in an organization and tasks needed for method engineering in this domain. Studies of the other people involved or tasks and organizational structures and mechanisms needed to carry out ME in practice are missing.

General impact of Method Engineering from industry perspective: Method engineering is a Technique for: 1) Increasing production per unit of time. e.g. increasing the number of customers that can be handled per cashier by installing bar code readers. 2) Decreasing cost per unit output. e.g. decreasing total cost of each cell phone by reducing the number of parts and thus the labor hours required for assembly. It is critical to look at impact on whole system.

5 Companionship between Method & tool

Tool support is essential instrument for evidence of ME research because tools can ensure that method knowledge is applicable or not and measurement of described method vs. method in use. So, we need ISD tools to implement method knowledge and contain parts of the conceptual structure as their schema definition, support modeling with certain notations, or support the process definition and management also. Some popular examples of methods and tools in ISD phase or tasks of ISD described in table 2. ISDMs without considering their support in ISD tools would be the same as designing an IS without implementing it.

Table 2: Different methods and Tools in ISD phase

S.No.	Category of Tools	Category of methods	ISD Phase / Tasks of ISD
1.	Upper-CASE, interface design tools	Data modeling, structured analysis, object-oriented analysis	System analysis
2.	GDSS, CSCW, requirements engineering	Brain-storming, interviews, requirements	Requirem ent Engineeri

	tools	definition and design techniques	ng
3.	Upper-CASE, interface design tools	Data modeling, structured design, object-oriented design	System Design
4.	4GLs, editors and compilers, debuggers, code generators, verifiers, performance analyzers	Mapping from high-level language to machine language, version control	Construction
5.	Documentation and reporting tools, reverse engineering tools	Version control, configuration management, reverse engineering	Operation and Maintenance
6.	Work flow modeling tools, simulators, business modeling tools	Business modeling, process modeling,Work flow modeling, task structures	Business process reengineering and development

A number of methods have been published ranging from generic methods to methods for specific projects or domains like Unified Process, RUP, DSDM, SSADM, Merise, OPEN Process Framework for general system development; ARIS and DEM for ERP implementations; Scrum, XP (Xtreme Programming), crystal in the agile movement; T-map for testing; Attribute Driven Design method and TOGAF for software architecture; Archimate and Dynamic Architecture (DYA) for enterprise architecture.

6 Meta Model

Models play a crucial role in ME, as in all engineering. A *metamodel* is a conceptual model of an ISD method. Meta-models have emerged as a principal means to understand the activities seen in the area of Information Systems. Meta-model is an abstraction of methods i.e.one level of abstraction higher than “normal” models. *Metamodeling* is a modeling process which takes place one level of abstraction and logic higher. A meta-model is a system of meta-concepts, inter-relationships between these, and constraints. It is possible to use meta-models as a basis for method engineering and CASE shell construction and in the development of CAME tools. Explicit output of meta-model is method. It can systematize and formalize method knowledge of exiting method in terms of CASE tool and repository implementation; mapping to generators and other tools; interfacing tools (e.g. CDIF & XMI); compare methods; support standardization efforts (e.g. OMG’s UML). Broadly, there are two types of meta-models: **Meta-data models** (static structure of method), and **process models** (dynamic part of method). No modelling is possible without explicit or implicit involvement of meta-model.

A number of meta-models have been developed in the last decade for a range of uses in understanding conceptual models of the 1970s and 1980s, identifying commonalties between models, representing product and process aspects of methods, understanding process models, representing quality aspects of methods, developing techniques for requirements engineering and investigating data warehouses.

Reflecting the product-process dichotomy of methods, two types of meta-models have been developed. The **first** of these are meta-data models for the **product aspects** of methods. Such models introduce a system of generic concepts in which the static, data aspects of methods as well as constraints defined in them can be represented Examples are meta-models of [1], and object Z. The meta-model of Smolder uses the concepts of Object-Property-Role-Relationship for this meta-level. The **second** kind of meta-models deals with the **process aspects** of methods. These meta-models are called meta-activity models and specify a system of concepts to define tasks and task transition criteria [1]. Motivated by the need for coupling the product and process aspects of methods, a **third** kind of meta-models, **integrated product/process** meta-models have been defined. Examples of these are the contextual meta-model and the decisional meta-model have been proposed**.**

By and large, meta-concepts have dealt with abstractions of product and process concepts only. ***The basis of the method engineering is meta-model instantiation***.

7 Situational Method Engineering

Method engineering aims at designing and building methods. For some time now researchers have been talking about method reuse. The earlier approaches to ME ignored the issue of project needs while developing methods. There was no way through which methods could be modified or adapted according to the needs of the project.

The notion of Situational Method Engineering (SME) was first proposed by Kumar and Welke in 1992 as a new method engineering discipline the aim of which is to construct new methods and the associated tools or to adapt existing ones to every Information System Development (ISD) project. In other words the emergence of the ME acknowledges the need for the construction of methods tuned to specific situations of developments projects.

A number of SME approaches have been already proposed from the last few years. Most of them use *assembly* techniques based on the reuse of existing method parts in the construction of new methods or in the enhancement of existing ones.

Recently, situational method engineering approach has been proposed by Harmsen et. al. [23] to adapt standard methods to specific situations. Three major proposals exist for construction of situational methods; one is based on fragments by Harmsen et al. [23], [24], the other on the notion of contexts by Rolland et al. [25], [26]. The third [22] is an extension of the early approach used in MetaEdit and is based on ME framework of Harmsen.

SME is well known that methods can be engineered from scratch, by modification of other methods, by assembly or Method Intention Architecture,(MIA) . Method construction from scratch requires long lead times for method development. Method modification and assembly rely on method reuse. Both of these assume the existence of a method repository from where the method(s) of interest are retrieved and modified or assembled into a new method that is subsequently stored in the repository for reuse.

Method engineering built methods from scratch and it was difficult to reuse previous method knowledge. This inhibited method adaptation and reuse to address needs of specific projects. Situational Method Engineering (SME) techniques were developed to handle development of situation specific methods. A central repository called the method base contained reusable method parts. The method base could be queried and appropriate method parts retrieved. These could then be assembled together to form the desired method. Proposals for querying the method base use descriptors, project contingency factors and multi-criteria search descriptors. These assume that it is possible to postulate a set of method attributes that are generic enough to be used for retrieving method parts of any new method to be constructed.

Several SME approaches have been described in the literature. The approach of Brinkkemper [27] relies on the experience and knowledge of the method engineer in ensuring well selected components and building the required method. Grundy [28] proposed an integrated facility for carrying out method development from scratch, method modification and method reuse. Gupta [29] proposed an instantiation algorithm that formed the basis for method development from scratch, by modification and by assembly.

Ralyte [30] suggests that method engineering is facilitated if the intention of the method can be determined and raises some questions: a) how can assurance be provided that the method to be enhanced, extended, or restricted is a good candidate method? b) What are the chances that at the method engineering intention stage, the method shall have to be discarded because its adaptation is very difficult? c) Should not some more exploratory work be done before committing to setting up method adaptation intentions?

To find the answers of above unanswered questions, we have developed a MIA approach [32]. MIA approach has three main notions, Intention, Architecture and Organisation and focused on elaborating the notion of method architecture [32] and its relationship with method organization [31] and shown that method architecture [31] is a functional abstraction of a class of organisations. Each organization [31] is a specification of the features that are to be built in the method. There can be many organizations [32] for a given architecture.

The basic process for engineering architectures and organisations is assembly based. Just as method chunks, fragments etc. can be assembled from re-useable parts, so also architectures and organisations can be assembled together from re-useable components.

8. Upcoming Functional Method Engineering

Let us bring out the difference between SME and FME being proposed here. In the fragment based SME proposal [27], we have two fundamental elements a) *product and their structures* b) *procedures and their execution order to develop the products.* It is clear from (a) that interest is the structure of products. Similarly, since the structure of a process is largely determined by the order of execution, interest is in process structure. Therefore, we can conclude that SME is centered around the structural aspects of methods.

This focus on engineering the structure of methods de-emphasizes what the method does, what task it is good for. In fact, the determination of whether the method structure can carry out the project task at hand is based on the experience of the method engineer. In other words, the method engineer determines the task of the project by some ad-hoc means selects the appropriate method structures and then assembles these together.

FME puts method structure subordinate to method functionality. FME asks for an explicit determination and representation of method functionality in the form of method architectures. It is only after the architecture has been built that the issue of method structure is to be considered. In this sense, SME occupies the, downstream, construction engineering stage of our life cycle [31] .

9. Conclusion & Future Work

Today, method development has become one of the most valuable assets of corporate world; development of ISDMs and non-ISDMs faces many problems out of which the most important are low productivity, a large number of failures, and an inadequate alignment of ISs with business needs. These problems are sometimes due to economical imbalance, such as cost overrun and delay in scheduled delivery, but surprisingly often due to poor product quality and insufficient user satisfaction. The major reason for the specified problems is selection of a good candidate method from method repository.

We are expecting as a result from FME, method selection for adaptation shall be more appropriate and give assurance that the SME task is progressing purposefully. The chance of method rejection at later stages shall be considerably reduced.

We believe that answer to the selection of a good candidate method from repository can be found outside the boundaries of methods, their concepts and structures. The moment we step back from methods, the question naturally arises as to whether we can address 'all kinds' of methods or only ISDMs?

Our future work for FME will move away from features and look at methods in a global sense. Our global view is derived from organizing method engineering in three levels, . We will concentrate on functional method. We will show that progression occurs within this level and a stage is reached where method features get identified

situational, functional and intentional. The last lays down the requirements, the second the functional architecture to meet these needs and the first provides the set of concepts and interrelationships to realize the functional. Progression occurs down the three levels. We will concentrate on functional method. We will show that progression occurs within this level and a stage is reached where method features get identified.

References

[1] Brinkkemper S., *Formalisation of Information Systems Modelling, Ph.D. Thesis*, University of Nijmegen, Thesis Publishers, Amsterdam, 1990.

[2] Wynehoop J.D., Russo N.L., *System Development Methodologies: Unanswered Questions and the Research-Practice Gap*, Proc. Of 14th ICIS, Orlando, USA, 1993

[3] Lyytinen K., Smolander K., Tahvainen V-P, *Modelling CASE Environments in Systems Work*, Conference on Advanced Information Systems Engineering, Sweden, 1989.

[4] Loucopoulos P., Zicari R., *Conceptual Modelling, Database and CASE*, Wiley (Pub.), 1992.

[5][Pra97] Prakash N., *Towards a Formal Definition of a Method*, Requirements Engineering Journal, Vol. 2(1), Springer Verlag, U.K., 199

[6] [Pra99] Prakash N., *On Method Statics and Dynamics*, Requirements Engineering Journal, Vol. 24 (8), Springer Verlag, U.K., 1999.

[7] Rodden T. et.al., *Process Modelling and Development Practice*, in Software Process Technology, Third EWSPT, Warboys B.C. (ed), LNCS, 772, Springer Verlag, 1994.

[8] Smolander, K., Tahvanainen, V.-P., Lyytinen, K., How to Combine Tools and Methods in Practise - a Field Study. In: *Lecture Notes in Computer Science, Second Nordic Conference CAiSE'90*, (eds. B. Steinholtz, A. S. lvberg, L. Bergman) Stockholm, Sweden, May, pp. 195-211, 1990.

[9] Davis, G.B., Strategies for information requirements determination. *IBM Systems Journal*, Vol. 21, No. 1, pp. 4-30, 1982

[10] Sullivan, C.H., Systems Planning in the Information Age. *Sloan Business Review*, Vol. 26, 2, pp. 3–11, 1985

[11] Kumar, K., Welke, R.J., Methodology engineering: a proposal for situation-specific methodology construction. In: *Challenges and Strategies for Research in Systems Development* (eds. W.W. Cotterman, J.A. Senn), John Wiley & Sons Ltd, pp. 257-269, 1992

[12] Olle, T.W., Hagelstein, J., Macdonald, I.G., Rolland, C., Sol., H.G., Van Assche, F., Verrijn-Stuart, A.A., *Information Systems Methodologies – A Framework for*

Understanding. (2nd edition) Addison-Wesley Publishing Company, The Bath Press, Avon, 1991.

[13] Brinkkemper, S., Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, 38, pp. 275-280, 1996.

[14] Harmsen Frank, Marnix Klooster, Sjaak,Brinkkemper, Gerard Wijers: Intranet Facilitated Knowledge Management: A Theory and Tool for Defining Situational Methods. CAiSE 1997: 303-317, 1997.

[15] CCTA (Central Computer and Telecommunication agency) *SSADM4+:reference manual*, NCC Blackwell, 1995

[16] FIPS, Integration definition for function modeling (IDEF0), *Federal Information Processing Standards Publication*, 183 (FIPS 183), 1993.

[17] Booch, G., Jacobson, I., Rumbaugh, J., *Unified Modeling Language: version 1.0*, Rational Software Corporation, 1997

[18] Vlasblom, G., Rijsenbrij, D., Glastra, M., (1995) Flexibilization of the methodology of system development, *Information and Software Technology*, Elsevier-Science B.V., 37, No. 11, pp. 595-607, 1995.

[19] Kotteman, J., Konsynski, B., Information Systems Planning and Development: Strategic Postures and Methodologies. *Journal of Management Information Systems*, Vol. 1, No. 2, pp. 45-63, 1984.

[20] Naumann, J., Davis, G., McKeen, J., Determining information requirements: A contingency method for selection of a requirements assurance strategy. *The Journal of Systems and Software*, 1, pp. 273-281, 1980.

[21] Baskerville, R., Wood-Harper, T., A critical perspective on action research as a method for information systems research. *Journal of Information Technology*, July, 1996.

[22**]** Kelly S., et.al., *Metaedit+ - A Fully Configurable Multi User and Multi Tool CASE and CAME Environments*, Conference on Advanced Information Systems Engineering (Caise96), Constantopoulos, Mylopoulos and Vassiliou (eds.), Springer Verlag, 1996.

[23] Harmsen F., et al, Situational Method Engineering for Information System Project Approaches, in Methods and Associated Tools for the Information Systems Life Cycle, Verrijn-Stuart and Olle (eds.), Elsevier, 169-194, 1994

[24] Harmsen Frank, Sjaak Brinkkemper: Design and Implementation of a Method Base Management System for a Situational CASE Environment. APSEC: 430-438, 1995.

[25] Rolland Colette, Véronique Plihon: Using Generic Method Chunks to Generate Process Models Fragments. ICRE 1996: 173-181, 1996

[26] Rolland Colette, Challenges in Object Oriented Modelling: From Conceptual Modelling to Requirements Engineering. In Proceedings of OOIS' 1996.

[27] Brinkkemper S., Saeki M., Harmsen F., Assembly Techniques for Method Engineering, Proc. CAiSE 98, Pernici B. Hanos C. (eds.) LNCS 1413, Springer, 381-400, 1998.

[28]] Grundy J.C. and Venable J.R., Towards an Integrated Environment for Method Engineering, in Method Engineering Principles of Method Construction and Tool Support, Brinkkemper, Lyytinen, and Welke (eds.) Chapman and Hall, 45-62, 1996.

[29] Gupta D. and Prakash N., Engineering Methods form Method Requirements Specification, Requirements Engineering Journal, 6, 3, 135-160, 2001

[30] Ralyté J; Deneckère R., Rolland C., Towards a Generic Model for Situational Method Engineering, Proc. CAiSE 2003, Eder J. & Missikoff M. (eds.) LNCS 2681, Springer, 95-110, 2003.

[31] Prakash N and Goyal S.B, Towards a Life Cycle for Method Engineering, Proceedings Eleventh International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'07), 27-36, 2007

[32] Prakash N and Goyal S.B, Method Architecture for Situational Method Engineering, 2nd IEEE International Conference on Research Challenges in Information Science (RCIS'08), Marrakech, Morocco, ISBN 978-1-4244-1677-6, p.p. 325-336, 2008