CREWS Report Series 98 - 21

ARE METHODS REALLY USEFUL?

C. Rolland

University Paris-1 Sorbonne, 17, rue de la Sorbonne, 75005 Paris cedex 5, FRANCE

email : rolland@univ-paris1.fr

Appeared in the proceedings of the 20th International Conference on Software Engineering ICSE'98, Kyoto, Japan, April 1998.

Are Methods Really Useful?

Colette Rolland
Université de Paris1 Sorbonne
17, rue de la Sorbonne
75231 Paris Cedex 05
France
rolland@univ-paris1.fr

There are a large number of methods available for information systems development (ISD). These include structured approaches, prototyping approaches, systemic approaches, object-oriented, etc. Many of these methods have been comparatively analysed in books [OLLE88] and journals (e.g [HIRS92]). Despite a large body of work concerning details of systems development methods, there is still a poor understanding of how such methods are actually used in practice [WYNE93]. Thus, there is a felt need for empirical evaluation of the use of methods.

Besides, there is an increasing feeling that methods are not well-suited [LYYT87] to the needs of their users, the IS developers. In particular, it is necessary to change methods from one business situation to another. Several survey based studies have revealed that ISD methods are developed or adapted locally. For example, a survey of method use in over 100 organisations' [RUSS95] shows that more than 2/3 of the companies have developed or adapted their methods in-house. Also, 89% of respondents believed that methods should be adapted on a project to project basis.Thus, there is a need to situate methods, i.e define methods as context dependent whereas they are today considerd to be domain independent.

Method engineering [WELK92] represents the effort to improve the usefulness of systems development methods by creating an adaptation framework whereby methods are created to match specific organisational situations. There are at least two objectives that can be associated to this adaptation. The first objective is the *production of contingency methods,* that is, situation-specific methods for certain types of organisational settings. This objective represents method engineering as the creation of a multiple choice setting. The second objective is one in which method engineering is used to produce *method "on-the-fly"*. Situational method engineering is the construction of methods which are tuned to specific situations of development projects. Each system development starts then, with a method definition phase where the development method is constructed on the spot.

There are four different concerns of method engineering : the *definitions of methods,* their *representations*, the *way of developing these representations*, and the *rationale* for using these representations. We comment them in turn.

Many definitions have been proposed and most of them converge to the idea that a method is based on models (i.e systems of concepts) and consists of a number of steps which must/should be executed in a given order. In other words a method is composed of a *product model* and a *process model*. However, it shall be noticed that in the past, method developers have concentrated on the definition of product models and therefore that the product aspect of methods has been favoured at the expense of the process aspect.

Representation of methods is based on *meta-modelling* around which the whole area of method engineering has developed. The more modern meta-models look for an integration of the process and product aspects of methods whereas earlier meta-models focussed on product aspects only. Meta-modelling per se does not tackle the important problem of *modular description of methods*. One proposal [HARM94] viewed a method component as either a *product fragment* or a *process fragment.* The drawback of the fragment based approach is the over-emphasis on the product aspect resulting in underdeveloped meta-process modeling. In addition, the process models underpining the meta-models are often activity-based Within the ESPRIT project NATURE, [ROLL95] proposed a *decision-oriented process meta-model* which places equal emphasis on the product and process aspects of methods. To our knowledge, these are the two proposals for defining method components. It is clear that additional work is needed before an agreed notion of method components can be arrived at. Defining appropriate modeling languages [SAEK91] is also an issue.

The method construction process calls for software support. Whereas Computer Aided System Engineering (CASE) tools support the development of information systems, Computer Aided Method Engineering (CAME) tools aim at supporting the development of methods. The design of CAME environments is a research issue involving a number of different problems such as *repository structuring* and *management*, *enactment mechanisms*, efficient *interpretation/execution* of process modelling languages, process descriptions *configuration management.* In addition, their integration with CASE environment remains unsolved. Besides, the need to integrate in both, CAME and CASE, enactment mechanisms to support process execution was shown. Finally even though the functionality of CAME tools has been rather well identified, implementation of tools with this full functionality has yet to be achieved.

The question of *why* we should use a meta approach in method definition must be addressed. It shall be pointed out that the straight-forward modelling of current methods is inadequate for solving any of the unsolved problems of IT acceptance in an organisation.

The challenge of method engineering is to understand why these problems are unsolved, relate them to organisational factors, and adapt methods to develop IT systems to the specific factors of the situation at hand.

[HARM94] Harmsen F et al, Situational method engineering for informational system project approaches, in Method and Associated Tools for the Information Systems Life Cycle, Verrijn-Stuart and Olle (eds.), North Holland, pp169-194, 1994

[HIRS92] Hirschheim R. and Klein H.K. Paradigmatic influences on information systems development methodogies:evolution and conceptual advances, Advances in Computers, 34, pp. 294-381, 1992

[HIDD94] Hidding G.J., Methodology information : who uses it and why not?, Proc. WITS-94, Vancouver, Canada, 1994

[LYYT87].Lyytinen K., Different perspectives on information systems : problems and solutions, ACM Computing Surveys, Vol 19, No1, 1987

[OLLE88] Olle T. W., J. Hagelstein, I. MacDonald, C. Rolland, F. Van Assche, A. A. Verrijn-Stuart, Information Systems Methodologies : A Framework for Understanding, Addison Wesley, 1988

[ROLL95] Rolland C., Souveyet C., Moreno M., An Approach for Defining Ways-Of-Working, Information Systems, Vol 20, No4, pp337-359, 1995

[RUSS95] Russo et al, The use and adaptation of system development methodologies, Proc. 1995 Intl. Resources Management. Association Conference, Atlanta, 1995

[SAEK91] Saeki M., Kaneko T. Sakamoto M., A method for software process modelling and description using LOTOS, Proc. 1st Intl. Conference on the Software Process, IEEE Computer Society Press, Los Alamitos, CA, USA, pp 90-104, 1991

[WELK92] Welke R.J, and Kumar K., Method engineering : a proposal for situation-specific methodology construction, in Systems Analysis and Design : A Research Agenda, Cotterman and Senn(eds), Wiley, pp257-268, 1992

[WYNE93] Wynekoop J., Russo N., System development methodologies:unanswered questions and the research-practice gap, in Proc.14th Intl. Conf. Inf. Syst., New York, ACM Pub. pp 181- 190, 1993