TWO FRAMEWORKS FOR UNDERSTANDING AND EVALUATING INFORMATION SYSTEMS METHODOLOGIES

Frances Bell, Dept of Computer & Mathematical Sciences, University of Salford, Salford, M5 4WT, F.Bell@mcs.salford.ac.uk

Briony J Oates, School of Computing & Maths, University of Teesside, Middlesbrough, TS1 3BA, B.J.Oates@tees.ac.uk

Abstract

We examine the use of two contrasting frameworks in the understanding and critical evaluation of single information systems methodologies. Criteria for comparison are the intentions or purposes of the framework as stated by the author(s); the apparent philosophy of the framework i.e. its view of 'methodology'; the 'process' or 'product' orientation of the framework; and the extent to which the stated purposes are achieved. We include some insights gained from our own experience of the use of the frameworks. Finally, we look briefly at current work in the combination of methodologies and raise some questions for framework users in this different context.

Introduction

It is by no means clear that information systems methodologies enable the delivery of information systems that satisfy users and meet organisational objectives. Methodologies are diverse, used differently (often piecemeal) and not universally accepted (Russo, Hightower, Pearson, 1996). Methodologies can be used for reasons *other* than increasing the effectiveness of the information systems development: "goal displacement", where following the methodology becomes the goal (Fitzgerald, 1995); or to alleviate stress, "methodology as a social defence" (Wastell, 1996).

This paper does not seek to justify the use of methodology but takes its use as given. What follows is based on the assumption that some *agreed* methodology will be chosen to support a given information systems development and that the choice of methodology may be supported by the use of a conceptual framework.

According to Reviron's and Hughes' research, there is a lack of critical reflection and reflection in practice, which may be alleviated by the inclusion of critical evaluation in full-time higher education and professional development courses (Reviron and Hughes, 1996). We have found that the use of methodological frameworks can provide some experience of critical evaluation.

Problem: which methodology?

There are now many information systems (IS) methodologies, hundreds according to Avison & Fitzgerald (1995) (thousands according to Jayaratna, 1994). The number and range of methodologies should help us find the "right" methodology for a given

situation. But like the British tourist at the French hypermarket cheese counter, we may find that increased choice can simply lead to bewilderment.

Which methodology do I choose?

How "good" is a given methodology?

Is it similar to another methodology, or completely different?

Will it meet my needs?

Solution: use a conceptual framework

"A conceptual framework ... is a meta-level model through which a range of concepts, models, techniques, methodologies can either be clarified, compared, categorised, evaluated and/or integrated." (Jayaratna, 1994, p42)

A conceptual framework can help us understand, compare and evaluate methodologies. However, there is now an increasing number of frameworks available to help us (Avison & Fitzgerald, 1996, Olle et al, 1991, Jayaratna, 1994, Bell & Oates, 1994 etc.). So which framework shall we use? We find we have replaced one set of questions with another, similar set:

Problem: which framework?

Which framework do I choose?

How "good" is a given framework?

Is it similar to another framework, or completely different?

Will it meet my needs?

Solution: read on ...

In this paper we present a review of two contrasting frameworks for the understanding and evaluation of information systems methodologies: Olle et al's Framework for Understanding (Olle et al, 1991) and NIMSAD - **N**ormative **I**nformation **M**odel-based **S**ystems **A**nalysis and **D**esign - (Jayaratna, 1994). We have found both of these frameworks useful, albeit in very different ways. Our analysis is intended to inform framework choice and use.

Background

Methodologies help us cope with reality by abstraction i.e. we build models, mental or physical, which focus on certain aspects of perceived reality and exclude others. For example, Soft Systems Methodology (Checkland, 1981, Checkland & Scholes, 1990) uses the notion of a 'human activity system' or holon. SSM users formulate some holons relevant to aspects of their perceived reality, and use them to investigate or consider a possible change to some aspect of the world around them. SSM does not, however, offer its users models to be used in, for example, designing the software architecture of a computer-based information system. Similarly, a framework for understanding methodologies takes a particular view of what a methodology is, and encourages the framework user to concentrate on some aspects of methodologies and not others. We demonstrate this in the following sections. Just as the choice of methodology is contingent on a particular situation, so must be the choice of framework.

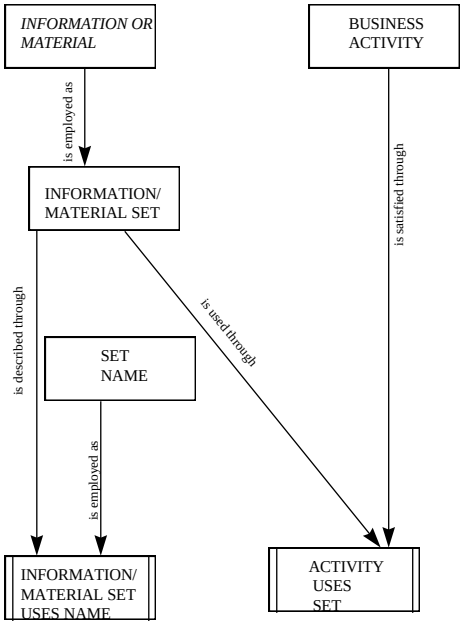
In this paper we examine two frameworks: Olle et al's Framework for Understanding (Olle et al, 1991) and NIMSAD (Jayaratna, 1994). They were chosen because they are in marked contrast. It is difficult to estimate the usage of these two frameworks by practitioners and academics as sales statistics for the books are not readily available, and there are no other commercial products associated with the frameworks whose sales could indicate their uptake. However, Web searches reveal that each framework has had some impact in academia, if such impact can be measured by the inclusion of the texts on course book lists, and personal recommendations.

In each case our findings are based on an analysis of the framework: as described in the cited texts and as used in our own work (Bell, 1996), (Oates, 1995), (Oates & Jayaratna, 1995), together with experience of their use with computing and IS students (final year undergraduates and postgraduates at two UK universities) during the past six years. For each framework we give a brief description and then concentrate on the intentions or purposes of the framework as stated by the author(s); the apparent philosophy of the framework i.e. its view of 'methodology'; the 'process' or 'product' orientation of the framework; and the extent to which the stated purposes are achieved. We include some insights gained from our own experience of the use of the frameworks.

Olle et al's Framework

This framework was initially published in 1989, and significantly updated in the second edition in 1990. It resulted from work in the 1980s at a series of CRIS (Comparative Review of Information System Design Methodologies) conferences (Olle et al, 1982, 1983, 1986,). We have used this framework with approximately 300 students in two different Universities over six years. Some of the students have gone on to use the framework in an industrial setting to support their project work.

At the highest level of abstraction, the text identifies 16 key concepts of a methodology (e.g. stage, step, design product). These are presented graphically using a data structure diagram where each concept is a data structure and the dependencies between these concepts are shown as the relationships between the data structures. Each of these key concepts is explained in the first three chapters. Ancillary concepts, not represented in the data structure diagrams, are also introduced e.g. analysis, design, situations and scenarios. The later part of the book deals with general issues such as representation, documentation and CASE tools.



Fragment of Business Analysis Stage Process Pesrpective Diagram, after Fig 4.3 from Olle (1991).

Figure 1

Most of the book (approximately 230 out of 400 pages) is concerned with component analysis. Possible design products of a methodology (e.g. a data flow diagram) are broken down into a set of related components or generic objects (e.g. business activity, information/material set, etc.) which are described in the text and modelled in a set of data structure diagrams (see Figure 1 for a segment of such a diagram). Olle et al provide 12 such diagrams: for each of three stages of development (information systems planning, business analysis and system design), and for each of three perspectives (data, process and behaviour) plus the cross-references between the three perspectives.

In component analysis, the components of the design products of a methodology are analysed against these data structure diagrams (which, it is claimed, represent the superset of all possible components) to see whether the methodology includes each of the components and their relationships. Worked examples are included but the methodologies are not named.

Olle et al see their framework as having three purposes, each for a specific audience (page v):

- Students and teachers already having an understanding of one information systems methodology - can use the framework to help understand methodologies relative to each other.
- Practitioners - can gain a perspective on methodological approaches free of the context of any specific methodology, and can also use the component checklist for evaluating possible methodological approaches.
- Researchers - can use the framework as a springboard.

The underlying philosophy of this framework is that of technical rationality, with systems development seen as a process of construction. A methodology can be analysed without regard to how it might be interpreted by methodology users, or be

changed in response to its context of use. The framework seems to assume that a methodology is primarily a technical artefact which can assist in the development of information systems with identifiable pre-existing requirements. Furthermore these requirements will be expressed diagrammatically through relationships between instances of generic objects, rather than expressed through natural language statements. Even a structured method like SSADM (Goodland with Slater, 1995) is enriched by textual techniques such as requirements and function definition which complement the diagrammatic techniques (e.g. Logical Data Structures and Entity Life Histories) more readily associated with the method. Olle et al's analysis does not address the textual components of a method.

The emphasis on component analysis gives this framework a predominantly product focus, at a fine level of detail. A product focus can be useful to CASE tool designers and those who need to re-use design products from other methodologies (Bell and Oates, 1994). However, no attention is paid to the *quality* of those products. For example, there is no support for examining how the quality of the design product components may be affected by the process and the user of the methodology.

When comparing one structured methodology against another, we have found that component analysis reveals differences at the level of 'Which components are produced by each methodology?' but conceals any differences in representations (e.g. graphical notations). Additionally it relies on the existence of neutral, generic object types, hence its emphasis on unambiguous definition of the components of design products. It assumes that the concept of business activity, for example, has the same meaning in any methodology, and to any methodology user or indeed component analysis user.

We have found that the framework does, to some extent, satisfy its first purpose in that it can help students understand a given methodology relative to a generic structural framework i.e. to answer the question: 'How does the output of this methodology at each stage compare with Olle et al's component diagrams?' However the answer gained does not complete the evaluation. Olle et al themselves stress that we should not necessarily assume that the greater the number of components that a methodology includes, the better the methodology. Hence it is not clear how we should interpret the findings of the component analysis. Where students already know a methodology, Olle et al's framework can enrich their understanding e.g. component analysis of the Yourdon Structured Method (Yourdon, 1989) has helped students to criticise that method's claim to support the behaviour perspective. However the framework was of no help to them in analysing the *process* or *usability* of the method.

Students generally find the style of the book difficult to read, and the diagrams and descriptions of low level components difficult to understand. The component analysis diagrams seem to have been derived 'bottom up' from the methodologies known to the framework authors at the time. The abstraction provided by stages and perspectives is only relevant to those methodologies which incorporate the same 'waterfall' life-cycle model and model a system from the same technical perspectives (data, process and behaviour). Where project managers are trying to map a different life cycle (e.g. a Rapid Application Development approach) the cross-references between *stages* (not covered by Olle et al) are as important as the cross-references between *perspectives*. Therefore, not only is Olle et al's framework only suited to conventional structured methodologies in the 'hard' or 'functionalist' paradigm (Hirschheim and Klein,

1989), but is further restricted to those which use a waterfall life cycle model. It is not applicable to 'softer' more interpretive methodologies, (e.g. SSM (Checkland, 1981, Checkland & Scholes, 1990)) nor can it be used for the newer object-oriented methodologies which are increasingly popular in systems development (e.g. Coad & Yourdon 1990,1991)

NIMSAD

Since its publication we have used the NIMSAD framework (Jayaratna, 1994) with approximately 160 students in two different universities, and in our own critical analyses of methodologies (Bell, 1996), (Oates & Jayaratna, 1995).

The NIMSAD framework considers three inter-dependent elements: the problem situation (methodology context), the intended problem solver (methodology user) and the problem-solving process (methodology). The fourth element of the framework is an evaluation of the other three elements, before, during and after the intervention i.e. the use of a methodology in a problem situation (see Figure 2). For each of the other three elements the evaluation element offers a large number of questions to prompt the framework user into considering the scope of the guidance provided by the methodology and its assumptions about the nature of reality, the role of the problem solver etc.

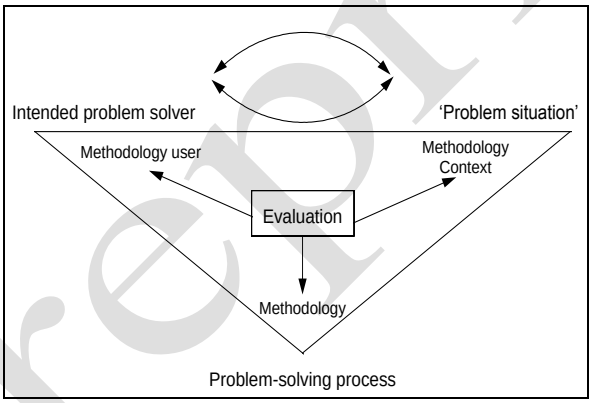


Figure 2 (after Jayaratna,1994)

The problem solver, situation and process elements are described in three chapters. Evaluation is covered in a further chapter which mostly comprises a large number of questions to be answered. Three example evaluations of methodologies (one chapter each) are also included: SSM (Checkland, 1981), (Checkland & Scholes, 1990), ETHICS (Mumford, 1983a, 1983b) and SASS (De Marco, 1979). The book is aimed at methodology selectors, users and clients and final year students and postgraduates. A number of aims of the book are listed (page xii):

- Help readers & users of methodologies break free from political constraints.
- Raise their consciousness to consider the desirability of their actions before their feasibility.
- Consider the effect of their actions on others as well as themselves.
- Enable more open debate and discussion.

- Minimise desire to achieve self-needs at the expense of others.

Hence the author seems very concerned with developing a strong ethical stance in his readers. The framework itself is described (page xvii) as a general framework that can be used for understanding and evaluating any methodologies, not simply those relating to information systems.

The underlying philosophy is based on interpretivism rather than positivism (i.e. a methodology is interpreted by humans, rather than being an independent physical entity). In order to understand a methodology we therefore have to consider its user(s) and the context in which it is being used.

Students have managed the analyses of problem-solver and problem context reasonably successfully, but have struggled to map NIMSAD's problem-solving process elements onto those of a methodology. The three phases of problem formulation, solution design and design implementation are usable, but the further breakdown of these three into eight stages is difficult for them to understand and apply. Only the *process* of problem-solving is considered. However, a methodology can also deal with, and produce, meta-data, which can be thought of as the products of a methodology (both intermediate and final) and are the focus of interest of Olle et al's (1990) component analysis. As noted in the previous section, the structure and relationships of this meta-data is of interest to CASE tool designers and method integrators.

The huge range of questions under Evaluation (Chapter 7) has promoted some rich analyses of methodologies. Students have successfully used NIMSAD to consider the kind of situation an existing methodology is suited to, and the demands it makes of its users. Their findings have been used to compare one methodology with another. With students, it has promoted discussion of the effect of a methodology on its users and the necessary 'mental construct' (Jayaratna, 1994) for using a particular methodology. It has also helped raise ethical issues. It also seems to enable students' realisation that individuals interpret a methodology, either consciously or unconsciously, rather than a methodology being fixed, and that a methodology may have to be changed during the course of an intervention.

In general, students seem to find the book fairly easy to read, as long as they have certain pre-requisite knowledge. Users of this framework need some knowledge of organisation theory, since no clear conceptual structure for the problem situation element is provided (Oates, 1995). Since NIMSAD is firmly based on systems theory, its users also need understanding and experience of systemic approaches. The NIMSAD user must also appreciate the different philosophical assumptions underlying methodologies. Anyone accepting the notions of technical rationality and positivism (Schon, 1983), who assumes that there can be objective knowledge free from subjectivity and normative constraint, would have difficulty in appreciating the need to consider other approaches to knowledge, reality and truth (Oates, 1995). The NIMSAD user should not assume that a methodology is static and independent of its users and context and must understand the need for critical self-reflection. NIMSAD is more likely to enable us to address the question raised by Flood and Romm

'How can we make choices between interests and purposes while remaining accountable?'(Flood and Romm, 1997),

Users of this framework are likely to be driven by a spirit of enquiry, rather than seeking firm guidelines on how to choose the appropriate methodology for a given situation. Our experiences with students confirm the findings of Biggam and Hogarth (1996) that NIMSAD helps by posing questions to assist with evaluation but offers little support for interpreting the answers. Also, the likely NIMSAD user is someone who is more interested in methodologies which handle organisational development issues and requirements analysis. Those wanting to understand methodologies which help their users (often a team) to design, construct and implement computer-based systems, will find the problem-solving process elements of NIMSAD covering design and implementation are not covered to the same depth as the earlier stages (Oates, 1995).

Conclusions

In conclusion we offer a comparison of Olle et al's framework with NIMSAD.

Olle et al take a 'hard' view of methodology, focusing on the design products of structured methodologies, whereas NIMSAD takes a 'soft' view, focusing on the process and context of use of problem-solving methodologies. Olle et al, in effect, provide a checklist of design components, a detailed, reductionist view. NIMSAD provides a multitude of open-ended questions and takes a more holistic view, asserting that a methodology cannot be considered separately from its users and the problem situation. Olle et al help us produce a low level description of a static technical artefact, whereas NIMSAD helps us analyse a dynamic human situation involving the use of a methodology. In contrast to the numerous and complex graphical models in Olle et al's framework, NIMSAD provides a simple, memorable model which builds on previous work on the problem solver, problem situation and methodology (Checkland, 1981), (Avison and Wood-Harper, 1990). A further strength of NIMSAD is its inclusion of evaluation before, during and after the intervention - encouraging a longitudinal study of the use of a methodology. Both frameworks can help us gain a richer understanding of a methodology, but it must be stressed that neither provides help in interpreting the findings derived from using the framework - this is left to the judgement of the framework user.

We have found that each of these frameworks has, in its own way, helped students to achieve a critical *understanding* of information systems methodologies. Practitioners are likely to be as interested in the evaluation and comparison of methodologies to support methodology *choice* and *use*. NIMSAD encourages the construction of a narrative before during and after about the role of methodology use in an intervention, as shown in Figure 3, whereas Olle et al's framework considers the methodology as separate from its context of use.

Links to Other Work

Both frameworks (Olle et al's and NIMSAD) tend to assume that a *single* methodology is being evaluated. In recent years interest has developed in the combination of methodologies, particularly in the disciplines of Software Engineering, Information Systems and Management Science.

In Software Engineering, work has been done in the integration of formal and structured methods (Kronlöf, 1993), (Polack, Whiston, Mander, 1993), (Semmens, Allen, 1991), and also in *method engineering* where a situation-specific methodology

is built, often within the context of a configurable CASE (Computer Aided System Engineering) tool, (Kumar and Welke, 1992), (Brinkkemper et al, 1996). A recent publication names this approach as *a new methodology paradigm*, (Hidding, 1997). Hidding advocates the "configuring of a one-of-a-kind methodology from common building blocks". Methodology material (for reading) is deconstructed into components (method processes, work products and job aids) which can be re-combined into a situation-specific methodology. The structure/syntax focus of this approach has much in common with Olle et al's approach except that these 'building blocks' may be at a coarser level of granularity than Olle et al's fine-grained 'components'. A difference is that Hidding is concerned with methodology process as well as product.

In information systems, pluralist approaches have tended to focus on the combination of Soft Systems Methodology with other techniques and methods, often from the functionalist paradigm (Wood 1992).

Jackson recognises Multiview (Avison, Wood-Harper, 1990) as "the longest running attempt to bring together soft and hard approaches to information systems development" but finds that Multiview shares with other 'soft/hard' combinations the theoretical uncertainties associated with dealing with a plurality of rationalities.(Jackson, 1997).

Methodological pluralism has been a matter of debate within Management Science. A recent publication (Mingers and Gill, 1997) explores the theory and practice of the combination of methodologies (and parts of methodologies) within a single intervention, termed *multi-methodology*. A wide range of approaches is covered: from Flood & Romm's triple loop learning (Flood & Romm, 1997); through Jackson's notion of pluralism as an essential part of "critical systems practice" (Jackson, 1997); to Mingers' "critical pluralism" (Mingers, 1997); and White and Taket's "pragmatic pluralism" (White and Taket, 1997).

The move away from the "simple" choice of a single methodology towards a more complex choice of combination (and tailoring) of methodologies for use in an intervention creates new demands.

What use can frameworks be in such a situation?

Can they guide us in our method integration or multi-methodology practice?

NIMSAD's emphasis on before, during and after intervention evaluations may help us to critique our practice but does not provide a theory of multi-methodology itself.

Our analysis has presented these two frameworks as a dualism between product and process, and has analysed their use in the education of future practitioners. Olle et al's framework's focus on fine-grained components at the expense of process and people limits its usefulness in the evaluation of methodology choice and use. NIMSAD's holistic approach can support rich evaluation of single methodologies in use to inform, but not fully support, the more complex methodology choice demanded by the combination of methodologies.

References

Avison, D.E. and Fitzgerald, G. (1995) *Information Systems Development: Methodologies, Techniques and Tools*, McGraw Hill, UK

Avison D.E. and Wood-Harper A.T. (1990), *Multiview: An Exploration in Information Systems Development*, Blackwell Scientific Publications.
Bell, F. (1996) Evaluation of the Multiview Methodology using the NIMSAD Framework, *Proceedings of 4th BCS Conference on IS Methodologies*, Cork.
Bell, F. and Oates, B.J. (1994) A Framework for Method Integration, *Proceedings of 2nd BCS Conference on IS Methodologies*, Edinburgh.
Biggam, J. and Hogarth, A. (1996) Normative Information Model-based Systems Analysis and Design (NIMSAD): A Preliminary Evaluation, *Systemist*, Vol. 18(1), Feb. 1996.
Brinkkemper, S., Lyttinen, K., and Welke R.J. (Eds), (1996) Method Engineering: Principles of method construction and tool support. In *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering. (*Aug. 26-28, 1996, Atlanta). Chapman & Hall, London, 1996.
Checkland, P.B. (1981), *Systems thinking. Systems practice*, Wiley, UK.
Checkland, P.B. & Scholes, J. (1990) *Soft Systems Methodology in Action*, Wiley, UK.
Coad, P. and Yourdon, E. (1991a) *Object-Oriented Analysis*, 2nd edition, Prentice Hall, New Jersey, USA.
Coad, P. and Yourdon, E. (1991b) *Object-Oriented Design*, Prentice Hall, New Jersey, USA.
De Marco, T. (1979) *Structured Analysis & Systems Specification*, Yourdon Press, New York.
Fitzgerald B. (1995), "A Descriptive Framework for Investigating Problems in the Application of Systems Development Methodologies", *Proceedings of 3rd BCS Conference on IS Methodologies*, Wrexham.
Flood, R. and Romm, N. (1997), From Metatheory to "Multimethodology", in *Multi-methodology*, ed. Mingers, J. and Gill, A., John Wiley, 1997.
Goodland M., Slater C. (1995) *SSADM - A Practical Approach*, Mc Graw-Hill, UK.
Hidding, G.J., Reinventing Methodology: Who Reads It And Why?. *Communications of the ACM*, vol 40, Number 11.
Hirschheim, R. and Klein, H.K. (1989) Four paradigms of information systems development, *Communications of the ACM*, vol 32(10).
Hughes, J., Reviron, E., Selection and Evaluation of Information Systems Methodologies: The Gap between Theory and Practice, *Proceedings of 4th BCS Conference on IS Methodologies*, Cork.
Jackson, M. (1997), Pluralism in Systems Thinking and Practice, *Multi-methodology*, ed. Mingers, J. and Gill, A., John Wiley,
Jayaratna, N. (1994) *Understanding and Evaluating Methodologies. NIMSAD: A Systemic Framework*, McGraw-Hill, UK.
Kronlöf K (1993), *Method Integration, Concepts and Case Studies*, John Wiley & Sons.
Kumar, K. and Welke, R. Method engineering: A proposal for situation-specific methodology construction. *Challenges and Strategies for Research in Systems Development.* W.W. Cotterman, and J.A. Senn, Eds, John Wiley, Washington D.C., 1992.
Mingers, J. and Gill, A. (ed.) (1997), *Multi-methodology*, , John Wiley, Chichester, England, 1997.
Mingers, J (1997),Multi-paradigm Methodology, *Multi-methodology*, ed. Mingers, J. and Gill, A., John Wiley.

Mumford, E. (1983a) *Designing Participatively*, Manchester Business School, Manchester.
Mumford, E. (1983b) *Designing Human Systems* Manchester Business School, Manchester.
Oates, B.J. (1995) Evaluation of the NIMSAD Conceptual Framework, *Proceedings of 3rd BCS Conference on IS Methodologies*, Wrexham.
Oates, B.J. and Jayaratna, N. (1995) Evaluation of Yourdon Systems Method Using the NIMSAD Conceptual Framework, *2nd Scandinavian Research Seminar on Information and Decision Networks*, Vaxjo, Sweden.
Olle, T.W., Sol, H.G. and Verrijn-Stuart, A.A. (eds) (1982) *Information Systems design methodologies: A comparative review*, North Holland, Amsterdam.
Olle, T.W., Sol, H.G and Tully, C.J. (eds) (1983) *Information Systems design methodologies: A Feature analysis*, North Holland, Amsterdam.
Olle, T.W., Sol, H.G and Verrijn-Stuart, A.A. (eds) (1986) *Information Systems design methodologies: Improving the Practice*, North Holland, Amsterdam.
Olle T.W., Hagelstein, J., Macdonald, I.G. et al (1991) *Information Systems Methodologies. A Framework for Understanding*, Addison Wesley, Second Edition.
Polack F., Whiston M., Mander K. (1993), *The SAZ Method Version 1.1*, University of York.
Russo, N.L., Hightower, R., Pearson, J.M. (1996), The Failure of Methodologies to Meet the Needs of Current Development Environments, *Proceedings of 4th BCS Conference on IS Methodologies*, Cork.
Semmens L. & Allen P. (1991), 'Using Yourdon and Z: an Approach to Formal Specification', *Proc. of 5th Annual Z User Meeting*, Oxford, Springer-Verlag.
Wastell D. (1996), "The fetish of technique: methodology as a social defence", *Information Systems Journal*, Vol. 6, pp25-40.
White, L & Taket, A., (1997), Critiquing Multimethodology as Metamethodology: Working Towards Pragmatic Pluralism, *Multi-methodology*, ed. Mingers, J. and Gill, A., John Wiley.
Wood, J.R., (1992), Linking soft systems methodology and information systems, *Systemist*, 14, 133-135.
Yourdon, E. (1989) *Modern Structured Analysis*, Prentice Hall, New Jersey, USA.