Keywords

Design, design concerns, creating diagrams, business rules

INTRODUCTION

Compared to the number of information systems professionals using a methodology, there are far fewer of us who actually develop a new diagram. That said, the contribution a new diagram can make to the industry is potentially enormous. Imagine the impact of a diagram that models entities or objects better than the ones we currently use.

But what do we actually know about how to go about developing a new diagram. Is there an established wisdom? Is there a body of knowledge that can be drawn upon to guide and assist us during the process? We might start by asking questions such as what makes a good diagram, who will use it and for what purpose, where does and should it sit within a methodology and so on?

This paper reflects on an action research study during which a diagram to capture business rules was developed. It reflects on the process and in particular describes three systemic design concerns that emerged.

The paper proceeds as follows. In the following section, some background to information requirements determination is covered and then this is followed by introducing the idea of a business rule. In these sections only sufficient discussion is provided here to enable the reader to appreciate the relationship with systemic design concerns. Further information on business rules is available in McDermid (2007). The next three sections each address one of the systemic design concerns. The paper closes with a brief summary.

BACKGROUND TO INFORMATION REQUIREMENTS DETERMINATION

There has been continuing attention to information requirements determination (IRD) in the literature, dating as far back as Miller (1964). Most authors divide the process of IRD into two main steps: **requirements acquisition** and **requirements modelling** (Valusek and Fryback 1987, Greenspan et al. 1994). There are exceptions however, for example Vitalari (1992) identifies three steps of requirements generation, assessment and specification. The problems of the early stages of IRD concerned with acquiring or eliciting requirements have always been a concern. For example, Vidgen (1997) considers stakeholder approaches while Darke and Shanks (1997) examine a related concept called viewpoint modelling and Flynn and Davarpanah Jazi (1998) recommend user-led construction of requirements.

Techniques which are designed for requirements **acquisition** or elicitation tend to use some form of informal representation, typically narrative statements of requirements which lack structure. Byrd et al. (1992) identify eleven requirement acquisition techniques, namely prototyping (Naumann and Jenkins 1982), open interview (Davis 1982), brainstorming (Davis 1982), goal-oriented approach (Zmud et al. 1992), cognitive mapping (Eden 1988), variance analysis (Mumford 1985), repertory grids (Gutierrez 1987), scenarios (Mittermeir et al. 1987), structured interviews (Davis 1982), critical success factors (Rockart 1979) and future analysis (Land 1982). To this list, Darke and Shanks (1997) add joint application design (Avison and Fitzgerald 1995) and focus groups (Leifer et al. 1994) and conceivably others could be added such as the Soft Systems Methodology of Checkland (1981).

Informal representation carries the significant disadvantage that it is unstructured i.e. it does not require the identification of specific types of constructs which are perceived to form part or all of a requirement. In all of the above techniques except prototyping, there is no structure in the end-point in terms of pre-defined constructs which specify requirements and so important aspects of an information requirement may well be omitted, ambiguous or assumed.

Unless natural language sentences are **structured** (or classified) so as to ensure that all aspects of completeness of an information requirement are declared i.e. so that constructs (e.g. states, events, conditions) are **explicit**, then it would be impossible to ensure that problems of incompleteness or ambiguity could not arise. The term **structured** is used here in preference to the term “semi-formal” and also to avoid confusion with the term “formal” that is often reserved for specification techniques based on mathematics (Pohl 1994).

While the advantages of structure are attractive, it has to be accepted that the relatively dynamic business of requirements acquisition probably needs to take place first in a comparatively unstructured fashion, but thereafter it is desirable to put structure on requirements. This latter process is typically referred to as **requirements modelling**. Requirements modelling also facilitates validation of earlier assertions about requirements (Vitalari 1992).

There are many views about what specific constructs are appropriate for defining requirements. Moreover, arguably most requirements models have been developed by those perhaps more concerned with the later stages of the system development life cycle and so a major epistemological and semantic mismatch potentially arises in terms of the consistency and integration between the (informal) **acquisition models** and (structured) **requirements models**.

As long as information systems can be smoothly developed from a structured information requirements specification, such a specification can be seen as an appropriate endpoint to IRD in that it would be a complete and structured representation of preceding effort in determining information requirements. This leads to the question of converting from one kind of model to another and about what specifically constitutes an agreed endpoint to IRD. A **first** systemic design challenge therefore is concerned with how to convert an informal representation of information requirements to a structured representation without loss of information. This is closely related to a **second** challenge, which is to identify those constructs that are necessary to define information requirements. A **third** challenge is apparent when considering the complete life cycle of information systems development. Necessarily, a number of models of information systems are required over the development life cycle. Raw information requirements have to be elaborated to include data item validation criteria, design dialogue information etc. A copious amount of information is generated and the collection and presentation of that information has to be carefully managed otherwise information overload ensues. Decisions on what information or constructs are relevant at a particular stage in the life cycle as well as how later constructs can be obtained from earlier constructs lie at the very heart of this challenge. In other words, not only is it important to identify the constructs of a structured requirements model, but some idea of the number and type of models (i.e. the levels of abstraction) required to develop an information system is also needed.

INTRODUCTION TO BUSINESS RULES

Discussion is based on an action research study into business rules in two different organisations. Only sufficient discussion is provided to enable the reader to appreciate the context of systemic design concerns. In the first organisation the researcher worked with a systems analyst on an existing system. The outcome of this study was the production of an initial definition of a "business rule" and a means of diagramming it. In the second organisation, an analyst and users were studied using this Business Rules Diagram (BRD). In this part of the study, the definition of a business rule was further refined. Over the course of the action research study, six different versions of a (structured) BRD were developed. Earlier versions contained fewer constructs; as each version was evaluated, it was concluded that there was a need for additional constructs to ensure completeness in the description of a business rule. In all, four major constructs were identified.

Business rules may be considered as those invariants in a human activity system which express an aspect of policy underpinning a human activity system. A major difficulty is the large number of models that may be argued to express some aspect of a business rule. In one way or another most models or diagrams popular in information systems can be perceived to do this. However, one useful approach is to classify models in terms of their predominant ontology as discussed by Ilvari et al. (1998). Accordingly models may be classified into:

- extended state-based models i.e. those whose predominant philosophy is based on the state construct or extensions thereof (e.g. Shlaer and Mellor 1992);
- condition-based models (e.g. Herbst 1996);
- extended static models (e.g. Kappel and Schrefl 1989);
- trigger-based models (e.g. Flynn and Davarpanah Jazi 1998); and
- event-based models (e.g. Martin and Odell 1995).

The Business Rules Diagram (BRD) is an example of the extended state-based models category. It should be noted however, that within any one classification there is still a wide range of opinion and variation possible.

While business rules may be considered a type of information requirement, business rules do not represent all types of information requirement in two ways. Firstly, a business rule is limited to any functional information requirement, in contrast to a non-functional requirement which, for example, may refer to a performance or economic requirement (Greenspan et al. 1994). Secondly, there are sources of information requirements which do not rely on business rules. For example, an executive information system may gather much of its data from external sources where clearly human activity systems are out of its control. Here, information requirements would not be generated from business rules. On the other hand, a significant portion of IRD activity focuses on organisations **tracking** their human activity systems (Wand and Weber 1995). For instance, a manufacturing system tracks the progress of production, a merchandising company would need to track the progress of orders, universities and hospitals need to track the status of students and patients respectively. In all these cases business rules arise out of the need to track human activity systems.

SYSTEMIC DESIGN CONCERNS

Systemic Design Concern 1 – Converting an Informal Model to a Structured Model

It was decided to follow a formal conversion process as rigorously as possible and here the work of Wand and Weber (1993) was used as a framework to assist in this conversion. This framework is based upon a particular ontological model developed by Bunge (1977, 1979). Ontology is a branch of philosophy concerned with articulating the nature and structure of the world. Many ontologies are possible and indeed different versions of the BRD developed during the action research study contained different ontological views of the world. Of particular relevance here is the notion of mappings of ontological constructs onto design constructs. Ontological constructs are those constructs which are asserted to represent relevant constructs in the real world. For example, it may be asserted that events are a fundamental construct of the real world and so the ontology would reflect that. Design constructs are the set of constructs which exist in the design world and these should represent the constructs in the ontological world. In other words, design constructs are syntactic representations of semantic concepts in the ontological world. So, in the above example, there would need to be a notational symbol to represent the semantic concept of event. This approach formalises the systemic design concern and through formalising it, forces the designer to consider possible ontologies as well as manage the process of selecting an ontology. Wand and Weber (1993) speak of the mathematics of mappings in which the real world is mapped via scripts into the machine world (figure 1). In the context of this research, the real world is a problem situation in which information requirements exist informally and script 1 could be a structured information requirements model (e.g. a Business Rules Diagram). Script 2 could be an analysis model (e.g. a data model with ancillary data dictionary), script 3 a design model etc.

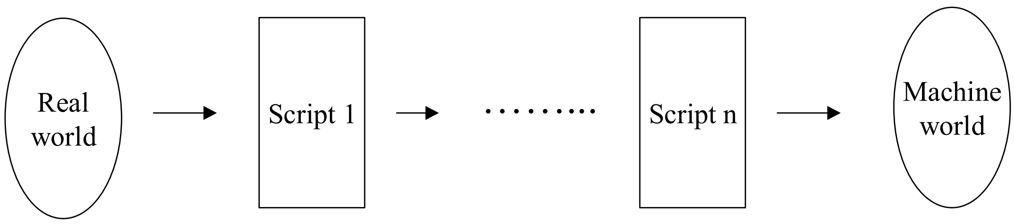
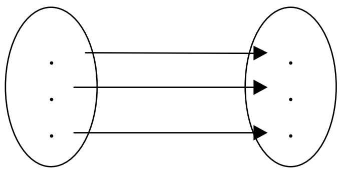


Figure 1 Mapping of Scripts from the Real World to the Machine World (after Wand and Weber 1993)

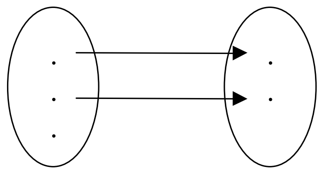
The power of the framework is derived from considering whether mappings between the ontological world and the design world are incomplete. The term world is used here to describe the complete set of constructs in a given domain of interest. Figure 2 shows five scenarios. In figure 2a, ontological completeness is shown. This means that there is an exact mapping between the ontological world and the design world i.e. all ontological constructs have a corresponding design construct and all design constructs have an ontological construct. In figure 2b however, there is ontological incompleteness. Here, one particular ontological construct does not have a corresponding design construct. Wand and Weber (1993) argue that ontological incompleteness is undesirable because the design world would not be able to represent all real-world phenomena. Figure 2c depicts a situation called construct overload. This occurs when one construct in the design world represents two (or more) constructs in the ontological world. Again, this is argued to be undesirable because it lacks clarity and may lead to confusion or ambiguity when reading design artefacts. Figure 2d shows construct redundancy which in one sense is the opposite of construct overload. In construct redundancy there are two (or more) constructs in the design world that can represent a single construct in the ontological world. This is considered undesirable because designers have to work with more constructs than necessary as well as leading to possible confusion.

Lastly, figure 2e describes construct excess. Construct excess arises when a construct exists in the design world that does not have a counterpart in the ontological world. The presence of construct excess may indicate three things. Firstly, it may indicate deficiency in the ontological world. Secondly, it may suggest that the design world contains an unnecessary construct i.e. one which is outside the scope of interest of the ontological world. Thirdly, it may be symptomatic of a general lack of understanding about the nature of the problem.

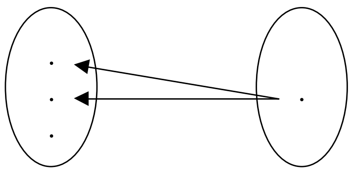
The Wand and Weber framework can therefore be seen as an intellectual tool to assist in obtaining a structured model from an informal one, though in itself this particular framework does not suggest which specific constructs may be of interest. The design aspects of this tool are clear. The tool provides a framework for reasoning about and evaluating the quality of a candidate diagram in terms of the degree to which it successfully models its problem domain. Few, if any, of the diagrams in use today within the information systems community have employed a rigorous systemically valid approach to their design such as this one has. As noted by Fitzgerald (1991, p663) some time ago, '*At this point we thought that we would examine the way that other systems development techniques had been validated. This was more difficult than at first thought because… those that were available did not appear to have been validated'.*



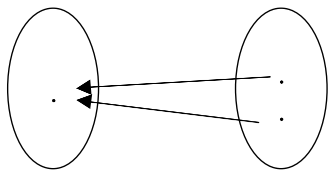
(a) Ontological Completeness



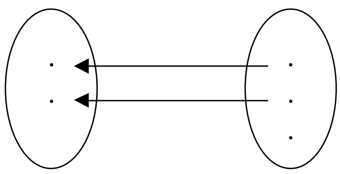
(b) Ontological Incompleteness



(c) Construct Overload



(d) Construct Redundancy



(e) Construct Excess

Figure 2 Five Scenarios in Ontological Framework Mapping, after Wand and Weber (1993)

Systemic Design Concern 2 – Identifying the Constructs

Perhaps not surprisingly, identifying the constructs was the activity that took up the most time in this research. The process that was followed was essentially one of brute force involving trialling permutations of different constructs to establish with stakeholders to what extent they modelled their world well and were fit for the purpose there were designed for. By the end, four constructs were considered to meet the goal of the research and these are outlined in figure 3.

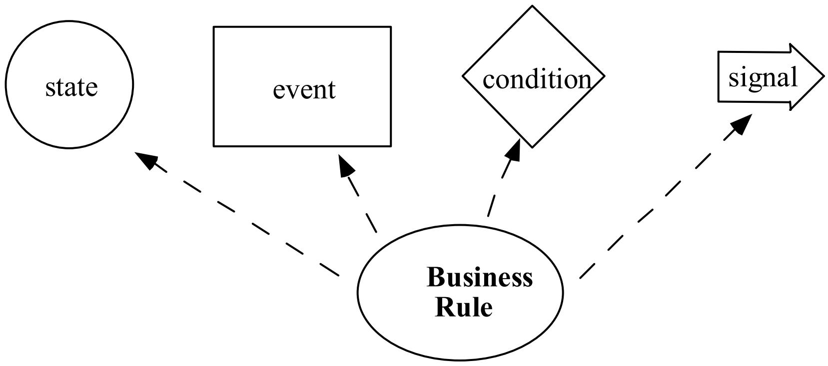


Figure 3 Abstraction of Key Constructs of a Business Rule

While the process of selection of constructs was brute force i.e. trial and error with different permutations of constructs, the question of complexity through cognitive overload was something that was attended to in a more formal manner. While there is little in the way of quantitative guidance on for example, how many symbols a human can absorb and manipulate in one diagram, it was clear that this was a key issue for a diagram to be accepted and widely used. Two techniques were chosen. One was a modification of McCabe's complexity algorithm (McCabe 1976) and the other from Rossi (1997).

McCabe (1976) provides a well-established graph-theoretic measure of complexity. The metric was developed as an aid for testing and maintaining computer programs and is based on quantifying the decision structure within a computer program. The cyclomatic number V(G) of a graph is:

$$V(G) = e - n + p$$

where e is the number of edges,
n is the number of vertices and
p is the number of connected components.

The theory is based on the assumption that each program has one entry and one exit point. The complexity of a graph is defined as 'the minimum number of paths that can, in (linear) combination, generate all possible paths through the module' (Watson and McCabe 1996, p11) or less formally the number of independent paths through a graph. Using McCabe's formula served to provide an albeit approximate measure of how complex the BRD was becoming as more pathways were being added.

Though not as well established as McCabe's metric, Rossi (1997) introduces a complexity metric that takes into account the number of 'different constructs' in a diagram. Based on the work of Rossi and Brinkkemper (1995), the metric seeks to take account of the inherent complexity in a diagram based on the number of different object types in a diagram, the number of different ways the object types may inter-relate and the total number of properties of object types in a diagram. The notational complexity of a diagram is defined as follows.

$$C = \sqrt{(X^2 + Y^2 + Z^2)}$$

Where C is the complexity,
X is the number of object types in the diagram,
Y is the number of relationship types in the diagram and
Z is the number of properties in the diagram.

Together they provided an additional means of managing complexity in a diagram by providing a quantitative indication for evaluation purposes of complexity.

Systemic Design Concern 3 – Selecting Levels of Abstraction

Levels of abstraction is a technique familiar to information systems and computer science because it allows the modeller to focus on what is important at that juncture and ignore what is not. While the Wand and Weber framework outlines the strategy for mapping between worlds it does not provide guidance in terms of how many mappings are appropriate or indeed how much intellectual distance there should be between mappings of worlds. Decisions on these matters need to be taken in the context of the problem domain - for example, how the stakeholders conceptualise their world, how much information exists in that world that could be reasonably captured in a diagram and so on. From a diagram developer's perspective the concerns include complexity and cognitive overload. What follows is what emerged in the action research study.

The results of the research suggest that it is desirable that at (at least) three types of rule in a business are identified and it is suggested that each might be better kept in separate repositories. However, the topmost level of business rule can be captured as a true conceptual model of the business in the sense that it can show rules which are unconstrained by how people in the organisation execute these rules or indeed how a computerised information system might implement these rules. Figure 4 depicts three levels of business rules.

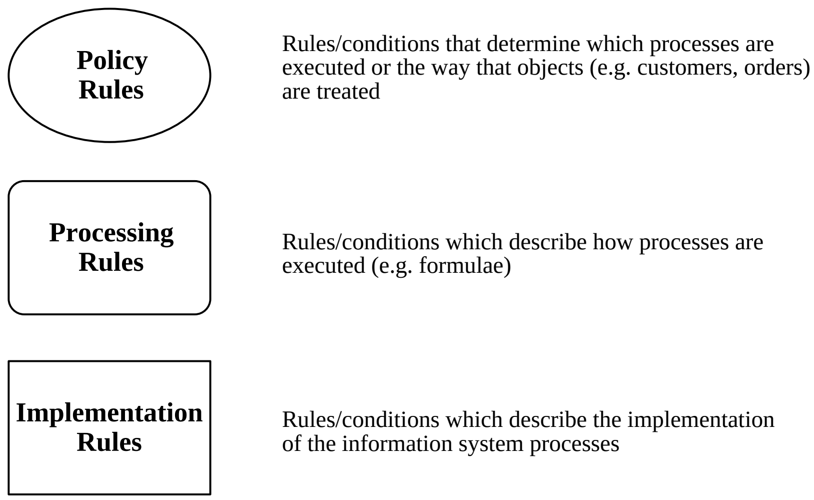


Figure 5 Levels of Abstraction of Business Rules

At the highest level such business rules may be termed policy rules in the sense that they reflect the essence or core of what the business is about. In other words, these may be considered raw information requirements unconstrained by the specifics of how that policy might be carried out. For example, in a banking system, customers in an overdraft state would typically be treated differently from customers with a positive account balance and further, different processes and procedures would be likely to apply. Observe that such rules would probably have to contain the pre-conditions which fire processes or activities in the business e.g. overdraft customers are given higher interest rates, reminder letters and so rules such as 'if overdraft customer then give higher rate, send reminder' would allow such policy rules to be applied. What the higher rate actually is, how it is calculated and how often a reminder is sent, may well **not** be part of the business policy. At the next level, what may be termed processing rules can be seen to exist. These rules arguably are also business rules. However, in comparison to policy rules they relate more to the detail of how the processes and procedures are actually carried out, rather than the condition(s) under which the business may choose to follow different processes. So, for example, a rule expressing the calculation of a sales tax or the detailed sequence of steps in accepting an order would be classified as a processing rule. In the latter situation, for it to be considered a processing rule rather than a policy rule, the organisational view would be that the specific steps in accepting an order were **not** in themselves a matter of business policy. Work on formal contracts (Kent 1997 and Kent and Gil 1998) illustrates the degree of detail and the level of formality considered necessary to specify processing rules.

At the lowest level of abstraction in figure 4 sits the implementation rule. Here the distinction lies in that the rule relates to how the process is actually implemented in the **information system** e.g. the batch printing of invoices for efficiency's sake would be an implementation rule rather than a processing rule.

Failure to distinguish and separate these levels of business rule would lead to confusion and overloading. Confusion would arise because various authors are working at different levels of abstraction. For instance, in the practitioner literature some examples of business rules are provided which seem more concerned with describing a level of abstraction equivalent to implementation rules as in '*employee ID is a numeric field between 1,000 and 3,000*' (Jones 1991, p9). The overloading problem is likely to occur because without these levels of abstraction any diagrammatic presentation of business rules would inevitably contain more than one type of rule if not all and therefore this would lead to too much information being presented. Indeed, this was a major design challenge in the research i.e. to completely model a business rule, yet to find a way to depict it as simply as possible as discussed in systemic design concern 2. On the other hand, the relationship between the levels can be subtle and difficult for the novice analyst and user to comprehend at first, because these levels are often elaborations of each other.

SUMMARY:

This paper has highlighted three systemic design concerns in the modelling of business rules and explained how these challenges have been explored in an action research study. This study culminated in the development of a Business Rules Diagram that benefitted from examining and dealing with these issues.

Three systemic design concerns were discussed in this paper. Firstly, the use of Wand and Weber's ontological framework as a tool to manage the conversion process from informal to structured is novel in the context of designing **new** diagrams though there are instances of evaluating existing diagrams and notations with their technique (Wand and Weber 1995). Secondly, the process of identifying the constructs was discussed. Of significance here is that quantitative tools were used to manage the complexity of the diagram under development. Thirdly, the idea of separating business rules into repositories which link together through levels of abstraction was discussed and thus demonstrated the concern of how a diagram under development might be considered in the context of a fuller methodology.

REFERENCES:

Avison, D. E., and Fitzgerald, G. (1995). *Information Systems Development Methodologies*. (2nd ed.) Oxford: Blackwell Scientific Publications.

Bunge, M. (1977). *Treatise on Basic Philosophy. Vol. 3, Ontology I: The Furniture of the World*. Vol. 3. Dordrecht, Holland: D. Reidel.

Bunge, M. (1979). *Treatise on Basic Philosophy. Vol. 4, Ontology II: A World of Systems*. Vol. 4. Dordrecht, Holland: D. Reidel.

Byrd, T. A., Cossick, K. l., and Zmud, R. W. (1992). 'A synthesis of research on requirements analysis and knowledge acquisition techniques.' *MIS Quarterly* 16, pp117-138.

Checkland, P. B. (1981). *Systems Thinking, Systems Practice*. Chichester, England: John Wiley.

Darke, P., and Shanks, G. (1997). 'User viewpoint modelling: understanding and representing user viewpoints during requirements definition.' *Information Systems Journal* 7, pp213-239.

Davis, G. B. (1982). 'Strategies for information requirements determination.' *IBM Systems Journal* 21, pp4-30.

Embley, D. (1992). *Object-oriented Systems Analysis: A Model-driven Approach*. Englewood Cliffs, New Jersey: Prentice-Hall.

Eden, C. (1988). 'Cognitive Mapping'. *European Journal of Operational Research, 36*, pp1-13.

Fitzgerald, G. (1991). 'Validating new information systems techniques: a retrospective analysis'. In H. E. Nissen, Klein, H. K. and Hirschheim, R. (Ed.), *Information Systems Research: Contemporary Approaches and Emergent Traditions* (pp657-672). Amsterdam: North-Holland.

Flynn, D. J., and Davarpanah Jazi, M. (1998). 'Constructing user requirements: a social process for a social context.' *Information Systems Journal* 8: pp53-83.

Greenspan, S., Mylopoulos, J., and Borgida, A., (1994). 'On Formal Requirements Modeling Languages: RML Revisited. *Proceedings of the 16th International Conference on Software Engineering.* (pp135-147) Los Alamitos. IEEE Press.

Gutierrez, O. (1987). 'Some aspects of information requirements analysis using a requirements grid technique'. *Information Analysis: Selected Readings* (pp347-62) R. D. Galliers (ed). Reading: Addison-Wesley.

Herbst, H. (1996). 'Business Rules in Systems Analysis: A Meta-model and Repository System'. *Information Systems, 21*(2), pp147-66.

Ilvari, J., Hirschheim, R. and Klein, H. (1998). 'A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies, *Information Systems Research*, 9(2), pp164-193.
Jones, B. (1991). Letter to editor. *Database Programming and Design*, Nov. p9.
Leifer, R., Lee, S. and Durgee, J. (1994). 'Deep structures: real information requirements determination'. *Information and Management, 27*, pp275-85.
Kent, S. and Gil, J. (1998). 'Visualising action contracts in object-oriented modelling'. *IEE Software*, Vol 145 (2-3), pp70-78.
Kent, S. (1997). 'Constraint diagrams: Visualising invariants in object-oriented models'. Proceedings of OOPSLA97. ACM Press.
Martin, J., and Odell, J. (1995). *Object-oriented Methods: A Foundation*. Englewood Cliffs, New Jersey: Prentice-Hall.
McCabe, T. J. (1976). 'A Complexity Measure'. *IEEE Transactions in Software Engineering, 2*(4), pp308-20.
McDermid, D. C. (2007). 'The Business Rules Diagram', *The Icfai Journal of Computer Sciences*, Vol. 1, No. 2, pp 47-64.
Miller, J. C. (1964). 'Conceptual Models for Determining Information Requirements.' *Proceedings of the Spring Joint Computer Conference* 25, pp609-20.
Mittermeir, R., T., Hsia, P., and Yeh, R. T. (1987). 'Alternatives to Overcome the Communication Problem of Formal Requirements Analysis'. *Chap. 9, pages 153-165 of*: Galliers, R. D. (ed), *Information Analysis: Selected Readings.* Reading: Addison-Wesley.
Mumford, E. (1985). 'Defining systems requirements to meet business needs: a case study example'. *Computer Journal*(28), pp97-104.
Naumann, J. D. and Jenkins, A. M. (1982). 'Prototyping: The New Paradigm in Systems Development.' *MIS Quarterly* 6(3): pp29-44.
Pohl, K. (1994). 'The Three Dimensions of Requirements Engineering: A Framework and Its Applications.' *Information Systems* 19, pp243-258.
Rockart, J. F. (1979). 'Chief Executives define their own data needs.' *Harvard Business Review* 57, pp81-91.
Shlaer, S., and Mellor, S. J. (1992). *Object Lifecycles: Modeling the World in States*. Englewood Cliffs, New Jersey: Yourdon Press.
Rossi, M. (1997). 'Evolution of OO Methods: the Unified Case'. *Australian Journal of Information Systems, 4*(2), pp55-62.
Rossi, M., and Brinkkemper, J. (1995). 'Metrics in Methods Engineering'. *Proceedings of the 7th Advanced Information Systems Engineering Conference*, Oulu, Finland, Springer-Verlag, Berlin.
Valusek, J. R., and Fryback, D. G. (1987). 'Information requirements determination: obstacles within, among and between participants'. *Information Analysis: Selected Readings* (pp139-51). R. D. Galliers (ed). Reading: Addison-Wesley.
Vidgen, R. (1997). 'Stakeholders, soft systems and technology: separation and mediation in the analysis of information systems requirements.' *Information Systems Journal* 7, pp21-46.
Vitalari, N. P. (1992). 'Structuring the Requirements Analysis Process for Information systems: A Proposition Viewpoint'. *Pages 163-79 of:* Cotterman, W. W. and Senn, J. A. (eds) *Challenges and Strategies for Research in Systems Development.* Chichester, England, John Wiley and Son.
Wand, Y., and Weber, R. (1993). 'On the Ontological Expressiveness of Information Systems Analysis and Design Grammars'. *Journal of Information Systems, 3*, pp217-37.
Wand, Y., and Weber, R. (1995). 'On the deep structure of information systems.' *Information Systems Journal* 5, pp203-223.
Watson, A. H., and McCabe, T. J. (1996). *Structured Testing: A Testing Methodology using the Cyclomatic Complexity Metric*. Gaithersburg, USA: National Institute of Standards and Technology.
Zmud, R. W., Anthony, W. P., and Stair, R. (1992). 'The use of mental imagery as a requirements technique'. *Pages 337-50 of:* Cotterman, W. W. and Senn, J. A. (eds) *Challenges and Strategies for Research in Systems Development.* Chichester, England, John Wiley and Son.

COPYRIGHT