

**DEPARTMENT OF COMPUTER SCIENCE**

**Technical Report**

**Towards A Framework for Selecting Notations  
For Modelling Requirements**

**Sara Jones, Carol Britton and Wing Lam**

**Report No: 307**

**January 1997**

# Towards A Framework for Selecting Notations for Modelling Requirements

Sara Jones, Carol Britton and Wing Lam  
Department of Computer Science  
University of Hertfordshire  
College Lane, Hatfield, Herts. UK  
AL10 9AB  
Tel: 01707 284370 / 284354 / 284337  
Email: S.Jones,C.Britton,W.Lam@herts.ac.uk

## Abstract

An acknowledged problem in software development is that a poor choice of notations may all too often have a detrimental effect on development activities, but full scale of evaluation of every notation in the context of every development project would be neither feasible nor sensible. This is particularly the case in the area of requirements modelling and specification, where the number and variety of available techniques makes selection of the most appropriate notation difficult and prone to error. There is currently little in the way of theoretical guidelines or empirical case studies to support the requirements engineer in making a sound choice of modelling notation in any particular situation. This paper presents and evaluates a principled approach to the selection of modelling notations, with the ultimate aim of assisting practising software developers in making quick but useful evaluations of available modelling notations in the context of particular development situations.

**Keywords:** modelling, notations, selection

**Category:** research

## 1 Introduction - the importance of using appropriate notations

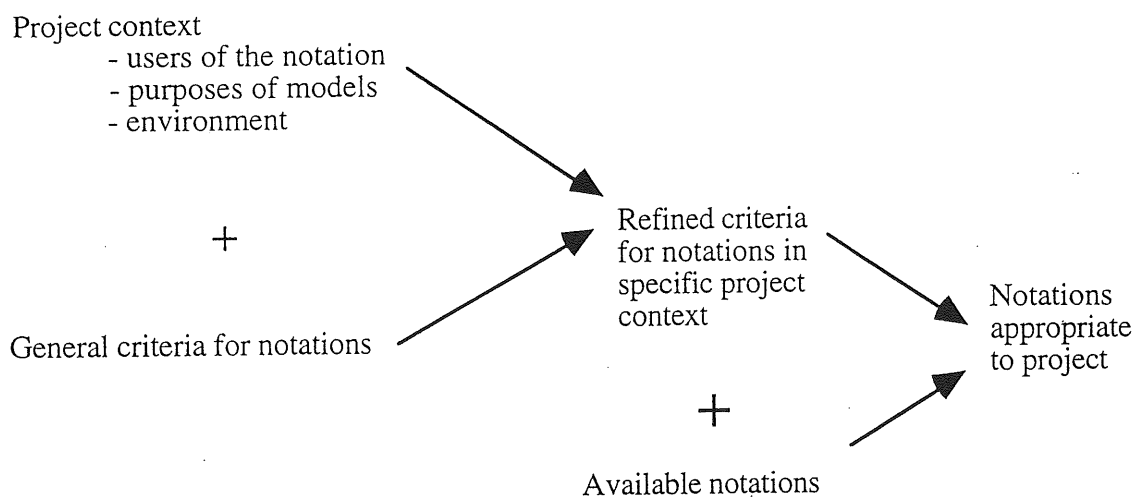
Although the relationship between the use of different notations and the effectiveness and efficiency of the system development process is not fully understood, the impact of the choice of representation on successful performance of many system development activities has been recognised for some time (Green 89 and 91, McCluskey et al 95, Modugno et al 94). However, little is currently known about what notations are likely to be most suitable for use in which contexts. The choice of notations for particular projects often reflects the experience or preferences of the development team more than an objective consideration of possible alternatives (McCluskey et al 95).

The problem of assessing and selecting notations, methods and tools is one that has attracted interest from both academia and industry. Projects such as RESCUED from Queen Mary and Westfield College (O'Neill et al 97) and the Evaluation Framework for Representations in Requirements Engineering from City University (Sutcliffe et al 97) are currently investigating the choice of representations at various stages of the software development process. The STARTS (DTI 87) Guide includes suggested criteria for effective modelling notations, while the DESMET project addresses the question of how to determine objectively the effects and effectiveness of methods and tools in general. Macaulay (Macaulay 96) has identified the question of 'What representation format should be used?' as a significant one for practising software developers working on the elicitation and specification of system requirements. Other authors have set out general guidelines on how to choose appropriate requirements representations and methods for particular situations. For example, Christel and Kang's report for the SEI identified a number of factors relating to the scope of a project, the need for understanding by various parties, and the volatility of requirements, which they suggested should influence the developer's choice of requirements techniques and notations (Christel and Kang 92). More recently, Sommerville and Sawyer have listed some generic guidelines for choosing models and methods (Sommerville and Sawyer 97) and the RESPECT project has made some general recommendations as to the stages of system development for which certain techniques and representations are most appropriate (Maguire 97). However with each of these authors taking a slightly different perspective on the problem, it is difficult for the practising software developer to know on what basis the choice of modelling notation for a particular project should be made.

Although one obvious approach to the problem would be to conduct a series of formal experiments, there are factors that mitigate against this. Experimentation is resource-intensive, particularly given the vast range of different situations in which requirements modelling takes place. Moreover, many notations for modelling requirements are relatively immature and are still evolving; results from current experiments with particular notations may only be useful in the very short term.

We argue in this paper that a more appropriate approach to the problem is to base the selection of notations on existing knowledge and experience. We therefore propose a framework within which this knowledge can be structured, developed and exploited. The framework suggested here aims to help practising software developers make quick, but useful evaluations of available modelling notations in the context of particular development situations. The framework is presented in the form of a simple questionnaire (for further details see section 5). It does not propose rigorous guidelines, but should be treated rather as a tool to promote discussion and evaluation. Figure 1, below, gives an overview of the proposed selection process, showing how general criteria for modelling notations (identified from the industrial

and academic literature) are considered in relation to a profile of the project. This profile is obtained from a detailed checklist of features of the project context provided in the questionnaire. Comparison of the general criteria for notations with the profile of the project context provides the basis for a refined list of criteria for notations which will be effective in the situation under consideration. Modelling notations available to the developer are then considered in relation to the refined list of criteria and from this the final selection of appropriate modelling notations is made.



**Figure 1: Overview of the selection process for modelling notations**

The rest of this paper describes the selection framework in more detail and summarises ways in which it has been evaluated to date. Section 2 of the paper provides working definitions for the most frequently used (and misunderstood) terms in this context; in section 3 we discuss the features of the development context which influence the choice of modelling notation and in section 4 we review the relevant literature and list the criteria for modelling notations which appear to be agreed by the majority of authors. In section 5 we briefly explain how the selection framework came to be developed, and in section 6 we outline ways in which it has been validated and comment on results to date. The paper concludes with a discussion of the points raised and indicates the contribution that this work makes to the problem of selecting effective modelling notations.

## 2 Definitions of terms

We have already said above that the principal role of the selection framework is as a tool for discussion and evaluation. No useful discussion can take place unless all participants have a shared understanding of the meanings of the main terms, and words such as 'model' are

particularly prone to a variety of interpretations. Clear definitions of terms help to eliminate vagueness and to reveal and resolve ambiguity; even if the definitions used are not agreed by all readers, they nevertheless help to clarify what the authors are trying to say. We therefore suggest the following as working definitions in the context of this paper:

**Notation:** a system of signs, symbols or characters used to represent concepts that are relevant to the development of a system; examples include the data flow diagramming technique, Z, CCS, natural language, storyboards.

**Representation:** an expression of a concept or concepts relevant to the development of a system which is created using notations such as those listed above; examples of models include particular data flow diagrams, Z specifications, and statements of requirements written in natural language.

**Model:** the information content of a representation.

### **3 Some Important Aspects of the Development Context**

A problem with the majority of current approaches to system development is that they place little, if any, emphasis on the context in which a system is developed. There is, however, increasing recognition among the system development community that characterising projects in order to determine appropriate methods may improve the effectiveness and quality of the system development process. Various authors have characterised particular development situations in different ways. Potts, for example, discusses the needs of software developers who build off-the-shelf application software (Potts 95), Sommerville and Sawyer characterise requirements engineering for critical systems (Sommerville and Sawyer 97), McCluskey et al consider various aspects of the development situation which affected their choice of notation for use in a project developing a prototype decision support system for air traffic controllers (McCluskey et al 95), and Jones and Britton have attempted to identify the distinguishing features of multimedia development projects which might have an impact on the choice of notations for use in this area (Jones and Britton 96, Britton et al 96, Britton and Jones 97a).

The complexity of a software development project means that there are many different factors which could potentially affect a developer's choice of modelling notations. The authors identified above have each described situations of interest in different ways. To bring together these different views, we suggest that the development context for a particular project may be characterised using something analogous to the notion of context proposed by Kellogg in relation to the use of interactive systems.

Kellogg argued that the usability of a computer system could only be properly assessed in relation to the context in which it was intended to be used (Kellogg 90). In this case, the notion of context consisted of elements relating to:

- the intended users of the system
- the tasks for which the system was to be used, and
- the environment (social, organisational and physical) in which the system was to be placed.

We suggest that the important features of a system development context which might influence a developer's choice of modelling notation could, by analogy, be considered in terms of factors relating to:

- the intended users of the representations (stakeholders)
- the purposes for which the representations are intended to be used, and
- the environment in which the representations are to be produced and employed, including specific features of the system under development.

Each of these factors is discussed in more detail below.

### **3.1 Stakeholders**

We consider a stakeholder in the context of selecting modelling notations as anyone who either uses a representation produced using the notation or who is affected by it. This will include, among others, the developers who create the models, the customers who read and discuss them and the programmers who use them as a basis for implementation. The characteristics of stakeholders which have a bearing on the choice of modelling notations are shown below. These should be considered for each of the different categories of stakeholder, in order to build up a complete picture.

- 1 Extent of stakeholder involvement in the requirements process
- 2 Stakeholder's understanding of software systems
- 3 Stakeholder's experience of requirements modelling notations
- 4 Stakeholder's experience of relevant maths
- 5 Feasibility of training stakeholders to understand new notations
- 6 Stakeholder's understanding of the problem domain

### 3.2 Intended Purpose of Models

During the development of a system, models may be produced for a variety of different purposes, such as those shown in the list below. This list has been compiled from a review of the literature on Requirements Engineering and a survey of developers of multimedia systems (Britton et al 96). Where models are to be used for more than one purpose, it is highly likely that these will demand different properties from notations; in this case it will be necessary for the developer to put the purposes identified into order of priority.

- 1 As a vehicle for communication and negotiation between the developer and other stakeholders
- 2 As a vehicle for communication between members of the development team
- 3 As a basis for informal agreements between the developer and other stakeholders
- 4 As part of a tender
- 5 As a basis for demonstrating the system to senior management
- 6 As a basis for a legal contract between client and developer
- 7 As a basis for detailed design of the system
- 8 As a basis for selection of an off-the-shelf package
- 9 As a basis for system implementation
- 10 As documentation for future maintenance and modification of the system
- 11 To carry out formal checks of correctness and consistency
- 12 To validate system requirements

### 3.3 Environment

The environment in which representations are to be produced and employed is influenced by a wide variety of factors, including the cultural and social aspects of the developing and client organisations, the nature of the particular development project, and the type of system under development. The features listed below are those which we consider are most likely to affect the choice of notations for modelling requirements in the early stages of development.

- 1 Stability of the requirements
- 2 Likelihood of conflicts / inconsistencies in requirements
- 3 Degree to which the project is constrained by time
- 4 Degree to which the project is constrained by costs
- 5 Tools that are available in the organisation to support the requirements process
- 6 Existing models which should be incorporated into the requirements process
- 7 Programming paradigm or language to be used for implementation of the system

The following features relate directly to the system under development:

- 8 Extent to which the system is safety-critical
- 9 Extent to which the system is security-critical
- 10 Complexity of processing in the system
- 11 Complexity of information presentation

Consideration of the features in the three categories above (stakeholders, purpose of models and environment) will not only lead to a better understanding of the system development context, but will also allow the developer to make a more soundly-based choice of modelling notation. As Green (89) puts it: "A notation is never absolutely good, but good only in relation to certain tasks". For example, if the requirements specification is to be used primarily as a vehicle for communication, ease of understanding of the notation will be a priority. On the other hand, if the system under development is safety or security-critical, then the ability to support formal reasoning may be more important. We would enlarge on Green's view and say that a notation is never absolutely good, but good only in relation to certain situations or contexts, where the notion of situation is broader than that of task and includes stakeholder, environment and type of system as well.

#### **4 General Criteria for Use in Selecting Notations**

Work on criteria for modelling notations has been carried out by authors from both the academic and industrial communities. Farbey (93) includes both criteria for notations, such as readability, modifiability and lack of ambiguity, and criteria relating to the notation in use, such as the ease of production of a well-presented specification, the cost in time to produce the model, and the amount of support available. Green (89) suggests that a notation should be able to support what he terms 'opportunistic planning', where high-and low-level decisions may be mingled, work may frequently be re-evaluated and modified and commitment to different decisions may be strong or weak. Although Green is writing about notations for programming, his point is equally relevant to the study of notations which are used earlier in the development process. In an article on hypermedia design (Garzotto 95), Garzotto evaluates notations in terms of what can be described: a useful notation should be able to model information content and presentation, system structure, and interaction with the user. Davis (88 and 93) suggests a list of criteria pertaining to the effectiveness of representations and the choice of notations. Davis' list includes criteria relating directly to modelling notations, such as that the notation should permit annotation and traceability, facilitate modification, and provide



a basis for automated checking and generation of prototypes and system tests. Finally, Green (80) makes the point that programming languages with a large number of features are more difficult to learn and understand than languages with fewer features. If we apply this to modelling notations, we can deduce that understandability is supported by using a notation which has a relatively small number of different symbols. This means that, to achieve ease of understanding, a relevant criterion of a modelling notation is to have relatively few symbols.

Among publications from authors in industry, criteria for modelling notations in the STARTS guide (DTI 87) incorporate qualities such as rigour, suitability for agreement with the end-user and assistance with structuring the requirements. Rigour comprises four separate features: how precisely the syntax of the notation is defined, the extent to which it is underpinned by maths and logic, whether the meaning of individual symbols is defined, and the extent to which the notation supports consistency checking of the requirements themselves. Suitability for end-user agreement refers to ease of understanding of the notation by an untrained user, and assistance with structuring the requirements assesses the extent to which the notation supports hierarchical decomposition and separation of concerns in the model. The STARTS guide also regards the range of requirements covered by the notation as important, including functional, performance, interface, system development and process requirements. Admiral Training's (95) guide to interactive multimedia development is similar to the STARTS guide in placing emphasis on what the notation should be able to model. Effective notations, according to Admiral, should be able to describe the current situation, the target audience, the actual and required level of user performance, the overall aim of the system, the environment, possible constraints and details of specific functions.

From the above discussion and our own experience, we can identify a set of agreed criteria for modelling notations in general as follows:

- 1 Coverage (the ability to model a variety of features, including data, processes, sequence, concurrency, time, the user, performance and different media)
- 2 Ease of producing a model
- 3 Ease of understanding a model
- 4 Degree to which the notation encourages meaningful debate
- 5 Level of structure inherent in the notation
- 6 Compatibility with other modelling notations
- 7 Modifiability of representations built using the notation
- 8 Degree of precision of the notation
- 9 Degree to which notation is based on maths and/or logic
- 10 Possibility of annotation and traceability in representations produced

- 11 Possibility of generating prototypes/animations
- 12 Level of available tool support

## **5 Development and Application of the Framework**

### **5.1 Background**

The need for some kind of guidelines or discussion tool to support the selection of representations for modelling requirements was identified originally from a survey of multimedia developers that was carried out under a 12-month EPSRC ROPA project (Britton et al 96). The survey found that relatively little time and effort are given to modelling requirements in multimedia systems and that by far the most widely used technique in this context is the storyboard. Many of the modelling notations currently used in modelling software systems are less appropriate for multimedia systems, but developers do not have the time to carry out a full investigation of possible alternatives. It was felt that a framework which offered practical support in assessing modelling representations for particular situations would be useful, not only for multimedia developers, but for anyone faced with the choice of modelling representations.

The approach taken in developing our framework was to carry out a literature search in order to produce a generally agreed set of criteria for effective modelling notations. This is summarised in section 4 above. Factors contributing to an overall profile of the project context were determined both from a further literature search of accounts of particular development projects and from conversations with practitioners (see section 3, above). At this point the first draft of the framework was produced and applied to an in-house project.

### **5.2 The MAISIE project**

The aim of the MAISIE project was to develop a multimedia system to discourage young children from starting to smoke. The system had three objectives: to educate children about the dangers of smoking, to promote a negative attitude in children towards smoking and to teach social skills needed to resist pressures to smoke. In the early stages of the project the developers encountered the same sorts of problems with choice of modelling representations as were reported in the survey of multimedia developers (Britton et al 96). Because of time pressures, the selection framework could only be applied to the MAISIE project relatively late in the development process and it is hardly surprising that the notations suggested (retrospectively) by the framework were the same as those actually used. However, using the framework did highlight certain features of the project which increased the developers' overall

understanding and provided a measure of confidence in the requirements process for MAISIE. The principal characteristics of the MAISIE project, as identified by applying the selection framework, are shown below.

Stakeholders:

- there was a wide variety of stakeholders, both among users of the eventual system (children, teachers and school nurses) and among the development team (programmers, project manager, graphics, video and audio experts);
- apart from the programmers and the requirements engineers, the stakeholders had virtually no technical knowledge of software systems;
- members of the development team had very little knowledge of the problem domain (smoking and health education);
- because of practical constraints, it was not possible for all stakeholders to have a high degree of involvement in the requirements process.

Purposes for which representations were to be used in the MAISIE project were:

- as a vehicle for communication and negotiation between the developer and other stakeholders;
- as a vehicle for communication between members of the development team;
- as a basis for detailed design of the system;
- to validate system requirements.

It was decided that, of the purposes identified, the most important in the MAISIE project were communication and negotiation between developer and other stakeholders and the validation of system requirements.

Environment:

- there was a likelihood of conflicts in requirements between the various user groups;
- presentation of information was of utmost importance and was likely to be complex;
- MAISIE was to be implemented using a multimedia authoring tool.

### **5.3 Refining the General Criteria in relation to the MAISIE project profile**

In order to be useful in a practical way, the general criteria for modelling notations, shown in section 4, needed to be refined in relation to the MAISIE project context. It was evident from

the MAISIE project profile that ease of communication was a major consideration, since most of the stakeholders had very little knowledge of software systems and members of the development team were unfamiliar with the problem domain. At the same time the complexity of information presentation and its importance in the MAISIE system called for a notation which could provide clear models of the system structure. Discussions among the development team about what was required in modelling notations for MAISIE led to the following wish-list of high priority criteria:

- 1 Ease of understanding a model
- 2 Degree to which the notation encourages meaningful debate
- 3 Level of structure inherent in the notation
- 4 Possibility of annotation and traceability in models produced
- 5 Possibility of generating prototypes/animations
- 6 Ability to model the integration of different media

Thus the refined list of criteria consisted of five from the list of general criteria, and one (the ability to model the integration of different media) which was added to reflect the specific needs of multimedia development. The remaining criteria in the general list were deemed to be less relevant in the MAISIE development context.

#### **5.4 Further Applications of the Framework**

Table 1, below, shows the criteria considered as important in the MAISIE project and the modelling notations chosen. For comparison, we also show two further development projects which were used as case studies for the framework. These were FAROAS, a prototype decision support system for air traffic controllers (McCluskey et al 95), and SPIRE, an information database to support disabled students at the University of Hertfordshire (Bearne et al 96).

In the table criteria from the general list that were not considered as high priority in the particular development context are shown as struck out, and new criteria identified as a result of considering the project profile are shown in italics.

PROJECT	FAROAS	SPIRE	MAISIE
CRITERIA OF IMPORTANCE	Coverage: data processes user interface	Coverage: data user actions user interface different media	Coverage
	Ease of producing models	Ease of producing models	Ease of producing models
	Ease of understanding models	Ease of understanding models	Ease of understanding models
	Degree to which the notation encourages meaningful debate	Degree to which the notation encourages meaningful debate	Degree to which the notation encourages meaningful debate
	Level of structure inherent in the notation	Level of structure inherent in the notation	Level of structure inherent in the notation
	Compatibility with other modelling notations	Compatibility with other modelling notations	Compatibility with other modelling notations
	Modifiability of models	Modifiability of models	Modifiability of models
	Degree of precision	Degree of precision	Degree of precision
	Foundation in maths/logic	Foundation in maths/logic	Foundation in maths/logic
	Possibility of annotation and traceability in models produced	Possibility of annotation and traceability in models produced	Possibility of annotation and traceability in models produced
Ease of prototyping/animation	Ease of prototyping/animation	Ease of prototyping/animation	
Level of available tool support	Level of available tool support	Level of available tool support	
<i>Ease of formal reasoning</i>		<i>Ability to model the integration of different media</i>	
<i>Ease of translation between forms</i>			
NOTATIONS CHOSEN	Many-sorted first-order logic	E-R diagrams State-transition diagrams	Storyboards State-transition diagrams

**Table 1: Results of using the selection framework in three case studies.**

Note that in all three of these development projects prototyping was used to validate requirements. In this paper, however, we are primarily interested in the choice of notations for models that precede prototyping.

## 6 Validation of the Framework

As well as applying the framework retrospectively to projects carried out by members of the research team, we have presented these ideas to both academics and practitioners in interviews and at two large meetings. The questionnaire has also been completed in relation to one other project by two researchers who are not part of our team. This section presents data collected in a workshop held at STEP97, a recent conference on software engineering (Budgen et al 97) . General points for discussion regarding additions or modifications to the framework and ideas for putting it into practice which have arisen out of this and other presentations of our ideas are raised in the final section.

The aim of the workshop at STEP97 was to promote discussion of any factors which may significantly influence a developer's choice of notations for requirements representation, and to evaluate the usefulness of the framework produced by our project. The workshop was attended by 10 practitioners and 12 academics and lasted for two hours. The framework was presented at the beginning of the workshop, and there was then a general discussion regarding the way in which notations are currently chosen, and the extent to which our framework might be used to assist in making such choices. At the end of the discussion, workshop participants were asked to fill out a rating sheet in which they rated each element of our framework, for example as follows:

### 1.1.1 *Extent of stakeholder involvement in the requirements process*

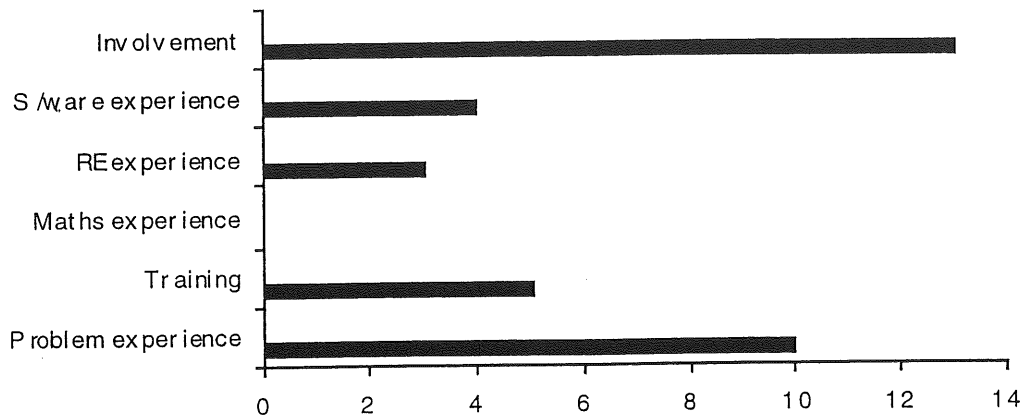
very important	quite important	not very important	not at all important
----------------	-----------------	--------------------	----------------------

### 1.1.2 *Stakeholder's understanding of software systems*

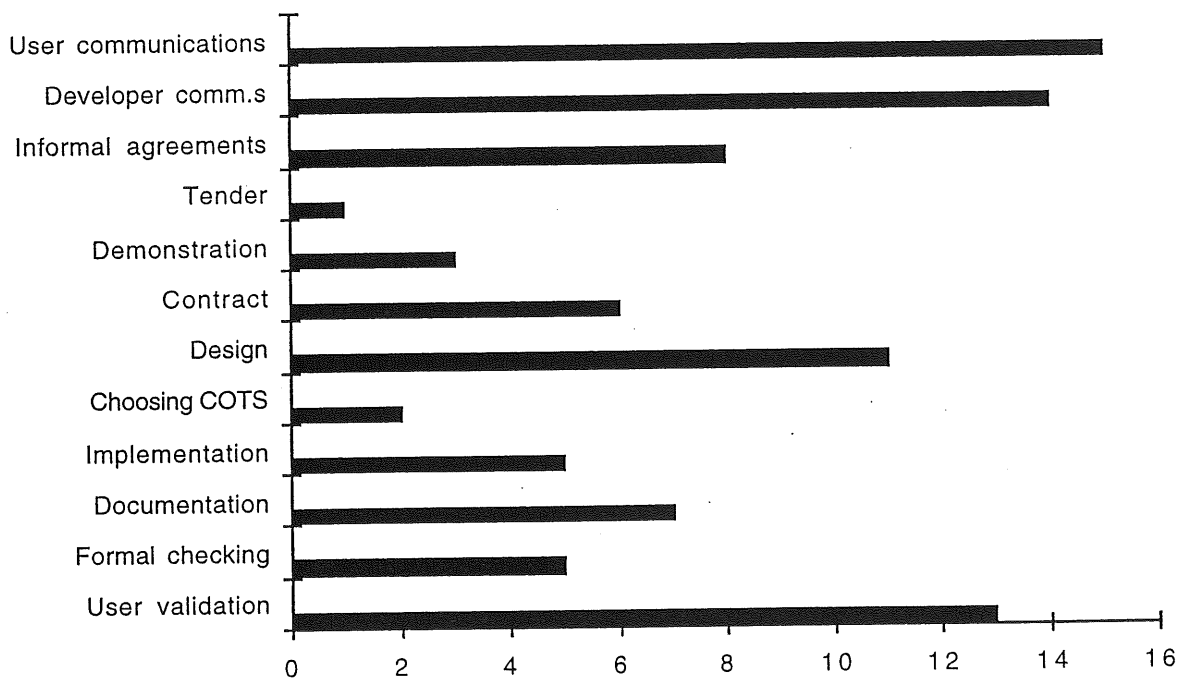
very important	quite important	not very important	not at all important
----------------	-----------------	--------------------	----------------------

**Figure 2: Extract from the rating sheet**

Eighteen of the participants completed the rating sheet. In the figures below, we show the number of participants who thought that each element of the framework was 'very important'. The complete set of data is shown in an appendix.

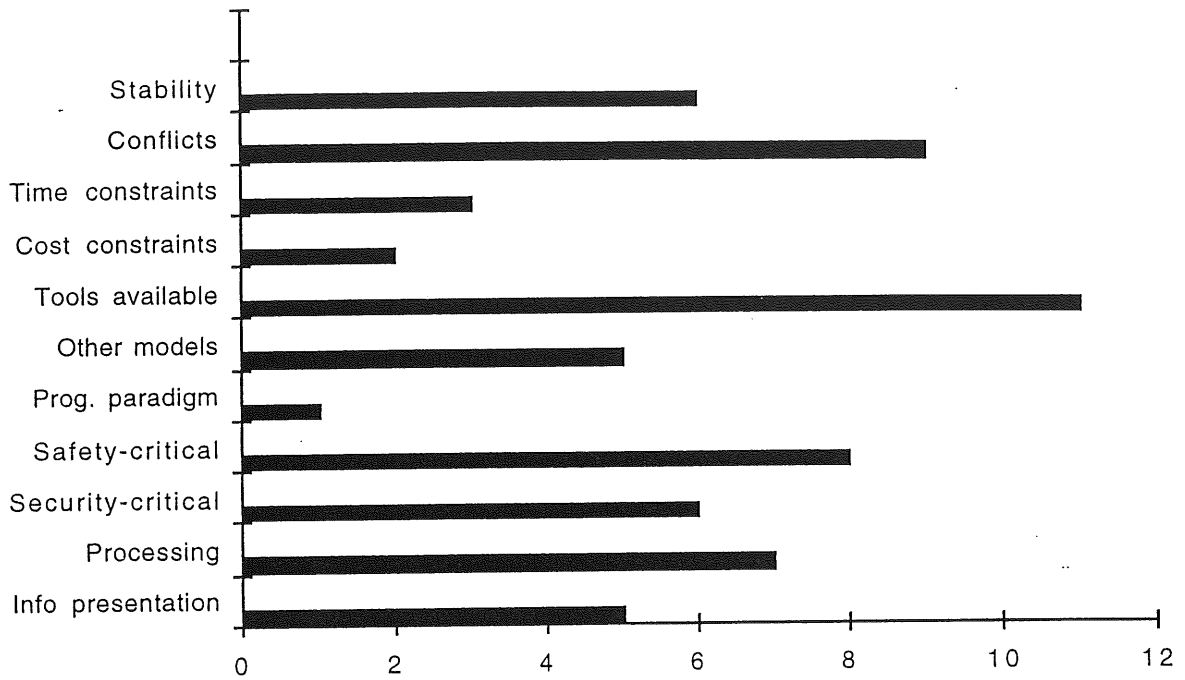


**Figure 3: Numbers of people rating characteristics of stakeholders as 'very important'** (labels refer to elements listed in section 3.1\*)

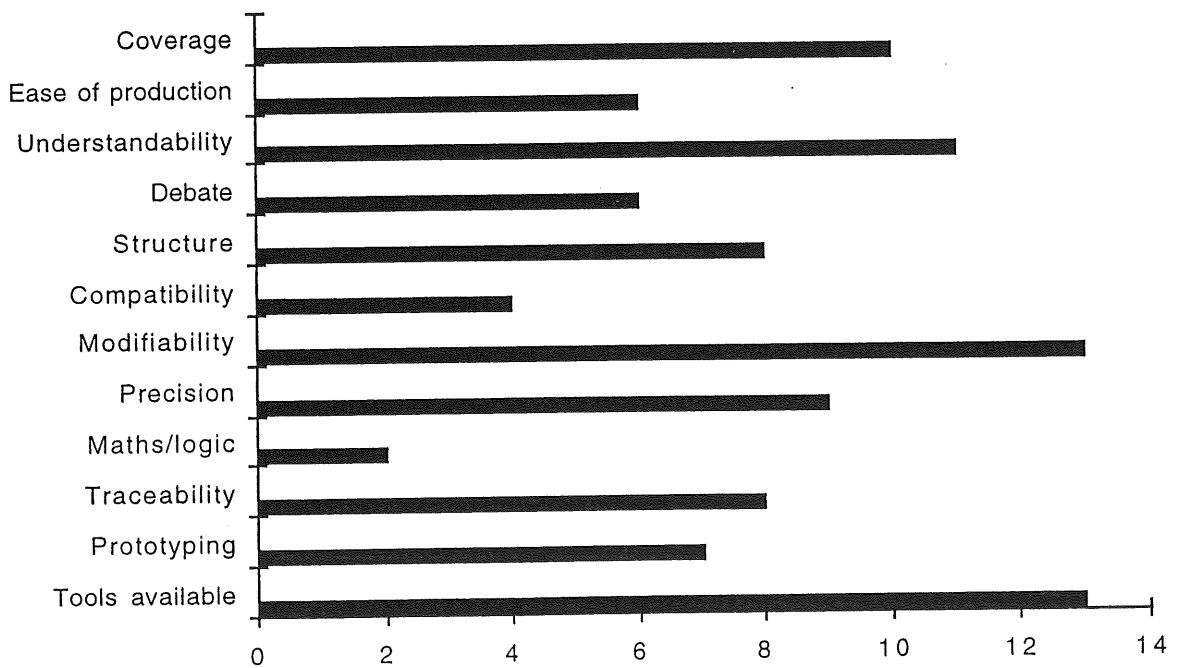


**Figure 4: Numbers of people rating purposes of models as 'very important'** (labels refer to elements listed in section 3.2)

\* In discussions after the rating sheet had been completed, it became clear that the majority of participants in the workshop had understood the term 'stakeholders' to refer simply to the clients and users for the system, rather than including the developers as intended in our definition.



**Figure 5: Numbers of people rating elements of the environment as 'very important'** (labels refer to elements listed in section 3.3)



**Figure 6: Numbers of people rating criteria as 'very important'** (labels refer to criteria listed in section 4)



Considering the data presented above, it appears that perhaps the most important factors in deciding what notations to use centre around the following issues:

- degree of user or client involvement in the requirements process, and in particular in requirements validation (see figures 3 and 4);
- the need for communication between developers (see figure 4);
- the need for support in system design (see figure 4);
- the modifiability of models produced (see figure 6); and
- the availability of tool support (see figures 5 and 6)

Issues relating to these factors were rated as 'very important' by two thirds or more of those who took part in the workshop.

Of the factors listed for consideration as part of the framework, very few were rated by participants at the workshop as 'not very important' or 'not at all important' (see appendix). In view of this, we feel a reasonable degree of confidence in the content of the framework as a vehicle for supporting decisions about modelling notations, although we still have reservations about how it can most effectively be applied; some areas of further research are identified in the discussion in the following section.

## **7 Discussion**

The use of effective modelling notations is crucial in the early stages of system development: a fact which is reflected in the range and variety of available notations. Developers, in need of a modelling notation, are faced with an ever increasing choice, too varied to try out in practice, all valiantly attempting to keep up with the accelerating development of technology and communication methods. However there is currently little theoretical guidance on how to choose an appropriate notation. How is the developer to select the combination of notations that will best suit the purpose at hand and have confidence in the choice?

In this paper we have presented the basis for a principled approach to the selection of modelling notations, with the ultimate aim of assisting practising software developers in making quick but useful evaluations of available modelling notations in the context of particular development situations. The framework is not a recipe for choosing modelling notations, but rather a scaffolding to support the developer's growing understanding of the project context. Our

approach encourages developers to think in depth about the system and its environment at an early stage in the development process. In addition to providing a sound basis for the selection of modelling notations, the framework promotes a process of reflection about the project context which will, in itself, enhance the quality of the development.

At present this work is at a formative stage; findings from validations to date indicate that the ideas in the framework are soundly-based (although a useful suggestion from the workshop at STEP97 was the addition of company policy and culture as an important consideration). Discussions at STEP97 indicate that more thought is needed as to how the framework can be put into practice most effectively. There are four areas to consider here:

- how to prioritise or weight the different elements;
- whether the framework would be most usefully applied to individual projects, project types or as part of a strategic organisational review;
- whether the most effective format for the framework is a paper questionnaire or a computer-based tool;
- how evaluation of modelling notations, which must currently be made on a subjective basis, can be put on a firmer footing through, for example, the use of appropriate metrics;
- how experience of requirements engineering practitioners in selecting notations can best be accumulated for future benefit.

Work on some of these topics has already begun (Myers et al 97, Britton and Jones 97b). Future research on the framework will aim to address the issues above in more depth.

### **Acknowledgements**

The authors are very grateful to all those who have taken part in evaluating these ideas, particularly Robert Clark and Ana Moreira.

## References

- Admiral Training Ltd. "Multimedia Design - A Newcomer's Guide" The Department of Employment, Sheffield, 1995
- Bearne, M., Jones S., Hewitt, J. and Sapsford-Francis, J. "Providing public access to information in complex and weakly structured domains: a three-layered model for hypermedia information systems", *The New Review of Hypermedia and Multimedia*, Volume 2, 1996.
- Britton, C., Jones, S., Myers, M. and Sharif, M. "A survey of current practice in multimedia system development" Technical Report No. 258, Faculty of Information Sciences, University of Hertfordshire, 1996, to appear in the *Journal of Information and Software Technology*.
- Britton, C. and Jones, S. (97a) "Using project characterisation in selecting notations for modelling requirements", Technical Report No. 302, Department of Computer Science, University of Hertfordshire, 1997. Submitted to the IFIP International Workshop on Situated Method Engineering, November 1997
- Britton, C. and Jones, S. (97b) "Which properties make a modelling notation easy for untrained users to understand?" in *Proceedings of the International Workshop on Representations in Interactive Software Development*, Queen Mary and Westfield College, July 1997
- Budgen, D., Hoffnagle, G. and Trienekens, J. (Eds.) *Proceedings of the Eighth International Workshop on Software Technology and Engineering Practice*, IEEE Computer Society 1997
- Christel, N. and Kang, K. "Issues in Requirements Elicitation", Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, 1992
- Davis, A. "A Comparison of techniques for the specification of external system behaviour" *Communications of the ACM*, Volume 31, Number 9, September 1988
- Davis, A.M. "Software Requirements. Objects, Functions and States" Prentice Hall International, 1993
- Department for Trade and Industry and National Computing Centre "The STARTS Guide", Second edition, volume 1, NCC Publications, 1987
- Farbey, B. "Software quality metrics: considerations about requirements and requirement specifications" in *Software Engineering: A European Perspective*, R. Thayer and A. McGetterick (Eds.), IEEE C.S.Press, 1993
- Garzotto, F., Mainetti, L., Paolini, P. "Hypermedia Design, Analysis and Evaluation Issues" *Communications of the ACM* Vol 38, No.8, August 1995
- Green, T.R.G. "Programming as a Cognitive Activity" in H.T. Smith and T.R.G. Green (Eds.) *Human Interaction with Computers*. Academic Press, 1980
- Green, T.R.G. "Cognitive Dimensions of Notations" in *People and Computers (HCI 89)* Sutcliffe and McCaulay (Eds.) C.U.P, 1989
- Green, T.R.G. (1991) "Describing Information Artifacts with Cognitive Dimensions and Structure Maps" in *People and Computers VI, Proceedings of the HCI'91 Conference*, Diaper, D. and Hammond, N. (Eds.), August 1991
- Jones, S. and Britton, C. "Early Elicitation and Definition of Requirements for an Interactive Multimedia Information System" *Proceedings of ICRE96, the Second International Conference on Requirements Engineering*, IEEE Computer Society Press, 1996

Kellog, W. "Qualitative Artifact Analysis", in *Human Computer Interaction: INTERACT90* Diaper, D., Gilmore D., Cockton, G. and Shackell, B. (Eds.) Elsevier Science Publishers, BV (North Holland) 1990.

Kitchenham, Barbara DESMET guidelines for Evaluation Method Selection : EMS / Barbara Kitchenham. - Version 2.0. NCC Services, 1994.

Macaulay, L.A. "Requirements for Requirements Engineering Techniques", Proceedings of ICRE96, the Second International Conference on Requirements Engineering, IEEE Computer Society Press, 1996

Maguire, M. "RESPECT User Requirements Framework Handbook" Version 2.2, HUSAT Research Institute, April 1997.

McCluskey, T.L., Porteous, J.M., Naik, Y, Taylor, C.N. and Jones, S. (1995) *A Requirements Capture Method and its use in an Air traffic Control Application*. Software practice and Experience, Vol 25 (1)

Modugno, F., Green T.R.G. and Myers B.A. (1994) "Visual programming in a Visual Domain: A Case Study of Cognitive Dimensions" in *People and Computers IX, Proceedings of HCI'94*, Glasgow, August 1994

Myers, M. ,Britton, C., Jones, S.and Sharif, M. "An investigation into the measurement of modelling notations", in Proceedings of The Eighth European Software Control and Metrics Conference, Berlin, published by ESCOM, May 1997.

O'Neill, E., Johnson, P. and Johnson, H. "Representations in co-operative software development: an initial framework", Proceedings of the International Workshop on Representations in Software Development, 2-4 July 1997, Queen Mary and Westfield College, University of London

Potts, C. "Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software", in Proceedings of RE95, the Second IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, 1995.

Sommerville, I. and Sawyer, P. "Requirements engineering: a good practice guide", Wiley, 1997

Sutcliffe, A.G., Maiden N.A.M. and Bright, B. "An Evaluation Framework for Representations in Requirements Engineering", Proceedings of the International Workshop on Representations in Software Development, 2-4 July 1997, Queen Mary and Westfield College, University of London

## Appendix

The table below shows the numbers of people who rated each element of our framework as 'very important', 'quite important', 'not very important' and 'not at all important' respectively. The rating sheet was completed by 18 practitioners and academics as described in section 6.

	very important	quite important	not very important	not at all important
<b>Stakeholders:</b>				
Extent of stakeholder involvement in the requirements process	13	3	2	0
Stakeholder's understanding of software systems	4	2	11	1
Stakeholder's experience of requirements modelling techniques	3	6	8	1
Stakeholder's experience of relevant maths	0	4	11	3
Feasibility of training stakeholders to understand new techniques	5	6	7	1
Stakeholder's understanding of the problem domain	10	6	2	0
<b>Intended purpose of models:</b>				
As a vehicle for communication and negotiation between the developer and other stakeholders	15	1	2	0
As a vehicle for communication between members of the development team	14	2	2	0
As a basis for informal agreements between the developer and other stakeholders	8	7	3	0
As part of a tender	1	6	7	2
As a basis for demonstrating the system to senior management	3	7	7	1
As a basis for a legal contract between client and developer	6	7	3	1
As a basis for detailed design of the system	11	7	0	0
As a basis for selection of an off-the-shelf package	2	10	4	2
As a basis for system implementation	5	10	3	0
As documentation for future maintenance and modification of the system	7	10	1	0
To carry out formal checks of correctness and consistency	5	7	6	0
To validate system requirements	13	2	3	0
<b>Environment:</b>				
Stability of the requirements	6	9	3	0
Likelihood of conflicts / inconsistencies	9	8	0	1
Degree to which your project is constrained by time	3	6	6	3
Degree to which your project is constrained by costs	2	6	8	2
Tools that are available in your organisation to support the requirements process.	11	6	0	1
Existing models which should be incorporated into the requirements process.	5	8	4	1
Programming paradigm or language to be used for implementation of the system.	1	6	6	4
Extent to which the system is safety-critical	8	7	3	0
Extent to which the system is security-critical	6	8	4	0
Complexity of processing in the system	7	5	5	0
Complexity of information presentation	5	7	5	1

	very important	quite important	not very important	not at all important
<b>Criteria:</b>				
Coverage (ability to model data, processes, process sequence, process concurrency, time, the user, the user's actions, the user interface, performance, different media)	10	7	1	0
Ease of producing a model	6	10	2	0
Ease of understanding a model	11	6	0	1
Degree to which the notation encourages meaningful debate	6	7	3	2
Level of structure inherent in the notation	8	8	1	0
Compatibility with other modelling techniques	4	7	6	1
Modifiability of models built using the technique	13	3	1	0
Degree of precision of the technique	9	9	0	0
Degree to which technique is based on maths and/or logic	2	7	4	5
Extent to which the notation encourages annotation and traceability in the models produced	8	8	0	1
Possibility of generating prototypes/animations	7	7	2	2
Level of available tool support	13	4	0	1

