DIVISION OF COMPUTER SCIENCE

Early Elicitation and Definition of Requirements for an Interactive Multimedia Information System

S Jones
C Britton

Technical Report No.236

October 1995

Early Elicitation and Definition of Requirements for an Interactive Multimedia Information System

S. Jones and C. Britton
School of Information Sciences, University of Hertfordshire, Hatfield, AL10 9AB, UK.
Tel: 01707 284321, Fax: 01707 284303, Email: S.Jones, C.Britton@herts.ac.uk

Abstract

The literature on requirements engineering contains little in the way of either theoretical guidance or empirical case studies relating to the specification of requirements for interactive multimedia information systems. This paper describes some of the special difficulties currently encountered in the early stages of developing such systems. It then presents a description of the methods and tools used in the early stages of the MARS project, whose aim is to develop an interactive multimedia system for use in school-based health education and anti-smoking campaigns directed at children aged 9 - 10. The paper ends with a discussion of lessons learnt with respect to some current issues in requirements engineering, and considers the extent to which the particular problems identified earlier in the paper were addressed by the methods and tools used in the MARS project. We conclude that further work in this area is badly needed, so that better guidance on the specification of requirements for interactive multimedia information systems can be given in future.

Keywords: requirements elicitation, requirements definition, multimedia, information systems, education

1 Introduction

This paper describes the elicitation and definition of requirements for an interactive multimedia educational system. It lists some of the special difficulties currently encountered in the early stages of developing such systems, and describes the methods used in a particular project to address these difficulties. It ends with a discussion of lessons learnt with respect to some current issues in requirements engineering.

The project discussed is the MARS project [Jones et al 95], whose aim is to develop an interactive multimedia system for use in school-based health education and anti-smoking campaigns directed at children aged 9 - 10. The MARS system is an example of the broad class of information systems which we will refer to here as 'information service systems' (ISS). Information service systems are defined as systems which provide information in the public domain to members of the general public in an easily accessible form. An increasingly familiar example of this type of system is the 'information kiosk' [Leventhal et al 94] or public access 'point of information' system found in museums, libraries, shops and stations. However this paper focuses on information service systems which are used not simply to inform, but also to educate, in the sense of changing their users' attitudes and behaviours. We will refer to these systems as educational information service systems (EISS).

The emergence of the 'information superhighway' and the rapid development of new technologies, such as multimedia, have lead to a sudden increase in the demand for computer-based information service systems. Unfortunately, however, the amount of guidance available as to how to develop such systems has not increased at a similar rate. Systems and software engineers are being forced to deal with the complexities of specifying and developing these systems in a largely ad hoc manner. The lack of suitable methods means that the costs of development are often high, and the quality and usability of the resulting systems are frequently low.

This paper describes the methods and tools used in the early stages of the MARS project, and presents some initial observations regarding their applicability in the development of other interactive multimedia information service systems. In section 2, we outline some of the particular problems faced by developers in eliciting and defining requirements for EISS. Section 3 describes the methods and tools used in eliciting and defining requirements for the MARS system, and the way in which the problems identified in 2 were addressed in the early stages of the project. Section 4 presents some thoughts regarding the relevance of our experience in the MARS project to current issues in requirements engineering. The paper ends with some general conclusions.

2 Problems of Defining Requirements for Interactive Multimedia Information Service Systems

Many of the most difficult problems of software system development are exemplified in the development of interactive multimedia information service systems, especially those which are intended to educate, rather than simply inform their users. The technologies employed in these systems are new and changing all the time so that the possibilities they present are easily both over- and under-estimated by developers and potential users of a system. The difficulty here is compounded by the fact that both the group of potential system users, and the development team may be large and disparate in the case of a multimedia project. The move to using computers for information presentation, rather than information processing, also means that many of the traditional system development tools are inappropriate in this context, and other methods and tools must be employed. Finally, it should be noted that the market for systems of this kind is extremely fast-moving, as companies need to corner lucrative niches before any of their competitors. There is therefore a need for requirements methods and tools which can maximise the effectiveness of a 'quick and dirty' development.

2.1 The use of new and rapidly evolving technologies

Many of the problems we have so far experienced in defining requirements for the MARS system have been due, we believe, to the fact that the system employs new and rapidly evolving technologies - in this case, those relating to the use of multimedia. Some particular issues identified in this context are as follows.

2.1.1 The need for effective communication

Effective communication, particularly between developers and clients or other stakeholders is always of paramount importance in requirements engineering. However, differences in the experience of members of the project team often tend to make this difficult, with, for example, developers knowing little about the problem domain and clients being unfamiliar with the technology which might be appropriate to solving their problem.

This kind of problem is exacerbated in cases where it is proposed that the system to be built should employ new technologies such as, currently, multimedia. In the MARS project, the developers were not only unfamiliar with the domain of school health promotion, but were also inexperienced with the particular configuration of hardware and software which the project was constrained to use: while relevant experience of, for example, Macintosh-based hypermedia development projects using Hypercard and FoxPro, and of PC projects using Knowledge Pro was easily available, there was, on the other hand, no-one with any previous experience of using Authorware Star on a PC. Most of the domain experts had little or no direct experience of multimedia applications. Problems of this kind are often further compounded by the evolution of related technologies during the course of the project. For example, as the cost of hardware (and memory) drops and new software (and better compression algorithms) become available, it may suddenly be feasible to include much longer videos in the MARS system than it has been until now.

Lack of experience with relevant technology means that members of the development team are likely both to over- and under-estimate the capabilities which might feasibly be provided. Different members of the team start the project with different expectations, depending on their previous experience, and each member's expectations evolve during the course of the project, probably at a different rate to those of others. All these factors make communication both more difficult and more necessary. Lack of any societal norms, or even experience and understanding common to all members of the project team, means that key terms (eg video) are easily mis-understood, or given different significance by different members of the team (the term 'video' might variously be interpreted as something like a television programme, a home video or a computer game). Yet key concepts and terms such as these must be accurately understood and agreed upon if realistic requirements are to be defined.

The problem of communication caused acute difficulties in the MARS project. While early discussions between developers and domain experts appeared to progress and lead to satisfactory outcomes with all parties agreeing and feeling that they had understood each other, it became clear at a later stage (during a demonstration of the prototype) that some aspects of the system had been fundamentally misunderstood by at least one of the domain experts.

2.1.2 Evolution of the problem domain

The development of experience and understanding of new technologies through the course of a project can lead to radical re-thinking of the domain, as new ways of doing things become apparent. For example, the use of interactive video in the MARS project means that the incorporation of skills training (to provide children with, for example, the social skills to say no to a cigarette if they want to) into school health education programmes is now a real possibility, whereas before it might not have been considered owing to a lack of resources. Methods for requirements definition must therefore cater not only for the evolution of requirements, but also for evolution in the domain itself.

2.2 Involving a large and disparate group of users, experts and developers

2.2.1 A large group of potential users

Information service systems (ISS) are defined as systems which provide information to members of the general public in an easily accessible form. Educational information service systems (EISS) may be directed at a particular subset of the general population - MARS, for example, is intended mainly for use by 9 and 10 year old children - but still have to cater for considerable differences in ethnic and social backgrounds, educational attainment, knowledge of the domain and experience or confidence with the use of computers. Considering other 'indirect' users of the system - including, in the case of MARS, teachers and school nurses - broadens the group of users from which requirements need to be solicited still further. Methods for defining requirements for EISS need to be able to record and integrate requirements from representatives of many kinds of system users.

2.2.2 Many different fields of expertise

As well as involving domain experts such as teachers, school nurses, developmental, educational, social and health psychologists and representative users, a project such as MARS should ideally draw on expertise in many different fields in developing material for use in the system. Specialists in human-computer interaction, graphic design and communication, 3-d modelling and animation and video production and editing would all make valuable contributions to the project. each of the team members will bring their own views and experiences to the project which can enrich the development process, but also lead to problems of management and communication.

2.3 Information presentation (rather than processing)

Many information service systems are better characterised as systems for information presentation, rather than information processing. This is certainly true of educational ISS such as MARS. In traditional terms, this means that more emphasis must be placed on non-functional requirements, than on input-output specifications of required functionality. MARS is also intended to educate - to change the user's attitudes and behaviours - rather than simply to present the user with information which s/he may choose to ignore. This means that low-level requirements may be very hard to identify, as education is not an exact science and there are wide-spread and conflicting views on what constitutes a 'good' educational tool.

2.4 Staying ahead in a fast-moving market

Related to the rapid development and evolution of multimedia technology described above is the fact that the market for multimedia systems is extremely fast-moving. Companies involved in multimedia development often find themselves developing to market, rather than developing a bespoke system for a particular client, so that they can capture a particular market niche before any of their competitors. There was no paying client for the MARS project, which began simply with the perception of a new opportunity for applying emerging technology in a particular domain. Developments obviously need to progress quickly in this context. We must therefore acknowledge the need for requirements practices and procedures which are flexible and efficient in delivering effective requirements with a minimum of cost, time and effort.

3 Eliciting and Defining Requirements for MARS

The domain of application for the MARS project was that of school-based health education programmes about smoking. Financial constraints meant that development had to be done using equipment already available at the University: in this case, the Authorware Star multimedia authoring package running under Microsoft Windows 3.1 on a 486 PC with a maximum of 20 MBytes of memory. The target user group included school children aged 9 - 10 years as well as teachers and school nurses, and the team of experts available for the project at this stage included experts in nursing, developmental psychology, social psychology and human-computer interaction. As there was no paying client for the project at this stage, we were effectively developing to market.

Having reviewed the literature, we found little direct guidance regarding methods, tools or techniques intended specifically for the elicitation and definition of requirements for interactive multimedia systems. Few relevant case studies were found. Some studies addressed particular aspects of the problem described above (for example Gough et al [Gough et al 95] considered the problems of developing to market rather than for a particular client), but none related to the problem of developing interactive educational multimedia systems. There seems to be an assumption in the community that the specification of requirements for multimedia systems is not problematic, but the reasons for this assumption vary from 'surely its just like developing any other system' to 'well, its not really software development, so why bother to think about it'.

In the absence of any specific guidance, we decided that our approach, during the early stages of requirements elicitation and definition, should focus on the use of incremental prototyping and structured natural language definition of requirements. We felt that the early use of prototyping would help us to address the possibility, which we foresaw, of problems with communication. We felt unsure as to the applicability of any particular requirements notation in our particular context, and therefore decided to opt for the flexibility and speed of using structured natural language, hoping that it would become clear at a later stage in the project what structured notations, if any, would assist us in refining the requirements for MARS. Having considered a number of possible structures [DTI 87, IEEE 84, Davis 93] for our initial requirements documents, we decided to use a slightly modified version of that suggested in the STARTS Guide for developing real time systems, noting that the emphasis it places on functional requirements would probably not be appropriate in defining requirements for MARS, and that some extra headings relating, for example, to 'Media Requirements' might be necessary.

The rest of this section describes our approach in more detail.

3.1 Elicitation

Our first step was to identify experts willing to contribute to the project in as many relevant areas as possible. We began the project with a meeting involving most of the domain experts, to discuss our initial ideas for the project and begin to familiarise ourselves with the domain. We also used a video of a discussion about smoking at a local youth club to get some idea of the knowledge and attitudes of some of our target user group.

As a result of this process, we established four high level objectives for the system:

- to increase awareness and knowledge of the risks and effects of smoking in children aged 9 - 12;
- to modify attitudes to smokers and smoking in those children;
- to teach skills which enable children to resist pressures to smoke;
- to reduce the prevalence of smoking

We then roughly specified a prototype system which would aim to address these objectives, and form a basis for further discussions regarding requirements for, and possible uses of, a system of the kind we envisaged.

As soon as the prototype was sufficiently developed to demonstrate most aspects of the kind of system we proposed to build (i.e. once we could demonstrate all forms of media and give examples of the ways in which they could be combined, as well as showing how the user would be able to interact with and control the system), and sufficiently robust that it could be operated smoothly by the developer, we began to invite individual domain experts to view the system. Sessions with domain experts took the form of semi-structured interviews (see, for example, [Cordingley 89]) of the kind commonly used in knowledge engineering.

Each interview lasted 1 - 2 hours and followed a standard format, involving the expert and two members of the development team. With the expert's consent, interviews were tape-recorded for later reference. At the beginning of the interview, the date and location of the demonstration were recorded as well as the name and area of expertise of the expert. One of the developers reminded the expert (or explained to those experts who hadn't been present at the initial project meeting) of the aims of the project, and stated the high level objectives listed above. Some description of the possible scenarios in which a system such as MARS might be used was also given. It was explained that the purpose of the demonstration was to illustrate the possibilities of multimedia, and to obtain comments and feedback from the experts on various aspects of the system. We took care to emphasise the fact that the system should be viewed simply as a first prototype, and that we were intending to throw it away and start again from scratch if need be. We asked experts to voice both positive and negative comments as they occurred to them during the course of the demonstration.

In previous projects, we have noticed a tendency for people evaluating prototype systems to get unduly caught up in fairly superficial details such as the colour of the background or the quality of graphics. Although such aspects of a system are clearly important, we also wanted experts to provide us with input on higher level issues and requirements for a future system. We therefore asked each expert to wear a particular 'hat' for the duration of the demonstration - i.e. to focus on the particular domain for which they were acting as our expert. The system was then demonstrated by one of the developers, while the other took notes on the expert's reactions and followed up any interesting or ambiguous comments s/he made.

After the demonstration, the expert was reminded of the overall objectives of the project, and, in the light of these objectives, was asked to answer some general questions in relation to his or her domain of expertise. For example, the expert in school nursing was asked questions relating to issues such as:

- whether, in her opinion, the health messages given out by the system were appropriate
- whether school nurses were likely to want to use a system like this
- whether it would fit in with their current practices etc

These questions were generated prior to the interview by using a standard requirements specification template (see below) as a checklist to ensure that all areas relevant to a particular field of expertise would be covered in the interview.

An expert's suggestions regarding the prototype were, where possible, implemented immediately after the interview, so that one expert's suggestions were evaluated by another as part of the next demonstration.

Because some parts of the prototype were initially very under-developed (for example, the part of the system demonstrating the concept of an interactive story consisted of only two screens with captions but no pictures, and two buttons on the second screen enabling the user to choose one of two possible outcomes), we drew up story boards to show to experts to explain more fully how these parts of the system might work. However, it is worth noting that we never needed to use these story boards as all experts seemed to understand the principles of using an interactive story from the bare minimum of information in the system.

One of the most difficult things about using a prototype to elicit requirements in this way was the need to remain objective, and not to try to defend the system when it was criticised by an expert. This would obviously have been counter-productive, as it might either have caused the expert to forbear from making any further negative comments (thus robbing us of valuable requirements for the future system), or have lead to ill-feeling and reduction in co-operation on the part of the expert. It is important to develop strong bonds between members of the development team, so that they can work together effectively, and accept criticism from individuals outside of the team without fragmenting. A natural part of developing such bonds is the creation of in-group agreements (for example, regarding the way the system should look) and norms, and a tendency to defend both oneself and other members of the group against perceived attacks from outside. Such agreements must, however, not be allowed to become too strong, and the tendency to defend against criticism must be curbed so that requirements from domain experts can continue to be absorbed. This is an important part of maintaining effective communication.

3.2 Definition

Requirements elicited from domain experts through the process described above were recorded in natural language, structured within a standard template for requirements specification. The requirements document was updated after each interview with an expert, and evolved in parallel with the prototype as illustrated in figure 1.

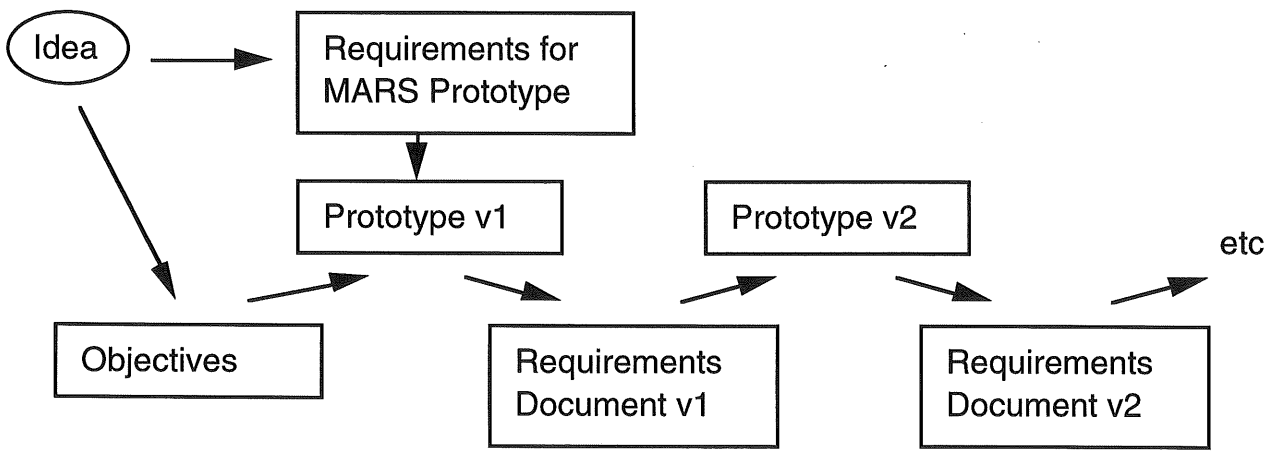

Figure 1: Overview of the Process of Eliciting and Defining Requirements for MARS

The main headings used in the template were those recommended in the STARTS Guide [DTI 87]. We focused, at this early stage in the project, on the first two of the three recommended levels:

Level 1: Introduction
- Background
- Outline System Requirement
- Outline System Environment
- Structure of the Document
- Definitions
- References

Level 2: User Description
- The Environment
- Function
- Operation
- Life Cycle Aspects
- Performance
- Constraints
- Assumptions

To support traceability, requirements recorded in level 2 of the document were tabulated with information of the kind suggested by the IEEE Guide to Software Requirements Specifications [IEEE 84]. An example of tabulated requirements for the MARS system is shown below. In tables such as these:

- each requirement was given a unique number;
- the origins of requirements were encoded in terms of initials (which identify individuals or particular project meetings) and date of origin;
- a natural language definition, and, where appropriate, a rationale, was given for each requirement;
- requirements were prioritised as necessary, desirable or optional;
- if the requirement stated was a refinement of an earlier requirement, the number of that requirement was given (if the refined requirement was part of the current document, only the requirement's number was given, but if the refined requirement was in another document, the ID for that document was given as well; and
- if a requirement was subsequently refined, the number of the refined version was given (a document number may also be given as above).

2.2 The Users of the System

No.	Orig.	Date	Definition and Rationale	Pty.	Src.	Ref.
2.2.1	PM3	9.2.95	The main users of the system will be children aged 9 and 10.	N	v0: 2.1	v2: 3.4.2
2.2.2	PM3	9.2.95	Children of both sexes should feel comfortable using the system.	N		
2.2.3	SJ	10.4.95	Children of any ethnic origin should feel comfortable using the system.	D		
2.2.4	PM2	30.1.95	The system may also be used by primary school teachers.	N		
2.2.5	PM2	30.1.95	The system may also be used by primary school nurses.	N		
2.2.6	SJ	29.3.95	Parents, head teachers and school nurse managers may also be affected by the system.	D		

Figure 2: Extract from the Requirements Document (v1) for MARS

4 Discussion: Lessons Learnt

We have described some of the particular problems involved in specifying requirements for Educational Information Service Systems, and the way we have approached the early stages of eliciting and defining requirements for one such system in the MARS project. We now present a number of observations which arise out of our experience and relate to current issues in requirements engineering.

4.1 The origin of the requirements process

There has often been an assumption that the software development process begins with at least a vague statement of requirements from a client who may be interested in purchasing a software system (see, for example, [Ince 94]).

The MARS project began with the simple idea that an interactive multimedia EISS might be useful in the domain of health education. After a small number of meetings between people likely to be involved in the project, a list of 4 main objectives was produced. The aim then, in the early stages of the project described above, was to begin to develop this idea into a comprehensive requirements specification for use in developing such a system.

Many companies, particularly those currently developing EISS and other multimedia products, develop 'to market' in this way, rather than having a specific client whose needs they are aiming to fulfil. The requirements process often starts with no more than a vague idea which needs to be developed into a viable product. Requirements methods should therefore not assume the existence of either a client or a vague statement of requirements to work to, and should provide support for flexible development of ideas, particularly in the early stages of requirements definition.

4.2 The role of natural language in requirements specification

There is much debate about the role of natural language in requirements engineering, with some arguing that requirements should be expressed solely in natural language (because, for example, clients will not effectively be able to validate requirements specified using any other notation), and others arguing that natural language should hardly be used at all because it provides too many opportunities for ambiguity.

Until a more appropriate conceptual framework is developed, the 'definition' of requirements in terms of natural language text (and prototype systems) seems adequate for the early stages of elicitation and definition of requirements as described above. It means that adapting the requirements document in line with the evolution of the problem domain, as well as that of the requirements, is relatively unproblematic. Using other, more structured, notations at this stage may indeed be problematic, in that requirements that can be easily be expressed in the chosen notations may inadvertently be given more weight than others.

For developers of systems such as interactive multimedia ISS which are aimed at fast-moving markets, it is worth noting that writing and working to partial requirements specifications may at present be less problematic using natural language than it is with a structured or formal notation, and may be better than using none at all. For example, the developer of the MARS system was able to use the natural language records of requirements elicited from domain experts to progress the prototype even before they had been fully tabulated for traceability. Any practical requirements procedures must allow for such flexibility if they are to be used in real development projects in areas such as those described here.

4.3 Structuring the Requirements Specification

Many different structures for requirements specification documents have been proposed (see, for example, [Dorfman and Thayer 90, Davis 93]) and numerous standards exist for use in specifying different kinds of systems. Almost all of these standards and proposed structures pre-suppose a distinction between what are called 'functional' and 'non-functional' requirements, often proposing that functional requirements should be specified in a separate section of the document.

The frameworks for structuring requirements specifications provided by the IEEE standard and the STARTS Guide have so far been found to be useful in providing a roughly hierarchical structure of concerns and a check-list of areas to consider, but the 'data-processing' view of computer systems which they embody may not be wholly appropriate to the development of interactive multimedia ISS such as MARS. As described above, we are, in the MARS project more concerned with the way in which information stored in the system is to be presented, than we are with the way in which inputs to the system are to be processed. In filling out the third, and most detailed, level of the requirements specification for MARS, we do not believe it will be appropriate to think of the system in terms of inputs, outputs and data flows, and intend to draw on the feature-based organisation proposed in the 1993 IEEE Revised Standard [IEEE 93 - see section A.5]. Each major part, or feature, of the MARS system will be specified separately in a section which describes its general aims as well as the precise learning objectives specific to that part before going on to map out the relevant 'interaction space' using, for example, state transition diagrams for the lowest level of detail. Specifications of components of the system employing media such as video may integrate state transition diagram specifications of interactions with tools such as story boards which have traditionally been used to map out sequences in films and animations.

We also note that what have traditionally been called non-functional requirements concerning the usability of the system will be very important in the development of EISS such as MARS, and will need to be linked directly with relevant functional requirements specified as suggested above.

4.4 The role of prototyping in requirements definition

Some form of prototyping is commonly used to help developers, potential system users and clients to understand more about a domain or the possible use of a computer-based system within that domain [Davis 93].

Our experiences of using a prototype in the MARS project correspond quite closely to those reported by Gomaa [Gomaa 83]. In our project, as in his, one of the main reasons for building the prototype was that some of our domain experts were finding it difficult to visualise how the proposed system might be used. We also found that many mis-understandings between the developers and domain experts were revealed as a result of evaluating the prototype and that it provided a useful basis for developing and refining our initial definitions of requirements. Demonstrating and discussing the prototype provided developers and domain experts from many different fields with an opportunity to build a shared understanding, and a common vocabulary for further discussion, thereby addressing the need for effective communication described above.

We would, however, caution that even after viewing a demonstration of the prototype, at least one expert was still confused about the way in which a final system might be used in the context of a classroom situation. A solution to this might be to allow experts to observe the use of the prototype in the environment for which it is intended, or simply to allow them to view the prototype for longer.

A final caution relates to what has been called 'functional fixedness'. It has been shown that once people are informed that an object has a particular purpose, they are less able to think of other ways in which that object might be used. Similarly, once people such as our domain experts have been shown a prototype, there is a danger that they will lose the ability to think of alternative designs for a final system, or other things a system might do. They may also find it difficult to challenge what they see as the developer's own personal ideas about the way the system could look. In the MARS project, we found that experts only rarely suggested that things in the system should be dropped or changed - most of their suggestions related to possible additions or enhancements of existing features. We would like once again to stress the importance, firstly of emphasising the fact that the prototype should not be viewed as a given, and secondly of attempting to remain objective during evaluations, not defending the prototype when criticised by an expert. We believe that this will be important whenever prototypes are used in requirements specification.

4.6 The use of scenarios

The use of scenarios is increasingly seen as a promising solution to the difficult problems of communication, particularly between developers and domain experts [Loucopoulos and Karakostas 95].

During the early stages of the MARS project, we felt quite strongly that an interactive multimedia system would be useful in school-based health education programmes, but had only a vague notion of exactly how and where such a system could be used. Domain experts agreed that such a system might be useful, but needed time to consider the ways in which the introduction of new technology would permit new forms of health education. Thus one of the aims in the early stages of the project was to generate scenarios of use which could later be employed to capture further requirements for the system. In this connection, we note that a trade-off must be made in the development of systems for fast-moving markets in which access to potential users and experts is limited: on the one hand, care must be taken not to home in on a particular scenario of use too quickly as other possibilities may inadvertently be precluded, but on the other hand it is important not to drag out the early stages of the project for too long, as the time, patience and enthusiasm of domain experts is limited, and the market for systems such as MARS is moving forward very quickly. A delicate balance must be struck. In MARS we opted to use scenarios only informally in generating material for the semi-structured interviews described in section 3.1.

4.7 Tools for requirements definition

For the early stages of the MARS project described above, we found it adequate to treat the requirements document in the same way as any other controlled document relating to the software development process. The document was produced using the team's standard word processor (Word v5.1), and within the standard document template provided by the team's quality management procedures. This situates the process of requirements definition conveniently within the normal software quality management framework and has so far meant that those involved in requirements definition have not had to spend time learning to use specialised requirements management packages, and can therefore produce initial requirements documents quite quickly. It remains to be seen whether the production of a complete specification of requirements for the MARS system can feasibly be achieved in the same way.

5 Conclusions

We have described some of the special difficulties currently encountered in the early stages of developing a particular class of systems referred to as interactive multimedia educational information service systems (EISS), and have presented a number of observations regarding current issues in requirements engineering. We have based our observations on our experiences of the MARS project which was begun last year at the University of Hertfordshire.

Before embarking on the MARS project, we carried out a search of the relevant literature but found little in the way of either theoretical guidance or empirical case studies relating to the specification of requirements for EISS or even of interactive multimedia systems in general. We feel that further work in this area is badly needed. To this end, we are intending to continue our work on the MARS project, for example to investigate the role of structured and/or formal notations in modelling systems such as MARS to support further definition of requirements, and the relative effectiveness of using specialised requirements specification tools in this context.

References

E. Cordingley, "Knowledge Elicitation Techniques for Knowledge-based Systems", in *Knowledge Elicitation: Principles Techniques and Applications* D. Diaper (Ed), Ellis Horwood, 1989.

A. Davis, "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.

M. Dorfman and R. H. Thayer, "Standards, Guidelines, and Examples on System and Software Requirements Engineering", IEEE Computer Society Press, 1990.

H. Gomaa, "The Impact of Rapid Prototyping on Specifying User requirements", ACM SIGSOFT Software Engineering Notes, 8(2), April 1983.

P. Gough, F. Fodemski, S. Higgins and S. Ray, "Scenarios - An Industrial Case Study and Hypermedia Enhancements", in Proceedings of RE95, the Second IEEE International Symposium on Requirements Engineering, March 1995, IEEE Computer Society Press, 1995.

IEEE, "Guide to Software Requirement Specifications", ANSI/IEEE Standard 830-1984, IEEE, 1984.

IEEE, "Recommended Practice for Software Requirement Specifications", IEEE Standard 830-1993, IEEE, 1993.

D. Ince, "ISO 9001 and Software Quality Assurance", McGraw-Hill, 1994.

S. Jones, C. Britton, D. Knight and N. Whiteley, "Requirements for a Multimedia Health Information System", University of Hertfordshire, School of Information Sciences, Technical Report No. 224 1995.

L. Leventhal, K. Instone and B. Teasley, "Hypertext-based Kiosk Systems: Seven Challenges and an Empirical Study" in Proceedings of EWHCI94, the East West International Conference on Human-Computer Interaction, 1994.

P. Loucopoulos and V. Karakostas, "System Requirements Engineering", McGraw-Hill, 1995.