Experience with SCRAM, a SScenario Requirements Analysis Method

Sutcliffe A.G. and Ryan M.
Centre for HCI Design, School of Informatics, City University, Northampton Square, London EC1V 0HB, UK. Email: sf328@city.ac.uk

Abstract
A method of scenario based requirements engineering is described that uses a combination of early prototypes, scenario scripts and design rationale to elicit and validate user requirements. Experience in using the method on an EU project, Multimedia Broker, is reported. Quantitative data on requirements sessions is analysed to assess user participation and quality of requirements captured. The method worked well but there were problems in the use of design rationale and control of turn taking in RE sessions. Lessons learned in using the method are summarised and future improvements for the method are discussed.

Keywords Requirements Elicitation, Validation, Prototyping, Empirical Study, Scenario analysis.

1. Introduction
In spite of the increasing attention that scenarios have attracted in requirements engineering [1], few methods have emerged to guide the practice of scenario based requirements analysis or validation. One of the exceptions is the Inquiry Cycle of Potts [5] which we developed in a scenario based requirements analysis method (SCRAM) that recommended a combination of concept demonstrators, scenarios and design rationale [8, 9]. Initial experience with the method was promising in analysing requirements for a decision support system with safety critical implications, although the empirical study of the methods used did uncover the need for thorough training for its effective use.

In this paper we report further experience with the SCRAM method which was used for requirements analysis on the Multimedia Broker project. The paper is divided into seven sections. First we introduce SCRAM, which is grounded in scenarios, prototyping and design rationale. The next section details how the requirements analysis sessions were conducted and briefly describes the Multimedia Broker prototypes. Section 4 reports the results of the requirements capture, while section 5 gives a quantitative analysis of user participation and quality of the requirements. This is followed by the experience and lessons learned in use of the method which are reported in section 6, and a brief discussion in section 7.

2. The SCRAM method
The method is based on four techniques for requirements capture and validation:
• Use of prototypes or concept demonstrators: a key concept is providing a designed artefact which users can react to.
• Scenarios: the designed artefact is situated in a context of use, thereby helping users relate the design to their work/task context.
• Design rationale: the designer’s reasoning is deliberately exposed to the user to encourage user participation in the decision process. The QOC (Questions, Options, Criteria) notation [3] is used to illustrate the various trade-offs; see Figure 1.
• Whiteboard summary: the designer’s requirements are summarised on a whiteboard to identify dependencies and priorities.

Figure 1. Navigation design rationale.

The techniques are combined with process guidance for the requirements analyst. The method consists of the following phases:
1. Initial requirements capture and domain familiarisation. This is conducted with conventional interviewing and fact finding to gain sufficient information to develop a first concept demonstrator.

2. Specification and development of the concept demonstrator. The concept demonstrator has limited functionality and interactivity, so it can only be run as a ‘script’ to illustrate a typical task undertaken by the user. Scripts illustrate a scenario of typical user actions with effects mimicked by the designer.

3. Requirements analysis-validation session. The users are invited to critique the concept demonstrator and interview the designer. The session is recorded for subsequent analysis.

4. Session analysis. Data collected during the analysis session is analysed and conclusions reported back to the users.

The end point of the method delivers a requirements specification comprising the concept demonstrator, a set of analysed design rationale diagrams expressing users’ preferences for different design options, and specification as text, graphics or more formal notations depending on the requirements analyst’s choice. In addition, video of the analysis sessions is available for requirements traceability analysis.

2.1 Session design

The analysis session is intended to encourage co-operative requirements capture between two, possibly three, users and two requirements engineers. One engineer acts as the operator of the concept demonstrator and the other fulfils an explainer-rapporteur role. The presence of at least two users helps balance the ‘ownership’ of the session away from the developers and is productive in producing conversation about the artefact, domain and requirements. The sessions are run according to the schedule:

1. Introduction and briefing, to put the users at ease, explain the developer roles and emphasise that it is the artefact, and not the users, that is on trial.

2. Demonstration and scenario run through. The concept demonstrator is illustrated in a scripted sequence, linked to the scenario. Probe questions are asked at key points in the demonstration script, while design rationale diagrams are used to explain design options.

3. Summary phase. The explainer-rapporteur summarises the key facts learned during the session and interactively constructs a concept map of requirements (both functional and non-functional) on a white board and requests any comments. If the users wish to take copies of the concept demonstrator away with them they are encouraged to do so.

Following the session the video data and audio soundtrack are analysed with notes taken during the session. The depth of analysis depends on resources available. The design rationale diagrams are used as a shared artefact to promote discussion, and gesture is used where possible to illustrate differences between the options by pointing to the screen. One obvious problem is bias towards the option implemented in the demonstrator. This can be counteracted by using storyboard sketches of the other options and by more vigorous critiquing by the developers of the implemented version. In particular, use of the criteria, which incidentally capture non-functional requirements, is a powerful way of promoting critical thought. The motivation of using design rationale is to explore the possible solutions with the user and amplify requirements via design exposure.

Apart from the key points in the demonstration script the developers are encouraged to cede the floor space to the users as much as possible (i.e. the conversation must not be dominated by questioning and explanation).

3. Case study setting

SCRAM was used to establish requirements on the Multimedia Broker Project that aimed to integrate Information Retrieval and Internet technologies, and develop brokerage services to enable authors to find resources for successful multimedia publishing over the World Wide Web. The terms of reference in the project proposal were to support and advise users when searching and retrieving information from distributed, multimedia databases. It should also support users to create multimedia documents and applications, and will enable publishers to provide resources for the Broker and charge users to access it.

A feasibility study carried out in advance of the project identified four functional groupings: information retrieval, brokering, payment and administration, and pre-press authoring support. This study focused on requirements information retrieval (see requirements 1-3 below), so brokering (requirement 4) and payment/pre-press authoring (requirement 5) received less attention as these were covered in other investigations. The requirements investigation attempted to elicit users’ views on both functional and non-functional requirements and to obtain their preferences.

3.1 Prototype system functionality

The prototypes were developed in Visual Basic but had limited functionality and were designed to run as a concept demonstrator script. However, the user interface did show the effect of carrying out commands even
though no back end functionality (e.g. for searching) was implemented. In the scenario several design issues were embedded relating to information search and retrieval. The issues followed the method’s basis in design rationale for requirements issues with several potential solutions.

1. Browsing for resources. The first issue was how to browse the broker’s information about multimedia databases and web resources. Four design options were proposed:
   - a virtual reality 3D walkthrough of indexing concepts describing the resources in a spatial rooms/building metaphor
   - a 2D Map Navigator showing entities and relations, based on our previous work [7].
   - a 2D category map indexing topics for resources associated with categories
   - a geographic view of the resources’ location in world and European maps
   The concept demonstrator for this part of the scenario is illustrated in Figure 2.

![Figure 2. 2D Map Navigator for browsing and query by pointing.](image)

2. Searching for resources. The second issue, searching the dataspace, showed a variety of query methods. Query by pointing was demonstrated on the 2D Map Navigator showing entities and relations, and a restricted natural language template interface, (see Figure 4) was included that allowed the user to form fill high level templates (e.g. find resources that explain < ...... > tasks). Another design option for this requirement was provision of intelligent support for iterative searching, by advice on whether to broaden or narrow queries, use synonyms, etc.

3. Results presentation. The third issue dealt with how retrieved data is displayed and linked to its originating query. Three options were shown for summary browsers of the retrieved results:
   - spatial juxtaposition of queries and of their results
   - interactive graphical summaries of results sets
   - relevance feedback, whereby users could mark items in the results set and then submit these as examples that the system analysed to form further queries.

4. Brokering advice and negotiation. Advising was portrayed as information displays about resource vendors and authors, and resources for particular domains. Brokering also entails support for negotiation. The system provided this with form-fill templates and intelligent dialogue to help negotiation with the user. Design options ranged from intelligent software to opening a simple communications channel to a human broker.

5. Payment and pre-press authoring support. Solutions for these requirements were illustrated by virtual transactions over the Internet (e.g. 1st Virtual) and dialogues, to guide the user on how and when to pay, maximum charges, most cost-effective resource retrieval, etc. Information on copyright, where items can be recombined into new products, and international regulations of data mobility were also presented. Finally, interfaces to authoring tools were illustrated so that users could create their own multimedia documents.

3.2 Requirements analysis sessions

Twenty users in total participated in four sessions:
1. PTRC - 19.08.96, with 3 users.
2. PTRC - 04.09.96, with 11 users.
3. IMT - 23.09.96, with 2 users.
4. PTRC - 25.11.96, with 3 users.

The user organisations were PTRC - Education and Research Services Ltd, a specialist transport research group who publish surveys and research reports, and the Institute for Media technology (IMT), Stockholm, multimedia title developers. The method was used in two formats; first as described in section 2 above, in sessions 1, 3 and 4; and secondly in a seminar format for session 2. The system was demonstrated in lecture format and participants were encouraged to ask questions. Participants filled in the questionnaires; however, no recordings were made so only preference data was captured. Session 3, with only 1 user, has not been included in the following analysis.

The users in all sessions were either members of staff of the user (publisher) partners in the Multimedia Broker consortium, or selected members of their current client...
base. Each user completed a pre-session questionnaire providing details on their computer experience. Users were then taken through each stage of the prototype using the SCRAM method (see section 2). The users also completed a post-session questionnaire in which they rated each design option and gave further details of their requirements. Collaborative construction of the requirements summary at the end of the sessions gave further opportunity for feedback after the demonstration.

4. Case study results

The requirements were synthesised from a mixture of notes taken during the sessions, listening to the audio tape log, questionnaires completed by the users, and the whiteboard summary held at the end of the sessions. For the browsing requirement users showed a preference for the 2D entity relation map and the Text Thesaurus interface; see Table 1. The 3D walkthrough was considered aesthetically pleasing but possibly time consuming and difficult to use.

Table 1. Post session questionnaire feedback

<table>
<thead>
<tr>
<th>Navigation</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Walkthrough</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2D Entities/relation map</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>2D Topic expansion map</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Text Thesaurus</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>World/Europe Maps</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Using the scale 5 = essential, 1 = don’t like at all, averaged for users in each session on exploring/browsing requirements for sessions 2, 3 & 4.

4.1 Information searching

The constrained natural language interface (see table 2) was the preferred interface for constructing queries using key phrases as well as key words. Users were keen to use natural language templates which they could customise to their own particular needs. Users approved of the form fill interface, and requested a query library for storing and retrieving re-usable queries, and a ‘Query by Example’ tool in order to retrieve similar queries. Text manipulation tools, such as an expanded thesaurus, were well received.

Table 2. Post session feedback on querying

<table>
<thead>
<tr>
<th>Querying functions</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Map showing entities/relations</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>2D Topic expansion map</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>4</td>
</tr>
</tbody>
</table>

Logical operators (AND, OR, etc.) were considered hard to use and caused confusion. The users required intelligent support to help reformulate queries pertinent to their needs. Prioritising input and output, and relevance feedback in searching by finding similar results to items marked by the user, were favoured.

4.2 Brokering support and results presentation

Offering different ways of sorting and presenting results (e.g. graphically using bar charts, and spatially linking queries to results) was considered useful. Searching for key words/phrases within results and marking items of interest were supported. The users wanted help to clarify their requirements and suggest possible search choices and alternatives (Table 3). However, the users did point out that these facilities should be user driven. Brokering by goodness of fit matching for suitable multimedia resources was rated favourably and the users were keen on the system advising them about the suitability and content of the resources. Compatibility with other applications (such as MS Word™, MS Excel™) emerged as another desirable requirement.

Table 3. Post session feedback on negotiation/brokering

<table>
<thead>
<tr>
<th>Support services required</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help clarify requirements</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Negotiate copyright</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Advise on/suggest possible choices</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Match user requirements to suitable resources</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Facilitate payment</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Help multimedia resource delivery</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>Advise on MM designs</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>Allow export to other applications (e.g. Word)</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>14</td>
<td>5</td>
</tr>
</tbody>
</table>

(5 = essential, 1 = don’t like at all).

4.3 Non-functional requirements
Speed and ease of learning were considered as the most important non-functional requirements; see Table 4. Different level users must also be considered.

Table 4. Post session feedback on non-functional requirements

<table>
<thead>
<tr>
<th>Important criteria</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to learn</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Portability</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Robust communication</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Quick response time</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>Integration (with authoring)</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

(5 = essential, 1 = don’t like at all).

Several usability issues were raised; for instance, that the screens must be simple and easy to use; the system should have a tutorial and an easy to use help system; the system should adapt to different levels of expertise; and users should have control over the brokering/negotiation support.

4.4 Requirements summary

The following points summarise the key requirements captured in the sessions:

1. The users liked the support services and intelligent tools. In particular, they wanted the system to help clarify their needs and advise on various resources and results which they may find interesting and useful. However, this dialogue should be user driven.

2. The users were very interested in any support tools which could help retrieve relevant information. They showed considerable support for the constrained natural language interface, which they could customise to suit their needs.

3. Support for iterative searching using pertinent and relevant reformulation advice to users’ needs was rated highly. Relevance matching, i.e., finding similar results to an already retrieved item, was considered useful.

4. The users liked graphical front ends but they were wary that 3D/VR interfaces might be difficult and cumbersome to use.

Figure 3 illustrates the requirements map constructed during the post-session discussion for the first walkthrough session. It illustrated the key issues identified by the facilitator/users during the de-briefing session.

5. Empirical analysis of sessions

5.1 User participation

The transcript analysis of the first session shows significant contributions from two of the three users. The Facilitator consistently talked throughout the discussion (55% of total utterances) as shown in the time line analysis in Figure 4 which shows the speech turns for the participants. He asked the group for their opinion and encouraged users to articulate their thoughts. The concept demonstrator operator spoke briefly to explain the scenario for each interface. Even though the users participated effectively, (18, 13 and 4% of utterances), the facilitator probably commanded an excessive proportion of the conversation and left few gaps for the users to take conversation initiative.
The pattern in the second session for which we have discourse data is more encouraging, with less facilitator dominance, who was the same individual. Both users participated frequently; however, the conversation is still dominated by the facilitator and operators overall (63%). This is also reflected in the pattern of conversational turns, illustrated in Figure 5. One of the problems is that explaining the scenario and design takes time, so the designers tend to dominate conversation if they are not careful. The method does encourage explicit breakpoints when user opinion is actively solicited as key points in the scenario; however, this may restrict user initiative in other parts of the sessions.

Figure 5. Participants’ speech turns shown over time in PTRC session 4.

Similar patterns were found in the IMT session 3 (not illustrated), so the conversation analysis does provide evidence of active participation, but with the caveat that we need to reduce the overall volume of the facilitator/operator conversation.

5.2 Quality of requirements captured

The requirements captured during sessions 1, 3 and 4 are summarised by category in Table 5.

Table 5. Requirements categories summed for sessions 1, 3 and 4.

<table>
<thead>
<tr>
<th>Requirements category</th>
<th>Total reqs</th>
<th>% (rounded)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>User’s task</td>
<td>24</td>
<td>37</td>
</tr>
<tr>
<td>Non-functional</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Usability</td>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td>User interface</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>User profile</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>65</td>
<td>100</td>
</tr>
</tbody>
</table>

Most requirements captured were related to the user’s task, which can be regarded as a sub-class of functional requirements. Functional requirements described needs for automated processes and algorithms. Few non-functional requirements were elicited; however, usability requirements were detailed for screen and dialogue design as well as the general, more non-functional requirements for ease of use and learning. Some requirements were also captured for the type of user.

Most of the requirements captured were clear and specific (52%), and only 10% were ambiguous. Most of the requirements were stated in the concept demonstrator; hence, the session validated existing requirements rather than captured new ones. Most of the requirements shown as design options were approved by the users, although the 3D graphics navigators/browsers and virtual reality extension of this idea were viewed with some scepticism. The general reaction was that while 3D graphics were attractive the medium was not appropriate for professional tools. Several requirements were elaborated during the sessions, and the topics that were commented on and for which the users provided new ideas are shown in Table 6.

Table 6. Topics in sessions 1, 3 and 4.

Additional requirements capture after discussion with the users.

<table>
<thead>
<tr>
<th>Requirements category</th>
<th>session 1</th>
<th>session 3</th>
<th>session 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browsing and navigation</td>
<td>user terms &amp; concepts</td>
<td>tags on map, image thumbnails</td>
<td>user views &amp; filters, info layers</td>
</tr>
<tr>
<td>Querying</td>
<td>simple searches, narrowing strategies, phrase + keywords, booleans, templates &amp; examples</td>
<td>feedback on results, templates + keywords, drill down searches</td>
<td>feedback on results, iterative queries</td>
</tr>
<tr>
<td>Intelligent advice</td>
<td>approx. matching, active thesaurus</td>
<td>auto extract terms, approx. matching</td>
<td>active thesaurus, auto extract terms</td>
</tr>
<tr>
<td>Presentation planning</td>
<td>----</td>
<td>----</td>
<td>interactive queries on result display</td>
</tr>
<tr>
<td>Brokering</td>
<td>----</td>
<td>negotiation templates</td>
<td>retrieve sample before paying, advice not auto matching</td>
</tr>
<tr>
<td>Usability &amp; NFRs</td>
<td>response times user adaptation</td>
<td>multiple languages</td>
<td>user configuration choice of UI style</td>
</tr>
</tbody>
</table>

As can be seen from Table 6, there were considerable inter-session differences and some topics evoked more contributions than others. Navigating and browsing part
of the demonstration provided feedback in all sessions with some new requirements for using users' terminology, and adding views and filters. Querying facilities provoked many comments and several new requirements on feedback and iterative querying, while the intelligent advice received similar comments favouring an active thesaurus to help the user. Presentation planning and brokering services received few comments and extra requirements were only captured in session 5. This may reflect the lower emphasis that was placed on these in the demonstration.

6. Feedback and lessons learned

All of the users gave positive feedback on the SCRAM method (average 6-6.5 on 1-7 point scale). They thought it was helpful to see alternative designs and the demonstration was useful for discussing requirements. They all felt they had been given the opportunity to say what they wanted about the designs.

1. Add as much functionality as time allows. The use of the concept demonstrators was effective at generating discussion regarding requirements. Users could clearly visualise how each task might be performed using the alternative designs; however, the inability to interact directly may have hindered their reaction. This was not possible as the session already took a long time (1.5 hours). There is no easy way out of this dilemma. Partially functional concept demonstrators can convey an extensive design vision in a short space of time, but they do so by sacrificing hands-on experience.

2. Focus scenarios. Generating scenarios for each issue and concept demonstrator sub-script was useful for users to understand system usage in a task context. We concentrated on straightforward and simple scenarios and did not tackle any complex scenarios. However, given the diversity of user backgrounds it was impossible to contextualise the scenario for each user’s experience, so we suspect that the users may not have seen the scenario as relevant to their current work. One possibility is to make use of analogies to anchor future usage scenarios in the users’ current experience, e.g. searching the Web, search in a library.

3. Discourage designer dominance. The requirements engineers tended to take up too much of the floor space in the session dialogues. This may have been caused by overloading the session with too many issues that needed explaining, or by the poor scenario focus that made the users less responsive. Whatever the cause, user participation could have been improved by enforcing less conversation from the designers.

4. Use simple representations. Although the users gave positive feedback on the use of design rationale diagrams to compare different interface designs, we observed that they found the QOC notation hard to use. We have therefore changed the representation to tables in subsequent studies. The simple requirements map on the whiteboard helped users actively participate in the brainstorming session for requirements summary.

5. Emphasise priorities and trade offs. A number of trade offs were considered by the users during the requirements sessions. Overall the method produced too many requirements as the users tended to give a default ‘both’ reply when presented with options. As most projects have limited resources, trade off judgements need to be encouraged more forcefully. One approach is to give users a set amount of fake currency and then ask them to buy the options they really want.

6. Prevent session overloading. The session took 1.5 hours to produce the requirements and covered a large range of issues. This meant the demonstration script took a long time to complete and the users had difficulty relating this to their work context and maintaining concentration. Current questions are structured according to keypoints in the demonstration scenario, however, this is inflexible. Filtering may be advisable to avoid refining the more obvious and predictable requirements that are better captured by interviews or questionnaires.

7. Mimic complex functionality to show its limitations. A limitation of concept demonstrators is that the user interface shows an idealised story of how a function should operate. This may mislead users into assuming that the system has capabilities that the designer cannot deliver. This was apparent in the brokering functionality which was invisible at the user interface. It is difficult for users to assess utility of functions unless they can experience their accuracy and how they fit in with their decision making. One way of exploring limitations might be to use Wizard of Oz techniques with a human operator executing the functions.

In spite of these problems, the overall experience was positive as demonstrated by the users’ feedback, and a large number of useful requirements were captured.

7. Discussion

The SCRAM method was successful at eliciting precise and clear user requirements. Empirical studies of RE methods inevitably suffer from limitations because it is difficult to establish how many essential requirements were captured and how many were missed until the implemented system is evaluated. However, the
empirical evidence does suggest that the method was effective as many requirements elaborations were triggered by reference to the concept demonstrator screens, and the design rationale.

Compared with our previous experience in using SCRAM [8, 9], the major differences were the lower volume of user participation in this study. There are two possible reasons for this. First we suspect the task focus and the scenarios were more immediate in the previous studies. Also the users fulfilled roles that encouraged extroversion and self expression resulting in a larger volume of detailed requirements and much more domain knowledge. Moreover there was no problem of designer dominance, indeed the converse was the case as users dominated the conversation.

The SCRAM method fits in the tradition of scenario based RE approaches, such as the Inquiry Cycle. SCRAM goes further than the Inquiry Cycle in integrating requirements techniques with the use of concept demonstrators, design rationale and questioning techniques and provides advice on how to structure scenario based requirements sessions that may augment commercial RE toolbox methods such as DSDM [2]. In conclusion, so far the method seems to be effective but improvements need to be made. We are increasing its effectiveness by adding more analytic heuristics for checking requirements specifications against scenarios [9] and providing tools to support walkthrough style analysis [4].

Acknowledgements
The authors would like to thank partners on the Multimedia Broker project (SISU, PTRC, LIBER, & IMT) for their cooperation and help in this study. The research was partially supported by the European Union in the Telematics Project Multimedia Broker and the Esprit Long term research project CREWS.

References