A Service Elicitation Process for Crisis Management Technologies

Appelman J.H.
Delft University of Technology
j.h.appelman@tbm.tudelft.nl

Kar, E.A.M. van de
Delft University of Technology
e.a.m.vandekar@tbm.tudelft.nl

Hengst-Bruggeling, M.
Delft University of Technology & TNO
m.den.hengst@tbm.tudelft.nl

Ven, J. van de
TNO & DECIS-lab
josine.vandeven@decis.nl

Burghardt, P.
DECIS-lab
paul.burghardt@decis.nl

ABSTRACT

New information technological applications, that aim to support better professional responses to incidents and crises' are being developed at an increasing rate. We observe in almost any disaster that actions by civilians also contribute substantially to alleviation of the effects of a crisis. We are largely unaware what crisis management services would be of use to civilians under these particular circumstances; just as it is difficult for users to specify requirements for services based on novel technologies. These insights guided the design of a service elicitation approach that can be repeatedly used for different user groups. In this paper we introduce a design methodology and discuss outcomes of a first test-session. We reflect in the conclusions on improvements to the design.

Keywords

GSS, Crisis Management, Requirements engineering

1. INTRODUCTION

It is increasingly acknowledged that crisis management and especially the rapid achievement of situational awareness and mitigation measures is best supported by a decentralized service oriented information architecture (see also Turoff et al., 2004, Burghardt, 2004, Ooms, 2004). Such architectures imply that many different actors: victims and by-standers, and professionals at different decision making levels can be supported by personalized services offered on a range of tools. These can range from full-fledged high-tech control and command centers to a simple PDA or cell-phone.

There are a host of methodologies that support the engineering of information systems. Some of these are highly codified sequenced steps of tasks that programmers or groups of programmers have to take others are more loosely defined methodologies that aim to support the competencies of programmers and leave relatively much room for autonomous decisions(Martin 1991). What these methodologies do not clearly specify is how a group of people involved in building such a system could collaborate to produce well grounded requirements that a system should be able to meet. Collaborative requirements elicitation is plagued by the same problems as any type of group work: dominant or shy participants, misunderstanding, free-riding, lack of consensus, poorly defined goals etc.

(Nunamaker et al. 1997). Our paper addresses the problem of a lack of specificity on how to execute a collaboration process by designing a standard process for requirement elicitation. Such a design specifies what activities people on a group should perform to come to the joint production of requirements for technologies specifically designed for the support of crisis management activities.

To do this we used thinkLets, a concept central to the field of collaboration engineering (CE). This is a design approach that aims to develop technology-independent process building blocks for collaborative activities. We show in the second section why we think requirements elicitation may benefit from the use of GSS and a CE inspired process design. In section 3 we introduce the case and in section 4 we use both streams of theory to present a design of a repeatable process for user requirements elicitation. We evaluated this repeatable process once in its current form. The findings of this first trial run are presented in section 5. We end with conclusions and a discussion of future research directions in section 6.

2. BACKGROUND

When considering the development of software systems it is apparent that there is a shift toward service oriented architectures. The services that the system produces define the function and boundaries of the technical components that the system comprises of. Put simply, what the users want the system to do determines how it operates. Of course there are limits to such flexibility; an information system cannot operate completely 'demand-driven'. But if such a system offers a range of services that users can select depending on their current needs it might function flexibly enough and thus be of use in a crisis for civilians, volunteers as well as professionals involved in its mitigation. We therefore use techniques from the requirements elicitation literature to define services that inform how components or tools can be used to empower users. In the sense that they are able to use information to their own or others benefit in a crisis situation.

The primary measure of success of a designed system, being it a product, a software system, or a service system, is the degree to which it meets the purpose for which it was intended. Requirements engineering is the process of discovering that purpose, by identifying stakeholders and their needs. Prior research showed the importance of requirements engineering (Standish Group 1995) at least for software systems: more than half of the 8000 software projects studied were compromised from the outset by shortcomings in the requirements definition process. Improvements in requirements engineering could therefore substantially reduce the risks.

Requirements engineering is a highly collaborative process and one of the first steps is requirements elicitation: understanding the stakeholders and capturing their requirements.

For eliciting requirements, many different techniques have been used. The techniques can be summarized into several categories (Nuseibeh and Easterbrook 2000):

- Traditional techniques include questionnaires, surveys, and interviews with the individual stakeholders.
- Group elicitation techniques aim to foster stakeholder agreement and buy-in, while exploiting team dynamics to elicit a richer understanding of needs. Techniques that can be mentioned here are brainstorming, focus groups, and RAD/JAD[1] workshops.
- Prototyping can be used when there is a great deal of uncertainty or when early feedback from stakeholders is required (Davis 1992).
- Model-driven techniques provide a specific model of the type of information to be gathered and use this model to drive the elicitation process.
- Cognitive techniques include a series of techniques originally developed for knowledge acquisition, such as thinking aloud and card sorting (Shaw and Gaines 1996).
- Contextual techniques are an alternative to both traditional and cognitive techniques. These include techniques such as participant observation and conversation analysis (Goguen and Linde 1993).

[1] RAD = Rapid Application Development; JAD = Joint Application Development

Each technique has its advantages and disadvantages and there is no best technique. The combination of several techniques is usually very productive. When taking a closer look at the categories above, we can distinguish between group techniques and individual techniques. Prototyping, model-driven techniques, cognitive and contextual techniques can all be used in a group setting or in an individual setting. Although eliciting requirements does not need a group setting, it can benefit from it (Langford and McDonagh 2003). It is more time efficient, it has a higher flexibility (depending on the group, the steps can be adapted along the way), the output is easy to understand, a higher richness of the information is gained, and information that would have stayed hidden tends to emerge. Some of the disadvantages are long preparation time, a group setting is more difficult to manage, participants could be very dominating or silent on the other hand, recruiting participants takes a lot of time and effort, the analysis of the results afterwards takes a lot of time, and the participants are not really representative for a larger group (Langford and McDonagh 2003). Some of these disadvantages can be resolved by using Group Support Systems. The next subsection will pay attention to this.

The combination of techniques to be used and steps to be taken to elicit requirements is highly dependent on the situation at hand. The stakeholders involved are an important indicator for this. Requirements elicitation involves 'capturing' the requirements of different stakeholders, such as customers, users, and developers. Users play a central role in the elicitation process (Nuseibeh and Easterbrook 2000). User requirements define what should be developed. The requirements of the other stakeholders mainly define the constraints to what has to be developed. In this paper, we focus on eliciting user requirements and thus potential users.

Users are not a homogenous group; different types of users can usually be identified (Sharp et al. 1999). Selection of participants for the requirements elicitation process is therefore important. The participants should be chosen carefully through purposive sampling (as opposed to random sampling for surveys) – by selecting participants belonging to specific user groups (Morgan 1998). The participants need to be reasonably knowledgeable about the topic and should be interested in talking about it. Ideally, the groups should not include too many different types of people (Maguire 2003), whilst a certain amount of diversity may be useful to encourage contrasting opinions (Bruseberg and McDonagh 2003). Participants need to be comfortable when talking to each other and should share a similar background to encourage a common understanding of more detailed insights. Preferably, requirements elicitation groups consist of participants who are not too familiar with each other: the more diverse the views that are represented, the more reliable or robust the results become. Some familiarity between participants may help to 'break the ice', but over-familiarity may adversely affect the synergy of the group (Morgan 1998). Furthermore, the participants do not take the role of designers. They do not have to find solutions, but provide suggestions for the designers.

It is often the case that users find it difficult to articulate their requirements and to be creative. It is very unlikely that users can tell what they want in the future. They can tell what they want now, and the requirements elicitation techniques should guide users from what they want now into future usage scenario's (Ireland and Johnson 1995, Johnson 1992). The users should talk through their daily experiences to end up with usage scenarios. Another way to support the users in thinking into the future is to demonstrate a possible future (Maguire 2003, Schneider and Winters 1998). The demonstration allows the users to understand the anticipated future more clearly. Disadvantage of demonstration is that the users might get unrealistic expectations, for example, because of lack of detail. Besides techniques to support users in articulating their requirements, techniques to increase the creativity of users can be used as well. A number of techniques should be taken into account, each with its distinct benefits and drawbacks (Bruseberg and McDonagh 2003): visual evaluation, product handling, mini-user trial, creating 3D-forms, users drawing their ultimate product, and nominal group techniques are examples of such techniques.

This means that for eliciting requirements, we can benefit from a group setting. But we have to choose users who will participate in the group session carefully and we need techniques to support creativity in order to help users explore possible future use of new technologies. More importantly, we would like to have an approach that would allow for a repetition of the same steps, to come to requirements for crisis management services, with different groups so the results can be pooled and compared. Collaboration engineering is a way to design a requirements elicitation process that meets these criteria (Briggs et al. 2003).

To pursue these ends collaboration engineering has produced a concept labeled a ThinkLet. "*A thinkLet is a named, tightly scripted, process for creating a single repeatable predictable pattern of collaboration among people working together towards a goal*". (Kolfschoten et.al. 2004) Patterns of collaboration are not visible in the user interface of the computer but are the result of a set of bounded actions each individual performs. ThinkLets aim to consistently describe this set of bounded actions. We used the thinkLets and the patterns of collaboration to design a repeatable process. This is possible because each thinkLet is described in the same way. For each thinkLet: an action, rule(-s), role, capability and parameters are specified. There are 5 patterns of collaboration: divergence, convergence, organizing, evaluating and consensus building. All thinkLets, there are about 70 now, fall under one of those 5 patterns. After assigning a pattern to a particular part of a group process the appropriate thinkLet is selected.

To support the obtainment of our research goals, the design and execution of a service elicitation process for new crisis response technologies, we employed a Group Support System (GSS) and used thinkLets to model the group process.

3. SYSTEMS FOR CRISIS MANAGEMENT

In 2003 researchers of various research institutes started a project to explore the basic concepts for large-scale systems of systems in chaotic open world environments. The project is named COMBINED Systems and this stands for "**C**haotic **O**pen world **M**ulti-agent **B**ased **I**ntelligent **Ne**twork **D**ecision support Systems". In this project it is imagined that future systems consist of both human actors and artificial agents that work together to achieve their common goals under chaotic circumstances.

The basic vision is to develop multi-agent systems that can cope with these circumstances by creating scalable ad hoc networks of actors and agents. Communities of actors and agents Observe the situation they are in, Orient themselves, make Decisions and take Action (OODA). To illustrate its results the COMBINED program is working on a crisis scenario in the Rotterdam International Harbor. Two ships collide causing poisonous gasses to spread over the Erasmus university area. An evacuation becomes necessary causing acute crisis and traffic management problems.[2]

We selected the following systems of the Combined program for this session.

Icon Map : a map with icons indicating scenes at a crisis. These icons can be placed by bystanders, for example when a bystander sees a fire, she clicks the fire-icon and then clicks on the map where the fire was seen. Because more people will do this, there will be an accurate report of the place of the fire due to the combination of all the information. A bystander will be able to plan a route out of the area thereby avoiding dangerous places.

Distributed Perception Network : sensors in the environment of the crisis can provide information about for example gasses in the area. Sensors provide input to a Bayesian network and this can send information to the people about a certain gas being detected. However it can also request information from the people in the crisis area. For example, it can sent a question to all mobile phones in a specific area whether people see other people with breathing problems (or whether they experience themselves breathing problems), yes or no? The number of reactions will provide information about the problems in that area.

First Aid Management (to help injured people): this system records all the information of the injured people in the (crisis) area, does the triage and supports the person providing first aid to take a decision which person to help first. When more people provide first aid in the same area, the tool helps to divide the work among them.

Icon Communication: People that speak different languages can communicate with each other via an icon-tool. A person can formulate short sentences choosing different icons from a set. This way foreign people can report the observation of an incident fairly easy, but they can also communicate about their own situation with for example the First aid provider.

Ant-based routing system : this system is based on the idea that each person (or vehicle) can contribute information to a routing system. Because I went from place A to place B, I can provide information about the distance I traveled and the time it took me. This information is added to the system, people that want to travel the same route can compare the different ways to travel from A to B, and take for example the shortest route.

The research issue presented in this paper concerns user requirements in crises situations and the match of these with the above mentioned systems. We identified four groups of users: Civilians/victims, first responders (professionals),

[2] www.combined.decis.nl

tactical teams, strategic teams. We executed a session for the first groups; the design of which is presented in the next section.

4. DESIGN OF REQUIREMENTS ELICITATION PROCESS FOR CRISIS MANAGEMENT

In this section we will the process design of a GSS session and shortly describe the different phases. The complete process is depicted in figure 1.

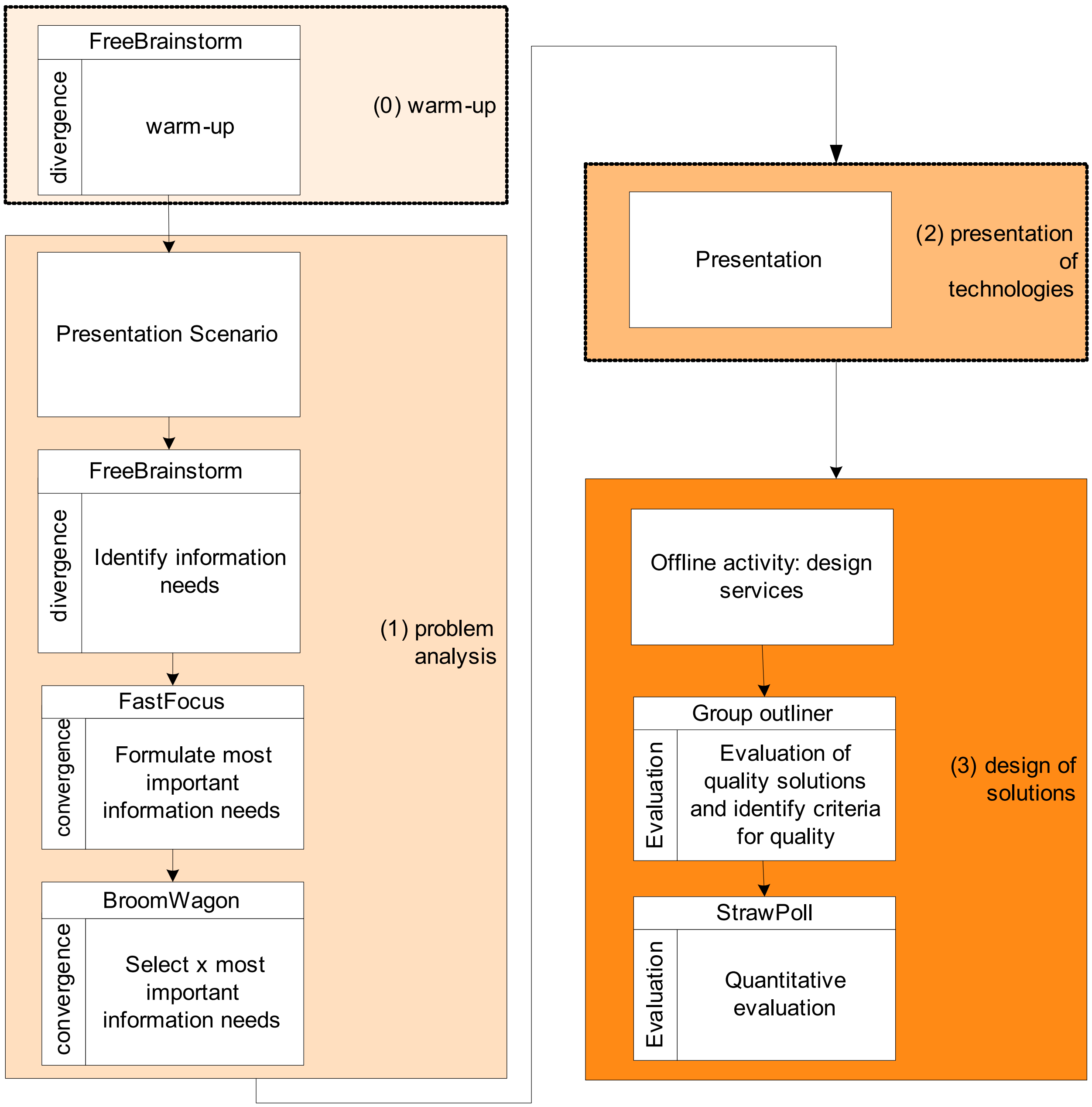

Figure 1. Thinklet based user elicitation process

In the first phase the context is presented and information needs are elicited: imagine you are in that situation: what are your questions, what are your needs? These are the prompts that trigger a brainstorm. After this step the participants are asked to select and give, in a round robin fashion the most important needs, in this convergence activity the redundancy the first brainstorm delivered is reduced. Now we can assume that the participants share a group memory, they have produced a shared definition of the most important needs. The list with most important information needs is then evaluated and the participants gain more focus by means of an assignment where the participants can select again the most important items.

In the second phase a selection of technologies deemed appropriate for that particular user group is presented. This might be extended by a demonstration.

In phase 3 the most important needs will then be used to define what services, linked to already existing technologies could be interesting to integrate in a crisis management system. In an offline activity sub-groups design a service for an information need. Next, they present the service idea to the group that evaluates the service. The evaluation activities produce a ranking via quantification and the designers of the tools also get qualitative feedback that is put into the GSS by the participants.

5. FINDINGS

The presented scenario was a crisis scenario in the Rotterdam International Harbor as used in the Combined project by the researchers who develop the systems. The students were asked to imagine being in a session at the Erasmus university when it becomes clear that a disaster is happing: there are sirens, strange gas smell, panic.

Figure 2. Crisis management scenario

The brainstorm in information needs delivered more than 60 different items related to needs of civilians/victims in this crisis situation. After convergence the group was left with 24 clearly understood information needs. They voted on these 24 needs so that we got overview of the most important ones. The top 5 is:

- If I leave, where do I go to?
- In what way(-s) can I leave?

- Do I have time? (how much)
- Do I need medical treatment?
- What Happened?

After an evaluation round the participants designed services and, subsequently listened to each others' presentations.

This presentation was followed by an offline activity in which the GSS was not used. Based on the questions and with the presentations as background, different services were further developed. With transparencies and pencils the participants designed a service by describing the context, the desired input and output of the service. There were four groups and the facilitator decided to leave out the question 'Do I have time'.

Each of the groups presented their designs and during their presentations the other participants put their comments in the GSS. This saved a lot of time. Urgent remarks could be discussed. It was not planned but some of the researchers were working in the next room and joined the presentations and the discussion. This way the researchers got direct feedback from potential users.

The solutions presented were aimed at receiving rather than giving information. The fact that they were mainly triggered to assume the role of victims led them to questions. These questions rendered the following services:

1. If I leave where can/should I go?

   A service for a Japanese tourist, that does not know any English and is confronted by a gas cloud. The tourist smells a chemical odor and develops stinging eyes. Icons allow him to communicate and he sends a message describing his physical health, the situation and other relevant information from his perspective, like the fact that he is Japanese. By means of a first aid support module and an Icon map linked to the routing system the tourist receives information on what to do.

2. How do I get out of here?

   The context for this service is a situation where students in Rotterdam take part in a GDR session. Again, a penetrating odor spreads, the air outside blackens, they get respiratory problems. They use their PDA to ask in the emergency module where they should go. The answer is a map with a route specified optically.

3. Do I need medical treatment?

   A poisonous gas leak is detected. Inhabitants of the area are sunbathing in the garden and they ask what they should do? They receive a location specific advice and advice on how to react to certain symptoms that might develop.

4. What happened? What is going on?

> Students are in a building and the air alarm is sounded. Some students panic others request information. They switch on their PEA (personal electronic assistant) in the emergency mode. Their location is certified and by means of icon communication the request information. They receive a color coded map of the area, icons that describe the situation, and icon based map and a spoken message. They find that the alarm is a false one while students that do not use the PEA jump, panic stricken, out of the window.

The comments provided via the GSS were later sorted and analyzed. Besides valuable feedback on the proposed services and the possibilities and impossibilities of actual implementation more fundamental questions popped up. These are worthwhile to mention because they inform the development of crisis management or response systems. The most important comments all revolved around the question: "what if the technology fails?". Participants were adamant in stating that they would never accept a new application in this area if they could not assured that it was extremely robust and relatively simple to use. These comments link very well to recognition primed decision making theory. We will return to these comments in the conclusion section.

We talked to participants afterwards about why they scored as they did and we reflect on their remarks in the last section.

6. CONCLUSIONS AND REFLECTIONS

We designed a session for users to design services based on new technologies that can be executed with different target groups. In this paper we present our experiences with the first test group.

We conclude that the first session was successful in meeting the objective of the elicitation of services. The design did what we hoped it would do, however we experienced some limitations. First the information needs can be considered obvious. Nevertheless, to verify this 'obviousness' more sessions with participants with different characteristics are needed. This is possible because we now have a repeatable design. A practical issue is that the organization of these sessions should be done by members of these groups instead of academics. E.g. a first aid group could participate in one of these sessions organized by their instructor. Reasons for this is that focus groups should share the same background, are reasonably knowledgeable, interested in talking about it and comfortable talking to each other. Secondly, we found that the focus of the session was squarely on the first minutes of a crisis. We know that it is more than likely that the demand for services will change over time. We therefore conclude that a more elaborate scenario for different phases and different groups in a crisis, like: detection, building situational awareness, developing mitigation measures, mitigation and evaluation, delivers a more diverse set of services.

Concerning changes to the design or methodology, we can say that the design held up very well. No big changes are considered necessary where it concerns the actual elicitation process. Improvements must mainly be sought in the area of representation. When we succeed in attracting groups of participants from different backgrounds the reliability of the results will increase. More executed sessions will eventually also deliver a further statistical validation. It seems worthwhile to explore distributed execution of the session because it is increasingly common that (internationally) distributed development teams need to collaborate. We are aware that there are already existent approaches but we also put forward, in the introduction, that these suffer from either a highly structured drawn-out process that consumes considerable time of facilitators and participants or rather vague instruction at the group level. We have designed the process as 'light' as possible and, in this way, we hope to have contributed to the need for more agile methods for requirements elicitation.

Secondly, we want to make a remark on the designed services. The participants combined the technologies nicely into a scenario. But this does not answer the question whether these are the supporting technologies they want (most). At the same time, it is impossible to ask them what technologies or services they want if they have no idea what is available. Therefore we recommend an incremental design process in which prototypes are built and used and tested by users in every stage of development.

The relevance for the field of this study is that we showed a relatively easy way for system developers to get into contact with the potential end-users and their ideas. It might be too ambitious to get clear user requirements with these kinds of sessions but it surely provides a good opportunity to get confirmation from the users on the direction.

The relevance for science is that it confirms how difficult it is to get user requirements for systems based on advanced technologies. The challenge is to find the right way of presenting technologies that are under development by researchers. On the one hand you want to show your achievements and get feedback, on the other hand you want to find the wants and needs of users independently from technology in development in labs.

Furthermore, all users stressed that they were concerned with robustness of technologies. The first and most important requirement is that a system or parts of the system should always work. This leads us to the conclusion that the principle of a fail-safe mechanism should be incorporated in the design of crisis management systems. We continue with our collaborative engineering research in this user requirement elicitation field for systems based on advanced technologies.

ACKNOWLEDGMENTS

We thank Sandra van der Hulst for her technical assistance during the trial-run.

REFERENCES

1. Appelman, J.H. & Van Driel, J. (2005) Crisis-response in the Port of Rotterdam: can we do without a facilitator in distributed settings? Proceedings of the 38th Hawaii International Conference on System Sciences. Los Alamitos: IEEE Computer Society.

2. Briggs, R.O., M. Adkins., D. Mittleman. J. Kruse, S. Miller, J. F. Nunamaker, Jr.., (1998-99) "A Technology Transition Model Derived From Qualitative Field Investigation of GSS Use Aboard the U.S.S. CORONADO", Journal of Management Information Systems, 15 (3), 151-196.

3. Briggs, R.O., de Vreede, G.J. (2001) "ThinkLets: Building Blocks for Concerted Collaboration", GroupSystems.com, Tucson.

4. Briggs, R.O., G.J. de Vreede, J.F. Nunamaker, Tobey D.H. (2001) ThinkLets: Achieving Predictable, Repeatable Patterns of Group Interaction with Group Support Systems (GSS), Proceedings of the 34th Hawaii International Conference on System Sciences.

5. Briggs, R.O., G.J. de Vreede, J.F. Nunamaker, Jr. (2003) "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems", Journal of Management Information Systems 19 (4), 31-63.

6. Bruseberg, A. and McDonagh, D. (2003)'Organising and Conducting a Focus Group: The Logistics', in: Focus Groups, Supporting Effective Product Development, J. Langford and D. McDonagh (eds.), Taylor & Francis, London, 21-45.

7. Burghardt, P. (2004). Combined Systems: The Combined Systems Point of View. In: ISCRAM 2004, Brussels.

8. Davis, A. (1992) 'Operational Prototyping: A New Development Approach', in: Software, vol. 9, no. 5, 70-78.

9. Endsley, M. R., Bolté, B. and Jones, D.G. (2003). Designing for Situation Awareness: An Approach to User-Centred Design. London: Taylor & Francis.

10. Fjermestad, J., Hiltz S.R. (1999) “An assessment of group support systems experimental research: methodology and results”, Journal of Management Information Systems 15 (3), 1999, 7-149.

11. Fjermestad, J., Hiltz S.R. (2001) “Group Support Systems: A Descriptive Evaluation of Case and Field Studies”, Journal of Management Information Systems 17 (3), 115-160.

12. Goguen, J. and Linde C. (1993), ‘Techniques for Requirements Elicitation’, in: 1st IEEE International Symposium on Requirements Engineering, San Diego, USA, 152-164.

13. Harder, R.J., H. Higley, (2004) “Application of ThinkLets to Team Cognitive Task Analysis”, Proceedings of the 37th Hawaiian International Conference on System Sciences, Los Alamitos: IEEE Computer Society Press.

14. Ireland, C. and Johnson, B. (1995) ‘Exploring the Future in the Present’, in: Design Management Journal, no. 6, p57-64.

15. Johnson, P. (1992) Human-Computer Interaction: Psychology, Task Analysis and Software Engineering, McGraw-Hill.

16. Kolfschoten, G.L., J.H. Appelman, R.O. Briggs, de Vreede, G.J. (2004) “Recurring patterns of Facilitation Interventions in GSS Sessions”, Proceedings of the 37th Hawaiian International Conference on System Sciences, Los Alamitos: IEEE Computer Society Press.

17. Langford, J. and D. McDonagh (2003) ‘Introduction on Focus Groups’, in: Focus Groups, Supporting Effective Product Development, J. Langford and D. McDonagh (eds.), Taylor & Francis, London, 1-18.

18. Maguire, M. (2003) ‘The Use of Focus Groups for User Requirement Analysis’, in: Focus Groups, Supporting Effective Product Development, J. Langford and D. McDonagh (eds.), Taylor & Francis, London, 73-96.

19. Martin J. (1991) Rapid Application Development, Macmillan Publishing, New York

20. Morgan, D.L. (1998) Planning Focus Groups, Sage Publication, Thousand Oaks.

21. Niederman, F., C.M. Beise, and Beranek P.M. (1996) “Issues and Concerns about Computer-Supported Meetings: The Facilitator’s Perspective”, MIS Quarterly 20 (1), 1-22.

22. Nuseibeh, B. and Easterbrook, S. (2000)‘Requirements Engineering: A Roadmap’, Imperial College, London, United Kingdom, www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf.

23. Ooms, A.H.J. (2004). Organization Awareness in Crisis Management: Dynamic organigrams for more effective disaster response, Proceedings ISCRAM 2004, 63-68. Brussels.

24. Schneider, G. and Winters, J. (1998) Applying Use Cases: A Practical Guide, Addison-Wesley.

25. Sharp, H., A. Finkelstein, and Galal, G. (1999) ‘Stakeholder Identification in the Requirements Engineering Process’, in: Workshop on Requirements Engineering Processes, Italy, 387-391.

26. Shaw, M. and Gaines B. (1996) ‘Requirements Acquisition’, in: Software Engineering Journal, vol. 11, no. 3, 149-165.

27. Standish Group (1995) CHAOS Report: Application Project and Failure.

28. Turoff, M., Chumer, M., Van de Walle, B. & Yao, X. (2004). The Design of a Dynamic Emergency Response Management Information System (DERMIS). The Journal of Information Technology Theory and Application, 5, 1-35.