Crisis Management Training: Techniques for Eliciting and Describing Requirements and Early Designs across Different Incident Types

Ebba Thora Hvannberg and Jan Rudinsky

University of Iceland, Dunhaga 5, 107 Reykjavik, Iceland
ebba@hi.is, janr@hi.is

Abstract. Crises occur seldom, but when they occur they have high impact on the enclosing organization and its stakeholders. Examples are plane crashes, train incidents and bomb threats, but the types of crises are virtually endless. We report on research of early phases of the development of a crisis management training simulator, with the goal of understanding different representations and transitions between steps of a development process. The focus of the research study was on how the different representations did align with a given process model and how these representations lent themselves to a consolidation activity. The results were that consolidation across data sources starts early during the understanding phase and that stakeholders like to validate abstract models. The consolidated conceptual models mostly addressed work and strategies. No formal attempt was to consolidate across management and organization structures.

Keywords: Crisis management, software systems, consolidation, process model, representations.

1 Introduction

Fortunately, crises occur seldom, but when they occur they have high impact on the enclosing organization and its stakeholders. Examples are plane crashes, train incidents and bomb threats, but the types of crises are virtually endless [1]. Crisis management is the core of response to such serious accidents. In this context, crisis management is about organizing work, resources and information in a secure and timely manner with the aim of rescuing people and ensuring their best health conditions. The challenges of crisis management are manifold, but include variability of available resources that are sometimes scarce and sometimes plentiful. Examples of those resources are rescue equipment, transportation, medical support and manpower. Additionally, the surroundings and contexts of work, such as in mountains or at sea in different weather conditions, vary greatly. One challenge of the problem domain is that there is a high demand for synchronization and decision making among

C. Stephanidis (Ed.): Universal Access in HCI, Part IV, HCII 2011, LNCS 6768, pp. 225–234, 2011.

teams with members in different organizations. Furthermore, time is a critical component, distinguishing between life and death.

To meet these demands of crisis management, organizations install a scheme of activities, command and flow of information. A systematic approach, relating to procedures and plans, and a systemic approach affecting the whole organization, have to be addressed [2]. Training a system of crisis management is typically performed in large exercises where an accident and its response are simulated but in a real environment. In such cases, a hypothetical training scenario is created which participants can act out [3]. Although life training is successful and results in trained responders and commanders, such training is expensive. Therefore, a software system to simulate training is under development.

A simulator to train crisis management has to take into account various aspects of operational and training concerns. It contains multidisciplinary perspectives drawing from psychology, technology, social-political and technological-structural views, including procedures, policies, practices and routines, in addition to the traditional view of technology as a machine or a tool [1].

A crucial part of designing such a simulator for training of crisis management is domain analysis. The objective is to learn about the different actors, their work, their interaction and information flow between them. Typically, different methods are applied to obtain data during domain analysis, such as interviews, empirical observations and surveys [4].

There seems to be a plethora of methods to analyse domains, but there is limited knowledge on how the output of elicitation is abstracted or consolidated, and how the output is formed into design. We know that data is abstracted, but there is a pendulum between using concrete representations such as scenarios to more abstract and formal ones such as task models [5]. It is essential to understand these activities, since in the transition from analysis to design it has to be checked whether all the scenarios of use are accommodated adequately by the design [6].

A domain analysis unavoidably requires consolidation of data across sources, workers, customers and contexts. There are several motivations for such consolidation. First, the aim of consolidation can be to make the models more detailed, by connecting data from e.g. different sources, but it is also to make them more abstract, thus allowing the designer and the developer to see the common factors in the software system. A second motivation is to tailor it to a broader market, resulting in a valuable product for its owner. Third, the recent increased interest in domain specific languages is a motivation to describe domain knowledge in generic models. This can encourage building a common platform for crises management simulation that will allow teams to share lessons learned, throughout their organization, to improve planning [2].

In this paper, we report on research of early phases of the development of a crisis management training simulator, with the goal of understanding different representations and transitions between steps of a development process. The focus of the research study was on how the different representations did align with a given process model and how these representations lent themselves to a consolidation activity.

2 A Process Model

The development team of the training simulator did not use a particular development process model as a reference. For the purpose of a reference process in this research study, we selected a scenario driven process of human-centered interactive system design, described by Benyon [7]. In it, scenarios are suggested and meant to help developers ease into the abstractions by allowing them to write concrete steps, which described tasks, contexts and users roles. In the scenario-driven process, Benyon shows developers how they first can develop stories, scenarios and personas and then write abstract requirements that make up a conceptual model. Once the requirements have been stated, a concrete scenario showing how the system is used can be developed. This can also be visualized with prototypes, mostly of low fidelity. Abstracting again from the visualization is a formalization of the design with use cases, and the final step is to develop an interactive system. The initial phase of the process is carried out to understand the problem domain, i.e. activities, tasks, roles and context. When several of these have been collected into a scenario corpus, it is time to abstract them, via generalization or aggregation, into a conceptual model. This completes the *understanding phase*. For each conceptual model, there are possibly many concrete representations. The responsibility of the *envisioning phase* is to recommend an appropriate concrete design, which is based on requirements and design constraints, such as the technology applied. The final step is the *implementation phase*. This research covered only the understanding and the envisioning phases. It should be noted that the phases are not serial, and can overlap.

3 Understanding Tasks, People and Contexts

Extensive gathering of requirements took place in the initial phases of development for the crisis management training simulator. Three sites were visited in three countries. Two of the sites are airports and their reference scenarios are an aircraft incident and a bomb threat, but the third one is a railway company with a train incident as a reference scenario. The instruments, which have been applied during requirements elicitation, and representations that are output of the process's activities are shown in **Table 1**. Abstract and concrete representations were created during the understanding phase. What is common to them is that they were representations of individual incident types of crisis, such as a bomb threat or an aircraft incident. There is little consolidation across incident types during this initial phase of understanding work. Four work processes were identified, Alert, Rescue, Transport and Triaging of casualties and Security. In addition, there is a separate Communication process describing communication and coordination between the different work processes. For clarity, communication is also included in the work processes.

There were attempts to divide the domain into facets or aspects. Work is separated from training, e.g. in the user requirements and the training requirements. There was an analysis of worker competencies, mainly targeted for training, in the requirements documents. Furthermore, there is an analysis of management and organization, like in

Table 1. Understanding people, tasks, contexts and requirements

Goal	Phase	Elicitation instrument	Representations	Concrete / Abstract
To gather initial information on work and training.	Preliminary	Informal meetings with stakeholders, presentations of stakeholders on work.	Written scenarios describing a series of timed events of work.	Concrete
		Manuals on work. Observations of the working environment.	Time series that show interactions between roles.	Roles are abstracted, times are concrete.
			Mind map of activities of different roles of actors.	Roles and activities are abstracted.
			Contextual Design Work models of individual customers.	Sequences of activities are abstracted, roles too, Physical and Artifact models.
To gather information on work and training and validation of representations of the previous phase.	Understanding	Observation of live training and table top exercise. Validation of work models with crisis managers.	Contextual Design Work models of individual customers.	Sequences of activities, control and roles are abstracted. Physical and Artifact models.
To gather data on stakeholders' priorities of system's capacities.	Understanding	Questionnaire on training competencies.	Training competencies required (hierarchy).	Competencies are abstracted.
	Understanding	Questionnaire on user requirements.	User requirements as text.	Abstract
Validation	Understanding	Interviews	Refinement of scenarios and models.	

Table 1. (*Cont.*)

Validate whether the stated training requirements are unclear, incomplete, ambiguous or contradictory.	Understanding	Translate textual scenarios into visual form.	Story boards of scenarios	Concrete scenario per site.

Björner's six facets: intrinsics, support technology, management and organization, rules, regulations and scripts and human behavior [8], and Cognitive Modeling Framework[9]. Emphasis on strategies could only be found in the Contextual design sequence models, which included decisions and decision support. There was little emphasis on information or data modeling, except in the Contextual inquiry artifact model, which had references to data forms used during work. The Contextual design flow model, which would have provided more information on data, was not created due to lack of detailed information. Technology support was analysed in the Physical and Artefact models. These were mostly from the physical world, and not from the world of information technologies, except for communication technology. Because of the importance of communication, including coordination in the domain, analysts decided to include the communication between roles or organizational units in a Communication model and in the individual sequence models, i.e. the Rescue work model, the Alert work model, the Transport work model and the Security work model. Communication is either modeled as a meta-level construct of a sequence model, similar to trigger, intent, decision making, decision support or performance or as a separate sequence.

Some lessons were learned from this first phase. The different representations show that analysts frequently go back and forth between abstract and concrete presentations. Thus, some consolidation across workers and data sources starts even at an early stage. Some stakeholders had difficulty validating the concrete written scenarios, but preferred to look at abstract models. The reason was that they did not trust the scenarios to cover each activity and coordination between them. For them, the abstractions were more accessible and easier to validate, because of the possible different alternatives even within an incident type, e.g. an aircraft incident. The reason may be that the crisis managers that were part of the stakeholder group were used to work with abstract procedures such as parts of plans, organization charts, role specifications etc. A third observation we made was that the representations show various abstractions, e.g. hierarchical structures, mind maps and contextual inquiry work models. This diversity of the representations reflects the developers' experience and practices. All of those are informal in nature. Might it mean that developers have not yet found a formalism which meets their demand?

4 From the Concrete to the Conceptual

Concrete representations, whether they are essential use cases, stories or scenarios, lend themselves to easy readability for users [10]. They are relatively well scoped and describe a series of action in a certain context. However, they rarely describe more difficult constructs such as alternative paths, iterations, parallel tasks, or relationship between data, tasks or roles, except as concrete examples. Besides, they may not lend themselves to be a model after which a system can be developed. Thus, we need abstract representations. Abstraction involves generalization and aggregation over different stories or scenarios. Generalization is about finding common tasks, objects and roles across different scenarios, but aggregation ensures inclusion even if components are only part of some stories and not others. Developers encounter difficulties when starting to abstract from the concrete scenarios as they need to cover wide ground and need to use more complex constructs as mentioned above. This abstraction, or consolidation of representations is described independently in Benyon [7] and Beyer and Holtzblatt [11], but the steps are similar. This phase is an interface between how work is carried out currently and the new system. Whether it is a conceptual scenario or a conceptual work model, all context and individual personas are removed [7]. The question then, how much of the context should be removed during the conceptualization? Are there, for example, technologies that are a part of the work environment which should not be a part of the future system? Are there organizations, e.g. teams, divisions or departments that should be removed from the future system? Will the user behave differently in the new system? If we think that there will be such changes, the particularities belong to the work's context and should be eliminated in the conceptual model. Instead of removing the technologies and support entirely from the domain description, it can be advantageous to separate it clearly as Björner [8] does, since the analyst realizes its purpose in the domain description. An example in the crisis management is triaging. Responders use small colored boards to tick off and count the seriousness of an injury. This task can be described in a user story or a scenario. However, if the developer wants the flexibility to change the technology, i.e. the colored boards, in the future system, it should be excluded in the conceptual scenario. Thus, it is a part of the current context but not the future one. Similarly, Benyon divides the domain into four areas: people, activities, context and technologies.

When the crisis management training simulator was developed, two conceptual models emerged, namely, a consolidated contextual design sequence model of work and topic maps (see **Table 2**). The latter is still under construction. A third one is a generic reference scenario, described with informal flow diagrams. The consolidation of the work models has been quite labor intensive, since the stakeholders are at three sites and the crisis management is different for the three incident types. Still, there are many commonalities, which could form a basis for a domain specific language for crisis management. The consolidation has been carried out on meta-level constructs such as triggers, goals and steps of activities. Also, the Contextual Design work models have meta-level constructs, called decisions and decision support. These were applied during consolidation as an activity step, but sequences were not merged on outcomes of decisions that crisis managers or first responders across incident types need to make. The generic reference scenario models decisions, so clearly it should be a meta-level construct in a conceptual model for crisis management.

The consolidated conceptual models mostly addressed work and strategies and little technology support. There was not a formal attempt to consolidate across management and organization structures. The conceptual model distinguishes commanders from first responders, but does not further divide commanders into levels.

Table 2. Understanding / Conceptual modeling

Goal	Phase	Instrument	Representations	Abstract or Concrete
Consolidation of work models	Conceptual modeling	Consolidation of sequence models	Contextual design Sequence models	Abstract
Consolidation of scenarios	Conceptual modeling	Consolidation of written scenarios	Flow diagram of activities. Management levels, Information flow, Events, Decision points, Actions.	Abstract
Formalizing concepts	Conceptual scenario	Conceptual design	Written text of operational and training concepts.	Mostly Abstract
Formalization of rules in the simulator	Conceptual modeling	Procedures, Organization, Simulation	Topic maps of events in simulator	Abstract

5 From Conceptual Models to System Interactions

Gradually, conceptual scenarios, constituting requirements, can be developed into larger conceptual models of the system with increasing detail. Objects are modeled with their characteristics and relations, tasks are modeled with structures of iterations, alternative paths and parallel threads. Tasks have pre-conditions, post-conditions and other invariants. As soon as the requirements are clear, the conceptual models can describe abstractions of the future system. This phase is still underway. We continue to follow Benyon's development process, where the third phase is envisioning the system through its user interface. The transformation from the conceptual models to system interactions involves adding context of the new system to the models. Some developers will choose to add parts of this context to the abstract representation, i.e. the conceptual model, others will prefer to attempt to envision the new systems in prototypes. In the development of the crisis management training system, low-fidelity prototypes have been created with power point slides. Some of them have had simple interactions but others a series of snapshots (see **Table 3**). As a part of future work, there needs to be an evaluation of how the different incident types require particularities in the concrete implementation.

Table 3. Envisioning and evaluation

Goal	Phase	Instrument	Representations	Abstract/ Concrete
To visualize interactions of parts of scenarios or work model	Envisioning	Develop power point slides based on a conceptual model	Power point slides with simple interactions	Concrete
To compare interaction choices	Envisioning	Develop power point slides based on a conceptual model	Snapshots of different choices of interactions, e.g. input devices	Concrete
To receive feedback on first ideas of interaction	Evaluation	Interview, discussing the previous two representations		

6 Discussion

During the understanding and the envisioning phases, a number of difficulties were encountered. The first one is the difficulty to separate training issues from the operational issues. One of the known objectives of building an information system is to improve processes. It is not only about developing a system to aid with an operation; it is also about improving work [5, 9]. The variation in a training system is that improvement should occur at the training level and not at the operational level. The separation of training from operation on the other hand opens up the opportunity to simplify operations and focus on training objectives. A second difficulty relates to the simulation. Developers are used to build systems which execute tasks on real objects, but do not simulate that execution. Another issue of the simulation is that modern work requires the use of information systems. When training for this work in a simulator, how are these information systems simulated, or integrated into the simulator? The third issue is the concept of a training scenario, which is the input model of the simulator. The training scenario in crisis management training may include a number of things such as the size and type of the event, number of resources initially available etc. Thus, the scoping of the simulation is essential. Finally, a potential difficulty with the development of the crisis management simulator is its evolution. During the understanding phase it became clear that the underlying operation was not a static system, but evolved easily when operational improvements were called for and as new information technologies were adopted and changed work.

7 Conclusion

This paper has attempted to analyse the work carried out by several analysts of a crisis management training simulator. The domain of crisis management is complex and the project is characterized by a project team with different experience and practices. This study showed that simultaneously developers address a number of issues at the macro

and the micro level, and they tend to use abstraction to help them. Probably, developers will attempt to create many such abstractions, to suit their needs.

One way of describing work is to divide it into processes, which communicate between them, eliciting triggers of actions, their intent, decisions to be taken, decision support, output of the actions, required performance and breakdowns. This is the approach analysts of the crisis management training simulator have selected with good results. Perhaps it is due to the type of work of crisis management, i.e. it is heavily activity oriented, where coordination, communication, decision making and different roles in a predefined management and organization structure is of utmost importance, but data, as input or output to functions, plays a less of a role. Another approach would have been to focus on functions and objects or data. Probably, these constructs are too constraining, at least in the initial phases, and do not give the developer enough flexibility. It is our conclusion that function or data-oriented specification can only succeed an activity and communication oriented analysis like the one which has been described during the initial understanding phase. Thus, the function or data-oriented specification takes place during the Envisioning phase.

We further conclude that the concrete representations such as storyboard scenarios and low fidelity prototypes have been useful. However, developers need a conceptual model which gives an overview of the domain and ensures that all parts are included. We propose that a domain specific language for crisis management training may be desirable, at least at the conceptual level. There have been some attempts to define a generic model for crisis management [12]. The danger is that such a generic model will too broad and not useful for an innovative system.

Similarly, it may be questioned whether one development process model for all is the most useful one for developers. This paper has used Benyon's process as a reference, and it has proven useful. However, it does not address issues specific to simulators, such as intelligence and realism. It does not address training or scenario based training. It is likely, that such a process would support developers better than a generic one. Already, process models for simulators and game development [13] have been suggested and one for developing training of decision-making in a synthetic environment [14], an area relevant to crisis management training.

References

1. Pearson, C.M., Clair, J.A.: Reframing Crisis Management. The Academy of Management Review 23, 59–76 (1998)
2. Pearson, C.M., Sommer, S.A.: Infusing creativity into crisis management: An essential approach today. Organizational Dynamics 40, 27–33 (2011)
3. Moats, J.B., Chermack, T.J., Dooley, L.M.: Using Scenarios to Develop Crisis Managers: Applications of Scenario Planning and Scenario-Based Training. Advances in Developing Human Resources 10, 397–424 (2008)
4. Zave, P.: Classification of research efforts in requirements engineering. ACM Comput. Surv. 29, 315–321 (1997)
5. Diaper, D.: Task scenarios and thought. Interacting with Computers 14, 629–638 (2002)
6. Stary, C.: Shifting knowledge from analysis to design: requirements for contextual user interface development. Behaviour & Information Technology 21, 425–440 (2002)
7. Benyon, D.: Designing Interactive Systems. Pearson Education, London (2010)

8. Björner, D.: Software Engineering 3 Domains, Requirements, and Software Design, vol. 3. Springer, Heidelberg (1998)
9. Vicente, K.J.: Cognitive Work Analysis. Lawrence Erlbaum associates, Mahwah (1999)
10. Benyon, D., Macaulay, C.: Scenarios and the HCI-SE design problem. Interacting with Computers 14, 397–405 (2002)
11. Beyer, H., Holtzblatt, K.: Contextual Design. Morgan Kaufmann, San Francisco (1998)
12. Kruchten, P., Woo, C., Monu, K., Sotoodeh, M.: A Human-Centered Conceptual Model of Disasters Affecting Critical Infrastructures. In: Proceedings of the 4th International ISCRAM Conference, Delft, The Netherlands (2007)
13. Raybourn, E.M.: Applying simulation experience design methods to creating serious game-based adaptive training systems. Interacting with Computers 19, 206–214 (2007)
14. Jenkins, D.P., Stanton, N.A., Salmon, P.M., Walker, G.H.: A formative approach to developing synthetic environment fidelity requirements for decision-making training. Applied Ergonomic 42(5), 757–769 (2011)