Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

Executive summary

A Scenario-Based Modelling Method for Simulation Systems

Problem area

The training of crisis management personnel is a complex task that involves many different agencies and participants. This is usually carried out in full-scale incident training exercises every couple of years with smaller scale sessions in between. This leads to infrequent training sessions of the full crisis management team that would be involved in handling a full-scale incident such as an aircraft or railway accident.

The aim of the CRISIS project, a European Framework Programme 7 project, is to address this gap by developing a simulation system that creates a training environment for crisis management teams.

Description of work

The CRISIS project partners include several partners who are involved in crisis management operations, as well as in the training of crisis management. Using the descriptions from these partners, reference scenarios were generated of example occurrences that require a crisis management response. These scenarios, in combination with the user requirements form the basis for the modelling method that is being used to define the functional requirements and the system plan for the CRISIS simulation system.

Results and conclusions

This paper presents the method that is being used in CRISIS for translating users' needs into functional design requirements. The five-step iterative method provides a systematic way of transforming user's requirements into functional requirements using storyboards, a scenario-based visualizing technique. The method is applied in the ongoing development of an

Report no.
NLR-TP-2010-638

Author(s)
A. Rankin
R. Kovordanyi
J.N. Field
M. Morin
J. Jenvald
H. Eriksson

Report classification
UNCLASSIFIED

Date
June 2011

Knowledge area(s)
Training, Simulatie en Operator Performance
Externe Luchtvaart Veiligheid en beleidsondersteuning

Descriptor(s)
Crisis Management Training
Requirements
Scenario-based design
Storyboarding

This report is based on a poster presented at the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Lisbon, May 8-11, 2011

interactive crisis management training system. Existing software has been used for visualizations, providing an easy way of circulate and refining ideas between different stakeholders.

Applicability

The method that is described in this paper is applicable to the design of simulation systems in general and is not specific to the crisis management application of this project. It is particularly applicable for the design of systems that are used for training or for evaluation of procedures and equipment within a pre-defined scenario.

The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement no. FP7-242474. As CRISIS is a collaborative effort the authors would also like to thank all CRISIS partners who might have indirectly contributed to this paper.

Nationaal Lucht- en Ruimtevaartlaboratorium, National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 20 511 31 13, Fax +31 20 511 32 10, Web site: www.nlr.nl

Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2010-638

A Scenario-Based Modelling Method for Simulation Systems

A. Rankin[1], R. Kovordanyi[1], J.N. Field, M. Morin[2], J. Jenvald[2] and H. Eriksson[1]

[1] Linköping University

[2] VSL

This report is based on a poster presented at the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Lisbon, May 8-11, 2011.

Customer	European Commission
Contract number	----
Owner	NLR + partner(s)
Division NLR	Air Transport
Distribution	Unlimited
Classification of title	Unclassified
	June 2011

Approved by:

Author	Reviewer	Managing department
12-05-2011	17-5-'11	26-5-'11

Abstract

A critical step in any system-development process is to identify and understand user needs and to translate these into requirements specifications. An increasingly common approach for eliciting needs and identifying requirements is to use scenarios and to model these scenarios using storyboards. The art of storyboarding is evolving as new technologies provide simple and innovative ways for designers and others to visualize events, activities and interactions. Currently there are commercial software and web-based tools for making low- and high-fidelity prototypes and for re-enacting an event or process. Such tools present a new era as ideas easily can be presented, adapted and circulating between all collaborators – from project managers to end users, interaction designers and software developers. This paper presents a modelling method that uses storyboarding as a technique for translating scenarios and end-user requirements into functional requirements for system development. Described is an iterative five-step method with provided examples of storyboards in each step.

Keywords

Storyboarding, Scenario-based design, Crisis Management training, Functional Requirements

Contents

.

1 Introduction

Determining the requirements for a new software product is a critical step for the success of development projects. Defining a sufficient level of detail for system design, taking various stakeholders into consideration, understanding the implications of the new system and dealing with conflicting requirements are some of the issues that must be managed. Agreeing on the constraints of the system and the prioritization of requirements such as the "must have" and "desirable" are two other issues. There are many techniques that can be used to elicit the requirements, for instance, interviews and focus groups, prototyping, use cases or scenario-based design to name a few (Carroll, 2000; Hackos & Redish, 1998; Schneiderman, 1997). Often, an analyst or designer will use more than one of these techniques throughout the system design process.

For emergency-management training systems, defining scenarios together with relevant stakeholders is a frequently-used method (Drury et al., 2009; Haynes et al., 2009). Such scenarios serve as medium for communication and as a starting point for further requirements engineering and design. Although scenarios are quite helpful in the initial development stages, such as user-requirements engineering, it is sometimes unclear how the scenarios should be transformed into specifications of functional-requirements and detailed system design. Unfortunately, there is a lack of methods, tools and techniques for defining functionality and creating designs systematically from scenarios.

This paper presents a method intended for extracting functional requirements from user requirements using *storyboarding*. Storyboarding is a scenario-based visualizing technique (Goodwin, 2009). The storyboards are based on generic scenarios derived from a large pool of user requirements collected in previous site visits, workshops and interviews with end-users. Translating the scenarios into functional requirements will be performed using the designed five-step method. The five steps are: (1) Concept development using storyboards, (2) Timelines depicting interactions, (3) Visualizations and refinement of user interactions, (4) Extracting functional requirements and, (5) Interactive prototyping.

The method is applied in the ongoing development of an interactive crisis management training system. The training system is being developed within the CRISIS project. The objective of the project is to develop an interactive simulated Virtual Reality (VR) environment to aid users to train crisis management. The VR-system is a desktop application and will encompass a collaborative and interactive simulation for individual and team training across multiple levels of an organization. The end users involved in the project are: ANA at Lisbon airport, British Transport Police in London and ISAVIA at Reykjavik airport.

2 Scenario Based Design

Using scenarios for design of human-centred systems can be a powerful instrument as it allows designers and analysts a simple way of exploring and testing new ideas. By describing the sequence of information exchange, actions and results it forces designers and analysts to reflect on human experiences, human behaviour and their implications on the system. According to Carroll (2000) scenarios are "*the language of all stakeholders*" (p. 58). Storytelling is a familiar narrative structure since childhood – giving people the advantage of already being experts at applying their imagination and overloading meaning into the story (Carroll, 2000; McCloud, 2006). A scenario is initiated by an event, and followed by a sequence of actions. For example, the "aircraft incident" scenario portrayed in Figure 2 describes how an airplane pilot reports to the air traffic control tower that "one engine is out". The air traffic control tower immediately alerts the airport emergency services, which are on-site when the plane crashes on the runway as it is coming in for landing. This alert is followed by a large rescue operation involving multiple organizations. The scenario ends when the fire is put out and all casualties have been transported from the scene.

Another advantage of using scenarios is that they can be used collaboratively. They can easily be circulated to end-users and other stakeholders such as project managers, software architects or graphical designers who can provide new ideas and directive feedback. This flexibility provides the freedom for innovative explorations of design requirements and alternative views without having to commit to one particular solution (Carroll, 2000).

Storyboarding is a visualizing technique in scenario-based design used to represent interactions based on a number of sketches or still images portraying significant actions, often with a matched-up dialogue or other textual description (Goodwin, 2009) . Storyboarding has its origins in the movie industry where a storytelling style of comic books to depict the storyline of a movie is used to visualize the actions pre-production (Saffer, 2006).

Traditionally sketching is done by hand and therefore requires some drawing skills. Lacking this skill could be a huge disadvantage for designers, users or any other stakeholders who needs to express his or her ideas in a perceivable way. The landscape for this may be changing as easy-to-use and inexpensive software tools that can aid the "artists" are becoming increasingly available. These tools can be found at a range of different complexity levels: from simpler tools like *Bubblr*, which supports basic adding of captions to photos in *Flickr,* to more complex (and more expensive) tools like *The Tarquin Engine*, which is a flash template for dynamic navigation. Tools, or a combination of these, are chosen depending on what type and level of visualizations and interactivity the designer wants.

The methodology described here is a combination of elements from other design methods and can be used in a variety of different applications. We are applying it in the CRISIS project for the development of crisis management scenarios and system requirements.

In the CRISIS project the software program *ComicLife* is used for storyboarding. *ComicLife* integrates photos, pictures, captions and lettering to create comic like images (see Figure 1). Our motivations for choosing this particular piece of software are that:

(1) No prior skills are needed such as sketching or programming competencies
(2) The software can be used collaboratively over the internet
(3) Pictures and sketches from various sources (e.g. images from exercises, simulations) can be implemented and made to look coherent
(4) The level of fidelity can be adjusted at different stages in the process by simple "style" –editing
(5) Actions and events can easily be added or removed throughout the whole design process
(6) The images can be reused later on in the scenario or in an altogether different scenario

This means that anyone within the CRISIS team can be involved in the storyboard development and can translate their experiences and expectations into illustrations for the system development. Figure 1 illustrates an example of a storyboard developed using ComicLife. Here, images are based on real photographs which have been modified using style and colouring. The reason for making the storyboard "comic-style" is to help the reader "experience" the story (McCloud, 2006) . The reader, guided by storytelling principles such as pace, clarity and communication, can help clarify ambiguities and contradictions in the story making the scenarios more realistic (McCloud, 2006).

In ComicLife different styles can be used adapted to the level of fidelity appropriate for each image, as well as using "mixed-fidelity" (McCurdy, et al., 2006). Mixed fidelity is when high fidelity is used on some dimensions and low-fidelity on others (McCurdy, Connors, Pyrzak, Kanefsky, & Vera, 2006; Yasar, 2007) . As specific functionalities of the system are determined, certain parts can be made to look increasingly authentic. Colouring of specific details can also be used to draw attention to the use of a specific event or tool. Keeping the story "comic-style" allows focus to stay on the story rather than on details in the picture which may bias the viewer's interpretation and imagination.

Figure 1 Sample storyboard for medical response developed using ComicLife.

3 Requirements Analysis: a Five-Step Method

The five steps presented in this section describe how to get from generic scenarios (e.g. textual scenario descriptions) to functional requirements specifications (see Figure 2). A prerequisite to this five-step method is end-user participation. In the example presented below user requirements have been gathered through interviews, workshops and site visits. These were performed in an iterative process over several months to produce a list of user requirements. The scenarios used for storyboarding are based on the results from the interviews and workshops.

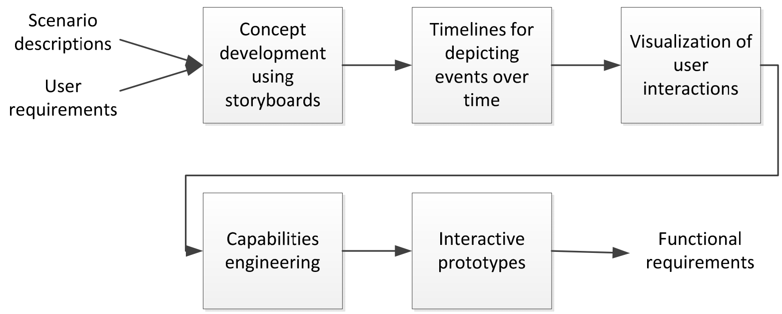


Figure 2 The five-step model

3.1 Concept development using storyboards

In this first step we will translate the textual scenarios into visual form. The purpose of this step is to capture and refine ideas and requirements. The storyboards provide an overview of the flow of actions and interactions taking place in the scenarios. They do not include any information on what goes on "behind the scene". This means that no information on how the user will interact with the system is provided. At this point, the storyboards solely include "the story". Working through each scenario helps determine whether the stated training requirements are unclear, incomplete, ambiguous or contradictory. Figure 3 illustrates the first two minutes of a scenario where an airplane reports engine problems to the air traffic control tower. In the example the air traffic controller raises the alert alarm.

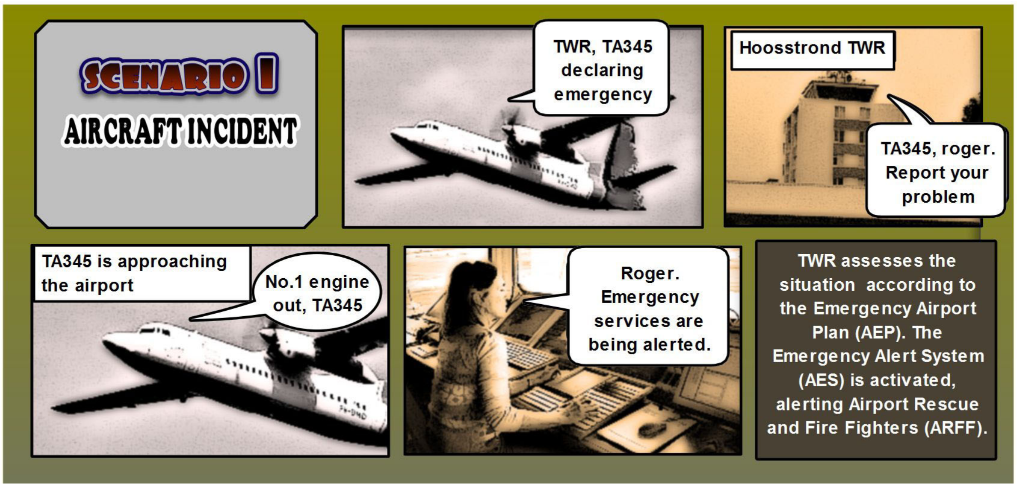


Figure 3 Storyboard from the scenario "The Aircraft Incident".

3.2 Time-lines for depicting events over time

The time-line shows the communication taking place between participants in the scenario. It also shows the communication method used—for instance, telephone and tetra radio. The information portrayed in the time-line provides a different view of the storyboards. Although times and interactions are to some extent represented in the storyboards, rendering them on a time-line gives a more comprehensive perspective. This information is useful when creating the list of functional requirements, to make sure all relevant channels for communication have been listed. The time-line provides a view of "communication clusters", which gives an indication of critical parts of the scenario. This information may lead to revisions in the "story" or placement of particular injects. It is also possible to identify communication needs within the groups represented in the exercise by further splitting the analysis. This can be used to identify the specific events that relate to the fire-fighters for example. The sample time-line in Figure 4 captures the part of the story portrayed by the storyboard in Figure 3.

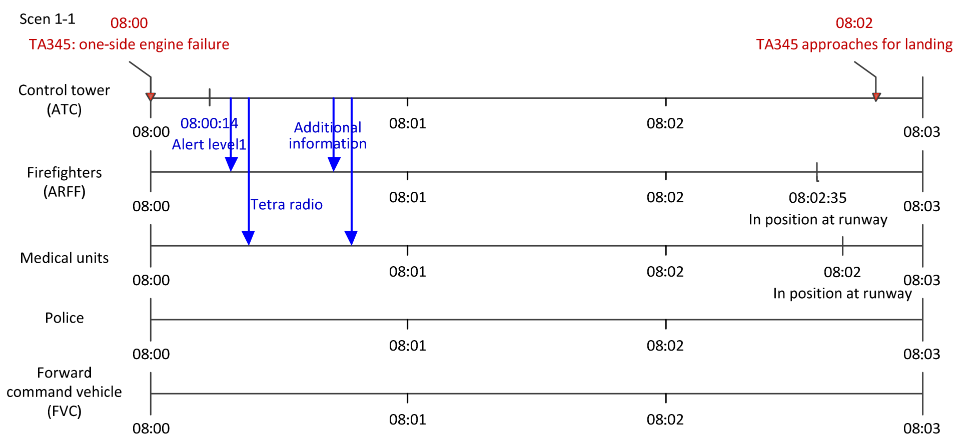


Figure 4 Time-line depicting the communication taking place.

3.3 Visualizations of user interactions

Exploring how the user interacts with the virtual world is carried out by visualizing the user interactions and extracting the functional requirements. First, visualizations of how the users interact with the system are made (Figures 5-9). Ways of interacting and communicating with the system and other participants are illustrated such as keyboard, head set, tetra radio or mobile phone. More than one storyboard can be made for each user-group to provide a variety of user interfaces that can be discussed with the end users. Figure 5 illustrates an airport medical unit receiving the airport emergency alarm (left image) and an on scene commander requesting additional resources (right image). Interactions with the VR-system are performed using a keyboard and mouse and a mobile phone.

Figure 5 In the left image the airport medical emergency services report that they are on site. Interaction with the VR system is performed using a keyboard and, mouse and a mobile phone.

Users at different levels in an organization will have a different set of requirements to execute their task during a crisis. At local, regional or national level command tasks are more abstract and concern strategic decision making. Usually Chief Commanders are present in the same room and utilize tools such as white boards, paper, pens and maps to create and overview and analyze the situation. Immersing this type of work completely into a VR system may provide some challenges as this affects the fundamental principles of teamwork and training (Jenvald, Morin, & Eriksson, 2010). An alternative is partial involvement using virtual windows and displays but including face-to-face communication in a setting resembling a command staff room. The command team receives information from the virtual world through other participants and the simulation staff via telephones, e-mail and fax.

Figure 6 demonstrate a *regional level command team*. They have maps, magnets, white boards available. Information from the virtual world is provided through telephone and radio communication with the outside worlds, i.e. participants and simulation staff interacting with the virtual world. An additional way of providing the command team with information from the virtual world could be video surveillance from the accident scene.

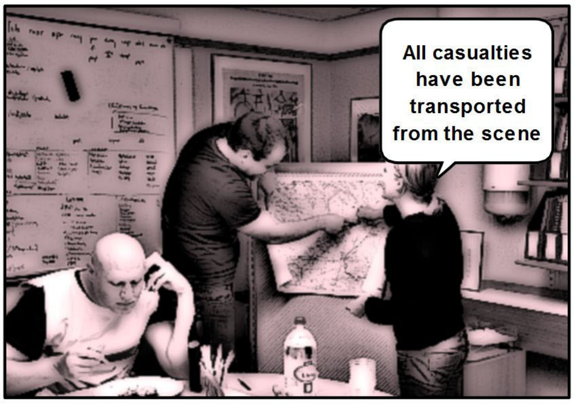


Figure 6 Command team working on strategic level utilizing maps, white boards etc (left image). They receive information and interact with the VR system through telephone and radio communication (right image).

Figure 7 is an example of an *operational level command team* such as an emergency operations centre. Monitors and displays are built into the VR-system and provide information from the virtual world. Further, video surveillance is projected onto a monitor (Figure 7). A head-set integrated with the system is used for phone and radio communication. A key issue is to make the working environment resemble their normal environment as much as possible. As other command teams , operational command staff members often sit in the same room, giving them the benefits of coordinating their work using face-to-face communication. In figure 6 (right image) we see the chief of staff talking to a one of the operational coordinators. White boards, pens and maps are important resources for creating an overview of the situation and information streaming in from different organization.

Figure 7 Operational command team receives information from the VR system through several displays. In the left image we can also see information being received through video surveillance (the right screen in the back). The right image shows the Chief of Staff the Chief of Staff interacting with the coordinator.

The *simulation staff* is able to directly influence the events taking place in the virtual environment. Through injects the scenario can be altered, creating more or less complex situations. Persons in the simulation staff also interact through telephones and radios acting as the “outside” world, i.e. any roles not partaking in the training.

Figure 8 Simulation staff handling external events during training sessions. Here, one member of the group is simulating the radio communication of the TA345 (cf. Figure 2).

Working with storyboards and images is a method of ensuring that people think about all of the details when describing the interactions with the system. When images from past exercises are used, other elements of the interaction between team-members or of their actions are illustrated and then need to be considered for the simulated system. It enables the end-users to represent their use through generating storyboards that can be used in discussions with the design and development team for the system.

3.4 Extracting Functional Requirements

The aim of step 4 is to create a list of functional requirements, derived from the user’s needs. In this case, this means analyzing the capabilities of each user group individually. This analysis is subsequently used when extracting a list of functional requirements.

The themes and content of this list have to be tailored to fit the specific system that is being developed. However, a basic outline containing a set of categories with different types of functionalities can be prepared beforehand. Categorize may be, for instance, user goals, physical abilities, environment etc (see Table 1). These categories for analysis of capabilities and functional requirements may be altered, added or deleted as new issues and questions arise throughout the whole process. Functionalities and features that appear crucial at first may turn out to not to be and vice versa. An example illustrating a list of functional requirements is given below (Table 1).

3.4.1 Capabilities Analysis

In this example the capabilities for "medical personnel" users will be explored. The categories are based on areas commonly referred to within scenario-based design and insights from capabilities engineering (Goodwin, 2009; Jungert et al., 2010): (1) User goals, (2) Skills, (3) Environment (4) Physical attributes and (5) mental models. *User goals* refer to goals the user might have with using the system. Specific user goals often concern a particular function of the system. One user goal could, for example, be better communication with team members during the first-response phase of the emergency operation. The second category, *skills*, defines the skills that various categories of users may have, for example, experience with handling a particular communication device or ability to operate an emergency vehicle. Example of an *environment* can be a command post, where the VR-based training system will help commanders to obtain information from the crisis site, and to communicate with the team leader in the field. *Physical attributes* of the planned system may include various interaction devices, colours of uniforms in the VR-environment, shapes and gender of computer characters or avatars, etc. *Mental models* refer to how the user will interpret the virtual world based on their knowledge and experience in the real world. These models include, for instance, knowledge about rules, procedures and organizational structures. This will affect the users' expectations on how the planned system should behave and is used to develop further detailed storyboards and functional requirements for the design team.

User Role	Goals (abilities)	Skills	Environment	Physical attributes	Mental models	User requirement
Medical personnel	Increased communication skills	– Operate vehicle – Operate tetra radio	– Triage area – Command post	Features on tetra radio	Communication channels	Train communication

Table 1 Analysis of user group (example)

Each category is analyzed and discussed, using the information provided in the previous steps. Options are explored and categories are, if necessary, refined. For example, "physical attributes" could be interpreted as the users attributes in the real world or as the character in the virtual world. Although they will not get too aquatinted with their own physical appearance (a "first person shooter" perspective will be used where the view of the simulated world is represented from the point of view of the player) other participants in the virtual world might. Colours used for clothing may, for instance, be very important. Gender, on the other hand, may not be. Another issue which might be considered is the mobility of the avatar; will the user experience lessened mobility when lifting a heavy patient? If a category appears to have several "layers", as in this case, it may be a good idea to create sub-categories.

3.4.2 Creating a functional requirements matrix

The final part is extracting functional requirements in goal-directed design is to compile a functional requirements matrix. This can be done on the basis of personas, or, in our case, on the basis of the user goals, skills, and environment descriptions, developed in the previous step. The functional requirements matrix should specify data needs, functional needs, system qualities, and other forms of constraints from a user-perspective. Throughout this process, it is important that the system designer remains solution-agnostic by refraining from proposing specific solution ideas. Functional requirements should instead describe abilities in general terms, such as “It should be possible to… “.

Sources can either be a mental model (e.g. the user’s subjective perception of how the system or artefacts within the system ought to function), goals (e.g. enjoy a meal as opposed to find a restaurant, e.g. train communication skills), environment (e.g., the physical environment where the system will be used, and any constraints that can be derived from that environment), skills (referring to the assumed skills and abilities of the user, and the need for system adaptation that arise from these) (Goodwin, 2009).

Prioritizing requirements as “must have” or “desirable” is an important part of defining requirements. The task may be difficult to perform in full when ideas and options are still being iterated. As the storyboards and capabilities analysis becomes more detailed, priorities will most likely change. A refinement of priorities should therefore be performed toward the end of the system design process.

Source	Data needs	Functional needs	System qualities	Constraints	Priority
Command post staff					
Mental model (The system is seen as part of the normal PC-equipment in the command post)			Should run on standard PC	Should be integrated with other means of comm. used by the staff	Must have
Goals (To be able to allocate resources optimally)	Database of available resources	Ability for user to update the database of resources			Must have
Environment (The command post)		Ability to get an overview of the crisis scene			Desired

Table 2 Functional requirements matrix (partial example, not including all types of potential sources). Source serves as a link to earlier established system capabilities and is needed for traceability.

3.5 Interactive prototypes

The aim of this step is to translate the user requirements into more concrete examples that can be discussed with the end-users in more detail. Prototypes are developed in Microsoft Office *PowerPoint* for elements of the system where interactivity is required to generate better feedback from the end users. This enables basic interface and software design without requiring programming competence. While advanced animations can be demonstrated using other tools, such as Flash, these tools are not as accessible for editing in a team collaboration environment. The static interaction that is possible within *PowerPoint* is sufficient for the majority of prototype applications. The use of *PowerPoint* also enables editing on site to incorporate comments and demonstrate their effect immediately.

Crisis management teams include a wide range of different personnel, from the national commanders and coordinators to rescue personnel. The use of prototypes to refine the requirements aims to ensure that all users understand the system concept, and how it applies to their task and training. This approach allows the user requirements to be further refined, and functional requirements can then be derived. Interactivity is an important element of the requirements in system design. It is also an important tool for clarifying the user requirements. Developing early prototypes of the software for evaluation may not be the most effective way when the requirements are not fully defined. The aim of this step is to develop basic prototypes quickly and allow users to see and understand the system as it is designed and developed. As the concept matures, and the details become more defined, it is possible to translate this into more detailed mock-ups, *PowerPoint* prototypes and eventually basic game prototypes before starting development on the actual system. Certain elements of the simulated environment (such as the virtual world scenery) may be developed sooner in the development process and can therefore form a background to the later prototypes.

A concept that has been discussed with the stakeholders during initial development discussions can be translated into a set of presentation slides using photos and screen mock-ups put together in *PowerPoint*. By using the same storyboarding software on the graphics, a uniform, “storyboard” appearance can be given to the prototypes as well. Creating an image of the system “screenshot” enables elements of the screen to be highlighted where the user can interact with the system. For example, a button on a screen, or stepping through a syllabus, can be implemented through the interaction capabilities of *PowerPoint*. This enables a relatively complete prototype of the system to be developed, and gives the user the ability to step through the screens.

As an example, the system that is being developed in this project combines the virtual environment with crisis management system tools to train the operational personnel. The interfaces between the virtual environment and the tools themselves can be illustrated in a prototype (Figure 9). The prototype demonstrates how the user accesses the different systems they work with, through clicking on the virtual screens, which call up the system interface in a format that they would normally use.

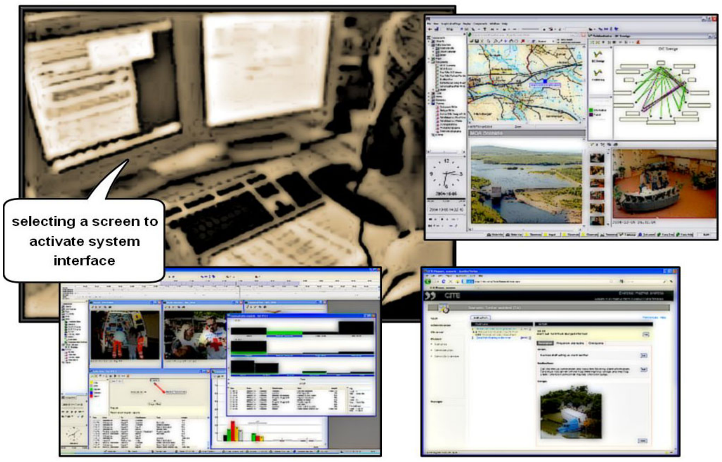


Figure 9   Introduction screen shots for virtual environment to system simulation screens from the system concept

Through a combination of photos and illustrations made externally, such as in *Adobe Photoshop* or any other image editing software, it is possible to mock up certain elements of the software interface, or to demonstrate how a training scenario would develop. The elements of the storyboarding from the previous steps can be included and used in the *PowerPoint*, and can form the basis of the "screenshots" or interface. It is also possible to develop examples of the software itself, such as illustrated in Figure 10 (left image) with a prototype developed for the EU FP7 "pplane" project.

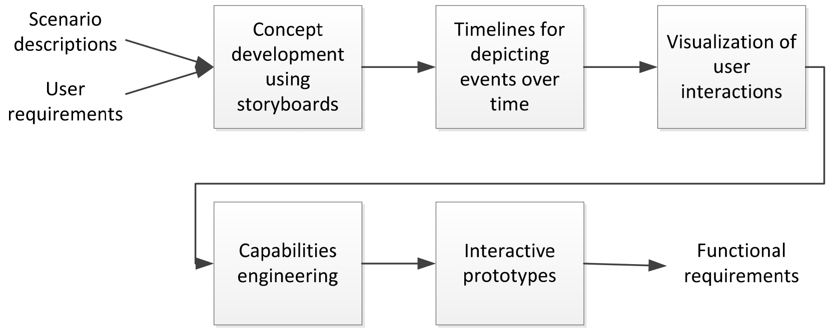
Figure 10   Example of login and selection screens (left image). Example of active screen with sub-screens (right image).

Figure 10 (left image) illustrates the login and startup screens for the mock-up of a conceptual PPlane interface created in *PowerPoint*. Through selecting the buttons on the display, it is possible to change to the associated slide which displays the next "screen" of the software. It is also possible to put together slides to illustrate how a drop down menu would be used, to select a destination in this case. The slides are developed to illustrate the aspects of the software interface that need to be evaluated, and can therefore focus on a particular usage of the "system". It is also possible to mock up more intricate displays, as illustrated in Figure 10 (right image). This demonstrates a map interface, and how the buttons on the side of the interface can be used to display additional information on the existing display or to change the displayed screen. For example, clicking on the button at the top left of the display, changes the display (the slide) to the weather information display. While this is a static display, it illustrates how the system can be used, and can aid both the users and the developers in the refinement of the requirements.

When developing a simulated environment or game system, it is also possible to create interactive, dynamic prototypes using the game engine that forms the core of the system. The virtual world can then be included as it is developed, while the focus of the prototype may be on elements such as the way that the user interacts with the system, or navigates around the scene.

When creating a prototype that is based on a relatively simple level of interaction it is important to be aware of the level of realism that a user may interpret. A user may believe that the functionality that is being demonstrated is already fully available, while it is only a concept that is yet to be translated into technical requirements. Therefore an element of "storyboarding" is useful in the interactive prototyping stage as well to reduce the apparent realism.

4 Discussion



Although the five steps are presented in a sequential order they should be viewed as parts of an iterative process. Questions will arise during the storyboarding process and as uncertainties become less uncertain the storyboards will become increasingly detailed. The first sets of storyboards include a variety of different design-options that can be tested in the interactive prototypes (Step 5). As the development moves forward the storyboards are refined and when necessary, tested again. Likewise the interactive prototypes will be refined as the requirements become more defined, and the feedback from the different users is combined. It may also be necessary to go further in the interactive prototyping and develop more advanced prototypes to demonstrate particular aspects of the system or virtual environment – either using other tools (such as Flash) or starting the software development process itself.

There is the possibility that the time required to carry out the full five steps analysis to define the functional requirements is too much for a project or application. It may therefore be necessary to simplify some of the steps, or to omit steps to simplify the overall process. The process presented in this paper is proposed as a complete process and intended to ensure that the functional requirements are fully defined. At this stage in the CRISIS project the storyboards, analysis of the scenarios and visualization of the user interactions have been valuable in understanding the user's needs and how they intend to apply the training system once it is available.

While we have encouraged the use of storyboards and imagery in the discussions with the end-users about their operations and training requirements, we have also tried to ensure that the discussions are open. There is a risk that images lead to constrained thinking when it comes to concept definitions, particularly when trying to define the use of a virtual environment for the first time. The possibilities envisaged by the end-users, or by the design team, may become limited by their current experience of the existing training environment – live exercises or otherwise. In some ways this can be compared with the ability to imagine a world when reading a book (without imagery provided), and the limitations that are imposed on that world when images are present or when watching a film. It has therefore been as important to hold discussions with end-users as it has been to provide storyboards of interactions. The storyboards are a useful tool for the project participants to clearly communicate their concepts and ideas with each other, accompanying the discussions that are held.

Keeping all project stakeholders involved in the requirements elicitation process is a key to success, from the end users to the technical developers. This feedback loop keeping the end-users involved in the process, throughout the requirements definition and refinement process, is important to ensure that the system that is being developed can be applied. This ensures that the system is driven by the training needs of the users, rather than providing a technology driven system where the training needs to be adjusted to meet the capabilities of the system. Using a visualizing technique such as storyboards ensures accessibility to all partners. As the storyboard format is electronic, it provides the opportunity for circulating ideas easily, allowing all partners to alter, adapt or try new ideas and to share them with others.

Two aspects in this process that deserve extra attention are keeping an open mind for exploring boundaries and at the same time making sure constraints are defined in time. Finding a balance between these is what, at the end of the day, will provide innovative and interesting design options, which do not exceed the limits of what is possible within the scope of the project. It is necessary to stay “outside of the box” as ideas and concepts are being formed. External constraints and preconceptions about existing technology may bias the exploration of design options. However, not taking, for instance, technological or economical constraints into considerations will create problems further down the line. A balance must be found between exploring and limiting.

This process is being applied within the CRISIS project to develop a full system concept of the virtual environment, the instructor’s tools and the training concept which is based on the user’s needs. This full system concept defines the complete system that the end-users currently require and will also be defined in the functional requirements for the system. Once the functional requirements have been defined, these will be analysed by the technical development team which is when the technical, time and economical constraints are considered. At this stage the storyboards and prototypes can also be used to assist the technical development team in determining the priority of functional requirements. In this way the end-users can also be consulted when defining the functional requirements that will be included in the first version of the system, and those that can be implemented at a later stage. This process is intended to maintain the end-users involvement in the system definition and ensure that the system that is developed is in line with their expectations.

5 Summary

An essential part of developing new systems is to understand the users' needs and being able to transform these into functional design requirements. This paper presents a five-step method transforming user's requirements into functional requirements using storyboards, a scenario-based visualizing technique. The method focuses specifically on eliciting functional requirements, a part of the development process often lacking in clarity. Further, it provides a way of allowing users and other stakeholders to participate in the development of the design, doing this in a timely manner and keeping the costs down. Using existing software for visualizations offers the opportunity to try different design options, circulate the ideas and refine them as details of the system performance are worked out.

6 Acknowledgements

We would like to thank all partners in the CRISIS project for contributing to the work presented in this paper. A special thanks to Kritian Boe Hansen, CrisWare, for insightful comments and suggestions. The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement n° FP7-242474.

7 References

Carroll, J. M. (2000). *Making Use: Scenario-Based Design of Human-Computer Interactions*. The MIT Press.

Drury, J. L., Pfaff, M., More, L., and Klein, G. L. (2009). A Principled Method of Scenario Design for Testing Emergency Response Decision-Making. *Proceedings of the 2009 International Conference on Information Systems for Crisis Response and Management (ISCRAM 2009).*

Goodwin, K. (2009). *Designing for the Digital Age: How to Create Human-Centered Products and Services*. Wiley.

Hackos, J. T., & Redish, J. C. (1998). *User and Task Analysis for Interface Design* (p. 512). John Wiley & Sons, Inc.

Haynes, S. R., Purao, S., & Skattebo, A. L. (2009). Scenario-Based Methods for Evaluating Collaborative Systems. *Computer Supported Cooperative Work*, *18*(4), 331-356. Springer.

Jenvald, J., Morin, M., & Eriksson, H. (2010). Challenges for user interfaces in VR-supported command team training, *Proceedings of 6th Nordic Conference on Human-Computer Interaction*, October 16-20, 2010, Reykjavik, Iceland.

Jungert, E., Grönwall, C., Hallberg, N., Kylesten, B., Lantz, F., & Eriksson, L. (2010). A Generic Architecture for Surveillance Systems, *Proceedings of the 16th International Conference on Distributed Multimedia Systems*, Illinois, USA.

McCloud, S. (2000). *Reinventing Comics: How Imagination and Technology Are Revolutionizing an Art Form*. Harper Paperbacks.

McCloud, S. (2006). *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels*. Harper Paperbacks.

McCurdy, M., Connors, C., Pyrzak, G., Kanefsky, B., & Vera, A. (2006). Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. *Proceedings of the SIGCHI conference on Human Factors in computing systems CHI 06*, Quebec, Canada.

Saffer, D. (2006). *Designing for Interaction: Creating Smart Applications and Clever Devices*. Peachpit Press.

Shneiderman, B. (1997). *Designing the User Interface*. Addison Wesley.

Yasar, A.-U.-H. (2007). Enhancing experience prototyping by the help of mixed-fidelity prototypes. *Mobility 07 Proceedings of the 4th international conference on mobile technology applications and systems and the 1st international symposium on Computer human interaction in mobile technology*. ACM.

NLR

Appendix A Poster presented at ISCRAM 2011

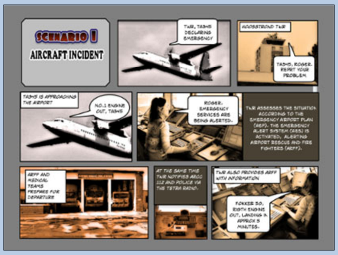
CRISIS

Amy Rankin ▪ Linköping University ▪ amy.rankin@liu.se
Joris Field ▪ NLR ▪ joris.field@nlr.nl
Rita Kovordanyi ▪ Linköping University ▪ rita.kovordanyi@liu.se
Magnus Morin ▪ VSL ▪ magnus.morin@vsl.se
Johan Jenvald ▪ VSL ▪ johan.jenvald@vsl.se
Henrik Eriksson ▪ Linköping University ▪ henrik.eriksson@liu.se

SEVENTH FRAMEWORK PROGRAMME

Scenario-Based Modeling for Simulation Systems

CRISIS is a distributed virtual reality system for training crisis management (CM) on multiple levels of command and across organizations. Interpreting and unifying the needs of all users therefore poses challenges. Using visualizing techniques to illustrate the training environment such as the scenario, possible interfaces, exchange of information, support tools, and how the training is organized can help overcome some of the difficulties. Using simple storyboards and prototypes enable the involvement of all stakeholders when creating and adapting ideas throughout the design phase, aiding the process of understanding and merging the user requirements.



Scenario Storyboards

Scenarios developed together with the end-users of the system are visualized using storyboards. These provide an overview of the flow of actions and interactions taking place in the scenarios.

Airport medical unit interacting with the training system using keyboard and mouse.

Command team utilizing maps and whiteboards.

The command team receives information and interacts with the VR system through telephone and radio communication

Visualizing the user interactions

Storyboards are used to illustrate how the user interacts with the system.

Interactive Prototypes

Prototypes are developed for elements of the system where interactivity is required to generate better feedback from the users.

Middlesex University · CRISWARE · spaceapplications · NLR · UNIVERSITY OF ICELAND · Object Security · VSL · Linköping University · ANA Aeroportos de Portugal · a + e solutions = (BI) · BRITISH TRANSPORT POLICE · isavia