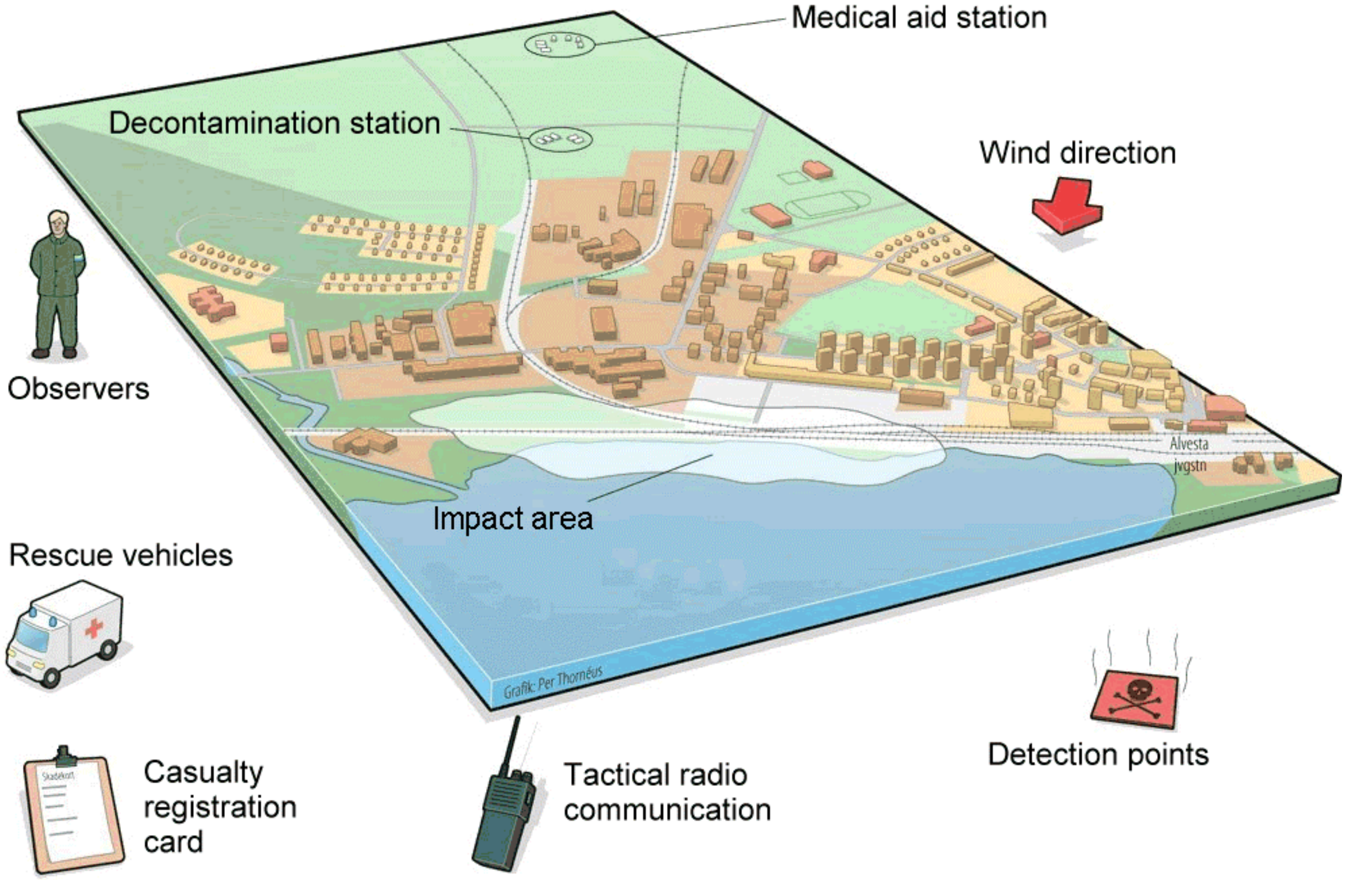
FOI-R--0496--SE
May 2002
ISSN 1650-1942

Technical report

Joel Fjordén

Representation and Visualisation of Casualty Flows in Rescue Operations



Command and Control Systems
SE-581 11 Linköping

SWEDISH DEFENCE RESEARCH AGENCY

Command and Control Systems
P.O. Box 1165
SE-581 11 Linköping

FOI-R--0496--SE

May 2002

ISSN 1650-1942

Technical report

Joel Fjordén

Representation and Visualisation of Casualty Flows in Rescue Operations

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	FOI-R--0496--SE	Technical report
	Research area code 2. Operational Research, Modelling and Simulation	
	Month year May 2002	**Project no.** E7042
	Customers code 5. Commissioned Research	
	Sub area code 21 Modelling and Simulation	
Author/s (editor/s) Joel Fjordén	**Project manager** Mirko Thorstensson	
	Approved by Hans-Åke Olsson	
	Sponsoring agency Swedish Armed Forces	
	Scientifically and technically responsible Mirko Thorstensson	

Report title

Representation and Visualisation of Casualty Flows in Rescue Operations

Abstract (not more than 200 words)

Effective tools for evaluating course of events from rescue operations are invaluable. This thesis describes the development of such a tool that enables dynamic specifications of rescue operation scenarios. Scenarios are descriptions of rescue operations, which include the geographical area and casualty flows. The scenario system supports external import of scenario description data, as well as logged events from operations. Replay of collected data is dynamically updated in the scenario system, allowing both continuous overview of changes as well as discrete snapshots of points in time. The system is implemented as a plug-in module, extending the existing data analysis and visualisation system MIND. The scenario system is explored using data collected from a rescue operation performed in Alvesta, in 1997.

Keywords

casualty flow, computer-supported analysis, data collection, description language, evaluation, MIND, modelling, rescue operation, simulation, visualisation

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 70 p.
	Price acc. to pricelist

Utgivare	**Rapportnummer, ISRN**	**Klassificering**
Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 Linköping	FOI-R--0496--SE	Teknisk rapport
	Forskningsområde 2. Operationsanalys, modellering och simulering	
	Månad, år Maj 2002	**Projektnummer** E7042
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 21 Modellering och simulering	
Författare/redaktör Joel Fjordén	**Projektledare** Mirko Thorstensson	
	Godkänd av Hans-Åke Olsson	
	Uppdragsgivare/kundbeteckning Försvarsmakten	
	Tekniskt och/eller vetenskapligt ansvarig Mirko Thorstensson	

Rapportens titel (i översättning)

Representation och visualisering av skadeflöden vid räddningsoperationer

Sammanfattning (högst 200 ord)

Det är ovärderligt med effektiva verktyg för att utvärdera händelseförlopp vid räddningsoperationer. Det här examensarbetet beskriver utvecklingen av ett sådant verktyg som tillåter dynamisk specificering av scenarion för räddningsoperationer. Scenarion är beskrivningar av räddningsoperationer och innehåller det geografiska området samt skadeflöden. Scenario-systemet stödjer inläsning av externa scenariobeskrivningar, liksom loggade händelser från operationer. Uppspelning av insamlad data uppdateras dynamiskt i scenario-systemet, vilket tillåter en kontinuerlig överblick av förändringar och även diskreta ögonblicksbilder. Systemet implementeras som en delmodul som utökar det befintliga analys- och visualiseringssystemet MIND. Scenario-systemet utforskas med insamlad data från en räddningsoperation genomförd i Alvesta, 1997.

Nyckelord

beskrivningsspråk, datainsamling, datorstödd analys, MIND, modellering, räddningsoperation, simulering, skadeflöde, utvärdering, visualisering

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 70 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Representation and Visualisation of Casualty Flows in Rescue Operations

by

Joel Fjordén

LiTH-IDA-Ex-02/59

Abstract

Effective tools for evaluating course of events from rescue operations are invaluable. This thesis describes the development of such a tool that enables dynamic specifications of rescue operation scenarios. Scenarios are descriptions of rescue operations, which include the geographical area and casualty flows. The scenario system supports external import of scenario description data, as well as logged events from operations. Replay of collected data is dynamically updated in the scenario system, allowing both continuous overview of changes as well as discrete snapshots of points in time. The system is implemented as a plug-in module, extending the existing data analysis and visualisation system MIND. The scenario system is explored using data collected from a rescue operation performed in Alvesta, in 1997.

Acknowledgements

I want to thank my supervisor Pär-Anders Albinsson for programming support, criticism and comments during this work. I also want to thank Magnus Morin for interesting discussions about rescue operations, Markus Axelsson for support on MIND specific programming matters, Johan Jenvald for comments in different areas, my project leader Mirko Thorstensson and also all the personnel in the MIND research group at the Swedish Defence Research Agency for their support in different phases of my work.

Finally I would like to thank my beloved Nina for her support and encouragement.

Linköping May 2002

Joel Fjordén

Contents

Appendices

Appendix A: The XML Scenario Graph for Alvesta 1997

Appendix B: The XML Scenario Persons for Alvesta 1997

Appendix C: The XML Scenario Events for Alvesta 1997

Appendix D: A Scenario Graph Example with Hierarchies

List of Figures

1 Introduction

This opening chapter briefly describes the background of the work, as well as the purpose and task of this thesis. This is followed by a short description of the result and the chapter closes with an outline of the report.

1.1 Background

In many situations it is important to be able to analyse the course of events from operations, especially in military and rescue operations. Data need to be collected from several sources that record events on the scene. This can be done automatically through technical systems, such as GPS receivers and different simulation systems, or by manual observations submitted in a form that facilitates automatic management and classification of reports (Thorstensson, 1997). The MIND system (Jenvald, 1996), is used for this kind of data collection, data presentation and in-depth analysis of efforts in training operations.

Rescue operations and emergency management in the case of large accidents or natural disasters are very demanding tasks, which involve numerous individuals and teams working together to save lives. Effective training and evaluation methods of simulated situations are crucial for the outcome of real disasters. Full-scale mission training is expensive in terms of equipment, personnel and time. Therefore it is important to make the most out of every training exercise, in order to learn as many lessons as possible. After-action reviews (Rankin et al., 1995) and exercise analysis allow the trainees and trainers to reflect on, and learn from, the performance of teams and individuals in relation to the overall mission objectives and in the light of the actual course of events (Jenvald, 1999).

The MIND system was used to support rescue operations for the first time in 1997. These field trials were conducted in Piteå and Alvesta (Jenvald et al., 1998), where rescue units, medical personnel and police responded to a simulated chemical-weapons attack on a potential target such as an airbase or a railway junction. The MIND system was used successfully (Thorstensson et al., 1999), but difficulties were encountered when adapting the system for new rescue operations.

Since 1997, a series of field trials have been conducted using the MIND system, for example in Orlando, Florida, in 2000 (Crissey et al., 2001) and in the Stockholm Underground, in 2000 (Thorstensson et al., 2001).

1.2 Purpose and Task

The MIND system has been used successfully for rescue operations (Thorstensson et al., 1999) and after-action reviews (Rankin et al., 1995), since 1997. A problem is that every new scenario, which describes a rescue operation in terms of geographical area, casualty flows and events, needs to be rebuilt into the system. In the previous version of the MIND system, version 2, there was only one pre-programmed scenario, Alvesta 1997 (Thorstensson et al., 1999). If the MIND system was to be used for rescue operations that differed from this scenario structure, then certain parts of the system had to be extended or rebuilt from scratch, which involved programming. In the new version of the MIND system, version 3, there are no pre-programmed scenarios at all and dynamic visualisation of casualty-flows during replay of rescue operations is not possible. Instead diagrams, generated from collected data at certain points in time, are created in Microsoft Excel and imported into the MIND system, in order to make it possible to show some snap-shot diagrams from rescue operations. The goal of this thesis is to make a process that avoids the rather time-consuming task of rebuilding each new scenario and instead makes the process more generic. This means that it should be possible to specify arbitrary scenarios in external files that later can be imported by the scenario system.

The main task is to create a language, specified in XML (Birbeck et al., 2001), for describing scenarios and events, limited to the domain of casualty flows in rescue operations (Morin et al., 2000). This implies a higher degree of flexibility, increased consistency and the possibility of automating a lot of work. It should also be possible to visualise scenario events through graphical views, which are limited to one in this thesis, during replay of the rescue operation. The largest part of this thesis will cover the actual implementation of the system in Microsoft Visual C++ 6.0. Furthermore, the system should be able to parse scenario and event data, in order to build an internal representation of components needed for replay of the whole rescue operation. The minimum requirement of the scenario system is at least to support the rescue operation performed in Alvesta, in 1997 (Thorstensson et al., 1999).

1.3 Result

This thesis has resulted in a plug-in module that extends the MIND system to support dynamic specifications of scenarios and events in the domain of rescue operations and casualty flows. The system has been explored with logged events, converted to the new structure, from the Alvesta rescue operation in 1997.

1.4 Structure of this Report

The following chapters cover these topics:

Chapter 2: Method and Limitations
This chapter describes the method used in this thesis, as well as some limitations.

Chapter 3: MIND and Rescue Operations
A short description of the MIND system is given in this chapter. After that comes an explanation of how field-trials work and an example of a rescue operation is also given.

Chapter 4: The MIND Scenario Extension
This chapter explains the current problem when creating scenarios in the MIND system, as well as the suggested solution.

Chapter 5: Analysis
Here the results from the analysis phase are discussed. Two topics mainly are covered, namely the XML structure of the scenario and that of the events.

Chapter 6: Design
This chapter is based on the previous chapter and describes the work and results during the design phase. Class diagrams over the scenario system are also presented.

Chapter 7: Implementation
Here some implementation issues are discussed, but the major part in this chapter is a description of the main functionality of the scenario system.

Chapter 8: Conclusions and Future Work
This final chapter discusses some conclusions and also gives examples of future work that can be applied to this thesis.

Chapter 9: References

Appendix A: The XML Scenario Graph for Alvesta 1997

This is the XML structure of the scenario graph for the rescue operation performed in Alvesta, in 1997.

Appendix B: The XML Scenario Persons for Alvesta 1997

This is an extraction from the XML structure of the scenario persons, acting as casualties in the rescue operation performed in Alvesta, in 1997.

Appendix C: The XML Scenario Events for Alvesta 1997

This is an extraction from the XML structure of the events retrieved from the rescue operation performed in Alvesta, in 1997.

Appendix D: A Scenario Graph Example with Hierarchies

This is an example of an XML scenario graph structure containing hierarchies.

2 Method and Limitations

This chapter describes the chosen design method, as well as the limitations in this thesis.

2.1 Method

The method used for this thesis is based on an object-oriented development approach (Fagerström, 1999), consisting of three phases: analysis, design and implementation.

2.1.1 Analysis

The purpose of the analysis phase is to decide what to implement, often from some kind of requirements specification. It is important that this phase only considers what the system should be able to handle and not how to do it. The domain of rescue operations with casualty flows needs to be analysed, in order to determine what factors need to be represented. The result from this analysis will be used when defining the language for describing scenarios and events.

2.1.2 Design

The design phase continues where the analysis phase trailed off, without any clear boundaries. This phase takes the problem to a deeper level, now focusing more on how to actually implement the result from the analysis phase.

2.1.3 Implementation

This is the final phase and here the actual implementation of the system begins, using all the material from the other phases. If the design result really holds, then it should be quite straightforward to do the implementation.

2.2 Limitations

There are limitations in the choice of programming language, since Microsoft Visual C++ 6.0 and sometimes also MFC, Microsoft Foundation Classes, have to be used. It would probably have worked to use C# and Visual Studio .NET to implement certain parts of the system, but since it is such a new product it was safer to use the older Visual C++ 6.0 for support in the area.

Another limitation is the domain of rescue operations and the concentration to casualty flows. There is also a limitation in the different visualisations of the scenario during replay of an operation. The limitation is that it is enough for this thesis only to implement one view and then if time allows this can be developed further to more general views.

3 MIND and Rescue Operations

This chapter opens with a short description of the history of the MIND system. It continues by describing how field-trials work. Finally, a description of a rescue operation is given.

3.1 History of MIND

In 1992 the National Defence Research Establishment began cooperating with the Army's Centre for Brigades, at that time the Army Armour Centre. The purpose was to start a project that would develop a system for simulation of mines and artillery. The project was named the MIND project and the purpose was to develop both technology and methods that would make it possible to measure and analyse the effect of different units. The technical system has got its name from the project and is called the MIND system (Örnberg et al., 1996).

From the start in 1992 until 1997, the main purpose of the MIND system was to support military training by providing simulations, data collection and presentation (Jenvald, 1996). Starting in 1997, a series of field-trials were carried out in order to adapt the MIND system to also support different kinds of rescue operations (Jenvald et al., 1998; Thorstensson et al., 1999).

3.2 Simulation

The purpose of simulations in the MIND system is to increase the realism of field-trials by simulating hazardous factors in the environment. The system supports different types of simulations in different application areas. In order to support emergency response, management training and evaluation the MIND system incorporates interfaces and tools for importing and presenting the results from simulations of the propagation of chemical agents in different terrains and under varying environmental conditions.

The MIND system does not include a proprietary simulator for chemical agents, but instead provides a flexible interface, which allows the results from different simulators to be imported. When simulations are used in training exercises, it is important how these results are fed back to the participants of the exercise. Ideally, the simulation results should affect their ordinary equipment to generate data consistent with the outcome of the simulation (Jenvald et al., 1998).

3.3 Data Collection

The MIND system supports dynamic data collection from alternative sources, which can be collected by computer-supported automatic equipment or manually by trained observers. Various exercise demands impose requirements and limitations on the methods and techniques for data collection and data transfer from the exercise field to the MIND centre (Section 3.5). These factors include the need for real-time monitoring, a desire to minimize the time between the exercise and the after-action review (Rankin et al., 1995) and various technical and financial restrictions. Geographical and environmental conditions can also be factors that imply restrictions on data collection methods. The rest of this section gives some examples of data collection methods (Jenvald et al., 1998).

GPS receivers are important for registration of the time and position of different units. This enables analysis of approach routes, force deployment and the utilisation of transportation resources. Field observers can be used where the typical task includes complex classifications and requires human judgement. With structured reports (Thorstensson, 1997) it is possible to overcome the difficulties related to the human ability to observe and formulate reports based on observations. Command post observers are also field observers, but specifically trained and equipped to monitor and register the activities at the command post (Thorstensson, 1998; Worm et al., 1998).

The recovery and treatment of casualties is paramount in rescue operations. The persons acting as casualties are often given a simulated injury through make-up before the scenario starts. A frequently used data collection method is for the casualties to observe themselves throughout the rescue operation by using a casualty registration card that is attached to the wrist (Thorstensson, 1997).

Detection is the primary means of initially establishing and continuously monitoring the propagation of a chemical substance during a rescue operation. This is performed by using fixed detection plates along the approach routes to potential targets and also by using mobile and portable chemical detection equipment. Weather reports are also essential in a scenario involving hazardous substances and they are obtained as structured reports from designated field observers.

Time-stamped recordings of the radio channels used for communication can provide valuable information of the taskforces’ activities. Field observers can also use digital cameras to take photographs and a photograph report card to record the time and location where the picture was taken. Photographs are used to illustrate interesting situations during the exercise and to record environmental factors. Video recording can provide continuous coverage of an unfolding situation and is very useful for supporting the analysis of procedures and routines.

3.4 Presentation

The ability to present recorded data from training exercises, during different parts of an overall training period, is most decisive for supporting effective training. Prior to exercises, presentation of similar scenarios and replay of previously conducted missions (Figure 3-1) can be used to motivate the trainees and to increase their understanding of complex interaction between different units (Jenvald et al., 1998).

During exercises, presentation can support monitoring of the units' activities, improving the trainers' ability to control the scenario development and to increase exercise situation awareness. After exercises, presentation is used to support feedback to trainees during after-action reviews (Rankin et al., 1995) and as an important part of in-depth analyses.

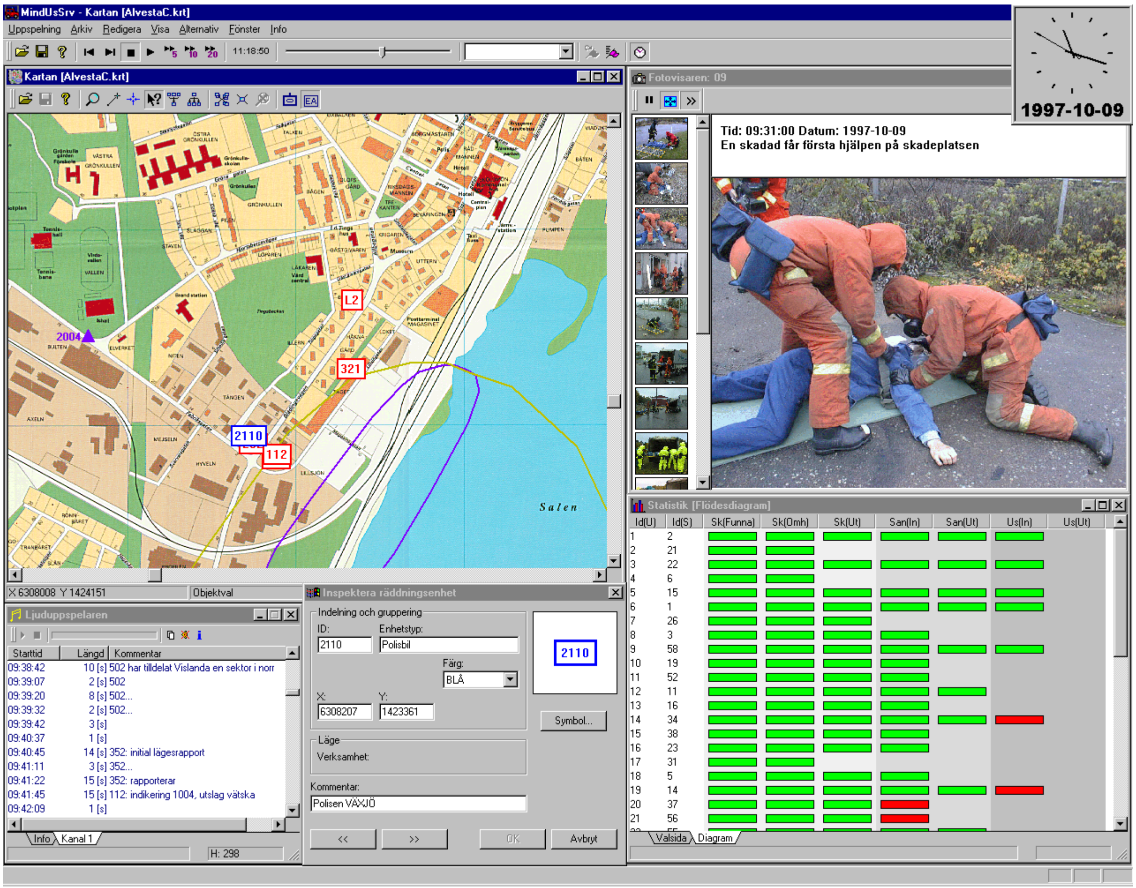


Figure 3-1: The graphical user interface of the MIND system's presentation tool. The figure shows rescue units (depicted as boxes with numbers) on the digital map, annotated photographs in the photograph tool, annotated radio recordings in the sound tool, a police patrol in the unit browser and a casualty flow diagram in the diagram view. All views show the situation at 11:18:50 A.M. on October 9, 1997.

3.5 The MIND Centre

The MIND Centre is established at a suitable location in or near the field well before the start of the exercise. The main requirements are that the premises should be suitable for office work and that there is sufficient electric power for several computers and peripherals.

The MIND Centre performs preparations of the instrumentation system before the exercise, and during the exercise it co-ordinates simulation and data collection activities throughout the exercise area. All data is compiled for the after-action review (Rankin et al., 1995), as soon as the exercise has finished and the data collection is completed. The MIND Centre supports the exercise commander and other officials at the after-action review by running the visualisation tool that replays the operation.

3.6 A Rescue Operation

In the morning of October 9, 1997, enemy aircraft attacked the railroad junction in the town of Alvesta, in southern Sweden (Thorstensson et al., 1999). The air strike included both conventional weapons and the chemical warfare nerve agent VX. The fire brigades from Alvesta, Växjö, Lessebo, Uppvidinge, Ljungby, Markaryd, Älmhult and Tingsryd were alerted and joined forces in an effort to rescue the casualties from the attack. Additional units from the medical service, the police and the county administrative board participated in the rescue operation.

On arrival to the disaster scene, the rescue force was divided into three sections with different tasks (Figure 3-2). One section deployed a decontamination station, with capacity to decontaminate casualties on stretchers and also rescue personnel leaving the contaminated area. Another section established a medical aid station for providing advanced first aid for casualties arriving from the decontamination station. The medical aid station, deployed in heated tents, stabilised and prioritised casualties for the final transport to a hospital. The major part of the rescue force was directed to the primary impact area where it recovered casualties of the air strike. Its main task was to find and evacuate casualties from the contaminated area to the decontamination station.

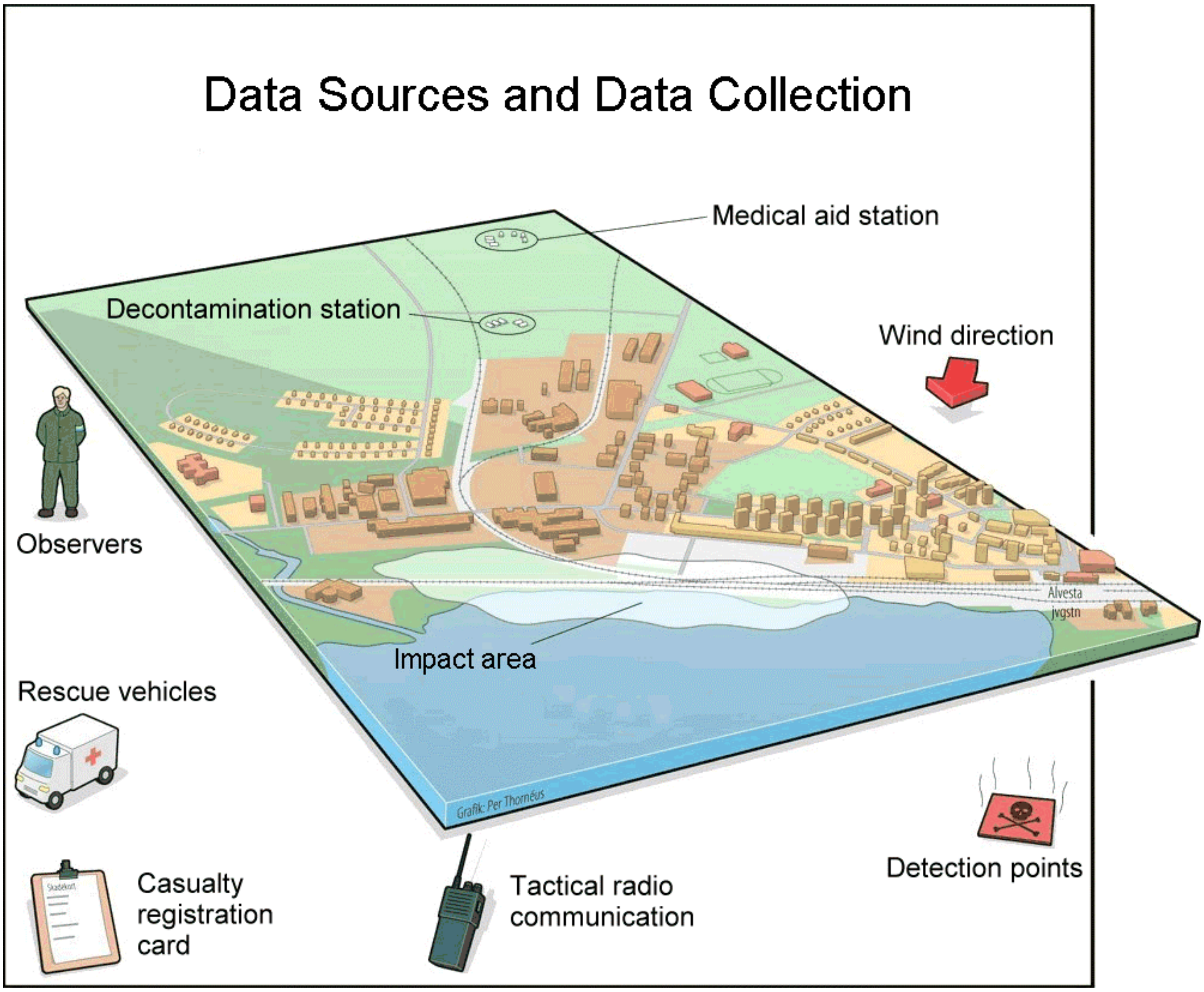


Figure 3-2: An overview of the main exercise area during a computer-supported rescue operation in Alvesta, together with the different data sources. Graphics: Per Thornéus.

4 The MIND Scenario Extension

This chapter describes the current situation of the MIND system regarding rescue operations. It also describes what the extension from this work solves and why this is needed.

4.1 A Current Problem with Rescue Operations

The MIND system has been used successfully in several rescue operations (Thorstensson et al., 1999). As mentioned before, one problem is that every new rescue operation demands rebuilding parts of the system. The previous version of the MIND system, version 2, mostly supported one kind of rescue operation with casualty flows, made for the exercise in Alvesta 1997 (Thorstensson et al., 1999). The components needed for this rescue operation were pre-programmed into the system and certain parts of the system had to be extended or rebuilt from scratch, if it was to be used for other kinds of rescue operations, for instance rescue operations containing more stations, like hospitals, or casualties.

In the new version of the MIND system, version 3, there are no such pre-programmed components for rescue operations at all and dynamic visualisation of casualty-flows during replay of the rescue operations is not possible. Instead diagrams, generated from collected data at certain points in time, are created in Microsoft Excel and imported into the MIND system, in order to be able to show some snap-shot diagrams from rescue operations.

The existing problem when adapting the MIND system to support new scenarios is time-consuming and requires both proficiency in programming and knowledge about the MIND framework. But, since a lot of this work has common underlying processes, it is desirable to facilitate the creation of rescue operations and to support dynamic visualisation of casualty-flows during replay, with the release of version 3 of the MIND system.

4.2 Solution

The problem description above suggests that a description model is needed for specifications of rescue operations with casualty flows. The description model will be a language for describing rescue operation scenarios and it should not require programming skills or knowledge in the MIND framework. With this in mind, XML, Extensible Markup Language (Birbeck, 2001) is chosen for defining the description language. XML is easy to learn and is becoming a standard when it comes to structured data storage. There exists a variety of XML parsers today and this is another reason for choosing XML.

4.2.1 Scenarios

The term scenario, in this thesis, describes a rescue operation with casualty flows. The scenario description language must support a structure for the specification of a geographical area, symbolising the area of the rescue operation, with different sites and transport routes. Since the domain of the scenario system is restricted to rescue operations with casualty flows, rules for specifying these casualties and the flow of transports between sites are needed as well.

4.2.2 Views

Views make it possible to visualise the events during replay of rescue operations. Views can, for instance, be certain types of diagrams (Figure 4-1) or presentations of the rescue operation area on a map (Figure 4-2 and Figure 4-3). It is desirable that the new MIND system includes some standard views, concerning casualty flows, in order to avoid the programming needs when views are needed for new scenarios.

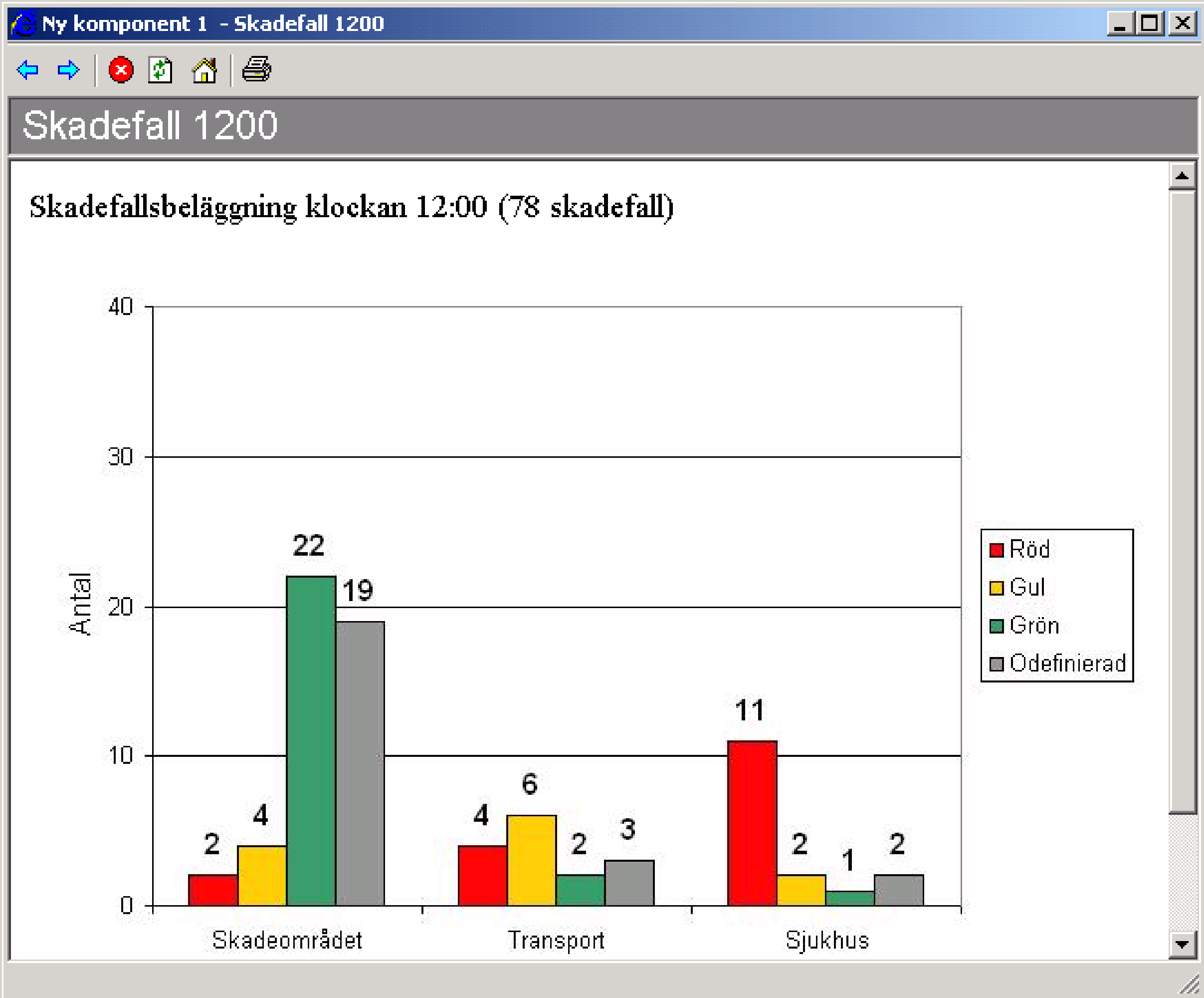


Figure 4-1: A view showing the number of casualties with certain priorities at different stages in the rescue operation.

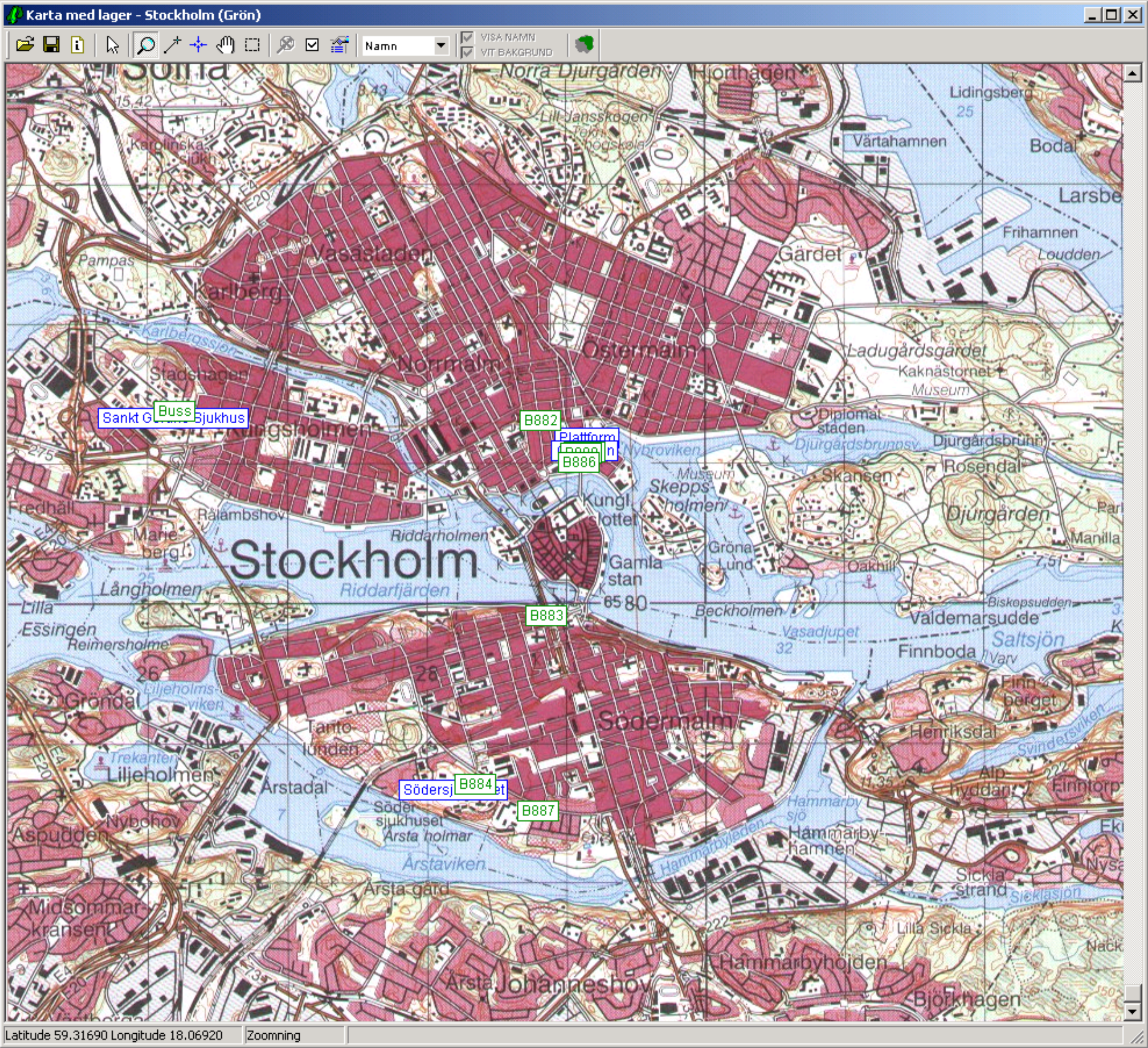


Figure 4-2: A view containing representations of rescue vehicles and sites on a map, which overviews the area of the rescue operation performed in the Stockholm Underground, in 2000 (Thorstensson et al., 2001).

Figure 4-3: A view containing representations of rescue vehicles on a close-up photo of the casualty area in the rescue operation performed in the Stockholm Underground, in 2000 (Thorstensson et al., 2001).

4.2.3 The Scenario Creation Process

When the description language is defined, it is possible to create XML files containing different scenarios. These scenario specifications will be created by the staff in the MIND research group before, during or after the rescue operation takes place. The next stage is to create the parser that interprets these scenario structures, in order to build the internal representation of the scenario in the MIND system. Recorded events regarding casualty flows, from the rescue operation are also needed before it is possible to replay the scenario in the MIND system. A specific XML structure is needed to support the creation of these scenario events and an additional parser is needed for the translation of this event structure to the internal representation of the events in MIND (Figure 4-4).

When the description language is defined for creating both the XML scenario structure (Section 5.1) and the XML scenario events (Section 5.2), and when the parsers have been implemented, it should be possible to import all data needed into the MIND system. Finally, it should be possible to replay the rescue operation, using the imported data from the XML files containing the scenario structure and events. A view is needed in order to visualise the course of events. In this work this will be a standard view, which can be used by all scenarios, containing a bar diagram presenting the number of persons at each geographical site.

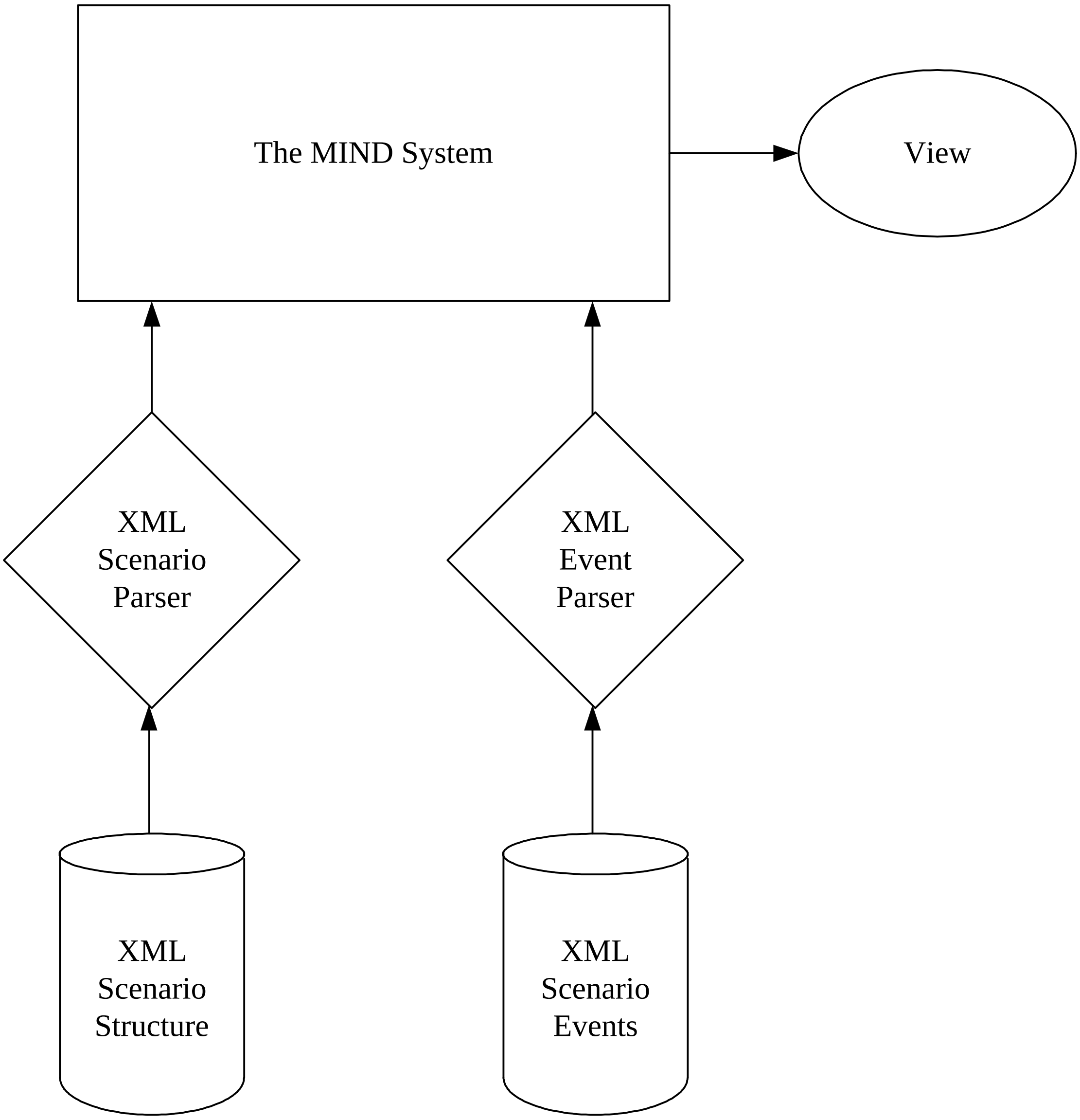


Figure 4-4: The Scenario Creation Process. The parsers build the components in the MIND system, from the data in the XML files. A view makes it possible to visualise the events during replay of the rescue operation.

5 Analysis

This chapter describes the work during the analysis phase, but also changes made during the design and implementation phases. There are mainly two topics discussed, namely the XML structure of the scenario and that of the events. The result from this analysis is a description language for scenarios and events.

5.1 The XML Scenario Structure

The main issue for the XML scenario structure is the possibility of a language for specifying dynamic representations of scenarios. This is needed because it is impossible to know all conceivable scenario information beforehand. Therefore it must be possible to specify an arbitrary geographical area, containing some kind of sites and connecting routes. It should also be possible to have any number of site hierarchies at any depth. Graph structures are appropriate for this, since it is easy to represent and visualise the sites as nodes and the routes as arcs. The casualties in the rescue operation must be able to hold any number of attributes, in order to simulate as many states as possible.

Each site is specified under a common tag called “Sites” and each path under a tag called “Paths”. The sites and paths are in turn specified under a tag called “Graph”. The root tag is a “Scenario” tag which holds both the graph and a tag called “Persons” that in turn contains one or more “Person” tags. The scenario root tag also holds a “Title” tag, which is a unique title of the scenario (Figure 5-1).

```
<Scenario>
        <Title></Title>

        <Graph>
                <Sites></Sites>
                <Paths></Paths>
        </Graph>

        <Persons>
                <Person></Person>
        </Persons>
</Scenario>
```

Figure 5-1: Outline of the structure for a scenario.

5.1.1 Sites

Each site, which can have certain attributes, is represented as a node in a graph and it is connected to other sites through paths (Section 5.1.2), or arcs in a graph. Sites can be nestled, that is a site can as mentioned above, contain a graph itself. Hierarchies might generate unnecessary high detail levels, which imply complications for the views, but some scenarios can make use of this. Therefore and for future expansions, any number of hierarchies at any depth is supported (Figure 5-2).

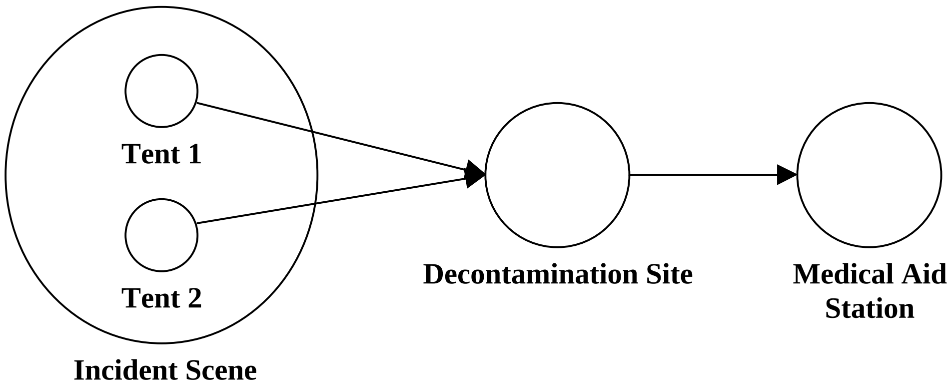


Figure 5-2: Site hierarchy example. The site "Incident Scene" consists of two sub-sites, "Tent 1" and "Tent 2".

There is only one attribute for the site that is required and must be specified in the site structure (Figure 5-3). This attribute is "Name" and it is the name of the site, for example "Incident site", which has to be unique. All other attributes are optional and among those is "Shortname" that indicates an abbreviation for the site consisting of a maximum of four characters. This is a straightforward solution to the labelling problem when it comes to diagrams. If the name is too long in a bar diagram, for example, it will not fit under the bar. This problem is avoided by using a shorter name under the bar.

<Site>

<Name></Name>

<Shortname></Shortname>

<Type></Type>

<Registrations>

<Registration Type="**In**"></Registration>

<Registration Type="**Out**"></Registration>

</Registrations>

<Position Type="**WGS84**">

<Latitude CardinalPoint="**North**"></Latitude>

<Longitude CardinalPoint="**East**"></Longitude>

</Position>

<Date></Date>

<Time></Time>

</Site>

Figure 5-3: Outline of the structure for a site.

The attribute “Position” is the geographical position for the site in WGS84 coordinates (LMV, 1996), since the MIND system internally handles coordinates in this format. “Date” and “Time” are other attributes that specify when the site is going to get the specified position, which means the time when the site is planned to be in place. If the position is specified but the date and time are not, then it is assumed that the site will have this position at the beginning of the scenario.

“Type” is an attribute that tells the type of the site, for example “Decontamination”, that can later be used by views for representing sites of the same type in some special way. A bar diagram can for example represent some data from sites of the same type, as bars that are close together or even as one single accumulated bar.

“Registrations” is a list of attributes, each of which is called a “Registration”, that indicates the registrations at the site. Registrations represent measuring devices, manual or automatic, that register some event in time. Casualty report cards can be used as a manual way of recording events and when they occur (Thorstensson, 1997). A GPS receiver, which registers time and position, is an example of an automatic method. First the idea was to make it possible to specify any number of registrations for a site. This turned out to be troublesome in the implementation phase and also unnecessary, because other kinds of registrations can be handled by the dynamic attributes in the person structure (Section 5.1.3). Instead there are only two registration points that can be specified. These are specified in a “Type” attribute and can be either “In” or “Out”, which specifies that the site will log the time for persons arriving and leaving the site, respectively. The system will log both “In” and “Out” events, if such an event would occur, even though they were not chosen as a registration. It is then up to the view if they should be taken into account or not, depending on the choice of registrations. It is possible to name the registrations, which can be a good idea for different views when setting different labels.

The rest of this section is devoted to attributes that were discarded during the analysis phase, due to redundancy, discussion or simply because they were not needed. From the beginning until the actual implementation of the parser there were “Path” attributes, containing the names of the paths (Section 5.1.2) that the site was connected to, specified in the site itself. This gave rise to redundancy, since this information can be retrieved by searching through the “Path” attributes in the “Paths” section. This was still the choice though, since presumably, it would be easier and more direct to look up the paths that connect to a certain site. When it was time for the implementation it turned out that this was not the case at all, because the parser runs in two separate stages, where the first one builds all the sites and the second one all the paths. The obvious choice was of course to remove the “Path” attributes from the site. This also results in less mistakes when creating the XML scenario structure, since there is no need to refer to the correct path names again, thereby misspellings are avoided.

At first all views that a site should support were to be specified as attributes under the site, so that the site would only have to keep as much information as needed for the chosen views. The views were to be predefined in the system and then choices could be made among those. This turned out to be quite awkward since some views do not only belong to a single site, but to the whole scenario. The choice then was to also have a global attribute list for those views, but this only got more and more complicated, resulting in quite a complex structure. The idea finally came to remove the view attributes once and for all, since all events exist in a large event list in the MIND system (Morin, 1996) and thus they exist and can be reached by the views anyway. This gave rise to the separation of views and the scenario structure.

In an early version of the site structure an attribute called “Parallel” could be specified in the site with a value indicating the name of the site it was parallel to.

This attribute resulted in some ambiguity, since it could be interpreted as physically parallel sites or sites that had parallel resources. The attribute also gave rise to redundancy, and therefore it was removed. Now it is instead the task of the view to use some graph searching algorithm if it needs to find out this information.

5.1.2 Paths

A path represents any kind of transportation route that connects two sites and is represented as a directed path in the scenario graph structure. Note that the term path is not to be associated with the normal meaning of the term path when concerning graph theory, where it actually means one or more arcs leading from one node to another. The term path in the XML structure refers to one arc, or for example road in the real world, only.

When the structure for the paths were discussed, some questions arose about what was going to be allowed and not. One of them was whether to allow cycles in the graph, but it turned out that this should not be allowed since the paths represent the direction of the flow that is planned for the scenario. In the situation of rescue operations where transports are made to move people from one site of treatment to another and finally to a hospital, for example, it is not planned to move people back again to an earlier treatment station. Cycles would also complicate things for certain views, like diagrams.

Another question turned out to be about parallel roads, and this is allowed in the sense of duplicate resources (Figure 5-4) (Morin et al., 2000). As mentioned before this can be specified by having the same type attribute on the sites, it is then the task of the view to find the parallelism, through some graph searching algorithm.

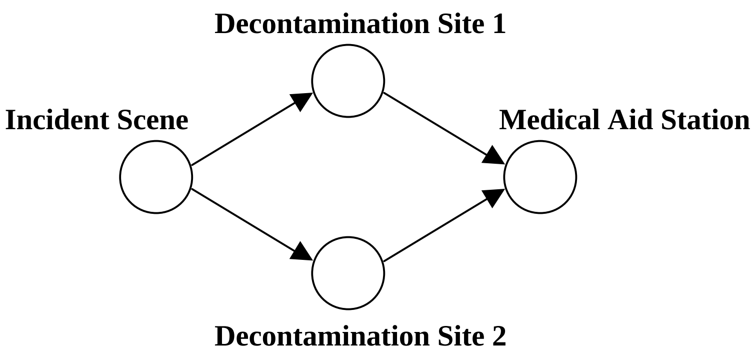


Figure 5-4: Example of a scenario with parallel sites.

The path structure (Figure 5-5) has three attributes that are all required to be specified. The first one is “Name” which indicates the name of the path. The other two attributes connect the path to two sites and make it directed. One of those is the “From” attribute that should have the name of the site that it comes from and the other attribute is “To” and it should instead have the name of the site that the path leads to. Currently there are no more attributes but it is easy to expand it in the future if need be, for instance with capacity or type of road.

When it comes to specifying the paths between sites, there are some rules that must be followed in order for the parser to create the scenario correctly. All paths that come from and go to a site on the same depth are to be specified under the same graph tag as the sites. When there are hierarchies involved and paths come from a site in a lower hierarchy level and go to a site in a higher hierarchy level or vice versa, these paths have to be specified under the same graph as the sites in the lower level hierarchy.

```
<Path>
        <Name></Name>
        <From></From>
        <To></To>
</Path>
```

Figure 5-5: Outline of the structure for a path.

5.1.3 Persons

A person is someone who participates in the scenario as a casualty, acting as realistically in the situation as possible. The persons are each specified under a “Person” tag (Figure 5-6), which in turn are kept in a tag called “Persons”. All persons that participate in the scenario and who later should be able to be affected through events (Section 5.2), need to be specified with all their specific attributes. There is of course a possibility in later expansions that it should be possible to specify one or more general persons with certain characteristics.

Some of the attributes that can be specified for a person are static and among these is the attribute “Id”, which is required to be unique for the person. All other attributes are optional, like “Name” that is the name of the person and “Sex” which indicates gender. There is also an “Age” attribute for holding information about the person’s age, as well as a “Comment” tag for arbitrary comments about this person. The last static attributes are for specifying the position, date and time for the person. The attribute “Position” can be either in WGS84 coordinates (LMV, 1996), or in the form of a site name that then indicates at which site the person is. It is also possible to specify both of them. The position will be set instantaneously if the attributes “Date” and “Time” are not specified, but if they instead are specified the position will not be set until that time.

It is also possible to specify any number of dynamic attributes for the person. These are each called a “Property” and are collected under a common “Properties” tag. Each property needs to be specified with its type, which can be an integer, a float or a string. It is also possible to specify an initial value for the property as well as giving the property a name. The name is required and needs to be unique among the person’s properties, because this name is specified in the XML scenario events file as the recipient of a certain event (Section 5.2).

```
<Person>
    <Id></Id>
    <Name></Name>
    <Sex></Sex>
    <Age></Age>
    <Comment></Comment>
    <Properties>
        <Property Desc="Priority">
            <Type Desc="string" />
            <Initvalue></Initvalue>
        </Property>
    </Properties>
    <Position Type="Name"></Position>
    <Date></Date>
    <Time></Time>
</Person>
```

Figure 5-6: Outline of the structure for a person.

5.2 The XML Scenario Events

The events for the scenario are specified in its own file, using a special structure. The root tag in this structure must be called “Scenarioevents” and it contains three child tags for activation, position and property events. Each event is encapsulated in a “Sample” tag, which contains the attributes “Date” and “Time” of the event.

All events have a site or a person as receiver. In the case of a person, the tag must specify an attribute “Type” with a value “Id” or “GUID”. If the event is instead for a site, then the type value can be either “Name” or “GUID”. Each of the values indicates how the system is to connect the event to the right object (Section 6.1.1). The value “Id” is the id of the person receiving the event, the value “Name” is the name of the site and “GUID” is a global unique identifier that each object gets when really created in the scenario system.

5.2.1 Activations

The main tag, of the activation events, is called “Activations”, which in turn contain one “Sample” tag for each active event. There can be a tag called “Person” or a tag called “Site” in each sample (Figure 5-7), indicating if the event is for a person or a site. In addition to these tags each sample also needs a tag called “Active” with a value indicating true or false. This is the status that the site or person will get when receiving the event. The active status indicates if the site or person exists in the scenario or not.

```
<Sample Date="" Time="">
        <Site Type="Name"></Site>
        <Active></Active>
</Sample>
<Sample Date="" Time="">
        <Person Type="Id"></Person>
        <Active></Active>
</Sample>
```

Figure 5-7: Outline of the structure for the active events.

5.2.2 Positions

Each position event is encapsulated in a “Sample” tag and collected under a common tag called “Positions”. There are two different types of positions that can be specified, one for setting the WGS84 coordinates (LMV, 1996) of a site or a person (Figure 5-8) and another for setting the abstract position of a person (Figure 5-9), in form of a site that indicates which site the person currently is at.

When the position is specified in WGS84 coordinates, the other tag in the sample must be either a “Site” or a “Person”. The structure is a little more complicated when the position event is in the form of a site, because registrations need to be considered. The first tag is a “Person” that indicates who will get the event. The second is a “Registration” tag with an attribute called type, which can have the value “In” or “Out”. Included in the registration node whose value specifies the site in which the person is to be registered. This means that the event holds information about what person comes “In” or “Out” from a certain site.

```xml
<Sample Date="" Time="">
        <Position Type="WGS84">
                <Latitude CardinalPoint="North"></Latitude>
                <Longitude CardinalPoint="East"></Longitude>
        </Position>
        <Site Type="Name"></Site>
</Sample>
<Sample Date="" Time="">
        <Position Type="WGS84">
                <Latitude CardinalPoint="North"></Latitude>
                <Longitude CardinalPoint="East"></Longitude>
        </Position>
        <Person Type="Id"></Person>
</Sample>
```

Figure 5-8: Outline of the structure for the WGS84 position events.

```xml
<Sample Date="" Time="">
        <Person Type="Id"></Person>
        <Registration Type="In">
                <Site Type="Name"></Site>
        </Registration>
</Sample>
```

Figure 5-9: Outline of the structure for the events with a site as position.

5.2.3 Properties

All events for the specified properties in the person structure are stored under a common tag called “Properties”. Each property event is in a “Sample” tag and contains two child tags. These are “Person” and a tag with the name of the property. The value of the property tag is the new value for that property (Figure 5-10).

<Sample Date="" Time="">

<Person Type="**Id**"></Person>

<PropertyName></PropertyName>

</Sample>

Figure 5-10: Outline of the structure for the property events.

6 Design

This chapter uses the information from the analysis phase and combines this information with the work during the design phase. The structure of the scenario system will be discussed, together with some diagrams over the needed classes.

6.1 Structure of the Scenario System

The two different XML structures were discussed in the analysis chapter. The data in these files need to be converted into different components in the MIND system, which is based on the component object model, COM (Grimes, 1999). There are numerous components in the MIND system (Axelsson, 2002a; Axelsson, 2002b), but only those needed for the scenario system will be discussed here (Figure 6-3).

6.1.1 Objects

An object is a model of a physical object, which might have a representation on a map. The state of the object is handled through events or through interaction by users. Site and person objects are the only MIND objects needed for the scenario system (Figure 6-3).

6.1.2 Sources

A source is a container of events, and every source has a number of events that are connected to an object or view. Each source can convert different data files into events. The scenario system needs two different sources, since there are two different XML structure files. One scenario source will be needed for converting the XML scenario structure, containing the graph and persons, into the needed component structure in MIND. A second source, the scenario event source, is needed for converting the XML scenario events into MIND events (Figure 6-3).

6.1.3 Documents

Pictures, sound and text are examples of documents and what they all have in common are the storage of data. When the XML scenario structure is to be translated into objects in the MIND system, some structure is needed in order to keep all relations between the sites and their corresponding paths. Since a MIND document is able to store data, it seems appropriate to have a scenario document to hold this information about the graph (Figure 6-3).

The structure needed for holding the sites is forming a tree, where the leaf nodes are actual sites and the other nodes are abstract sites containing a hierarchy of sites (Figure 6-1). The root node in the tree is only an abstract container of all sites, needed in order to hold all sites together. It is not a real site itself but can be interpreted as the world, which contains different sites.

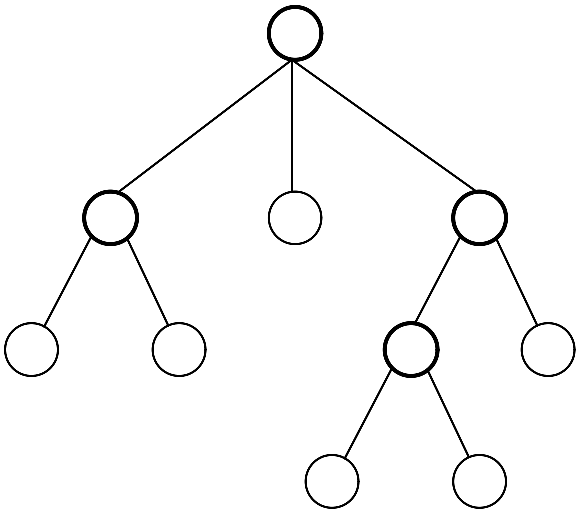


Figure 6-1: A tree structure, showing abstract site hierarchy containers (bold circles) and actual sites (thin circles).

Each node in the site tree is an object of a site class and the relation with paths between them is represented through objects of a path class (Figure 6-2). Each site object needs three lists of pointers, in order to keep the relations. Two of these contain pointers to paths leading to the site and leading from the site. The other one contains pointers to the sites in the hierarchy below, if there is one. The path objects also have pointers, where one points to the site that the path comes from and the other points to the site that the path leads to.

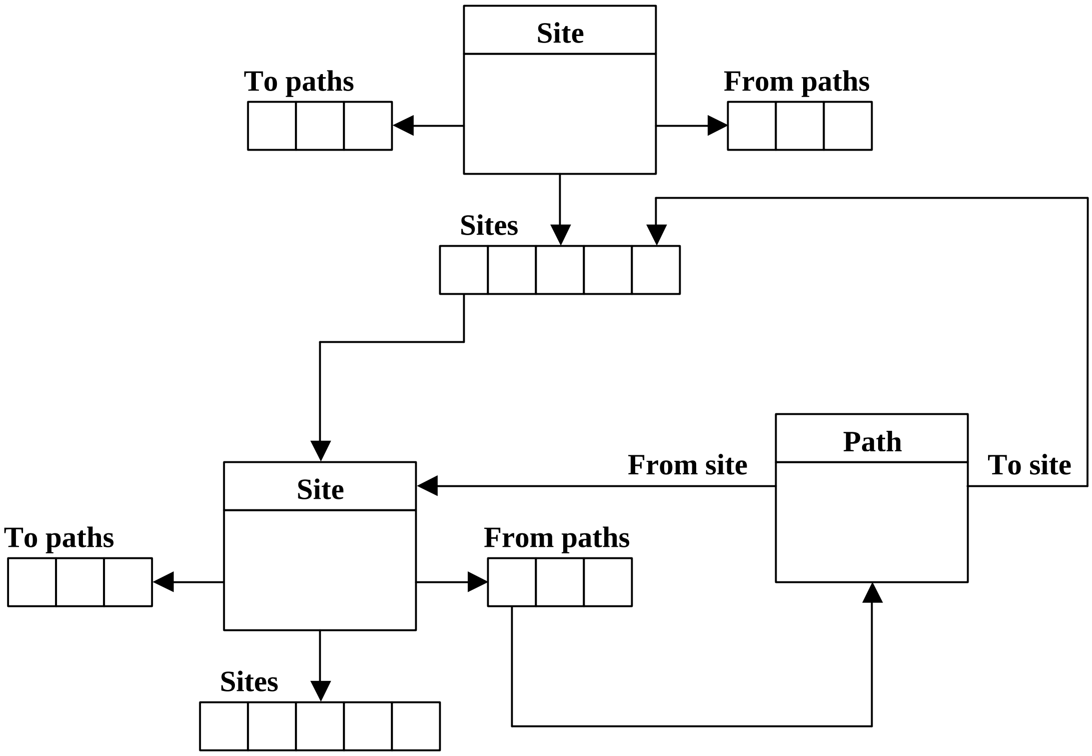


Figure 6-2: Class structure showing the relations between sites and paths.

Except from holding the graph structure, the document also needs to hold information about all the persons. It can do this in form of a list of pointers to person objects.

6.1.4 Views

The purpose of a view is to interact with the user, by showing components in different ways. Views receive events in order to change its content during replay of an operation. The scenario system can have any number of views, but in this thesis only one is created. This scenario view is a bar diagram, showing the number of persons at each site. The view uses the information stored in the scenario document, together with events during replay (Figure 6-3).

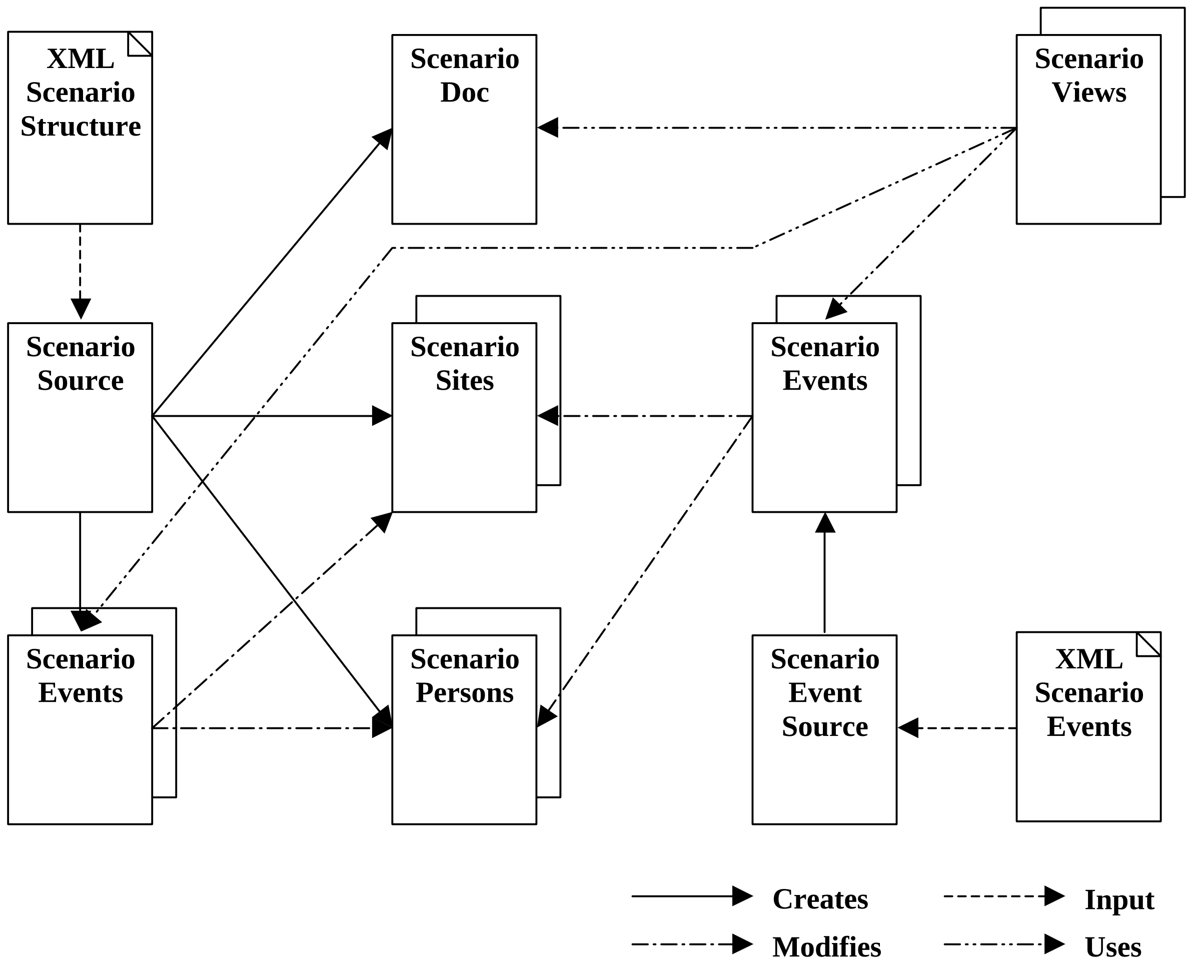


Figure 6-3: Component structure of the scenario system. The scenario events are created at two different points in time, which is indicated by the two scenario event components.

6.2 Scenario System Classes

Each component, in the previous section, has its own class (Figure 6-4). There is only one object of some classes for a single scenario, but most of them have several objects, like sites and persons.

The scenario view (Figure 6-5) is separated from the scenario system, since the system is not dependent on the view. The view is needed though, in order to present the data in some way during replay of the operation. It is possible that the scenario view contains many different kinds of views.

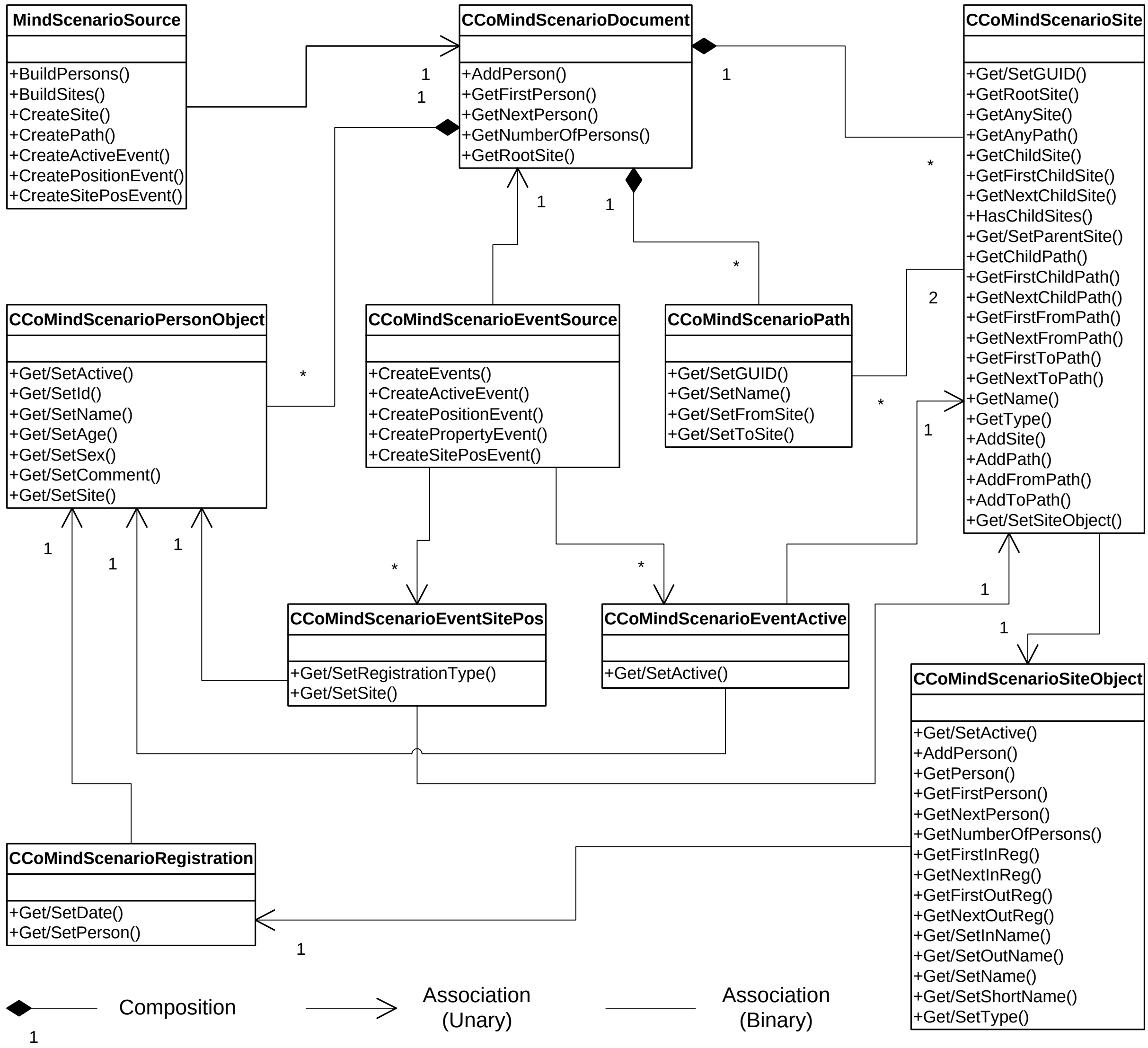


Figure 6-4: Class diagram of the scenario system.

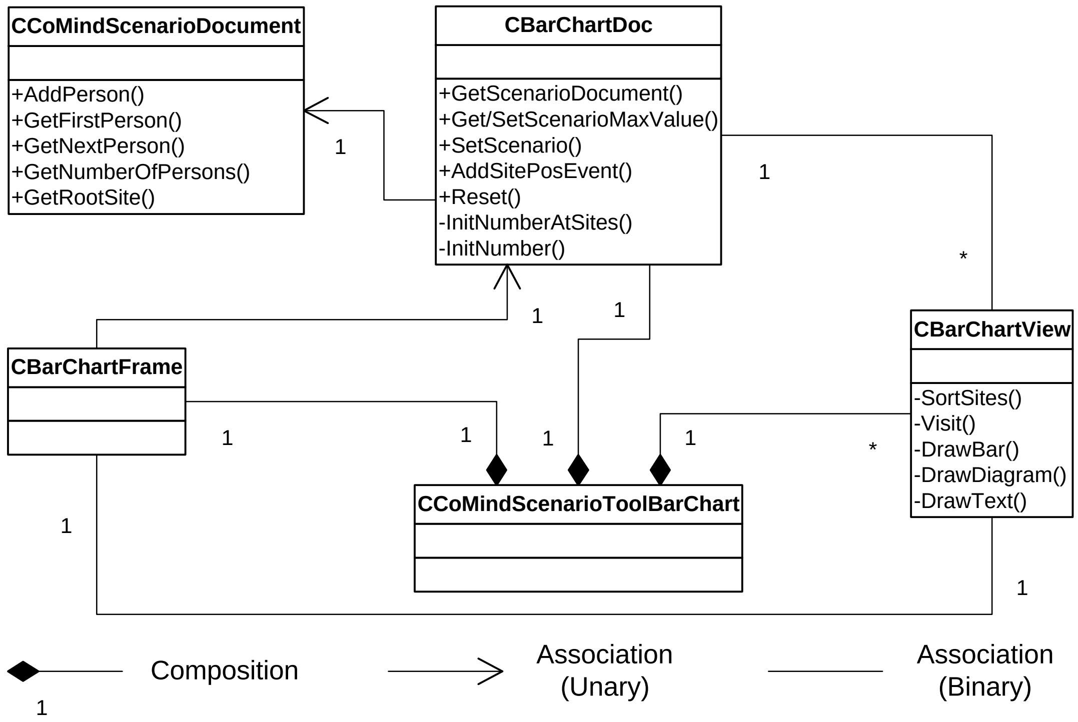


Figure 6-5: Class diagram of the scenario view.

7 Implementation

This chapter opens by briefly describing some implementation issues and ends with a presentation of the functionality of the scenario system.

7.1 The XML Parsers

There are two different XML parsers, where the first is implemented in the scenario source, in order to parse the scenario structure. The second is implemented in the scenario event source, parsing events. They are both implemented by using Microsoft's MSXML parser (Arciniegas, 2002) and their purpose is to create the internal representation of MIND components, needed for replay of the operation.

7.2 Conflicts

The scenario system does not handle any conflicts that may occur, for instance misspellings in the XML structures. In this thesis it is assumed that all input data are correct and that perhaps some higher level tool has been used to create the XML structures, in order to avoid misspellings and erroneous tags. Another problem is the consistency of the collected events, for example if a person gets an "In" event on a site before getting an "Out" on the site the person currently is at. The person will now obviously be in two places at the same time, which can be hard to interpret during replay. There is no easy way to correct errors like this, because assumptions can be very misleading.

In order to detect and solve conflicts like these, an extra process is needed. This lies outside the work of this thesis, but this process should be able work in two stages, where the first one detects conflicts, either online or offline. That is, either during the actual replay or before the replay of the operation begins. The second phase needs some heuristic solutions for solving the conflicts. It is hard to determine what these heuristics should be and a lot of work probably needs to be done in order to find acceptable ones.

7.3 Functionality of the Scenario System

The functionality is described by making a walk-through of the scenario system in the chronological order a presumptive user would follow, when creating a new scenario.

The scenario system needs to be loaded with the right data before it is possible to replay the scenario. This is done by first creating a scenario source, which is capable of importing external data in the form of an XML scenario structure (Figure 7-1). When the import is done, the system has created a scenario document that holds the internal representation of the graph and persons. Each site and person is also created as its own object in the system (Figure 7-2).

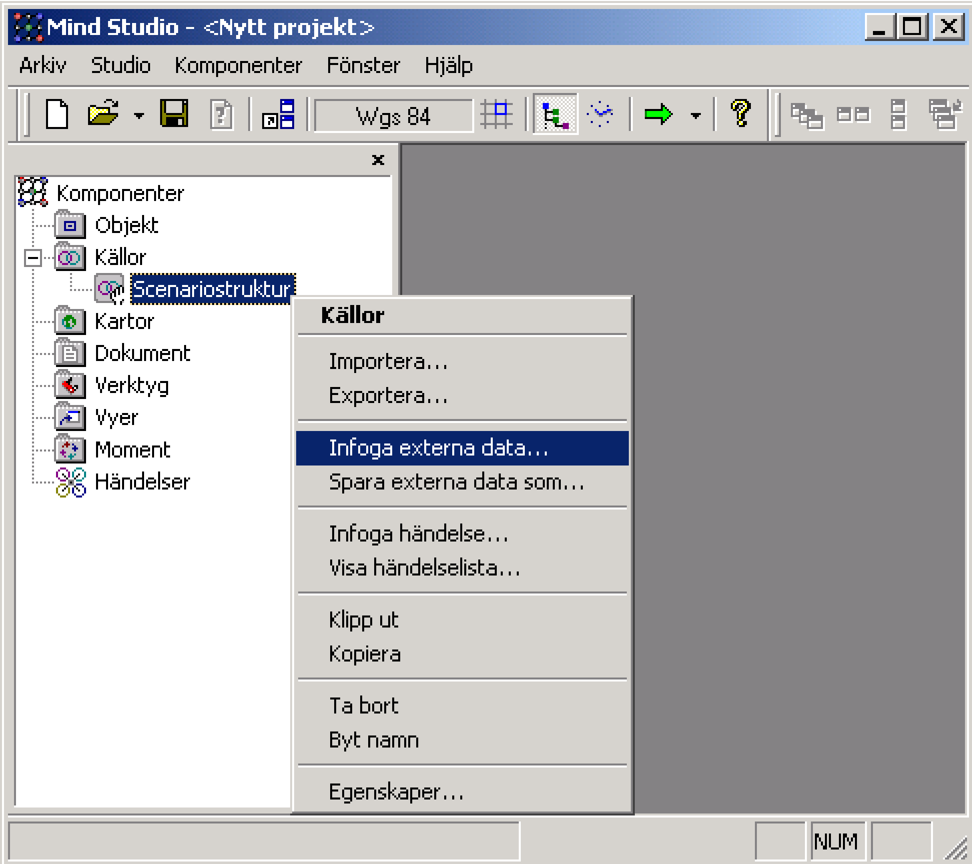


Figure 7-1: Importing a scenario structure.

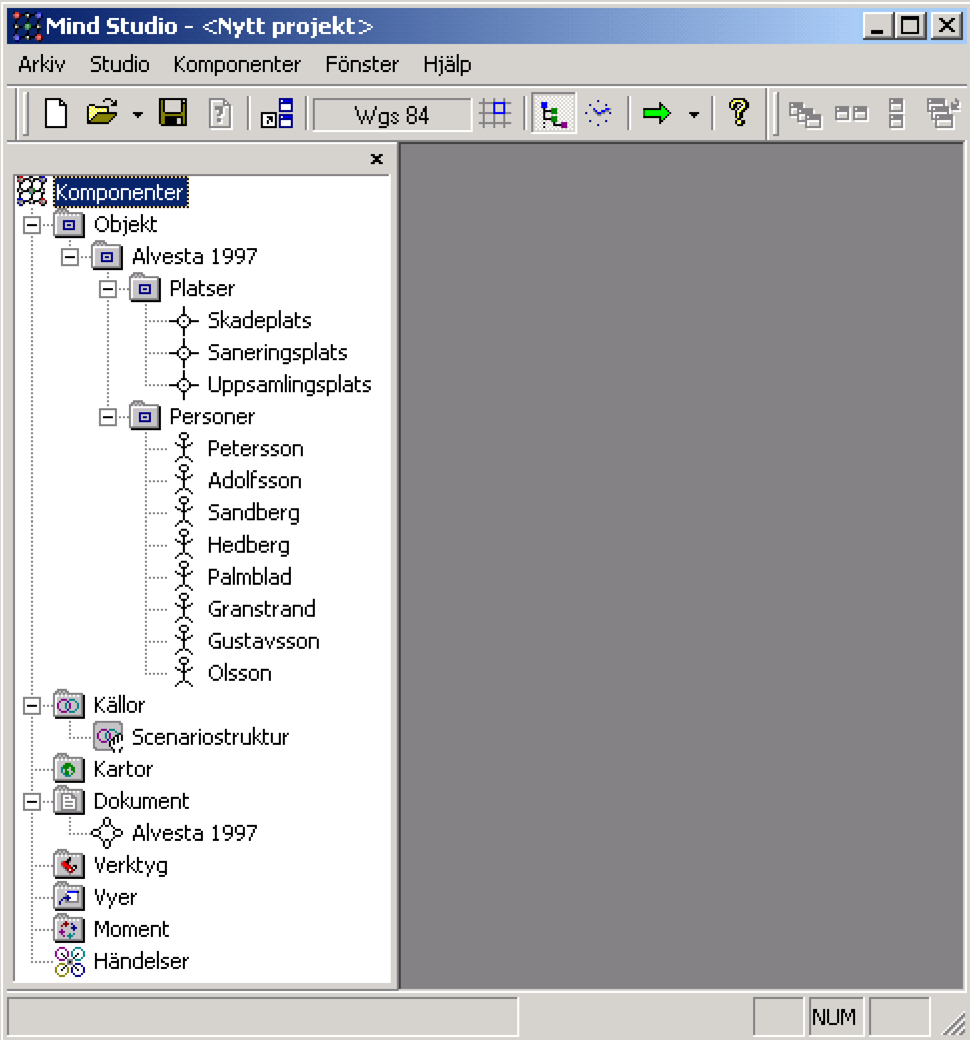


Figure 7-2: The result after importing a scenario structure.

When the scenario structure has been imported, it is time to import the scenario events. In order to do this, a scenario event source first needs to be created. This source is also capable of importing external data, but in the format of XML scenario events (Figure 7-3). When the events have been imported, it is possible to see them all in an event list (Figure 7-4).

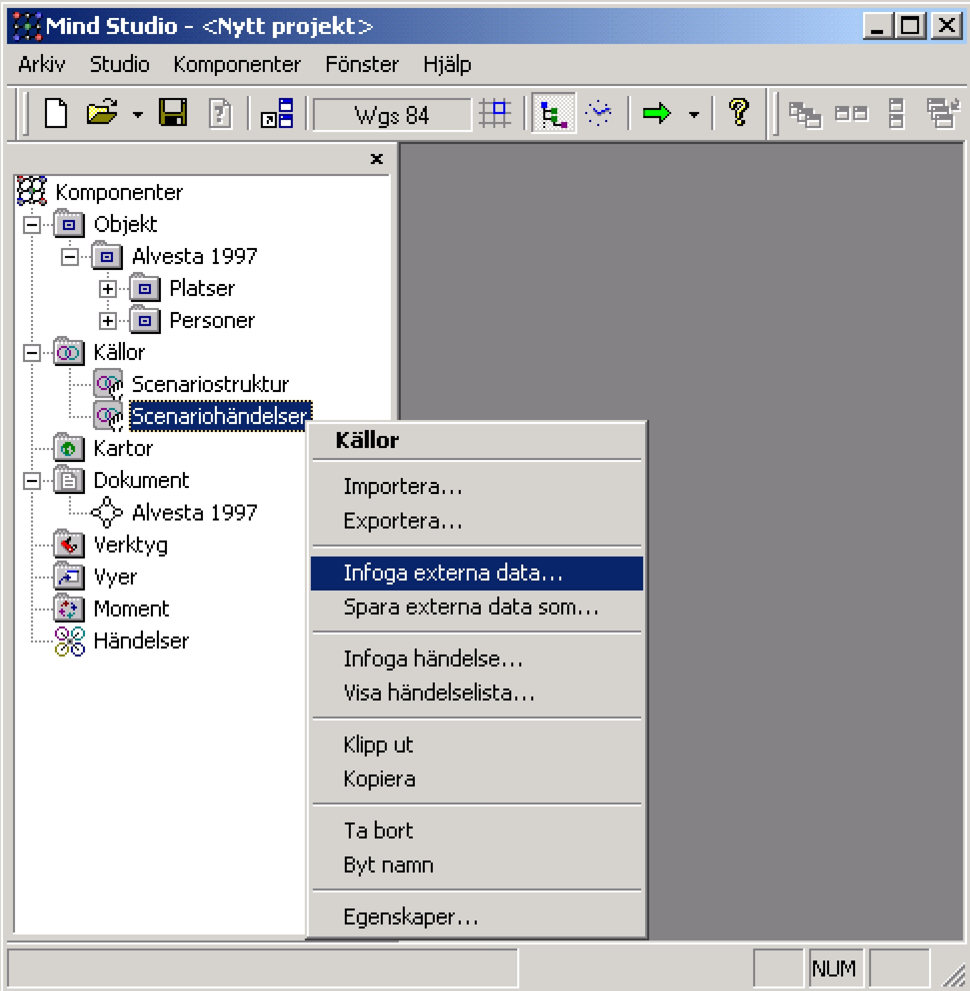


Figure 7-3: Importing scenario events.

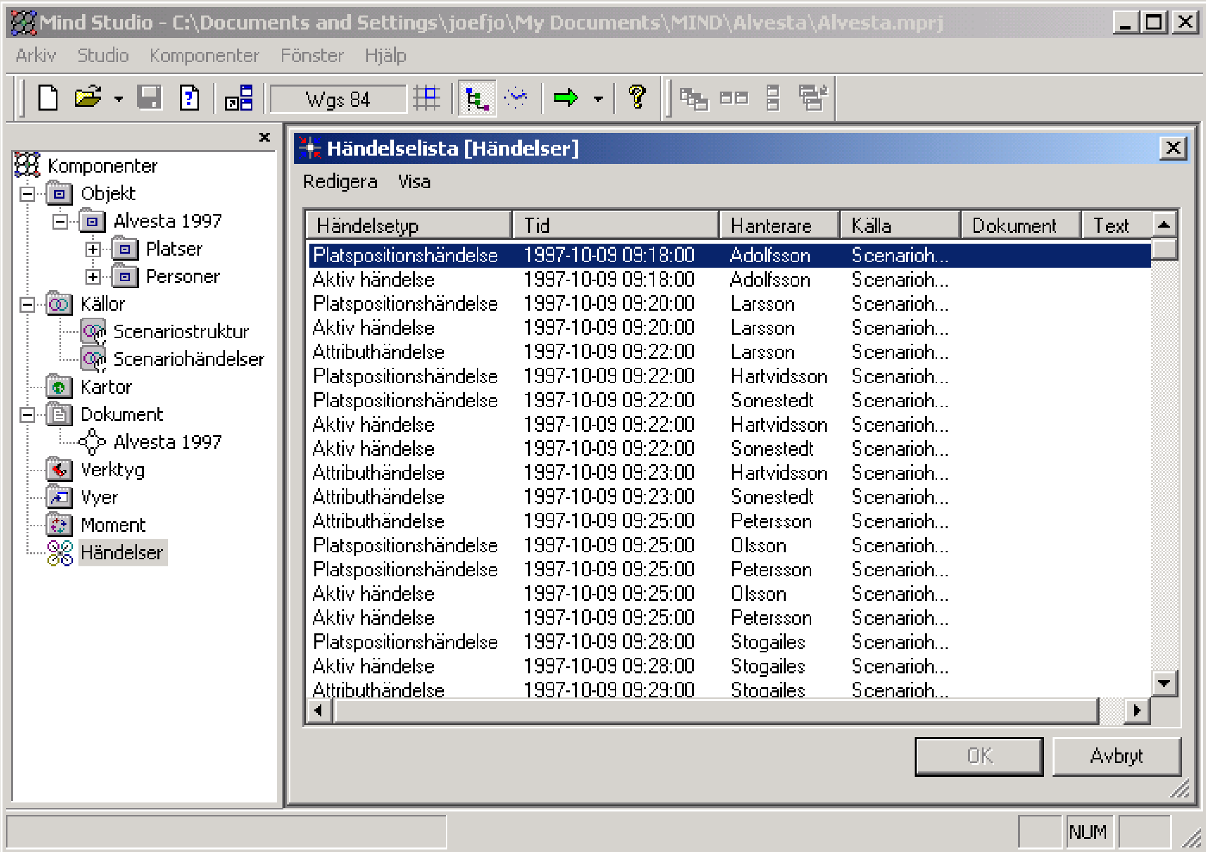


Figure 7-4: The event list showing all events.

Now the system has all the information that is necessary for replaying the scenario, but there is no view created that could present the data. Each object has a specific property page, so even though there is no view created it is possible to see the status of each object by using the property pages (Figure 7-5, Figure 7-6 and Figure 7-7).

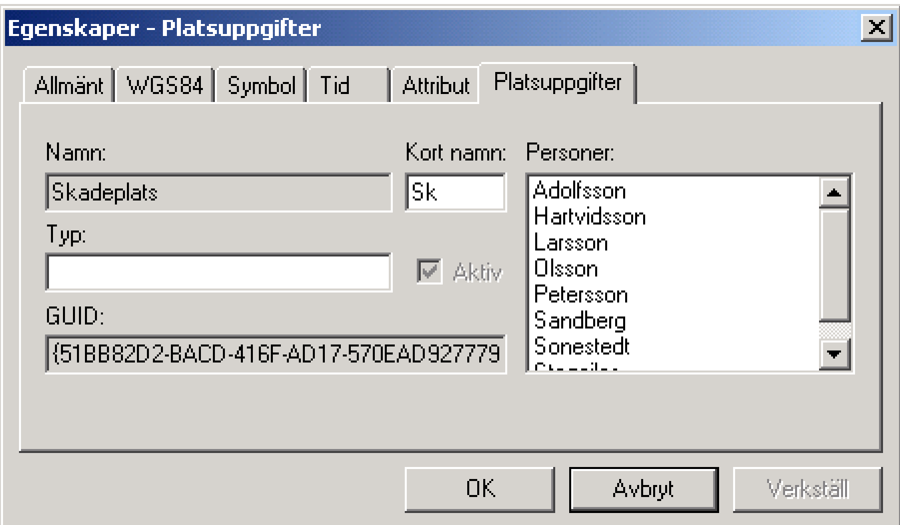


Figure 7-5: A property page showing information of a site.

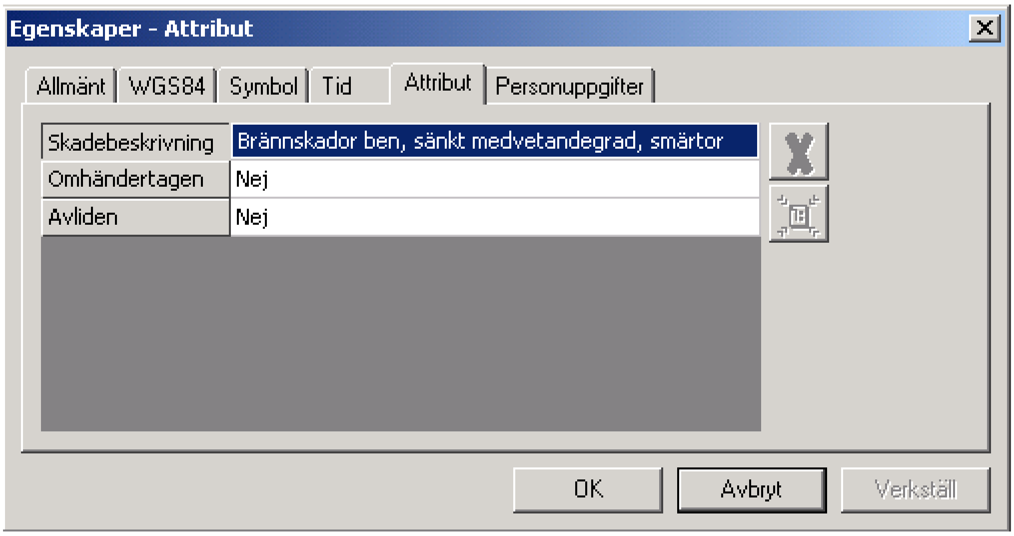


Figure 7-6: A property page showing the attributes of a person.

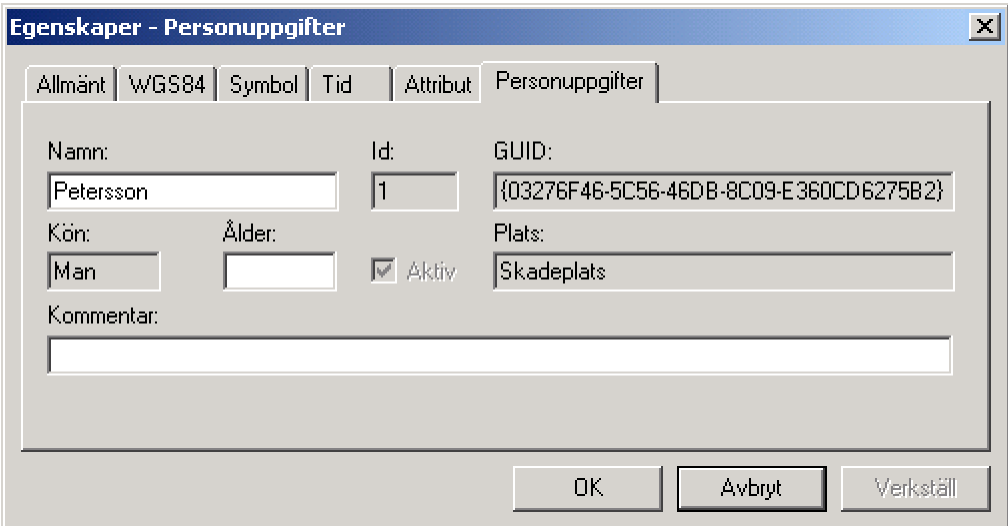


Figure 7-7: A property page showing information of a person.

Finally a predefined view can be added to the system, in this case a bar diagram that shows the number of persons currently at each site (Figure 7-8). It is possible to get a good visualisation of the scenario by using specially adapted views. The scenario system can have any number of scenarios loaded into it and it is possible to choose which one to visualise in the view. The view also has a configuration dialog, where it is possible to choose which sites should be presented in the diagram.

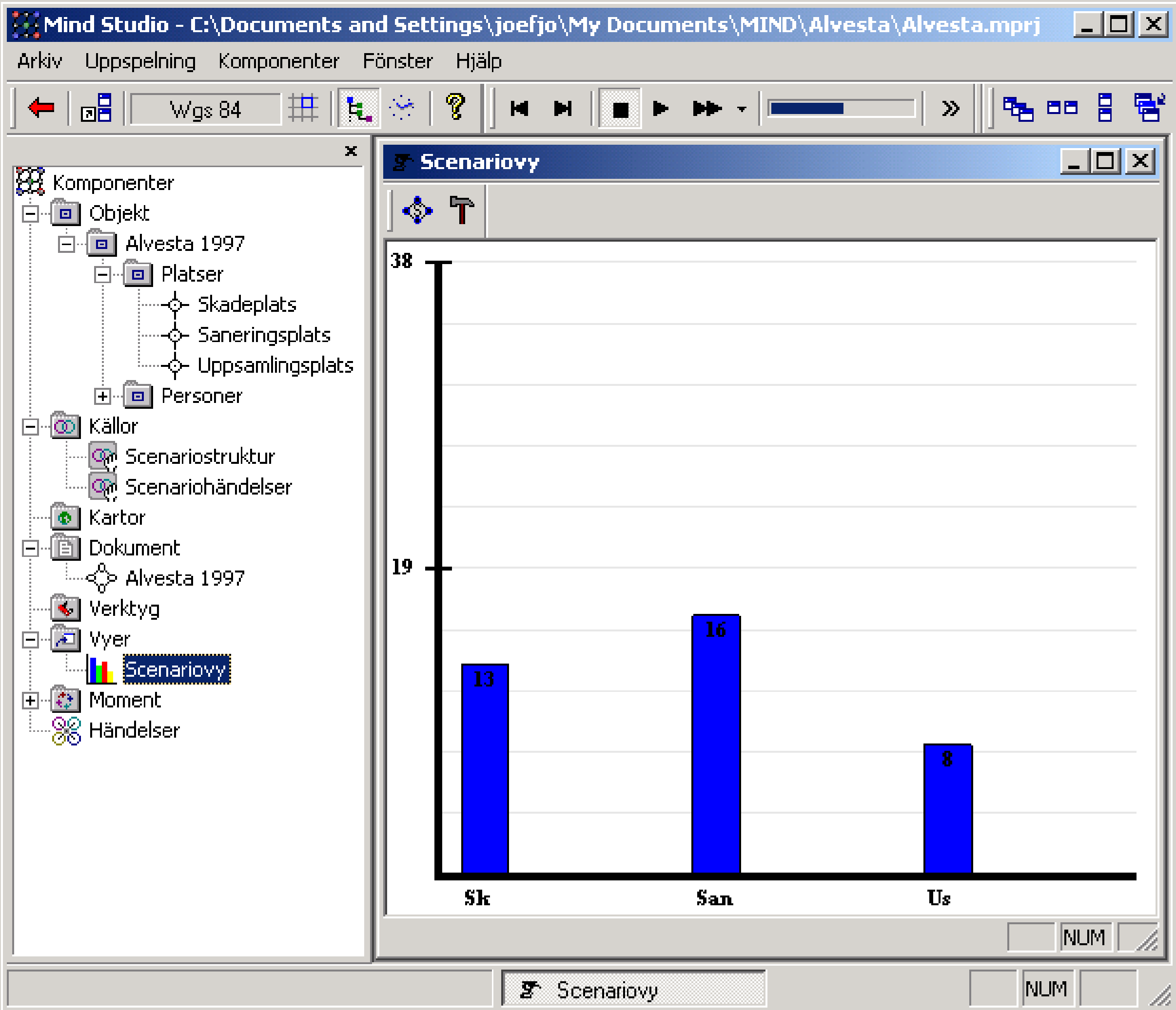


Figure 7-8: A scenario view showing the number of persons at each site.

It is not possible to add a new site or person by simply adding a new object of this type to the scenario system. The reason for this is that the document will not get the required information, like the graph structure and so forth. The only way to add sites and persons is by importing external data. Though it is possible to add new events by hand, this is not recommended for large amount of events, since it can be very time consuming. It can be a handy way to add single events to the scenario though.

8 Conclusions and Future Work

This concluding chapter briefly summarises this thesis and presents some conclusions drawn. It ends by describing some expansions or future work that can be added to this work.

8.1 Summary and Conclusions

The purpose of this thesis has been to develop a system that enables a dynamic approach when it comes to creating and visualising scenarios for rescue operations with casualty flows. Since the developed system only has been tested with one scenario, from a real field-trial, it is hard to say if it can support all conceivable scenarios. It is possible to specify a graph structure, which represents the area of the rescue operation, of any size and complexity and each person can also have any number of attributes. Therefore it is probable that the system will be able to support most of the rescue operations, with casualty flows, that it will be used for. The scenario system will not directly support an operation where it is desirable to evaluate other kind of flows, like ambulances for instance. On the other hand, ambulances transport casualties, which makes it possible to follow the flow of ambulances indirectly.

Advantages with the system are the support for a dynamic specification of scenarios, as mentioned above, but also that the system is prepared for expansions. A disadvantage, on the other hand, is that it has not been proven that the system is dynamic enough to support every kind of scenario. Another disadvantage is the lack of different standard views.

8.2 Future work

There is no end to the extension that can be made to and around this system. First of all, a program could be created for specifying the scenario on a higher level, which in turn generates an XML scenario structure. It could for example be possible to create the graph structure by drawing sites as circles and attaching paths between them by adding lines. There exist programs for drawing graphs and some of them also support the process of saving graphs to file in certain formats, which points at the possibility to convert that file format to the format of the scenario structure.

Another extension is to expand the path structure with a road type and capacity, for example. This implies the possibility to visualise different kinds of terrains, but also to specify if the transport is by boat or by air. A lot of work can also be made to create some custom views that can be used by the more general scenarios.

9 References

Arciniegas, F. (2002). *C++ XML*. Indianapolis: New Riders. ISBN 0-7357-1052-X.

Axelsson, M. (2002a). *MIND: Systemöversikt.* Unpublished.

Axelsson, M. (2002b). *MIND: Programmeringsmodell.* Unpublished.

Birbeck, M., Ducket, J., Gundmundsson, O. G., Kobak, P., Lenz, E., Livingstone, S., Marcus, D., Mohr, S., Ozu, N., Pinnock, J., Visco, K., Watt, A., Williams, K., & Zaev, Z. (2001). *Professional XML 2nd Edition.* Birmingham: Wrox Press. ISBN 1-861005-05-9.

Crissey, M. J., Morin, M., & Jenvald, J. (2001). *Computer-Supported Emergency Response Training: Observations from a Field Exercise.* In Proceedings of the 12th International Training and Education Conference, ITEC 2001, Lille, France.

Grimes, R., Templeman, J., Stockton, A., Watson, K., & Reilly, G. (1999). *Beginning ATL 3 COM Programming*. Birmingham: Wrox Press. ISBN 1-861001-20-7.

Fagerström, J. (1999). *Objektorienterad analys och design – en andra generationens metod.* Lund: Studentlitteratur. ISBN 91-44-00764-7.

Jenvald, J. (1996). *Simulation and Data Collection in Battle Training.* Linköping Studies in Science and Technology, Thesis No 567, Linköping University, Linköping.

Jenvald, J., Morin M., Rejnus, L., & Thorstensson M. (1998). *Computer–Supported Assessment of Emergency Planning for Rescue Operations*. FOA-R--98-00910-505--SE. Defence Research Establishment, Linköping.

Jenvald, J. (1999). *Methods and Tools in Computer-Supported Taskforce Training.* Department of Computer and Information Science Dissertation No. 598, Linköping University, Linköping.

LMV, Lantmäteriverket, (1996). *Handbok till mätningskungörelsen.* Gävle: Trycksam AB.

Morin, M. (1996). *Händelsestrukturen i MIND-systemet.* Technical documentation in the MIND group.

Morin, M., Jenvald, J., & Thorstensson, M. (2000). *Computer-supported visualisation of rescue operations.* Safety Science, 35, 3-27.

Rankin, W. J., Gentner, F. C., & Crissiey, M. J. (1995). *After Action Review and Debriefing Methods: Technique and Technology*. In Proceeding of the 17th Interservice / Industry Training Systems and Education Conference, Albuquerque, New Mexico, 252-261.

Thorstensson, M. (1997). *Structured Reports for Manual Observations in Team Training.* MSc Thesis, LiTH-IDA-Ex-97/64, Linköping University, Linköping.

Thorstensson, M. (1998). *Development of Methods for Support of C^3I Systems Analysis.* Methodology Report FOA-R--98-00837-503--SE, Defence Research Establishment, Linköping.

Thorstensson, M., Morin, M., & Jenvald, J. (1999). *Extending a Battle Training Instrumentation System to Support Emergency Response Training.* In Proceedings of the 10th International Training and Education Conference, ITEC'99, pp. 550-562, Haag, The Netherlands.

Thorstensson, M., Björneberg, A., Tingland, B., & Carelius M. T. (2001). *Computer-Supported Visualisation of an Inter-Agency Exercise in the Stockholm Underground.* In Proceedings of the International Emergency Management Society's Eight Annual Conference, TIEMS 2001, Norway.

Worm, A., Jenvald, J., & Morin, M. (1998). *Mission Efficiency Analysis: Evaluating and Improving Tactical Mission Performance in High-Risk, Time-Critical Operations.* Safety Science, 30(1-2), 79-98.

Örnberg, G., Jenvald, J., Morin, M., & Worm, A. (1996). *The MIND Project 1995-1996.* Status Report FOA-R--96-00288-3.4--SE, Defence Research Establishment, Linköping.

Appendices

Appendix A

The XML Scenario Graph for the rescue operation performed in Alvesta, in 1997.


```
<Scenario>
	<Title>Alvesta 1997</Title>

	<Graph>
		<Sites>
			<Site>
				<Name>Skadeplats</Name>
				<Shortname>Sk</Shortname>
				<Registrations>
					<Registration Type="In">Sk(Funna)</Registration>
					<Registration Type="Out">Sk(Ut)</Registration>
				</Registrations>
				<Position Type="WGS84">
					<Latitude CardinalPoint="North">56.89198</Latitude>
					<Longitude CardinalPoint="East">14.548</Longitude>
				</Position>
			</Site>

			<Site>
				<Name>Saneringsplats</Name>
				<Shortname>San</Shortname>
				<Registrations>
					<Registration Type="In">San(In)</Registration>
					<Registration Type="Out">San(Ut)</Registration>
				</Registrations>
				<Position Type="WGS84">
					<Latitude CardinalPoint="North">56.97149</Latitude>
					<Longitude CardinalPoint="East">14.566</Longitude>
				</Position>
			</Site>

			<Site>
				<Name>Uppsamlingsplats</Name>
				<Shortname>Us</Shortname>
				<Registrations>
					<Registration Type="In">Us(In)</Registration>
					<Registration Type="Out">Us(Ut)</Registration>
				</Registrations>
				<Position Type="WGS84">
					<Latitude CardinalPoint="North">56.97786</Latitude>
					<Longitude CardinalPoint="East">14.566</Longitude>
				</Position>
			</Site>
		</Sites>
```

```
            <Paths>
                  <Path>
                        <Name>Sk-San</Name>
                        <From>Skadeplats</From>
                        <To>Saneringsplats</To>
                  </Path>

                  <Path>
                        <Name>San-Us</Name>
                        <From>Saneringsplats</From>
                        <To>Uppsamlingsplats</To>
                  </Path>
            </Paths>
      </Graph>
</Scenario>
```

Appendix B

This is an extraction from the XML structure of the scenario persons, acting as casualties in the rescue operation performed in Alvesta, in 1997.


```
<Scenario>
	<Title>Alvesta 1997</Title>

	<Persons>
		<Person>
			<Name>Petersson</Name>
			<Sex>Man</Sex>
			<Id>1</Id>
			<Properties>
				<Property Desc="Skadebeskrivning">
					<Type Desc="string" />
				</Property>
				<Property Desc="Omhändertagen">
					<Type Desc="string"/>
					<Initvalue>Nej</Initvalue>
				</Property>
				<Property Desc="Avliden">
					<Type Desc="string"/>
					<Initvalue>Nej</Initvalue>
				</Property>
			</Properties>
		</Person>

		<Person>
			<Name>Adolfsson</Name>
			<Sex>Man</Sex>
			<Id>2</Id>
			<Properties>
				<Property Desc="Skadebeskrivning">
					<Type Desc="string" />
					<Initvalue>Brännskador ben, sänkt medvetandegrad, smärtor</Initvalue>
				</Property>
				<Property Desc="Omhändertagen">
					<Type Desc="string"/>
					<Initvalue>Nej</Initvalue>
				</Property>
				<Property Desc="Avliden">
					<Type Desc="string"/>
					<Initvalue>Nej</Initvalue>
				</Property>
			</Properties>
		</Person>
```

```
        <Person>
            <Name>Sandberg</Name>
            <Sex>Man</Sex>
            <Id>3</Id>
            <Properties>
                <Property Desc="Skadebeskrivning">
                    <Type Desc="string" />
                    <Initvalue>Brännskada rygg</Initvalue>
                </Property>
                <Property Desc="Omhändertagen">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
                <Property Desc="Avliden">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
            </Properties>
        </Person>

        <Person>
            <Name>Hedberg</Name>
            <Sex>Man</Sex>
            <Id>4</Id>
            <Properties>
                <Property Desc="Skadebeskrivning">
                    <Type Desc="string" />
                    <Initvalue>Brännskada benen</Initvalue>
                </Property>
                <Property Desc="Omhändertagen">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
                <Property Desc="Avliden">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
            </Properties>
        </Person>

        <Person>
            <Name>Andersson</Name>
            <Sex>Man</Sex>
            <Id>5</Id>
            <Properties>
                <Property Desc="Skadebeskrivning">
                    <Type Desc="string" />
                    <Initvalue>Brännskador: hals, bröst buk</Initvalue>
                </Property>
                <Property Desc="Omhändertagen">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
                <Property Desc="Avliden">
                    <Type Desc="string"/>
                    <Initvalue>Nej</Initvalue>
                </Property>
            </Properties>
        </Person>
    </Persons>
</Scenario>
```

Appendix C

This is an extraction from the XML structure of the events retrieved from the rescue operation performed in Alvesta, in 1997.


```
<Scenarioevents>
	<Title>Alvesta 1997</Title>

	<Activations>
		<Sample Date="1997-10-09" Time="09:55:00">
			<Person Type="Id">33</Person>
			<Active>True</Active>
		</Sample>

		<Sample Date="1997-10-09" Time="09:50:00">
			<Person Type="Id">55</Person>
			<Active>True</Active>
		</Sample>

		<Sample Date="1997-10-09" Time="09:38:00">
			<Person Type="Id">11</Person>
			<Active>True</Active>
		</Sample>

		<Sample Date="1997-10-09" Time="09:22:00">
			<Person Type="Id">22</Person>
			<Active>True</Active>
		</Sample>
	</Activations>

	<Positions>
		<Sample Date="1997-10-09" Time="09:55:00">
			<Person Type="Id">33</Person>
			<Registration Type="In">
				<Site Type="Name">Skadeplats</Site>
			</Registration>
		</Sample>

		<Sample Date="1997-10-09" Time="10:05:00">
			<Person Type="Id">33</Person>
			<Registration Type="Out">
				<Site Type="Name">Skadeplats</Site>
			</Registration>
		</Sample>

		<Sample Date="1997-10-09" Time="10:20:00">
			<Person Type="Id">33</Person>
			<Registration Type="In">
				<Site Type="Name">Saneringsplats</Site>
			</Registration>
		</Sample>
```

```
        <Sample Date="1997-10-09" Time="10:56:00">
            <Person Type="Id">33</Person>
            <Registration Type="Out">
                <Site Type="Name">Saneringsplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="11:00:00">
            <Person Type="Id">33</Person>
            <Registration Type="In">
                <Site Type="Name">Uppsamlingsplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="09:50:00">
            <Person Type="Id">55</Person>
            <Registration Type="In">
                <Site Type="Name">Skadeplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="09:56:00">
            <Person Type="Id">55</Person>
            <Registration Type="Out">
                <Site Type="Name">Skadeplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="10:20:00">
            <Person Type="Id">55</Person>
            <Registration Type="In">
                <Site Type="Name">Saneringsplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="10:35:00">
            <Person Type="Id">55</Person>
            <Registration Type="Out">
                <Site Type="Name">Saneringsplats</Site>
            </Registration>
        </Sample>

        <Sample Date="1997-10-09" Time="11:35:00">
            <Person Type="Id">55</Person>
            <Registration Type="In">
                <Site Type="Name">Uppsamlingsplats</Site>
            </Registration>
        </Sample>
    </Positions>

    <Properties>
        <Sample Date="1997-10-09" Time="09:56:00">
            <Person Type="Id">33</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>
```

```
        <Sample Date="1997-10-09" Time="11:00:00">
            <Person Type="Id">33</Person>
            <Avliden>Ja</Avliden>
        </Sample>

        <Sample Date="1997-10-09" Time="09:55:00">
            <Person Type="Id">55</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="09:41:00">
            <Person Type="Id">11</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="09:23:00">
            <Person Type="Id">22</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="10:30:00">
            <Person Type="Id">12</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="09:48:00">
            <Person Type="Id">56</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="11:15:00">
            <Person Type="Id">56</Person>
            <Avliden>Ja</Avliden>
        </Sample>

        <Sample Date="1997-10-09" Time="09:40:00">
            <Person Type="Id">23</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>

        <Sample Date="1997-10-09" Time="12:45:00">
            <Person Type="Id">23</Person>
            <Avliden>Ja</Avliden>
        </Sample>

        <Sample Date="1997-10-09" Time="09:50:00">
            <Person Type="Id">34</Person>
            <Omhändertagen>Ja</Omhändertagen>
        </Sample>
    </Properties>
</Scenarioevents>
```

Appendix D

This is an example of an XML scenario graph structure containing hierarchies.


```
<Scenario>
    <Title>Hierarchies Example</Title>

    <Graph>
        <Sites>
            <Site>
                <Name>Incident scene</Name>
                <Shortname>Inc</Shortname>
                <Position></Position>
                <Graph>
                    <Sites>
                        <Site>
                            <Name>Tent 1</Name>
                            <Shortname>Tnt1</Shortname>
                            <Type>Incident</Type>
                            <Registrations>
                                <Registration Type="In">In</Registration>
                                <Registration Type="Out">Ut</Registration>
                            </Registrations>
                            <Position Type="WGS84">
                                <Latitude CardinalPoint="North">44.106367</Latitude>
                                <Longitude CardinalPoint="East">15.34385</Longitude>
                            </Position>
                        </Site>

                        <Site>
                            <Name>Tent 2</Name>
                            <Shortname>Tnt2</Shortname>
                            <Type>Incident</Type>
                            <Position></Position>
                            <Graph>
                                <Sites>
                                    <Site>
                                        <Name>Room 1</Name>
                                        <Shortname>Rom1</Shortname>
                                        <Registrations>
                                            <Registration Type="In">In</Registration>
                                            <Registration Type="Out">Ut</Registration>
                                        </Registrations>
                                        <Position Type="WGS84">
                                            <Latitude CardinalPoint="North">54.106367</Latitude>
                                            <Longitude CardinalPoint="East">14.34385</Longitude>
                                        </Position>
                                        <Date>2002-02-15</Date>
                                        <Time>14:15:13</Time>
                                    </Site>

                                    <Site>
                                        <Name>Room 2</Name>
                                        <Shortname>Rom2</Shortname>
                                        <Registrations>
                                            <Registration Type="In">In</Registration>
                                            <Registration Type="Out">Ut</Registration>
                                        </Registrations>
                                    </Site>
                                </Sites>

                                <Paths>
                                    <Path>
                                        <Name>Tent1-Room1</Name>
                                        <From>Tent 1</From>
                                        <To>Room 1</To>
                                    </Path>

                                    <Path>
                                        <Name>Room1-Room2</Name>
                                        <From>Room 1</From>
                                        <To>Room 2</To>
                                    </Path>

                                    <Path>
                                        <Name>Room2-Dec2</Name>
                                        <From>Room 2</From>
                                        <To>Decontamination site 2</To>
                                    </Path>
                                </Paths>
                            </Graph>
```

```
                        </Site>
                    </Sites>

                    <Paths>
                        <Path>
                            <Name>Tent1-Dec1</Name>
                            <From>Tent 1</From>
                            <To>Decontamination site 1</To>
                        </Path>
                    </Paths>
                </Graph>
            </Site>

            <Site>
                <Name>Decontamination site 1</Name>
                <Shortname>Dec1</Shortname>
                <Type>Decontamination</Type>
                <Registrations>
                    <Registration Type="In">In</Registration>
                    <Registration Type="Out">Ut</Registration>
                </Registrations>
            </Site>

            <Site>
                <Name>Decontamination site 2</Name>
                <Shortname>Dec2</Shortname>
                <Type>Decontamination</Type>
                <Registrations>
                    <Registration Type="In">In</Registration>
                    <Registration Type="Out">Ut</Registration>
                </Registrations>
            </Site>

            <Site>
                <Name>Medical Aid Station</Name>
                <Shortname>MAS</Shortname>
                <Graph>
                    <Sites>
                        <Site>
                            <Name>Station 1</Name>
                            <Shortname>St1</Shortname>
                            <Registrations>
                                <Registration Type="In">In</Registration>
                                <Registration Type="Out">Ut</Registration>
                            </Registrations>
                        </Site>

                        <Site>
                            <Name>Station 2</Name>
                            <Shortname>St2</Shortname>
                            <Registrations>
                                <Registration Type="In">In</Registration>
                                <Registration Type="Out">Ut</Registration>
                            </Registrations>
                        </Site>
                    </Sites>

                    <Paths>
                        <Path>
                            <Name>Dec1-Station1</Name>
                            <From>Decontamination site 1</From>
                            <To>Station 1</To>
                        </Path>

                        <Path>
                            <Name>Dec2-Station1</Name>
                            <From>Decontamination site 2</From>
                            <To>Station 1</To>
                        </Path>

                        <Path>
                            <Name>Station1-Station2</Name>
                            <From>Station 1</From>
                            <To>Station 2</To>
                        </Path>
                    </Paths>
                </Graph>
            </Site>
        </Sites>
    </Graph>
</Scenario>
```