AMSIDO-1

AGENT BASED MICRO WORLD SIMULATION OF INFORMATION DISTRIBUTION IN ORGANIZATIONS

(CASID PHASE)

By

MSc. Egonsdotter Gunilla, gunilla.egonsdotter@mh..se
MSc. Palmius Joel, joel.palmius@mh.se

2002-01-28
Mid Sweden University

TABLE OF FIGURES

EXECUTIVE SUMMARY

The AMSIDO project is a joint project between the Swedish National Defense College (SwNDC) and Mid Sweden University (MSU) since August 2001.

In the beginning the project was called CASID (Cellular Automata -based Simulation of Information Distribution), which was used as a work name during the whole first phase. At the end of the phase it was decided that a better name of the product would be "agent based micro world simulation", thus the name change into AMSIDO, "**A**gent based **M**icro world **S**imulation of **I**nformation **D**istribution in **O**rganizations".

The observed problem underlying the project was that new solutions for information distribution were implemented even when the knowledge about their effects was incomplete or lacking. When formulating the project plan for CASID it was thought that one way to get a better base for decisions would be through simulating the possible information systems in order to see their effects.

The purpose of the project could be summarized: *The purpose with the first phase in the AMSIDO project is to create a practical and theoretical base for a decision support system measuring the projected efficiency of different ways of constructing an information system.* This did in practice mean that the purpose of the first phase was do an overview of what to simulate in an organization, how to simulate it, and then to construct a prototype doing this.

The key method for reaching the goal and the purpose for this project has been an iterative explorative modeling. The method tools for data collection have been questionnaires, deep interviews, videotapes and all types of digital documents from exercises conducted during the year 2001.

During the first phase, the project has produced a number of software applications: Editors for controlling the simulation, the simulation software in itself and visualization layers for the simulation output. The editors produce XML-based descriptions of entities to be simulated in the simulator. The simulator reads these, starts the simulation, and produces output information of what took place during the simulation. The visualization layers reads this output and produces graphical displays and statistics.

At the time of the presentation of this report (January 2002), the basic infrastructure of the simulation is implemented and working, although in a somewhat crude manner. The example simulation runs and produces graphical output, which makes it possible to study the effects of the information points in the simulation.

ITM
Egonsdotter Gunilla
Palmius Joel

INTRODUCTION

During the fall 2001, the first phase of the AMSIDO project has been conducted as a co-operation between Mid Sweden University (MSU), and the Swedish National Defense College (SwNDC).

When the project started, it was called "CASID" (Cellular Automata -based Simulation of Information Distribution), which became the work name during the whole first phase. The idea behind the name was that the project started with the idea of constructing the software with a Cellular Automata perspective, in a manner similar to what had been underlying previous simulation projects, as an example of combat[1].

However, when working with the project, it soon became obvious that there was some confusion regarding the actual meaning of the term Cellular Automata. The product of the project would not follow the strict definition (in which behavior and properties are placed in the underlying coordinate grid itself, rather than in objects moving on top of the grid). At the end of the project it was therefore decided that a better name of the product would be "agent based micro world simulation", thus the name AMSIDO, "**A**gent based **M**icro world **S**imulation of **I**nformation **D**istribution in **O**rganization" for the whole project.

The project was formally a part of AQUA, which in its turn was started as a complement to (or part of) ROLF 2010. The *Mobile Joint Command and Control System 2010* (ROLF 2010) has the purpose to: "test alternative Command and Control (C2) concepts, work forms, decision-making support and C2 organizations adapted to mobile or movable joint C2 systems for the year 2010" [2].

AQUA (which is a name, not an acronym) is a research and development project with the "Visionarium[TM]", a specially designed room with technical infrastructure for supporting dialogue and decision-making, as its main product (see as an example Artman and Persson, for description)[3]. AQUA has over time developed into an umbrella covering several sub-projects concerning specific issues within the science of Command and Control.

The observed problem underlying AMSIDO was that new solutions for information distribution were implemented even when the knowledge about their effects was incomplete or lacking. It was observed that it was difficult to know in advance whether a new tool, such as a new set of computers, would actually help spreading information in a better way, or if it would be a pointless but expensive investment.

When formulating the project plan for CASID it was thought that one way to get a better base for decisions would be through simulating the possible information systems. This way it would be possible to see the effects of different information system structures without actually having to implement them. Thus, the goal of CASID (and later for AMSIDO) was to construct

---

[1] Dockery and Woodcock, 1993
[2] Sundin and Friman, 2000
[3] Artman and Persson, 2000

a prototype software for simulating information distribution within a geographically limited organization.

The use of simulations had successfully been utilized within projects more or less related to AQUA, something that was another source of inspiration. Examples of such projects are STRATMAS[4] and C3Fire[5].

The purpose of the project phase described in this report could be summarized: *The purpose with the first phase in AMSIDO project is to create a practical and theoretical base for a decision support system measuring the projected efficiency of different ways of constructing an information system*. This did in practice mean that the purpose of the first phase was do an overview of what to simulate in an organization, how to simulate it, and then to construct a prototype doing this.

This report summarizes the results of the first phase of the AMSIDO project, or in practice, the results of the effort during the fall 2001. The report will begin with an overview of the background, method and data, and end with a system overview of the software, and a practical guide to installing and testing it.

---

[4] Woodcock, Hitchins and Cobb, 2002
[5] Granlund, 1995

PROJECT FRAMEWORK

This section describes the context of the project, such as involved personnel, organizational belongings and the case scenarios.

Project organization

As mentioned above, the project was a joint project between SwNDC and MSU. The members of the project have principally been the two authors of this report. Apart from these a project leader was stationed at each organization: MSc Anders S. Christensson at SwNDC and PhD Viveca Asproth at MSU. The main initiator and facilitator for the project have been Christensson.

Another important person who was tied to the project (although not as a project member) early during the project was Claes Sundin. Sundin works as a consultant today but has worked for many years in the Swedish defense, and he was one of the authors of *"ROLF 2010 A Mobile Joint Command and Control Concept"*. When he became involved with this project he was working in a parallel project[6] that will hopefully be used in next phase of AMSIDO. His main contribution was in support and expertise for modeling the organization and the behavior of the automata.

The members of the project have a few times during the period had meetings for reporting and generating ideas. During these seminars other researchers (principally members of the AQUA group) have been invited to participate. These meetings have been important parts of the work, since they have brought new ideas to the project.

Exploratory modeling

During the planning of the project a vision, a purpose and a goal was constructed. The project started with an exploratory modeling in the sense of Stoltermann and Lövgren[7]. This method has been the key method during the project. The first step was to make a decision about what parts of reality to study. Next step was an iterative design process of the decided scope. This method can not be seen as strictly scientific but rather as a design or development method. However, this was necessary in order to study the information distribution. The validation of the results during the first phase largely consisted of heuristic evaluation (asking personnel at SwNDC if the results looked believable).

The modeling brought on a significant reduction of the real situation, which was necessary in order to create an operative model usable as a base for the simulation. However, because of this, the reliability and validity of the results could be questioned.

---

[6] Sundin, 2002
[7] Löwgren and Stolterman, 1998

The case scenario

SwNDC annually arranges exercises in strategic command as a part of the Program for Advanced Command Studies. In these exercises, headquarters of command are formed in each of two groups "battling" each other. The exercises aim to represent some sort of crisis during a wartime situation. The staff is modeled according to current North Atlantic Treaty (NATO) standards, which means that the staff is organized strictly hierarchical. During the exercises there are operational and tactical command levels. The exercises try to implement the organization during a situation of crisis, such as during a war, but in a somewhat smaller scale. As an example, a few individuals will sometimes represent a whole staff.

The people participating in the modeled exercise were a very homogeneous group. They have the same type of education; the same military language and good knowledge of the said hierarchical organization. Thus they can easily understand and communicate with each other, something that might differ from a real life situation.

The staffs communicate internally via an information system consisting of a combination of IT tools (email, phones…), meetings and manual routines (messengers, bulletin boards...). One important part of the exercises was to use and test new methods and tools for staff meetings.

Limitation and clarification

The environment and reality delimited in the simulation were the SwNDC premises and the activities of the exercise OPSÖ 01. The plans for the basic simulation was limited to keeping the whole organization restricted to a certain geographical area, such as a single building. This does by necessity exclude some levels of organization. The simulation aims to reflect human behavior on an operation level with information flow from tactical and strategic levels.

It should be noted that when the term "information system" is used in this report, the human individuals using the physical infrastructure are considered part of that system. It was necessary to define the behavior of the humans, which were represented in the simulation. Further, it is important to notice that we can only study behavior, not intentions, thoughts, needs or any other kind of background parameters. Further, only the behavior originating from information collection was of interest in this study.

Outside of the scope of this project was the actual content of information. To actually model the significance and consequences of information content would prove far too complex to possibly handle. Therefore, all material have been studied with the intention to, as far as possible, exclude all information content. The authors realize that this delimitation was a drastic reduction of the situation.

THE DESIGN PROCESS

The work to create the basis for the software has been an iterative design process in the form of an exploratory modeling. This was conducted along two parallel lines: one investigating the organization and the other investigating ways for implementing the simulation software.

During the development process these two lines were a closely integrated effort. Every step led to development of the material and the two parts had to act close together.

The sections below describe both the planned activities and the realization.

Activity line

When the project was planned a time line for the activities was created for the period of the project. At the end of the project, the opinion was that most of the activities had been more iterative than our planning shows.
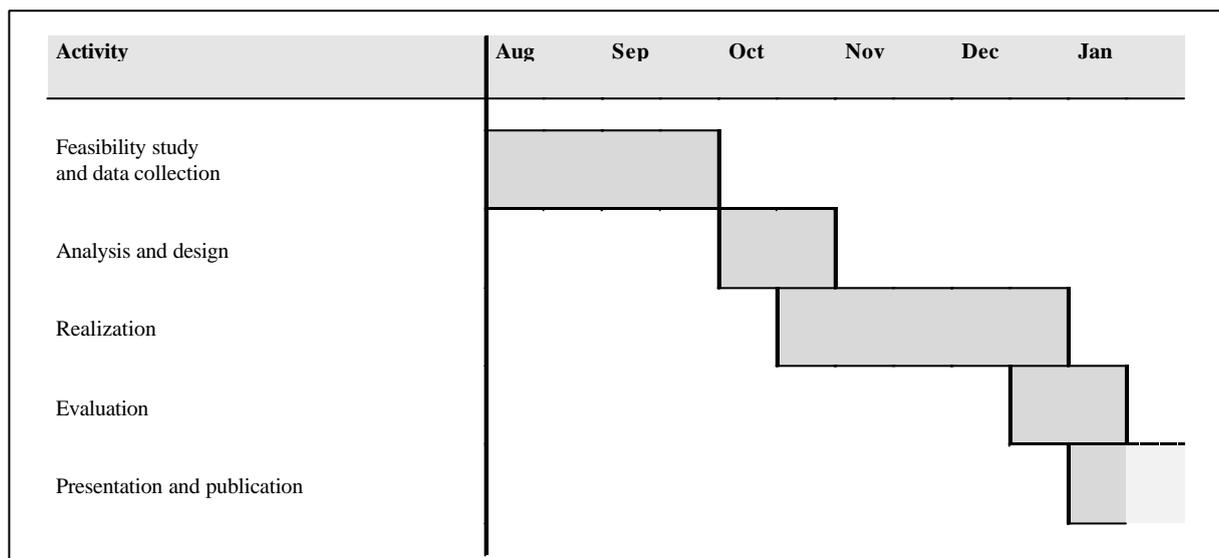
Figure 1. Activity line for the project

Feasibility study and data collection

The first two months of the project consisted of charting, analyzing and making operational observations of information flows in an exercise organization. It was also a time of theoretical studies of relevant literature. In other words, the first part of the project consists of an exploratory phase with the purpose of deciding whether it was reasonable to believe that a simulation could be created. In October '01, it was decided that this was so.

The thought, when the project was planned, was that most of the material would be collected from observations during the performed exercises. However, this was not possible since there were only two small exercises where the Visionarium[TM] was used, and what happened during

these small exercises represented only a small fraction of what takes place when a large exercise is carried out. Instead, most of the methods in generating data in this activity involved reading documents from the exercises MFSÖ01 and OPSÖ01. Videotapes from the Visionarium[TM] show the work in command and control for one part of the exercise (Appendix C). At this stage Claes Sundin became involved with the project as a support for constructing a picture of the behavior in the studied exercise groups.

During the planning of the project a vision, a purpose and a goal was formulated. The first actual step of the project was to create an operative picture in accordance to Löwgren and Stolterman [8]. In practice this consisted of an exploratory modeling of the different entities identified within the situation. This could be viewed as a kind of exploratory research in accordance to Patton [9].

After having constructed a basic operative picture of the situation, the next step consisted of complementing the material with structured deep interviews of exercise leaders, and back-comparisons with the existing documentation. Further, the results had been continually discussed with Sundin and other senior representatives of SwNDC, in order to ensure basic reasonableness.

The results of the back-comparison were fed into a new modeling that was then compared to the documentation in order to form iterative steps in an evolutionary design of the models.

Rather early in the project it was decided that four entities were to become the most important to study. These entities were:

? The **arena entity** described the "physical" side of the simulation, or in other words walls, rooms, stairs and similar. The arena links in information points.
? The **information point entity** was a physical object that can receive, store and transmit information. Example of an information point is a bulletin board.
? The **person entity** was a representation of a person walking around in the arena, looking for information.
? The **information package entity** was a representation of a package of information flowing in the simulation.

The CASID simulation consists of several applications, "talking" to each other via XML files. The simulation application in itself is not very flexible. It will simply read XML files describing the simulation project, and then support very basic functions such as "start simulation", "stop simulation" and "show data from simulation".

The configuration and production of the XML files takes place within a number of editor applications, which constructs the various aspects of the simulation. The data and description foundations are completely written in XML, to allow for a standards-following data exchange procedure. The editor and simulation applications are all written in Java to allow for platform-independence.

---

[8] Löwgren and Stolterman, 1998
[9] Patton, 1990

ITM
Egonsdotter Gunilla
Palmius Joel

Analysis and design

The modeling started with the least complex part, the arena entity. The materiel underlying the modeling was documents and an observation of the building in which the exercises took place.

The next step was the information point entity. The same material as earlier was used and complemented with videotapes from the exercises (Appendix C).

After the first report at SwDNC, the work continued with analysis and design of the organization and of the person automatons that represented the human behavior. The idea on how to represent human behavior was to construct meta schemata, schemata, triggers and instructions as the picture below illustrates.
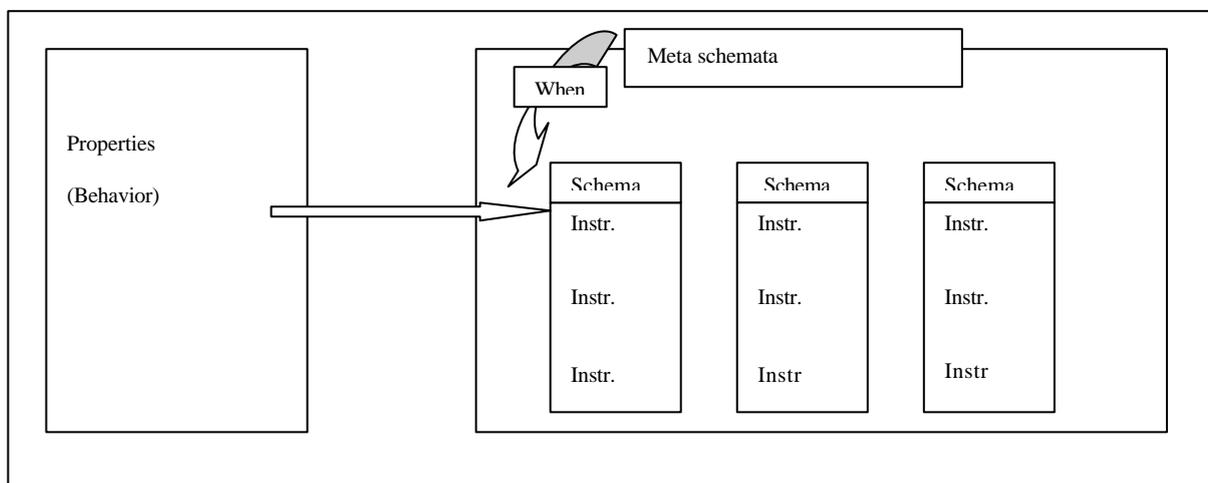
Figure 2. The work model for a construction of a person automaton

The person automaton consists of several properties and a general behavior. The whole behavior is "inherited" from the definition of the behavior group to which the automaton belongs.

The behavior is an aggregate of a number of schemata (such as a schema for "take coffee break"), and a meta-schema controlling the overall schema execution of the schemata. The schemata are triggered through events in the surrounding ("interact with automaton met in corridor"), according to schedule ("take lunch break", "sleep") and according to the overall strategy ("look for information through email in personal office").

The schemata consist of "instructions" or in other words a low-level code language with atomic functions such as "go to area", "wait" and "interact with point". Apart from the behavior, the automatons have properties such as "position in organization", "energy" and "home area".

This part of the project has been characterized by structured deep interviews of officers and modeling in an iterative process. The discussions with Sundin proved invaluable. He

described the staff work over twenty-four hour periods and that description became the base for the behavior of the automatons. The list below described the steps taken in the method for deciding the behavior for the person automatons:

1. Sundin constructs a base for a couple of stereotyped behavior groups for example experts. These groups were created from the functions people have in a ROLF staff.

2. The behavior groups are discussed with a student at SwDCN (a participant from the exercises) and after that the results were further processed. A seminar with a couple of participants was planned, but unfortunately it was impossible to conduct since it was not possible to find volunteers for the task.

3. A couple of behavior groups were determined from the result of the base and discussion.

4. The determined behavior groups were broken down into a couple of short instructions, this was an iterative process.

5. Next step was to create schemata through putting instructions together. The construction of the schemata was based in the behavior groups.

6. Triggers were created in order to execute the schemata.

7. Finally the modeled behavior groups were created by combining different schemata and triggers.
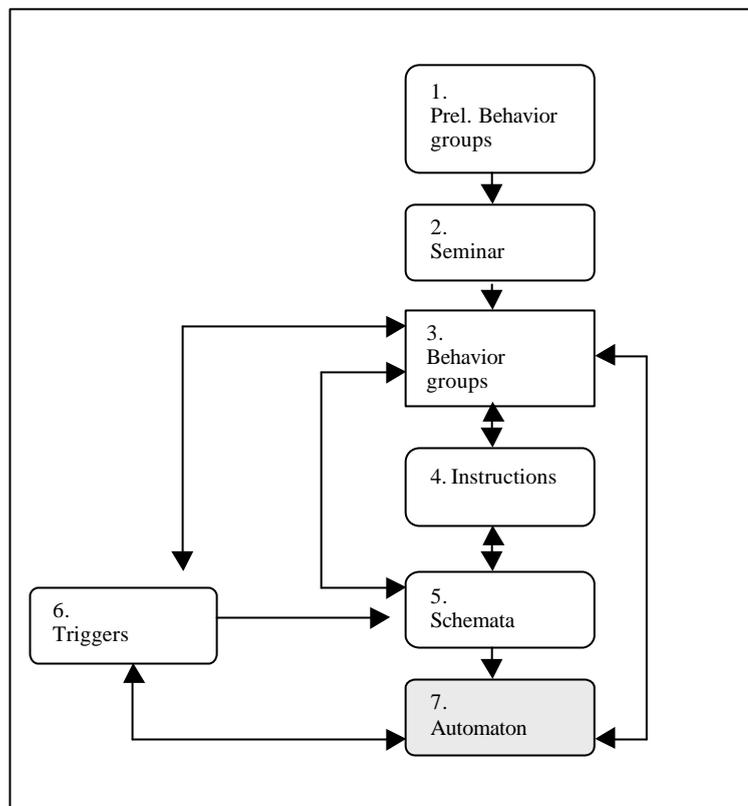
Figure 3. The method for modeling an Automaton.

One part that was not possible to read about in the exercise documentation was the meetings. Of course there was documentation of schedules and participants in meetings, but nothing about what had actually occurred. The material that became the basis for the modeling was instead a report about a study of meetings in the Visionarium made by the consult firm *The Synthetic Family* [10]

Realization

When the overall picture of the organization and different behavior was clear, the work started to create the specific instructions, triggers and schemata in XML code. This was also an iterative process, because the different parts effect and depend on each other.

The entity not as well examined before in the project was the information package, which was now examined through a questionnaire put to participants in the OPSÖ01 (Appendix B). Unfortunately, since a long period of time had passed since the exercise, it became hard to get any answers at all. Six people answered and these answers became a support for the modeling of the information packages. The answers gave a view over the information flow and addresses of the information.
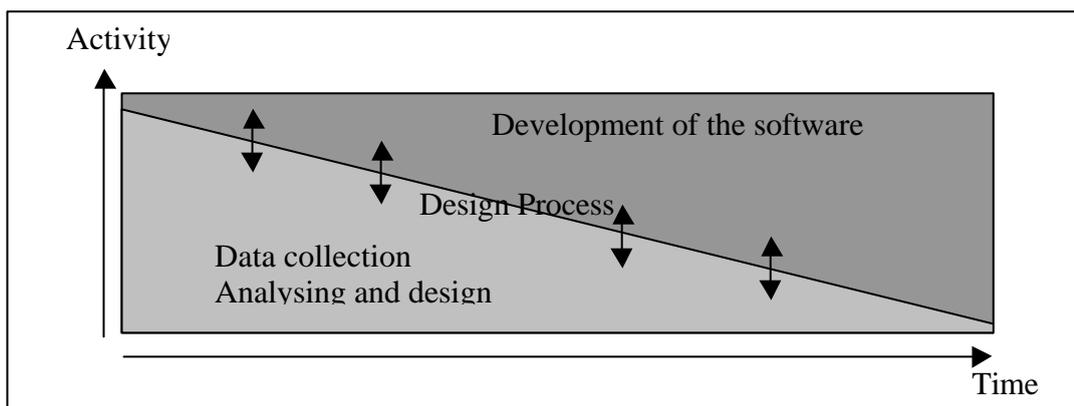
Figure 4. The method for the project, the design process.

Parallel with the modeling, the software was constructed and iteratively compared with the models. A graph of the interaction and extend of the two can be seen in figure 4.

The results of this are described in the system overview later in this document.

---

[10] Lundberg and Gunnarsson, 2001

Evaluation

After the core of the realization was finished, the software was tested and refined iteratively, and compared with the goals stated in the project plan for CASID. This evaluation concentrated mainly on code stability, as a more formal validation is planned for later in the spring.

Presentation and publication

The presentation of the project took place at the SwNDC in late January '02. It was also planned to make a presentation of the first step in this project at a conference in York, July '02. The paper for the conference "*A BASE FOR SIMULATING INFORMATION DISTRIBUTION*" was written in September '01 and describes the feasibility study.

THE PRODUCT

The AMSIDO project has produced a number of software applications: Editors for controlling the simulation, the simulation software in itself and visualization layers for the simulator output. The editors produce XML-based descriptions of entities to be simulated in the simulator. The simulator reads these, starts the simulation, and produces information of what took place during the simulation. The visualization layers read this information and produces some graphs, statistics and displays.

Definitions of concepts

Based on the initial exploratory modeling of the situation, it was (as mentioned above) agreed that descriptions of four entities could provide a sufficient basis for simulating an information system. These were the physical structure, the information points, the person automatons and the actual information. Of course, more objects than these have been modeled, but mainly as a support for the four main entities.

Arenas (physical structure)

The first editor constructed was the arena editor. It makes it possible to create a floor plan drawing. The physical structure was the delimiter of the arena for which the simulation was created. In practice this was a representation of a building or a floor with walls, stairs and doors.
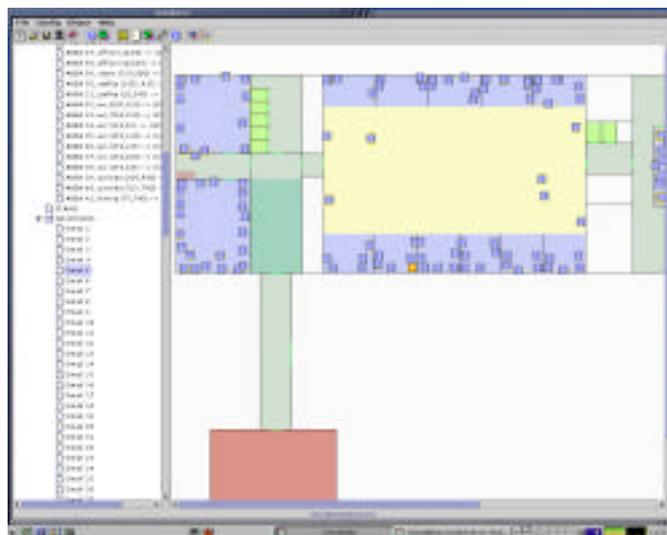

Figure 5. The arena editor

The physical structure describes the "arena" within which the person automatons move and operate. It was noted that a significant part of the information distribution was done in chance encounters within corridors and in the coffee room. Therefore it was decided that the

simulation needs to recapture the movements of the individuals, even when they do not explicitly work with information distribution.

Within the physical structure there were specially designated areas. These can be (as an example) "briefing room", "corridor", "coffee area" and "WC". The special areas do not have properties of their own apart from location and name. Rather, it was the behavior in the person automatons that decides for what purpose the areas were used.

The representation of the arena was modeled in the editor described above. In practice this work consisted of trying to represent a plan sketch of the scenario building. Within the arena, the information points were placed according to the modeled situation.

Information points

The first and most obvious part of an information system was the physical infrastructure, the points through which the information spreads. In this study the information points were defined as "physical objects with the ability to receive, store and transmit information packets".
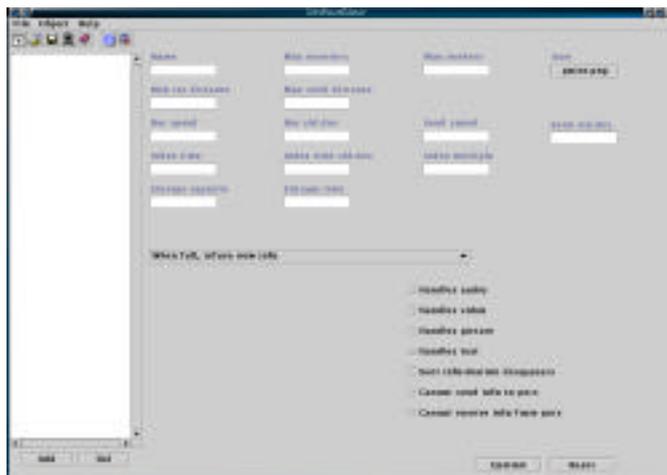
Figure 6. The information point editor

The information points were found to be possible to describe using a few general properties or qualities. The most notable of these were "send rate", "receive rate", "storage capacity", "medium" and "area of influence".

The send and receive rates were measured in packets per second and defines the speed of information transmission between the point and a person automaton. The storage capacity decides how many information packets the point can contain before being full. The medium was which type of information packets the point can handle. As an example, a bulletin board can accept pictures and text, but not sound or video.

The area of influence had two values: one describes how close an automaton must be to receive an information packet from the point, and one how close it must be to add an information packet to the storage of the point.

The videotapes became the basis for the qualities of the information points. It was sometimes difficult to see what was happening on the videotapes, and they do not show all types of information points. Other points mentioned in the matrix were web, printer, fax, bulletin board, whiteboard and information screen. In the case of the later points, we could only use our own experience of the qualities of the points.

Information point	Mail	Phone	Visioscope	Retroscope
Qualities				
Member of recivier/transmitter				
One to one	X	X		
One to many	X		X	X
Many to many				
Handling information				
Transmitted information disapear		X		
Storage information disapear				
Distance recivier/transmitter				
Maxdistance to receiver	1,5m ?			5-6m?
Minidistance to recivier	o,5m ?			1-1.5m?
Could recivie/transmit information from point				
Could transmit	X	X	X	X
Could receive	X	X		
Speed and storage capacity				
Transmit speed	23s-127s	8-144s		9-158s
Storage capacity				
Recivier speed		"		"
Time to find right information	0s-			
Handling this type of media				
Picture/pictures	X		X	X
Text	X		X	X

Figure 7. Matrix over information points

Note however that the above is the theoretical base for the implementation, and that the actual implementation have not got this far. See the system overview for more information.

Information package

The definition of an information package was in this study "a discreet quantity of data". While in the simulation completely devoid of actual content, they *represent* the information in the case scenario. Therefore the term was "information package" rather than "data package".
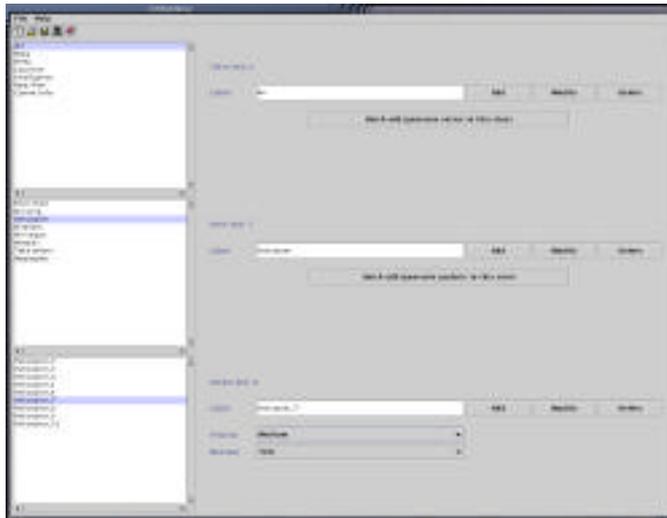
Figure 8. The information package editor

The information packages have four main parameters: A serial number, a type, a priority and a class. The serial number is a plain unique identifier to enable tracing of specific packages. The type is the media through which the information can be spread, "picture", "video", "text" or "audio". This decides through which information points the package can be spread. The priority was "Inconsequential", "normal", "medium" or "high", and provides a base for selection when the automatons had several information packages to choose between.

The class was primarily a way of grouping a stream of information packages, in order to provide a base for statistics after the simulation run. The class also decides whether the information package exists when the simulation starts, or if it had to be constructed through an aggregation of other information packages.

To find realistic information flows, a total of 46 reports from OPSÖ was studied in order to construct a summary. This was not a complete survey of all existing documents, but enough to get a view of the report system in the case scenario.

Information packages from strategical level		
From	Number	To
PJHQ (strategical level)	1	J3 (denied)
PJHQ (strategical level)	1	J4 (debriefing)
PJHQ (strategical level)	6	Control, CJTF (Implementation)
Information packages from OPIL,OP		
From	Number	To
Control ,CTJF	7	ACC, LCC, MCC (SUM,)
Information packages from different units		
From	Number	To
J3	3	ACC (denied)
	10	ACC, LCC, MCC (implementationer)
	2	ACC,MCC
	3	PJHQ (strategical level) (request)
J4	2	PJHQ (strategical level) (debriefing +request)
ACC	2	Control, CTJF (Request)
ACC	1	J3 (clarification)
MCC	1	Control, CJTF (request)
LCC	1	Control, CJTF (request)
Information packages from tactical level.		
From	Number	To
CTG (tactical level)	1	MCC (request)

Figure 9. Summmury over the information flow.

Information packages to every specific unit:	
Unit	**Number**
J3	2
J4	1
OPIL/control	10
ACC	22
MCC	20
LCC	17

Figure 10. The number of information packages a specific unit get.

Person automaton

The by far most complex modeling was that of the behavior of the person automatons.
The person automatons represent the individuals working within the organization. As a
simulation might contain several hundred person-automatons, it was early decided that it
would be impossible to describe all the represented automatons individually. Instead, the
preliminary modeling was directed towards finding possible groups of behavior patterns, and
form these groups into a number of distinct classes. As these groups do not necessarily equate
with a formal function within the organization, the term "behavior group" was chosen rather
than "role" in order to avoid confusion when discussing with personnel at SwNDC.

The person automatons existed in a context, in this case a specific organization: the OPSÖ01 exercise. The exercise organization was rather complex, but the representation that was developed is illustrated in figure 10.

At the top of the figure we see the strategic level, which contained both a political and a military level. The strategic level gives the initial instructions and has continued planning dialogues with the operational level. It also approves the CONOPS. The big arrow illustrates this communication and represents the flow between the strategic and the operative level.

In the middle of the figure the operative level is represented. This level is the main scope of this report. In the operative level there is planning and a realization units. The participants in these units come from the joint forces units, in this case J2-J6 and the different defense units, Air Craft Command (ACC), Marine Craft Command (MCC), Land Craft Command (LCC). The arrows between the different circles represent the participant unit affiliations and also partly the base for the communication between units.

In the realization and planning units it was important to have a close co-operation for a common goal, which is illustrated in the large circles. There was also communication between the two units, illustrated with an arrow.

At the bottom of the figure the tactical level is represented. Between that level and the operational level there had been a large information flow, consisting of orders and information. Therefore it is represented with big arrows for input and output in the limited scope.

The responsibility in the organization was an important issue and therefore it is illustrated with a red line from the tactical level through the operative level to strategic level.
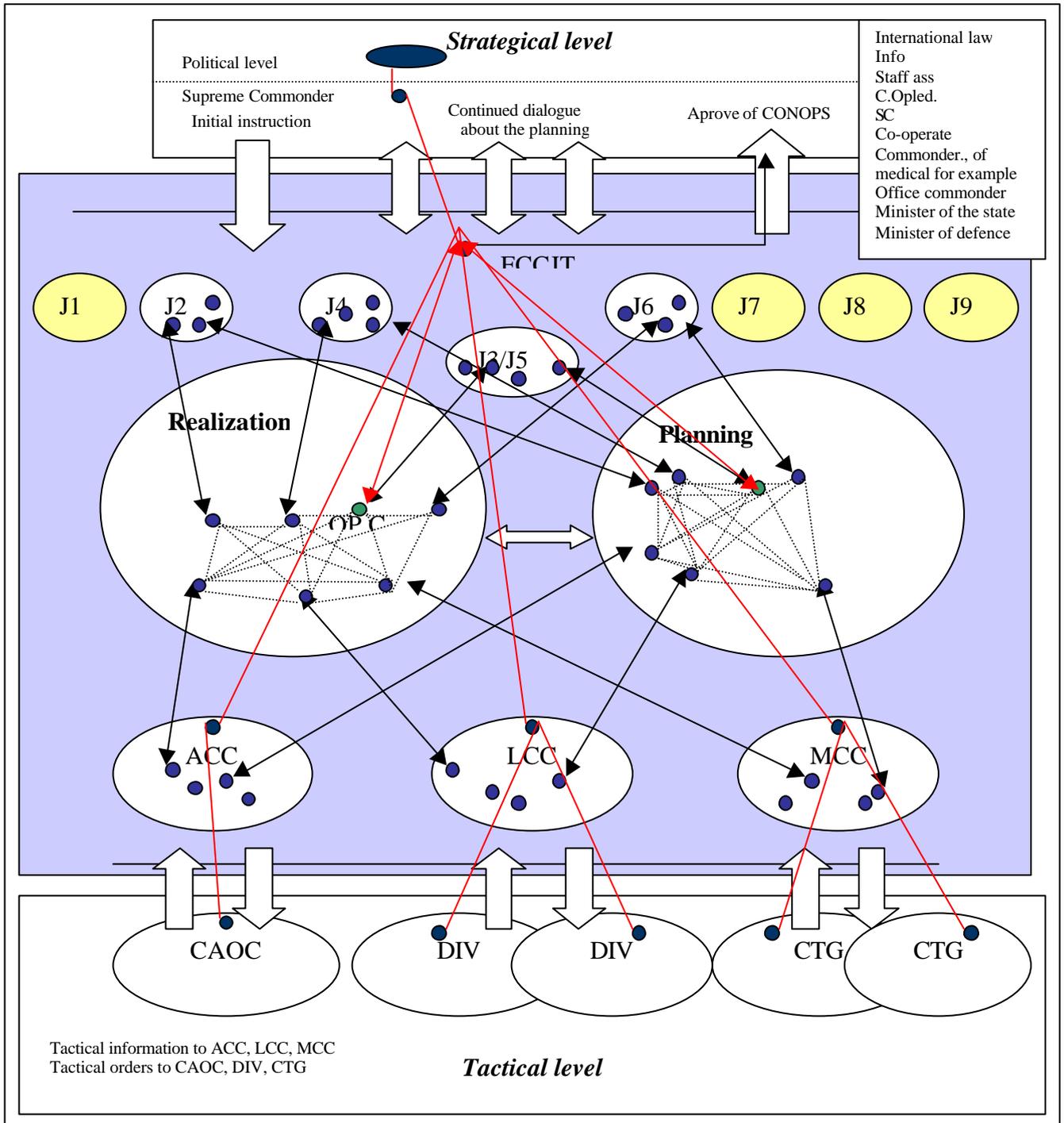
Figure 11. The organization

This picture over the exercise became the base for behavior groups and the information flows. The middle part of the picture, the operative level, shows the organization that the simulation represents.

Seven different behavior groups were constructed to represent the different behaviors in the organization.
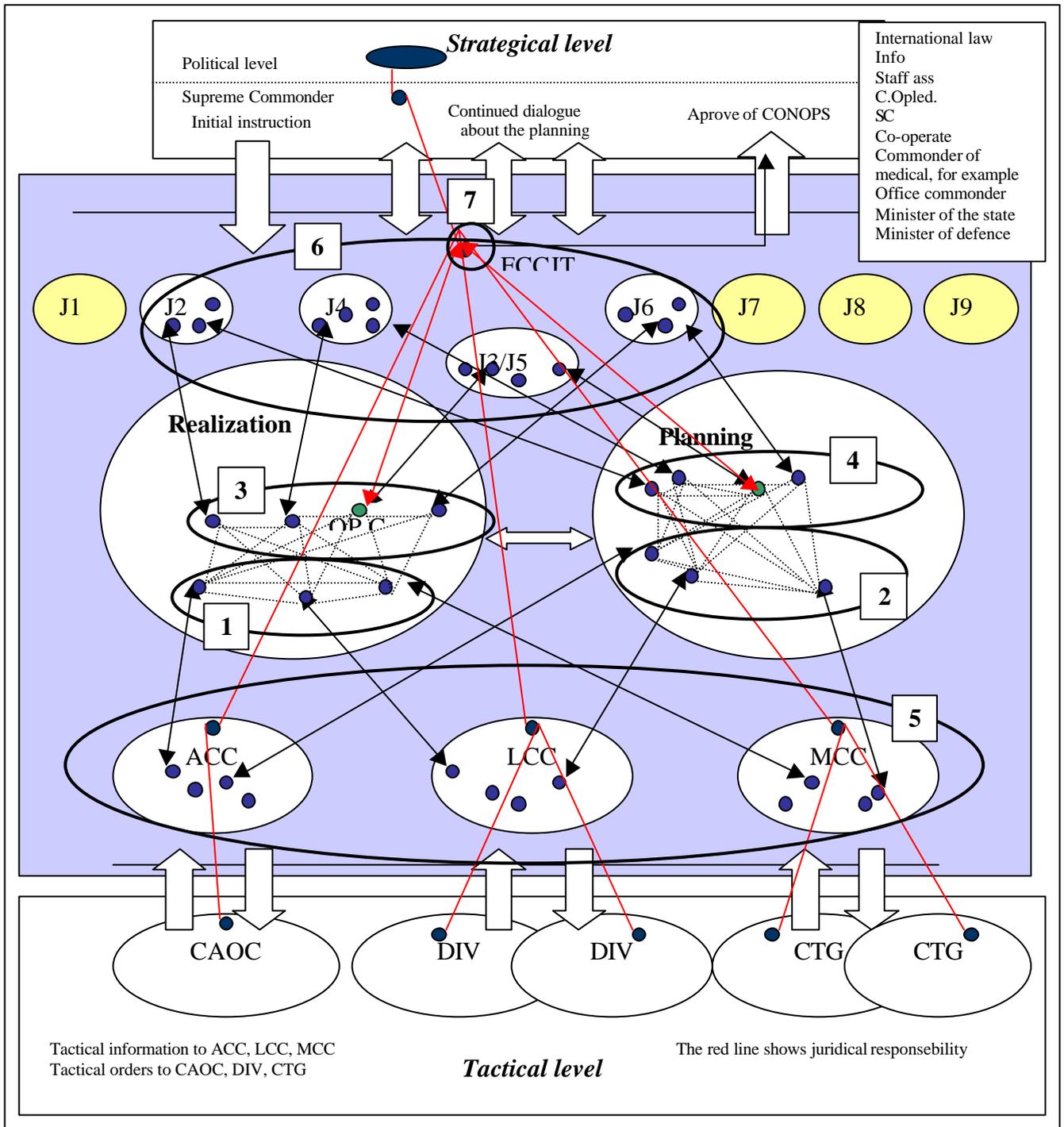
Figure 12. The organization with the behavioral groups

1. Participants from Air, Marine and Land which was part of the realizing unit.
2. Participants from Air, Marine and Land which was part of the planning unit.
3. Participants from different joint units which was part of the realizing unit.
4. Participants from different joint units which was part of the planning unit.

5. Participants from Air, Marine and Land working in their own unit.
6. Participants from different joint units working in their own unit.
7. Special Commander as Commander Operation Instruction Leader (C OpIL)

What was in common for the participants in the specific behavior group was the way they worked, with focus on how they collect and use information. It also gives an indication about what kind of information they wanted.

When constructing the person automatons, a basic data modeling was conducted. The resulting data model can be seen in figure 13.

In the organization there were a number of units like ACC. In this project "belonging" means a unit or function, to which an individual belongs. An individual in this organization always has a position, as for example commander, and he has a home area.

A person also has a behavior and this behavior is a result of the behavior groups. The behavior group is a combination of triggers and schemata. The schemata are aggregations of smaller instructions.
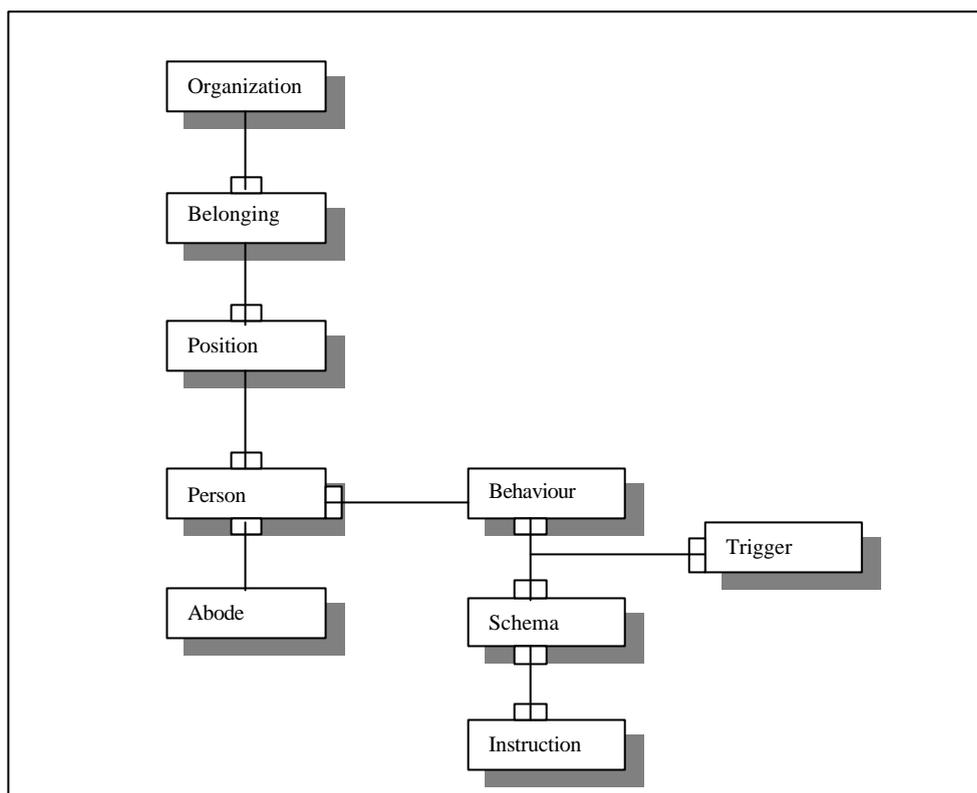
Figure 13. Data model for the Automaton

One example of a schema is "coffee". The schema describes how an automaton moves to an area and stay there for five minutes. The automaton interacts with every other automaton or information point on its way. The two automaton-parameters "energy" and "wc" chance value.

After the coffee break the automaton return to his home area.

```
<SCHEMA ID="coffee">
        <GOTOAREACLASS GOAL="coffee" AUTOMATA="same" POINTS="all" />
        <PASSIVE TIME="300" AUTOMATA="same" POINTS="all" />
        <SETVALUEABS NAME="energy" VALUE="100" />
        <SETVALUEREL NAME="wc" ADD="20" />
        <GOTOAREA GOAL="$source" AUTOMATA="same" POINTS="all" />
        <CONTINUESCHEMA />
</SCHEMA>
```

Figure 14. XML-code, coffee schema

In figure 15 we can see an example of a behavioral group, as represented in XML code. It was rather clear that in this stage of the project the daily schedule was important part of the behavioral groups.

```
<GROUP ID="realization_function">
        <EVENTTRIGGER TYPE="takeover" SCHEMA="overwork" />
        <INFOTRIGGER TYPE="oneinfo" INFO="wanted" SCHEMA="sendinfo" />
        <INFOTRIGGER TYPE="giveorder" INFO="wanted" SCHEMA="order" />
        <TIMETRIGGER TIME="08:00" SCHEMA="form_meeting" PRIORITY="3" />
        <TIMETRIGGER TIME="08:15" SCHEMA="tv" PRIORITY="3" />
        <TIMETRIGGER TIME="08:30" SCHEMA="work" PRIORITY="3" />
        <TIMETRIGGER TIME="11:55" SCHEMA="tv" PRIORITY="3" />
        <TIMETRIGGER TIME="12:10" SCHEMA="bigfood" PRIORITY="3" />
        <TIMETRIGGER TIME="13:00" SCHEMA="work" PRIORITY="3" />
        <TIMETRIGGER TIME="16:25" SCHEMA="tv" PRIORITY="3" />
        <TIMETRIGGER TIME="16:40" SCHEMA="work" PRIORITY="3" />
        <TIMETRIGGER TIME="17:00" SCHEMA="bigfood" PRIORITY="3" />
        <TIMETRIGGER TIME="17:30" SCHEMA="work" PRIORITY="3" />
        <TIMETRIGGER TIME="18:00" SCHEMA="bigstaff_meeting" PRIORITY="3" />
        <TIMETRIGGER TIME="18:55" SCHEMA="form_meeting" PRIORITY="3" />
        <SETVARIABLE NAME="meeting" VALUE="1" />
</GROUP>
```

Figure 15. XML-code, behavioral groups

The exercise documentation, and the other material presented in this chapter, form the base for the summary in figure 16. This became the base for the implementation of the automatons in the simulation. As an example, the "Commander" in function "J2" is located in the room "SÖ", and he belongs to the behavior group "own_function".

Position	**Function**	Location	Count	Behaviour group	Wanted inf.class
Commander	J2	SÖ	1	own_function	Intelligence
Hdl(handling officer)	J2	SÖ	3	own_function	Intelligence
Hdl	J2	203	1	planning_function	Intelligence
Commander	J3/J5	SÖ	1	Own_function	Real/Plan
Hdl	J3/J5	AQUA	1	realization_function	Real/Plan, Intelligence
HKP	J3/J5	AQUA	1	realization_function	Real/Plan, Air
Hdl	J3/J5	SÖ	1	own_function	Real/Plan
Hdl	J3/J5	203	2	planning_function	Real/Plan
Commander	J4	SÖ	1	own_function	Logistics
Hdl	J4	203	1	planning_function	Logistics
Hdl	J4	AQUA	1	realization_function	Logistics, Land 30%
Hdl	J4	SÖ	2	own_function	Logistics
Hdl	J4	SÖ	2	own_function	Logistics
Commander	J6	SÖ	1	own_function	Communication/Information
Ledsavd	J6	SÖ	1	own_function	Communication/Information
Commander	Land	SÖ	1	own_lma	Land. Logistics, Intelligence, Real/plan 10% Marine ,Air 20%
Hdl	Land	AQUA	2	realization_lma	Land. Logistics, Intelligence, Real/plan 10% Marine ,Air 20%
Hdl	Land	SÖ	1	own_lma	Land. Logistics, Intelligence, Real/plan 10% Marine ,Air 20%
Commander	Air	SÖ	1	own_lma	Air, Real/Plan 10%
Hdl	Air	AQUA	2	realization_lma	Air, Real/Plan 10%
Hdl	Air	SÖ	1	own_lma	Air, Real/Plan 10%
Commander	Marine	SÖ	1	own_lma	Marine Real/Plan 10%
Hdl	Marine	AQUA	2	realization_lma	mMrine Real/Plan 10%
Hdl	Marine	SÖ	1	own_lma	Marine Real/Plan 10%
C OpIL	FCCJTF	SÖ	1	control	Aggregat
C Op	FCCJTF	SÖ	1	control	Aggregat

Figure 16. Table over the organization

SYSTEM DESCRIPTION, OVERVIEW AND STATE

The system could roughly be described as consisting of three parts: The configuration editors, the core simulation, and the presentation layers. The configuration editors produce XML code files, which are read by the simulation. The simulation produces state information, which is sent to the presentation layer(s) via interface classes.

The Code as a Whole

As of the report meeting in the end of January 2002, the system consists of a source code hierarchy of about half a megabyte of code, not including the XML files. To give a bit of perspective, this is about 20000 lines, or if printed with 50 lines per page, about 400 pages.

The code is only described in a cursory manner below, as to give an idea of its general structure. However, included on the CD that should accompany this report, you should be able to find a JavaDoc web hierarchy describing the code more closely.

The Configuration Layer

The configuration files consists of XML code, describing thing such as the arena or the behavior of the automatons in the simulation. The main file is the "project" file, which links to the other files.
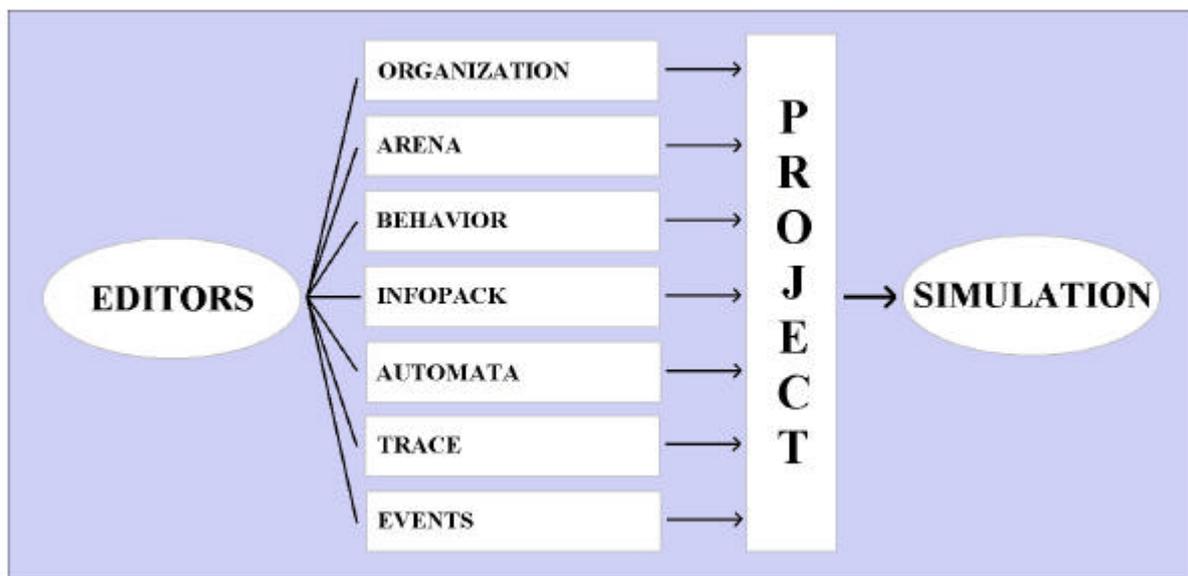

Figure 17. The configuration layer

The **organization** file contains information about organizational units, and available positions. None of these have any significance attached to them, but are rather a base for sorting statistics output.

The **arena** file describes the physical side of the organization: the rooms, walls, doors and similar. The arena file also contains descriptions of information points and types of information points.

The **behavior** file contains behavioral schemata, describing how the person automatons should behave in the simulation. It also contains the available groups of behavior.

The **infopack** file contains all available information packets, ordered in classes and series.

The **automata** file describes all the person automatons included in the simulation. The automatons have properties like where their home area is, to which organizational unit they belong, and to which behavior group.

The **trace** file contains a list of information packets to trace in the simulation, and which color-codes to use when coding them in the visualization.

The **events** file currently controls when and where information should be introduced into the simulation.

In the cases where editors exist, these produce the files. However, as of today, many editors are still unimplemented. In those cases, the XML files will have to be written by hand. For further information about this, see "Unimplemented features", and "User guide".

The Simulation Layer

The simulation layer reads the configuration files and constructs a representation of them. The diagram below is a rather crude description of how the simulation components relate to each other, but should provide a reasonable overview.
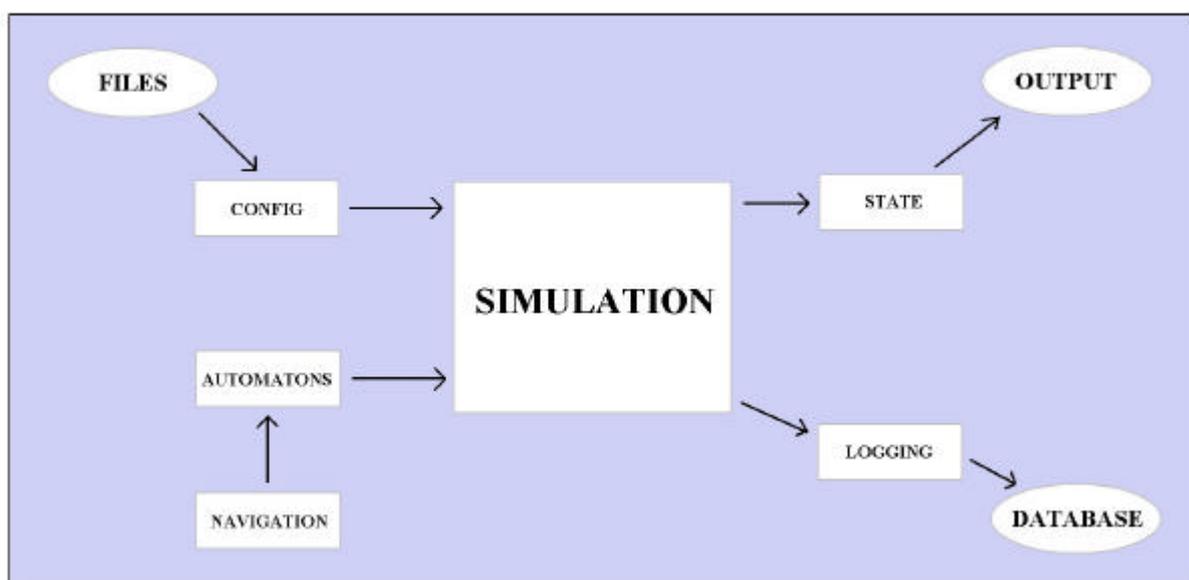

Figure 18. The simulation layer

The parsing of the configuration files result in a configuration object, which handles all static entities, such as the arena. The dynamic entities, such as the person automatons, are in practice handled by a "simulator state" object, but do conceptually act as separate objects.

The simulation works in discreet steps. The state is updated a few times per second, depending on the load of the calculations. During the update, the state information is sent to a relational database as to make later analysis possible.

The heaviest part of the simulation, both in regard to calculation intensity and size of code, is the navigational behavior of the person automatons. The navigational routines are collected into a separate code hierarchy, and are available for the person automatons through a "NavigatorModel" interface.

Every time the simulation has updated its state, a snapshot of that state is sent to the presentation layer(s). These decide how the state information is to be visualized.

The Presentation Layer

There are currently two different presentation layers: one for console mode, and one for graphical output. Both communicate with the simulation core via an interface class called a "SimulatorListener".

The ConsoleSimulator Presentation Layer
The ConsoleSimulator presentation layer currently do not present any information at all, and is as such possibly not a "presentation" layer. The main purpose of the ConsoleSimulator layer is to run the simulation without spending power on heavy visualization calculations.

The GraphSimulator Presentation Layer
The GraphSimulator offers several runtime graphical visualizations of the simulator state. These are implemented in three different sub-windows which themselves implement the SimulatorListener interface.
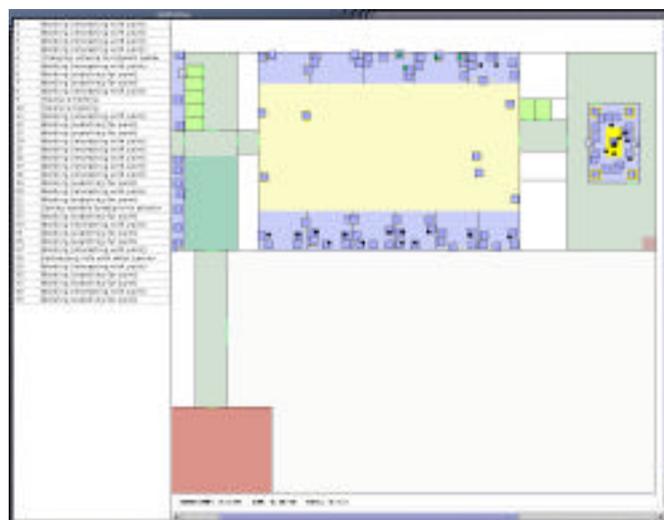

Figure 19. The GeoDisplay

The **GeoDisplay** presents the state of the simulation as a geographical overview of the organization. In this view, it is possible to see how the person automatons navigate and search for information. The automatons are color coded to mark whether they have been reached by an information specifically having been marked for tracing. The information points in the simulation are marked with icons depicting their class (as an example a monitor icon for a TV). The rooms are coded with a color corresponding to
their area class. In addition to the purely geographical view, there is also a table listing the current activities of the person automatons.
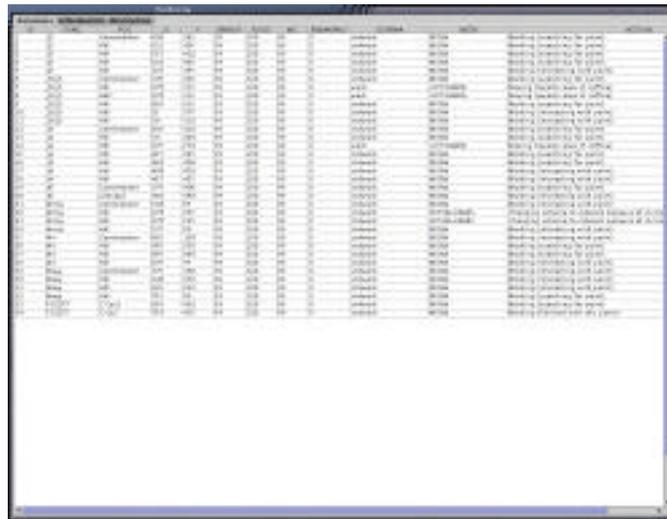
Figure 20. The StateDisplay

The **StatDisplay** describes the state of the simulation in tables with numbers. The StatDisplay is in itself divided into three different tables: The Automata table, the Information table, and the Descriptive table.

The *Automata* table lists all known information about the automata, such as their position, activity, state and organizational belonging.

The *Information* table lists all information packets, and presents statistics about how far they have spread in the organization. In the start of the simulation, no information packets have been introduced. After they have been introduced, they are presented in the form "numbers reached" / "size of target group".

The *Descriptive* table lists some general statistics about the simulation, such as how much time has passed, how many automatons there are all in all and similar.
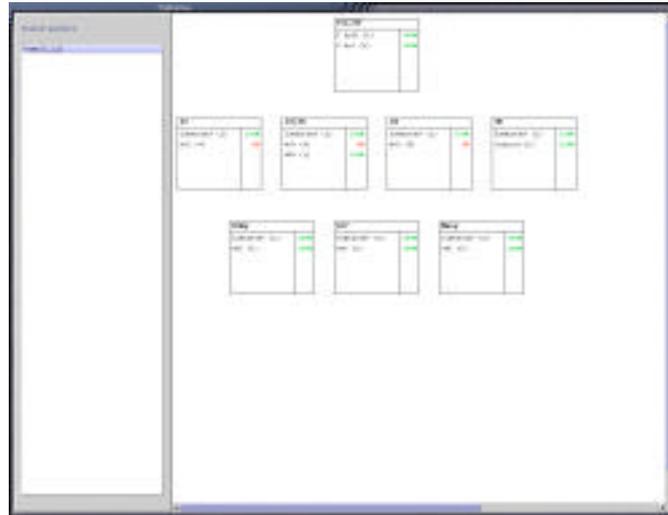
Figure 21. The OrgDisplay

The **OrgDisplay** presents information about how far an information packet has spread through the organization. The display is separated into boxes representing the available organizational units ("functions"). These boxes are then separated further into the positions available. The positions are marked with a number describing how many of the persons of that position having been reached by the information. A table lists all information packets that are being traced.

Implemented features (highlights)

Below are some of the major achievements in the development described.

Working and Reasonably Stable Basics
While there is no doubt at all that there are a number of unknown bugs (some of them likely horrendous) in the code, the basic infrastructure is there, and it is working. The automatons move around, and they find information. It is possible to model and start a simulation. The visualization layer describes what happens in an understandable and correct way.

During the later stages of the development, "crashes" has been very rare. While the code might not always do things in the most efficient (or even most correct) way, it usually does not terminate abnormally.

Open, transparent file formats
All underlying data files are in the XML format. As such, they are human-readable and easy to both import and export to/from other applications.

Modular code structure
Effort has been put into making the code as modular as possible. The presentation and simulation layers are almost completely separated and communicate via a special interface. Thus it is not difficult to add new visualization layers.

Further, most parts of the simulator core are mostly self-contained and possible to communicate with via small interfaces. As an example, the NavigatorModel specifies only four methods, which is enough for the communication with the simulation. The currently used navigation implementation (the "BruteForceNavigator") works well but is very processor-intensive. To exchange this with a new implementation of the navigational routines would not require many code changes inside the main simulator classes.

Unimplemented features (lowlights)

While the project has come far, it is far from finished. There are several issues with features having been planned, but which has not yet made it into the code.

Behavior
The current implementation of the automata behavior is very crude, and there are a few things missing as compared to what was planned.

The most important part missing is that the automata currently ignore the fact that there are differences among the information packets. The automata will accept all information as if it was important to them.

Further, the automata do not realize if they have got all the necessary information. Ideally, they would know this and do something with it (such as send it to someone else).

All in all, the behavior would probably gain significantly from being overhauled and extended.

Information Points and Information
In the same way that the automatons ignore the type of information they receive, the information points also treat all information as being of the same type. While the XML files in fact describe which points should handle what types of information and in what manner, the implementation of information points currently ignore this. Thus, an information of the type "sound" could well be placed on a bulletin board.

All in all, very few of the properties of information points are actually being handled, in spite of having been modeled.

Measurement
In the beginning of the project, it was planned that the efficiency of the information should be measured with a collected value. Currently, no such value exists. The efficiency is currently only possible to read as a combination of for how long an information has been in the system, and how many persons it has reached during that time.

Editors
Currently, there are only efficient implementations of editors for the arena, for the information points, and for the information packets. The other entities in the simulator must be modeled through writing XML code with a common text editor.

Logging

While the logging exists as such, there are no tools whatsoever to read the logs. Further, not all data are saved to the logs. Since there is currently no use for the logs, they are switched off per default.

CONCLUSIONS

The end of the project AMSIDO-1 (or the working name CASID) can only be seen as the end of a first phase in a bigger project. The results consist of prototype software that can crudely demonstrate information flows in a world that had been rather reduction, and of a modeled reality which has not yet been entirely implemented.

Evaluation of the method

During the first phase, the modeled reality was drastically reduced in order to be possible to implement in the simulator. This meant that important aspects like individual differences, information contents and more complex behavior were left outside the software. While being a necessary reduction at the time, we were well aware that this was a reduction with conceivably important implications for the validity of the results of the simulation.

Our opinion is that it had been a better representation if we would have had the possibility to do some observation during the exercises. During the time of the project there was no exercise. Therefore we started an exploratory design process based on existing documentation, and on interviews.

The documentation of the exercises obviously had a purpose other than being used as a base for the implementation of a simulation, but must be described as a bit unstructured anyway. It is difficult to trace the information flow inside the organization only through the documents that is stored in the available databases. The information flows are much more than different types of reports by mail and the presentation of staff meetings. It was often difficult or not possible to see who has done what and when. What has happened outside the technology scope, is another: there is no documentation of phone calls, personal contacts or if someone put a note on a bulletin board.

Another aspect is the question about how general the results can be said to be. As mentioned, the modeled situation is a military exercise organization in crisis (in this case a war situation). The authors cannot, because of this, claim that the results are representative for "normal" organizations.

The project plan in retrospect

The overall goal in the project plan has been to create a prototype for a tool that could be used to simulate information distribution in a limited geographical area. It would also create a possibility to see what is happening when there is some chance in the structure. The product is described as software that demonstrates the distribution in real time and gives a measure of efficiency of the distribution.

The different parts in the software are said to be a physical structure, a logical structure and an organization representation. This part of the simulation is said to contain information points, things like coffee machines and persons.

The product presented in this report is a prototype that makes it possible to see the information flow. While being far from a finished product, it still lives up to the stated goals underlying the initiation of the project.

Further research

The continuation of the project must of course include implementing features that was left out during the first phase, but ideally also a few other aspects, as described below.

? *Validation of the software*:
One of the most important points falling outside the scope of the first phase of the project, was a large-scale quantitative effort to validate the results of the produced software. The software from the first phase should be compared with the real world, and where it is found lacking it should be improved. "Valid" also has some implications for the parts included in the simulation. During the first phase, the modeled reality was drastically reduced in order to be possible to implement in the simulator. In the validation it is likely that it will be found that excluded parts need to be implemented eventually.

? *Extending information point quality*:
A better and way of describing and comparing the qualities of information points is needed. The measurement has from the beginning had a focus on the hardware in the system, as one goal is to be able to discern the value of a single point, or a cluster of points, in regard to the information distribution efficiency. To be able to do such an evaluation, the points have to be described in a fine-grained and comparable manner.

? *Visualization of events and enabling runtime interventions*:
Why and when certain events take place should be possible to see in intuitive graphs, and it should be possible to what-if projections based on interventions in the visualized variables. While the first phase of the project contains basic run-time visualizations, it does not produce much in the order of intuitive event graphs. Such would, however, be a valuable tool for analyzing simulation outcomes.

? *Macro organization*:
It should be possible to connect several simulations or separate organizations as to simulate a larger organization. The basic simulation is based on the limitation that the whole organization should be contained in a geographically limited area, such as a single building. This does by necessity exclude some organization levels. One idea on how to address this limitation is to connect several low-level simulations in order to describe a macro-organization.

? *Evolutionary optimizations*:
The software should be able to calculate the optimal structure of the information system, based on generations of evolving simulations. Currently, the system only projects a likely information distribution based on a modeled organization. It does not automatically find faults in the efficiency of the distribution, and it does not suggest improvements.

? *Extending information content*:

In the first phase of the project it was decided that all information content (apart from priority) should be excluded from the simulation. In order to produce a more fair simulation of the situation, it would be desirable to increase the amount of information content to be included. The simulated information should be more than a meaningless token. It should be able to carry orders, source/path stapling and instructions.

? *Extending behavior and individual differences*:

In order to make the simulation more realistic, individual differences and a more elaborate behavior should be implemented in the automatons in the simulation. Most of the language is in the "move-take-wait-return" level of detail. There are no individual differences in behavior other than those given by different tasks. While this is enough for drawing a picture of the information flows, it is still a drastic reduction of the modeled situation. Further, the reduction of describing all behaviors as inherent in the individual has been felt to remove some important properties of group interactions. A survey would have to be made to see if it is possible to describe groups as entities rather than aggregates of individuals.

REFERENCES

Artman, H., and Persson, M. (2000). *"Old Practices – new technology: Observation of how established practices meet new technology".* From *"ROLF 2010 the way Ahead and The First Step"* (C. Sundin and H. Friman, eds.) The Swedish National Defense College, Department of Operational Studies and the Centre for Command and Control Science, Stockholm, Sweden.

Dockery, J., and Woodcock, A. (1993). *The Military Landscape - Mathematical Models of Combat*. Woodhead Publishing Limited, England.

Granlund, R. (1995). Knowledge Centered Simulation In Emergency Management Training Systems. http://www.ida.liu.se/~reggr/research/myres-09.html [Checked January 2002]

Lundberg, A., and Gunnarsson, H. (2001) *AQUA meeting model, Complementary study I.* The Synthetic Family; Report to SwNDC, Stockholm, Sweden.

Löwgren, J., and Stolterman, E. (1998). *Design av Informationsteknik - Materialet Utan Egenskaper*. Studentlitteratur, Lund, Sweden.

Patton, M.Q. (1990). *Qualitative Evaluation and Research Methods*. SAGE Publications, Inc.,USA

Sundin, C., and Friman, H. (2000). *ROLF 2010 the way Ahead and The First Step.* The Swedish National Defence Collage, Department of Operational Studies and the Centre for Command and Control Science, Stockholm, Sweden.

Sundin, C. (2002). *PEDRAG*, MONOKERES AB; Report to SwNDC, Stockholm, Sweden.

Woodcock, A., Hitchins, D., Cobb, L. *The strategic Manangment System(STRATMAS) and the Deployment of Adaptable Battle Staffs.* http://www.dodccrp.org/Proceedings/DOCS/wcd00000/wcd0005c.htm [Checked January 2002]

APPENDIX A - INSTALLATION AND GETTING STARTED

As the code is currently in an early pre-alpha state, it has not yet been collected into any sensible package, and there is by no means any installer. However, it should not be difficult to get it up and running once the basic requirements have been fulfilled.

It would be a good idea to read "System Description and Overview" before continuing with this.

Basic Requirements

The system is completely written in Java, using J2SDK 1.3. Thus, to run the system, at least J2RE 1.3 is required. If the installation is intended as a base for further developing the system, the whole J2SDK 1.3 package is required. Both J2RE and J2SDK are available from Sun (see link collection below.

Further, most of the backend of the system utilizes XML, via the JDOM implementation. This is an extension of the Java XML classes (JAXP). The system will therefore require that at least JDOM beta-7 is installed and in the class path. This package is included on the CD accompanying this report, but is also available from the JDOM homepage.

To build the source code via the build script (build.xml) in the source hierarchy, you will also need Jakarta Ant, which is a build tool. The installation of this should be reasonably straightforward.

By default, the database logging is switched off, and if this is satisfactory, no database system is needed. However, to use the logging, a RDBMS with transaction capabilities, and a JDBC implementation for this system, is required. The installation of such looks very different depending on the RDBMS, and will not be commented upon further. The connection string and the database URL are set in the project file.

The software can be acquired from:

J2SDK/J2RE: http://java.sun.com
JDOM: http://www.jdom.org
ANT: http://jakarta.apache.org/ant/

Installation

Install Java using the instructions available on Sun's home page. The installation instructions presuppose that this has been done, and that you are familiar with concepts such as "class path".

Install JDOM using the instructions contained in its archive. Take care to check that:
? The jdom, jaxp, xalan, xerces and crimson jar files are included in the class path.
? The jdom.jar file is before all the other XML-related jar files in the class path.
? The jaxp.jar file is before the xalan, xerces and crimson jar files in the class path.

Optionally, install Jakarta-ant using the instructions contained in its archive. Place the "ant" executable in your system path.

On the CD you will find an archive called "casid.tgz". Unpack this where you wish the project to reside.

Add the "classes" folder in the casid folder where you unpacked the casid archive to your class path.

In the casid folder where you unpacked the archive you will find a folder called "source", and within that a script called "startit". Edit this script so that the CASID_HOME path is set to the casid folder.

Having followed these steps, you should be able to start the system.

Running the system
Always when starting the system, the "source" folder under casid should be current directory. On a unix system, "cd" to this directory before following the instructions below.

There are basically two ways to start the system. The easiest is to simply run the "startit" shell script in the source folder. This will display a menu and let the user choose what to start. Note however that several of the options in the menu are unimplemented.

The second and preferable way to start a part of the system is to run the Java class immediately without passing the menu. This is done by running:

```
java -DCASID_HOME=[path to casid folder] CASID.Main [option]
```

...where [option] is one of the following:

      -ArenaEditor
      -InfoPointEditor
      -InfoPackEditor
      -ConsoleSimulator
      -GraphSimulator

Note that these are case sensitive and that the "-" should be included. If "-GraphSimulator" is invoked, a window prompting for displays will be shown.

Changing the Configuration Files
The package includes configuration files for running a simulation of an exercise at SwNDC. Before attempting to remodel the simulation, these should be used as examples and be read through thoroughly. With the exception of arena, infopoints and infopacks, the configuration files must be written by hand.

All files are contained under the XML directory under casid. When reading these, a hint is to start with "project.xml", as that links to the other files. All file paths specified in the files are (for other XML files) relative to the XML directory or (for images) to the source directory.

Working with the Source Code
Before changing the code, please study the JavaDoc web hierarchy on the CD.

The code is easily rebuilt by invoking "ant" with "source" as current directory.

Mid Sweden University   *AMSIDO-1, the CASID phase*
ITM
Egonsdotter Gunilla   APPENDIX B
Palmius Joel

APPENDIX B - QUESTIONNAIRE (in Swedish)

Enkät

Under våren 2001 deltog ni i övningen OPSÖ. Denna övning ligger till grund för enkätundersökningen. Undersökningen kommer att utgöra ett statistiskt underlag vid konstruktion av en mjukvara som skall simulera informationsdistribution i er organisation. Eftersom vi inte kommer att hantera innehållet i informationen, jämställer vi i vissa fall order med information.
Källa är ett begrepp som nämns flera gånger i enkäten och med detta avser vi den informationskanal som medverkat till att ni erhållit/sänt information t.ex. e-mail.

Vi vill att ni försöker göra en så realistisk **uppskattning** som möjligt på de olika frågorna. För att vi skall kunna placera in denna statistik på rätt plats i mjukvaran behöver vi följande uppgifter om er roll i övningen:

Befattning:……………………………………………………

Funktion:.……………………………………………………..

Stabstillhörighet:.……………………………………………..

Avdelning:…………………………………………………...

Fråga 1

Fördela 100% mellan olika informationskällor under övningen, utgå från **antal gånger** du **erhållit** information/order från källan.

Fysisk person		Telefon		Anslagstavla	
E-mail		Mobil Telefon		Whiteboard	
Hemsida		Fax		Visioscope	
Informations-skärm		OH-kanon		Retroscope	Annat

Om annat tilldelats procent, vad är då annat:…………………………………………………

Fråga 2

Fördela 100% mellan olika informationskällor under övningen, utgå från **antal gånger** du **sänt** information/order från källan.

Fysisk person		Telefon		Anslagstavla			
E-mail		Mobil Telefon		Whiteboard			
Hemsida		Fax		Visioscope			
Informations-skärm		OH-kanon		Retroscope		Annat	

Om annat tilldelats procent, vad är då annat:……………………………………………………

Fråga 3

Fördela 100% mellan de olika grupperna, utgå från **antal gånger** du **erhållit** information/order.

	Information	Order
Överordnad		
Medarbetare		
Underställd		
Annan person		

Fråga 4

Fördela 100% mellan de olika grupperna, utgå från **antal gånger** du **sänt** information/order.

	Information	Order
Överordnad		
Medarbetare		
Underställd		

Annan person		

Fråga 5

Uppskatta **antalet gånger** du under **ett** arbetspass vid övningen:

Läst mail?   ………………………………………….

Skrivit mail ?………………………………………….

Studerat en hemsida?…………………………………

Lagt upp information på en hemsida?………………..

Studerat en informationsskärm?………………………

Givit en information via en informationsskärm?……………..

Studerat en anslagstavla?…………………………………..

Satt upp en information på en anslagstavla?……………….

Studerat en Whiteboard?………………………………

Skrivit på en Whiteboard?………………………………

Tagit emot en information från en person?…………………

Givit en information till en person?…………………………...

Tagit emot en information via en telefon?……………………

Givit en information via en telefon?………………………..

Tagit emot en information via en Fax?………………………

Givit en information via en Fax?…………………………..

Fråga 6

Om du tjänstgjort i "Akvariet" under övningen, uppskatta hur stor del av informationen som kommit från :

Retroscope?……………………………………………………

Visioscope?……………………………………………………

Fråga 7

Fördela 100%, med tanke på **var** du generellt befunnit dig när du **mottagit** information/order:

Föreläsnings sal		"Akvariet"		Skrivbord		Formellt möte	
Matsal		Korridor				Annat	

Om annat, var?…………………………………………………………………………..

Fråga 8

Fördela 100%, med tanke på **var** du generellt befunnit dig när du **givit** information/order:

Föreläsnings sal		"Akvariet"		Skrivbord		Formellt möte	
Matsal		Korridor				Annat	

Om annat, var?………………………………………………………………………….

<u>Fråga 9a</u>

Nordland

Markera den avdelning du själv tillhörde med en **cirkel.** Markera de avdelningar du direkt erhöll information från samt sände information till med **pilar**. En **streckad linje** motsvarar **information** och en **heldragen** linje motsvarar **order**.

Mid Sweden University   *AMSIDO-1, the CASID phase*
ITM
Egonsdotter Gunilla   APPENDIX B
Palmius Joel

Sydland

Markera den avdelning du själv tillhörde med en **cirkel.** Markera de avdelningar du direkt erhöll information från samt sände information till med **pilar**. En **streckad linje** motsvarar **information** och en **heldragen** linje motsvarar **order**.

SÖ-hallen

Akvariet
1 milplan

J3/5 bemannar akvariet.
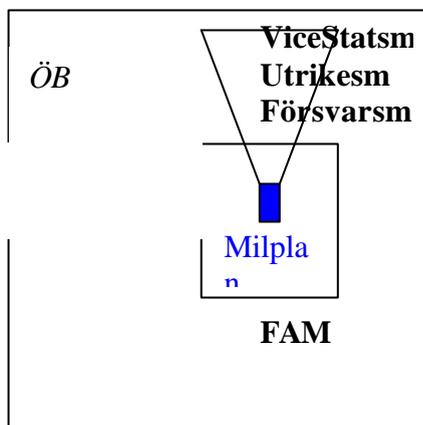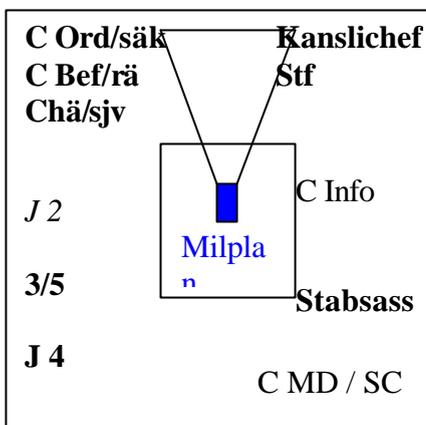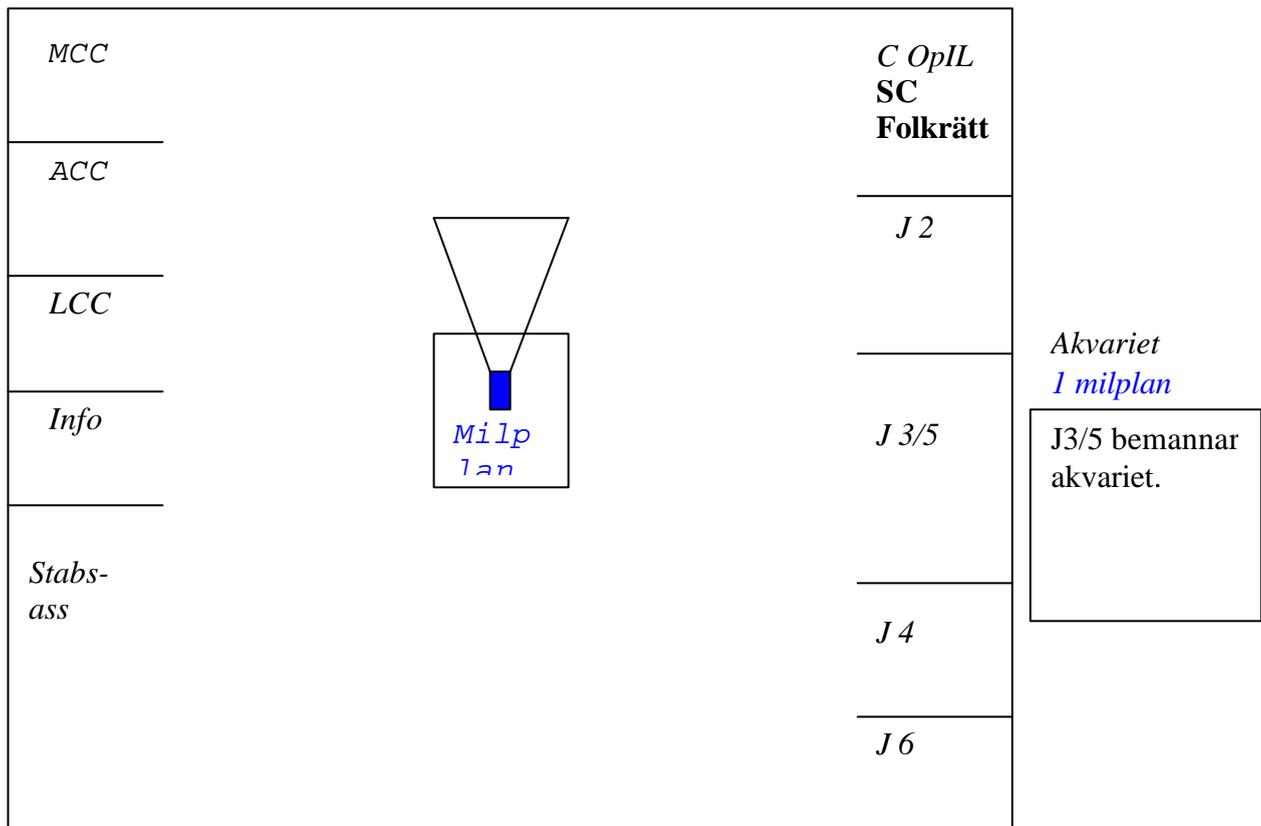
Fråga 10

Fördela 100%, mellan de olika avdelningarna/enheter du **mottagit** information/order från.

ÖB		MCC		J6		Vice Statsm	
C OpIL		ACC		C Ord/		Försvarsm	
SC		LCC		C Bef/rä		Utrikesm	
Folkrätt		J2		Chä/sjv			
Info		J3/J5		Kanslichef Stf			
Stabsass		J4		FAM			

Fråga 11

Fördela 100%, mellan de olika avdelningarna/enheter du **sänt** information/order till.

ÖB		MCC		J6		Vice Statsm	
C OpIL		ACC		C Ord/		Försvarsm	
SC		LCC		C Bef/rä		Utrikesm	
Folkrätt		J2		Chä/sjv			
Info		J3/J5		Kanslichef Stf			
Stabsass		J4		FAM			

Vi tackar dig för din medverkan

Gunilla Egonsdotter
Joel Palmius

Mid Sweden University   *AMSIDO-1, the CASID phase*
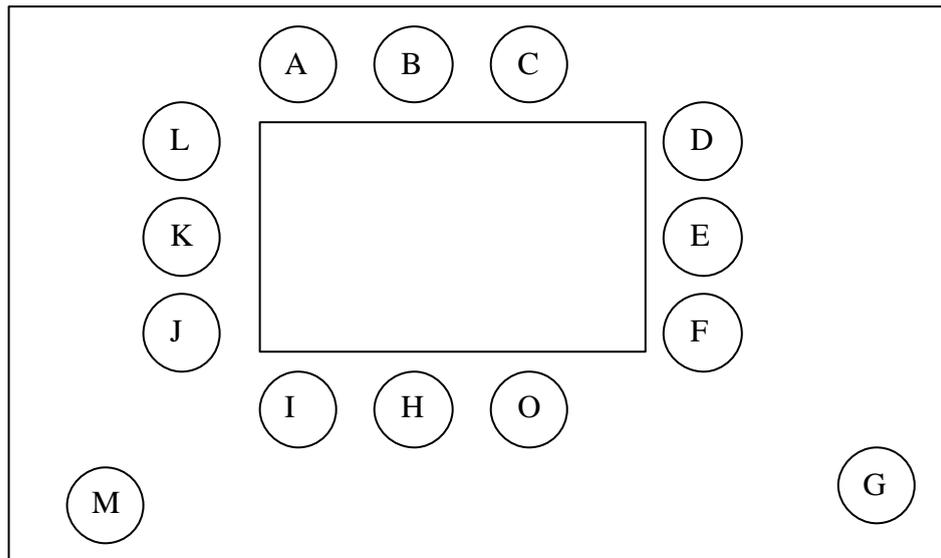ITM
Egonsdotter Gunilla   APPENDIX C
Palmius Joel

APPENDIX C – EXAMPLES OF VIDEO PROTOCOL (in Swedish)

Videoinspelning 2001/06/12
Tak
8:05:55- 8:40:03

Person som kommer utifrån med information betäcknas X
I rummet ges och tas information kontinueerligt 1:M
8:20:04-8:20:18 sker diskussionen i två och två grupper.1:1
X :
8:23:51-8:24:29 Ger information använder Visioscope+muntligt

A:
Blädrar i papper hela tiden, söker infomation, tar förmodligen in information,
8:14:14 Informerar med pappersinf.
8:16:47-8:17:01 mot datorn
8:20:36 Får pappersinf från X
8:22:43-8:22:25 mot datorn
8:23.30 mot datorn
8:23:51 lämnar rummet

B:
8:20:18 Ber K att få se bild (pappersinf.)
8:30:49-8:31:25 Lämnar rummet

C:
8:14:03-8:14:33 mot datorn
8:24:45- 8:25:38 mot datorn

D:
8:34:49- 8:35:07 lämnar information muntligt +retroscope

E:
8:22:35-8:31:25 Lämnar rummet
8:32:21 Lämnar rummet

F:
8:06:10 Mot Visioscope
8:06:25-8:07:58 mot datorn
8:08:08 Utbyter information med F.
8:12:14-8:15:49 lämnar rummet
8:15:49 ger muntlig information
8:19:52 mot datorn
Sammanfattar information vid 2 tillfällen.

G:
8:13:30 muntlig kommunikation med O
8:13:57 lämnar gemensamt rummet med O

H:
8:12:49 Informerar muntligt med stöd av Visiscope
8:16:46- 8:16:54 Informerar muntligt med stöd av Visiscope

I:
8:06:31-8:06:47 lämnar rummet
8:09:01-8:09:11 ger muntlig information
8:17:41 ger skriftlig infomation till K
8:22:16-8:22:19 får skriftlig inf. av K lämnar rummet

J:
8:12:59 Hänvisar gruppen till Retroscope

K:
8:06:08-8:06:25 aktiv muntlig information
8:06:59 aktiv muntlig information
8:14:00- 8:15:11 lämnar rummet
8:15:11 lämnar muntlig information
8:17:41 lämnar skriftlig infomation till I
8:31:16- 8:32:48 lämnar information med stöd av Retroscope.

L:
8:06.27-8:06:52 aktiv muntlig information.
8:22:40 lämnar rummet

Mid Sweden University *AMSIDO-1, the CASID phase*
ITM
Egonsdotter Gunilla   APPENDIX C
Palmius Joel

M:
8:07:00- 8:07.22 Lämnar skriftlig information till I.

N:
8:07:56-8:08:05 Lämnar skriftlig information till I.

O:
8:12.35- 8:12.44 mot datorn
8:13.13 Muntlig kommunikation med G
8:13:57 – 8:20:18 lämnar rummet tillsammans med G
8:20:18- 8:20:52 mot datorn
8:25:23 –8:27:04 mot datorn (diskett för retroscope?)
8:33:33 mot datorn
8:36:35- 8:37:09 lämnar rummet

Video………………………………….
Tid……………………………………..

Informationspunkt/namn	E-mail	E-mail	E-mail	E-mail	E-mail	E-mail	E-mail
Kvaliteér							
Antalet mottagare/sändare							
En till en							
En till många							
Flera till flera							
Hantering av information							
Sänd information försvinner							
Lagrad information försvinner							
Avstånd mottagare sändare							
Maxdistans till mottagare							
Mindistans till mottagare							
Kan sända/ta emot information från punkt							
Kan sända information							
Kan ta emot information							
Hastighet och lagringsförmåga							
Sändningshastighet							
Lagrings kapacitet							
Mottagningshastighet							
Tid att hitta rätt information							
Hanterar dessa typer av informationsformat							
Bild/bilder							
Text							
Ljud							