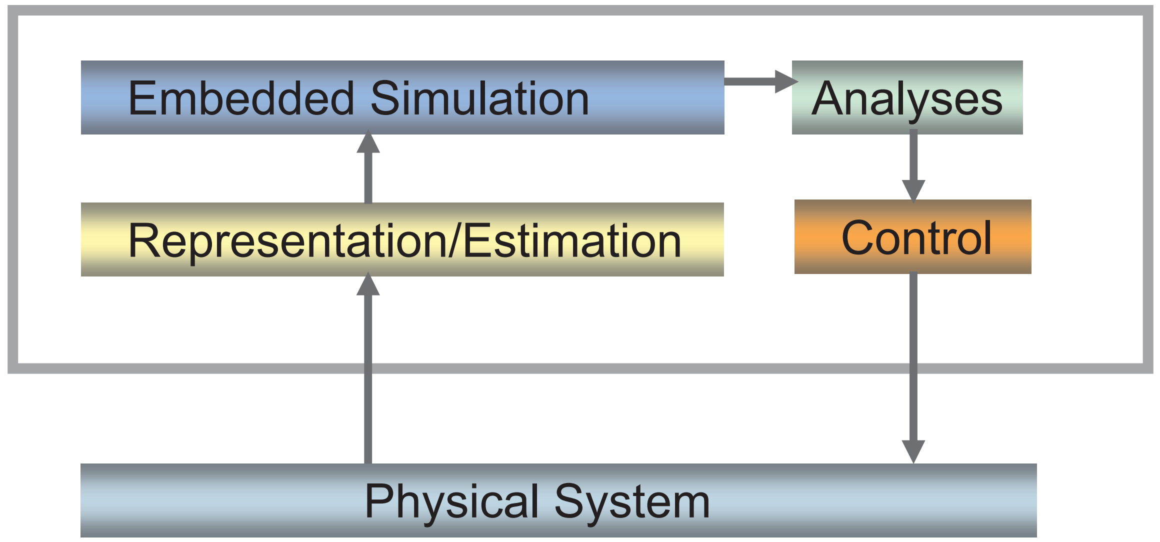
FOI
Defence Research Agency
Systems Technology
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se

Embedded Simulation Systems for Network Based Defense

ROBERT SUZIC



FOI-R-- 1828 --SE User report
ISSN 1650-1942 February 2006

Systems Technology

Robert Suzic

Embedded Simulation Systems for Network Based Defense

FOI-R--1828--SE
ISSN 1650-1942
Systems Technology
User report
February 2006

<table>
<tr><td rowspan="5">Issuing organization
FOI – Swedish Defence Research Agency
Systems Technology
SE-164 90 Stockholm</td><td>Report number, ISRN
FOI-R--1828--SE</td><td>Report type
User report</td></tr>
<tr><td colspan="2">Research area code
2. Operational Research, Modelling and Simulation</td></tr>
<tr><td>Month year
February 2006</td><td>Project no.
E60710</td></tr>
<tr><td colspan="2">Sub area code
21 Modelling and Simulation</td></tr>
<tr><td colspan="2">Sub area code 2</td></tr>
<tr><td rowspan="4">Author/s (editor/s)
Robert Suzic</td><td colspan="2">Project manager
Robert Suzic</td></tr>
<tr><td colspan="2">Approved by
Monica Dahlen</td></tr>
<tr><td colspan="2">Sponsoring agency
FM</td></tr>
<tr><td colspan="2">Scientifically and technically responsible
Pontus Svenson</td></tr>
<tr><td colspan="3">Report title
Embedded Simulation Systems for Network Based Defense</td></tr>
</table>

Abstract

Simulation has been a useful tool for the defence within fields of education, training, analyses, development of tactics etc. However, to date most simulations are considered to be separate, off-line tool that do not interact on-line with the physical (defense) systems. In the future a new kind of simulations, called embedded simulations can be used. An *embedded simulation* uses an on-line simulation as a means to support controll or optimize a physical system. New technological advances such as enhanced computer power and common operation picture represented in computer understandable manner enable the use of embedded simulations.

However, to use embedded simulations in a system such as Network Based Defense (NBD) requires efficient simulation methods as well as good human-computer interfaces In most cases the embedded simulations have to run much faster than real time and have to be adapted to NBD's different decision cycles. The focus of this study was on summary of related research, a proof-of-concept implementation and identification of Network Based Defense's (NBD) requirements on embedded simulations. We explaine and exemplify the use of embedded simulations as a real-time effect s-based decision support tool in a military operation other than war.

Keywords

simulation, embedded simulation, symbiotic simulation, decision support, composability, bayesian networks

Further bibliographic information	**Language** English
ISSN 1650-1942	**Pages** 40 p.
	Price acc. to pricelist

Utgivare FOI - Totalförsvarets forskningsinstitut Systemteknik 164 90 Stockholm	**Rapportnummer, ISRN** FOI-R--1828--SE	**Klassificering** Användarrapport
	Forskningsområde 2. Operationsanalys, modellering och simulering	
	Månad, år Februari 2006	**Projektnummer** E60710
	Delområde 21 Modellering och simulering	
	Delområde 2	
Författare/redaktör Robert Suzic	**Projektledare** Robert Suzic	
	Godkänd av Monica Dahlen	
	Uppdragsgivare/kundbeteckning FM	
	Tekniskt och/eller vetenskapligt ansvarig Pontus Svenson	

Rapportens titel

Simuleringsbaserade inbyggda system för nätverksbaserad försvar

Sammanfattning

Simulering är ett användbart verktyg för försvaret inom områden som utbildning, analys, träning, taktikutveckling osv. Hittils har simuleringar varit ett separat, frånkopplat, verktyg som inte interagerar med verkliga system.
I framtiden kan man använda inbyggda (inbäddade, symbiotiska) system som använder simulering i syfte att *direktkopplat styra* och *optimera* det fysiska systemet. Nya tekniska möjligheter som gemensam lägesbild möjliggör användningen av inbyggda simuleringar som t ex kan prediktera systemets tillstånd och effekter för att kunna optimera systemet. Att använda inbyggda simuleringar i system som nätverksbaserade försvaret ställer dock höga krav på prestanda och användargränssnitt. Nätverksbaserad försvar ställer tids - och presentationskrav på inbyggda simuleringar. I de flesta sammanhang måste inbyggda simuleringar gå betydligt snabbare än realtid och de behöver anpassas till nätverksbaserade försvarets beslutscykler.

Den här rapporten är ett resultat av forskningen om inbäddade simuleringar där man sammanställer, undersöker och utvecklar metodik för hur inbyggda simuleringar kan optimera systemets beteende och användas för nätverksbaserat försvar. Rapporten visar och exemplifierar ett koncept med tillhörande proof-of-concept modell på hur inbäddade simuleringar kan användas för effektbaserat beslutsfattande i realtid för fredsbevarande operationer.

Nyckelord

simulering, inbäddad simulering, symbiotisk simulering, beslutstöd, sammansättningsbarhet, bayesianska nätverk

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 40 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Table of Contents

Executive summary

Simulation is a useful tool for defense systems such as education, analysis, decision support, and development of tactics and planning. However, to date most simulations are used as separate, off-line tools that do not interact on-line with the physical (defense) systems. A grand challenge, identified by the simulation community, is to build embedded simulations for control and optimization of physical systems. New technological advantages and a common operational picture, with its vast amount of information, represented in computer understandable manner enable the use of embedded simulations. Network Based Defense (NBD) with its different decision cycles put requirements on embedded simulations to work efficiently and timely. Also, in most cases embedded simulations have to run faster than real time.

An *embedded simulation* uses an on-line simulation as a means to support control or optimize a physical system. In our case, with physical system we mean a part of the NBD system. The domain of embedded simulations (ES) involves several different methodologies such as data fusion, knowledge representation, composability of simulation models, human-computer-interface, reusable and meaningful what-if models, and efficient control of ES.

There are several potential benefits by using ES in future NBD. ES can enable on-line decision support. ES can serve the commander in tactical situations where different "what if" types of course-of-action analysis can be performed faster than real time. ES simulations can also be used for automatic control of systems. A potential use can be in controlling unmanned air vehicles (UAVs). Moreover, ES simulations can be used to simulate parts of the common operational picture.

However, in common operation picture example, the user should have knowledge of what are real and what are simulated data. In what-if types of analyzes a robust way of thinking is required to compensate for (possible) model errors. The increased robustness in ES results can be achieved by using stochastic models and uncertain data. Generally, it is recommended that in cases where uncertainty arises, embedded stochastic simulations should be used.

In contrast to traditional (off-line) simulations, to be efficient ES demand reusable and composable models valid within a particular simulation problem domain. Also a "deeper" (machine understandable) interoperability between C4I systems and ES applications is required. One way to support integration of ES with the physical (other) systems is by the use of architectures.

In future research we recommend development of new methods for connecting simulation and physical systems. In the NBD case, this could be made by connecting ES with the common operational picture.

Also, in field by using tools such as Tablet-PC a user could trigger an ES and obtain timely decission support. However, in some cases where simulation is a part of an automatic (sub-) system the end user may not need to know that an ES is running. The verification of the simulation results on-line puts both higher demands on the user and on the (new) methods for on-line verification and validation.

Embedded Simulation Systems for Network Based Defense was a one-year project with the overall objective of investigating and developing methods for how simulations, as part of physical systems, could analyze, and optimize the behavior of the target systems in real-time.

The focus of this study was on summary of related research, a proof-of-concept implementation and identification of Network Based Defense's (NBD) requirements on embedded simulations. We explain and exemplify the use of embedded simulations as a real-time effects-based decision support tool in a military operation other than war.

1 Background

Simulation is a useful tool for defense systems such as education, analysis, decision support, and development of tactics and planning. However, to date most simulations are used as separate, off-line tools that do not interact on-line with the physical (defense) systems. A grand challenge, identified by the simulation community, is to build embedded simulations for control and optimization of physical systems. New technological advantages and a common operational picture, with its vast amount of information, represented in computer understandable manner enable the use of embedded simulations. Network Based Defense (NBD) with its different decision cycles put requirements on embedded simulations to work efficiently and timely. Also, in most cases embedded simulations have to run faster than real time.

Embedded Simulation Systems for Network Based Defense was a one-year project with the overall objective of investigating and developing methods for how simulations, as part of physical systems, could analyze and optimize the behavior of the target systems in real-time.

The focus of this study has been to summarize related research, build a proof-of-concept implementation and identify what requirements the Network Based Defense (NBD) concept imposes on embedded simulations. Providing optimization and control of complex systems in real-time requires some kind of embedded intelligence. In our proof-of-concept model we use Bayesian Networks for embedded simulations (ES).

Chapter 2 contains definitions of ES and a generic architecture for it. In Chapter 3, we summarize related national and international research and identify some common themes; work presented in Chapter 2, 3 was made in cooperation with Simulation Based Decision Support project members. Effects-based operations is a new way of thinking on how military operations should be carried out; a description of this concept and how it influences ES is given in Chapter 4. In Chapter 5 we present a concept model of how ES can be used in riot situations; this research is a product of joint research within this project, the information fusion project at FOI, and KTH.

2 Embedded simulations: An overview

2.1 Embedded simulations as a concept

To date, most simulations are used as separate, off-line tools that do not interact on-line with the physical (defense) systems. However, computational power has increased significantly and is much cheaper today than 15 years ago. ESs, as parts of the physical systems, are aiming to provide real-time decision support by running faster than real-time *what-if* simulations and in some cases even perform optimization of the target system's behavior, see Figure 1.

An *embedded simulation* uses an on-line simulation as a means to support control of or optimize a physical system. In our case, with physical system we mean a part of the NBD system. The simulation and the physical system interact in real-time and both benefit from the interaction. The physical system benefits from the optimized performance that is obtained from the analysis of simulation experiments. The simulation system benefits from the continuous supply of the latest data and the automatic validation of its simulation outputs. Several different terms have been used to describe the same idea such as symbiotic (i.e. mutual-benefit) simulation, embedded simulation and simulation based embedded systems.

By using ES continuous reevaluations of the situation can be performed and new (context relevant) knowledge can be used to support decision making. ES can generate pieces of knowledge given different decision alternatives by using online what-if simulations. Such knowledge can be further combined with other knowledge (information) fragments and a (information) model that represents the evolving situation would be obtained. The information model explains relations and mechanisms between decisions and their (predicted) effects. Such a model would be a useful tool for providing real-time effects-based operation analyses on tactical level in order to support decision-making on-line.

2.2 Methodological Requirements

In contrast to offline simulations, online simulations require machine interpretable and more flexible representations of the physical systems. In most off-line simulations, the user configures and starts simulations via a graphical interface. An example of machine to machine embedded simulation is a simulation that is triggered by some machine system such as an on-line C4I system that requires some data to be simulated. Such a process which triggers and/or configures an ES is called an instantiation process, see Figure 1. The results of the simulation are delivered to the C4I system that makes machine or machine-human interpretation.

The ES has to have the ability to adaptively represent a physical system that changes with time. For example, new units that enter the scene have to be interpreted by the ES in a consistent manner.

The domain of ES involves several different methodologies such as data fusion, knowledge representation, composability of simulation models, human-computer-interface, reusable and meaningful what-if models, and efficient control of ES, see Figure 1.

Composability is the capability to select and assemble reusable simulation components in various combinations into simulation systems that meet user requirements [Petty, 2004]. There are two types of composability, syntactic and semantic. Syntactic composability is concerned with the compatibility of implementation details, such as parameter passing mechanisms, external data accesses, and timing mechanisms. It is the question of whether a set of components can be combined. Semantic composability, on the other hand, is concerned with whether the models that make up the composed simulation system can be composed and remain valid. Both types of composability are important for ES.

To be useful, some ES also need different types of artificial intelligence complementing (end user) human intelligence. Examples of AI methods where the results of what-if simulation can be used are different optimization methods (e.g., stochastic dynamic programming [Johansson, Mårtenson, Suzić, and Svenson, 2005], simulated annealing [Kirkpatrick, Gelatt, and Vecchi]), Bayesian Networks [Jensen 1996], Game Theory [Osborne, Rubinstein, 2001] and data processing/mining.

Contextualized information and models are required to enable embedded simulations and efficient decision support for rapid control of complex systems in real-time. Semantic Web technologies provide means of representing knowledge in a structured and formalized manner. Furthermore, these technologies are capable of knowledge reasoning which can be employed for contextualization and instantiation of embedded simulations, see Figure 1. Semantic Web Technologies such as Resource Description Framework (RDF) and its extension the Web Ontology Language (OWL) can be used for contextualization.

Data fusion is the process of combining sensor data and a priori knowledge (e.g., environment, behavior recognition models) in order to obtain the best possible estimates of the states of agents/phenomena or object of interest. In the context of ES the task of data fusion is to estimate a physical system's states of interest in real-time. Those estimates are then used as a input into embedded simulations.

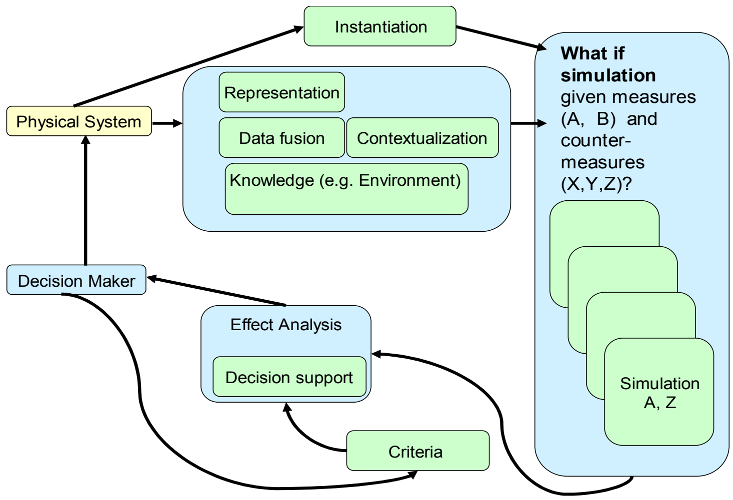


Figure 1. Embedded simulation concept

3 Summary of relevant publications

The papers described in this section can be classified as having either civil or military applications. The criteria used when choosing articles were their potential benefits for ES in NBD. In the concluding part of the summary, we bring up some generic features. This chapter was written in cooperation with Simulation Based Decision Support project members Farzad Kamrani and Marianela Garcia Lozano.

3.1 Related military research about ES

The publications summarized here originate from work in different groups in Sweden and the United States. In Sweden, research work with embedded simulations has been performed at Royal Institute of Technology (KTH) lead by professor Stefan Arnborg, at FOI within the information fusion projects and at Saab Systems. The publications from USA, summarized here, present US Navy’ s test and implementation work with ES.

3.1.1 US Navy Research for ES

The US Navy Embedded Simulation Infrastructure (ESI) Program [Daly, and Layman, 2002], [Layman, and Weatherly, 2003], [Layman, Weatherly, and Robinson, 2005] has been established with the goal of combining powerful embedded computer simulations with existing C4I functionality and to produce enhanced C4I tactical applications. This work also involves architecture work and developing composable M&S components.

According to [Daly, and Layman, 2002], one needs to look beyond the paradigm of the electronic map that C4I offers today and think about the integration of useful information in an operational picture representing something other than troop geo-locations over time. “Simulation based mission applications such as a course of action analysis application, or visual representations of complicated operational assessment models, can significantly improve the quality and timeliness of tactical decisions for military operations. Simulation based applications have the ability to process C4I data beyond human cognition, and display it in an intuitive and meaningful way for key tactical decision makers" [Daly, and Layman, 2002]. A Common Operational Picture (COP) can utilize embedded models or simulations to present synthetic views of information other than "tracks".

The task of the C4I system is to provide a COP to the tactical decision maker. Here, the COP is proposed to be divided into two parts. The first part is COP that is based on current processed and aggregated data. The second part, which is based on simulated data, is called simulated common operation picture (SIM-COP). The SIM-COP can also insert simulated data in place of and complement real-world data.

Generally, the enhanced information processing in C4I enables the development of processes for the use of embedded simulations within tactical applications. E.g., simulated information about electronic sensor performances and atmospheric conditions has been available in stand-alone computer systems. However, their tactical use is limited due to lack of integration with the C4I system. The huge amount of information available in C4I systems makes them ideal hosts for advanced tactical mission applications such as the use of embedded simulations. An important example is the Predictive Battlespace Awareness [Piccerillo, and Brumbaugh, 2004] that rely on accurate, timely access to C4I information and data that ES are assumed to be a part of.

Benefit for US navy

Simulations used in tactical decision aids, operation planning, and course of action (COA) analysis are often based on executing simulated scenarios that provide insight to the tactical decision maker [Daly, and Layman, 2002] .

There are some examples of the use of ES such as assessing vulnerability of own communications networks, and assessing probability of detection of own forces during various conditions, assessing best protection from hostile force activities such as deploying weapons of mass destruction (WMD).

According to [Daly, and Layman, 2002], there are several benefits of using ES such as:

1) Projecting operations into the future for planning purposes or analyzing potential courses of action.

2) Adaptive Operational Domains that influence the tactical situation. (e.g.: radar coverage, acoustic conditions, WMD effects, Information Operations etc.) These may require the simulation to operate faster or slower than real-time.

3) A combination of both such as simulating a WMD event and time projections in the future to plan evasive action.

4) Effects-based analyses (The ES capability will add much needed defense analysis input to operational planning, as well as allow consideration of effects currently too complex or time consuming to analyze by manual means.)

Decision Support Requirements

In order for decision support to serve the commander in tactical situations, a close coupling with the C4I data is required. Tactical decision support can have many different flavors, from "what if" types of course-of-action analysis, to a commander's inference from using good situational awareness. However, it is important to ensure that simulation results are (on-line) valid. "Simulation based applications have traditionally been used in decision support in a "what if" war-gaming type of analysis. While this can be operationally useful, it is also subject to debate on validity of results" [Daly and Layman, 2002] .

Other challenges that ES impose

Interoperability (links are required between embedded simulations and C4I Applications)

Representation challenge (representation of simulated COP and SIM-COP at current time step; this is also an human-system interation issue)

Time (in some cases faster in some cases slower than real time)

Architecture and composability issue

According to [Layman and Weatherly, 2003], most commonly M&S are poorly integrated into C4I systems and therefore we need a useful architecture. The paper focuses on this common architecture and the issue of simulation model composability.

There are few (generic) architectures for ES and C4I together. Models are rarely reused and often lack composability. According to [Layman and Weatherly, 2003], one of the reasons for the modest success of integrating models and simulations in C4I system applications is due to the lack of Modeling and Simulation (M&S) services within the common operational environment (COE). There is a need for M&S software components that are available to generate training and planning scenarios, provide simulations links to internal C4I data bases and functions, manage the simulated data within the C4I system, manage variable time bases, display simulated data, integrate simulations with other applications and perform communication tasks. A proper selection and development of M&S components and an *architecture* that manages complexity and reduces development workload is a way towards efficient integration of M&S into C4I.

There are three critical elements that are necessary to compose simulation-based C4I applications out of common components [Layman, and Weatherly, 2003]:

- Selection and design of sharable software components that can be used within a domain
- Flexible architecture structure that promotes reuse within a domain or even between domains
- Development environment that promotes the combined use of the common operation environment and the M&S components.

An architecture that is aimed to satisfy the requirements above is proposed by [Layman, Weatherly, 2003]. In the scenario generation part there are Common Operation Picture (COP) Capture, Mission Editor, Time Line Editor, Virtual Track Manager (VTR) and Scenario Preview to develop a plan for analysis or training a scenario.

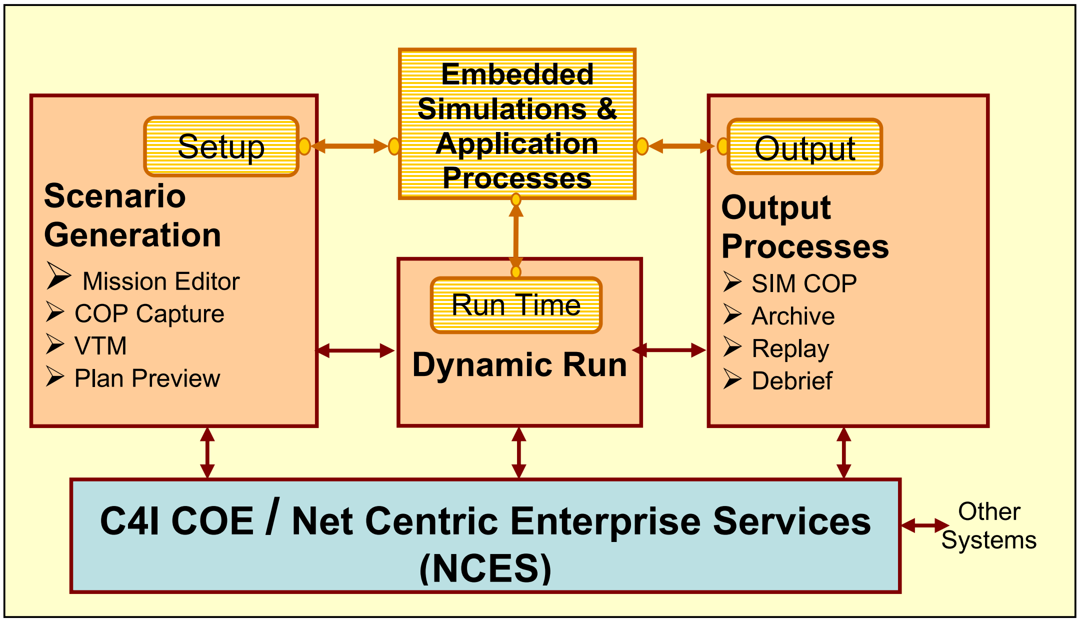


Figure 2. Host Application Architecture [Layman, and Weatherly, 2003].

Many large complex applications can best be developed and maintained using small, limited scope embedded simulations rather than a single large simulation. The Dynamic Run and the simulated COP services interact with the embedded simulations during runtime and the Archive/Replay/Debrief segment to capture and present the application's resulting analysis, see Figure 2. The Host Applications organize M&S services into shell programs to support a specific application domain. Moreover, the services can be developed to support a wide range of simulation based Mission Applications.

A Host Application has the following characteristics [Layman, and Weatherly, 2003]:

- Contains a set of M&S services relevant to a particular C4I Mission.
- Hosts one or more simulations or processes
- Readily modifiable to host additional simulations of the same category. (e.g., Sensor Fusion, Strike Planning)
- Provides interface standards for embedded simulations.

Mission Application Software Development Kit (MASDK)

In the paper [Layman, Weatherly, and Robinson, 2005], architecture and features of the MASDK is described; a software developed by Naval Research Laboratory within the C4I Embedded Simulation Infrastructure (ESI) program.

The MASDK is an architecture framework, a set of sharable C4I software components and a composability tool. It is written in Java and integrated with the Common Operating Environment (COE). The aim of MASDK is to reduce the complexity of the design and the developers' workload by composing systems from a collection of reusable elements. One can choose to use Mission Editor to develop a scenario rapidly by creating simulated forces and positioning them on a map or use COP

Capture to extract the real-time C4I Common Operational Picture (COP) as a starting point for a scenario. A scenario can be saved as an XML file and be run or modified later.

3.1.2 Research of Decision support group

The decision support group (URL: http://www.nada.kth.se/theory/dsg/) is an informal network of researchers in Sweden from KTH/Nada, Saab Systems and FOI. The research at the decision support group is about providing decision support tools for Command and Control applications, focusing on decision theory, data fusion and knowledge representation. Here, we summarize two the group's publications that are relevant for ES.

Simulation-Based Decision Support for C2

In [Huang *et. al.*, 2003] a brief motivation and introduction to simulation-based decision support for command and control in joint operations is performed. In joint operations where different units are involved the commander can not always predict the outcome of different tentative plans and compare these alternatives since the situation is complex and plans have multi-attributes. The paper suggests that ES should be part of the future C2 systems which gives the commander the opportunity to simulate tentative decisions and evaluate their consequences during the planning process. Embedded simulation-based decision support is composed of multi-agent simulation used to predict the consequences of a tentative plan and multi-attribute evaluation to evaluate and compare the multi-attribute consequences.

The ES support is a network resource that would be connected to other functional units of the command and control system. The aim of this design is that it can also provide decision-making services for training. In the multi-agent simulation each resource is substituted by an agent which is an autonomous system that has the capability of performing tasks. A plan defines a set of agents each agent may be involved in different tasks but is associated with only one role. Tasks are composed of several actions.

In [Brynielsson*,* 2002] and [Huang *et. al.*, 2003] authors propose functions such as environment and an information manager. Multi-attribute evaluation is modeled as a consequence matrix C , a utility matrix A and utility functions U which maps C to A. The elements of the matrix C are consequences of the tentative plan s_i on the attribute a_j. For each attribute these values can be summarized using a step function with limited number of utility levels. This will give the commander and staffs an understandable situation assessment.

Since other agents may take different actions to meet the commanders' plan the model has an inherent uncertainty that can not be eliminated. The suggested solution to this uncertainty is to calculate one consequence matrix and respective utility matrix for every possible opponent plan. Summarizing the probability weighted by these utility matrices yields a total utility matrix which should be used for evaluating the tentative plans.

Particle filter-based information acquisition

In [Johansson, and Suzić, 2004], a stochastic simulation method called particle filter [Doucet, Freitas, and Gordon, 2001], has been used to represent uncertainty in state to *predict* the future state.

Uncertainty arises due to lack of observations. Therefore an agent is represented as a set of possible or alternative states called particles. Those state uncertainty particles are depicted in left part of Figure 3 (red particles). Particles in the particle filter can be seen as hypotheses representing alternative states.

In [Johansson, and Suzić, 2004] each particle represents a state of possible agent's (enemy platoon's) position and velocity.

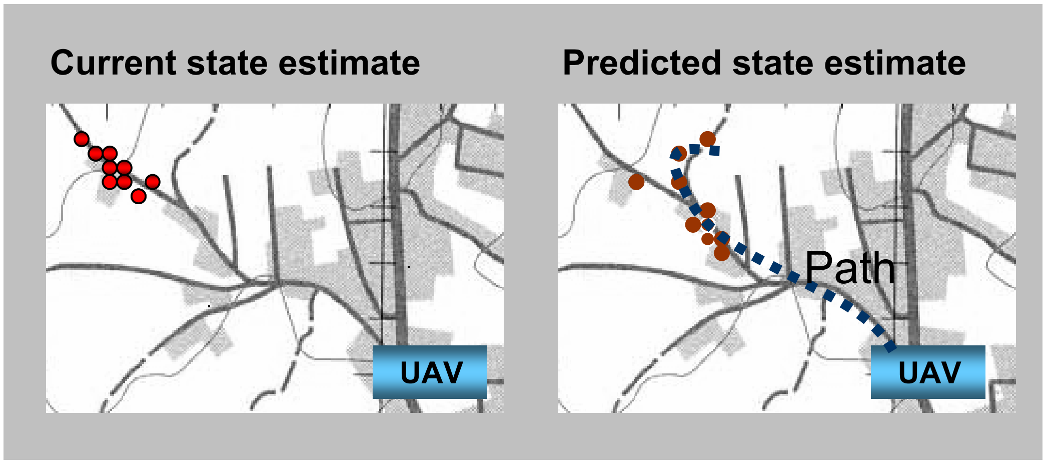


Figure 3. State uncertainty particles (red) and prediction particles (brown)

Generally, to reduce uncertainty about agent's (enemy's) state, position and velocity in this case, sensors are used. From ES point of view we found valuable in [Johansson, and Suzić, 2004] a *pro-active approach* where on-line simulation has been used to predict situation repetitively and on-line. The prediction time has been set in relation to how long time is required to reach an area of interest. This is done by using a transition model that takes state uncertainty particles as input and simulates an output prediction state, see right part of Figure 3. Consequently, by running those simulations the route planning of sensors (e.g. UAVs) can be done with information based on predicted state when the sensor would enter the area. This process has been run repetitively as soon as new sensor task was assigned. The use of particles for decision making has also been investigated by [Brynielsson *et. al.,* 2005].

3.1.3 Other related research at FOI

<u>A Simulation-Based Embedded System Prototype</u>

The main goal of the master thesis [Gülich, 2004], supervised by M&S's research director Rassul Ayany, has been to investigate how simulation-based embedded system can be constructed and used. For this purpose a prototype for optimizing the performance of a computer network is developed. Running simulation in parallel with the network, the system predicts the effect of different strategies on the network and chooses a strategy that minimizes the occurrence of bottleneck links in the network. Instead of a real computer network the system is connected to a model of a computer network implemented in Network Simulator 2 (NS-2).

Three different forwarding policies are compared; using routing table lookup, choosing one of the outgoing links randomly and selecting the outgoing link with the shortest queue. The parallel simulation interacts with the "physical system" periodically and the best chosen strategy is applied to all "routers". Both simulation and the "physical system" run parallel for whole periods of times and interact at the end of these periods. The simulation is ahead of the "physical system" and predicts the state of the system at the beginning of the next period. Based on this state three simulations with different routing policies are run and the best routing policy for the coming period will be chosen. When interacting, if the predicted state is validated against the real state of the network, the best policy is given to the "physical system" to be applied for the next period. Otherwise the "physical system" chooses a default policy and the simulation updates its picture of the state of the network.

Evaluating sensor allocations using equivalence classes of multi-target paths

In [Mårtenson, and Svenson, 2005], an algorithm is introduced for evaluating sensor allocations using simulation of equivalence classes of possible futures. The method that the authors propose is meant to be used for pre-planning allocation of sensors, e.g., for choosing between several alternative flight-paths for UAV's or for deciding where ground sensor networks may be deployed. Given a situation picture, possible future paths for objects of interest are generated by simulation. Then these possible paths are evaluated against each other and are considered equivalent if the sensors observations are the same. Different sensor schemes are then ranked according to how many objects they discover. The work of [Mårtenson, and Svenson, 2005] is based on previous work presented in [Ahlberg, *et. al.*, 2004].

Knowledge representation, modelling of doctrines and information fusion

[Suzić, 2003] presents a conceptual model of agent based simulations as one part of the information fusion process. Information fusion is an on-line (C4I) process, hence this agent based simulation is also one type of ES.

The process named above is assumed to consist of agents which store and simulate different behaviors on different aggregation levels, see Figure 4. Each agent represents a military unit which can consist of other units. It is important to be able to build such a hierarchy in a consistent and reusable way. Agent one is in this case a platoon but it can be a company or any other type of agent, see Figure 4.

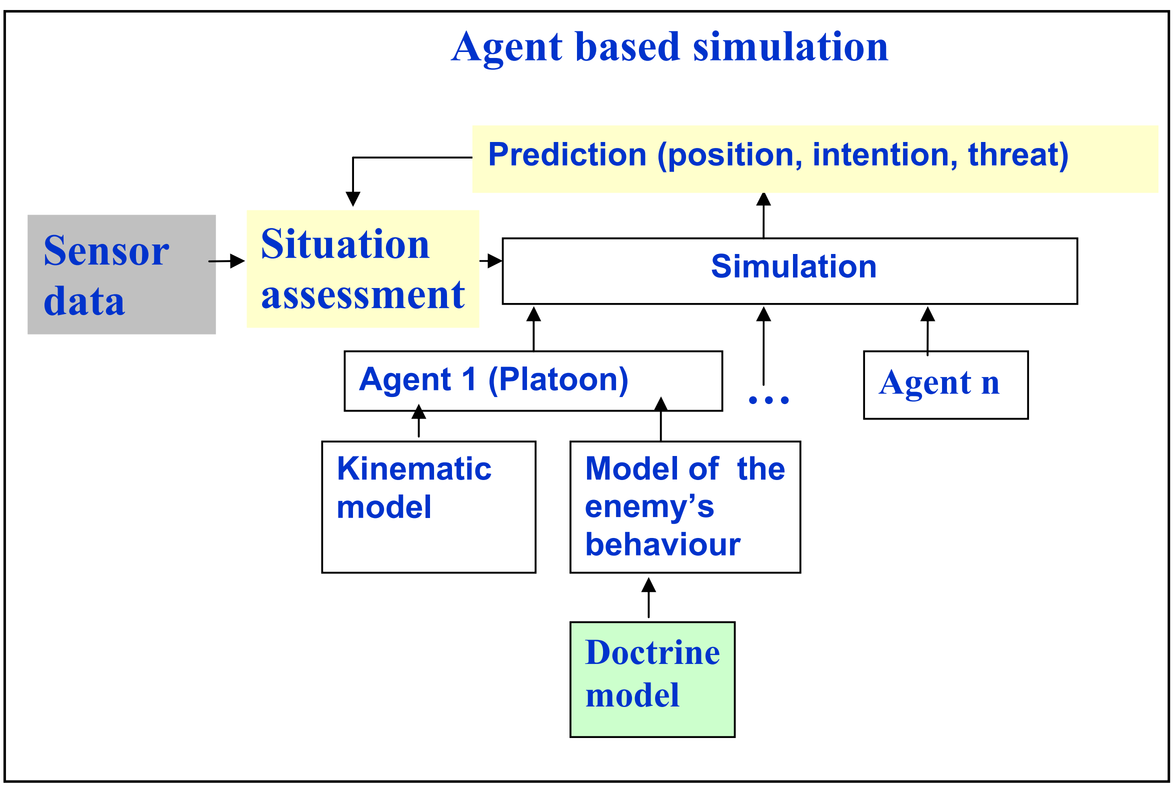


Figure 4. One conceivable way of using doctrine models and integrating them in an agent based simulation that is a part of an on-line system (information fusion)

The goal of this process is to:

1) Collect all relevant sensor data and information
2) Combine this information with a priori knowledge, e. g. terrain, doctrines, performance of vehicles etc.
3) Represent improved estimates of world state of interest (situation assessment)
4) Predict the most likely course of the enemies´ actions by the *agent based simulation* and connect them to the most probable and most destructive outcomes (impact analysis). Connect those results to situation analyses when new sensor reports arrive.

3.2 Related civilian research about ES

The related civil research contains three mainly publications.

Dagstuhl Seminar Grand Challenges in M&S

The aim of the Grand Challenges for Modeling and Simulation Seminar at Dagstuhl in 2002, [Fujimoto, Lunceford, Page, and Uhrmacher, 2002], was to condense ideas and proposals into a set of Grand Challenges that would help propelling the research progress in modeling and simulation. At this Seminar the participants were divided into several working groups where the group for Parallel / Distributed Simulation came up with the idea of Symbiotic Simulations (later called also ES) [Fujimoto, Lunceford, Page, and Uhrmacher, 2002]. This seminar is interesting since it is the first time (to the best of our knowledge) the term symbiotic simulations and definition of it is mentioned.

The group's definition of a symbiotic simulation is as follows, "A symbiotic simulation is a simulation that interacts with Physical System in a Mutually Beneficial Way. The Physical Systems benefits by optimized performance and the simulation benefits by having continuous data and automatic validation of simulation outputs. This implies real-time interaction and a control or decision support function." This process description is depited in Figure 5.

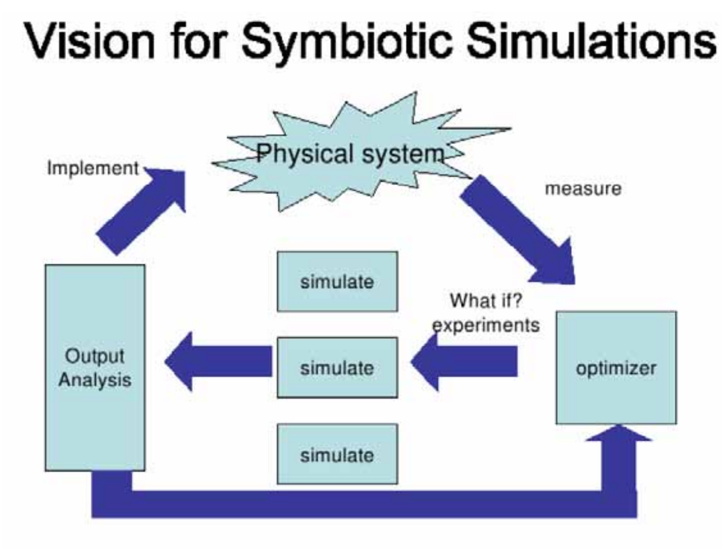


Figure 5. A proposed high level architecture view of how simulations and physical system interact [Fujimoto, Lunceford, Page, and Uhrmacher, 2002].

Simulation Software Component Architecture for Simulation-Based Enterprise Applications

Tools such as CORBA and DCOM together with internet, distributed data bases and simulations, enhanced processing power enable the use of embedded simulations for on-line decision making/support. However, one of the obstacles [Harrell and Hicks, 1998] is the low rate of simulation model reusability. This means that, from a detailed modeling perspective, every situation is a unique situation, and cannot be modeled generically with a reusable simulation. Most simulations include discrete-event simulation that has been viewed as a standalone, project based technology. In previous approaches simulation models were built to support an analysis project, to predict the performance of complex systems, and to select the best alternative from a few, well defined alternatives. Single use models where poorly reused in other future applications.

To be efficient, ES demand models that are valid within a particular simulation problem domain. In traditional simulation model building, software is designed to handle an infinite use case. That makes simulation software development environments extremely difficult to make robust and error free. In traditional simulation development environments, a graphical user interface is the primary means of interacting with the application. On the other hand embedded simulations do not need to involve direct intervention by humans. Instead, the simulation can extract data from existing sources and apply embedded simulation to the problem as defined by the simulation problem domain, and create simulation results that can be used by humans or by some machine processes. The user who takes an action that triggers a requirement for simulation may not even know they are asking for a simulation to be run [Harrell and Hicks, 1998].

An Agent-based Approach for Managing Symbiotic Simulation of Semiconductor Assembly and Test Operation

In the paper [Low, Lye, Lendermann, and Turner, 2005], the authors describe their work in developing a prototype proof-of-concept symbiotic simulation system. The system uses software agents for monitoring, optimization and control of semiconductor assembly. According to this paper, semiconductor manufacturing experiences an increasing complexity which results in rapid changing business environment and shortening of the life cycle of products. A typical semiconductor backend and test facility has a large number of process flows with short cycle-times, roughly one week.

Simulation has been an important decision making tool in evaluating scenarios in semiconductor manufacturing. However, since the physical system is changing very often the modeling and simulation process can not cope with these rapid changes. Symbiotic (embedded) simulation addresses this problem and will enable the system to respond promptly to abrupt changes of the physical system.

In contrast to military systems the physical system here is a factory consisting of two assembly and test conveyors. Each of these conveyors is a composition of several machines working serially. Whenever the in-house capacity of machines can not handle the workload, some portion of the workload should be out-sourced to an external vendor. The decision is evaluated considering the penalty cost for order delays and cost of out-sourcing.

The paper [Low, Lye, Lendermann, and Turner, 2005], emphasizes the use of agent-based technology to realize the symbiotic integration between a simulation-based decision support optimization module and a model of a semiconductor backend system.

Agent is a software program which is autonomous, proactive, responsive and adaptive. Using multi-agent where each agent makes decision based on local knowledge and cooperates with other agents for global optimal performance of the system gives the system the characteristics of distribution of control, resources and experience among different agents.

According to [Low, Wei Lye, Peter Lendermann, Turner, 2005], this approach makes the simulation a tool for not only strategic and tactical but also operational decision making in the semiconductor manufacturing shop floor. However, it should be underlined that this symbiotic simulation model is not integrated with a real semiconductor back-end factory but with a software model of such manufacture emulated in the commercial software package Witness.

Different agents in the system are implemented using JADE agent toolkit. System Management Agent (SMA), see Figure 6, is the central unit of the system which interacts with operators (human users), the physical system (via Monitoring Agent and Control Agent), and the simulations (via Optimization Agent).

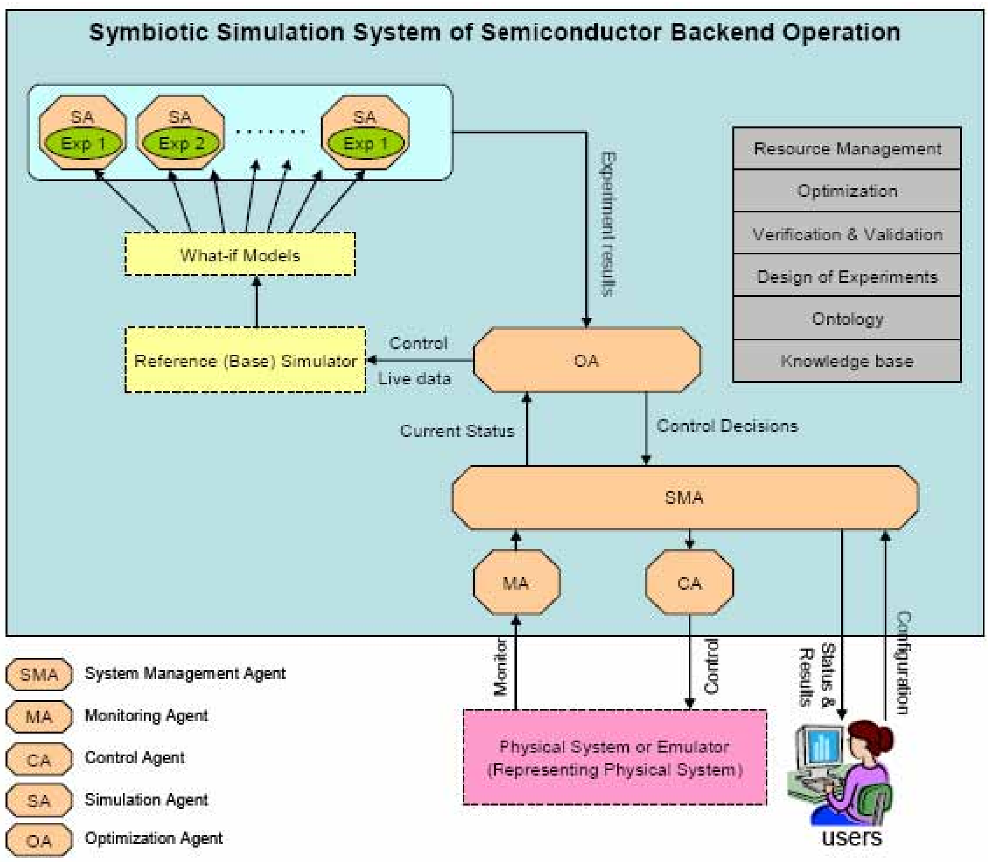


Figure 6. Symbiotic (embedded) simulation architecture presented in [Low, Wei Lye, Peter Lendermann, Turner, 2005]

Whenever the utilization of the machines exceeds 80%, the Optimization Agent carries out simulation-based optimization by comparing different "what-if" scenarios. The process begins with reading the current state of the system from System Management Agent and creating a Base Model of the system. This Base Model is run in parallel with the physical system and used by different Simulation Agents to carry-out "what-if" analysis. The Optimization Agent summarizes these results and if necessary instructs the Control Agent to make modifications in the physical system.

4 Embedded Simulations as a Tool for Efficient Decision Making

The concept of effect based operations (EBO) is a new way of thinking where desired effects (higher order goals at strategic level) are put in focus when planning, executing, and assessing military operations. EBO forces decision makers to look at outcomes and their explanations more than actions taken [Smith, 2002], [McCrabb, 2002], [Ho, 2004]. According to U.S. Joint Forces Command, EBO is a process for obtaining a desired strategic outcome or “effect” on the enemy, through the synergistic, multiplicative, and cumulative application of the full range of military and non-military capabilities at the tactical, operational, and strategic levels. EBO considers effects beyond tactical goals and may facilitate the consideration of conflicting interests.

EBO comprises four fundamental aspects [Eklöf, Yi, and Suzić, 2005]:

- *Planning (EBP)*
- *Execution (EBE)*
- *Assessment (EBA)*
- *Knowledge Base Development (KBD)*

Also, the careful analysis of the extent to which effects are expected to occur plays an essential part in the EBO concept. Accordingly, there is an emphasis on the *mechanisms* describing the causal relationships between actions and achieved effects, see Figure 7. Several previous authors have discussed methods to perform such an analysis for EBO planning [Wagenhals *et. al.,* 2003]. The focus of EBO is on causal explanations (models of mechanisms) that represent relations between actions and cause (effect). ES is methodology that is aimed to facilitate a *real time* EBO process by several means. One of them is presented in the next chapter where ES are used to support a tactical commander’s decision making by modeling the context dependent causes between different own actions, hostile agents actions and (local) effects. The second benefit is in online what-if analyses to make automatic (routine) decisions in a system such as NBD. The third benefit is that ES simulations can compensate for some of missing data in data bases (simulation of data); such simulations may run even without knowledge of the user and transform data from one database to other given relevant simulation models and data that describe current context. Also the use of embedded simulation to support decision-making has been discussed in Chapters 3 and 5.

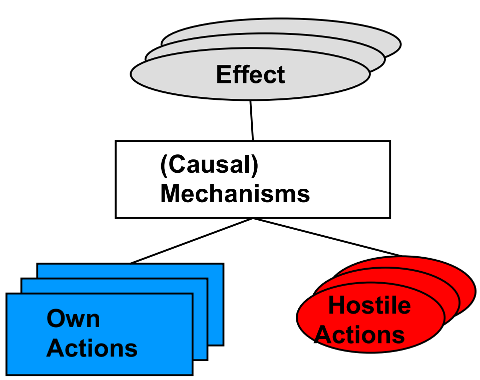


Figure 7. Relation between actions and effects via mechnisms

5.0 Effects Based Decision Support for Riot Control Employing Influence diagrams and Embedded Simulations: A case study

There are numerous potential benefits of using ES in NBD, see Chapter 3. Here, considering the recent shift in the role of Swedish armed forces we have focused on military operations other than war (MOOTW) case. The MOOTW case used here is a riot control case in a foreign country where Swedish military has (UN) mandate to keep public order. In this chapter we explain and exemplify the potential use of ES to support a MOOTW operation of keeping security.

The research presented in this chapter is a product of cooperation between SBIS project members, members of the FOI information fusion project and a PhD student Klas Wallenius from KTH/Saab Systems. The reference persons have been Stefan Arnborg (KTH) and Michael Malm (FOI).

We show here a concept of real-time EBO support for a commander involved in riot control to evaluate different strategies in real-time against conflicting goals. A first prototype of ES's proof of concept has been implemented by SBIS project member Martin Jerberyd. In this chapter we describe some of its functionality. Moreover, we describe how Bayesian Networks (BNs) are applied; BNs are combined with embedded simulation to facilitate a continuous reevaluation of the situation in order to optimize the actions performed by own units. Here, we also give an example of how the suggested approach will appear from an operational point of view.

5.1 Decision support in riot control

Riots put very important societal values at risk. We assume a generic scenario where at a summit (top level meeting) in a city, thousands of protesters from different countries are expected to show up to exercise their rights to demonstrate for or against different issues. It is also expected that some of these protesters will be trouble-makers intending to disturb the meeting. In the case of a domestic event, the police is responsible for the security at the summit, while in the case of an international peace-keeping operation, military units may be responsible. Anyhow, the overall goals for the security are to <u>protect the meeting</u> and its <u>participants</u>, and to <u>defend public and private property</u> from being damaged. At the same time, the police or the military must be very cautious neither to interfere with the integrity of the protesters and their rights to free speech, nor to unnecessarily provoke potential trouble-makers to cause further harm. In the case of riot control at a summit, effects include the rights of societal institutions to meet and to make decisions, as well as the right of the people to protest against these decisions. Here, we propose a concept for a commander involved in riot control to evaluate different strategies in real-time against these conflicting effects.

The concept of EBO, see Chapter 4, may facilitate this consideration of conflicting effects. EBO emphasizes actions performed to achieve effects beyond the most obvious tactical goals, i.e., in this case not just to prevent protesters from disturbing the meeting, but to maintain the rights of societal institutions to meet and to make decisions, as well as to maintain the rights of the people to protest against these decisions. Actions performed by units belonging to the police or the military need to be carefully synchronized to achieve these effects.

The embedded simulation supports the cumbersome work of maintaining the causal relationships represented by the BNs. By utilizing an *agent-based* approach to this simulation, the problem of designing a complex simulation model of the whole situation is reduced to the problem of designing less complex performance models of the interacting agents. Furthermore, some sort of risk management is required, as there are uncertainties in the predicted development of the situation due to

both uncertainties in the knowledge of the current situation and uncertainties in the agents' performance models. To address this issue, we apply a stochastic approach to the agent-based simulation, resulting in probability distributions of the causal relations between actions and effects, which in turn are used to maintain the BNs. Included in the concept, the different services and types of information used by the commander are presented in Figure 8, in which we by an arc relation mean "uses". Hence, the tactical commander is considering alternative tactical orders to the subordinated units in order to comply with the strategic priorities from the strategic commander. To facilitate this assessment, the commander gets information about the situation by means of the tactical situation picture, depicting the locations and activities of own units as well as of demonstrators. The tactical picture is based on information from field reports, being processed by a data fusion service as described in for example [Bergman and Wallenius, 2001]. However, the main emphasis in the proposed concept is on how the tactical commander can make use of services dedicated to generate and maintain BNs and get a "feeling" of how a situation *might* develop. By this model, the uncertain effects of different actions on the tactical level could be analyzed by the commander, in order to optimize decisions on how to utilize subordinated units. The resulting model could also be the basis for computer based optimization. Hence, this chapter we will describe how, conceptually, the different embedded simulation techniques need to be applied in order to implement this functionality. In Section 5.4.3., we will also give an example on how the suggested approach will appear from an operational point of view. The example given deals with effects-based riot control, but the approach could be useful for other types of security activities as well, such as other police work, disaster relief, and other types of military operations.

Figure 8. Services and types of information used by the commander through the proposed concept. The arrows depict agents using services and services using other services [Suzić, and Wallenius, 2005].

5.2 Riot Control Simulator (RCS): an embedded simulation prototype

Within the SBIS project we have developed a proof-of-concept model that is aimed to exemplify and give new research findings of embedded simulations. Here, our concept [Suzić, and Wallenius, 2005] has been to a certain extent exemplified. An SBIS's project member Martin Jerberyd has implemented the proof-of-concept simulation model that we called *Riot Control Simulator (RCS).* It consists of interface, stochastic agent models, embedded simulations and the results (predicted effects). In following subsections we describe some of the features of the RCS.

5.2.1 RCSs interface

The RCS has interface that has several functions concerning setup of scenario and simulating own and protesters decisions. Here follows the list of those functions:

<u>Environmental specification (Map)</u> It enables the user to build up environment or to import a certain type of environment. In future application an environment should be able to be directly imported from some C4I system containing a COP.

<u>Rioters function</u>. It involves placing protesters in number and setting an uncertain estimate of their internal state that is anger in this case. This functionality should be integrated in future with COP.

<u>Insertion of measures at different places (Peace keeping force)</u>. Here, the user can insert peace keeping force's barrier with different strengths at various places; the alternative "barrier" can be interpreted as well as military force with shields.

<u>Attraction points (A-point function)</u>. The A-point function consists of several alternative places where the simulated protesters (agents) can be *attracted* to. The attraction point can be places of meeting/meetings, a car, or building/buildings. The degree/strength of attraction can be also specified in the interface.

Repetitive simulation in simulation

Given a specific scenario and combination of own decisions and assumed protesters decision an embedded simulation can be applied. Since each simulation is stochastic-agent based simulation, different results (effects) may occur. Under tab SimTools/ScenarioRepeat, there are several parameters that can be set by the user, see Figure 9.

In the field <u>number of simulations</u> the user should be able to specify how many times an embedded simulation will run.

By specifying the number in range zero to one the user can enter its own belief of the medium value for the statistical distribution of <u>people anger</u>.

The <u>stop time</u> is the time on how long the prediction of situation evolution will be simulated.

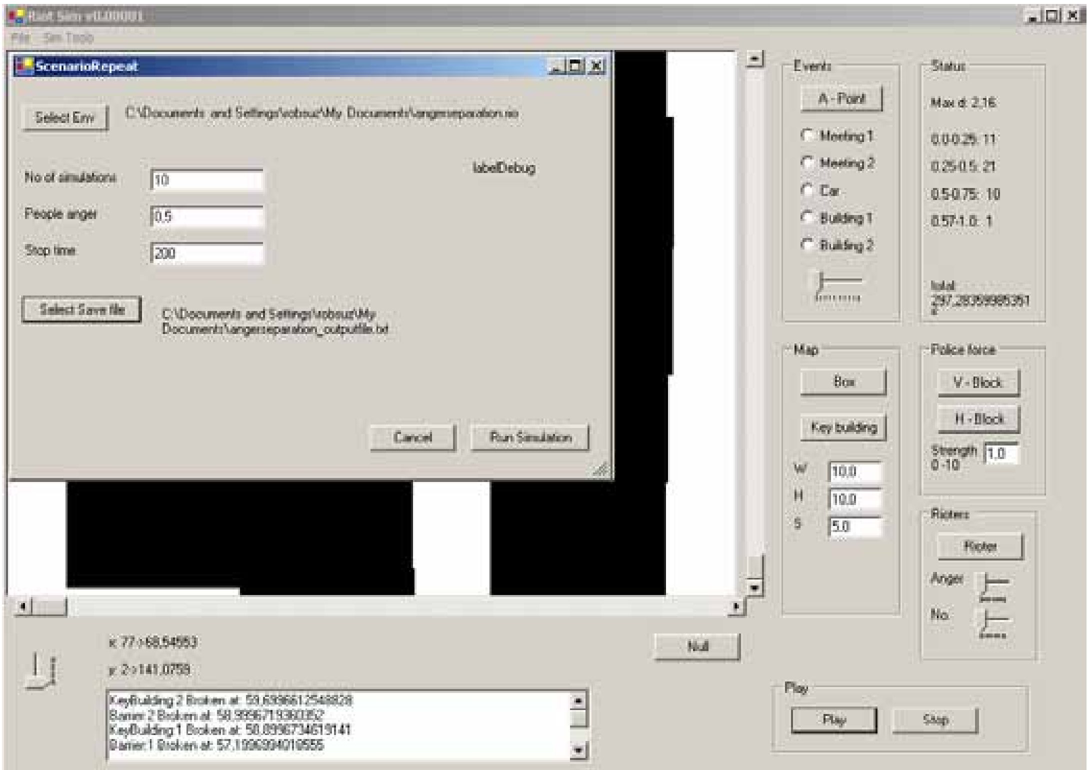


Figure 9. Scenario (situation) repeat function

In Appendix A we describe how we model rioters. In Appendix C we describe an error estimation for RCT could be implemented.

5.2.2 Results of RCS

The results of embedded simulations are written in a log file of simulated occurrences of effects see, Figure 10.

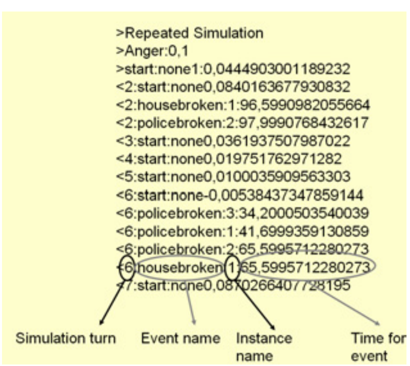


Figure 10. Result file of embedded simulation

In our case the effects are the negative effects of broken key buildings and broken (police/military) barriers. This file is a consequent description of events and it enables efficient analyses and decision refinement, improving and making decisions. Here, we have used the results of embedded simulations into BNs. The BNs as concept are desribed in Appendix B and error estimiation procedure in Appendix C.

5.3 Embedded simulations and Bayesian Networks

By deploying BNs, the situation can be separated into tractable pieces, see Figure 11. The CPTs represent the probability of a certain effect to occur, given that a potential combination of actions is executed. Consequently, a rich and transparent model can be expressed, greatly facilitating predictive situation awareness. The representation power of BNs can be further extended by introducing utility nodes by which the utility of a certain state distribution can be measured. Since the situation evolves over time, there is a need for changing the mechanisms that explain the uncertain causal relations between actions and effects in real time. Hence, to support the continuous re-evaluation of the joint probability distribution (JPD), *P(Decisions, Local Effects, Aggregated Effects),* we need adaptive and *context-dependent* BNs. The work of defining the JPDs is, however, very cumbersome and time consuming. In [Das, 2004] a mental simulation is proposed when assessing CPTs. Instead, in [Suzić, and Wallenius, 2005] we propose using *agent based* embedded simulations to model the mechanisms, and thereby to adapt the predicted effects as the time passes. Hence, the BNs are used to represent the whole tactical situation in order to synchronize the actions of all own units involved in the operation. In turn, the JPDs of the BNs are calculated by embedded simulations predicting the effects given possible actions performed by separate own units and groups of protesters on the sub-tactical level.

Here, we explain how the output (results) ES can be used to maintain a model of an ongoing situation. This model contains of a BN and the commanders interface.

5.3.1. A concept of combining ES and BNs

Considering the fact that it is not possible to be certain of how a riot is going to develop but, at best, to know only the probability distribution of potential future situations given measures and counter-measures, we propose the use of embedded stochastic agent-based simulations. Riot control is a difficult and emerging issue [Gaskins *et. al.*, 2004]. It has become a research challenge to support riot control by the use of simulation and other artifacts. To model mechanisms that explain the causality between actions and effects, we need recurrent embedded simulations based on data describing the physical system. An embedded simulation incorporates a number of agent models with stochastic properties. To this end, it uses the computer based representation of the current situation to run a number of predictive simulations with different outcomes. In the next step, the results of outcomes are used for calculation of the local effects' conditional probabilities; i.e. the probability of an effect given pairs of measures and countermeasures.

ES uses the computer based representation of the current situation to run a number of predictive simulations with different outcomes. In the next step, the results of outcomes are used for calculation of the local effects' conditional probabilities; i.e., the probability of an effect given pairs of measures (e.g., restrict and separate) and countermeasures (e.g., Move close to X and Attach unit 1), see Table 1. A possible effect could be the destruction of an object of importance, *X,* given that rioters will take the action to move closer to *X* and our own forces will take the action to rope off the area nearby:

Local effect Building X damaged	Restrict	Separate
Move close to X	P(Building X damaged| Restrict movement , Move close to X)	P(Building X damaged | Separate, Move close to X)
Attack unit 1	P(Building X damaged | Restrict movement , Attack unit 1)	P(Building X | Separate, Attack unit 1)

Table 1. Conditional probabilities of an effect

Hence, the probability of local effects is the statistical average of simulations running *N* times, each testing whether the effect has occurred or not, given a particular combination of measures and countermeasures. Let us denote a variable *effT*, representing the total number of times that an object has been destroyed given a certain measure A and a certain countermeasure R. We then define the simulated conditional probability according to the Laplace estimator [Wang, 2003] as:

$$P(EffectX = True \mid A, R) = \frac{effT + 1}{N + 2}$$

$$P(EffectX = False \mid A, R) = 1 - P(EffectX = True \mid A, R)$$

By running the embedded simulation several times, the resulting CPTs for the BNs are constructed, and thereby a context dependent JPD is being modeled. The simulations performed at a certain decision point in time are repeated to get a successively better estimation of the JPD. This continues as long as the time for making the decision allows, and as long as it is considered meaningful depending on whether additional simulations are expected to improve the estimation.

The simulation introduced here is a combination of two simulation concepts. The first one is the non-homogeneous agent-based simulation. The second property is that the simulation is stochastic, having its analogy to Monte-Carlo simulations, by which sampling is used to represent prior knowledge; i.e. the agents' properties are sampled from prior distributions. In RCS as mentioned the only distribution that we sample is level of anger.

Each member of the crowd or a smaller group of crowd members can be represented in the computer simulation. In [Jager *et al*., 2001] the use of non-homogeneous simulation agents is recommended. They also suggest a prior distribution of 1 % of "hard core" (aggressive agents), 10 % of "hangers-on" (potentially aggressive agents), while the rest of the agents are "bystanders". The prior distribution could be valuable when we, without having any prior knowledge about participants, instantiate simulation agents by sampling their properties. E.g., depending on prior knowledge, values of properties such as fatigue can be sampled from a distribution and assigned to agents.

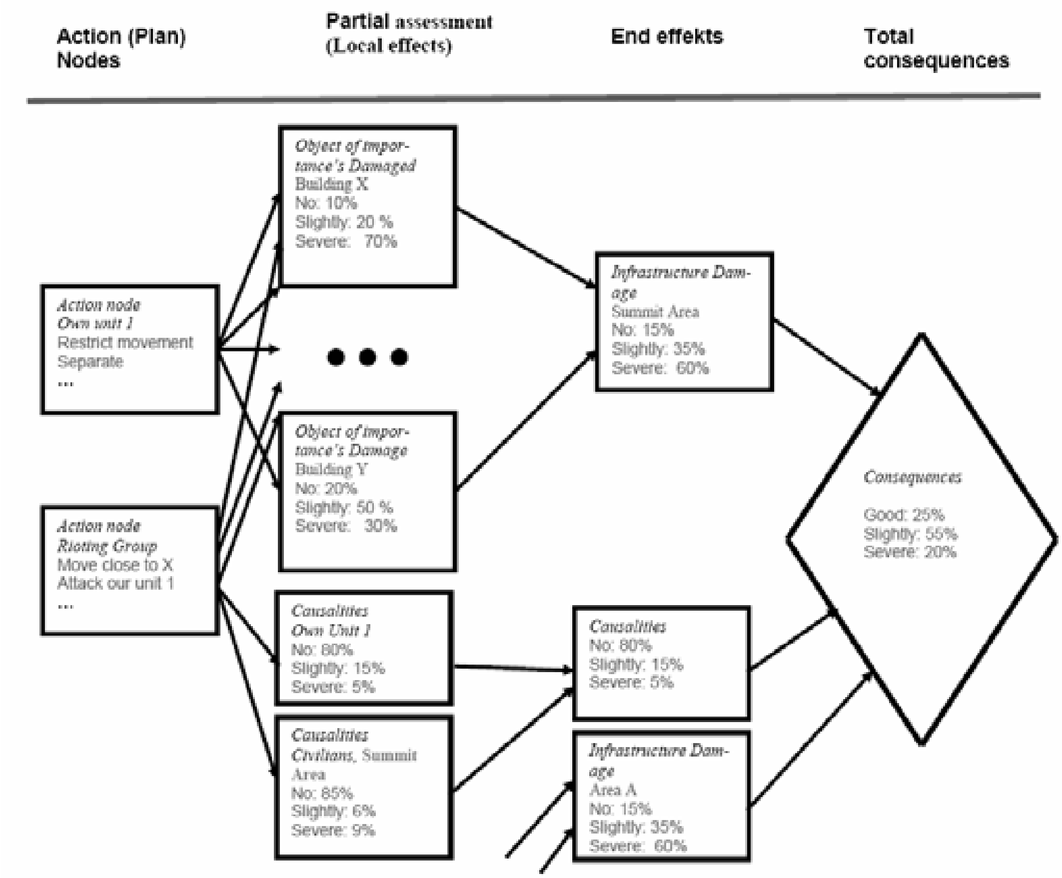


Figure 11. An example of a BN aimed to support mission planning and risk assessment [Suzić, and Wallenius, 2005]

We conclude this section by providing an example of a BN representing a subset of a riot situation. To this end, Figure 11 illustrates a small part of the BN that underlies the user example presented in Figure 15. On the left (in the first column) in Figure 11, the *action nodes* represent the plan alternatives that are to be evaluated. By the embedded simulation, as described in Section 3, the CPTs for combinations of own forces and rioting group actions are calculated. Those CPTs describe uncertain causal relations between the action nodes and *partial assessment nodes* (in the second column) representing predicted local effects. The local effects of interest could be a about particular objects of importance (such as buildings), the number of causalities of own units and of civilians in a certain area. Such effect could be the summit area in Figures 15. The *end effects* (in the third column), i.e. the effects in for whole area of interest depicted in Figure 11, are obtained given the states of the local effects. Due to its generic properties, uncertain causal relations between partial effects and end effects, i.e. their corresponding CPTs are assumed to be possible to model before a riot has occurred. E.g., given that several buildings (objects of importance) are going to suffer severe damage, the end effect on the damaged infrastructure would also be considered as severally damaged. The *utility node* on the right of Figure 11, represents the overall assessment of the predicted end effects considering the strategic priorities, stated by the superior commander. The superior commander can use the BN network to examine causalities between different effects. E.g. the superior commander can set different requirements using probabars (see 5.3.3) and get a “feeling” for how the effects would be related.

When context-dependent BN is built it could be complemented with plan recognition [Suzić, 2005], [Johansson and Suzić, 2005]. It performs a statistical guess what an agent or group of agents, rioting group here, is doing. By using plan recognition we would gain prior information about rioting group’s activities, i.e. plan (riot) nodes get a prior. This would even further enhance BNs context dependent representation and subsequently commander’s predictive situation awareness.

5.3.2. Experimental results with RCT's and BNs

We used RCS to asses the local effects in an ongoing situation. In the next step we used the output of the RCS, see Figure 10, to automatically construct a situation-relevant (context-dependent) BN.

Here we explain an experiment where we used RCS to swiftly construct and test different what-if cases that are a combination of different decisions. In an ongoing situation it is possible to test a *limited number* of cases (plans) where the human capacity is used to make judgment about cases that the user wants to test. The simulation took significantly shorter time than real time and we showed that even in complex scenarios embedded simulations can be used.

We applied four different cases for the same COP, see Figure 12. The user, tactical commander in this case can test different strategies of placing own forces (dark blue line), responsible for keeping security. The light blue squares represent buildings of importance while black square boxes represents buildings and restrictions in moving in the environment.

The goals set by the strategic commander are protecting the summit, right to demonstrate, protect private property and minimize number of casualties. To evaluate (see C2 model in [Wallenius, 2005]) different alternatives (plans) by using the embedded simulation that satisfies the goals, the tactical commander in his turn tests two strategies "Plan A" and "Plan B". The "Plan A" is a strategy of separation. The separation means to place own forces in a place where protesters passes by and where only "angry" (troublemakers) attacks own forces and protesters passes by; here, the peaceful protesters are separated from the angry protesters. The "Plan B" is the blocking alternative where protesters are prohibited of their right to protest but this alternative provides increased security for summit. However, if pressed into a small area the peaceful protesters may turn into angry protesters and a peaceful demonstration may turn into riot. Those can act against own forces in much larger proportions than initially and with much higher motivation. They could eventually break down commanders forces and threaten the summit seriously. We can also assume that the protesters have some alternative plan(s) to beside the plan, denoted here as "Plan One", to protest outside the summit building. The "Plan Two" is a plan of a part of protesters (troublemakers), plan to demolish private property in a shopping street.

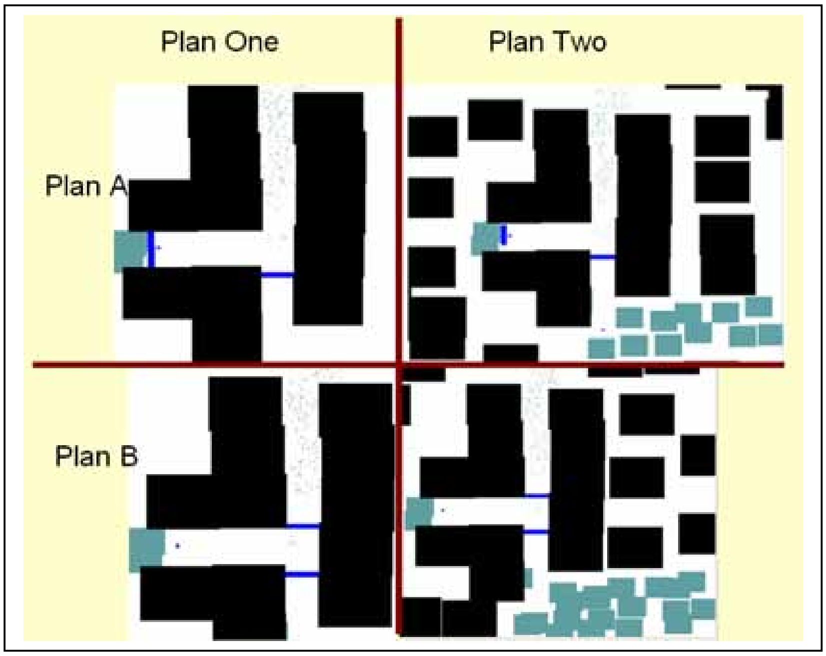


Figure 12. Four What-if cases with different combinations of plan alternatives

All events during the embedded simulations run are reported in a form of the (text) output file, see Figure 10. The RCS's file is machine-interpreted and converted by a C++ program that calculates probabilities according to Laplace estimator for different effects given different combinations of plans. Given the tables of probabilities, CPTs, for local effects; depending on the number of effects and their type, the end effect tables are automatically generated. In the next step, a XML file is also automatically generated by the C++ program. The XML file is, by the C++ program call, opened and interpreted by open source tool Genie as a BN, see Figure 13.

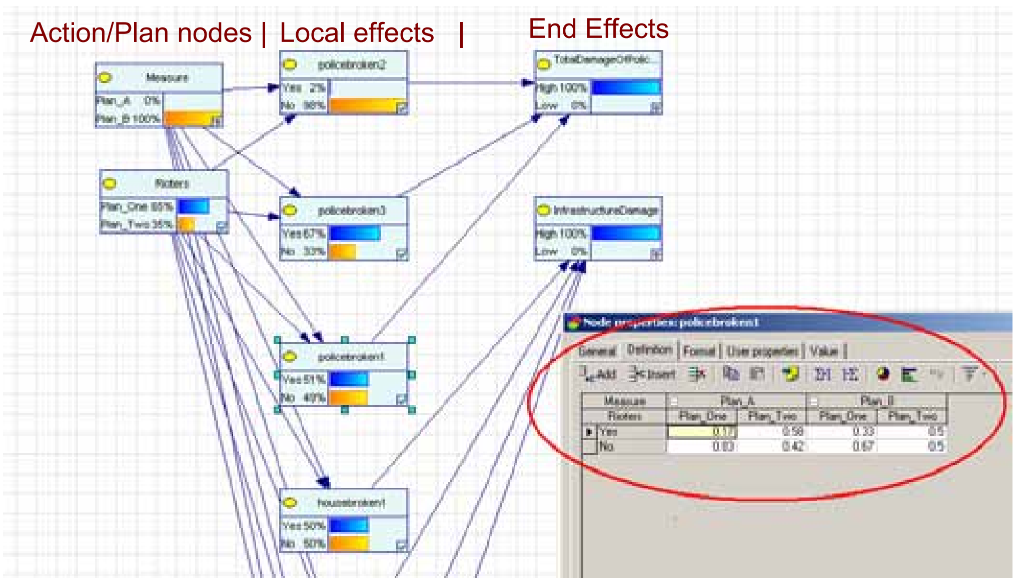


Figure 13. A context-relevant BN in Genie that is aimed to support assessment of effects given different plan alternatives

By elaborating with the context-relevant BN, the analyst or the tactical commander can use the network to get "felling" how a situation can evolve; e.g. to get an answer to question which decisions may lead to disastrous outcomes?

5.3.3. The tactical commander's operational view by using probabars

In this subsection, we summarize the concept of probabars introduced by [Wallenius, 2005]. A probabar is alternative way of graphical representation of probability distribution(s).

Let us look closer at a chance (Bayesian network) node with the name "Goal: No Casualties" in Figure 14.

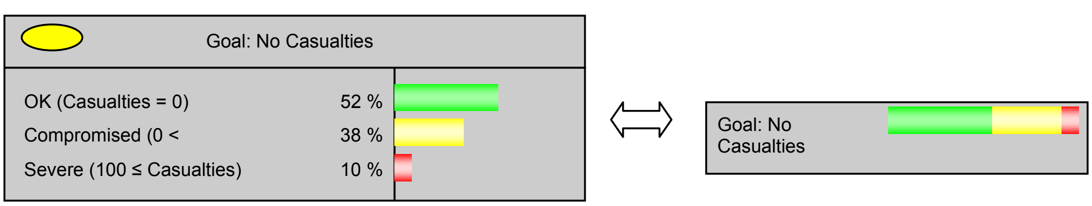


Figure 14. From Chance node to a single probabar.

The representation on the left is Bayesian node with bars and probabilities while a *probabar* on the right side contains one bar with, out any numbers. However, the user can *visually* asses proportion of state "OK" contra other states.

This concept is a practical way of mapping BNs, probabilistic, representation to a more convenient representation that is suited for personnel that is not familiar with large probabilistic models. Moreover, probabars aims at facilitating predictive situation awareness by a brief glance at the user's interface.

As a part of setting the strategic priorities, all potential outcomes need to be classified according to this scheme in advance. Thus, since the probabilities will always sum up to 1, it is possible to present a rough indication of the relations between the probabilities by use of the probabars.

Figure 15 suggests how a BN resulting from the simulations could be used by a tactical commander. On the left in the figure, the situation picture is depicted with positions of, e.g., important objects, protesters and own units. On the right, the assessed outcome is presented, given the current tasks assigned to the own units. For each of the strategic goals, the probabilities of that

- the outcome will be in accordance with the goals (*Ok*),
- the goals will not be fulfilled (*Compromised*),
- the outcome will be disastrous (*Severe*),

are presented.

For example, the probabar in connection labeled 'Protect Meeting and VIP' indicates that the probability that the top meeting will be performed without disturbances is around 80 %, while the probability of that the meeting will have to be stopped for some time is around 10 %, and finally, that the probability of that, e.g., the meeting must be cancelled or that some of the delegates will be injured is also around 10 %. Also the overall assessment is presented, indicating that the probability for the total operation to fulfill the goals is around 40 %, and the probability for a disastrous outcome is as around 10 %. These are seriously bad odds, and thus some new decisions need to be made quickly by the commander.

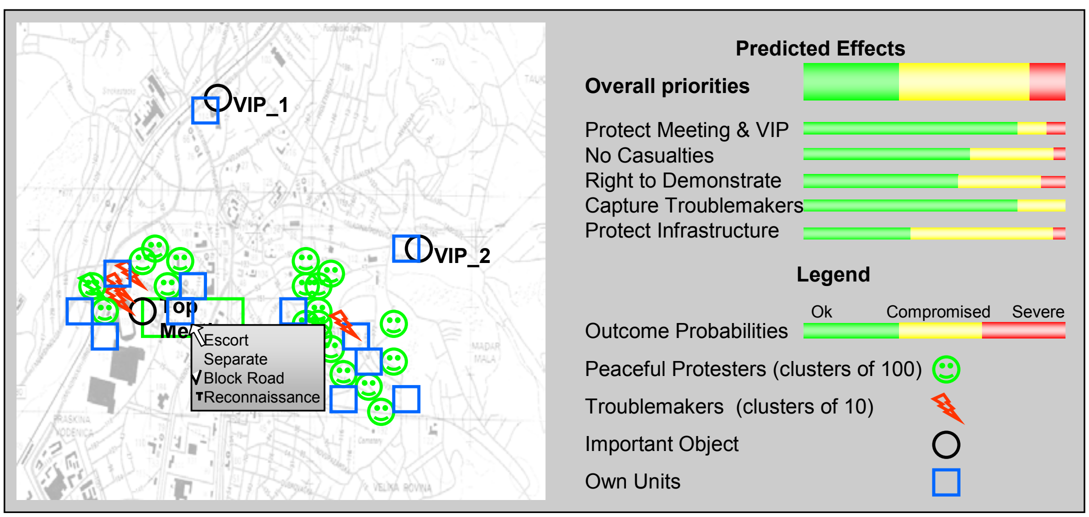


Figure 15. An example of a tactical commanders interface service presenting the assessed consequences of current actions, along with the situation picture. [Suzić and Wallenius, 2005]

By right-clicking on any of the own units, as depicted in Figure 15, new assignments could be given to them. At a first step, this could be made tentatively to test out new strategies for dealing with the threatening situation. Changing the assignments will give new information to be processed by the Influence Diagram and the Monte Carlo simulations. Possibly a strategy can be find that reduces the probability for severe consequences, which then will be indicated by the probabars. If so, the tentative assignments could be changed to definite assignments and thus ordering the subordinated units to perform according to the new strategy. Please note that the user interface presented in Figure 15 is designed merely to give an indication of how it might look like from a commander's point of view. However, to ensure the usability of the final design, iterative user studies will be essential for the future development.

6. 0 Conclusions and Future Work

In contrast to traditional (off-line) simulations, the ES are aiming to be efficient demand reusable and composable models valid within a particular simulation problem domain. Also a "deeper" (machine understandable) interoperability between C4I systems and ES applications is required.

One way to support integration of ES with the physical (other) systems is by the use of architectures. Two apparently valuable architectures from ES point of view have been summarized; those architecture originates from *different* domains (civil and military), different levels of how much the human controls the system and they are different in scale; the architecture from [Layman, Weatherly, 2003], see Figure 2, is more general than the architecture the architecture from [Low, Lye, Peter Lendermann, and Turner, 2005], see Figue 6.

However, there are some similarities between the architectures, beside the most obvious that they involve a physical system and simulation processes; the interface in [Layman, Weatherly, 2003] between the physical system and the embedded simulation is C4I system. The interface between a physical system in [Low, Lye, Lendermann, and Turner, 2005], covers some C4I's functions such as monitoring and controlling. System management agent in [Low, Lye, Lendermann, and Turner, 2005] integrates control of the user with the embedded system control and the physical system. Both archiectures has (some) functionallies of C2 systems.

Pro-active control of sensors enabled by (embedded) stochastic simulations by [Johansson and Suzić, 2004] is one of the examples of possible applications in the future NBD. However, this approach is demonstrated in a simulated environment and it needs further investigation.

In [Suzić and Wallenius, 2005] we have introduced a concept for consequence assessment in riot control on the tactical level. This concept has two major advantages. Firstly, the modeling of a complex system such as a rioting situation is performed by agent-based simulation that incorporates modeling of small subsystems interacting with each other, also stated in [Grieger, 2003]. This admits that the problem can be reduced to less complex problems that depend on properties that are more easily described. Secondly, the approach regards risks and uncertainties, since the sampling of agent properties admits the analysis of a variety of possible situations. By running a set of simulations we not only enhance the predictive situation awareness corresponding to the desired effects stated by the strategic priorities, but we will also be able to evaluate the robustness of our own decisions to different alternative actions performed by the rioters.

The concept presented in Chapter 5 needs to be further studied and integrated with a (emulated) decision system. The RCS needs verification and validation and further considering regarding complexity. Moreover, further research aims at getting a deeper understanding of the advantages and limitations of the concept. The interesting issues include how to deal with complexity in uncertain situations and to find out more about the challenges imposed by the use of embedded simulation. Also the need for user studies to structure the problem in terms of potential actions, important effects, etc., and to design appropriate user interfaces, calls for further emphasis.

7.0 References

Ahlberg, S., Hörling, P., Jöred, K., Mårtenson, C., Neider, G. Schubert, J., Sidenbladh, H., Svenson, P., Svensson, P., Unden, K. och Walter, J.
The IFD03 Information Fusion Demonstrator -- requirements, methodology, design, and experiences FOI-R--1413--SE, Avdelningen för ledningssystem, Totalförsvarets forskningsinstitut, Linköping, Sweden, 2004.

[Bergman, and Wallenius, 2001] N. Bergman and K. Wallenius. Providing the common view of the situation – the WASP approach. In *Information Management Challenges in Achieving Coalition Interoperability,* Quebec City, Canada, May 29-30, 2001.

[BNet] http://ilab.cs.ucsb.edu/BayesNets/Murphy_MIT_AI/BNT

[Brynielsson, 2002] Joel Brynielsson. A Decision-Theoretic Framework Using Rational Agency. In Proceedings of the 11th Conference on Computer-Generated Forces and Behavioral Representation, number 02-CGF-047, pages 459-463, Orlando, Florida, May 2002.

[Brynielsson, and Wallenius, 2003] J. Brynielsson and K. Wallenius. A Toolbox for Multi-Attribute Decision-Making. Technical Report TRITA-NA-0307, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, 2003.

[Brynielsson *et. al.,* 2005] J. Brynielsson, M. Engblom, R. Franzén, J. Nordh, and L. Voigt. Enhanced Situation Awareness using Random Particles. In Proceedings of the Tenth International Command and Control Research and Technology Symposium (ICCRTS), McLean, Virginia, June 2005.

[Daly, and Layman, 2002] J. Daly, G. Layman. C4I Tactical Applications Utilizing Embedded Simulations. Command and Control Research and Technology Symposium (CCRTS), June 11-13, 2002; Naval Post Graduate School, Monterey, California.

[Das, 2004] B. Das. Generating conditional probabilities for Bayesian Networks: Easing the knowledge acquisition problem. In *Journal CoRR cs.AI/0411034*, 2004.

[Doucet, Freitas, and Gordon, 2001] A. Doucet, N. D. Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice. Springer Verlag, 2001.

[Eklöf, Yi, and Suzić, 2005] M. Eklöf, C.H. Yi, and R. Suzić.Network Based Intelligence (NBI) – Knowledge Base Development and Use. User Report, FOI-R-1757-SE, Swedish Defence Research Agency (FOI), Stockholm, Sweden, 2005.

[Fujimoto, Lunceford, Page, and Uhrmacher, 2002] R. Fujimoto, D. Lunceford, E. Page, and A. M. Uhrmacher, editors. Grand Challenges for Modeling and Simulation. Dagstuhl Report, August 2002.

[Gaskins *et. al.*, 2004] R. C. III Gaskins, C. M. Boone, T. M. Verna, and M. D. Petty. Cognitive and behavioral psychological research for crowd modeling. In *NATO Modeling and Simulation Group Conference,* Koblenz, Germany, October 2004.

[Grieger, 2003] D. Grieger. An Overview of Crowd Control Theory and Considerations for the Employment of Non- Lethal Weapons. DSTO System Sciences, Technical report DSTO-GD-0373, Laboratory, Edinburgh, AU, 2003.

[Gülich , 2004] M. Gülich .A Simulation-Based Embedded System Prototype. Master thesis at Royal Institute of Technology (KTH), Stockholm, Sweden, 2004.

[Harrell, and Hicks, 1998] C. R. Harrell and D. A. Hicks. Simulation Software Component Architecture for Simulation-Based Enterprise Applications. *Proceedings of the 1998 Winter Simulation ConferenceD.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds*

[Helbing, and Molnar, 1995] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. Phys. Rev. E 51, 4282-4286, 1995.

[Helbing, Illes, and Tamas, 2005] D. Helbing, I. J. Farkas, T. Vicsek. Simulating Dynamical Features of Escape Panic. Nature 407, 487-490, 2000.

[Ho, 2004] J. Ho. The Dimensions of Effects-based Operations. Australian Army Journal. Volume II, Number 1, AU, 2004.

[Howard, and Matheson, 1984] R. A. Howard and J. E. Matheson. Readings on the Principles and Applications of Decision Analysis. pp 721—762, Strategic Decisions Group, Menlo Park, CA, US, 1984.

[Huang *et. al.*, 2003] Q. Huang, J. Hållmats, K. Wallenius, and J. Brynielsson. Simulation-based decision support for command and control in joint operations. In *European Simulation Interoperability Workshop 2003,* 03E-SIW- 091, pages 591-599, Stockholm, Sweden, June 16-19, 2003.

[Jager, 2001] W. Jager, R. Popping and H. van de Sande. Clustering and fighting in two-party crowds: Simulating the Approach-avoidance Conflict. In *Journal of Artificial Societies and Social Simulation* vol. 4, no. 3, 2001.

[Jensen, 1996] F. V. Jensen. An introduction to Bayesian Networks., UCL press, London, ISBN: 1-85728-332-5, 1996.

[Johansson and Suzić, 2005] L. R. M. Johansson, and R. Suzić. Particle filter-based information acquisition for robust plan recognition. In *proccedings of* 8th *International Conference on Information Fusion*, Philadelphia, PA, US, July 2005.

[Johansson, Mårtenson, Suzić, and Svenson, 2005] R. Johansson, C. Mårtenson, R. Suzić, and P. Svenson. Stochastic dynamic programming for resource allocation. Technical report, FOI-R--1666--SE, Swedish Defence Research Agency, Stockholm, Sweden, Linköping, Sweden, September 2005.

[Kirkpatrick, Gelatt, and Vecchi] S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi .Optimization by Simulated Annealing. In *Science*, Number 4598, 13 May 1983",Vol. 220, 4598, pp. 671--680, 1983.

[Layman and Weatherly, 2003] G. Layman, J. Weatherly. C4I Host Applications Simulation Interoperability Standards Organization 2003 Fall SIW Conference, Orlando,14-19, 2003 Paper No. 03F-SIW-053, September, 2003.

[Layman, Weatherly, and Robinson, 2005] G. Layman, J. Weatherly, D. Robinson. C4I Mission Application Software Development Kit (MASDK). Paper 05S-SIW-083, Simulation Interoperability Standards Organization, Simulation Interoperability Workshop, April 3-8, 2005; San Diego, USA, 2005.

[Low, Lye, Lendermann, and Turner, 2005] M. Y-H. Low, K. W. Lye, P. Lendermann, S. J. Turner. An agent-based approach for managing symbiotic simulation of semiconductor assembly and test operation. AAMAS Industrial Applications 2005: 85-92

[McCrabb, 2002] M. McCrabb, Effects-based Coalition Operations: Belief, Framing and Mechanism. Second International Conference on Knowledge Systems for Coalition Operations KSCO-2002 - 23rd and 24th April 2002 - Tolouse, France, 2002.

[Murphy, 2002] K. P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. University of California, Berkeley, USA, 2002.

[Mårtenson and Svenson, 2005] C. Mårtenson, and P. Svenson. Evaluating sensor allocations using equivalence classes of multi-target paths. In *Proceedings of the Eighth International Conference on Information Fusion (FUSION 2005)*, Philadelphia, USA, 25-29 July 2005. IEEE, Piscataway, USA, 2005.

[Osborne, and Rubinstein, 2001] M. J. Osborne., and A. Rubinstein . A Course in Game Theory. Cambridge (MA.): MIT Press., USA, 2001.

[Piccerillo, and Brumbaugh, 2004] R. A. Piccerillo and D. A. Brumbaugh. Predictive battlespace awareness: Linking intelligence. surveillance and reconnaissance operations to effects based operations", In *Proc. of the C2 Research and Technology Symposium*, San Diego, US, 2004.

[Petty, 2004] M. D. Petty. Simple Composition Suffices to Assemble any Composite Model. *Proceedings of the Spring 2004 Simulation Interoperability Worksho*p, Orlando FL, April 18-23 2004, 04S-SIW-048. (http://www.sisostds.org/)

[Schubert, Mårtenson, Sidenbladh, Svenson, and Walter, 2004] J. Schubert, C. Mårtenson, H. Sidenbladh, P. Svenson, and J. Walter. Methods and System Design of the IFD03 Information Fusion Demonstrator. In Proceedings of the 9th CCRP , Denmark, 14-16 September 2004.

[Suzić, 2003] R. Suzić. Knowledge representation, modelling of doctrines and information fusion. In Proceedings of the CIMI conference, Enköping, Sweden, May 20-22, 2003.

[Suzić and Wallenius, 2005] R. Suzić and K. Wallenius. Effects Based Decision Support for Riot Control: Employing Influence Diagrams and Embedded Simulation. In Proceedings of Workshop on Situation Management SIMA 2005, US, October 2005.

[Wagenhals *et. al.*, 2003] L. W. Wagenhals, A. H. Levis, and M. McCrabb. Effects-Based Operations: A historical perspective for a way Ahead. In *proccedings of 8th International Command and Control Research and Technology Symposium,* National Defense University, Washington, DC, June 17-19, 2003.

[Wang, 2003] J. Wang. Data Mining: opportunities and challenges. Idea Group Publishing, ISBN 1-59140-051-1, US, 2003.

[Wallenius, 2005] Klas Wallenius. Generic Support for Decision-Making in Effect-Based Management of Operations. PhD thesis, ISBN 91-7178-234-6, Royal Institute of Technology, Stockholm, Sweden, November, 2005.

Appendices list

Appendix A: Behavior of agent models that is used in RCS

The behavior of agent models of rioters and peaceful demonstrators contain three parts.

The first part is crowd dynamics part. The research of professor Dirk Helbing, [Helbing, and Molnar, 1995], has been a guideline for describing how a group of agents moves through an environment and interacts with each other. According to him, the motion of pedestrians can be described as if they would be subject to `social forces'. The social force is according [Helbing, and Molnar, 1995], [Helbing, Illes, and Tamas, 2000] a sum of inertial, repelling and attracting forces. Repelling force between agents is modeled in RCS as exponential function according to formula 13 in [Helbing, and Molnar, 1995]. The attraction force in our case is dependent on distance to places of interest. Those places include meeting points, key buildings and places where own (police or military) forces are located. In the RCS the inertial forces has not been used. Helbing's computer simulations of interacting pedestrians show that the social force model is capable of describing the self-organization of several observed collective effects of pedestrian behavior very realistically. I.e. Helbings results have been submitted to verification and are proven to have high accuracy in crowd modeling.

The second part of the behavior model is about interaction between different agents and objects. It is modeled mainly in heuristic manner and additional studies are needed. The summary study of [Grieger, 2003] modeling has been studied as well. The interaction between rioters and own force is matter of force balance, i.e. number of own forces and (angry) rioters. If the own forces are stronger than rioters then own force barrier is not broken and there are no casualties. In other case a broken barrier is related to number of own force casualties.

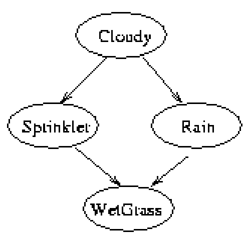
The third and final part is the statistical sampling part. It is limited here to sampling just one property of each agent. Given an initial distribution of demonstrators anger values for each agent are sampled different behaviors of agents are manifested depending on level of anger; this implies that different outcomes (effects) occur at each turn of the embedded simulation.

Appendix B: Bayesian networks

Here we give a short description of BNs that are aimed to support a decision maker in making decisions that involve uncertainty and several (potentially conflicting) goals.

Bayesian Networks (BN) is a statistical modelling method used to represent uncertain causal relations between different statistical variables. By using BN methodology it is possible to deal with uncertainty in a uniform and scientifically correct manner. Each node represents a variable that can be either discrete or continuous. Variables and its states are represented by conditional probability distributions also called subjective probabilities. BN is also denoted *belief network* since they describe our belief about the state of the variables. An advantage of the BN is that our knowledge is implemented in a piecewise manner. We only have to "explain" how a particular node depends on its parents. E. g. in Figure A, we define the probability density function of the variable *WetGrass*. The variables that make direct influence on the variable are called parents. *WetGrass* in this example has parents *Sprinkler* and *Rain*. The a priori probability density function of variable *WetGrass* does not model influence of the variable *Cloudy*. However, let us say that new evidence arrives. The statement of the new evidence is that we know that the weather is cloudy, *Cloudy* = True, this evidence will propagate through the network and make influence on our belief about if grass is wet or not. The process where weighing the new evidence with our subjective, a priori, knowledge is performed is denoted statistical inference.

P(C=F)	P(C=T)
0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Figure A. An Example BN [BNet]

When using BN we can infer evidence in all directions by using Bayes rule. For example, we can answer the question what is the probability that the sprinkler was on if we perceive that the grass is wet. The causal relations give only a description of the model.

Uncertainty in action, result in EBO and cause requires a probability-based approach. BN provide such an uncertainty based representation of mechanisms. BNs represent a model of the evolving situation by dealing with uncertainties in scientifically sound manner.

According to [Jensen, 1996] the formal definition of BN is:

- A set of variables and set of directed edges
- Each variable has a finite set of mutually exclusive states
- The variables together with the directed edges form directed acyclic graph (DAG)
- To each variable A with parents B1 .. Bn there is attached conditional probability table P(A| B1 .. Bn), abbreviated as CPT.

Mathematically expressed:

$$P(X_1..X_n) = \prod_{i=1}^{n} P(X_i \mid par(X_i))$$

Where $\boldsymbol{n}$ is the number of the nodes in the network and $\boldsymbol{X_i}$ represents a stochastic variable no. $\boldsymbol{i}$ of the BN.

When we describe a time-dependent BN we speak about Dynamic Bayesian Networks (DBN). It consists of several layers of BN with the same structure. The additional influences in DBN are the variables of the previous step(s) that make influence in variables for future step(s). Note that the term "dynamic" means that we are modelling a dynamic system, not that the network changes over time [Murphy, 2002]. The variable values changes over time but the network topology remains same.

Appendix C: Error estimation

There are two different types of errors in our model. The first one is the simulation model error that occurs due to lacking knowledge or model approximation of realty. This error is hard to estimate. The tests that compare reality and the simulation model output would be a first step towards the verification of the model. However, even if a verification process would be applied it is hard to obtain a quantitative estimate.

The second type of error is the noise estimate and it is possible to calculate. It is dependent of the number of the embedded stochastic simulations. In the scenario where embedded simulations are used the assumption is made that the effects are conditionally independent. The output of the each simulation loop during an embedded simulation tells whatever a local effect has occurred or not. After running repetitive simulation let as say M times we achieve the result a beta-distribution. The error of noise estimate becomes a standard deviation of a beta-distribution.