FFI RAPPORT

SYNTHETIC ENVIRONMENT OF EXPERIMENT "AD HOC ORGANIZATION OF PICTURE COMPILATION" - Technical Evaluation Report

GUSTAVSEN Richard Moe, MEVASSVIK Ole Martin,
SKJELTORP Arild

FFI/RAPPORT-2004/02554

SYNTHETIC ENVIRONMENT OF EXPERIMENT "AD HOC ORGANIZATION OF PICTURE COMPILATION" - Technical Evaluation Report

GUSTAVSEN Richard Moe, MEVASSVIK Ole Martin,
SKJELTORP Arild

FFI/RAPPORT-2004/02554

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
P O Box 25, NO-2027 Kjeller, Norway

FORSVARETS FORSKNINGSINSTITUTT (FFI)
Norwegian Defence Research Establishment

P O BOX 25
N0-2027 KJELLER, NORWAY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

REPORT DOCUMENTATION PAGE

1) PUBL/REPORT NUMBER FFI/RAPPORT-2004/02554	2) SECURITY CLASSIFICATION UNCLASSIFIED	3) NUMBER OF PAGES 22
1a) PROJECT REFERENCE FFI-II/892/912	2a) DECLASSIFICATION/DOWNGRADING SCHEDULE -	

4) TITLE

SYNTHETIC ENVIRONMENT OF EXPERIMENT "AD HOC ORGANIZATION OF PICTURE COMPILATION" - Technical Evaluation Report

5) NAMES OF AUTHOR(S) IN FULL (surname first)

GUSTAVSEN Richard Moe, MEVASSVIK Ole Martin, SKJELTORP Arild

6) DISTRIBUTION STATEMENT

Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)

7) INDEXING TERMS

IN ENGLISH:	IN NORWEGIAN:
a) Blue Game 04	a) Blue Game 04
b) Synthetic environment	b) Syntetisk miljø
c) Simulation	c) Simulering
d) High Level Architecture	d) High Level Architecture
e) Experiment	e) Eksperiment

THESAURUS REFERENCE:

8) ABSTRACT

This report documents the synthetic environment developed for experiment "Ad hoc organization of picture compilation" conducted during the NATO exercise Blue Game 04, with emphasis on technical issues. The main challenges are described, most of them introduced as a result of the use of satellite communications. Several possible designs are presented, together with discussions of whether or not they satisfy specified requirements. The selected solution is described in detail.

9) DATE	AUTHORIZED BY This page only	POSITION
2004-08-12	Vidar S Andersen	Director

ISBN-82-464-858-5

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE
(when data entered)

CONTENTS

		Page
1	INTRODUCTION	7
1.1	Report outline	8
2	EXPERIMENT OVERVIEW	8
3	SCENARIO	9
3.1	Sensor models	10
3.2	Visual database	11
4	CHALLENGES	12
5	SE ARCHITECTURE	12
6	SOFTWARE CONFIGURATION	15
6.1	Data exchange between the SE and the technology demonstrator	16
6.1.1	Redesigning the Reporter	17
6.2	Centralized RTI server	18
6.3	Using a different RTI implementation	19
7	DATA LOGGING AND ANALYSIS	20
8	SUMMARY	20

SYNTHETIC ENVIRONMENT OF EXPERIMENT "AD HOC ORGANIZATION OF PICTURE COMPILATION" - Technical Evaluation Report

1 INTRODUCTION

In May 2004 FFI conduced the exploratory experiment “Ad hoc organization of picture compilation”. The experiment was conducted during the NATO exercise Blue Game 2004. This report documents experiences with development and use of a synthetic environment (SE) in the support of the experiment.

The “Ad hoc organization of picture compilation” experiment had two primary objectives:

- Explore the potential operational value of ad hoc organization
- Run a picture compilation technology demonstrator and its synthetic environment in an operational setting

The experiment addressed aspects of ad hoc organization of picture compilation applied to the Common Operational Picture (COP), by utilizing Commercial Off The Shelf (COTS) technologies and open standards.

Three FFI projects participated in the experiment:

- **Network Based Defense (NBD) Decision Support**: The project explored the operational value of the ad hoc organization concept.
- **SIMUTREX**: Developed a synthetic environment stimulating the ad-hoc demonstrator with track messages. SIMUTREX is an ongoing project at FFI investigating how SEs can be assembled and utilized for training and experimentation.
- **NBD Grid**: Provided the network support for the ad hoc demonstrator. In addition they investigated communication network protocols.

The development of the SE was conducted during three months from the beginning of February 2004. The experiment took place 22-31 April 2004. An effort of 1200 man-hours was spent on developing and evaluating the SE. The evaluation report of the experiment is found in (1).

Sensor data originating from a realistic scenario was needed in order to conduct the experiment, as no “live” sensors were available. For that reason, an SE was developed in order to stimulate the picture compilation technology demonstrator with track messages.

The SE was based on the High Level Architecture (HLA), a standard architecture for interconnecting simulation components. The environment was developed using ‘commercial off-the-shelf’ (COTS) components and existing simulation models developed at FFI. This report documents the design of the SE, together with experience gained through out the development process.

1.1 Report outline

Chapter 2 gives an overview of the SE. The scenario is briefly described in chapter 3. Chapter 4 explains challenges and obstacles identified for realizing the SE, while chapter 5 describes the physical architecture, including alternative architectures. Chapter 6 describes various software configurations, including discussions of design decisions versus identified obstacles. A short summary is given in chapter 7.

2 EXPERIMENT OVERVIEW

The technology demonstrator and hence the SE was distributed at three different locations (Figure 2.1): The naval vessels HNoMS Otra and HNoMS Vidar and the National Joint Headquarters (NJHQ). Each location consisted of a Picture Compilation Node (PCN) and part of the SE that stimulated the PCN with simulated track data. The PCNs collaborated using service discovery middleware (JXTA), to support the ad hoc organization of picture complication participants and distributed picture compilation.

The PCN on board HNoMS Otra acted as the frigate HNoMS Bergen, the PCN located onboard HNoMS Vidar simulated an Unmanned Aerial Vehicle (UAV), while the PCN at NJHQ simulated a Maritime Patrol Aircraft (MPA). The MPA arrived late in the scenario and hooked into the network on an ad hoc basis.

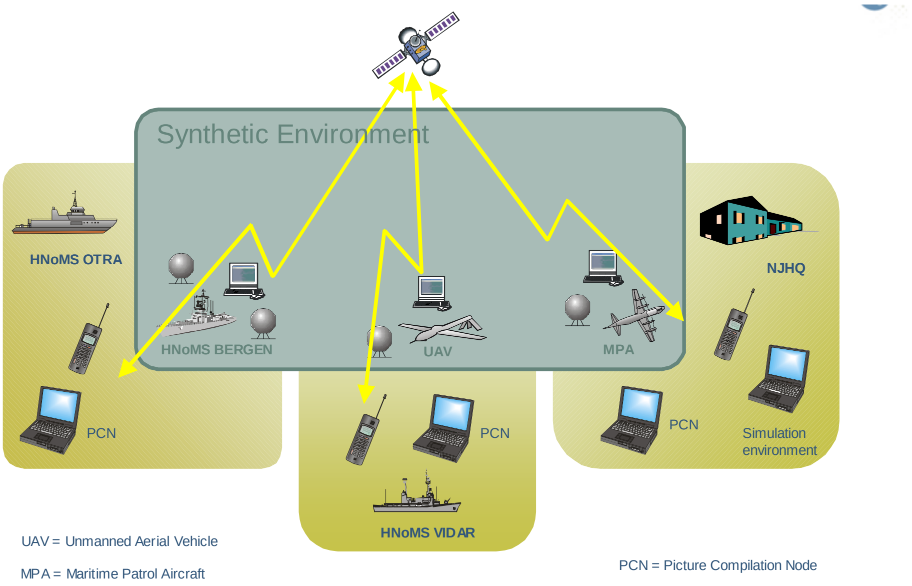

Figure 2.1 Technical set-up

The network consisted of LANs interconnected using satellite telephones. There were separate satellite connections for the SE and the technology demonstrator. In addition, NJHQ had access to the Internet via NATO Unclassified Local Area Network (NULAN). The SE did not use NULAN.

Both HNoMS Otra and HNoMS Vidar had a GPS feeding real-life positions into the picture compilation technology demonstrator and the SE. Thus, the SE was a mixed live and constructive simulation.

3 SCENARIO

The scenario was based on the Blue Game 2004 order of battle and planned events: A Mine Countermeasures (MCM) force escorted by the frigate HNoMS Bergen was subject to coordinated attack from two F-16s and a jet ski while transiting up the Oslo fjord (see Figure 3.1). The order of battle included 11 blue naval vessels, where 7 made the main body. An MPA and an UAV conducted surveillance in support of the blue force. The opponent was four fast patrol boats, a jet ski and two fighter aircrafts. Civilian traffic was also present in the area as shown in Figure 3.1.

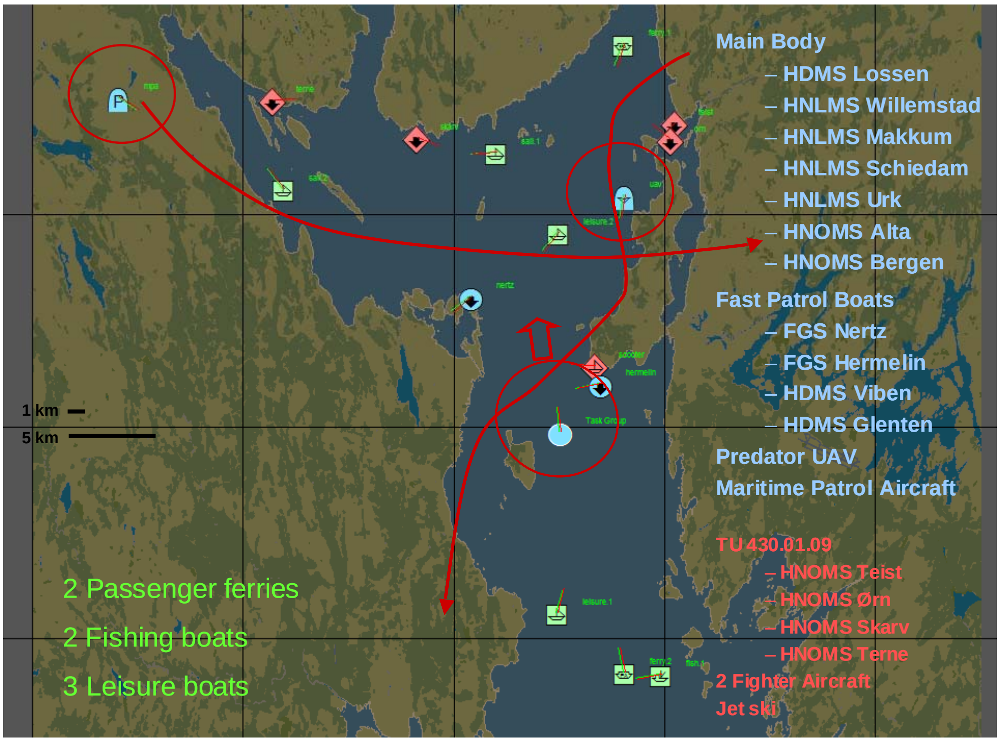

Figure 3.1 Scenario overview

The sensors on three military platforms, an UAV, an MPA and HNoMS Bergen, were simulated. The red lines in Figure 3.1 show the general movements of the simulated sensor platforms.

The following sensors and sensor platforms were simulated (see Figure 3.2):

- An UAV fitted with an electro-optical (EO) sensor.
- An MPA conducting standoff surveillance using EO sensor and Electronic Support Measures (ESM) sensor.
- HNoMS Bergen produced a local picture using radar (covering both surface and air) and an EO sensor. The local picture included own reports from the blue vessels.

Synthetic sensors and simulated data fusion nodes produced tracks and classification reports which were distributed to the PCNs.

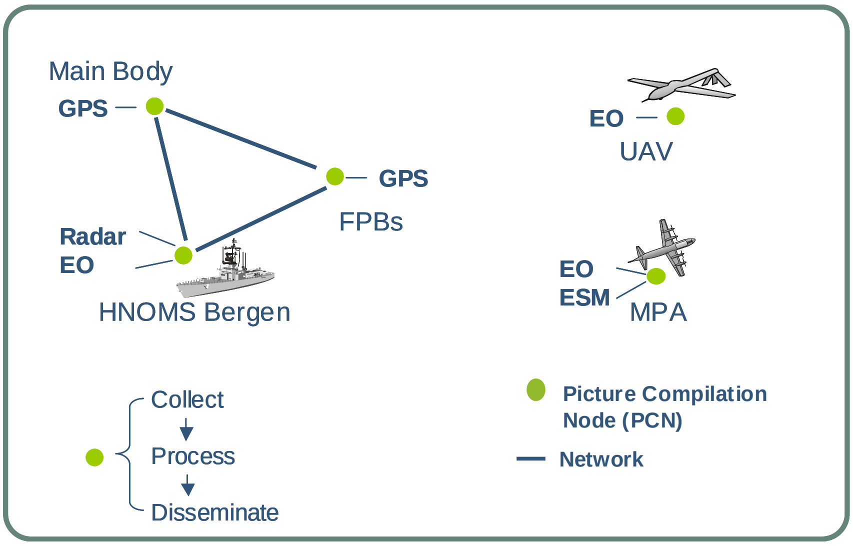

Figure 3.2 The simulated sensors connected to the PCNs

3.1 Sensor models

An FFI developed simulation, SensorSim (6)-(8), was used to simulate sensor detection, tracking and classification. The movement of the platforms included in the scenario was simulated by VR-Forces (4).

The generic sensor model implemented by SensorSim uses a line of sight detection model and 360 deg sensor coverage. The terrain database used was DTED Level 1. The sensor detection model has a minimum and a maximum detection range. These detection ranges apply to targets of all types (and in all environments). Sensors with classification capabilities have 5 levels of classification spanning from detection only (type of object unknown) to unique identification of a specific object. The level of classification produced by visual sensors is range dependent.

The sensor models do not include false detections, correlation errors (plot-to-track or track-to-track) or erroneous target classification reports.

Note that the sensor configuration data used in the experiment did not correspond to the real capabilities of the platforms/sensors. The capabilities were chosen in order to produce a fairly realistic picture and an appropriate load on the picture compilation demonstrator.

HNoMS Bergen was equipped with surface and air surveillance radar with maximum ranges of 20 km and 15 km respectively. An EO sensor with 5 km range was used to identify vessels and air targets.

The EO and ESM sensor on the MPA had a maximum range of 10 km and 80 km respectively. The ESM sensor was capable of recognizing vessel type but had no fingerprinting capability (not able to recognize vessel identity).

The UAV was equipped with an EO sensor able to detect land and surface objects. The EO sensor had a maximum range of 8 km.

3.2 Visual database

The visual database used by the 3D stealth viewer (MÄK Stealth Viewer) was built using the TerraVista software from Terrain Experts Inc (3).

The following data sources were used to generate the visual database:

- DTED Level 1 elevation data
- A subset of the N50 geographic feature data
- Landsat satellite images (30 m resolution)

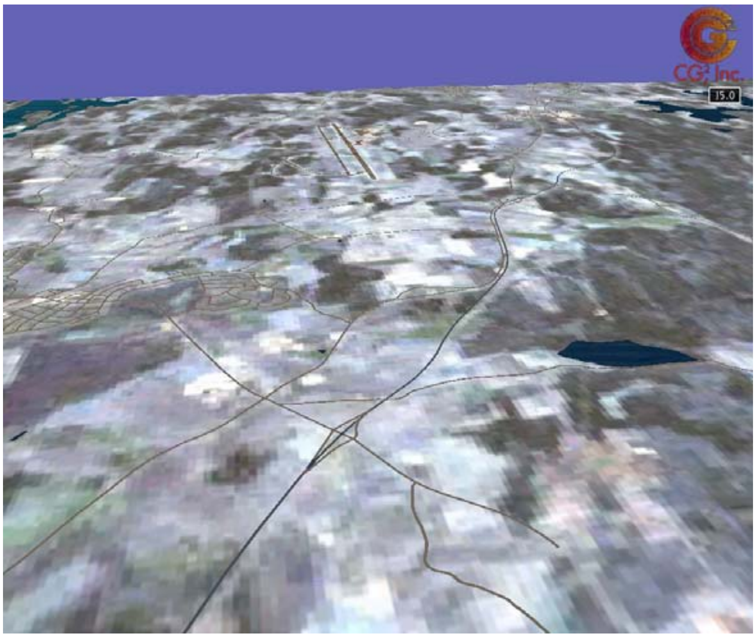

Figure 3.3 Lowest level of detail used in terrain database

The visual database was built with two different levels of detail and covered the area Moss-Horten. As a base for the whole area of interest (size 50 km N x 57 km E), the main data was the DTED elevation data with the satellite images used as texture. To build this area, some basic feature data (shorelines, lakes/ponds, airports and roads) were used as shown in Figure 3.3.

The most interesting part of the database (size 17 km N x 21 km E) was built with a higher level of detail, using additional data from the N50 features (forest, urban areas, crop land and buildings) as shown in Figure 3.4.

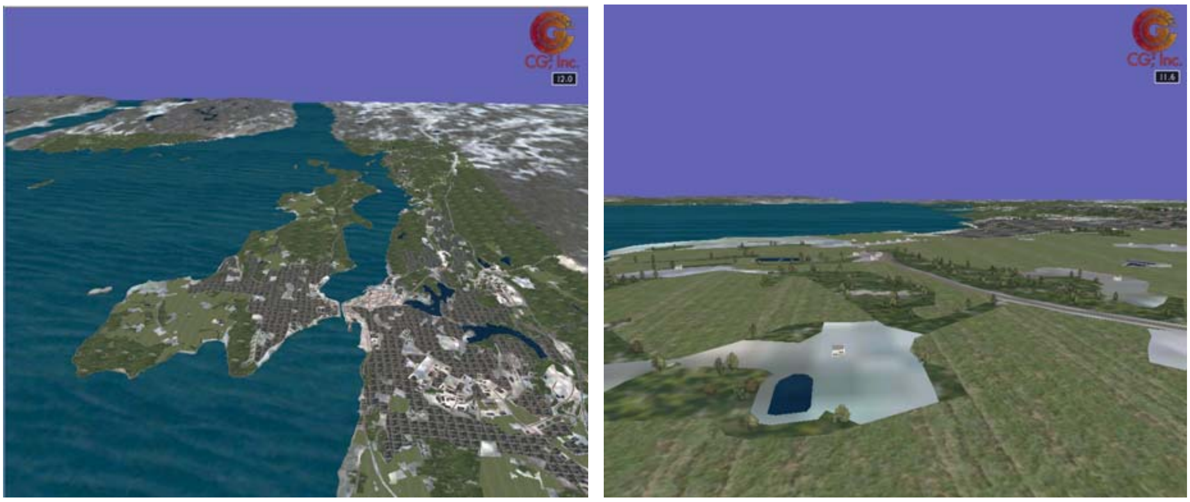

Figure 3.4 Highest level of detail

The 30 m resolution of the satellite image proved too low except for observations of a large area from high altitude.

The final visual database in OpenFlight format had a total size of 220 MB. One man-week was spent building the visual database.

4 CHALLENGES

The main challenge developing the SE, was to interconnect independent simulation components using satellite communication. Satellite communication was needed in order to transmit track messages from the main site of the SE, NJHQ, to HNoMS Otra and HNoMS Vidar. Until Blue Game 2004, most of our experience was limited to simulations connected to the same LAN. Such networks usually have high bandwidth (100 Mbs), and small latency.

For this experiment, two different satellite communication systems were considered as possible data carriers: Globalstar and Iridium. They both were, however, constrained to the same limitations. They offered a bandwidth between 2.4 kbs and 20 kbs, and had latencies as high as 6 s. Moreover, when connecting to either of the satellite systems, the computers were given temporary, non-routable, IP addresses. This meant that other computers could not reach them from the Internet unless they were connected to the same satellite communication system. Furthermore, if a computer got disconnected from the satellite network, chances were that it would receive a different IP address upon reconnect. Finally, the satellite systems did not support IP multicast.

To resolve the abovementioned limitations, both physical architecture and software had to be carefully configured. The next two chapters will describe in detail the solutions that were tested together with alternative solutions that were not tested.

5 SE ARCHITECTURE

The SE was implemented using six laptop computers running Windows XP and Red Hat Linux. Four computers were connected to a local area network (LAN) at NJHQ, and one was connected to each of the LANs onboard HNoMS Otra and HNoMS Vidar (see Figure 5.1). The latter two computers and one of the computers at NJHQ also served as Wide Area Network (WAN) gateways for the SE by using the low bandwidth/high latency satellite communication system Iridium. Chapter 6.1 contains a discussion regarding Iridium versus Globalstar.

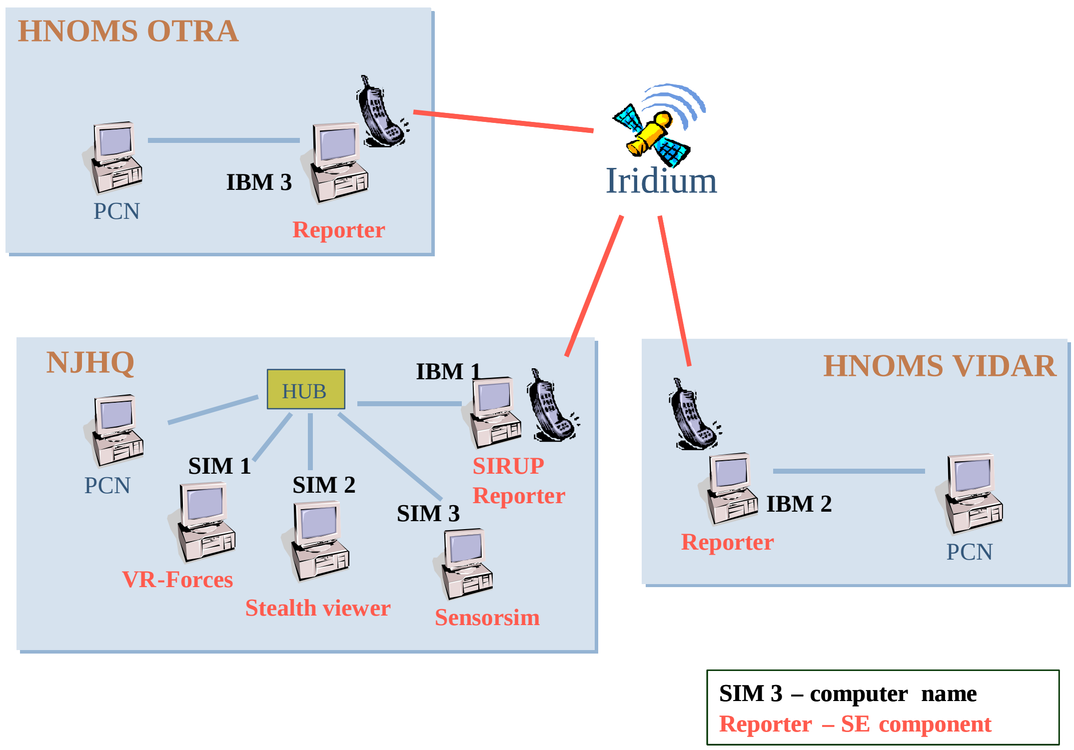

Figure 5.1 Physical architecture

Although the before mentioned design was chosen, other designs were considered early on. Firstly, we would have preferred the LAN at NJHQ to be connected to the Internet rather than satellite communication (see Figure 5.2). In this way the SE could have been implemented using fewer Iridium/Globalstar phones. The solution would imply (although not proven) higher throughput, and the network to be more stable. Such a design would also be more scalable. The reason for this is that all data flows through the single satellite telephone at NJHQ due to the distribution of the SE components.

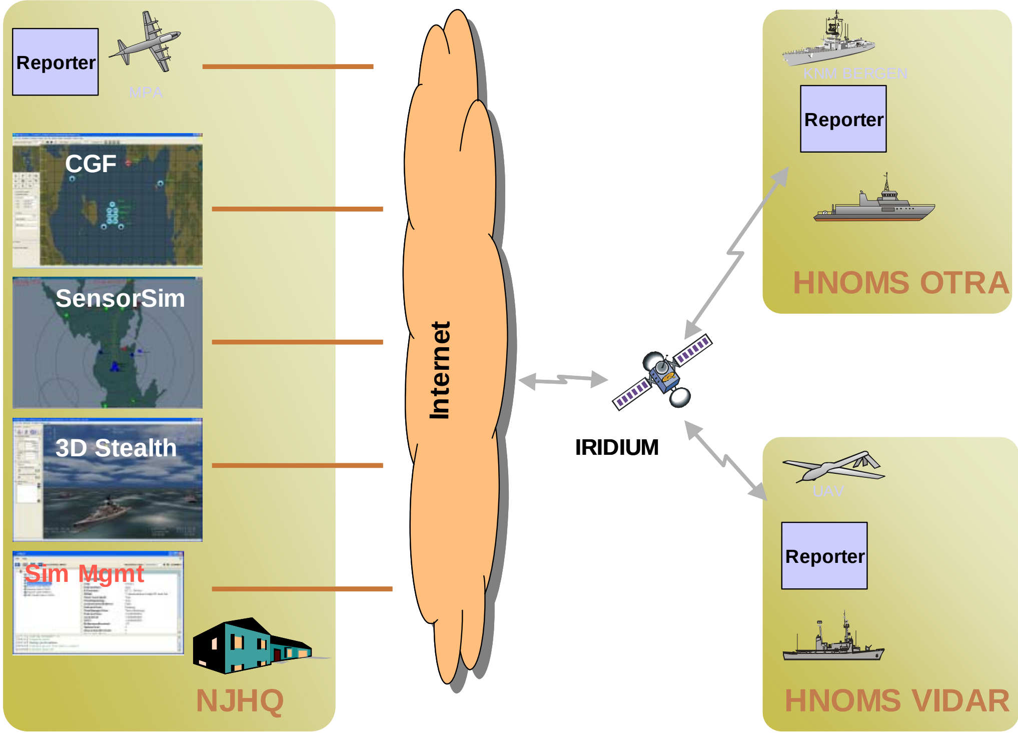

Figure 5.2 An alternative, not valid, architecture using Internet.

Although the Internet solution might seem applicable, it has a shortcoming that made it unworkable. When a computer connects to Iridium or Globalstar, it receives an IP address that is not routable outside the satellite communication network (class B private address space). This means that if two federates connect to each other, and one of them is using Iridium, they both need to be connected to Iridium unless they are connected to the same LAN[1]. This is also true for the Run Time Infrastructure (RTI) execution (exec), that is the central server component for the HLA federation. For federates to be able to join the federation execution, they need to be able to locate the RTI exec which implies that the RTI exec must be reachable through Iridium (i.e. its endpoint has to be an Iridium IP address). As a result, the RTI exec listens for joining federates at a non-Internet-routable IP address, which forces all federates to be connected to Iridium.

There is, however, an exception to the statement above. On the LAN at NJHQ we were in control of all routing tables on the computers. In this way, we could configure Iridium messages from other computers on the LAN to be routed through the Iridium gateway computer. As a result, federates on the other computers did not need to be connected to Iridium. For this solution to work, however, the RTI exec had to be set up as a mediator between all federates, preventing them to connect to each other directly (using the centralized event channel option of DMSO RTI 1.3 NG). It is also important to note that if our RTI execution was able to listen on more than one network interface (i.e. both Iridium and Internet), the described problem might have been avoided.

Due to scalability and the obstacles discussed above we were forced to develop a Java Remote Method Invocation (RMI) tunneling solution to bridge federates over Iridium. By using this

---

[1] This limitation is related to the RTI implementation (DMSO RTI 1.3 NG-V6).

tunnel we were able to bypass the problems described above. Due to time constraints we did not try alternative solutions in order to connect federates using Internet for this experiment.

6 SOFTWARE CONFIGURATION

The SE consisted of five main components of which the first four in the list below were located at NJHQ (see Figure 6.1):

1. MÄK VR-Forces (4), a COTS Computer Generated Forces (CGF) tool. The CGF was used to simulate the behaviour of the entities participating in the scenario.
2. An FFI developed picture compilation simulation, SensorSim. SensorSim simulated communication and sensor detection, tracking and classification of the sensors connected to the PCNs, producing tracks and classification messages.
3. A 3D stealth viewer (MÄK Stealth Viewer) (4). MÄK Stealth was used to monitor the movement of simulated entities by visualizing them in a 3D environment.
4. SImulation RUn-time sUPervisor (SIRUP) (2), a Federation Management tool for controlling the start-up and execution of a geographically dispersed SE.
5. Reporter forwarding track messages to the ad hoc picture compilation demonstrator and feeding live GPS positions of HNoMS Otra and HNoMS Vidar into the SE.

All simulation components communicated using the HLA RTI, a leading standard for building networked simulations. A centralized RTI component, called RTI exec, executed on the gateway computer at NJHQ, enabling components to interconnect.

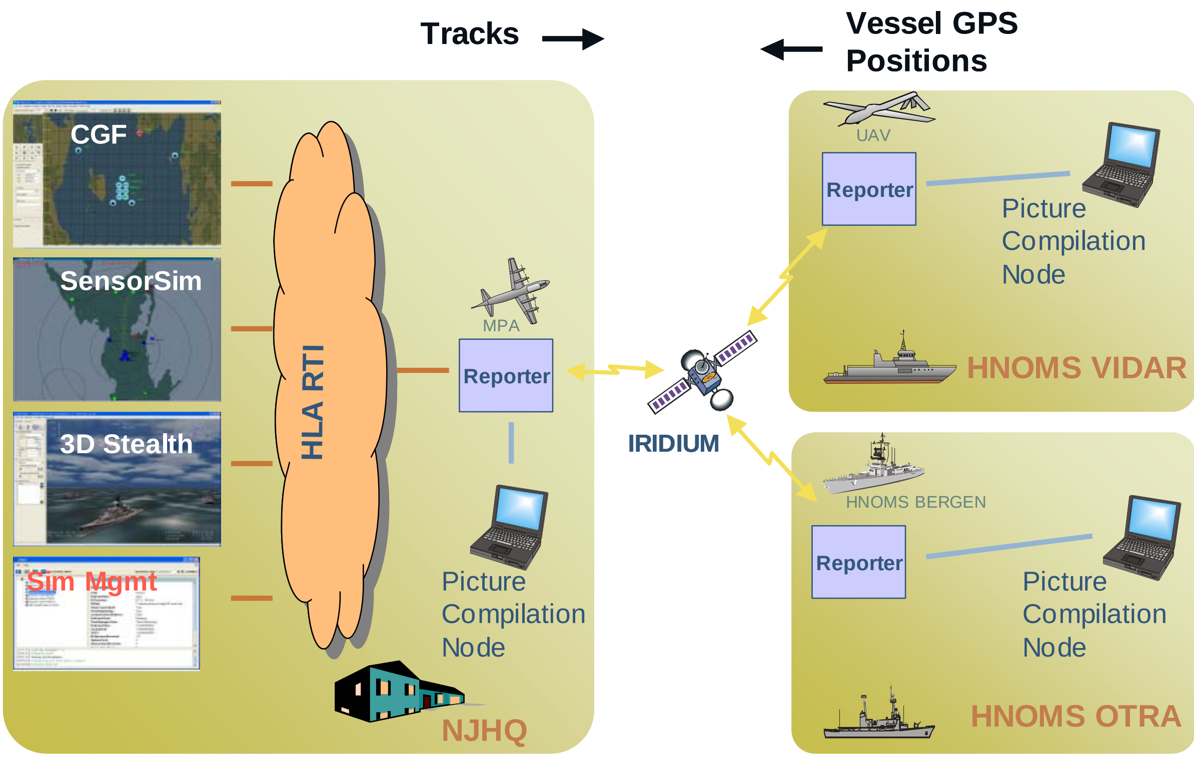

Figure 6.1 The set-up of the SE

Most of the Blue Game 2004 scenario was implemented in MÄK VR-Forces. This application was located at NJHQ, and was our main component with respect to simulation of the scenario. VR-Forces enabled monitoring (and editing) of the scenario during simulation execution, including planned routes and actions.

The main component subscribing to CGF updates was SensorSim[2]. SensorSim's primary task was to simulate the sensors on the UAV, MPA, and HNoMS Bergen. When position updates and classifications were received, updated track messages were sent using HLA interactions. Reporters connected to each of the PCNs received the tracks. In addition, the same reporters received GPS position updates from HNoMS Otra and HNoMS Vidar. The GPS updates were transformed into HLA messages, and picked up by SensorSim, VR-Forces and the stealth viewer. The result was a mixed simulation environment of live and constructive entities.

The middleware used during the Blue Game 2004 experiment was a mix of HLA and Java RMI. An initial solution based solely on HLA proved not to be scaleable. Most of the obstacles described in chapter 4 was related to the Reporter component. The next chapter will explain this component in more detail, and clarify the initial solution together with the problems experienced and how they were solved.

6.1 Data exchange between the SE and the technology demonstrator

The Reporter component can be considered as a bridge connecting two different systems: the SE and the picture compilation technology demonstrator. The Reporter subscribes to HLA track messages from the SE, transforms them into Common Operation Picture (COP) tracks, and publishes the tracks to the technology demonstrator as a Jini service. The Reporter also receives the GPS positions from the picture compilation demonstrator, transforms them into HLA interactions, and publishes them to the SE.

During the Blue Game 2004 experiment, the main SE components were executing at NJHQ and connected to a LAN. The picture compilation technology demonstrator was distributed at three locations: HNoMS Otra, HNoMS Vidar and NJHQ, each executing a PCN. One Reporter was located at each location feeding the PCN with track messages. During previous lab set-ups the PCNs were installed on the same LAN as the SE. The main challenge in Blue Game was to move the Reporter from the LAN to locations reachable only by satellite communication. This meant that they would have to join and communicate with the RTI using satellite communication.

To begin with, we tried to use the original software configuration designed for LAN. The only extension was to use Globalstar to connect the Reporters. The reason we chose Globalstar as our primary carrier, was due to its reported bandwidth and latency compared to Iridium (we measured a throughput of 7.5 kbs and a latency of 0.8 s). Unfortunately, this solution did not work very well. For unknown reasons, the components used on average 10 min to connect to the federation execution (compared to only a few seconds on a LAN), and often were not able to connect at all. We tried to monitor the network during connection, but did not reveal any

[2] The other main component subscribing for the CGF events was MÄK Stealth Viewer.

large amount of data flow. We also experienced frequent disconnections. This might have been due to network congestion.

The next approach was to use Iridium as the satellite communication carrier. Fortunately, Iridium proved to be more stable than Globalstar. Its latency was larger (2.5 s) and the data rate more variable. The data rates measured varied from 1 kbs - 20 kbs with 11 kbs on average using the built-in compression of the Iridium modem software. Infrequent disconnections and long set-up time did still occur, but at least the set-up phase was more reliable.

Although Iridium was chosen, disconnections and network congestion caused problems. The HLA RTI had serious problems accepting components that did not respond timely. The components still used 10 min on average to connect to the federation execution, and sometimes connecting was unsuccessful. The option of using HLA was finally abandoned, as it proved not scalable when all simulation components were connected to the network. The main reason for this is probably that even though HLA uses a publish-subscribe pattern in order to reduce network load, the RTI (DMSO RTI 1.3NGv6) was designed to filter data at the recipients making the satellite link to saturate.

6.1.1 Redesigning the Reporter

The solution to the problem was to redesign the Reporter. The Reporter component was split into two parts, one to execute on the LAN at NJHQ (HLA side), and one to be used onboard the vessels (PCN side) (see Figure 6.2). The two components were tightly coupled, and communicated using Java RMI. In this way, we could control what kind of messages that were being transmitted over Iridium. The Reporter HLA side was a HLA federate, subscribing for messages to be forwarded the PCNs. When a message was received at the Reporter HLA side, the message was encapsulated as a remote method call to the Reporter PCN side in accordance with the COP data model.

The computers executing the PCN Reporters acted as gateways between Iridium and the PCN LANs and thus had two different IP addresses. This caused an unforeseen problem. The Reporters PCN side was bound to the LAN address, and failed to connect to the Reporter HLA side (on Iridium). This problem was solved by explicitly binding the RMI client on the Reporter PCN side to the Iridium IP address. Because the Iridium address changed whenever the gateway computer connected to Iridium, the Iridium IP address was supplied as a command line option. Furthermore, since the Reporters also provided a Jini service on the PCN LAN, the components had to be carefully configured in the connection setup phase.

Since the Reporter HLA side was acting as a federate, all RTI communication was restricted to the LAN, and any problems belonging to long set-up times or congestion were removed. Moreover, the RTI would not be exposed to the Iridium connection, as the Reporter HLA side encapsulated this connection. This meant that even if the connection broke and we had to redial Iridium manually, the SE would be physically uninterrupted. The reporter PCN side would also listen for such disconnections, and if so happened seamlessly try to reestablish the connection to the Reporter HLA side. The Reporter components were now specially designed to cope with unstable networks, and this functionality was transparent to the simulation LAN at FOHK, as well as the ad-hoc picture compilation system. Connection set-up time between the Reporter peers was between 5-30 s.

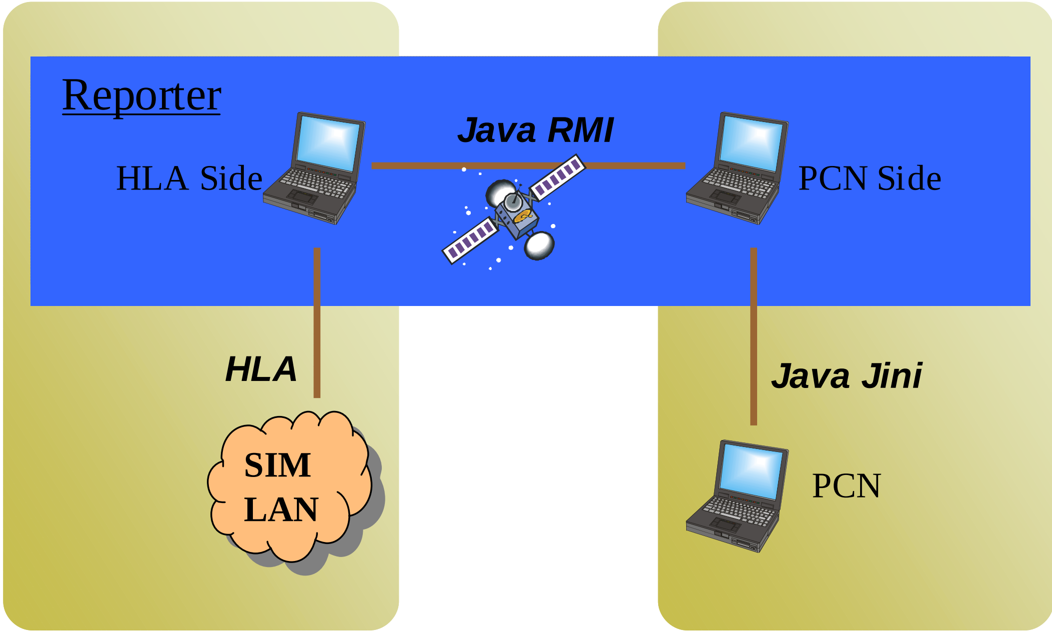

Figure 6.2 The figure shows how the Reporter was divided into two applications. HLA messages were communicated between them using a Java RMI tunnel.

6.2 Centralized RTI server

Before redesigning the Reporter we also experienced problems due to the private IP-address space used by Iridium. As default, the RTI execution is usually playing the role as a central look-up server, enabling simulation components to interconnect, and returning initial configuration data upon joining. After the components have joined, they will start to communicate directly with each other, either by sending multicast messages, or connect in a peer-to-peer fashion.

As explained in chapter 4, neither Globalstar nor Iridium support multicast. Peer-to-peer communication also proved to be problematic. The primary reason is that many of the simulation components in the SE were connected to the LAN at NJHQ, and given non-routable addresses unreachable outside the Iridium network. As a result, when the Reporter at e.g. HNoMS Otra tried to establish a connection to SensorSim, it would fail[3]. Fortunately, the DMSO RTI 1.3 NG provided a configuration that solved the problem by using the RTI exec as a centralized event channel (mediator). This means that no simulation component communicates directly, but always through the RTI exec. This will slow down the simulation, but bypass the problem of non-routable IP-addresses.

It is important to note that the solution above will only work under the condition that the RTI exec is reachable from all simulation components (and vice versa). In the case of the Blue Game 2004 experiment the only computer that met this condition was the gateway laptop

[3] The problem would have been solved if we had been able to control that the LAN components should initiate the connections. We did however not succeed in finding such a solution. Another solution could be to use a Network Translation Server (NAT server).

between the LAN and Iridium. This was the only computer known both to Iridium and to the LAN (see Figure 5.1).

Using a centralized RTI event channel solved the problem of interconnecting federates. We were now able to run a distributed simulation, at least for a few minutes. After a while, however, Reporters connected to Iridium stopped receiving track messages. The reason for this we believe originated from the fact that the DMSO RTI did not perform sender side filtering of messages. As a result, the Iridium link got saturated by the data traffic generated by the CGF after a few minutes.

The abovementioned problem, together with frequent Iridium disconnections, forced us to alter the Reporter, as described in chapter 6.1.1. When using the redesigned Reporter, the problem of connecting peer-to-peer also disappeared. From the perspective of the RTI exec, all simulation components would then reside on the LAN, and peer-to-peer connections, as well as multicast, would work as normal. However, altering the Reporter was a secondary solution, and like the above, many different configurations were tested before this was carried into effect.

6.3 Using a different RTI implementation

An alternative way of coping with the challenges posed by the network was to use another RTI implementation. During the experiment, the DMSO RTI 1.3 NG-V6 was used as the underlying middleware platform, offering the advantage of being highly configurable and stable. But it also has a serious shortcoming by not performing sender-side filtering of messages. This means that all federates in a federation will receive all messages, regardless of whether or not they use them (probably except for the case of using routing spaces). This is unproblematic when all federates are located at the same LAN (as all Ethernet messages will be forwarded to all computers anyway). But obviously a bad solution when federates are distributed on Internet, and even worse when using satellite communication.

Initially we planned to only use HLA to interconnect the SE components on a mixture of LAN, Internet and satellite communication. As mentioned earlier, our former experience of creating SEs were constrained to a single LAN, and we therefore found such an expansion to be challenging. The idea was that if we were able to geographically distribute the SE without having to modify any of the simulation components (only configuring the RTI), we would gain important knowledge. For that reason, the solution of altering the Reporter was considered more or less as a last way out. We knew, however, that this solution would solve many of the challenges described early on, but we also knew it to be a tailor-made solution only valid for Blue Game 2004.

As described in chapter 6, the initial design was not successful. We were not able to establish a satisfying solution solely using HLA. It is important to realize, however, that this shortcoming is not related to the HLA but to the RTI implementation[4]. Several different RTI vendors exist,

---

[4] HLA is not concerned with *how* messages are distributed between federates; only what kind of messages that *can* be distributed, together with corresponding rules and semantics. How federates interconnect is left to the RTI implementation.

and we obtained a trial license from Pitch AB on pRTI1.3 (5). The pRTI 1.3 implements sender-side filtering. Unfortunately, initial experimentation revealed that this pRTI 1.3 was not fully compatible with VR-Forces. It also seemed that the central RTI component (CRC) was unable to act as a mediator to bypass the IP addressing obstacles. As a result of the CGF incompatibility we did not pursue this solution any further.

7 DATA LOGGING AND ANALYSIS

In order to analyze the throughput of track messages, all data were logged at the producer and at the recipients. The Reporter components and SensorSim produced local logs containing data sent and received. The logs were stored as XML files and local timestamps were added to each data item.

The main goal of the data analysis was to estimate the delay of track messages, that is, the time elapsed from a message was being sent by SensorSim until received by the Reporter PCN side. In order to be able to compare local timestamps some common time reference were needed. Due to time constraints the use of time synchronization services like Network Time Protocol (NTP) or the use of external clocks on the internet through dedicated software were not implemented. Instead the time differences between the computer clocks were measured before analyzing the data and compensated by clock-drift. This strategy was however unfortunate as estimating the clock-drift on the different computers as a linear function was unsuccessful. As a result we were only able to estimate the order of magnitude of the track message time delay.

The analysis of the data indicates that the end-to-end latency between the ad hoc picture compilation system and the synthetic environment was in the range of 15-45 s. The update interval was 30 s for track messages and 10 s for the GPS reports. Table 7.1 shows the average number of tracks reported in each message (corresponding to one HLA interaction). Note that this number does not correspond to the number of tracks held by the sensors, because some messages may only contain a classification update.

Bergen	UAV	MPA
13	15	4

Table 7.1 Average number of tracks reported in a message

8 SUMMARY

This report has documented a distributed SE developed for an experiment conducted during Blue Game 04, with emphasis on technical issues. The aim of the SE was to stimulate an ad hoc picture compilation technology demonstrator with track data originating from a realistic scenario.

The main challenged faced during the development of the SE was related to the use of satellite communications in order to network with naval vessels. Several different solutions were examined in order to overcome these obstacles, regarding both technical infrastructure and software.

The initial design was to use a HLA to interconnect the simulation components. After several different configurations, we ended up with a solution diverging partly from this requirement. While most of our components used HLA and the DMSO RTI 1.3 NG, a Java RMI tunneling component was created for transferring messages over Iridium satellite communication. The developed tunneling solution was tailor made with respect to the data to be transferred and was fault-tolerant regarding satellite communication disconnections.

Figur 8.1 Screen dump from 3D stealth viewer

The tunneling solution did not experience noticeable problems during all experiment runs. In order to analyze the throughput of track messages, all data were logged at the producer and recipients. The analysis of the data indicates that the end-to-end latency between the ad hoc picture compilation system and the synthetic environment was in the range of 15-45 s. The update interval was 30 s for track messages and 10 s for the GPS reports.

References

(1) Rasmussen R et. Al (2004): Exploratory experiment ”Ad hoc organization of picture compilation conducted during Blue game 2004: Evaluation report, FFI/RAPPORT-2004/01940, Forsvarets forskningsinstitutt.

(2) Gustavsen R M (2004): Simulation Runtime Supervisor (SIRUP) - a federation manager, FFI/NOTAT-2004/01327, Forsvarets forskningsinstitutt.

(3) http://www.terrex.com.

(4) http://www.mak.com.

(5) http://www.pitch.se.

(6) Mevassvik O M (1996): SensorSim - simuleringsprogram for sjøovervåkning, FFI/NOTAT-96/05092, Forsvarets forskningsinstitutt.

(7) Hansen B J, Sanden H (2000): SensorSim - brukerveiledning, FFI/NOTAT-2000/06434.

(8) Mevassvik O M, Hansen B J, Sanden H (2003): Utvidelse av SensorSim, FFI/NOTAT-2000/00006, Forsvarets forskningsinstitutt.