



NLR-TP-2000-412

**Divide and control:
Making distributed real-time simulations work**

E. Kessler, A.A. ten Dam, R.P. van Sterkenburg and

E. van de Sluis



NLR-TP-2000-412

**Divide and control:
Making distributed real-time simulations work**

E. Kessler, A.A. ten Dam, R.P. van Sterkenburg and
E. van de Sluis

This report is based on an article to be published in the Proceedings of the 6th International Workshop on Simulation for European Space Programmes, SESP'2000, by ESTEC, Noordwijk, The Netherlands, 10-12 October 2000.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Division: Information and Communication Technology
Issued: August 2000
Classification of title: Unclassified



Abstract

Traditionally real-time simulators are single-site facilities. Nowadays, the use of simulators for engineering and training requires increased capabilities to simulate total system behaviour by including accurate rendering of the constituent subsystems. For complex systems, like those encountered in the space domain, various organisations have good simulators for their own subsystems. To save costs and reduce time-to-market it is advantageous to combine these proprietary simulators of the various companies into a single facility, a federation. For these situations NLR has developed a tool, SmartFED (Scenario MAnager for Real-Time FEderation Directing), which facilitates that proprietary simulators at various geographic locations participate in a joint simulation. The result is a single simulation operating a shared scenario. The High Level Architecture (HLA) based SmartFED accomplishes this by performing three main tasks: control, monitor and scenario definition and execution. Each task is implemented in an independent software component. This approach to distributed simulation allows the human experiment leader to control the execution of the simulation federation while each organisation retains full control over its proprietary simulator. The general applicability of this approach is illustrated by applying SmartFED to four unrelated geographically distributed federations. Software metrics indicate the significant costs savings and time-to-market reduction of this approach. The applicability to the space domain is addressed by providing some promising space applications including MPTE (Mission Preparation and Training Equipment).



Contents

1	Introduction	4
2	Distributed simulations	5
3	Overview of SmartFED capabilities	7
4	Federation Manager	8
5	Federation Monitor	10
6	Scenario Definition and Execution Manager	11
7	Highlights of selected SmartFED applications	11
8	SmartFED details and experience	13
9	Relevance to space applications	14
10	Concluding remarks and future work	15
11	Abbreviations	17
12	References	18

(18 pages in total)



1 Introduction

Traditionally real-time simulators are single-site facilities. Nowadays, the rapidly expanding use of simulators for engineering and training requires increased capabilities to simulate total system behaviour by including accurate rendering of the constituent subsystems. For complex systems, like those encountered in the space domain, various organisations have good simulators for their own subsystems. To save costs and reduce time-to-market it is advantageous to combine the proprietary simulators of the various companies into a single facility, a federation. For these situations NLR has developed a tool, SmartFED(Scenario MAnager for Real-Time FEderation Directing, see [1]), which facilitates that proprietary simulators at various geographic locations participate in a joint simulation. The result is a single simulation federation operating a shared scenario. The High Level Architecture (HLA, see [2]) based SmartFED accomplishes this by performing three main tasks:

- Control task: Simulation control is realised by the Federation Manager, which controls the execution state of all federates in the entire simulation;
- Monitor task: The Federation Monitor allows an operator (e.g. the experiment leader or the instructor) to monitor the entire federation on a screen from a single location;
- Scenario definition and execution task: The Scenario Definition and Execution Manager specifies which federates participate in the particular simulation and provides all federates with the appropriate simulation events.

This approach to distributed simulation allows the human experiment leader to control the execution of the simulation federation while each organisation retains full control over its proprietary simulator.

Some key features of SmartFED, which are substantiated in the following chapters, are the following:

- Generic tool: it can be used for any HLA-based simulation that has a federation object model;
- Training: it is very useful for training purposes as it allows the instructor to repeat the same scenario for a number of trainees. In combination with an assessment tool, a trainee can be guided through a training programme;
- Geographic distribution: it operates on geographically dispersed simulators;
- State-transition diagram: it guides a distributed simulation through a common state-transition diagram;
- HLA compliant: it has been developed for HLA-based simulations;
- CORBA: the Federation Manager has been ported successfully to operate in a non-HLA Common Object Request Broker Architecture (CORBA) based simulation.



The paper continues with a description of distributed real-time simulations. Chapter 3 provides a global description of SmartFED. Chapters 4 through 6 are devoted to a more detailed description of each of the SmartFED components i.e. the Federation Manager, the Federation Monitor and the Scenario Definition and Execution Manager respectively. Chapter 7 highlights some applications in which SmartFED has been deployed. Chapter 8 documents the details and experience gained, including software metrics. The applicability to the space domain is addressed in chapter 9 by discussing promising applications in MPTE (Mission Preparation and Training Equipment). Chapter 10 contains the concluding remarks. Finally the acronyms can be found in chapter 11 and the references are listed in section 12.

2 Distributed simulations

In the real-time simulator industry emphasis has for a long time been put on stand alone, real-time, high performance simulators. The operational parts of these simulators had to be placed close to each other at the same location in order to be connected through high-speed local networks. In recent years, networked simulations have emerged. A networked simulation consists of a number of simulators (or simulators parts) that are geographically dispersed, possibly far apart, and are connected through lower bandwidth wide area networks. Nevertheless, the networked simulators have to operate in a shared simulation as a seamless facility in order to allow man-in-the-loop simulation optionally complemented by hardware-in-the-loop simulation. In HLA terminology a simulator is called a federate and a collection of federates is called a federation. This terminology will be used throughout the remainder of this paper.

Distributed real-time federations require, amongst others, an architecture to support communication between simulators and a tool to control and monitor the federation, i.e. the scenario management tool. Both architecture and tool have been developed within the Simultaan [3] project. In the Simultaan project, institutes and a university collaborated to achieve a common view on the development of distributed federations using state-of-the-art technology. The objective has been to encourage re-use to improve affordability and improve time-to-market of distributed real-time simulation federations, see [4]. Such federations are applied in fields like training, engineering, research and even in the preparation of real systems operation.

Distributed federations consist of a number of independent simulators that collaborate to achieve these common goals. These simulators are usually geographically dispersed and are connected through wide area networks. A number of standardised communication protocols exist. In general these protocols are based on the broadcasting principle, which causes a significant amount of communication overhead. At present wide-area networks face difficulties

in supplying the required bandwidth. To reduce the communication overhead, new architectures have been designed that use a publish-and-subscribe mechanism. With this mechanism, data is only sent from one simulator to another if the receiving simulator has declared its interest in (i.e. is subscribed to) that type of data. Commercial Of-The-Shelf (COTS) implementations of HLA are available which work on heterogeneous networks. They offer real-time support for distributed federations.

With HLA, communication is realised through the use of the Run-Time Infrastructure (RTI). An important collection of data made available through the publish-and-subscribe mechanism, is the Management Object Model (MOM) data. The MOM provides global information about the federation (e.g. the list of participating federates) and information about the constituent federates (e.g. the federates hostname) of the federation.

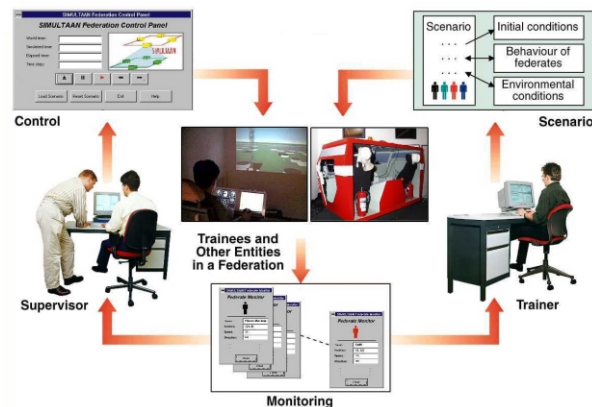


Figure 1. Simultaan federation: an example of a distributed simulation

Figure 1 shows the Simultaan federation, a distributed simulation that connects three proprietary man-in-the-loop simulators. Scenario Management plays the central role in the federation as it controls and monitors the entire distributed simulation. The combination of SmartFED and HLA allows the reuse of existing proprietary simulators on heterogeneous environments whilst making efficient use of available bandwidth. Moreover, also HLA itself is available on several platforms, which is an important asset. To operate in a federation, a federate only needs to comply with the HLA interface standard, which is a relatively simple addition to an existing simulator.

Use of HLA allows each simulator in a federation to subscribe to and publish two types of data, namely (portions of) the federate state, and interactions between federates. For Simultaan, an example of federate state data is position information of a simulated entity. Examples of interactions between federates are early collision warnings between two simulated entities.

3 Overview of SmartFED capabilities

The Federation Development and Execution Process (FEDEP) of HLA based distributed simulation is described in [5]. The SmartFED capabilities cover all of the activities in the Federation Execution phase. Additionally some support is provided for the Scenario Development phase. An important benefit of SmartFED is that it allows engineers to concentrate on project specific aspects while ensuring reuse of available components. For instance, SmartFED paves the way to incorporate existing federates in complex training scenarios.

Control over the execution of a simulation is needed for a number of reasons. It ensures co-operation between federates that participate in a joint simulation. Another reason is that the same scenario may need to be performed more than once. For example, in training simulations the environmental conditions must be identical in multiple runs to ensure that trainees practise under similar circumstances. In a distributed simulation, SmartFED interactively controls and monitors all federates that participate in a particular scenario run, and executes the scenario. This is illustrated in Figure 2.

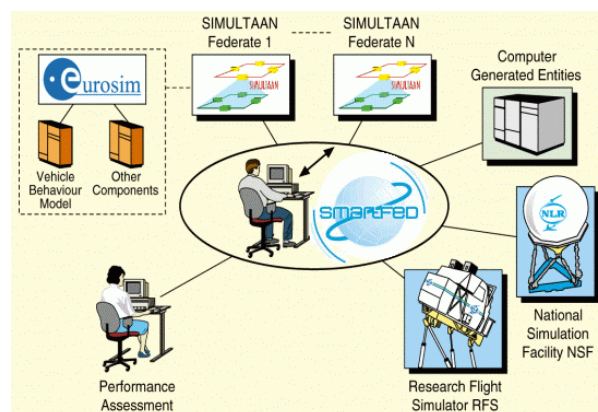


Figure 2. The central role of Scenario Management, i.e. SmartFED in Distributed Interactive Simulation (DIS)

SmartFED performs three tasks. Firstly, there is the control task. In SmartFED this is realised by the Federation Manager. The Federation Manager controls the Execution State of all federates in the entire simulation. Secondly, a human operator (e.g. the experiment leader or instructor) needs to be able to monitor the entire federation. This task is realised in SmartFED by the Federation Monitor that enables the operator to watch the entire distributed simulation on a screen in one geographic location. Different views on the federation and the participating federates are provided. When a federation is operational, the operator may want to watch the position of all entities and the interactions between them. The third task concerns definition and



execution of scenarios. In SmartFED this is realised by the Scenario Definition and Execution Manager. In a federation, the scenario may contain the initial positions of all entities and unprecedented events like the generation of a failure at a certain time in the simulation. All these tasks are made available to the users through graphical user interfaces. SmartFED has its own Graphical User Interface (GUI) to communicate with the human operator of the federation. Figure 3 shows an example combining windows for all three tasks.

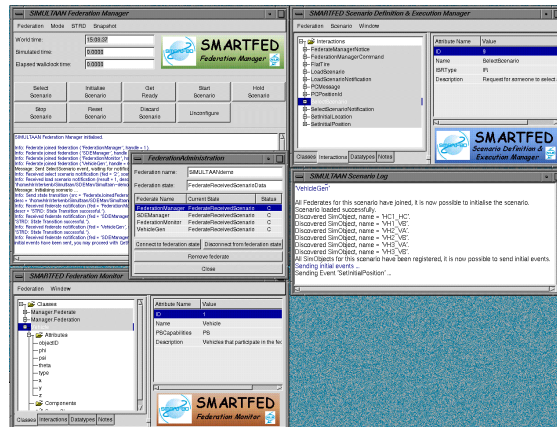


Figure 3 SmartFED Graphical User Interface

The following chapters discuss in more detail how the SmartFED components realise each of the above mentioned scenario management tasks.

4 Federation Manager

The Federation Manager provides central control over the distributed real-time simulation. The experiment leader operates the Federation Manager. The experiment leader decides when certain commands are sent to the federation. As depicted in Figure 2 the experiment leader can react on signals displayed by the Federation Monitor (which is discussed in chapter 5). Usually the same person will use the Federation Manager and the Federation Monitor, but this is not necessary.

A general state-transition diagram has been designed [6], depicted in Figure 4. Each federate must comply with this state-transition diagram. However, federates may well possess an internal state-transition diagram that differs from the one depicted in Figure 4. The main issue is that from a scenario management point of view, a federate complies with the depicted state-transition diagram. The Federation Manager sends state-transition commands to all federates. Federates reply with success or failure notifications to the Federation Manager.

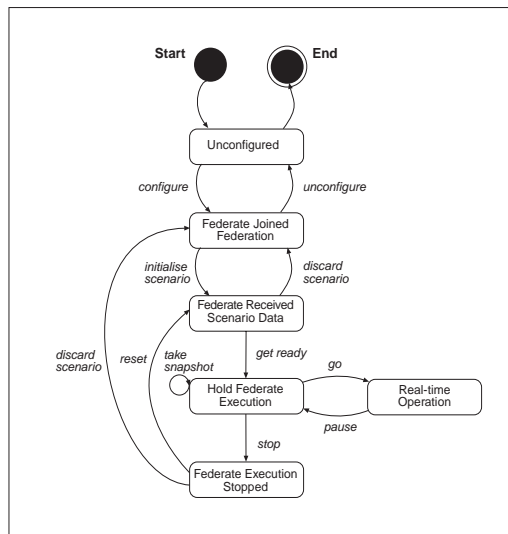


Figure 4. Federation state-transition diagram

The user of the Federation Manager (e.g. the experiment leader) decides whether a scenario must be executed or not and when a scenario execution will start and stop.

The Federation Manager subscribes to some federate data and federation data provided by HLA's MOM in order to know the states of each federate present in the federation. The Federation Manager may only send a state-transition to the federation when all federates are in the same state. State-transition commands are made available to the experiment leader by means of a graphical user interface, depicted in the top left window in Figure 3. A message window is available to notify the experiment leader of warnings or errors that occur during the federation execution, for instance when a federate loses the network connection with the rest of the federation.

The Federation Manager supports the initiation of snapshots by forwarding the snapshot invitation to the HLA RTI of the participating federates. A snapshot contains a dump of the entire internal state of a federate. Of course this is only possible if a federate is able to take a snapshot. In order to preserve the real-time nature of the simulation, snapshots can be generated only when a federate is in the 'Hold Federate Execution' state. The same holds for restoring a federate by means of a previously created snapshot.

During 'After Action Reviews', parts of the scenario can be replayed and analysed. Registering bookmarks during scenario execution can facilitate such reviews as they serve as easy to find starting points for the review. The experiment leader adds a bookmark at a point in time during the scenario run that might be of interest to the evaluation of the scenario afterwards.



5 Federation Monitor

The Federation Monitor provides the user with a view of the entire federation. This includes both a graphical overview showing the positions of the simulated entities with respect to each other and a textual view of the federation containing detailed information of the participating federates.

As mentioned earlier a federate can publish and subscribe to two types of data: federate state data and interaction data. Published data can be viewed using the Federation Monitor. As soon as its owner updates data from a federate, the Federation Monitor will receive the new values. The data attributes are viewed in textual format. An example is shown in the bottom left part of Figure 3. Additionally numerical attributes can be viewed graphically.

The data collection that contains the information of all data and interactions available in a federation is called the FOM (Federation Object Model), conform HLA terminology. A FOM is composed of the collection of available federate data and interaction data. The data and interactions subscribed to and published by a single federate is called the SOM (Simulation Object Model). The FOM is used as a basis for the implementation of the Federation Monitor. For HLA a Backus-Naur Form notation of a FOM is defined, see [2]. The Federation Monitor can read FOM files that comply with this notation. The Federation Monitor will enable the operator to browse graphically through the FOM. With this browser the operator can subscribe and unsubscribe to federate data and federate interactions whenever the operator wants.

When new data is published by a federate, it will appear as an icon in the Federation Monitor using information contained in the FOM. Detailed information on federate data and its attributes will be displayed. The Federation Monitor will subscribe to federate interactions and display those incoming events that the user is interested in. A separate interaction view provides the user with an overview of all interactions that have occurred during the federation execution. For each interaction the detailed information on its parameters are available.

Since in principle, the Federation Monitor can subscribe to all data that are made available in the federation, network congestion may occur when a user actually does so. The level of accuracy of the monitoring depends on the network load and may decrease as the number of monitored objects increases, see also [7]. It is expected that this part of SmartFED will be able to exploit the increased bandwidth and reduced latency capabilities of the upcoming wide area networks.



6 Scenario Definition and Execution Manager

The Scenario Definition and Execution Manager obviously has two main tasks: scenario definition and scenario execution. The Scenario Definition component enables the user (experiment leader, operator or instructor) to specify a scenario. A scenario is defined for a particular federation. A scenario consists of the following parts:

- federation composition: defines the federation name and which federates in the federation participate in the specific scenario;
- environmental conditions definition: the geographical environment in which the federation is operating (e.g. European airspace) and the meteorological conditions (e.g. fog);
- initial condition definition for each federate: the initial values of the federates data attributes (e.g. a vehicles position, speed);
- stimuli definition during scenario execution: which events must occur at what time during the scenario (e.g. failure at $t=10:30:00$).

The Scenario Execution component reads the predefined scenario and sends the initial conditions to the federation when Federation Manager generates the 'initialise scenario' command (see Figure 4). During the 'Real-time Operation' state (see Figure 4) the Scenario Execution component will send events to the federation at times specified in the scenario. The top right window in Figure 3 shows an example of the GUI.

The implementation of the Scenario Definition and Execution Manager is based on the implementation of the Federation Monitor. While the Federation Monitor allows the user only to watch the federate data and interactions, Scenario Definition allows the user to set the values of federate data available in the FOM and generate interactions during the scenario definition phase.

7 Highlights of selected SmartFED applications

SmartFED's general applicability and re-use potential are demonstrated by its successful deployment in four unrelated distributed real-time simulation federations. In the Simultaan project three different proprietary simulators have been coupled into a single federation, as shown in Figure 1. During the simulation each proprietary simulator remained located at its owners site. This distributed real-time simulation has been controlled from a single site.



The second distributed federation concerned a classified distributed simulator involving two proprietary simulators at different geographic sites controlled from NLR as the third site. Also in this case SmartFED has been reused successfully.

In the third distributed simulation federation only the Federation Manager has been used for an airport for A-SMGCS (Advanced Surface Movement Guidance and Control System) simulation. The SMGCS Airport Movement Simulator (SAMS) used 17 non-HLA compliant servers located in two sites in two countries. In this simulation CORBA was used for the communication between the servers. SmartFED successfully centralised execution control of the simulator by guiding all servers through the state-transition diagram and provided services to synchronise the time between all servers involved. It also supported the simulator composition at the start of each simulation. The Federation Manager saved a lot of time during actual simulations and by automating repeating tasks prevented human error. More information on the SAMS project can be found in [8].

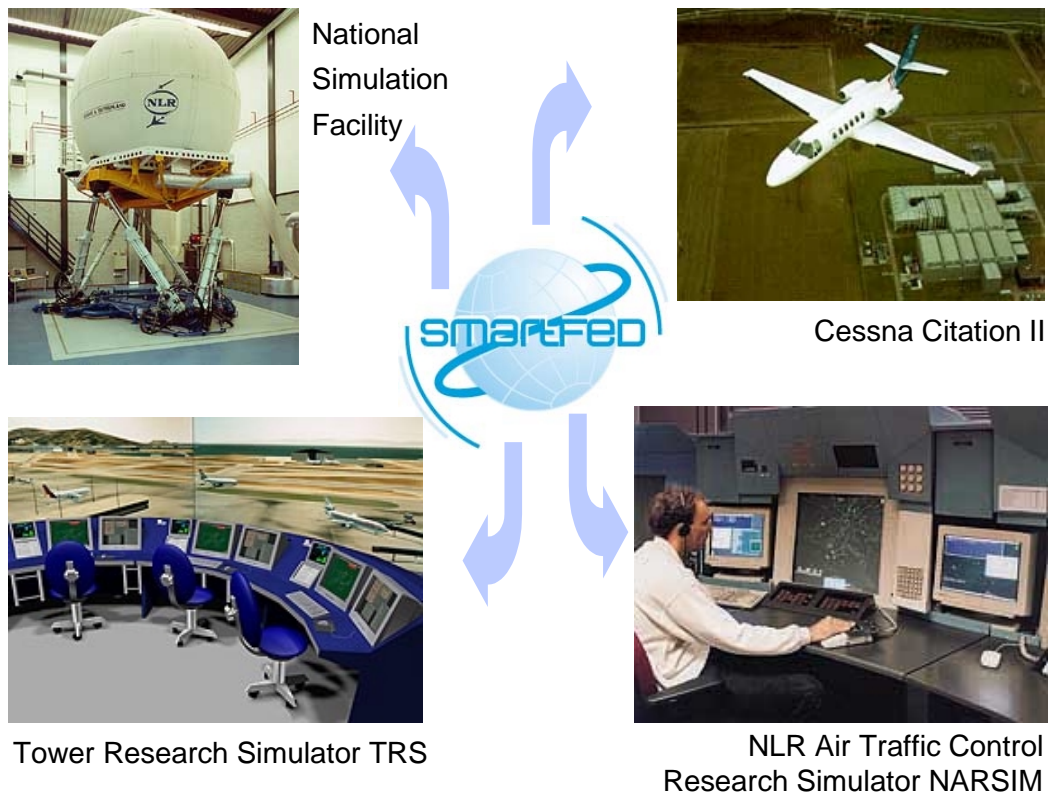


Figure 5 DELTA concept: unifying some NLR real-time simulation facilities



Figure 5 depicts the DELTA concept: the unification of part of NLR's real-time facilities in a joint simulation. As the central aerospace research and development organisation in the Netherlands, NLR operates a number of advanced facilities, among which are the Full Flight Simulator NSF, the Cessna Citation II aircraft, the NLR Air Traffic Control (ATC) Research Simulator (NARSIM), and the Tower Research Simulator (TRS). Use of SmartFED will effectively result in a DELTA federation proper. The DELTA concept will allow a user to customise a simulation to its specific needs by combining any combination of these facilities into a single, seamless federation. More information on this federation can be found in [1]. More information on each of NLR's participating facilities is provided in [9] and [10].

8 SmartFED details and experience

Based on actual experience obtained in the four federations introduced above, the following description details the effectiveness of SmartFED in controlling distributed real-time simulations, consisting of HLA compliant simulators. In the Simultaan project none of the three simulators used were specifically developed for use in a distributed simulation. The generic state-transition diagram used allowed all three proprietary simulators to comply with it. For simulation control the communication needs of SmartFED's Federation Manager were modest. The additional load on the standard ISDN (Integrated Services Digital Network) lines used was negligible. The HLA interface description of SmartFED itself consisted of around only 100 lines of non-comment RTI (Run Time Interface) definition.

The Federation Manager has a size of 9 Kloc (thousand lines of non-comment code) of C++. Approximately half of this size is related to the state-transition diagram functions provided to the participating simulators. Figure 4 depicts the state-diagram used. Although the state-transition diagram has proven to be quite generic, it might need modification when re-using SmartFED for other applications based on different state-transition diagrams. To re-use SmartFED in the DELTA federation, an extension is needed to support some specific communication protocols used in the aerospace field. The generic state-transition diagram functions or the Federation Manager would not require any significant changes. Adapting the Federation Manager for the re-use in the classified simulation took only a couple of working days. In the SAMS federation, the modification of the SmartFED Federation Manager including its CORBA extension to accommodate the non-HLA compliant servers, took less than 10 working days to complete, making it a very cost-effective solution. The additional real-time load of SmartFED on the dedicated ISDN communication lines was negligible.



In order to promote re-use, not only for simulator components, but also within SmartFED, the Federation Monitor is based on a few minor extensions to the Scenario Definition and Execution Manager. The Federation Monitor can display all standard data types including composite types and arrays. Consequently hardly any modifications were needed for the various distributed simulations. Also for the classified distributed simulator involving two proprietary simulators controlled from NLR as the third site, SmartFED was re-used successfully. To obtain a first working demonstrator less than 5 working days had to be spent. Most of these working days have been spent on adding the XDR (External Data Representation) communication protocol needed by one federates legacy models. These 5 working days included an update of the RTI to the latest version supplied by DMSO. This experience substantiates the significant gains in costs and time-to-market made possible by re-using SmartFED.

The Scenario Definition and Execution Manager has a size of 14 KLoC in C++, making it somewhat larger than the Federation Manager. In special situations it is required to run a simulation without allowing any modifications to the scenario. This is useful when an identical simulation has to be reproduced several times. In such cases the Scenario Definition and Execution Manager will not be used during simulation execution. In the case of the classified simulation this feature has been successfully exploited. For all three SmartFED components, the Graphical User Interface (GUI, see Figure 3) is based on the open source GTK+ toolkit. This package preserves SmartFED's platform independence acquired by choosing HLA. The experience with the various platforms used in the four federations in which SmartFED has been deployed confirms its support for heterogeneous networks.

9 Relevance to space applications

The description of SmartFED's potential deployment in the space domain will be based on a large multi-site ESA simulation project in which NLR is involved.

A very cursory description of Mission Preparation and Training Equipment (MPTE) follows. More information on MPTE is provided by e.g. [11]. MPTE provides a real-time simulation of the European Robotic Arm (ERA) which will be deployed on the International Space Station (ISS). The simulator will be used for a variety of purposes including mission preparation, training of ERA operators, on-line mission support and mission evaluation. MPTE consists of three identical real-time simulators, located in three different geographical locations, one in the Netherlands and two in separate geographic locations in the Russian Federation. The basic configuration of the three simulators is identical. Some specific interfaces and equipment



connected to MPTE are only available in one Russian site. MPTE already has personnel performing the SmartFED roles of experiment leader, instructor and trainee.

Combining SmartFED with MPTE would allow to have an instructor based in one site, with the trainee based in another site, thus saving significant travel costs and time. Also, a mission prepared in one site can be executed from another site involving additional experts and expertise. Mission evaluation will benefit as a mission can be replayed on any of the three sites. The SmartFED feature to allow the exact repetition of any scenario is very useful for these training purposes.

MPTE intends to simulate the European Robotic Arm (ERA). However in certain situations it might be profitable to add realistic simulations of other system(s) of the International Space Station. SmartFED can be used to realise such a distributed federation, with the only requirements being a HLA compliant interface description for each simulated ISS (sub) system added to envisaged federation. The required data link capacity will be mainly determined by the exchange of information between the participating simulators and the needs of the federation monitor. This approach maximises the re-use of existing simulators and models and consequently minimises time-to-market and costs for such combined simulations.

Deploying SmartFED for MPTE allows realising even more new options. In Houston the International Space Station Training Centre is located. A federation of this facility with the MPTE sites would allow to incorporate active payload modules. Again the advantage of a SmartFED based approach is the re-use of existing models and simulators to fulfil specific simulation requirements at low cost and at short notice.

10 Concluding remarks and future work

The user demand for more accurate real-time simulations of complex real-time systems at reduced costs and reduced time-to-market can be met by incorporating existing real-time simulators into a distributed simulation. Often these simulators are geographically dispersed and access to source code is restricted due to their proprietary nature or for security reasons. SmartFED paves the way to incorporate these existing federates in complex real-time distributed simulations.

SmartFED is implemented for application in a distributed simulation that communicates through HLA. HLA has proven to be well suited as communication architecture for controlled real-time distributed simulations. For a simulator to be HLA compliant, it only needs to satisfy the



interface requirements as specified by HLA. This means that equipping a simulator with a HLA interface is sufficient for the simulator to participate in a distributed real-time simulation that is controlled by SMARTFED. An asset of HLA is the availability of COTS products resulting in HLA availability on a number of platforms. This allows SmartFED to interact with federates in a heterogeneous network. SmartFED also supports XDR encoded data as well as CORBA based simulations adding to its general nature.

Based on the MPTE example, the applicability of this type of technology for space applications has been shown. It will improve the affordability of distributed simulations and reduce the time-to-market, i.e. SmartFED can make distributed real-time simulations work.

Future research will address extensions like a faster than real-time option to allow a user to execute a scenario as fast as possible. This option can be profitable in the pre-processing phase and the post-processing phase. The fast-time option can also be used during replay to skip less important parts of a simulation. Other work is being done to extend the state-transition diagram used. More information on NLR's approach to generic simulation tools can be found in [12].



11 Abbreviations

A-SMGCS	Advanced Surface Movement Guidance and Control System
ATC	Air Traffic Control
CORBA	Common Object Request Broker Architecture
COTS	Commercial Of-The-Shelf
DIS	Distributed Interactive Simulation
ERA	European Robotic Arm
FEDEP	Federation Development and Execution Process
FOM	Federation Object Model
GUI	Graphical User Interface
HLA	High Level Architecture
ISDN	Integrated Services Digital Network
ISS	International Space Station
KloC	thousand lines of (non-comment) Code
MOM	Management Object Model
MPTE	Mission Preparation and Training Equipment
NARSIM	NLR Air Traffic Control (ATC) Research Simulator
NLR	National Aerospace Laboratory NLR
NSF	National Simulation Facility
RTI	Run-time Infrastructure
SAMS	SMGCS Airport Movement Simulator
SmartFED	Scenario MAnager for Real-Time FEderation Directing
SMGCS	Surface Movement Guidance and Control System
SOM	Simulation Object Model
TRS	Tower Research Simulator
XDR	External Data Representation

12 References

- [1] R. P. van Sterkenburg, A.A. ten Dam, 1999, The scenario management tool SmartTFED for real-time interactive high performance networked simulation, NLR Technical Report NLR-TP-98577 or proceedings of HPCN Europe '99 Conference
- [2] HLA web-site: <http://www.dmsomil.com> Click on "Enterprise", next on "High Level Architecture"
- [3] Simultaan web-site: <http://www.nlr.nl/public/hosted-sites/simultaan>
- [4] J. Kos, A.A. ten Dam, G.W. Pruis and W.J. Vankan, 1998, Efficient Harmonisation of Simulation Competence in a CACE Working Environment, Proceedings of the 5th International Workshop on Simulation for European Space Programmes (SESP'98), ESTEC, The Netherlands, NLR Technical Report NLR-TP-98417
- [5] K. Hunt, J. Dahmann, R. Lutz, J. Sheehan, Planning for the evolution of automated tools in HLA, <http://www.dmsomil.com>
- [6] R.P. van Sterkenburg, 1998, Simultaan State-Transition Diagram for Federations, NLR Technical Report NLR-TP-99131, Amsterdam, The Netherlands.
- [7] D. Prochnow, E.H. Page and B. Youmans, 1998, Development of a Federation Management Tool: Implications for HLA, Simulator Interoperability Workshop (SIW) Spring 98.
- [8] H. Hesselink et al, SAMS, Final report for publication, NLR SAMS C/NLR/00/06, Amsterdam, July 2000
- [9] A.A. ten Dam, P. Schrap and W. Brouwer, 1994, Programme and Real-time Operations Simulation Support Tool PROSIM: The Simulation Program of the Dutch National Simulation Facility NSF, Proceedings of the 3rd International Workshop on Simulation for European Space Programmes (SESP'94), ESTEC, The Netherlands, NLR Technical Report NLR-TP-94524
- [10] NLR web-site: <http://www.nlr.nl/public/facilities>
- [11] Mission preparation and training facility for the European Robotics Arm (ERA), Z. Pronk, M. Schoonmade, 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Noordwijk, The Netherlands, 1-3 June 1999
- [12] W.F. Lammen, A.A. ten Dam, W.W.M. Heesbeen, R. Franco, MOSAIC: Model-Oriented Software Automatic Interface Converter, 6th International Workshop on Simulation for European Space Programmes (SESP2000), ESTEC, The Netherlands