



NLR-TP-2002-524

## **Challenges for Distributed Exercise Management: The SmartFED Approach**

M.F.R. Keuning, E. van de Sluis and A.A. ten Dam



NLR-TP-2002-524

## Challenges for Distributed Exercise Management: The SmartFED Approach

M.F.R. Keuning, E. van de Sluis and A.A. ten Dam

This report is based on a presentation held on the Modelling & Simulation Group Conference on Future Modelling and Simulation Challenges, Breda, The Netherlands, 12-14 November 2001.

This report may be cited on condition that full credit is given to NLR and the authors.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	I.1.D.2
Owner:	National Aerospace Laboratory NLR
Division:	Information and Communication Technology
Distribution:	Unlimited
Classification title:	Unclassified
	July 2002



## **Summary**

Distributed simulation requires a novel approach to exercise management and Verification, Validation and Accreditation (VV&A). With the introduction of (geographically) distributed simulations, exercise management consists of managing a multitude of simulators in a common scenario. This imposes new challenges with respect to managing the distributed responsibilities of the simulation. As with exercise management, distributed simulations impose new challenges on VV&A with respect to distributed responsibilities. NLR's exercise management tool SmartFED (Scenario Manager for Real-Time Federation Directing) is designed to meet these new challenges. This paper provides insight into SmartFED's concepts and practical experiences in the field of distributed real-time (training) simulations.



## Acronyms

ATM	Air Traffic Management
CORBA	Common Object Request Broker Architecture
DMSO	Defence Modelling and Simulation Office.
FEDEP	Federation Development and Execution Process.
FedMan	Federation Manager.
FedMon	Federation Monitor.
FOM	Federation Object Model.
GUI	Graphical User Interface
HLA	High Level Architecture.
RTI	Run-Time Infrastructure.
SEDEP	Synthetic Environment Development and Exploitation Process.
SDEMan	Scenario Definition and Execution Manager
SmartFED	Scenario Manager for Real-Time Federation Directing.
SOM	Simulation Object Model.
STD	State-Transition Diagram.
VV&A	Verification, Validation and Accreditation



## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Distributed Real-Time Simulation</b>	<b>6</b>
<b>3</b>	<b>Exercise Management: The Concepts</b>	<b>7</b>
<b>4</b>	<b>SmartFED Supported Distributed Exercise Management</b>	<b>8</b>
<b>5</b>	<b>Federation Manager</b>	<b>10</b>
<b>6</b>	<b>Federation Monitor</b>	<b>12</b>
<b>7</b>	<b>Scenario Definition and Execution Manager</b>	<b>13</b>
<b>8</b>	<b>VV&amp;A Support</b>	<b>15</b>
<b>9</b>	<b>FEDEP Support</b>	<b>16</b>
<b>10</b>	<b>Concluding Remarks and Future Work</b>	<b>17</b>
<b>11</b>	<b>References</b>	<b>18</b>

(18 pages in total)



## 1 Introduction

There is a growing interest in geographically distributed training/exercising using distributed simulation. Recently, applications have been published in the military [1], space [2] and civil aerospace [3] domain. Among the reported advantages are the (new) possibility to perform team training, the possibility to include real entities in the simulation, and cost reduction by saving on travel and subsistence.

Advances in research and technology create more and more new opportunities to make cost-effective use of distributed simulations. Standardisation efforts have resulted in novel intercommunication architectures and standardised processes for distributed simulation development. To take full benefit of distributed simulation in defence application areas like training, rehearsal, planning, acquisition and technology development, there are still challenges to be conquered.

Key challenges for distributed simulations include exercise management and verification, validation and accreditation. Distributed simulation requires a novel approach to exercise management. Traditionally, exercise management of single-site simulations consists of managing a single simulator. With the introduction of (geographically) distributed simulations, exercise management consists of managing a multitude of simulators. Also a novel approach is required for verification, validation and accreditation of distributed simulations. In practise, legacy simulators are adapted to be employed in a distributed simulation with other legacy and newly developed simulators. Implementation of requirements is, like the simulators, distributed, which is a challenge for verification, validation and accreditation.

To cope with these challenges, the National Aerospace Laboratory NLR has developed a High Level Architecture (HLA) based software tool-suite, Scenario MAnager for Real-Time FEderation Directing (SmartFED; see also [4], [5]). The Defence Modelling and Simulation Office (DMSO) initiated HLA. SmartFED is a facility (i.e. tool-suite) to couple various autonomous, geographically dispersed real-time (legacy) simulators into one distributed real-time simulation. In HLA parlance, such simulators are called federates that collaborate in a federation to achieve the distributed simulation. SmartFED manages the distributed responsibilities of the federation, whereas each federate remains responsible for its own internal affairs. The same tools that are used for controlled distributed exercise management and monitoring are also used to facilitate structured, controlled and repeatable verification, validation and accreditation. As such, SmartFED supports several aspects of the HLA Federation Development and Execution Process (FEDEP) Model.

SmartFED has been successfully used as an indispensable core element in several civil and military programmes. The paper provides detailed insight into the concepts of SmartFED, the practical experiences with SmartFED in the field of distributed real-time (training) simulations, also in defence, and how these concepts and experiences translate back to resolving challenges in modelling and distributed simulations.

The remainder of this paper is organised as follows. In section 2 distributed real-time simulation and the use of SmartFED in a typical aerospace application is described briefly. Section 3 presents the concepts of exercise management. SmartFED-supported distributed exercise management is detailed in the section 4. At present the SmartFED tool-suite consists of three distinct tools: a federation manager tool, a federation monitor tool and a scenario definition and execution manager tool. These three tools are described in more detail in the sections 5, 6 and 7. Section 8 describes how SmartFED supports Verification, Validation and Accreditation of distributed simulations. Section 9 elaborates on how SmartFED supports the FEDEP process. Finally, concluding remarks and items for future work are presented in section 10.

## **2 Distributed Real-Time Simulation**

An artist's impression of a SmartFED application pursued within NLR is given in Figure 1. The application deals with a total solution concept in the area of Air-Traffic Management (ATM)-gate-to-gate. Individual players, e.g. aircraft, airport, and ATM, are supported by dedicated facilities at NLR.

To aid simulated entities to interact in the virtual world, in a similar fashion as the real players, proper exercise management is required. For this, SmartFED has been developed.

The following is essential to fully exploit the advantages of distributed simulation exercises:

- 1) A standardised intercommunication mechanism. This has been addressed, first with DIS (Distributed Interactive Simulation) from which evolved HLA (High Level Architecture) [6]. At present SmartFED is based on HLA. In HLA parlance, simulation entities are called federates that collaborate in a federation to achieve the distributed simulation.
- 2) A standardised process for federation development and execution. Besides intercommunication standardisation, HLA also brought the FEDEP process [7] to address this aspect.
- 3) Exercise management. There is no standardisation yet on this aspect, though HLA offers some useful handholds. Distributed simulation requires a novel approach to exercise management. Traditionally, exercise management of single-site simulations



consists of managing a single simulator. With the introduction of (geographically) distributed simulations, exercise management consists of managing a multitude of simulators. This imposes new challenges with respect to managing the distributed responsibilities of the simulation.



Figure 1: SmartFED in an ATM gate-to-gate federation concept.

SmartFED has been successfully used as an indispensable core element in several programs since its inception in 1996. A more detailed insight into the concepts of SmartFED and the practical experiences with SmartFED in the field of distributed real-time (training) simulations is given in the remainder of this paper. First however the concepts of exercise management will be discussed.

### 3 Exercise Management: The Concepts

Exercise management, for both single-site and distributed simulations, can be split into four distinct functionalities grouped into two major responsibilities:

- 1) Simulation execution state management
  - a) Monitor the execution state
  - b) Control the execution state



- 2) Simulation scenario management
  - a) Monitor the simulation objects
  - b) Control the simulation objects

Whilst both single-site and distributed simulation exercise management comprise the same functionalities, exercise management for distributed simulations is decisively more complex than for single-site simulations.

Whereas a single-site simulation usually has a well-defined execution state, the concept of Execution State of a distributed simulation can often not be defined uniquely. Depending on the simulation exercise at hand the concept of execution state can be very strictly or more loosely defined. For example, when deploying legacy single-site simulators in a distributed simulation exercise, a very strict definition of state could very well be unfeasible, so that the application of a more loosely defined execution state is necessary.

As is the case for state of execution, also scenario management of distributed simulations is more complex when compared to single-site simulations. Simulation objects in a distributed simulation can be controlled by two different concepts:

- 1) Request driven concept, where the scenario manager requests state changes of simulation objects from their controlling simulator.
- 2) Active control concept, where responsibility of simulation object attributes is transferred to the scenario manager. The scenario manager then has unrestricted direct control over those attributes.

Of course a mixture of these concepts is necessary, especially since federates may need special safety-measures, e.g. the safety of a pilot in a full motion flight simulator. The choice of which concept is used thus depends on the federates involved in the federation. Especially legacy simulators impose limitations on the amount of external influence that can be inflicted on the simulation objects under their control.

#### **4 SmartFED Supported Distributed Exercise Management**

A typical distributed exercise management situation utilising SmartFED is depicted in Figure 2. In this case two human entities are identified, which together collaborate to perform exercise management. Whereas the supervisor controls the progress of the simulation execution, the trainer controls the content of the simulation execution.

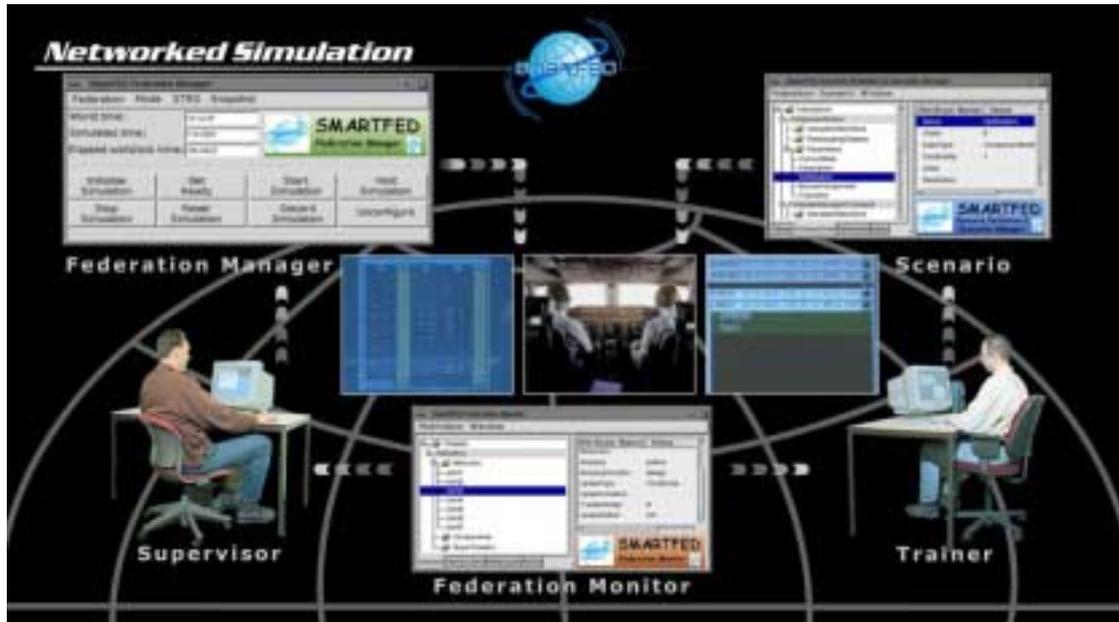


Figure 2: Typical (simplified) distributed exercise management situation involving aerospace federates.

SmartFED is a generic reusable tool-suite that provides support to the human exercise manager(s) controlling a real-time distributed simulation execution. At present the tool-suite encompasses three tools:

- 1) Federation Manager (FedMan): this tool implements support for simulation execution state management. It encompasses both monitor as well as control functionalities.
- 2) Federation Monitor (FedMon): this tool implements support to monitor simulation objects. It is remarked that more than one instantiation of FedMon is possible in a federation.
- 3) Scenario Definition and Execution Manager (SDEMan): this tool implements support to control simulation objects by means of both repeatable and interactive scenarios.

Some of the important properties of SmartFED are:

- HLA compliance. SmartFED is a tool-suite where each of the tools operates as an HLA compliant federate. Although SmartFED has been designed to be fully HLA compliant, it has also been successfully ported to use a custom intercommunication protocol based on CORBA.
- Simulation scenario management support. Within SmartFED this support has been separated into two tools (i.e. FedMon and SDEMan) to facilitate multi-site monitoring, whilst preventing conflicts due to multiple controlling entities.
- Control concept. Currently SmartFED supports the request driven concept. A future version of SmartFED will also support the active control concept.



FEDEP support. SmartFED supports the Integrate and Test Federation and the Execute Federation and Prepare Results steps (5 and 6 respectively) of the FEDEP model. The FEDEP support of SmartFED will be further discussed in section 9.

## 5 Federation Manager

The SmartFED Federation Manager (FedMan) provides central control over the distributed real-time simulation. The human supervisor (see also Figure 2) operates the Federation Manager from any one of the participating sites.

FedMan has the ability to monitor the execution state of each of the participating federates. This enables the supervisor to take informed decisions on his control strategy and to monitor the effects of his actions. FedMan supports control of federation execution state by means of a general state transition diagram (STD), which is depicted in Figure 3.

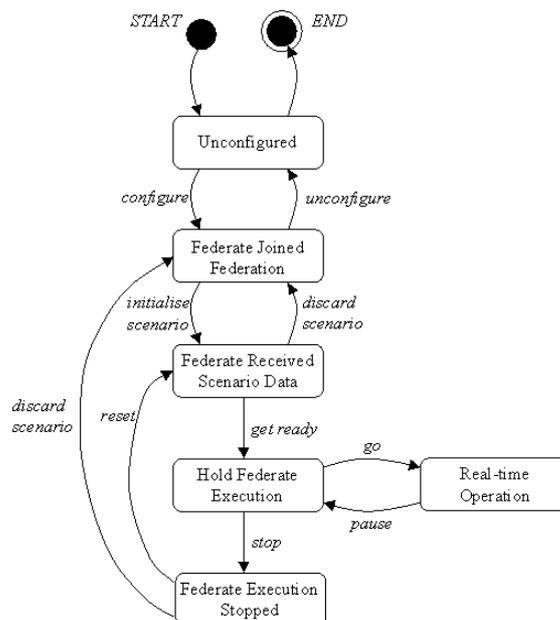


Figure 3: Federation state-transition diagram.

It is important to note that SmartFED does not impose any restrictions on a federate's internal STD. Federates may well possess an internal STD that differs from the one depicted in Figure 3. The main issue is that from an exercise management point of view, a federate complies with the depicted STD. FedMan sends state-transition commands to all federates. If applicable with regard to the selected control mode, federates reply with success or failure notifications.

During federation development it may appear that federates cannot comply with a federation STD since federates may have their own internal STD. Especially legacy simulators are made HLA compliant by building an HLA data gateway, which does not support external influence on flow of control. To deal with federations that utilise these kind of federates; FedMan supports two modes of control.

A strict control mode is available that enforces all federates to transfer into a requested state before the supervisor can forward execution to a next state. The second mode of control is a free-running mode that doesn't enforce federation wide state synchronisation. An example of a federation executing in free-running mode is depicted in Figure 4; note that different states are indicated for participating federates. The mode of control is selected at start-up of a federation and cannot be changed during federation execution.

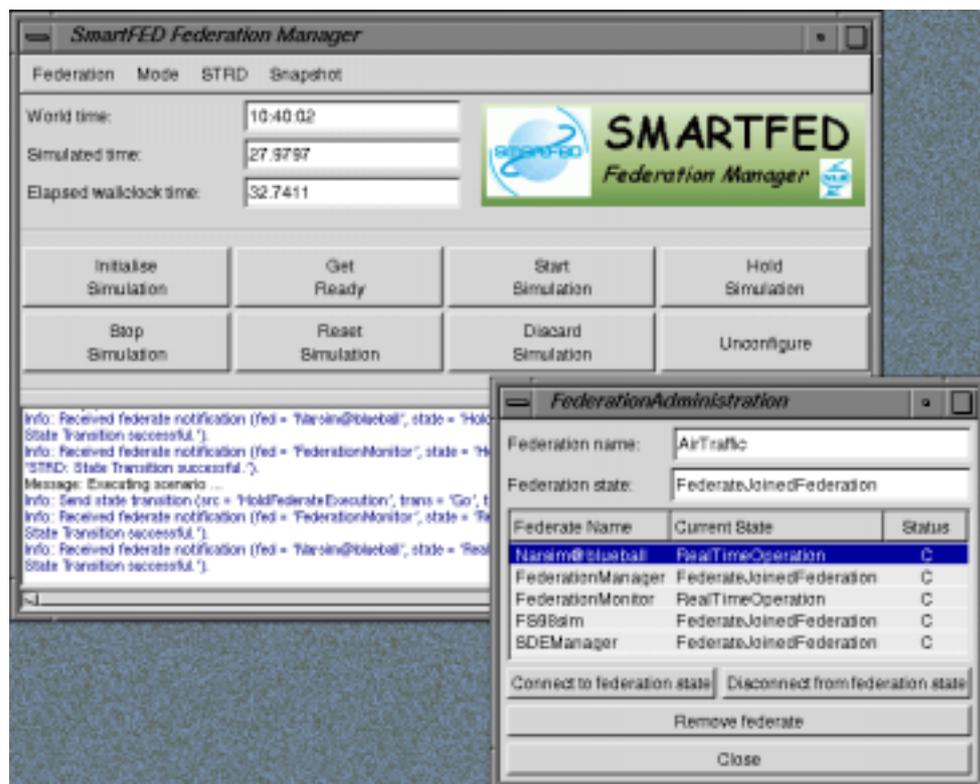


Figure 4: SmartFED Federation Manager.

For monitoring purposes FedMan provides a dedicated message window to notify the supervisor of warnings or errors that occur during the federation execution, for instance when a federate does not comply with a state transition request.

The Federation Manager supports the initiation of snapshots by sending a snapshot request to the participating federates. A snapshot usually contains a dump of the entire internal state of a federate. Of course this is only possible as far as a federate supports snapshots. 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.

## 6 Federation Monitor

The SmartFED Federation Monitor (FedMon) provides information about simulation objects within an entire federation. The supervisor and the trainer (roles identified in Figure 2) both take advantage of the FedMon monitoring facilities, though they are by no means the only possible beneficiaries of the use of FedMon. FedMon can be instantiated as many times and on any location as is deemed beneficial. An example screenshot of several of these monitoring facilities and their display formats is depicted in Figure 5.

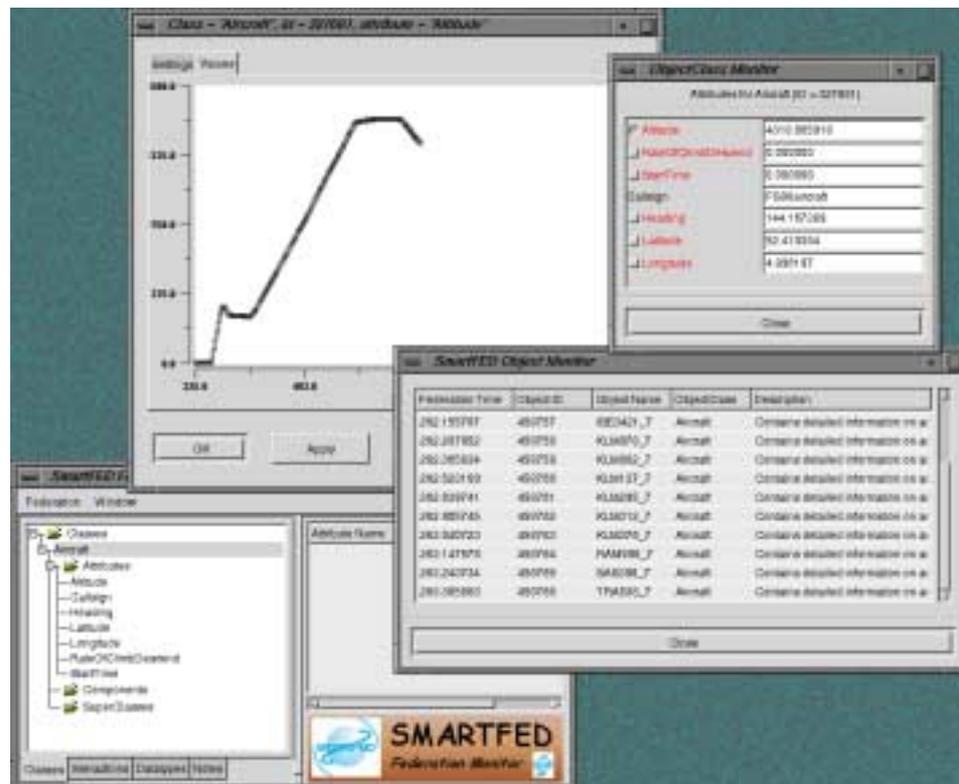


Figure 5: SmartFED Federation Monitor.

In HLA parlance, a simulation object can be either an instantiation of an object class or an instantiation of an interaction class. The difference is principally that an object has a significant lifetime, whilst an interaction takes place at a moment in time after which the interaction ceases to exist. HLA provides a standardised means to describe object and interaction classes that exist within a federation. For each federate a Simulation Object Model (SOM) must be defined. The SOM defines object and interaction classes that can be published or subscribed by that federate.



Federation wide, a Federation Object Model (FOM) must be defined. The FOM describes object and interaction classes from a federation point of view.

FedMon uses the FOM to structure access to all information available within the federation. A graphical representation of the FOM enables users to subscribe to information of their interest. FedMon provides both textual as well as graphical facilities to represent information about the federation and its simulation objects. Information monitoring can be categorised in three abstraction levels:

- 1) Simulation object/interaction class level. An overview can be displayed of all simulation objects in the federation of an indicated class.
- 2) Simulation object/interaction level. This level is supported by:
  - a) A so-called Planview oversight. This monitoring facility is aimed at providing a 2D-overview of simulation objects that possess a simulated geographic location on earth, in the air or in space.
  - b) An overview of all attribute/parameter values. The attributes/parameters are represented by the textual values.
- 3) Simulation object attribute/interaction parameter level. The user has the capability to configure views for specific attributes. For example, it is possible to view an attribute change over time.

## **7 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 (e.g. the trainer in Figure 2) to specify a scenario prior to simulation execution. A scenario consists of the following parts; an example is depicted in Figure 6:

- Federation composition: defines the federation name and defines which federates in the federation participate in the specific scenario;
- Initial condition definition for each federate: the initial values of the federates data attributes (e.g. an aircraft position, speed);
- Stimuli definition during scenario execution: which events must occur at what time during the scenario.



```

VIM - ~/DEMO.scn
File Edit Tools Syntax Buffers Window Help
// Declarations section
// =====
ScenarioID "FlightSim Scenario";

Federation "FlightSimDemo" {
  FOM "FlightSimDemo.omt";

  ParticipatingFederates {
    "FederationManager" [required];
    "SDBManager" [required];
    "FederationMonitor" [required];
    "FlightSimFederate" [required];
    "ATCFederate" [desired];
  };
};

Federate "FlightSimFederate" {
  SimObjects {
    "FlightSimAircraft" Boeing737_400;
  };
};

// Initial values and events section
// =====
InitialEvents {

  // At initialization, place Boeing737
  // on Schiphol, runway 01L

  SendEvent "SetInitialPosition" {
    ["AircraftID"] = "Boeing737_400";
    ["Latitude"] = "52.363181";
    ["Longitude"] = "4.737601";
    ["Altitude"] = "0.0";
    ["Heading"] = "10.0";
    ["Pitch"] = "0.0";
    ["Roll"] = "0.0";
  };
};

// Stimuli section
// =====
Stimuli {
  SendEvent "SetFailure" when [time=150.0] {
    ["AircraftID"] = "Boeing737_400";
    ["FailureID"] = "EngineInoperable";
    ["FailureSwitch"] = "FailureON";
  };
};

1,1 A11

```

Figure 6: SmartFED Scenario file example.

SDEMan reads the predefined scenario file and sends all initial events to the federation when the Federation Manager generates the ‘initialise scenario’ command (see Figure 4). During the ‘Real-time Operation’ state (see Figure 3) the Scenario Execution component will send events to the federation at the times specified in the scenario.

The scenario definition capability gives exercise management the possibility to (re-)play predefined training exercises. However, during exercise execution it may often be necessary to provide the trainee(s) with ad-hoc generated events. Examples are the generation of failures or the generation of additional (friend and foe) objects.

The SmartFED scenario execution manager supports this capability by allowing the exercise manager to generate in principle all events that are defined in the FOM. In this way the scenario execution manager is more or less “symmetric” to the federation monitor: the monitor allows subscribe/unsubscribe actions with respect to the FOM classes and events, while the scenario execution manager allows publish actions.

## 8 VV&A Support

The increase of interest in (distributed) simulation has also lead to an increase in Verification, Validation and Accreditation (VV&A) needs. Today, important decisions are often made that rely heavily on simulation results. "Bugs" in simulation could therefore have considerable economic or even safety effects. VV&A for distributed simulation is described in detail in [9].

There is a remarkable similarity between scenario management and VV&A. In practice, VV&A will result in the definition and execution of numerous tests. In general, test definition consists of defining test cases and test procedures. Test cases consist of sets of input stimuli and expected output responses. The test procedures are the actions to be performed to execute the test cases. During test execution, the test cases are executed and the actual results are compared with the expected ones, giving pass/fail results. From the discussion on the SmartFED SDEMan capabilities it is clear that the scenario mechanism can be used to define the input part of the test cases. The actual test results can be obtained by monitoring the data and events using SDEMan.

Although not designed to perform formal VV&A, it is observed in practice that SmartFED is often used as a useful federate and federation testing tool. Usually, three levels of testing are distinguished for a federation (see also [7]):

- Federate testing: to verify compliance of each federate with its allocated requirements (as documented in for instance the FOM).
- Integration testing: to verify a basic level of interoperability between the federates comprising a federation.
- Federation testing: the ability of the federation to interoperate to the degree necessary to achieve federation objectives is verified.

Federations can be tested using SmartFED on all levels. At the moment it being investigated whether the SmartFED capabilities should be enhanced to incorporate more testing capabilities. An example of such a capability is the possibility to automatically compare actual obtained federate/federation responses with expected ones. The scenario format as described in section 7 could be easily expanded to include this capability. However, in practice this would require for test case definition an exact description of the expected outputs, and, as illustrated in e.g. [10], the verification and validation of simulators is usually performed with respect to reality, which is most difficult to specify. Moreover, the behavior of simulations often results into graphic or mechanical effects, and not by (HLA based) object data or events that could be monitored.



## 9 FEDEP Support

The HLA Federation Execution and Execution Process (FEDEP) model [7] describes a high-level framework for the development and execution of HLA federations. The intent of the FEDEP model is to specify a set of guidelines for federation development and execution that federation developers can use to achieve the needs of their application.

The FEDEP process is depicted in Figure 7. SmartFED supports the Integrate and Test Federation (step 5), the Execute Federation and Prepare Results (step 6) and partially the Develop Federation steps (step 4) of the FEDEP model with the following capabilities:

- Federate testing is supported to validate the various federates with respect to the FOM. By performing this kind of (stand-alone) validation before the federates are integrated into the overall federation (usually by a “big bang” integration) faults can be detected and repaired at an early stage, thereby saving time and reducing costs.
- Federation integration testing is supported where the integrated federation is tested to “verify a basic level of interoperability”. Testing the state transition diagram of FedMan can easily test this basic level of interoperability.
- Validating the complete integrated federation against the FOM.
- Scenario instances (input to step 5) can be implemented by the scenario file mechanism of SDEMan as discussed in section 7.
- SmartFED provides logging files to support after action reviews of federation execution.

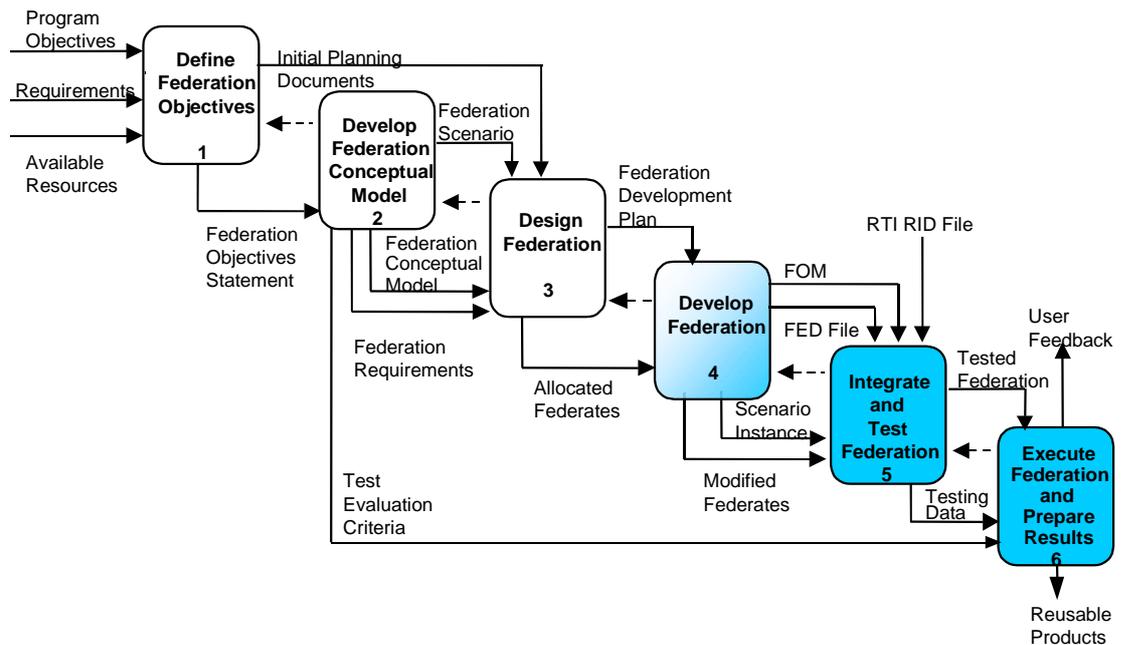


Figure 7: FEDEP's six-step model (from [7]).



## 10 Concluding Remarks and Future Work

The growing interest in utilisation of distributed simulations has led to the development of a standardised intercommunication mechanism as well as a standardised process for federation development and execution. Exercise management has not been standardised yet. This has led to the development of the Scenario MAnager for Real-Time FEderation Directing (SmartFED) tool-suite, which provides tool support for distributed exercise management in real-time. Though SmartFED was primarily designed for distributed exercise management, it can also be used as a valuable tool for VV&A. The SmartFED tool-suite is successfully used in a number of aerospace projects.

SmartFED utilises the standardised intercommunication mechanism HLA and supports the standardised FEDEP process. Several practical applications utilise SmartFED from which experiences are gathered and used for product improvement. The Generic State Transition Diagram (STD) deployed by the FedMan tool will be enhanced by the support for a user-defined federation execution STD. The current generic STD will still be available as a default instantiation of such a user-defined STD.

A limiting factor in worldwide simulation through connecting simulation facilities using for instance HLA is often the available bandwidth. The SmartFED tool-suite will be extended with a so-called Federation Timing tool (FEDTim) that can be used to perform specific measurements on data flows between federates in a federation.

The use of an automated tool to validate a federate/federation (as discussed in Section 8) may raise questions about the quality of the tool. To anticipate this, SmartFED is in the process of being qualified as verification tool in the sense of [8]. In [8], software verification tools are described as tools that cannot introduce errors, but may fail to detect them (this in contrast to development tools, that can introduce errors). SmartFED tool qualification now consists of demonstrating that “the tool complies with its Tool Operational Requirements under normal operational conditions”. Basically, this means that SmartFED is undergoing a stringent verification process, with several test federations.



## 11 References

- [1] Dr Peter Crane, Dr Winston Bennett Jr, and Dr Dutch Guckenberger: “Distributed Mission Training – Bringing Teamwork to Air Force Simulators”, in: *Military Training & Simulation News (MT&SN)*, Vol. 2, Issue 6 – November 2000, pp 58-62.
- [2] L. Argüello, J. Miró: “Distributed Interactive Simulation for Space Projects”, in: *ESA bulletin 102* – May 2000, pp 125-130.
- [3] H. Hessselink et. al.: “SAMS: Final Report for Publication (deliverable D5.5)”, SMGCS Airport Movement Simulator (SAMS) project document number C/NLR/00/006, July 2000, NLR Technical Publication NLR-TP-2000-552.
- [4] R. P. van Sterkenburg, A. A. ten Dam: “The scenario management tool SmartFED for real-time interactive high performance networked simulation”, NLR Technical Publication, NLR-TP-98577 or proceedings of HPCN Europe '99 Conference, 1999.
- [5] E. Kessler, A. A. ten Dam, E. van de Sluis, R. P. van Sterkenburg: “Divide and Control: Making Distributed Real-Time Simulations Work.”, NLR, proceedings of the SESP 2000 conference, 2000, NLR Technical Publication NLR-TP-2000-412.
- [6] IEEE standard 1516: “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules.” 2000.
- [7] Department of Defense (DoD) Modeling and Simulation Office (DMSO): “High Level Architecture – Federation Development and Execution Process (FEDEP) Model”, version 1.5 – December 1999.
- [8] EUROCAE ED-12B/RTCA DO-178B: “Software Considerations in Airborne Systems and Equipment Certification.” EUROCAE WG-12/RTCA SC-167, 1992.
- [9] IEEE Std 1278.4-1997 Trial-Use Recommended Practice for Distributed Interactive Simulation-Verification, Validation, and Accreditation.
- [10] Risk based Validation & Verification Levels Definition, Dirk Brade & Andreas Koester, In: proceedings EUROSIV 2001 (paper 01E-SIW-013)