



NLR-TP-2004-382

A three pronged approach towards collaborative engineering

E. Kessler, E.H. Baalbergen and W.F. Lammen

This report has been based on a paper presented at the first Concurrent Engineering for space applications Workshop 2004, ESTEC Noordwijk, September 30th – October 1st 2004.

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: AV.1.J.2
Owner: National Aerospace Laboratory NLR
Division: Aerospace Vehicles
Distribution: Unlimited
Classification title: Unclassified
September 2004

Approved by author:  8/10/2004	Approved by project manager:  8/10/2004	Approved by project managing department:  14/10 2004.
---	--	---



Summary

This paper describes the experiences obtained with three complementary approaches towards the integration of distributed engineering centres. The discussed cases involve aerospace engineering activities. Given the similarities between this domain and the space domain, the obtained results are applicable for concurrent engineering of space applications as well. The first example describes the integration of black-box systems models into a single-site simulation model at the top system-level. The second example presents the workflow concept supporting organisations to collaborate in the evaluation and validation of new design concepts. The last example discusses the integration of geographically dispersed real-time simulators of co-operating centres to arrive at a real-time synthetic environment to design and evaluate the product.



Contents

1	Introduction	4
2	Simulation case study on co-operative level	5
2.1	State of the Art Description	5
2.2	Aircraft domain collaboration example	6
3	Workflow case study on co-ordinated level	8
4	Geographically distributed simulation case study on concerted level	12
4.1	State Of The Art Description	12
4.2	Air Transport Domain Example	15
5	Conclusions	17
	Acronyms	17
	References	18

(20 pages in total)



1 Introduction

Collaborative engineering requires the participating parties to co-operate on the two conceptually distinct levels:

- On subject level. This level is also referred to as main business by [1] or, based on notions from activity theory [2], as design rationale or future artefact by [3];
- On organisational level, which [1] cites as meta communication or [3] cites as hypothetical user activity.

For the organisational-level collaboration, [4] defines a five-level taxonomy for classifying supporting systems. The taxonomy starts with communicative collaboration up to concerted collaboration. Each higher level extends the capabilities provided by the lower levels. On the communicative level, collaboration is limited to exchanging subject-level information via a shared, usually web-based repository. On collective level, also organisation-level information, like planning, resource tracking, etc, is exchanged via the shared repository. The co-operative level adds the sharing of common applications. However, each application is used locally on local copies of the information. The first case study elaborates this type of co-operation. The co-ordinated level adds co-ordination between the distributed team members. The result is a co-ordinated flow of activities from one team member to another, where each team member uses the result of the previous team member and in turn delivers its results to the next. This is also referred to as workflows. The second case study presents an example of this type of collaboration. The concerted level extends this collaboration to support synchronous and asynchronous problem solving by a distributed team. The third case study falls in this category.

Between various distributed collaborative projects, organisational and technical situations vary greatly. To provide an overview of the variety of approaches to match these diverse challenges, the sequel elaborates three carefully chosen complementary case studies from the aerospace domain. Each case will elicit the merits of the approach taken, concentrating on the collaboration, to increase the relevance of the findings for the space domain.



2 Simulation case study on co-operative level

This section is based on experience obtained in projects on co-operative level, which simulate integrated systems.

2.1 State of the Art Description

In system design, numerical simulation and optimisation of design objectives are commonly used. Aerospace systems can be quite complex and are usually part of one or more "higher level" systems. To analyse the behaviour of the integrated system, the system model may be composed of subsystem or component models from various suppliers. The physical behaviour of the sub-systems and components is a key determinant for the system behaviour. Therefore, it is important that this physical behaviour is adequately modelled, both with respect to the component behaviour and with respect to system behaviour. Sometimes the model of the physical behaviour of a (sub-)system or component may be too complex to be simulated efficiently within the constraints of the integrated system model. Within such a system model one or a few component models of extreme complexity may exist among several relatively simple component models, resulting in an undesirable and unbalanced system model.

Alternatively, in collaborative development projects, sub-system or component information may be supplied from one company to another and therefore proprietary constraints may prevent the use of detailed models of the physical behaviour of sub-systems or components. Some component models are, for example, represented by no more than a table with measurement results of the component behaviour. In such situations approximate representations based on system data sets can be used as effective and efficient alternatives.

A large variety of methods and tools is available for approximating system behaviour that is given by data sets. In order to collect and streamline some of the readily available functionality for approximation, a number of approximation methods has been integrated in a Matlab based software tool. This tool, named MultiFit and developed at NLR, provides a coherent and intuitive Graphical User Interface (GUI) to a variety of approximation methods based on polynomial functions, splines, neural networks, radial basis functions and kriging models. Furthermore the MultiFit tool is equipped with facilities for inspection and analysis of the data sets and for automatic export of the approximate models to Modelica code. Modelica [5] is a modelling language for multi-physics applications. NLR uses Modelica as modelling platform for the analysis of integrated system behaviour [6, 7, 8].

The functionality of MultiFit is illustrated in the Figure 2-1 below. On the left the MultiFit GUI window for data selection and inspection is shown. In the left hand side of this window the user can divide the considered data set into separate data sets for the approximation (fitting) and for

the assessment of approximation errors (verification). The graph in the right hand side of this window shows a plot of the selected fitting and verification data sets projected in a plane spanned by dependent and independent variables that can be selected from pull down menus by the user. It should be noted that data sets of any dimension can be approximated, i.e. $y=f(\mathbf{x})$ where a scalar ($y \in \mathbb{R}$) is fitted to the considered data as a continuous function (f) of an arbitrary number (n) of independent variables ($\mathbf{x} \in \mathbb{R}^n$). The right hand side of Figure 2-1 shows the MultiFit main window. In the left hand side of the window, the user can select and control the approximation method to be applied. On the right the quality of different fits to the considered data set can be compared and assessed. In the presented graphs, a comparison of the RMSE values of 2nd and 3rd order polynomial fits and a kriging model approximation of a data set is shown. Also shown in this MultiFit main window, is the File-pull down menu, from which a resulting approximation model can be directly exported to a Modelica source code file. A more elaborate description of the functionality of the MultiFit tool, and in particular of certain aspects of the export to Modelica, is given in [7].

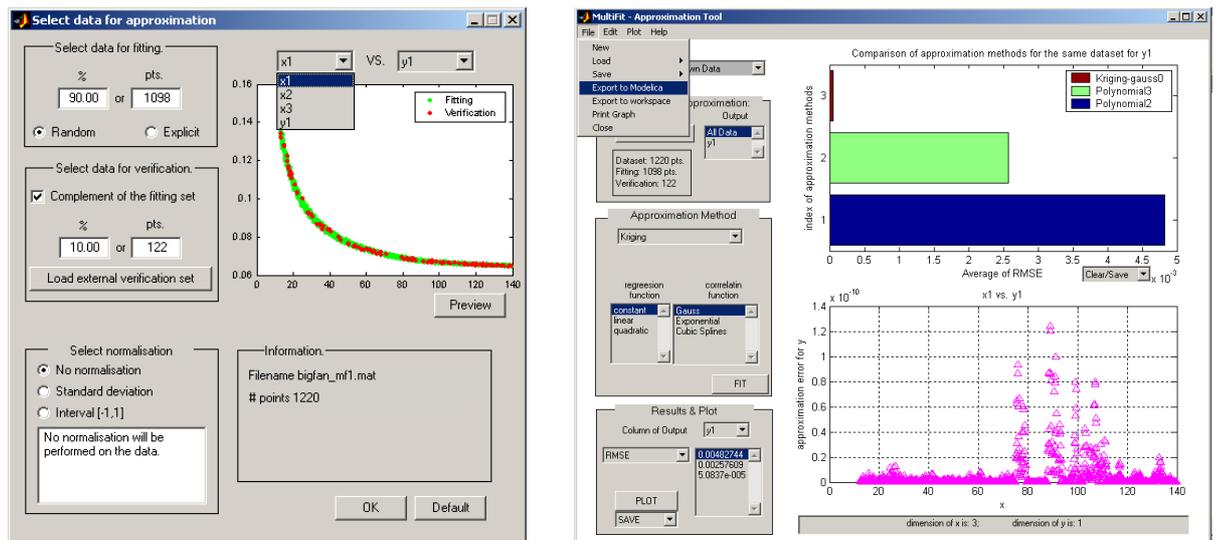


Figure 2-1 Illustration of the main functionality of the MultiFit Tool.

2.2 Aircraft domain collaboration example

The POA (Power Optimized Aircraft) project [9] investigates the optimisation of non-propulsive power consumption in civil aircraft, see Figure 2-2. POA is developing an integrated aircraft-level system model, the so-called “Virtual Iron Bird” (VIB). Because of the multi-physical nature of the considered system models, the Modelica modelling language is used as the basis for the integrated aircraft level system model of this POA VIB [10]. These models are supplied by various organisations.

Power supply and consumption aboard modern aircraft involves a large variety of systems that have to operate in a wide range of conditions. Globally there are (groups of) systems of electric (e.g. generators, cabin equipment, avionics), mechanic (e.g. engine shafts, gearboxes), pneumatic (for example air-conditioning, which is part of the environment control system (ECS), wing ice protection system (WIPS)) and hydraulic (for example flight control system (FCS)) nature, and of combinations thereof. To investigate the operational behaviour of these systems under many different conditions, it is important to have integrated models incorporating these different systems.

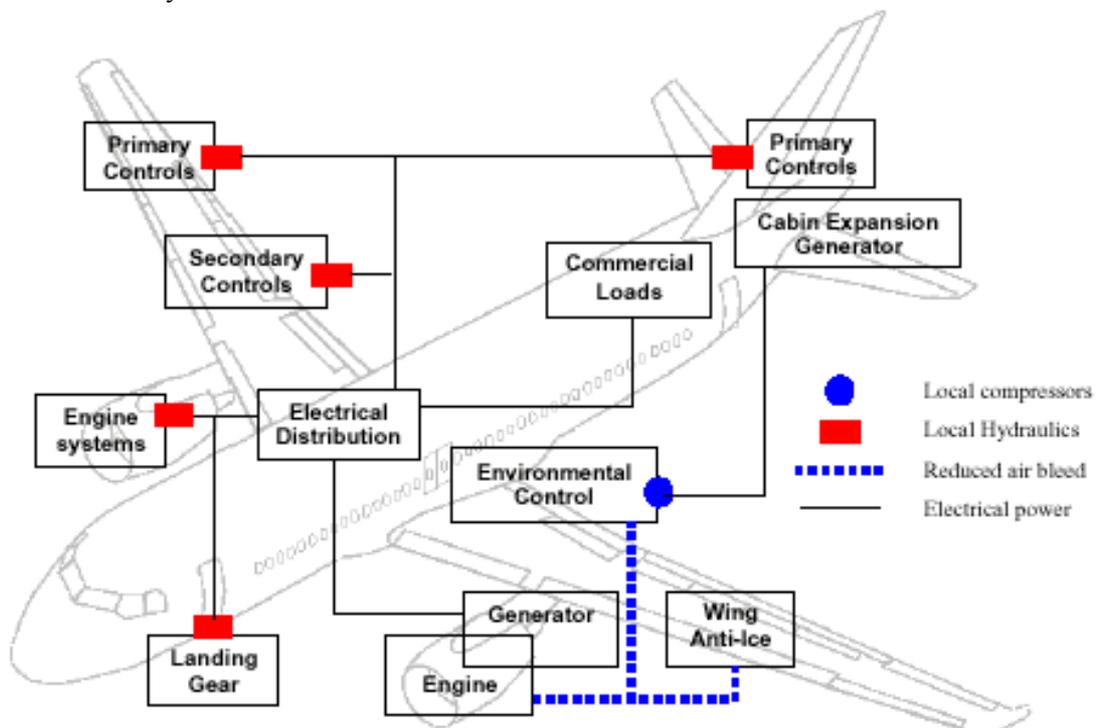


Figure 2-2 Overview of the aircraft systems considered in the POA project.

Figure 2-3 illustrates an implementation of a simplified integrated aircraft model in the Modelica based modelling and simulation environment Dymola [11]. In order to provide the engine power to the electric consumers, the (rotational) “mechanic” power of the engine shaft is converted via a gearbox and a generator, which are also included in the integrated model. To connect the consumers to the engine, appropriate physical connector objects as provided by Modelica are used. In this case the quantitative behaviour of some of the components (ECS and Engine) is based on data sets. These data sets may arise from complex system simulations (e.g. computationally expensive Computational Fluid Dynamics (CFD) simulations of the cabin airflow in the case of the ECS model) or experiments, which represent the underlying system behaviour. A more detailed description on this case study can be found in [8].

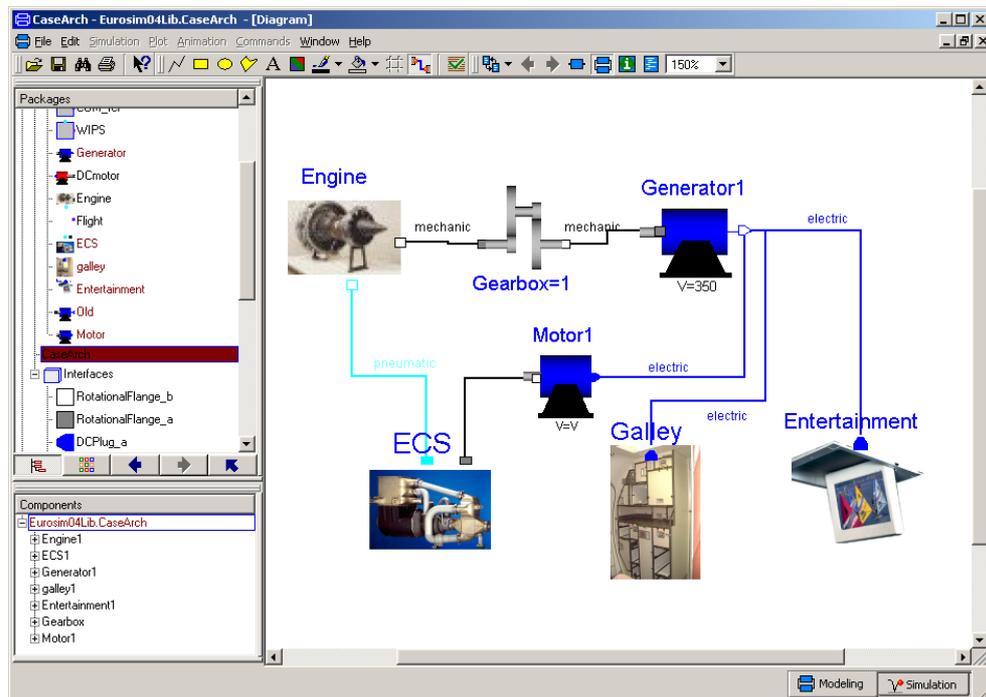


Figure 2-3 Dymola window showing the aircraft level integrated system model in Modelica.

To conclude, the combination of the NLR tool MultiFit with Modelica supports the full process from approximation of data sets to the integration with other multi-physics components for system optimisation. Specifically when models, delivered by various organisations, restrict information access or are computationally complex, the approximation approach promises to be useful for integrated system design in a co-operative setting.

3 Workflow case study on co-ordinated level

Experiences gained by NLR through the years, with multi-partner projects aimed to support multidisciplinary and collaborative engineering on the co-ordinated level, have been collected in a framework and supporting product suite. This framework, depicted in Figure 3-1, serves as blueprint for collaborative engineering platforms. Such platform provides the engineer with integrated access to the resources (computers, tools, data sets) that support the engineering activities. A key element in the platform is the “workflow”, which allows definition and controlled and automated execution of a chain of tools. This concept supports the definition and application of engineering procedures. Realisation of a collaborative engineering platform is supported by a dedicated product suite, in which expertise in the area of constructing collaborative engineering platforms has been translated to a set of generic tools. As an example of a collaborative engineering platform, we present the Integrated Technology Evaluation

Platform (ITEP) developed in the large integrated European Union project FACE (Friendly Aircraft Cabin Environment) [12].

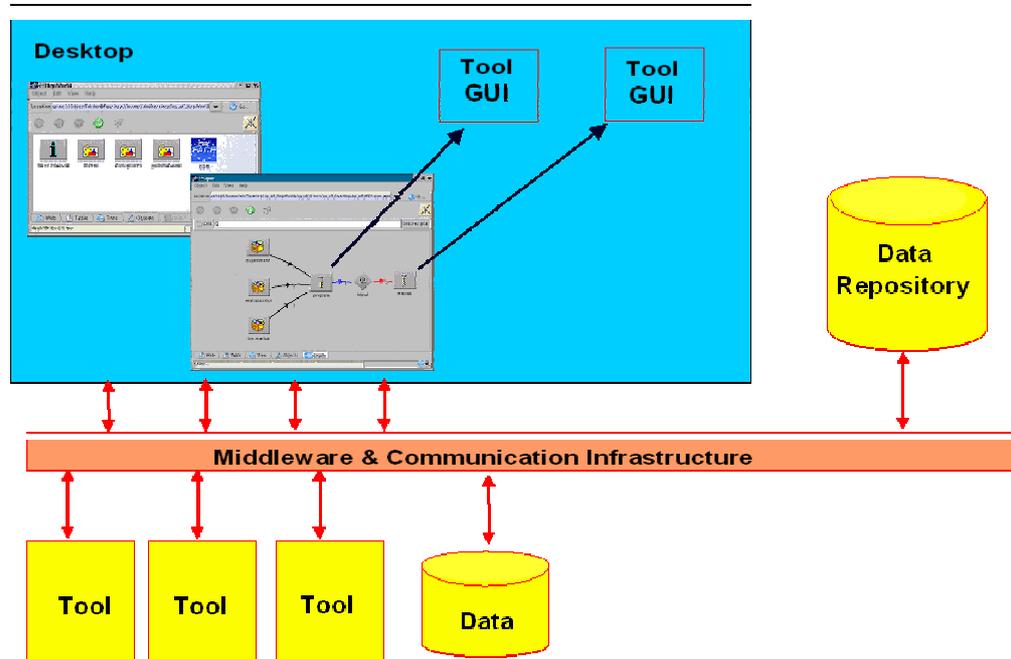


Figure 3-1 Framework for collaborative engineering platforms.

The FACE project started in 2002, lasts 4 years, and is carried out by a consortium of 30 partners from European aircraft industry, aerospace research, and several universities. The project focuses on development and evaluation of new multi-disciplinary technologies to treat noise, vibration, and air quality in order to improve environmental passengers and crew comfort in civil turbofan aircraft cabins. The new technologies will be applied in European aircraft projects in the near future. Activities in the project include testing of new concepts “in hardware”, through hardware experiments, as well as evaluations of these concepts “in software”, through numerical simulations. New concepts are assessed by validating the numerical simulation results against the experimental results. For the validation and assessment of the concepts, implemented and tested in the FACE project, evaluation procedures have been developed.

ITEP is developed in the FACE project to support the project partners to collaborate on the validation and evaluation of new design concepts. ITEP provides the engineer with an environment that supports integrated access to, organisation of, and easy exchange and reuse of tools, data, simulation models, evaluation procedures, and test results involved in the evaluation procedures and located with the various partners. ITEP is accessible to the project partners with appropriate security measures, thereby meeting the project as well as the partners’ requirements with respect to authenticated access and secure communication. ITEP provides the engineer



with an easy-to-operate, single computer that is tailored for the execution of evaluation procedures.

ITEP comprises three main subsystems: Web Portal, Evaluation Data Repository, and Working Environment. The Web Portal is collection of web pages providing documentation, access to the help desk, and entry points to the other subsystems. The Evaluation Data Repository provides a single repository for storing, searching, and retrieving the information (data, models, code, manuals) involved with the evaluation procedures, irrespective of the used file format. The repository provides world-wide access via its web interface, controlled access to its contents, and automated configuration management, and hence is suited for the management of data in collaborative engineering environments. The Working Environment combines the available resources, including computers, tools and data, into an easy-to-operate, single computer. The engineer operates this computer using an intuitive graphical user interface, available via a web interface. The GUI enables the user to manipulate data, to execute tools, and to define and execute workflows, via simple point&click and drag&drop operations on icons in windows. The major goal of the Working Environment is to let the engineer concentrate on the actual job instead of struggling with computers, networking, scripts, programs, files, operating systems, and interfaces. An arbitrary tool may be integrated easily into the Working Environment through tool wrapping, which allows integration of any legacy (including commercial) tool as is, without need for rebuilding the tool. The GUI of an integrated tool is that of the tool proper, but the Working Environment enables remote operation of the tool's GUI. Integrated tools provide a uniform way for the engineer to start tools (thereby providing input data and tool options, and dealing with output data) and combine and use tools in workflows.

A workflow is a chain, or more precisely, a graph of tools. It may be composed by positioning tools and so-called data containers (representing data sets involved with the tools) on a canvas, and drawing connections among the tools and data containers; cf. Figure 3-2. In ITEP, a workflow is used for definition and execution of evaluation procedures in terms of scenarios comprising tools and data, which may be located with and owned by the various partners. Execution of a workflow typically involves the execution of tools running on, and exchange of data in a network of heterogeneous computing systems, which potentially spans parts of the partners' networks. Mechanisms are available to co-ordinate the concurrent use of workflows and workflow elements. As such, the workflow supports collaboration with respect to evaluation procedures, and, consequently, is an essential building stone for collaborative engineering platforms.

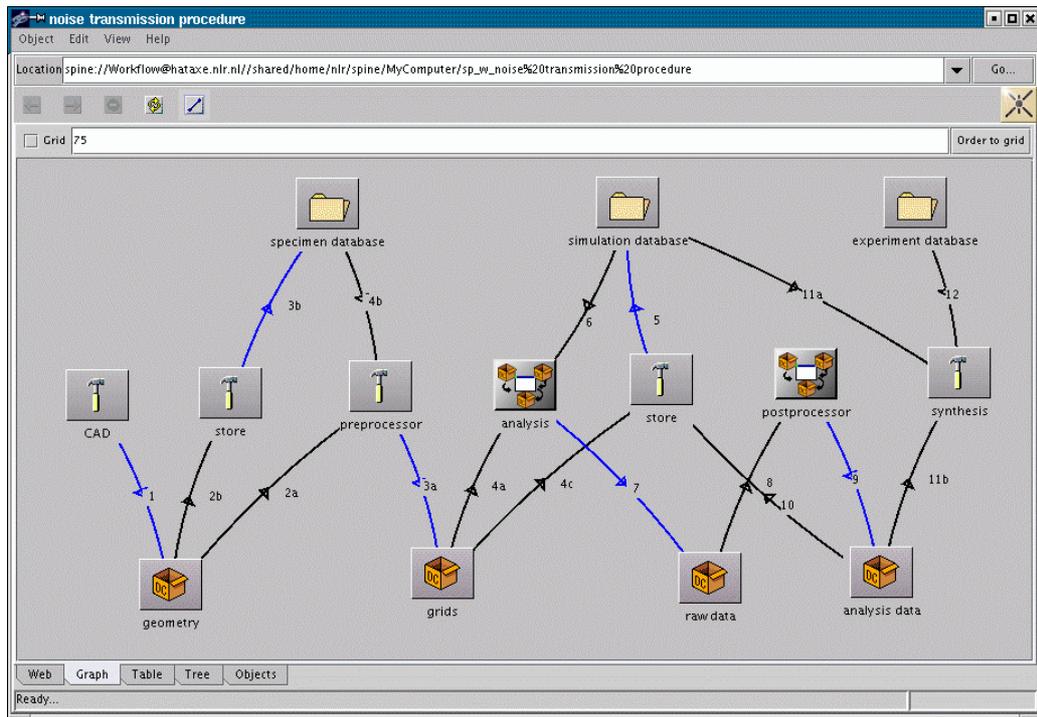


Figure 3-2 Example workflow in ITEP.

ITEP is implemented using the aforementioned product suite for realisation of collaborative engineering platforms. The suite applies and combines state-of-the-art and off-the-shelf techniques (such as CORBA, JAVA and the web), middleware products, and security technologies (e.g., virtual private networks, Secure Shell, Secure HTTP, signed applets) to support the realisation of platforms on top of existing infrastructures. The ITEP Evaluation Data Repository is realised as an instance of NLR's Model and Simulator Repository (MSR) [13]. The ITEP Working Environment is implemented using NLR's SPINeware [14, 15]. Complete ITEP is available to the project partners as a web application. It is accessible over the Internet, via its Web Portal, *faceitep.nlr.nl*. The web interfaces of the subsystems collectively enable the users to access the platform using the local desktop's Java-enabled native browser, and hence make the ITEP easily accessible to the engineers in the project. The web interface and the provisions for concurrent access to and use of its contents, ensure that ITEP is a suitable platform that serves the collaborative engineering needs of the FACE project, classified at the co-ordinated level.



4 Geographically distributed simulation case study on concerted level

This section describes concerted collaboration, i.e. problem solving by a distributed team, each of which owns unique assets like the real-time simulators needed to solve the collaborative task.

4.1 State Of The Art Description

Modelling is used extensively in aerospace engineering to study the behaviour of subsystems before realisation. When in the engineering process models are combined with realised subsystems, real-time simulation is required. Examples are hardware-in-the-loop and human-in-the-loop simulation. To simulate a subsystem with the required fidelity, often parts of its environment, that is, the other subsystems, have to be simulated as well. Simulation fidelity will increase when for these adjacent subsystems the simulators of the developing organisation can be used. Usually such simulators are available as engineering process artefacts of the adjacent subsystem and represent a significant effort. For many reasons, including non-portable facilities and concerns about guarding the knowledge embedded in the simulator, connecting existing simulation facilities is a preferred option. The resulting network of distributed real-time simulators provides a synthetic environment for the system being developed.

The various types of use of a synthetic environment are:

- Training or e-learning, in which a group of people (the trainees) are subject to a curriculum containing a fixed set of scenarios;
- Mission planning, in which a group of operational experts plans or optimises its task, by simulating the mission, analysing the result and retrying using a modified scenario, for example, cosmonauts planning the optimal use of a robotic arm within applicable safety constraints;
- Networked research and development, in which a group of engineers create or improve a product or service by changing the scenario as well as modifying the characteristics of the participating real-time simulators. Even the participating entities of the networked simulation can be changed. Collaborative engineering requires these capabilities.

To illustrate potential benefits of distributed simulation an example related to the air transport domain is elaborated.

The small volume of the aerospace market with respect to other markets implies that Commercial Of-The-Shelf (COTS) products and standards have to be used wherever possible to contain costs, project risk and time-to-market. As a result our approach to distributed simulation, called SmartFED (Scenario Manager for Real-time Federation Directing), is based on existing standards, the High Level Architecture (HLA). The use of HLA continues to increase, from being adopted by the US military in 1995, via selection by NATO in 1998 to a



general domain IEEE standard in 2000 [16, 17, 18]. Reflecting this HLA user base, each year three dedicated conferences are held [19]. Already in 2000, ESA determined that for the large Automated Transfer Vehicle programme, the distributed simulation approach is feasible. ESA estimated this approach would reduce costs and reduce time-to-market with 20% [20].

HLA provides an architecture to support component based simulation, with each real-time simulator being a component. On each simulator site, the so-called Run Time Infrastructure (HLA-RTI) takes care of all communication. The COTS approach efficiently ensures that new hardware and software platforms will be supported, whenever they become available. Furthermore it facilitates provision of supporting tools, which any single application could not afford.

HLA allows each participating simulator to select the set of attributes of objects it will make available to the other distributed simulators. Other objects can remain private and hence inaccessible for the other simulators amongst others to protect proprietary knowledge. The Simulation Object Model (SOM) describes all data needed by and provided by a real-time simulator. All data published in or subscribed to in the entire distributed simulation, or federation in HLA parlance, is contained in the Federation Object Model (FOM). As long as the FOM vocabulary contains all data subscribed to by a simulator, that simulator can participate in the federation. When another federate provides the same Simulation Object Model (e.g. an updated, faster or more accurate version), it can replace the original simulator without affecting the entire distributed simulation. This feature greatly improves the re-use of both simulators and distributed simulations.

As the process of developing a distributed HLA simulation remains quite complicated, it evolved into the 7-step Federation Development and Execution Process (FEDEP) process. To incorporate sufficient practical experience, FEDEP took an additional 3 years to mature before being standardised [21]. The FEDEP is depicted in Figure 4-1.

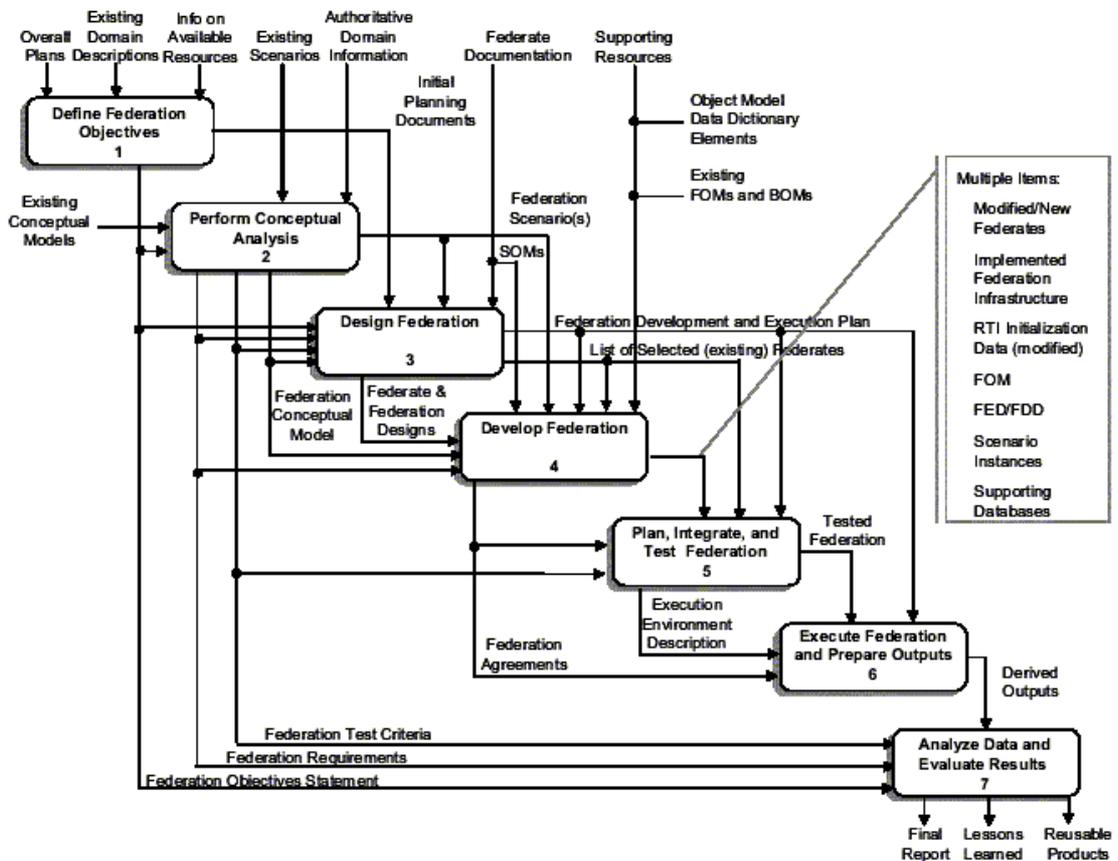


Figure 4-1 Federation Development and Execution Process (FEDEP) process.

The 7 FEDEP steps are:

- **Step 1: Define Federation Objectives.** The federation user, the sponsor, and the federation development team define and agree on a set of objectives and document what must be accomplished to achieve those objectives;
- **Step 2: Develop Federation Conceptual Model.** Based on the characteristics of the problem space, an appropriate representation of the real world domain is developed;
- **Step 3: Design Federation.** Existing federates that are suitable for reuse are identified, design activities for federate modifications and/or new federates are performed, required functionalities are allocated to the federates, and a plan is developed for federation development and implementation;
- **Step 4: Develop Federation.** The Federation Object Model (FOM) is developed, federate agreements are established, and new federates and/or modifications to existing federates are implemented (as required);
- **Step 5: Integrate and Test Federation.** All necessary federation implementation activities are performed, and testing is conducted to ensure that interoperability requirements are being met;



- **Step 6: Execute Federation and Prepare Outputs.** The federation is executed and the output data from the federation execution is pre-processed;
- **Step 7: Analyse Data and Evaluate Results.** The output data from the federation execution is analysed and evaluated, and results are reported back to the user/sponsor.

The FEDEP process is already supported by a number of tools. Recent work [22] lists 80 tools for the FEDEP process, with at least one tool per FEDEP step. Tool support for the first two steps and the last is scant though. During a 13-nation federation development the guidance provided by FEDEP was appreciated [23]. Still, the first FEDEP steps take long, as they involve organisational level communication for which FEDEP tool support is limited. However, these first FEDEP steps are crucial as they determine the success of the project [23].

The work on exercise management described in this section concentrates on FEDEP steps 5 and 6. The resulting tool, Scenario Manager for Real-time Federation Directing (SmartFED) [24] has three main functions, each implemented in its own independent component:

- Federation Management, to control the execution state of all participating simulators in the entire federation. For this all simulators must comply with a common state transition diagram, even when several states might not be implemented in a particular simulator;
- Federation Monitoring, to allow any user in the federation to monitor any object defined in the Federation Object Model of the distributed simulation. This monitoring can be performed on any location, of the distributed simulation. The data can be presented graphically or textually;
- Scenario Definition and Execution Management, the definition part supports the off-line definition of the participating simulators initial conditions. Also run-time events like the generation of simulated failures or meteorological conditions (e.g. fog) are defined. Once the simulation is being executed, the execution part of this component will activate the events at the predefined times or predefined conditions or on operator request.

4.2 Air Transport Domain Example

This section describes how SmartFED supports concerted collaborative engineering. In the air traffic management domain, real-time simulations are a necessary step between a research solution and a costly multi-national implementation. Some air traffic management tool sets can only be evaluated by exposing the air traffic controllers to them in a real-time simulation. This involves using geographically distributed simulators, or collaborative engineering.

After concentrating on improving the capacity of the airborne part of the air transport system, airport capacity limitations are becoming more visible. Current stand-alone systems are already optimised, so additional capacity improvement depends on optimising several connected systems. Air traffic management tools being developed include an arrival manager for incoming



aircraft and a departure manager for departing traffic. To assess whether the proposed tool set co-operates as expected, a collaborative scenario is conceived in which an incoming flight is delayed. In order for the transfer passengers to catch their connecting flight, the airline might request landing priority for this flight with respect to its other incoming flights combined with a request for a specified delay of the departing flight. Such scenario implies rescheduling of incoming and departing aircraft. All of this has to be accomplished with minimal impact on the overall air traffic system capacity and without compromising safety. A distributed simulation can be made, consisting of an approach simulator containing the arrival manger, a flight simulator for the late aircraft and an airfield simulator including the departure manager. Figure 4-2 shows the resulting distributed simulation. More information on this example can be found in [24]. In a similar case, integrating independent tools yielded the surprising finding that in special conditions the expected capacity gain could not be realised [25]. These conditions occurred sufficiently frequent to necessitate a redesign of the proposed tool set. Such results vindicate the need for collaborative engineering via distributed simulation prior to actual trial that could endanger human life. Others, [23], also confirm that distributed simulation is a cost-effective and safe way to examine new concepts for systems-of-systems.

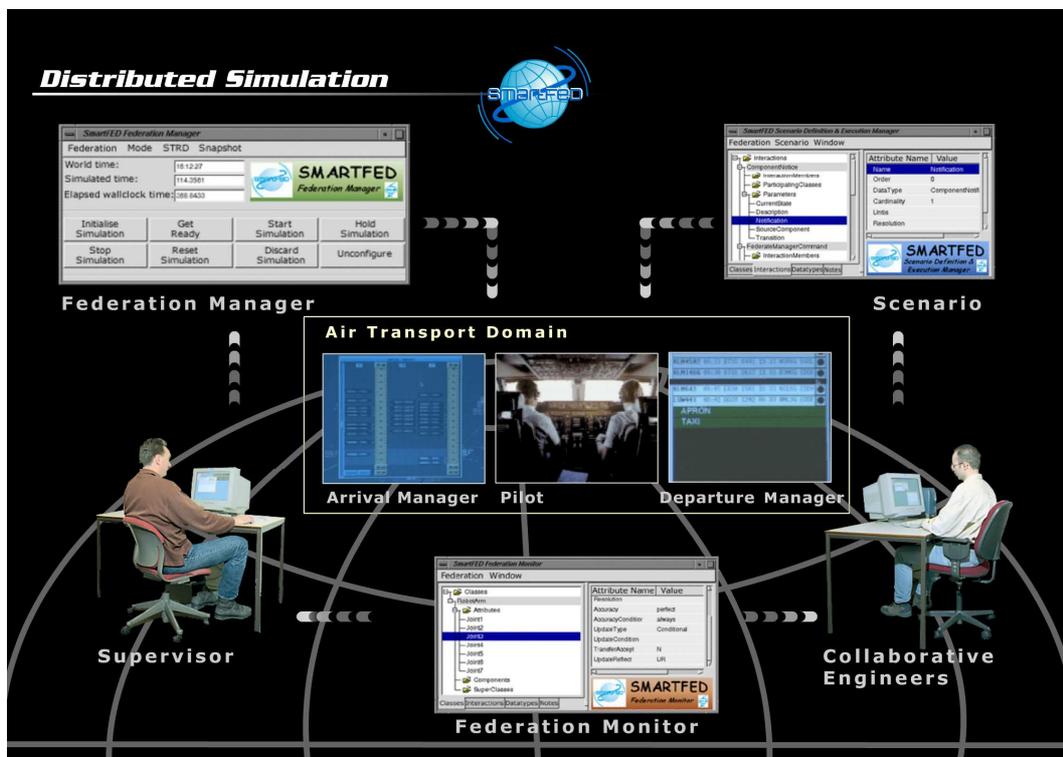


Figure 4-2 Example of collaborative engineering in air transport domain.



5 Conclusions

Collaborative engineering can be beneficial in various situations. Examples are provided for collaborative engineering on co-operative, co-ordinated and concerted levels. As the case studies show, these different environments are best served with dedicated tools.

The combination of the NLR tool MultiFit with Modelica supports the full process from approximation of data sets to the integration with other multi-physics components for system optimisation. Specifically when models, delivered by various organisations, restrict information access or are computationally complex, the approximation approach is useful for integrated system design in a co-operative setting.

Co-ordinated collaboration supports the flow of activities between distributed team members. Suitable supporting products such as SPINeware and MSR affordably provide significant value to the user.

Concerted collaboration supports problem solving by a distributed team. As an example, properly deployed distributed simulations help the design of large interacting systems-of-systems without jeopardising human life. High Level Architecture with its supporting FEDEP processes and tools supports the creation of the synthetic environment involved.

These case studies illustrate NLR's comprehensive experience with the various types of collaborative engineering.

Acronyms

CFD	Computational Fluid Dynamics
COTS	Commercial Of-The-Shelf
ECS	Environment Control System
FACE	Friendly Aircraft Cabin Environment
FCS	Flight Control System
FEDEP	Federation Development and Execution Process
FOM	Federation Object Model (
GUI	Graphical User Interface
HLA	High Level Architecture
ITEP	Integrated Technology Evaluation Platform



MSR	Model and Simulator Repository
POA	Power Optimized Aircraft
RTI	Run Time Infrastructure
SmartFED	Scenario Manager for Real-time Federation Directing
SOM	Simulation Object Model
VIB	Virtual Iron Bird
WIPS	Wing Ice Protection System

References

- [1] Geisler C, Rogers E.R, “Technological mediation for design collaboration”, IEEE professional communication society international professional communication conference and 18th annual ACM international conference on Computer documentation, September 2000.
- [2] Leontjev A.N, Activity, consciousness and personality, Prentice-Hall, Englewood Cliff, NJ, USA, 1978.
- [3] Tuikka T, “Remote concept design from an activity theory perspective”, 8th *European Conference of Computer-Supported Co-operative Work*, September 2003.
- [4] Romano N. C, Chen F., Nunamaker J. F, “Collaborative project management software”, 35th *HICSS*, January 2002.
- [5] Modelica Association (2002). *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling - Language Specification (Version 2.0)*, <http://modelica.org>.
- [6] Vankan W.J., Kos J, Lammen W.F, Approximation Models for Multi-Disciplinary System design - Application in a Design Study of Power Optimised Aircraft, *Proc. Eurogen 2003 Conference*, Spain, 2003. Also published as NLR-TP-2003-369.
- [7] Lammen W.F, Vankan W.J., Maas R, Kos J, Approximation of black-box system models in Matlab with direct application in Modelica , *Proc. 3rd International Modelica Conference*, Linköping, Sweden, <http://modelica.org/Conference2003.shtml>, 2003. Also published as NLR-TP-2003-583.
- [8] Vankan W.J, Lammen W.F, Kos J, Maas R, “Complementary Approximate modelling in Matlab and Modelica”. To be published in the *proceedings of the EuroSim 2004 conference*, 2004
- [9] POA - Power Optimised Aircraft Contract G4RD-CT-2001-00601 under the European Communities 5th framework Programme for Research - Competitive and Sustainable Growth - Key Action, New Perspectives in Aeronautics. <http://www.poa-project.com>, 2001.

- [10] [Bals J., Hofer G., Pfeiffer A, Schallert C. Object-oriented inverse modelling of multi-domain aircraft equipment systems with Modelica, *Proc. 3rd International Modelica Conference*, Linköping, Sweden, <http://modelica.org/Conference2003.shtml>, 2003.
- [11] Dynasim AB Developers of Dymola, <http://www.dymola.com>, 2004.
- [12] Paonessa A., "Friendly Aircraft Cabin Environment - FACE", *5th EURONOISE Conference*, Naples, May 19-21, 2003.
- [13] Moelands J.M, "Key features of the web-based Model and Simulator Repository MSR. Version 2, *NLR Technical Report NLR-TR-2004-262*, June 2004.
- [14] Baalbergen E.H., Ven H. Van der, "SPINeware - a framework for user-oriented and tailorable metacomputers", *Future Generation Computer Systems 15*, (1999) 549-558.
- [15] "SPINeware more than just middleware", <http://www.spineware.com>.
- [16] IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, *IEEE Std 1516-2000*, September 2000.
- [17] IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) Federate Interface Specification, *IEEE Std 1516.1-2000*, September 2000.
- [18] IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification, *IEEE Std 1516.2-2000*, September 2000.
- [19] Simulation Interoperability Workshop, <http://www.sisostds.org/>.
- [20] Argüello L., Miró J., "Distributed Interactive Simulation for space projects", *ESA bulletin 102*, May 2000.
- [21] IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP), *IEEE Std 1516.3-2003*, April 2003.
- [22] Federation Development and Execution Process (FEDEP) tools in support of NATO modelling and simulation (M&S) programmes, *NATO TR-MSG-005*, May 2004.
- [23] Reading R. et al, "Results and lessons learned from a multi-national HLA federation development supporting simulation based acquisition", *EURO-SIW 2002*, June 2002.
- [24] Kessler E., "Deploying networked real-time simulation, putting the virtual enterprise to work: Some aerospace experience", 35th *HICSS*, January 2002. Also published as NLR-TP-2002-003.
- [25] Kessler E., Knapen E, "Designing future advanced controller displays", *Information Design Journal 2002/2003*, p, 32-43, John Benjamins Publishing Company, ISSN: 0142-5471 E-ISSN: 1569-979X, 2003.



Acknowledgement

The POA project has been partly funded under contract G4RD-CT-2001-00601 under the European Communities 5th framework Programme for Research – Promoting Competitive and Sustainable Growth – Key Action 4: 'New Perspectives in Aeronautics'.

The FACE project has been partly funded under contract G4RD-CT-2002-00764 under the European Communities 5th framework Programme for Research, – Promoting Competitive and Sustainable Growth – Key Action 4: 'New Perspectives in Aeronautics'.