

# GeneSyS Project: Supervision of Distributed Systems (03E-SIW-043)

*Jean-Eric BOHDANOWICZ*

*Antoine LAYDIER*

European Aeronautic Defence and Space Company - Launch Vehicles (EADS-LV)  
66, route de Verneuil, BP 3002, 78133 Les Mureaux, France

[JeanEric.Bohdanowicz@launchers.eads.net](mailto:JeanEric.Bohdanowicz@launchers.eads.net), [Antoine.Laydier@launchers.eads.net](mailto:Antoine.Laydier@launchers.eads.net)

*Petr CHLIAEV*

*Alexander VANKOV*

*Vladimir VOLOCHINOV*

D-3-Group GmbH

Rudower Chaussee 29, D-12489 Berlin, Germany

[Petr.Chliaev@d3group.com](mailto:Petr.Chliaev@d3group.com), [Alexander.Vankov@d3group.com](mailto:Alexander.Vankov@d3group.com), [Vladimir.Volochinov@d3group.com](mailto:Vladimir.Volochinov@d3group.com)

*Andrey SADOVYKH*

Laboratoire d'Informatique, Université Paris 6 (LIP6)

8, rue du Capitaine Scott, 75015 Paris, France

[Andrey.Sadovkykh@lip6.fr](mailto:Andrey.Sadovkykh@lip6.fr)

Keywords:

GeneSyS, Supervision, Middleware, Agent, Web Services, SOAP, HLA, Simulation.

**ABSTRACT:** As complexity of distributed engineering, simulation, training and learning systems permanently grows, it is necessary to have a capability to effectively monitor and supervise remote components of such distributed systems. For this purpose, solutions suggested in the Generic System Supervision (GeneSyS) project could provide a common set of supervising agents and middleware to interconnect them. GeneSyS is aimed at developing a new generic middleware for supervising distributed systems at different levels such as Application, Network, System. An architecture based on agents and cores was selected to provide this functionality. The cores are middleware components that manage interactions providing look-up service to agents. It is planned to use SOAP-RPC and SOAP messaging standards as transport for GeneSyS messages. A simple XML-based protocol is used for communications. GeneSyS version 1 includes the following physical components:

- Core: a GeneSyS entity capable of storing information about other GeneSyS components and facilitating service discovery.
- Monitoring agent: a GeneSyS entity capable of measuring and publishing monitoring messages.
- Monitoring console: a GeneSyS entity capable of displaying monitoring messages in a human readable form.
- Repository: a GeneSyS entity capable of consuming monitoring messages, logging them, retrieving and publishing the log messages.

## 1. Introduction

GeneSyS is a European Union project (IST-2001-34162) co-funded by the Commission of the European Communities (5th Framework). EADS Launch Vehicles of France is the project Coordinator, with University of Stuttgart (Germany), MTA SZTAKI (Hungary), NAVUS GmbH and D-3-Group GmbH (both of Germany) as participants. GeneSyS started in March 2002 with

planned completion in September 2004 [1]. This paper presents basic concepts and achievements of GeneSyS project. Because the project is yet in its initial phase, the paper is more focused on architectural issues with only some preliminary implementation results discussed. The following sections show motivation and objectives of the project.

### 1.1 Constraints of existing systems

There are many commercial supervision systems currently available on the market such as like Tivoli (from IBM), OpenView (from Hewlett-Packard), etc. They are aimed at different aspects of system monitoring starting from operating systems to network and up to some commercial standard applications.

However most of them have several common constraints [2], which GeneSys should overcome:

- ? proprietary interfaces;
- ? proprietary protocols;
- ? operating system dependent implementation;
- ? non-flexible architecture;
- ? obstinate dedication to particular commercial application (Oracle, SAP, etc.).

These constraints complicate integration between third-party monitoring tools to achieve system control on all the levels. At the mean time, proprietary solutions slow down pace of development of the whole domain.

The next section introduces major objectives of the GeneSys project.

## 1.2 Objectives

In order to solve problems listed in the previous section, GeneSys consortium is developing an open generic concept of distributed system supervision.

The main challenges the consortium is facing include:

- ? Specifying and developing an open, generic, modular and comprehensive supervision concept,
- ? Integrating and validating this supervision structure within various industrial contexts,
- ? Achieving adoption or taking into serious account of the GeneSys concepts by appropriate simulation communities and major market stakeholders, and ensuring that the vision of the proposed generic structure could become a basis for the new emerging standard.

The next section presents main architectural design issues of the project.

## 2. Concept

### 2.1 General Architecture

To achieve a high level of flexibility, it has been decided to separate monitoring tools from visualisation application and communication bus. During the design

phase, several components were identified and designed for future implementation, as follows:

- ? Monitoring Agent implementing an interface to the entity being monitored;
- ? GeneSys Connector plugging agents to the communication bus;
- ? Console Agent visualising monitoring data in a user-friendly form;
- ? Directory Server providing publish/subscribe and look-up services;
- ? Repository purposed to log the data collected.

General architecture of the system is depicted in Fig. 1.

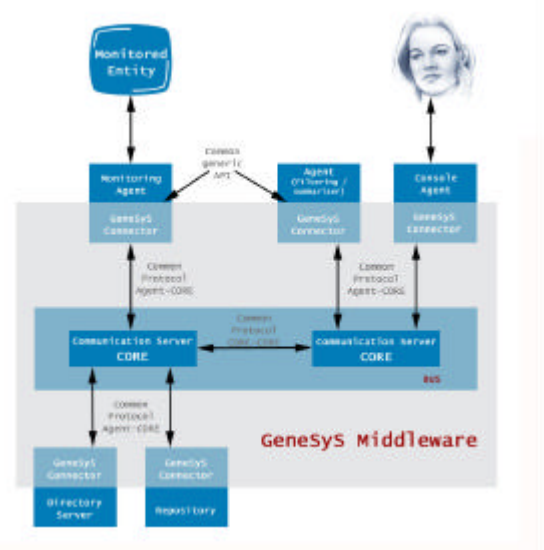


Fig. 1: General illustration of GeneSys concept

Communication bus has a platform independent implementation to manage data exchange in a standardised manner. Certain flexibility is achieved by separating visualisation facilities from the instrumented applications connected to monitored entity.

The Core is the major part of Communication bus. It provides routing, lookup, publish/subscribe and logging services using Repository and Directory Server.

All interfaces will be open and well documented. The components and protocols will be independent from the operating system and available under one of types of open source licenses. Standardisation activities are essential and important part of the project to be undertaken throughout its whole duration.

### 2.2 Implementation

Version 1 of GeneSyS implementation uses SOAP-RPC [3] as transport for monitoring and service messages. These messages are XML encoded using standardised forms. It means that agents and clients can be implemented using numerous programming languages from PHP to C/C++, Java, and Fortran.Net. In addition, the messages are easy readable and do not need complex parsing.

Agents were classified in Producers and Consumers. Producers provide monitoring data to Consumers, which process the data. Complex agents can realise both Producer and Consumer communication models, they can be used for intermediary data analysis, decision logic, etc.

Each agent should publish its capabilities to Core to be available for other agents. The Core is used to register/unregister, publish/subscribe and look up agents. Figure 2 illustrates interactions via GeneSyS Core.

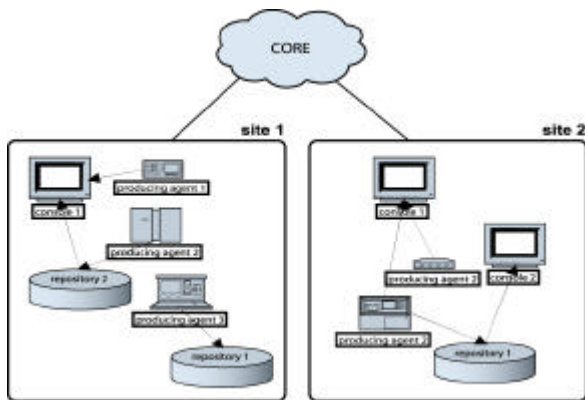


Fig. 2: Remote interaction via GeneSyS Core

So, the life cycle of a producer is as follows:

1. Register to a core
2. Publish capabilities (monitoring data types)
3. Send the data to consumers
4. Unregister.

On the other hand, the life cycle of a consumer includes:

1. Register to a core
2. Look up an agent
3. Subscribe to monitoring messages/Query the Agent
4. Handle incoming data
5. Unsubscribe
6. Unregister.

For GeneSyS version 1, the following “general monitoring purposes” agents will be implemented:

- ? System agents for Windows and Linux workstations, which provide generic information about CPU load, memory and disk space usage;
- ? Network agent retrieving generic information using SNMP.

Additionally, for the “PDR validation scenario” (see section 3), the following specific agents are being implemented:

- ? Tomcat servlet engine agent,
  - ? EDB (Engineering Data Base) agent,
- both for monitoring special distributed PDR application users’ activity.

For all data-producing agents, appropriate “console” agents (monitoring data consumers) will be implemented.

All consoles will be capable to interact with multiple agents, and, vice versa, all agents will be capable to interact with multiple consoles.

It is worth mentioning that main data flows (carrying monitoring data) occur between producing and consuming agents directly. The core is used for agents’ registration/discovering only. As a result, the GeneSyS architecture principles resemble the “Peer-to-Peer” ones [4], with central “agents’ addresses database” service (GeneSyS core components).

In GeneSyS version 1, the core will be represented by special unique Web-service application accessible for any agent (to be involved in validation scenarios) via Internet.

The next section introduces validation applications. For each scenario specific agents will be implemented.

### 3. Validation Aspects

Three validation scenarios were selected for GeneSyS. Among them, there is Preliminary Design Review as a part of (distributed) engineering process.

#### 3.1 Distributed Engineering

Preliminary Design Review (PDR) is an essential phase of any spacecraft development. For the Automated Transfer Vehicle (ATV), which aim is to provide transportation services to the International Space Station, up to 200 reviewers from different countries are

involved in PDR. Fig. 3 gives illustration to ATV (right side) and geographical distribution of main development sites across Europe (left side).

Reviewers meet on-line to discuss Review Item Discrepancies using GroupWare facilities: Engineering Database (EDB) and video conferencing tools.

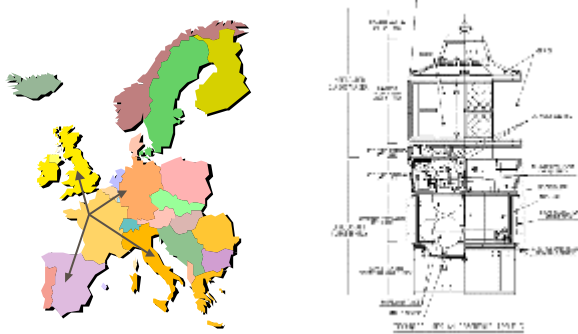


Fig.3: Distributed engineering of ATV spacecraft

Specific agents were developed to monitor Reviewers activities and to gather statistics. Generic system and network monitoring agents are used for maintenance. This allows for achieving of full control over the system starting from physical layer to application layer, groupware and workflow.

GeneSyS middleware interconnects generic components as well as specific PDR agents and consoles.

This should help Review Manager to better control PDR process using information about reviewers activities, work statistics.

Engineering Database (EDB) is a front-end to documentation repository helping to manage tracing all changes and updates through the review process. Small amount of code was injected to EDB to be notified about all activities concerning users and documentation. EDB monitoring agent (complex agent) interconnects EDB with a monitoring console. It also keeps logs in SQL database and transforms the data to a user friendly form. The monitoring data are processed to keep users informed about the status of the repository. EDB Agent uses GeneSyS compatible messages for communication, this means that EDB console can be easily implemented on different platforms including mobile solutions like mobile phones and PDAs.

### 3.2 Distributed Training

Distributed training scenario is planned for implementation in version 2 (scheduled for mid 2004), therefore only preliminary architectural issues are addressed hereafter.

The scenario will be based on a network of several remote simulators running with operators in the loop in the frame of space mission rehearsal and interconnected using HLA. It is essential to simulate in advance the contingency situations and train cosmonauts and ground controllers for off-nominal operations [5].

Each partner involved in the validation process will host one or several simulators (federates) and/or GeneSyS Consoles and will be connected to other partners through the WAN that will be used for the simulation data as well as for the supervision data.

The federation to be used for validation of GeneSyS software in distributed training scenario is illustrated by Fig. 4.

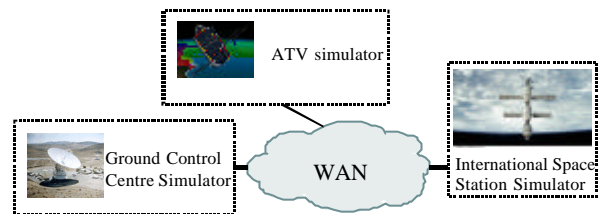


Fig. 4: Federation for distributed training

Three different kinds of users will use this application and GeneSyS:

- ? The trainees: they will be plugged to their simulation application through their specific federate: e.g. International Space Station federate for the astronauts and to the Mission Control Centre federate for the ground controllers.
- ? The instructors: they will be plugged to GeneSyS Core through specific GeneSyS Consoles and Federation Manager in order to have a synthetic view of the training simulations and the status of each trainee they are responsible for. They will interact together to select the given exercise to be run and for instance introduce contingency situation to verify that the trainees recognise this event has arrived and timely react by applying the proper procedure.
- ? The technical operators: they are in charge of controlling whole infrastructure (network, systems, RTI and involved applications). They

will be supported in their task by using the GeneSyS Consoles gathering information on different levels.

The GeneSyS agents will implement a front-end to Management Object Model of HLA to supervise RTI and Federations. Besides them, some other Agents will be deployed for monitoring of:

- ? WAN,
- ? Simulation hosts,
- ? Groupware tools/facilities.

#### 4. Conclusions

GeneSyS proposes a common monitoring framework which can be useful in different kinds of distributed systems and scenarios such as distributed simulation, distributed system engineering and ADL.

#### 5. References

- [1] GeneSyS project official web-site: <http://genesys.sztaki.hu>
- [2] A. Sadovykh, S. Wesner, J-E Bohdanowicz: "GeneSyS: A Generic Architecture for Supervision of Distributed Applications" EuroWeb 2002 Conference Proceedings, pp. 38-42, December 2002.
- [3] WebService Activity of W3C: <http://www.w3.org/2002/ws/>
- [4] Karl Aberer, Manfred Hauswirth: "Peer-to-Peer Information Systems: Concepts and Models, State-of-the-Art, and Future Systems" 18th International Conference on Data Engineering (ICDE), San Jose, California, 2002.
- [5] D. Michel, D. Claude, F. Dubuc, P. Shlyayev, A. Vankov, L. Arguello: "Spacecraft Operator Training Using Infrastructure Developed in EDISON Project" 2000 Spring Simulation Interoperability Workshop. Paper No. 00S-SIW-016.

#### Author Biographies

**JEAN-ERIC BOHDANOWICZ** graduated from the ENSEA in computing science and electronics with a speciality in the network and telecom fields. He joined EADS Launch Vehicles in 2000 where he managed several studies on networking supervision both in fields of R&D and industrial applications and in an international context.

He is now the IST-2001-34162 GeneSyS Project Manager.

**ANTOINE LAYDIER** graduated from the ISIMA engineering school in computing science and mathematics. He joined EADS LV in 1997 to participate in various international developments in distributed simulations, and developed a wide expertise in the most innovative technologies of information: CORBA, HLA, UML, ISDN, Linux, real-time systems, NTP, etc. He used to actively participate in Linux improvements (KDE project). He now manages projects at European level in the field of distributed systems and real-time simulations.

**PETR CHLIAEV** is the technical manager at D-3-Group GmbH. He has about 10 year research experience in simulation, information technology and computer science, mainly for space systems. He participated in several large international projects in the area of distributed interactive simulation and operator training (such as EDISON and DSE for EU, DIS-RVM and ESA-NASA HLA demo for ESA, etc.) where he was responsible for the development of some critical middleware components and participated in integration of training facilities. He is now actively involved in ATV-ISS spacecraft interface verification facility integration and testing.

**ALEXANDER VANKOV** is the business development manager of D-3-Group GmbH. Before joining D3 Group he worked at the Gagarin Cosmonaut Training Center for over 18 years and was involved in all major Russian manned aerospace programs. He participated in several large international projects in the area of distributed interactive simulation and operator training (such as EDISON and DSE for EU, DIS-RVM and ESA-NASA HLA demo for ESA, etc.). His current research activities are focused on new areas of using advanced simulation technologies.

**VLADIMIR VOLOSHINOV** is a senior research scientist having over 15 years of experience in mathematical modelling in economics and engineering systems; theory and numerical methods of optimisation. Participated in large international projects (including DSE for EU and ADL for ESA) in the area of distributed computer systems and ADL.

**ANDREY SADOVYKH** is a Ph. D. student of Laboratoire d'Informatique, Université Paris 6. He carries out researches at EADS Launch Vehicles participating in the GeneSyS project. His Ph.D. thesis is aimed at developing innovative middleware technologies for supervision of complex systems in the aerospace industry.

