GeneSyS – Generic Systems Supervision

Jean-Eric BOHDANOWICZ, EADS Launch Vehicles, Les Mureaux (EADS-LV), jeaneric.bohdanowicz@launchers.eads.net
Andrey SADOVYKH, Laboratoire d'Informatique de Paris 6 (LIP6), andrey.sadovykh@lip6.fr
László KOVÁCS, Balázs PATAKI, Zoltán TÓTH, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, (kovacs,pataki,zoltan.toth}@sztaki.hu
Stefan WESNER, High Performance Computing Centre Stuttgart (HLRS) wesner@hlrs.de

Abstract

This paper describes the design and architecture of the GeneSyS IST-2001-34162 project, which aims to define, implement and standardize a new supervision middleware for distributed systems and applications.

1. Objectives

GeneSyS is an Information Society Project (IST-2001-34162) started in March 2002. The GeneSyS project is done by a consortium including EADS-LV (France), NAVUS (Germany), D3Group (Germany), High Performance Computing Centre Stuttgart (Germany) and MTA SZTAKI (Hungary).

The aim of the GeneSyS project is to specify and develop a new supervision middleware for distributed systems and applications, as the need for a global and generic supervision solution has arisen among various industries. In fact, today solutions are mainly focusing on the supervision of low levels of a distributed system, i.e. hardware and network availability and monitoring. But these systems are becoming more and more complex and therefore, they need to be supervised in a new way:

- The supervision shall implement the control and monitoring not only of low level but also of higher levels (i.e., application, QoS, GroupWare, …) of the distributed system.
- The global supervision shall range from a passive monitoring of the resources to an active control of the applications running over these distributed systems.
- The supervision shall be applicable to a various set of different distributed systems and applications, i.e. it must be generic and open.

The top-level objectives of the GeneSyS project are:

1. To specify and develop an open, generic, modular and comprehensive supervision concept,
2. To integrate and validate this supervision structure within various industrial contexts,
3. To achieve the adoption of the GeneSyS concepts by all stakeholders (internal and external to the consortium), and to ensure that the vision of the proposed generic structure will become a new emerging standard.

2. Overview

The GeneSyS project prototype is composed of:

- a dedicated middleware to address the communication problem between heterogeneous platforms. It will be made of a Communication Server Core and Connectors,
- Several types of Agents to be plugged into the middleware. Some Agents interface with a Monitored Entity and transfer the information and functionality of a Monitored Entity via the use of GeneSyS Connectors into the GeneSyS middleware. Other Agents build the interface to Applications such as a Supervision Console with a graphical user interface or via other clients intended for users.
- Several "special" agents: one identified as the Directory Server is used by data producing agents to publish their capabilities. Consuming agents use the Directory Server to discover and locate other agents that publish the information they are interested in. Another one is the Repository. The Repository component is expected to store all forms and types of information that is produced and handled within the GeneSyS middleware.

Figure 1 GeneSyS Component Model
3. GeneSyS Architecture Elements

**Producer Agent:** This kind of components interfaces with components of the distributed application through different means. For network elements the Simple Network Management Protocol seems natural. For application monitoring proprietary protocols or an Application Programmer Interface can be used. The Agents are classified in types and the data format that is produced is specified using XML Schema. 

**Consumer Agent:** That data produced must be either stored or presented to a human. Consumer Agents can analyse the XML data transmitted by the Producing Agents. Consumer Agent can either be specialised for a specific data set or can be of general nature using XML Parsing technologies for a generic presentation or persistency mechanism. 

**Complex Agent:** It is very common that in large applications it is not useful to present all details available to the user. It seems much more reasonable to use the concept of a complex agent that collects the information from several Producer Agents, makes an analysis of the data and present only the derived information to Consumer Agents.

An example can be “Simulation Supervisor Agent” that collects information from the System Monitoring Agents responsible for the status of the involved systems, Network Monitoring Agents that collect the information on the network quality and the Application Monitoring Agents that provide information on the status of the simulation programs itself. The “Simulation Supervisor” can now decide if the simulation should be stopped or if it is still in good condition.

**Core:** The Core offers basic functionality for all agents that participate in a distributed monitoring scenario. All Agents register with the Core in order to provide information about themselves such as type, data that is produced or consumed. The Core must not be necessarily a single central entity.

**Communication protocols:** GeneSyS defines an XML based communication protocol for Service Messages and Data Messages. Service Messages are for example used to subscribe to a set of data produced by a Producing Agent, register as Producing Agent for specific set of data at the Core or query for a Agent. All data sets are specified using XML Schema and are therefore of a very flexible nature and can be validated on the fly. Within the current prototype SOAP-RPC is used for transmitting the XML messages but it is very likely that SOAP-Messaging will be used if it reaches a wider acceptance and interoperability.

4. Examples of GeneSyS

The GeneSyS project defines a number of Validation Scenarios to prove the viability of the GeneSyS concept. PDR is one such scenario that simulates a contingency situation during important phase of spacecraft development, called Preliminary Design Review. PDR involves several Distributed Engineering tools, including a distributed document repository and a video conferencing tool. GeneSyS helps to maintain involved systems, applications and monitor entire workflow of the review process involving more than 100 participants.

The Document repository, containing review material is called the Engineering Database (EDB) with a web front-end to access the data. This front-end is implemented using the Tomcat servlet engine. GeneSyS agents integrated with EDB allow to monitor on-line users activities, to log them and to gather statistics. The Tomcat agent provides both system level (Java VM) and server specific (Tomcat application) monitoring information. System level data include the number of threads in the VM or the actual memory utilization, while the Tomcat server specific monitorable items include the number of servlet contexts, containers, servers and requests.

The PDR scenario includes a video-conferencing facility. The GeneSyS core and specific agents make it possible to track the availability, activity of users and possible loss of their data.

The agents (Tomcat & Vido Conferencing) and their associated consoles for displaying monitoring information are all connected via the GeneSyS core and communicating via SOAP embedded GeneSyS messages.

Another validation scenario of Genesys project aims at the monitoring of remote web servers and web based on-line services. MTA SZTAKI project partner maintains dozens of web servers within the governmental sphere day by day. This maintenance requires continuous attention to the operating system resources used, as well as the maintenance of server configurations and content developments. Operators need to check the QoS parameters of these services regularly (hit and visit counts, response times, etc.). Many occasions these services are implemented using multi-tier architectures thus a simple monitoring of a single entity is not possible. Genesys concept and architecture will be introduced in this context for the monitoring purposes. Actual validation tests are planned within the next phase of Genesys project.

5. References

