Name of the Workshop: ECOOP 2004
Name of Session: Poster

SICSDA: an Adaptive Configurable Distributed Software Architecture Applied to Satellite Control Missions

Adriana Cursino Thomé[1], Mauricio G. V. Ferreira[2], João Bosco S. Cunha[3]

[1] Positivo Universitarian Center (UNICENP)
Rua Prof. Pedro Viriato Parigot de Souza, 5300 - Curitiba - PR - CEP 81280-330 - Brazil
adriana.thomé@unicenp.br
[2] National Institute for Space Research (INPE)
Av. dos Astronautas, 1758 - São José dos Campos - SP - CEP 12227-010 - Brazil
mauricio@ccs.inpe.br
[3] School of Engineering of Itajubá (UNIFEI) – Brazil
Caixa Postal 50 – CEP 37500-903 - Itajubá – MG - Brazil
bosco@unifei.edu.br

Abstract. This work was proposed as a Doctoral Thesis of Applied Computing Course at Brazilian National Institute for Space Research (INPE) and is a ongoing work. The main purpose of SICSDA is to allow that satellites control can be done through the same set of machines, enabling the choice of which one of the satellites is desirable to be monitored in a determined piece of time. Another important fact is the necessity of having an architecture that allows that a new mission could be settled without the necessity of creating a specific software system to the satellite being launched, minimizing the effort necessary to adapt the system to this new requirement. Besides, it is desirable that domain specialists and software developers are able to configure, if necessary, attributes and business rules to the satellites already launched, adding new elements to business domain without the necessity of extra codification.

1 Introduction

SICSDA architecture models the satellite control application based on adaptive object models (AOMs). Therefore, in SICSDA architecture, problem domain objects, for example, telemetry, telecommand, ranging; instead of being located in the code that implements the application, will be implemented in an object oriented database to be interpreted and instantiated at runtime. It means that the system will have a generic code that will be able to handle the different satellite object models too.

SICSDA needs to be fault tolerant, so this architecture will be distributed, and the functionalities offered by the application; for example, telemetry visualization and the sending of telecommands, can be distributed into a network pre-defined domain. It means that application objects can be instantiated in different machines of the network, causing a distribution of the system code. The system charge distribution service will define these objects location, that is, in which machine of the network they will be available. It means that each machine in the network will be able to have a different view of the metadata stored in the database. A "view", in this context, is the piece of the adaptive object model that will be instantiated in that machine. Therefore, a given machine in the network will only have access to a part of the object model, and it will instantiate only the objects pre-defined and allocated to it by the system charge service.

In addition, SICSDA architecture suggests the use of distributed databases through the replication of the database for each node of the network. It is important to remark that it will be necessary to have mechanisms that help the control and the storage of metadata recording, and mechanisms that help the versions control.

SICSDA architecture will be configurable because it will be able to: **(1)** allow domain specialists and developers to re-configure the database to handle new classes, by creating these classes, their attributes and methods at runtime; and **(2)** allow, at runtime, the choice of which satellite metadata to use, causing a new object model instantiation in the generic code each time that this kind of context change is asked by the user, that is, each time the user wants to control a different satellite.

SICSDA architecture will be dynamic or adaptive because it will be able to handle possible changes in the business domain, allowing it to match the evolution of the business requirements and to adapt itself to the users necessities. Following, is showed the elements that compound SICSDA:

- **Satellite Control Application:** contains the objects of the software that controls the satellites (telemetry, telecommand, ranging, etc).
- **Persistence Service:** responsible for storing and retrieving the metadata from the database.
- **Configuration Service:** responsible for maintaining the metadata and the views of the distributed objects. It will be implemented through a generic code.
- **Connection Service:** responsible for locating an object in the network.
- **Satellite Simulator:** software that simulates the interaction with satellites.
- **Charge Service:** responsible for charging the satellite metadata. It will be implemented through a generic code.
- **User Service:** responsible for offering to the user of the satellite control system an appropriate interface.

Thus, the charge service will charge the satellite metadata corresponding to the satellite that were chosen to be monitored in a given period of time. To do that the charge service will activate the persistence service, which will be responsible for recovering the metadata from the database. The charge service is going to instantiated the objects recovered from the database, creating the objects of the chosen satellite application, and constructing, in this way, the real object model of the application.
The user can interact with the application through the user service. If the user wants to initiate the monitoring of a satellite, the user service will activate the charge service to charge the appropriate objects from the database. The data from satellites can be obtained through the satellite simulator.
The application objects will be distributed and can be accessed through middleware. The connection service will be responsible to locate an object in the system. If the satellites metadata need to be changed to reflect changes in the domain, the user responsible for the configuration of the system can do these changes using an appropriated interface. When this kind of change occurs the configuration service will be activated. The configuration service, as a consequence, will activate the persistence service, that will be responsible for reflecting in the database these changes.

2 SICSDA Development Issues

As AOMs have been pointed as an evolution of MDA (Model Driven Architecture), one is trying to incorporate its concepts to ones proposed in MDA [2]. In this way, one is going to produce, as the MDA suggests, platform independent models (PIMs) and platform dependent models (PSMs) to represent the satellite control system, trying to adapt these concepts to dynamic systems.
When one is developing a software system is necessary to have a deep knowledge about the business domain, and this must not be different if the development involves a dynamic system. Although the intention is to produce a generic model to the satellites being controlled, it is necessary to model each satellite in separate to capture its own features. Therefore, one is building a platform independent model (PIM) for each satellite, called specific domain PIMs. These domain PIMs will be implemented as metadata in the database, as they represent the information about each satellite.
From the specific domain PIMs, will be derived a generic model that will represent all the satellites. This model is still independent of platform, and for that reason, it will be a PIM, too. However, this PIM will be also independent of a specific satellite domain, so one called this PIM, generic PIM. The generic PIM will be obtained from mapping rules that apply the use of TypeObject pattern, Property pattern, Strategy pattern, showed in [1].
After that, one will obtain the dependent platform model (PSM) from the generic PIM. The PSM will be implemented through a programming language, and is going to work as a “machine” responsible to instantiate the objects of the application. That is, this “machine” will be able to instantiate specific domain PIMs that are in the database, according through the satellite users desire to control in a specific period of time.
Besides, this “machine” will be able to reflect changes made in the domain in the metadata that is stored in the database (specific domain PIMs). This means that, if the business domain changes, these changes can be incorporated to specific domain PIM

through PSM. As a consequence, it is not necessary to do the generation code process anytime changes happen in the business domain, as the changes will be done in the metadata stored in the database by the generic code.
Mapping the specific domain PIM to generic PIM will be done manually, accordingly to the patterns specified in [1]. Mapping the generic PIM to PSM can be done using existing tools. SICSDA architecture is being implemented using Java as the programming language, an object-oriented database to store the satellites metadata, and J2EE as the distribution framework.

3 Expected Results

When this work is finished, one hopes to have elaborated an adaptive configurable distributed architecture for the development of satellite control systems that are based on adaptive object models.
Despite of the fact that many aspects point out that, at least in the beginning, architectures based on AOMs require more effort to be built than the traditional architectures, this work intends to take a significant step towards systems reusability, since adaptive systems have the property of matching business requirements evolution, and at the same time, they have a general-purpose architecture, what make them highly configurable and adaptable.
Another important step can be taken in the direction of systems maintainability. This is possible because the effort to make changes in the system can be minimized, since changes in the code can be minimized substantially. In addition, with this kind of architecture it is possible to allow domain specialists to make some changes in the system by themselves, increasing the maintainability and minimizing developer intervention in the system evolution. Besides, one also hopes to take an important step towards economy, since future missions will be able to use all the hardware and software investment that has been made for previous missions.
The work proposed here has a multidisciplinary nature, since it is a binding of some features that have been explored independently in different research areas until now. This way, one hopes to be collaborating with to the advance of computational research in Brazil, and to be contributing to the success of the Brazilian space mission, by proposing a new alternative architecture for the development of satellites control software, and mainly, by proposing a new approach for the development of adaptive systems.

References

1. Johnson, R.; Yoder, J. W. The adaptive object-model architectural style. IEEE/IFIP Conference on Software Architecture 2002 (WICSA3´02). Canada. August of 2002.
2. Poole, J. D. Model Driven Architecture: vision, standards and emerging technologies. ECOOP´2001 Workshop on Metamodeling and Adaptive Object Models. April of 2001.