Abstract

Currently, the telecommunications management systems are powerful, however, maintain centralized and proprietary platforms, limiting this way, the growth of these telecommunications networks. This factor generates the need for scalable and open systems, stimulating the sectors of the telecommunications area to seek for solutions for the problem in distributed, heterogeneous and totally scalable environments. The recent advances in the area of distributed objects, has created many positive expectations for the development of client/server systems which use, as if they were local, resources available through the network. This way, the objective of this work consists in the experimental evaluation of the potential of the main CORBA services in conjunction with the Java language, through the necessary requirements for the main functionalities proposed by the TMN architecture. For the implementation of the telecommunication services with the reusability of codes we use computational reflection with meta-classes substituting the manager and classes substituting the agents

Keywords: Distributed Systems, Distributed Objects, Computational Reflection, CORBA, JAVA, TMN

1. Introduction

In the last decades, we have seen the results of one of the greatest revolutions in the history of humankind: the information revolution. The Information Technology developments penetrated the large organizations, supplying solutions in a complex and highly competitive business world. The argument in favor of investment in information technology will be simple: the guarantee of a competitive advantage.

∗ This work is part of the AUTOGERE Project funded by CNPq - Brazilian National Research Council.
This work intends to propose a telecommunications services management model and offer the telecommunication companies a proposal to interact their services with information technology, that is, integrate them with Internet and Intranet technologies in a distributed computer environment. Supplying them with management of services will further open the way for new business and other services, thus making them more competitive in the new business environment. To reach our objective, we will implement of service management objects using the computational reflection paradigm with the JAVA language [GOS95] and the CORBA environment [OMG91][OMG96]. The recent advances in the area of distributed objects has created many positive expectations for the development of client/server systems which use, as if they were local, resources available through the network. The CORBA architecture is built with a set of basic and network objects, which are reliable and act as an intermediate software level, enabling distributed management applications to be built in heterogeneous environments, independent of the details and the characteristic problems of the distribution. The characteristics of the main CORBA services are similar to the functionalities proposed for TMN. Within our objective, this work also consists in the experimental evaluation the potential of the main CORBA services in conjunction with the Java language. Our service management model considers the necessary requirements for the main functionalities proposed by the TMN architecture, for implementing the telecommunication services with the reusability of codes by using computational reflection. This implementation technique has as basic function to build a meta-object to monitor how other objects react to messages, enabling intervention in the computational state. This intervention is the essence of dynamic reflective computation, conducted in a transparent manner, in the sense of collecting and registering information on the process in execution. For example, performance statistics, debugging information and monitoring of the execution, with the purpose of modifying the course of the process which is under execution.

The initial goal consists in simulating services offered by a telecommunications company using Internet, generating manageable information bases (MIBs) that will make information available, so that we can implement the management functionalities using computational reflection. These data bases should also provide information to various telecommunications company administrative segments, so that may by means of this information, companies may consult, generate graphs and statistics, workflow and decision-making, using existing computational tools. In the follow, the section 2 approches the telecommunication
Service Management; section 3 explains the Computational Reflection in the Object Model; section 4 shows a model specification for service management and, finally, in section 5 we mention some conclusions.

2. Telecommunication Services Management

In telecommunications, services are what the system resources offer to the enterprise’s clients [BAR96]. For example, teleconferencing, telecards, electronic mail, directory assistance, phone bill, and other services. Compared to the network elements, the telecommunication services have the following characteristics:

- The services are distributed, while the network elements may not be;
- Services are implemented over the network elements;
- There are always less services than network elements;
- The services are heterogeneous while the network elements are not;
- The services are more complex when compared with network elements. In many cases, the services involve people as part of the processes;
- Services are dynamic while network elements are static. New services frequently need to be created, implemented and made quickly available [KON96].

2.1 Services Management

The administration of the functions in the management of services has different characteristics and a great scope of application when compared to network management. Service management is more interested in supplying services, such as:

- How to supply a service to the subscriber/ client;
- To determine which functions need to be managed to have the service quality foreseen in a contract;
- How to define an information model, so that the same may preserve a unique and consistent vision for different services users;
- How a service is managed and implemented;
- How to manage services by means of telecommunications jurisdictions which may have different systems of services offered, price policies, commercial and administrative environments and a different technological structure.
• How to manage work-force and the flow of necessary tasks for the services;
• What is the relation between services and networks management;

The management of TMN networks based on OSI was designed to manage a great deal of simple objects. This may not be sufficient to provide a complete solution for the management of services [KONG96]. For example, the facilities of objects modeling currently used (for example GDMO) are not sufficient to model objects with services management functions. So that one may quickly control the changes in the services and in the interoperability, the need was shown for new technologies.

2.2 Necessities for Services Management

Necessities shared by Distributed Systems:
• Distribution;
• Scalability;
• Heterogeneous characteristics;
• Common Services.

Special necessities for the Management of Services:
• Consistency;
• Support for the Integration of Services;
• Interoperability.

2.3 Requirements of the Services Management

Services Management applications are necessary to administer geographically dispersed services with heterogeneous computers and very complex telecommunication services. The purpose of the management process is to access and modify resources in a transparent, consistent and reliable manner. It is important for the industry to develop new technologies to guarantee that the resources of the global network may be shared and managed by applications in an efficient and consistent manner. Telecommunication companies differ one from another in their management of and kind of services. The greater needs for the management of services are:
• Distribution – Management of telecommunication services, which are geographically dispersed;
• Scalability – The need for a scalable platform due to the fact that the resources, information, services and networks to be managed are always growing.

• Heterogeneous Nature - The diversity of resources, in computation and human processes requires support for a heterogeneous system in the management realm.

• Consistency - The consistency in the data is an important point for the success in the business of the telecommunication companies and necessary for a complete management of distributed telecommunication services.

• Support for Integration of Services - Functions for management of services are necessary so that the new services may be easily introduced and managed; and where services exist, may be integrated.

• Interoperability - Different and business administrative domains should conduct interoperability integrating the management of services. This management domain normally has different enterprise policies and business environments, a subjacent technological architecture and organizational infrastructure.

As the proposal is only a simple abstract simulation without a commitment with the real world, and also aims at an academic clarity, (although its applicability is possible), the application consists of two sub-systems:

• A managed part, which simulates the service provided (base-level);

• The manager part, which conducts some supervisory activities over the first (meta-level).

As previously mentioned the manager application should seek to touch on classic aspects of distribution and reusability that may permit the experiments with distribution platforms, as in the case of CORBA, with implementation in the JAVA language.

3. Computational Reflection in the Objects Model

The concept of computational reflection in the objects model was introduced in 1987 by Patti Maes [MAE87a,MAE87b]. In the concept of Maes, computational reflection is the activity executed by a computational system when it conducts operations upon (and possible affecting) its own computations. A computational system is understood as a system based on a computer, with the objective of conducting computations and supplying information in a particular realm of application. In this concept, reflection is
defined as a form of introspection, in which a system attempts to make conclusions about its own computations, which may be affected later on. In summary, by the term computational reflection we understand:

**Definition:** Computational Reflection [FER89] is every activity of a computer system conducted on itself, and in a separate manner from the computations underway, with the objective of solving its own problems and obtain information about its computations. In this last definition the concept of levels of computation is introduced, in which the meaning is very close to the physical significance of reflection, because the occurrence of an event in the execution process reflects itself in a superior computational level (the meta-level).

The computational reflection defines a new software architecture. This reflective architecture is composed of a meta-level, where the structure of data and actions to be conducted on the system-object are located in the base-level. This way, in a reflective architecture, a computational system maintains two inter-related components [LIS97] [ANC95]:

- an **subsystem-object**, which performs computations upon a domain which is external to the system;
- a **reflective subsystem**, which performs computations upon a system-object.

The separation of domains is the most attractive aspect of reflective architecture, not only for motivating the reusability, but mainly for allowing the application programmer to concentrate on the solution to the programming problem specific to the application domain. Reflection has been adopted to express non-functional proprieties of the system, such as reliability and security, in a independent manner from the application domain [LIS97].

The multiple meanings of the term reflection require special insight in this work: the first meaning has to do with the computational reflection objective, which is the capacity of a system to act upon itself and not on what the system should produce, conducting deductions and computations on internal data of the system itself. The second meaning relates more directly with a form of computational reflection implementation in the objects model: as a message in sent to an object, it is diverted to its corresponding meta-objective, and, as a result of this reflection, begins to conduct computations at the meta-level.
In Figure 1 is shown the schematic form of a reflective architecture, showing that the actions in the meta-level are executed upon data that represent information about the base-level program, while the latter executes actions upon its own data, with the purpose of attending the users of its services.

Figure 2 shows a proposal of a telecommunication service management model, which may be used and re-utilized for any service offered by telecommunication companies.

4. Model Specification
The initial task of this work consists in the development of objects (services), capable of performing computations on their own domain (agent).

The service management model presented in figure 2 shows the possibility of flexible computations, capable of altering and dynamically adapting components of the system. This corresponds to the capacity of the system to interrupt the execution process, perform computations, be capable of generating failure of installed components [ANC95], and beyond this, make information available pertinent to the behavior of the same during the course of time. Network and business administrators have the conditions to access this information and be able to infer, for instance, on statistics, auditing, observe the evolution of the execution process in a given period of time, and return to the execution level, translating the impact of the decisions.

These objects (Figures 2 and 4) shall be specified in CORBA, which has established a standard for the exchange of services among distributed objects, and implemented in JAVA [CUR97], being a language adequate to the heterogeneous nature of the system. Therefore, in concept, JAVA maintains reflective characteristics, where information on program objects are supplied at run-time by its environment and made available on the Internet, for permitting global availability of resources, as shown in Figure 3 [VOG97][ORL97][UMA97].

![Figure 3 - Invocation of a JAVA object (service) using CORBA.](image-url)
4.1 Description of the Objects

Telecommunication equipment address provider: This component implements the address component, which allows the equipment to know where they are located and where they should connect. This artifice is necessary due to the need to explicitly address processes, which are running in a network environment, so it can access them. The location of this provider is only one, and is known by all the equipment.

Telecommunication Equipment: This is the meta-class from which we inherit the classes that make-up the three types of telecommunication equipment which will be simulated on the system.

- **User terminal**: Component responsible for the simulation of a telephone terminal; and will be simulated by an HTML page with JAVA calls; through this component, a user can connect and exchange messages with other terminals or conduct the programming of services.

- **Switching center**: Component responsible for the connection between equipment, which needs to communicate. All the equipment (user terminals and service provider) connect to the switching center, whose function is to establish temporary circuits between 1 or n terminals and the service provider.
• **Service provider**: This system object simulates a piece of equipment or software which provides the service, as previously described. It shall be programmed by users through the user terminals, and then shall execute the services for the solicitors.

**Equipment Agents**: This is the meta-level from which are inherited the agents specific to each of the equipment components of the simulation of telecommunication networks (user terminal agent, central switching agent, and the service provider agent).

**Equipment Manager**: This is the meta-level from which are inherited the specific managers from each of the equipment components of the simulation of telecommunication networks (user terminal manager, central switching manager, and the service provider manager).

**Repairs Manager**: Responsible for the maintenance of the system; in case an error event occurs, indicated by one of the equipment managers, or a maintenance call, indicated by the repairman supervisor, the repairs manager should file a maintenance report, (super-class or meta-level). This consists of an entity, which contains information pertinent to the proceeding of the maintenance of one of the telecommunications simulation equipment of the network.

A maintenance report may be of two types (hereditary classes):

• preventive maintenance report;

• corrective maintenance report.

**Maintenance Entity**: This entity is responsible for the interaction of human maintenance with the system. By means of this element, a repairman may verify maintenance reports given to him and update them, according to the proceeding of the equipment maintenance and the technician may also request a management operation by means of this component.

**Technician Supervisor Entity**: Component responsible for the interaction between the supervisor of the human technicians with the system. By this element, one may request maintenance of the equipment and digitize data about the available technicians for the maintenance of the system.

**Event Report**: This component receives requests to register events pertinent to the functioning of the system. An event (meta-level) may be of the following types (hereditary classes):

• **normal event**, which is obtained from the managers of the equipment by means of notifications or;
• maintenance event, which is obtained from the equipment manager. A maintenance event may be of the following types: preventive maintenance or corrective maintenance, analogous to as described previously for the preventive and corrective maintenance entities;

Monitor: Entity responsible for a human network operator; by means of this component one can obtain information and statistics regarding the functioning of the system.

5. Conclusions

This work does not intend to faithfully simulate the characteristics, the behavior and the management of the real equipment of the telecommunication environment, which provides services to the users, and much less to provide such services. The proposed telecommunication service management model consists in focusing on some points which are relevant to the distributed applications in combination with heterogeneous environments (some times proprietary) and so this way may promote sufficient and necessary subsidies for the evaluation. This work seeks for academic clarity even though its applicability is possible.

The manager application should seek to touch on distribution aspects, which may permit the study and experiences with platforms such as CORBA. Such experiences should permit the verification of the possibilities, facilities and difficulties related to the interaction between CORBA and JAVA and also TMN, as well as code reusability with the use of computational reflection. In this manner here is proposed a model for service management, which is flexible enough for new implementations.

So that such goals may be reached, the intention is to simulate a service and a management application. This application should conduct simple management activities upon the simulation of the service, using reflexive characteristics. This way, it is possible to separate the management aspects from the construction of the service applications, trying to obtain Computational Reflection, and also offering the conditions to use and test the main characteristics of CORBA and JAVA.

6. References

[ANC95] Ancona, M.; Doderoa, G.; Gianuzzi, V.
Reflective Architecture for reusable Fault-Tolerant Software
Latin American Conference in Informatics, 21., 1995, Canela, BRAZIL.

[BAR96] Barros, Claudius D’Artagnan C.
Excelência em Serviços, Questão de Sobrevivência no Mercado

[CUR97] D.Curtis
Java, RMI and CORBA
OMG White paper - 1997

[FER89] Ferber J.
Computational Reflection in Class Based Object-Oriented Language.

[GOS95] Gosling, James & Mcgilton, Henry
The Java Language Environment - A White Paper

[GRA89] Graube, N.

[KON96] Kong Qinzheng & Graham Chen
Integrating CORBA and TMN Environment
CITR Pty Ltd. Queensland, Australia 1996.

[LIS97] Lisboa, Maria L. B.
Computational Reflection in an Object Model
Federal University of Rio Grande do Sul

[MAE87a] Maes P.

[MAE87b] Maes P.

[MAR93b] Martin, Roger
Changing the Mind of The Corporation

[OMG 91] Object Management Group

[OMG 96] Object Management Group
The Common Object Request Broker 2.0/IIOP specification Revision 2.0, OMG Document, August, 1996.

[OMG 97] Object Management Group
Java Language Mapping RFP

The Essential Distributed Objects - Survival Guide
John Wiley & Sons, Inc, 1996

[ORL97] R. Orlafi, D. Harkey
Client/Server Programming with JAVA and CORBA
John Wiley & Sons, Inc, 1997

[SUN 96] Sun Microsystems,
Java Remote Method Invocation Specification
Microsystem Computer Corporation, Revision 0.9, Draft may 1996.

[UMA97] Amjad Umar
Object-Oriented Client/Server Internet Environments

[VOG97] Andreas Vogel & Keith Duddy
Java Programming with CORBA