A Web Based Inventory System for Telecommunication Networks

Manoel Camillo Penna e Osvaldo Doederlin Pinalli

Abstract - **Configuration of telecommunication network is a complex problem, and network management systems (NMS) are not able to address the emerging issues in a consistent way. The point is, the NMS of a particular vendor handles only its own network. Integration is necessary, providing a whole vision of the network. Although the problem has been addressed by commercial inventory systems, there is very little research work on the theme. Main telecommunication operators are handling the problem with big inventory projects, which are expensive and not afforded by smaller companies. In this paper we discuss an alternative solution that allows inventory creation and which has been applied to a local company in Brazil located at state of Paraná.**

Index Terms **— Telecommunication Inventory, Telecommunication Management Network, Network Management System.**

I. INTRODUCTION

Management of telecommunication networks is a known complex problem, which has been addressed by many research teams, telecommunication equipment vendors, information systems companies and standardization organizations and forums. Particularly, ITU-T is a major player on the subject, and has defined a very important set of concepts, structured in a comprehensive architecture named TMN (Telecommunication Management Network) [1]. One main concept defined on TMN is the layer structuring of the architecture, allowing one to focus on pertinent concepts when handling related problems. This layered architecture introduces four management layers: Network Element Management Layer, Network Management Layer, Service Management Layer and Business Management Layer. This paper focuses on problems arising at Network Management Layer.

The Network Management Layer addresses the issues involved on building a consistent view of the whole network, allowing multiple aspects of management to be seen under this perspective. In this way, fault network management, performance network management, and configuration network management, are considered under the network viewpoint. Configuration management is particularly relevant at the Network Management Layer, because it is at this level that an end-to-end view for the telecommunication circuits should be provided. Thus, many important and non-trivial configuration management functions are necessary at this level, for example, path setup, path restoration and path control.

A possible generic architecture for configuration management at network management layer is depicted on Figure 1. Typically, a set of Network Element Management Functions (NEMF) is provided for interacting with Network Elements, fulfilling the needs of NE configuration and monitoring. They control the interactions between an NE and the Management System, but provide a segmented view of the telecommunication network, because of their nature.

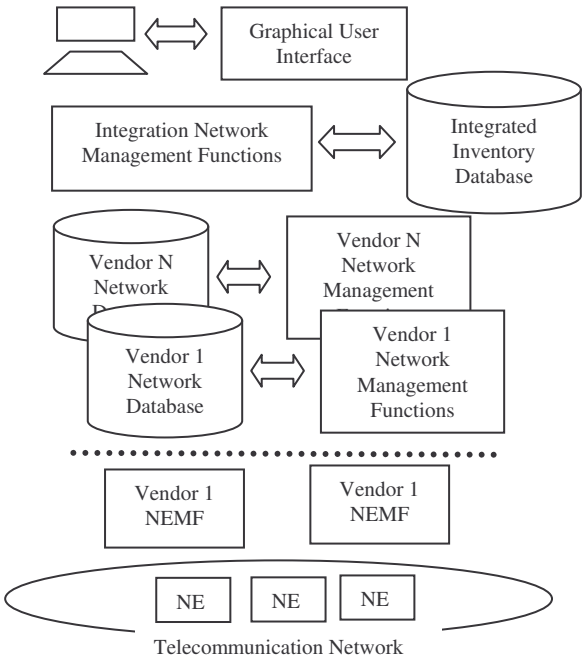

Figure 1. Generic architecture for network management layer

At Network Management Layer, another set of functions (NMF) provides the end-to-end view related to the managed network. They use the NEMF to interact with NEs and rely on a Network Database for storing the network data. The problem here is that the network information and network

Manoel Camillo Penna, Programa de Pós-Graduação em Informática Aplicada da PUCPR, Curitiba, Brasil. Osvaldo Doederlin Pinali. Visionnaire Informática. Diretoria de Tecnologia. E-mails: penna@ppgia.pucpr.br, osvaldo@visionnaire.com.br

management functions are usually proprietary, and not open. The difficulties arise when the telecommunication network is multi-vendor, and there is no way to provide an end-to-to-view. For dealing with this problem, the generic architecture suggests a set of integration network management functions (INMF), and an integrated network database for providing a multi-vendor end-to-end view. A set o interfaces between each NMF and INMF should be defined. The generic architecture also shows the need of Graphical User Interfaces (GUI), to make easy the interaction between users and network information.

This paper describes a system conceived under these principles, putting the focus on its Web based implementation architecture. The implementation architecture puts in perspective the fitness of object-oriented techniques for implementing the on memory network management functions, combined with a relational data model for storing network inventory data. Moreover, the architecture emphasizes the use of Web technology for presentation purpose, resulting in a robust and scalable environment for solving this complex problem. This paper is organized as follows. The next section introduces the general concepts and discusses the related work. The third section, presents the system design. The object model is fully discussed and the architecture of the system is presented in detail. Finally section five concludes de paper.

II. Related Work

A. Architecture of Telecommunications Inventory

Management of telecommunication networks has been addressed since the operation of earliest networks. The definition of TMN concept by ITU-T [1] was particularly influencing as a systematic approach for solving the problem. The ITU-T has recognized the magnitude of the involved issues, and has defined a general architecture for handling them. As said before, one important contribution of ITU-T architecture was the proposition of a layered architecture.

Another important contribution made by ITU-T was the use of object oriented techniques for dealing with the information model, and the creation of a set of communication protocols for dealing with message exchange [2]. A distribution architecture based on manager-agent concepts was also proposed [3], and together with the protocol stack, it has guided the communication and distributed computing technology on earliest TMN compliant industry implementation. Today the IT industry is developing a distributed computing technology based on object distribution. This provides a programmatic interface to distribute software modules, similar to the constructs used by programmers within monolithic systems. Most relevant are the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA) [4], Sun Microsystems Enterprise Java Beans (EJB) [5], and Microsoft Distributed Common Object Model (COM+) [6]. There is now a trend for telecommunications management industry to migrate to object technology for purposes of cost and support.

Other major efforts are being taken on the definition of common application architecture for telecommunication management. Most notable is the TeleManagement Forum (TMForum) Telecom Operations Map (eTOM) [7], delivers a business process model or framework for use by service providers and their suppliers within the telecommunications industry. It describes the enterprise processes required by a service provider and analyzes them at multiple levels of detail, according to their significance and priority for the business. It is part of the NGOSS toolkit, a comprehensive, integrated framework for developing and deploying operational and business support systems and software. NGOSS principles and tools apply to service providers, software suppliers and system integrators, providing *de facto* standards for business process mapping and automation.

B. Network Model

There are few efforts running in order to define an object model for telecommunication networks. The Interim European Telecommunication Standard [8] is an important example, where we can found a managed object library, designed under the network perspective. The proposed model abstracts the main aspects from existing telecommunication resources necessaries for management at network level, including equipment, networks and services.

Another important network model is Alcatel/Telefonica Qnn, which defines the requirements that must be matched by managers willing to connect to a network manager. The focus at network layer is connectivity management between network elements and between sub-networks. It is partially derived from [8], with extensions to some object class (e.g. extensions to Termination Point object class), and with new object class definitions (e.g. Domain, Area, SubNetwok, and Protection).

C. NMS Interoperability

One important issue that arises at Network Management Layer is the possible need for integrating multiple vendor infrastructures. It is common to find in telecommunications companies, a large number and diversity of equipment in operation that does not have interoperable interfaces. Although the integration of telecommunications network management is a known problem, most of the solutions that are still in use are based on proprietary products, made up of hardware and/or software specifically oriented to combination of products that already exist in the network.

A possible approach to solve the interoperation problem is the confinement of the solution of management to one single vendor, adopting its conventions and methodologies to set up and manage networks. It may be said that this solution, adopted without many alternatives, can lead to a scenario of total dependence on the resources of a specific vendor.

Another approach is the use of standards for handling the heterogeneity problem. The introduction of standard

networks management solutions have come up with the inclusion of "Management Framework" [9] in the reference model ISO/OSI (International Organization for Standardization Open Systems Interconnection). One of the concepts found within this recommendation is the management environment, which is in general made up of two or more open systems that cooperate to monitor and control the network resources.

After 1985, ITU-T has focused on TMN architecture. The initial choice for standard interfaces in TMN has introduced the Q3 concept, a set of communication protocols, known as CMIP, plus a common view of the structure of exchanged information, named MIB (Management Information Base). Besides Q3 effort the Internet industry proposed the SNMP protocol, the first non-proprietary management protocol, public and easily implemented, which enables the effective management of heterogeneous environments [10].

Other standardization options have attracted the interest of the telecommunications community, specially the CORBA standards [4], which involve all the necessary functionality for building the management systems distributed and oriented to objects. CORBA standard was developed when the technologies of telecommunications network management were already in a mature state. Advances on object technology led to a group of specifications that are more adequate to implement distributed objects. Another point that makes the CORBA standard interesting is that the products built according to those specifications aim at corporation systems, and consequently reach a wider market. The result is that, as the commercialization schedule is larger, the price of products based on this technology tends to be smaller than the price of the so-called Q3 management platforms.

In general, the implementation of an integrated management system does not aim to replace the already existing systems. The objective is to allow their interoperation and their integration with other existing or future systems, with some possible improvement in functionality. This is a complex task, which is worsen by the difficulty in "opening" the existing systems, and expanding them to interoperate with interfaces, protocols and already implemented mechanisms.

No matter the approach chosen, a definition of the information model for the integrated system becomes necessary. This is also is a complex task and the effort made by ITU-T for the production of information models for TMN cannot be discarded. Even though it is not always feasible, the model should contain, as much as possible, the information suggested by ITU-T, within the proposed structure, and should be described through the same syntax [2].

Nevertheless, the reutilization of a standard model, at anyone of the management levels previously mentioned, is not an ordinary task. The difficulty rests on mapping the functionality of the existing equipment in the suggested objects by international organizations. That mapping is crucial and characterizes the reutilization of the objects already defined.

The system described in this article is an application of configuration management that works at network management layer and supplies support for the operational planning of the provisioning activity. The basic functions are:

- Interface with the clients to hire new services;
- Construction and maintenance of a integrated network database;
- Objects and algorithms to define the trails involving equipment managed by distinct systems in the original management.

The system, named TIS (Telecom Inventory System), has been planned as a modular system, which might be changed in relation to its functionality through configurable parameters to satisfy its needs. It has a multi-vendor capacity, that is, it was developed so that its basis was an open architecture, in order to receive systems and applications from any vendor. The user's interface planned is friendly and easily accessed through Web architecture.

III. SYSTEM DESIGN

TIS information model is based on three main ITU/T specifications:

- M.3101 – Generic Network Information Model [2];
- X.721 – Structure of Management Information: Definition of Management Information [11].
- G.805 –Architecture of Transport Networks [12];

These standards provided a starting point, and we model the system's core technology components after ITU/T's models. Figure 2 is a block diagram of the entire system.

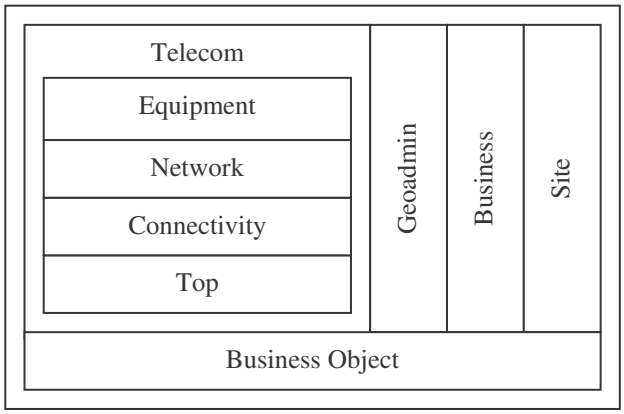

Figure 2. TIS core

The Telecom packages contains a rendering of concepts like circuits, links, connections, and ports; networks, sub networks, and network connections; managed objects and equipment. Some changes and additions were necessary to the standards, either to reduce complexity (removing unneeded features) or to make the model more concrete and complete. Specifically, there aren't standards to model the structure of transmission equipment, which is quite complex. Then, our Equipment package includes such details like sub racks and cards, speeds and timeslots, and ports.

A. *Equipment object model*

A major problem we faced while modeling the Equipment package was the diversity and complexity of transmission equipment – PDH, SDH, Radio, and Distributors are currently supported. Every model contains a specific layout, supporting different units arranged in different ways. A collection of object types hardwired for the equipment that our client was using when we started modeling could be already obsolete when the system got deployed.

The solution was abstracting the structure of such equipment, and coming up with a generic framework, which supports current and future equipment models. Figure 3 resumes the transmission equipment modeling.

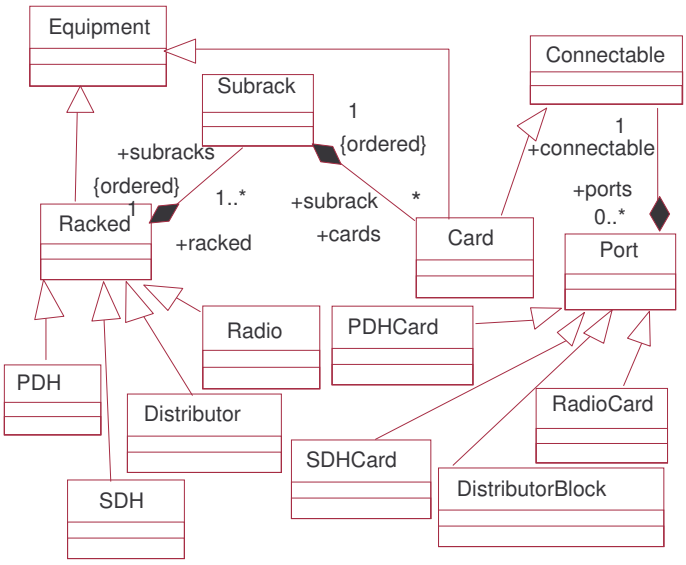

Figure 3. Equipment object

The fundamental ideas are:

- Transmission equipment contains *Cards*, and they are all *Racked* equipment – organized into *Sub racks*. (If some very simple equipment won't need sub racks, we put all cards in a single, default sub rack.).
- *Connectable* is any object having ports; so far, cards and cables (not shown) are connectable objects. The connectable object specifies the bit rate and directionality for all its ports; the Lines connecting ports will sub-allocate their bandwidth (by specifying timeslots or sections). A *Port* is any endpoint for a data signal: this includes ports of I/O cards, ends of cables, and connectors in distributors.
- One subclass of Racked and one subclass of *Card* are needed to support an entire technology. These subclasses handle technology-specific metadata (see later) and behavior. For example, a PDH Multiplexer contains I/O cards with real ports arranged in a simple list (p1..pN), while distributors have blocks containing connectors, which are arranged into bi-dimensional matrices and need attenuation data Both cases are handled uniformly.
- Equipments are built following meta-definitions previously fed into the system.

The meta-model is structurally parallel to the equipment model: there are Equipment, Connectable and Racked meta objects. The Equipment meta-objects define trivial items like names and category, compatibility with other equipment (which cards fit into which transmission equipment). The Racked meta object defines sub rack layout (how many sub racks, which ones are default, how many slots per sub rack, which cards are supported in each slot, etc.). The Connectable meta-object supports both I/O units and non-I/O (control, power, and other) units; for I/O cards, it defines the number of ports and supported transmission speeds, and differentiates line and tributary cards.

Before the system is put in production, it should be populated with metadata correspondent to the specific equipment being used by the client. TIS offers a GUI for this task, and the application's administrator is able to handle it. The system can be update to support new models of equipment without any reprogramming, and without any downtime. Any user with administrator privileges and knowledge about the equipment does the update online. This abstract framework also enables multi-vendor support.

B. *System architecture*

TIS was implemented as a "monolithic N-tier" application. The system is split into persistence (database), behavior (business objects) and user interface tiers, but they all run in a single process, in a centralized server. The Java™ platform was employed to implement the entire system.

The GUI is a dynamic HTML site, built with Java Servlet and JSP (Java Server Pages) technology. JSPs are HTML pages containing embedded Java code; these pages are automatically compiled into Servlets – Java classes that produce HTML pages. All processing happens in the server side, and clients only see pure HTML compatible with ordinary Web browsers.

This scheme is similar to other products (like Microsoft's ASP for VBScript/JScript, or PHP for PERL), but it supports Java. One important advantage of Servlets/JSPs is high performance: servlets, like other Java classes, are compiled to native code by high-performance Java Virtual Machines (JVMs), so they run faster than interpreted scripts. Additionally, if the business objects are also written in Java (like in TIS), the code inside servlets is free to manipulate them directly, without the overhead of any middleware or native interface. This setup – front-end code directly invoking the core app functionality – is typical of 2-tier, fat-client systems, and usually not recommended because code running in client sites should not be trusted and is cumbersome to manage. But servlets run in the server; all GUI is actually deployed in the server, so we found out that having the client code to obtain references to server objects and perform straight invocations on them provides unmatched

performance and is perfectly robust – as far as the server objects are properly encapsulated, and a safe language is employed. (Since early alpha stage testing, we have had thousands of errors caused by bugs, but never a single "cross-layer" corruption.) We also gain in simplicity: the same language is used for client and server, and they might share common behavior and data easily, and cross-layer refactoring is easy. The implementation architecture is depicted in figure 4 below.

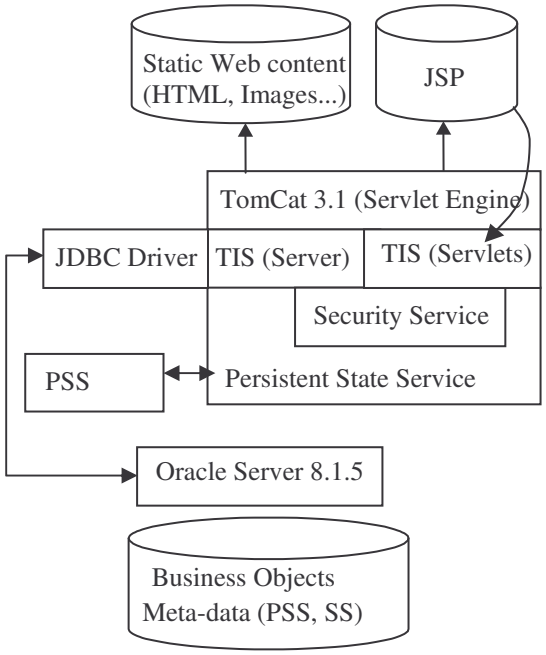

Figure 4. Implementation architecture

C. Administration and interoperability

The GUI, in pages restricted to administrators, support common administration tasks, such as managing users, equipment meta-data, equipment, and the business model. But the system offers an advanced, command line shell, which is part of the Persistent Service and connects to the business objects (of this application, or any other application using the PSS). An advanced administrator (more likely, a developer or support person) is able to fully manipulate all business objects; for example, all methods might be invoked. Some problems caused by bugs might be fixed without requiring the server to go offline, and many repetitive tasks are scriptable. This shell is also remotely accessible (as a webpage), enabling an interesting level of remote support work through simple HTTP access to the client site.

The system is planned to talk to multiple vendors' proprietary management systems, such as Marconi's MV-36 and MV-38, but this module is not yet implemented.

In addition to CORBA's IIOP, it's worth noticing that this application – like many other apps with a dynamic HTML user interface – supports HTTP as an open, interoperable protocol. Any program is free to open an HTTP connection to the server, and perform all tasks that users might execute through the GUI, by just sending similarly-formed GET requests. The format of such requests is stable because their decoding is actually supported by server objects; this helps keeping the client code (JSPs) simple and efficient. The webserver's response would be discarded, it is not interesting for clients other than a web browser, but this is easy to change if we structure result data with XML.

IV. Conclusion

The ITU/T standards provide standard, powerful modeling specifications, which we used successfully as the basis for a new network management application. We had to extend the model to support details of equipment available in the market and used by our clients, and we created a flexible meta-layer to support multiple vendors and equipment models seamlessly.

The implementation of this design effort relies not only on ITU/T's, but also on additional open standards – CORBA and Web technologies – to achieve qualities such as interoperability and portability. The Java platform was used for all implementation, using a setup that's aimed at performance and scalability.

The system described in this paper was fully implemented and is currently in production at a small telecommunication operator at Paraná state in Brazil.

References

[1] ITU-T, *Recommendation M.3020 – Principles of telecommunication management network*, 1996.
[2] ITU-T, *Recommendation M.3101 – Generic network model*, 1996.
[3] ITU-T, *Recommendation M.3020 – TMN interface specification methodology*, 1995.
[4] Object Management Group, *The common object request boker – architecture specification*, 1998.
[5] V. Matena and B. Stearns, *Applying enterprise java beans – based development for J2EE platform*, Addison Wesley, Boston, 423p., 2001.
[6] D. Box, *Essential COM*, Addison Wesley, New York, 386p., 1997.
[7] TM Forum, Enhanced telecom operations map: the business process framework for the information and communications service industry – GB921 v3.0, 2002.
[8] European Telecommunication Standard, *I-ETS-300-653 – Telecommunication management network: generic managed object class library for the network level view*, 1996.
[9] ITU-T, *Recommendation X.200 – OSI basic reference model*, 1995.
[10] D. Mauro and K. Schmidt, *Essential SNMP*, O'Reilly, New York, 300p., 2001.
[11] ITU-T, *Recommendation X.721 – Structure of management information: definition of management information*, 1992.
[12] ITU-T, *Recommendation X.805 – Architecture of transport networks*, 1995.