

MARCELO DUTRA ÖS

**UMA ARQUITETURA ABERTA PARA GERENCIAMENTO DE SET-
TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL**

**Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
título de Mestre em Engenharia.**

São Paulo
2006

MARCELO DUTRA ÖS

**UMA ARQUITETURA ABERTA PARA GERENCIAMENTO DE SET-
TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL**

**Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
título de Mestre em Engenharia.**

São Paulo
2006

MARCELO DUTRA ÖS

**UMA ARQUITETURA ABERTA PARA GERENCIAMENTO DE SET-
TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL**

**Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
título de Mestre em Engenharia.**

Área de Concentração:
Sistemas Digitais

Orientadora:
Profa. Dra. Graça Bressan

São Paulo
2006

FICHA CATALOGRÁFICA

Os, Dutra Marcelo

Uma Arquitetura aberta para gerenciamento de *set-top boxes* e serviços em redes de TV digital / M.D. Os – São Paulo, 2006, 200p.

Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Gerência de Redes 2.Televisão Interativa 3. Middleware 4. Arquitetura de Software I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

DEDICATÓRIA

Aos meus pais, José Aureliano e Maria do Carmo, por todo o apoio, compreensão e amor que me deram ao longo de minha vida, e que me fizeram chegar até este ponto.

AGRADECIMENTOS

À Profa. Dra. Graça Bressan , que me deu esta oportunidade e acreditou em meu trabalho. Agradeço pelos seus valiosos conselhos e por toda a orientação e atenção dispensada.

A toda equipe de TV digital do LARC, que sempre fizeram deste laboratório um ambiente produtivo e de contínua evolução.

Aos colegas do LARC, principalmente Raoni, Luiz e Daniel, pela contínua e frutífera troca de idéias e pelo apoio dispensado.

Aos meus amigos, pelas longas conversas e ótimos momentos juntos, que sempre me ajudaram a retomar as energias para terminar este trabalho.

À minha noiva, Ana Maria, por toda a compreensão, amor e carinho que me ofereceu durante todo este período de Mestrado.

RESUMO

A TV digital é hoje uma realidade em muitos países e no Brasil as pesquisas já estão atingindo um estágio avançado.

Muita expectativa gira em torno desta tecnologia, que oferece uma ampla variedade de serviços multimídia, abrindo as portas deste modo para criativas propostas e novas possibilidades de relacionamento com o usuário final.

Esta complexidade e as novas opções de serviços que podem ser oferecidos demandam, obrigatoriamente, uma arquitetura adequada de gerenciamento, capaz de controlar e otimizar os terminais desta rede bem como os serviços oferecidos pela mesma.

Neste trabalho são apresentados os requisitos funcionais e não funcionais necessários para a construção de um sistema genérico de gerenciamento de *set-top boxes* e serviços em TV digital, baseados no estudo de pesquisas semelhantes e nas particularidades deste ambiente. Também para este sistema genérico são detalhados todos os casos de uso de interesse.

Além disso, uma arquitetura híbrida é proposta para a solução deste problema, utilizando para tanto padrões abertos e considerando alguns cenários possíveis de implantação no mundo real. Nesta arquitetura, é adotado o modelo clássico de gerenciamento gerente-agente, através do uso de um servidor de gerência, responsável pelo armazenamento e análise de todos os dados de gerenciamento bem como pelo disparo de comandos e recebimento de respostas (síncronas e/ou assíncronas) dos agentes.

Tanto na definição deste sistema genérico como no desenho da arquitetura final, procurou-se na medida do possível adotar a utilização de princípios clássicos de gerenciamento que já são aplicados em várias arquiteturas existentes, inclusive de mercado. Deste modo, foi possível encontrar uma solução para um problema complexo através do uso de conceitos amplamente conhecidos, o que facilita o entendimento final.

ABSTRACT

Nowadays, digital TV systems are a reality in many countries and in Brazil the research in this field is reaching an advanced stage.

Much excitement is expected from this technology, which may offer a wide variety of multimedia services, opening the doors for creative proposals and new possibilities of relationships with the final user.

This complexity and the new options of services that can be offered demand a suitable architecture for management, which should be able to control and optimize the set-top boxes of this network as well as of the services offered by it.

In this work, the requirements for the building of a generic management system for the digital TV environment are presented, based in the study of similar research and considering the specifics involved. Besides that, a hybrid architecture based on open standards is proposed for the solution of this problem. In this architecture, it is adopted the classical manager-agent model, through the use of a management server, which is responsible for the storage and analysis of all the management data as well for the triggering of commands and receptions of answers and traps sent by the agents.

The conception and design of this architecture has followed as much as possible, classic principles of network management. With this approach, it was possible to find a simple solution to a very complex problem.

LISTA DE FIGURAS

| | |
|--|-----|
| Figura 2.1 Multiplexação de sinais em TV digital | 33 |
| Figura 2.2 Arquitetura de rede padrao para TV digital..... | 34 |
| Figura 2.3 O middleware dentro da arquitetura de uma set-top box | 37 |
| Figura 3.1 Camadas funcionais de gerenciamento TMN..... | 49 |
| Figura 3.2. Documento XML representando um objeto CIM..... | 55 |
| Figura 3.3 Arquitetura JMX | 60 |
| Figura 3.4 Gateway de serviços OSGi | 62 |
| Figura 3.5 Arquitetura OSGi | 63 |
| Figura 3.6 Ciclo de vida de um serviço | 66 |
| Figura 4.1 Arquitetura do sistema de monitoração TESEUS..... | 71 |
| Figura 4.2 Pilha de protocolos DAVIC..... | 75 |
| Figura 4.3 Pilha de protocolos de gerenciamento proposta por IBCoBN..... | 78 |
| Figura 4.4 Arquitetura de gerenciamento CABLEHOME..... | 80 |
| Figura 6.1 Diagramas de casos de uso | 98 |
| Figura 8.1 Arquitetura de Rede | 118 |
| Figura 8.2 Aplicação de gerenciamento relativa ao middleware | 124 |
| Figura 8.3 Diagrama de componentes - gerente | 128 |

| | |
|--|-----|
| Figura 8.4 Diagrama de componentes - agente | 133 |
| Figura 8.5 Diagrama de estados - agente | 134 |
| Figura 8.6 Diagrama de estados - gerente | 135 |
| Figura 8.7 Diagrama de classes pacote configuração - gerente..... | 139 |
| Figura 8.8 Diagrama de classes pacote contabilização - gerente | 140 |
| Figura 8.9 Diagrama de classes pacote falha - gerente..... | 141 |
| Figura 8.10 Diagrama de classes pacote desempenho - gerente | 142 |
| Figura 8.11 Diagrama de classes pacote segurança - gerente | 143 |
| Figura 8.12 Diagrama de classes pacote configuração - agente..... | 145 |
| Figura 8.13 Diagrama de classes pacote contabilização - agente..... | 146 |
| Figura 8.14 Diagrama de classes pacote falha - agente | 147 |
| Figura 8.15 Diagrama de classes pacote desempenho - agente..... | 148 |
| Figura 8.16 Diagrama de classes pacote segurança - agente..... | 149 |
| Figura 9.1 Topologia..... | 156 |
| Figura 9.2 Estado inicial - gerente | 158 |
| Figura 9.3 Estado administrador autenticado - gerente | 159 |
| Figura 9.4 Estado altera configuração set-top box - gerente..... | 159 |
| Figura 9.5 Knopflerfish - OSGi..... | 161 |

| | |
|---|-----|
| Figura 9.6 Console Knopflerfish - OSGi | 162 |
| Figura 9.7 CIMOM HTTP | 164 |
| Figura 9.8 Portas TCP – CIMOM HTTP | 165 |
| Figura 9.9 Visualização das classes e instâncias CIM..... | 166 |

LISTA DE TABELAS

| | |
|---|-----|
| Tabela 1.1 Escopo desta pesquisa..... | 22 |
| Tabela 4.1 Variáveis de estado e status da especificação ITU-T X.731..... | 73 |
| Tabela 4.2 Resumo comparativo de soluções de gerenciamento para TV digital..... | 83 |
| Tabela 5.1 Quality Function Deployment Inicial..... | 89 |
| Tabela 5.2 Quality Function Deployment Refinado..... | 100 |
| Tabela 8.1 Testes de verificação..... | 150 |
| Tabela 8.2 Estratégia de integração..... | 151 |
| Tabela 8.3 Testes de integração..... | 152 |
| Tabela 9.1 Configuração do agente no estudo de caso..... | 155 |
| Tabela 9.2 Configuração do gerente no estudo de caso..... | 155 |

ACRÔNIMOS E ABREVIATURAS

| Acrônimo / Abreviatura | Significado |
|-------------------------------|--|
| AAC | Advanced Audio Coding |
| API | Application Programming Interface |
| ARIB | Application Engine Platform for Digital Broadcasting |
| ATM | Asynchronous Transfer Mode |
| ATSC | Advanced Television Systems Committee |
| BNDES | Banco Nacional de Desenvolvimento |
| CIM | Common Information Model |
| CIMOM | Common Information Model Object Manager |
| CMIP | Common Management Information Protocol |
| CMP | CableHome Management Portal |
| CORBA | Common Request Broker Architecture |
| CPqD | Centro de Pesquisa e Desenvolvimento em Telecomunicações |
| CTP | CableHome Test Portal |
| DASE | Digital Television Application Software Environment |
| DAVIC | Digital Audio Video Council |
| DCE | Distributed Computing Environment |
| DHCP | Dynamic Host Configuration Protocol |
| DME | Distributed Management Environment |
| DMTF | Distributed Management Task-Force |
| DNS | Domain Name Service |
| DSM-CC | Digital Storage Media – Command and Control |
| DTVIA | Digital Television Industrial Alliance |
| DVB | Digital Video Broadcasting |
| ECMAScript | European Computer Manufacturers Association Script |
| EPG | Electronic Programme Guide |
| ETSI | European Telecommunications Standard Institute |
| GEM | Globally Executable MHP |

| | |
|----------|--|
| GSM | Global Systems Mobile |
| GUI | Graphical User Interface |
| HD | Hard-Disk |
| HDTV | High Definition Television |
| HTML | Hyper-Text Markup Language |
| HTTP | Hyper-Text Transfer Protocol |
| IBCoBN | Integrated Broadband Communication over Broadcast Networks |
| IDE | Integrated Development Environment |
| IEEE | Institution of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPTV | Televisão sobre IP |
| ISDB | Integrated Services Digital Broadcasting |
| ITU | International Telecommunications Union |
| J2EE | Java 2 Enterprise Edition |
| J2ME | Java 2 Mobile Edition |
| J2SE | Java 2 Standard Edition |
| JAXP | Java Application for XML Parsing |
| JMX | Java Management Extensions |
| JVM | Java Virtual Machine |
| LDAP | Lightweight Directory Active Protocol |
| MHP | Multimedia Home Platform |
| MIB | Management Information Base |
| MPEG | Moving Pictures Expert Group |
| MPEG2-TS | MPEG2-Transport Stream |
| MPLS | Multi-Protocol Label Switching |
| NTSC | National Television Systems Committee |
| OCAP | Open Cable Application Platform |
| OSF | Open Software Foundation |
| OSGi | Open Services Gateway Initiative |
| OSI | Open Systems Interconnection |
| PAT | Program Association Table |
| PCR | Program Clock Reference |

| | |
|----------|--|
| PID | Program Identifier |
| PMT | Program Map Table |
| PTS | Presentation Time Stamps |
| QFD | Quality Function Deployment |
| QoS | Qualidade de Serviço |
| RAM | Random Access Memory |
| RFC | Request for Comments |
| RFP | Requisição Formal de Proposta |
| RTCP | Real-Time Control Protocol |
| RTP | Real-Time Protocol |
| SBTVD | Sistema Brasileiro de TV Digital |
| SDTV | Standard Definition Television |
| SE-CMM | Systems Engineering – Capability Maturity Model |
| SI (DVB) | Service Information (DVB) |
| SNMP | Simple Network Management Protocol |
| SSL | Secure Socket Layer |
| STFC | Sistema Telefônico Fixo Comutado |
| TCP | Transport Control Protocol |
| TINA | Telecommunications Information Networking Architecture |
| TMN | Telecommunications Management Network |
| UDP | User Datagram Protocol |
| UML | Unified Modeling Language |
| UPnP | Universal Plug and Play |
| URL | Universal Resource Locator |
| VoD | Video on Demand |
| WBEM | Web-based Management Protocol |
| XML | eXtensible Markup Language |
| XSL | eXtensible Stylesheet Language |

SUMÁRIO

| | |
|--|-----------|
| 1. INTRODUÇÃO..... | 20 |
| 1.1 Objetivos e Motivação..... | 20 |
| 1.2 Escopo | 22 |
| 1.3 Metodologia e etapas de desenvolvimento | 24 |
| 1.4 Resultados esperados..... | 26 |
| 1.5 Conteúdo e organização..... | 28 |
| 2. TV DIGITAL..... | 30 |
| 2.1 Histórico | 30 |
| 2.2 Conceitos | 32 |
| 2.2.1 Interatividade | 33 |
| 2.2.2 Multiplexação e codificação | 33 |
| 2.2.3 Canal de <i>broadcast</i> e canal de retorno..... | 34 |
| 2.2.4 MPEG2-Transport Stream (TS)..... | 36 |
| 2.2.5 Middleware..... | 37 |
| 2.2.6 MPEG-21..... | 39 |
| 2.3 Sistema Americano - ATSC | 40 |
| 2.4 Sistema europeu - DVB..... | 41 |
| 2.5 Sistema japonês - ISDB..... | 43 |
| 2.6 Sistema brasileiro de TV digital | 44 |
| 2.7 Convergência entre TV digital e outras redes..... | 46 |
| 3. GERENCIAMENTO DE REDES | 47 |
| 3.1 Introdução ao gerenciamento de redes | 47 |
| 3.2 Arquiteturas de gerenciamento | 51 |
| 3.2.1 Simple network management protocol - SNMP | 51 |
| 3.2.2 Web-based enterprise management - WBEM..... | 53 |
| 3.2.3 Distributed Management Environment - DME..... | 58 |
| 3.2.4 Java Management Extensions - JMX | 60 |
| 3.2.5 Open services gateway initiative - OSGi..... | 63 |
| 3.2.6 Gerenciamento de serviços..... | 66 |
| 4. GERENCIAMENTO DE REDES DE TV DIGITAL..... | 70 |
| 4.1 Gerenciamento da transmissão de TV digital no sistema EUTELSAT | 70 |
| 4.2 Gerenciamento de set-top boxes baseadas em Java no sistema ATSC..... | 73 |
| 4.3 Arquitetura de gerenciamento do sistema DAVIC | 74 |
| 4.4 Gerenciamento de redes européias de TV digital a cabo..... | 78 |
| 4.5 Arquitetura de gerenciamento do projeto CableHome da CableLabs..... | 79 |
| 4.6 Gerenciamento proposto pelo Governo Brasileiro para o sistema brasileiro de TV digital..... | 82 |
| 4.7 Panorama geral dos sistemas de gerenciamento para TV digital..... | 83 |
| 5. ANÁLISE DOS REQUISITOS DE UM SISTEMA DE GERENCIAMENTO DE SET- TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL..... | 86 |
| 5.1 Quality function deployment (QFD) / Especificação de requisitos funcionais..... | 88 |
| 5.2 Requisitos não funcionais..... | 92 |
| 5.2.1 Requisitos de hardware da <i>set-top box</i> | 92 |
| 5.2.2 Requisitos da rede de televisão digital..... | 93 |
| 5.2.3 Requisitos do servidor de gerência | 93 |
| 5.2.4 Requisitos do middleware da <i>set-top box</i> | 93 |
| 5.2.5 Requisitos de interatividade com o usuário final..... | 94 |
| 5.2.6 Requisitos do projeto de software da aplicação..... | 96 |

6. DETALHAMENTO OPERACIONAL DE UM SISTEMA DE GERENCIAMENTO DE SET-TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL 98

| | |
|-----------------------------------|----|
| 6.1 Diagrama de casos de uso..... | 98 |
|-----------------------------------|----|

7. ANÁLISE E ESCOLHA DA SOLUÇÃO TÉCNICA 101

| | |
|---|-----|
| 7.1 Tecnologias disponíveis | 101 |
| 7.1.1 Comunicação física do canal de <i>broadcast</i> | 101 |
| 7.1.2 Mecanismo de multiplexação do canal de <i>broadcast</i> | 102 |
| 7.1.3 Rede de transporte do canal de <i>broadcast</i> | 103 |
| 7.1.4 Identificação da aplicação de gerenciamento nas tabelas MPEG-2..... | 104 |
| 7.1.5 Modo de transmissão de dados no canal de <i>broadcast</i> | 105 |
| 7.1.6 Canal de retorno | 105 |
| 7.1.7 Protocolo de comunicação de gerência..... | 106 |
| 7.1.8 Modo de armazenagem das informações de gerenciamento..... | 107 |
| 7.1.9 Linguagem de programação utilizada no lado gerente | 108 |
| 7.1.10 Linguagem de programação utilizada no lado agente..... | 109 |
| 7.1.11 Plataforma de middleware | 109 |
| 7.1.12 Arquitetura de controle de serviços | 110 |
| 7.1.13 Localização física da aplicação a ser executada no lado agente | 111 |
| 7.2 Cenário de utilização | 111 |
| 7.3 Tecnologias utilizadas | 112 |

8. PROPOSTA PARA UMA ARQUITETURA ABERTA DE UM SISTEMA DE GERENCIAMENTO DE SET-TOP BOXES E SERVIÇOS EM REDES DE TV DIGITAL..... 117

| | |
|--|-----|
| 8.1 Arquitetura de rede..... | 117 |
| 8.1.1 Escalabilidade..... | 119 |
| 8.1.2 Mobilidade..... | 121 |
| 8.1.3 Modelo de negócios..... | 121 |
| 8.2 Modelagem..... | 122 |
| 8.2.1 Posicionamento da arquitetura em relação ao middleware | 122 |
| 8.2.2 Arquitetura interna..... | 125 |
| 8.2.3 Modelagem dos estados do sistema | 134 |
| 8.2.3 Pacotes – Gerente | 137 |
| 8.2.4 Diagramas de classe – Agente | 144 |
| 8.3 Testes de verificação | 151 |
| 8.4 Estratégia de integração..... | 152 |
| 8.5 Testes de integração | 152 |
| 8.6 Testes de validação..... | 153 |

9. ESTUDO DE CASO DE GERENCIAMENTO DE TV DIGITAL: UM CENÁRIO E SUA IMPLEMENTAÇÃO 154

| | |
|--|-----|
| 9.1 Objetivo do estudo de caso e testes propostos..... | 154 |
| 9.2 Ferramentas utilizadas | 155 |
| 9.2.1 Agente | 155 |
| 9.2.2 Gerente | 156 |
| 9.3 Topologia..... | 156 |
| 9.4 Resultados da Implementação | 157 |
| 9.4.1 <i>Framework</i> baseado em estados | 157 |
| 9.4.2 MVC..... | 161 |
| 9.4.3 OSGi..... | 161 |
| 9.4.5 WBEM..... | 164 |
| 9.4.6 CIM Schema..... | 167 |
| 9.4.7 Persistência..... | 169 |
| 9.4.8 MPEG-21..... | 169 |
| 9.5 Visão geral da implementação..... | 171 |

10. CONCLUSÃO E TRABALHOS FUTUROS..... 173

| | |
|------------------------------|-----|
| 10.1 Conclusão | 173 |
| 10.2 Trabalhos Futuros | 176 |

REFERÊNCIAS BIBLIOGRÁFICAS 179

APÊNDICE A – Detalhamento dos Casos de Uso 189

| | |
|--|-----|
| A.1 Caso de uso macro 1 – realizar a gerência de configuração | 189 |
| A.2 Caso de uso macro 2- realizar a gerência de contabilização | 198 |
| A.3 Caso de uso macro 3 – realizar a gerência de falhas..... | 202 |
| A.4 Caso de uso macro 4 – realizar a gerência de desempenho | 205 |
| A.5 Caso de uso macro 5 – realizar a gerência de segurança | 209 |

1. INTRODUÇÃO

1.1 Objetivos e Motivação

Atualmente um dos principais focos de pesquisa nas áreas de telecomunicações e software é voltado ao desenvolvimento e à implantação da TV Digital em todo o mundo. Estas pesquisas começaram a surgir no fim dos anos 80, época na qual a tecnologia digital começava a ser amplamente explorada e dominada. Em um primeiro momento, elas se concentraram em 3 pólos distintos: Estados Unidos, Japão e Europa (EVANS, 1995).

Um sistema de TV Digital abriga não só a transmissão dos sinais de *broadcast* em formato digital, mas também a criação e recepção digital destes sinais. Mais do que isto, oferece a possibilidade de interação com o usuário final, ampliando em enorme escala a possibilidade de oferecimento de novos produtos e serviços. Tudo isto é realizado pela chamada *set-top box*, que será o elemento utilizado durante o período de transição entre o mundo digital e o analógico. De fato, ela será responsável por receber os sinais digitais do *broadcast*, decodificá-los para sinais analógicos e entregá-los ao aparelho de televisão analógico do usuário final. Nesta *set-top box* também estão contidos os elementos de software necessários para o oferecimento de serviços aos usuários (WHITAKER, 1999). Quando do fim da migração para o sistema digital, a *set-top box* será substituída pelo terminal de acesso, que é um elemento que oferecerá as mesmas funcionalidades de software que a *set-top box*, porém que já virá embutido no próprio aparelho de televisão digital. Portanto, todas as referências feitas neste trabalho às *set-top boxes* podem ser aplicadas também para terminais de acesso.

Quando estavam sendo projetados, na década de 80, os sistemas de TV Digital ainda não levavam em conta uma outra rede que foi crescer muito apenas no meio da década de 90: a Internet. Esta, que tem como princípios a interatividade total com o usuário e a digitalização de sua forma de comunicação e transmissão, fincou um novo paradigma na cultura tecnológica de todo o mundo. Apesar de tamanho avanço, a Internet não ocupa hoje nem 10%

das casas que possuem uma TV analógica em todo o mundo. A partir daí é que termos como “exclusão digital” passaram a ser utilizados, para diferenciar aqueles que têm acesso irrestrito à Internet daqueles que ainda possuem como principais meios de comunicação o rádio, a televisão analógica e o telefone (ZUFFO, 2001).

Nos últimos anos também começaram a aparecer várias pesquisas e projetos que propunham a integração de diversas redes diferentes, a fim de melhorar o custo para os operadores destas redes e de poder ampliar o leque de produtos oferecidos. A primeira iniciativa desta convergência foi a possibilidade do transporte de voz sobre redes de pacotes.

Do mesmo modo, hoje se acredita que o sistema de TV digital possa agregar valor à Internet, ao permitir que mais usuários tenham acesso a seus serviços. Por outro lado, a TV Digital também pode tirar proveito das características de interatividade da Internet (SCHWALB, 2003).

Deste modo, o panorama hoje existente de exclusão digital pode começar a ser quebrado com a inserção de uma nova rede – Internet - na casa dos usuários que já possuem um aparelho de televisão – quase 90% das casas, no caso do Brasil (BNDES, 2000).

Por outro lado, um dos desafios mais frequentes no cenário de pesquisa de redes é o de como controlar todos os recursos da mesma, a fim de que se possa ter uma utilização econômica, confiável e otimizada destes recursos (CARVALHO, GABOS; 2003). Para tanto, sempre o projeto e a pesquisa de redes foi – ou pelo menos deveria – ser acompanhado pela pesquisa de gerenciamento desta rede. O crescimento de todas as redes – sem dúvida principalmente a Internet e a TV digital (ambas ainda gozando de grandes taxas de crescimento) – fez com que este gerenciamento tivesse sua importância mais do que dobrada, a fim de que a operação e o planejamento destas redes possa ocorrer da forma mais otimizada e controlada possível. Apesar de toda esta importância, nenhum dos 3 sistemas de TV digital atualmente utilizados no mundo considerou este aspecto de gerência como peça principal do mesmo desde o início de seu desenvolvimento. Hoje, pesquisadores do mundo inteiro tentam adequar estes sistemas já em operação a necessidades sempre crescentes de usuários e portanto a novas necessidades de gerenciamento. Obviamente uma solução sempre é encontrada, apesar de muitas delas não se apresentarem satisfatórias.

No caso do Brasil, que ainda está em fase de escolha e desenvolvimento de seu sistema próprio, existe uma grande oportunidade de se projetar esta integração de redes visando o gerenciamento das mesmas, isto desde as primeiras pesquisas até a construção do produto final.

Dentro deste contexto, este trabalho tem como objetivo principal apresentar uma arquitetura aberta de gerenciamento das *set-top boxes* e serviços oferecidos por esta nova rede de TV Digital, levando em consideração sua integração com a Internet e o uso de padrões abertos. Acompanhando o desenvolvimento do projeto de TV Digital do Governo Brasileiro e também seguindo princípios clássicos do desenho de requisitos de gerenciamento e de redes de gerenciamento, esta pesquisa visa propor um produto final útil e condizente com as necessidades atuais dos sistemas de TV Digital.

1.2 Escopo

O escopo desta pesquisa é voltado, primeiramente, ao delineamento dos requisitos de gerenciamento necessários para o controle otimizado das *set-top boxes* e serviços de uma rede de TV Digital, integrada se possível com algum outro tipo de rede (por exemplo Internet). Com estes requisitos em mãos, é possível projetar a arquitetura de rede que atenderia a esta necessidade de gerenciamento, utilizando padrões e protocolos abertos.

Para que esta arquitetura de gerenciamento fosse desenhada e os requisitos fossem definidos de forma satisfatória, foram utilizados princípios clássicos de gerenciamento a fim de determinar o escopo deste projeto. Na verdade, cruzaram-se as áreas funcionais de gerenciamento definidas no modelo Open Systems Interconnection (OSI) de gerenciamento (STALLINGS, 1997) com as camadas funcionais de gerenciamento definidas na Telecommunications Management Network (TMN) (ALENCAR, 1999).

Estas áreas funcionais OSI bem como as camadas funcionais de gerenciamento TMN são definidas em detalhes mais adiante nesta pesquisa porém as mesmas são:

1. Áreas funcionais de gerenciamento OSI: falhas, configuração, segurança, desempenho e contabilização
2. Camadas funcionais de gerenciamento TMN: elemento de rede, rede, serviço e negócio

A partir da definição do escopo da pesquisa baseada nestes princípios, foram considerados também em conta os objetivos do projeto do Governo Brasileiro de TV Digital, chamado de Sistema Brasileiro de TV Digital (SBTVD) (FINEP, 2004a), sendo que em muitas vezes alguns objetivos deste projeto foram deixados de lado neste trabalho, por não se tratarem de fatores fundamentais. Por outro lado, alguns outros itens importantes foram ampliados e detalhados com mais rigor. Com estes elementos em mãos, foram definidos os pontos de interesse desta pesquisa (ver tabela 1.1, onde os elementos em laranja foram retirados de (FINEP, 2004a)):

TABELA 1.1 – ESCOPO DESTA PESQUISA

| Área Funcional OSI / Camada de Gerenciamento TMN | Elemento de Rede | Rede | Serviço | Negócio |
|---|------------------|------|---------|---------|
| Falha | x | x | x | |
| Configuração | x | x | x | |
| Segurança | em parte | x | x | |
| Performance | x (passiva) | x | x | |
| Contabilização | | x | x | |
| Em laranja estão marcados os requisitos que foram mapeados na RFP do governo | | | | |

O processo de definição deste escopo também foi realizado paralela e iterativamente com a utilização do método Quality Function Deployment (QFD), detalhado mais adiante neste trabalho.

Como pode ser notado, o escopo recai principalmente sobre o gerenciamento tradicional de elementos de rede (sendo neste caso considerado como elemento de rede apenas a *set-top box* – não são considerados elementos de transporte) e da nova tendência do gerenciamento de serviços, este abrangido nesta pesquisa nas 5 áreas funcionais OSI. O gerenciamento de serviços é atualmente uma forte e consistente tendência, pois o mesmo visa melhorar a entrega do produto ou serviço ao usuário final de telecomunicações, sendo que é sabido que o

objetivo deste mercado é o de aumentar receitas através do oferecimento de novos serviços realizando o mínimo possível de novos investimentos em infra-estrutura. É ainda considerado no escopo o gerenciamento da contabilização do uso da rede bem como a análise de desempenho da mesma.

As áreas de gerenciamento de falhas e desempenho são tradicionalmente as mais aplicadas em arquiteturas disponíveis, sendo que neste trabalho questões de falhas e desempenho são resolvidas pelos gerentes, que recebem dados periódicos das *set-top boxes*. Eventualmente, um agente também pode resolver alguns destes problemas localmente. Para a área de contabilização foram apenas definidos os gerenciamentos da rede e dos serviços já que um cenário de uma rede de TV digital não incorpora a contabilização da utilização do terminal de acesso. Menos tradicionais, as áreas de configuração e segurança são na verdade o maior desafio deste projeto, sendo que a função de segurança em nível de elemento de rede não estará completamente definida, por tratar-se de assunto realmente amplo.

Com relação à rede propriamente dita considerada no escopo desta pesquisa, é admitido que a mesma oferece um canal digital de *broadcast* e um canal de retorno, acessado através de alguma rede de comunicação específica (por exemplo linha discada) e contando, neste caso, com o uso de uma pilha Internet Protocol (IP).

1.3 Metodologia e etapas de desenvolvimento

O desenvolvimento desta pesquisa baseou-se no uso do Systems Engineering Capability Maturity Model (SE-CMM) (SEI, 1995) para definição do cronograma e de todas as atividades necessárias para o desenho da arquitetura aqui proposta.

O SE-CMM é uma metodologia desenvolvida pelo Software Engineering Institute, onde diversos passos necessários ao projeto de sistemas são definidos para que o produto final apresente-se com qualidade e atendendo aos requisitos iniciais do usuário. Mais do que isto, a metodologia é flexível o bastante ao não definir passos obrigatórios nem mesmo a ordem nos

quais os mesmos devem ser empregados. Com isto, cabe ao pesquisador determinar qual a melhor ordem para o desenvolvimento de seu sistema.

Para esta pesquisa, foram adotados 3 princípios para a filosofia de trabalho:

1. O produto final deve estar de acordo com os requisitos iniciais definidos por usuários ou possíveis usuários.
2. O problema deve ser tratado de forma ampla, sendo que a utilização de novas tecnologias, mesmo que estas não estejam em estado tão avançado de uso, porém sejam inovadoras e caracterizem-se como uma tendência, será considerada.
3. Serão levadas em conta e estudadas propostas de outras pesquisas que possuam o mesmo objetivo que esta, garantindo sempre que esta levará em conta as peculiaridades do contexto nacional de desenvolvimento de sistemas de TV Digital.

Com esta filosofia definida, e embasado na metodologia SE-CMM, foram definidas as seguintes etapas para o desenvolvimento desta pesquisa:

1. Definição do escopo (já apresentada no item anterior).
2. Estudo bibliográfico de trabalhos relacionados.
3. Entendimento das necessidades dos usuários finais, baseando-se fortemente no desenvolvimento de um mapa de Quality Function Deployment (QFD) .
4. Desenvolvimento da especificação de requisitos (funcionais e não-funcionais).
5. Desenvolvimento dos detalhes da operação do sistema, baseando-se na definição dos casos de uso (já que esta pesquisa baseia-se na metodologia orientada a objetos para desenvolvimento de software).
6. Análise e escolha da melhor solução técnica a ser utilizada .

7. Desenvolvimento da arquitetura propriamente dita.
8. Identificação e definição das interfaces (sejam elas de software ou hardware) internas e externas à arquitetura.
9. Definição de uma estratégia de integração dos módulos do sistema.
10. Definição de um estudo de caso.
11. Desenvolvimento (programação) propriamente dita.
12. Integração do sistema.
13. Validação do sistema e execução do estudo de caso, com análise de resultados.

Vale citar que o plano de verificação foi utilizado durante todo o processo de desenvolvimento da arquitetura aqui proposta, e que todos os modelos de software, essenciais para o desenho da arquitetura e operacionalidade da solução, foram definidos utilizando-se a metodologia Unified Modeling Language (UML) (BOOCH, RUMBAUGH, JACOBSON; 1999).

1.4 Resultados esperados

Dentro do contexto apresentado, são esperados os seguintes resultados para que esta pesquisa possa ser considerada bem sucedida:

1. Apresentação de uma arquitetura de gerenciamento realmente aberta, que utilize ferramentas e padrões de domínio público e que seja componentizada, para que os módulos da mesma possam ser reutilizados em outras pesquisas.

2. Apresentação de uma arquitetura que englobe o gerenciamento de serviços, item pouco explorado em produtos de mercado ou mesmo em pesquisas voltadas à TV digital, porém de vital interesse atualmente.
3. Definição de um gerenciamento projetado especificamente para o elemento de rede *set-top box*, que apresenta restrições claras de desempenho mas que deve possuir um mínimo de inteligência, sendo portanto um elemento que deve ser modelado de modo distinto a outros elementos de rede hoje existentes.
4. Utilização de um protocolo de comunicação independente de IP e ATM no canal de *broadcast*. Muitas das redes de TV digital não suportarão estes protocolos na rede *broadcast*, pelo menos em um primeiro momento (o uso de IP *unicast* e IP *multicast* no canal de *broadcast* ainda é limitado).
5. Definição de uma arquitetura de gerenciamento de segurança da *set-top box* e também das aplicações (serviços).
6. Definição de uma arquitetura de gerenciamento de configuração da *set-top box* e também das aplicações (serviços).
7. Consideração da presença de um canal de retorno, no mínimo intermitente. A maioria das propostas considera canais de retorno permanentes, o que não ocorre para um sistema de TV digital terrestre. O sistema gerente também deveria estar ciente de que, devido à presença do canal de retorno não ser permanente, a gerência deve tolerar atrasos no envio de respostas da *set-top box*. Por outro lado, o terminal de acesso deve estar preparado para administrar o envio das mensagens de gerenciamento apenas quando este canal de retorno estiver disponível, armazenando as mesmas para envio posterior caso contrário. **Isto implica também na consideração de que algumas ações de gerenciamento podem não requerer confirmação.**
8. Compatibilidade com uma arquitetura de gerenciamento mais flexível que o SNMP, como por exemplo WBEM, que facilita muito o processamento inteligente de informações na *set-top box*, através do uso de XML.

9. Real integração com um middleware aberto e de mercado, aproveitando recursos e APIs oferecidas pelo mesmo para o bom desempenho da arquitetura.
10. Integração com um sistema de autenticação e autorização de usuários (externo ao sistema de gerenciamento). Esta integração deveria permitir ao usuário selecionar o tipo de autenticação a ser usada por ele, sendo função do sistema de gerenciamento controlar estas informações e do sistema de autenticação e autorização armazenar e verificar senhas.
11. Realização de um estudo de caso onde elementos de rede e serviços possam ser gerenciados utilizando-se a arquitetura projetada e onde o tráfego de gerência possa ser transportado por uma rede de TV digital. Caso a mesma não possa ser utilizada, admite-se o uso de simuladores para esta rede.

Caso estes resultados sejam atingidos, acredita-se que a pesquisa aqui apresentada tenha agregado valor ao tema de gerenciamento de redes de TV digital, já que estas contribuições não estão presentes em nenhum sistema de gerenciamento de TV digital existente, conforme será detalhado ao longo deste trabalho.

1.5 Conteúdo e organização

Este trabalho está estruturado em 10 capítulos: introdução, TV digital, gerenciamento de redes, gerenciamento de redes de TV digital, análise de requisitos de um sistema de gerenciamento de TV digital, detalhamento operacional de um sistema de gerenciamento de TV digital, análise e definição da solução técnica, proposta para arquitetura aberta de um sistema de gerenciamento de TV digital, estudo de caso e conclusão e trabalhos futuros.

No capítulo 2 são apresentados os conceitos de TV digital e no capítulo 3 os conceitos de gerenciamento de redes. Estes formam o embasamento teórico desta pesquisa. No capítulo 4, sobre gerenciamento de redes de TV digital, é realizada uma revisão bibliográfica das soluções propostas e atualmente existentes para o gerenciamento de redes de TV digital. No fim deste mesmo capítulo é apresentado um panorama geral destas soluções e uma visão

crítica das mesmas é realizada, identificando assim oportunidades de pesquisa que são o foco deste trabalho.

O capítulo 5 apresenta os requisitos necessários para o desenvolvimento de qualquer sistema de gerenciamento de TV digital. Nele também são definidas quais as vantagens que um sistema deveria apresentar com relação aos apresentados no capítulo anterior.

O capítulo 6, através da técnica de definição de casos de uso, detalha a operacionalidade necessária a qualquer sistema de gerenciamento de TV digital, novamente de modo genérico, como no capítulo 5.

As alternativas tecnológicas para a solução do problema apresentado nesta pesquisa são apresentadas no capítulo 7, sendo que é definido também um cenário factível para a aplicação da arquitetura proposta. Com base neste cenário e nas alternativas tecnológicas apresentadas, é justificada a escolha da solução técnica que mais se adequa aos requisitos antes traçados.

No capítulo 8 a arquitetura propriamente dita é apresentada, bem como toda a modelagem, baseada no método Unified Modeling Language (UML). Testes de verificação e integração também são definidos neste capítulo.

No capítulo 9 é traçado um estudo de caso que possa comprovar que a arquitetura proposta atingiu os objetivos iniciais da pesquisa. A análise dos resultados deste estudo de caso é apresentada neste mesmo capítulo.

O último capítulo, 10, relata as conclusões desta pesquisa e quais os trabalhos futuros propostos para a continuação da mesma.

2. TV DIGITAL

2.1 Histórico

A televisão analógica, no Brasil e no mundo, transformou-se no principal meio de comunicação com as massas do último século e da atualidade. Baseada em um sistema de transmissão *broadcast*, onde os mesmos sinais de TV enviados a partir de uma emissora são captados por milhares de antenas e satélites presentes nas mais de 40,6 milhões de residências no Brasil (87,7% de cobertura, com tendência de crescimento), a televisão analógica modificou – e continua modificando - não só padrões tecnológicos existentes anteriormente, mas também, e principalmente, padrões culturais e sociais (ZUFFO, 2001).

Tamanha abrangência e influência ainda não foram atingidas por nenhum outro meio de comunicação; a Internet, por exemplo, ainda tem muito a conquistar para chegar ao mesmo nível da televisão analógica. E talvez esta conquista nunca seja realizada pelos métodos de acesso existentes hoje, concentrados nos grandes centros apenas – ao contrário da televisão analógica e mesmo do telefone fixo, que hoje ocupam praticamente todo o território nacional e o mundo (ZUFFO, 2001).

A TV digital é um projeto global, já realidade em alguns países que se adiantaram e definiram modelos de negócios e padrões tecnológicos a serem utilizados. A principal motivação inicial foi a melhoria da qualidade de som e imagem das TVs analógicas, que passariam a ser digitais, desde a geração até a recepção, passando pela transmissão destes sinais. Isto permitiria não só esta melhoria na qualidade na recepção, mas também a possibilidade de criação de efeitos especiais, possibilidade de realização de cópias sem perda de qualidade, entre outros benefícios. Com este objetivo em mente, foram pesquisados e produzidos muitos padrões de codificação digital de vídeo e áudio, que deram origem, por exemplo, à High Definition Television (HDTV) e à Standard Definition Television (SDTV), entre outros. Para o usuário final padrão brasileiro, esta melhoria de áudio e vídeo já representa uma enorme

vantagem e motivação suficiente para o investimento na troca de tecnologia (BNDES, 2000). De acordo com pesquisa realizada pelo CPqD (Centro de Pesquisa e Desenvolvimento em Telecomunicações) em 1999 (BNDES, 2000), a aceitação da TV digital pelos telespectadores deverá ser plena, na medida em que os consumidores terão a possibilidade de escolher sua programação, o horário mais conveniente para assisti-la e ainda poderão receber informações adicionais sobre os programas através do controle remoto.

Entretanto, estas mesmas pesquisas (BNDES, 2000), (BNDES, 2002) atestam que a aceitação de serviços interativos em TV Digital tem um caminho mais difícil a trilhar do que a melhoria na qualidade de áudio e vídeo, já que exige o aprendizado rápido de novas funções. É importante lembrar que uma aplicação interativa de difícil aprendizado desestimula o usuário final e acaba por não despertar o interesse do mesmo (ZUFFO, 2001). Algumas aplicações interativas que são temas de pesquisa ou mesmo estão já em ambiente de produção de TV digital são (SCHWALB, 2003):

1. Electronic Program Guide (EPG)
2. Video on Demand (VoD)
3. Jogos *on-line* via TV Digital
4. Game Show
5. Programas de música interativos
6. Programas de educação interativos (T-learning)
7. Eventos Esportivos interativos
8. Viagens (hospedagem e reserva de vôos)
9. Home Shopping
10. Home Banking

11. Apostas *on-line* (exemplo Loteria Esportiva *on-line*)
12. T-Government
13. Correio Eletrônico via TV digital
14. Personalização da experiência multimídia do usuário final

As primeiras pesquisas para a adoção de um sistema de TV Digital começaram a surgir no fim dos anos 80, sendo que estas não contemplavam apenas a geração, transmissão e recepção de sinais digitais via *broadcast*, mas também a possibilidade de interatividade (completa ou não) com o usuário final. Estas pesquisas concentraram-se em 3 grandes centros: Europa, Estados Unidos e Japão (EVANS, 1995). Estes 3 centros atualmente já possuem o seu sistema de TV Digital em uso comercial de larga escala, cada um com suas respectivas vantagens, restrições e pontos em comum, como por exemplo o uso de um terminal de acesso (*set-top box*) para recepção dos sinais digitais e a adoção de conversores durante o processo de transição da TV analógica para a TV digital (o que ainda está em andamento). Um quarto padrão, que na verdade utiliza como base estes 3 existentes hoje, é o chinês, que surgiu no ano de 2004, chamado de Digital Television Industrial Alliance (DTVIA). O padrão utilizado hoje na Europa é o Digital Video Broadcasting (DVB), nos Estados Unidos é o Advanced Television Systems Committee (ATSC) e no Japão é o Integrated Services Digital Broadcasting (ISDB).

São analisadas as principais características destes padrões adiante neste trabalho, e também é apresentada a situação do desenvolvimento do Sistema Brasileiro de TV Digital. Primeiramente, são introduzidos os principais conceitos que formam a base de qualquer sistema de TV digital.

2.2 Conceitos

2.2.1 Interatividade

Com um sistema digital de geração, transmissão e recepção de sinais de televisão, é possível também realizar uma **interação** com o usuário final. Esta interação, conceito simples e muito utilizado hoje na Internet, poderia também gerar uma grande evolução nas comunicações atuais da TV, levando até mesmo à **convergência** entre meios, por exemplo, Internet e Televisão. Segundo (SCHWALB, 2003), “interatividade é a propriedade que um sistema tem de emitir respostas em tempo real de acordo com os estímulos realizados pelos usuários do sistema”. Além disto, um sistema que tem a propriedade da interatividade motiva seus usuários a não adotarem uma posição passiva diante dos mesmos (SCHWALB, 2003). Deste modo, podem existir várias modalidades de interatividade em um sistema de TV Digital, desde as locais até as que buscam dados remotamente. Adiante neste trabalho, é mostrado como o sistema de TV digital europeu – Digital Video Broadcasting (DVB) – disponibiliza 3 tipos distintos de interatividade, sendo uma completamente local e as outras 2 baseadas no uso do canal de retorno.

2.2.2 Multiplexação e codificação

Este conceito de interatividade com o usuário final também foi e é uma das chaves para o desenvolvimento da TV Digital no mundo, e exige o desenvolvimento de um mecanismo de transporte de dados bi-direcional, no antes apenas mundo analógico unidirecional. Além disto, a TV Digital deve ter a capacidade de transmitir áudio, vídeo e dados simultaneamente (ver Figura 2.1), para que a interatividade e a experiência do usuário seja completa. Esta capacidade de transmitir diferentes mídias simultaneamente utiliza atualmente a mesma técnica por todos os padrões existentes (ORZESSEK; SOMMER, 1998), seguindo o modelo de referência do ITU-T definido em (ITU-R, 1996), que utiliza como transporte o MPEG2-Transport Stream (MPEG-2 TS).

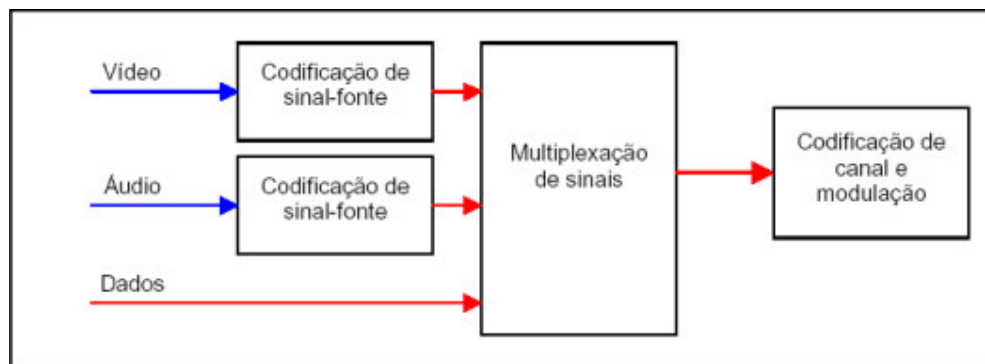


FIGURA 2.1 – MULTIPLEXAÇÃO DE SINAIS EM TV DIGITAL

No momento da demultiplexação, são utilizadas as tabelas MPEG2-Transport Stream (MPEG-2 TS) para a correta identificação e decodificação dos programas (as tabelas MPEG-2 TS são definidas adiante neste trabalho).

Conforme indicado também na figura 2.1, os *streams* de áudio e vídeo são primeiramente codificados antes de serem multiplexados. Estes também devem ser decodificados após a demultiplexação do sinal no lado cliente. Este processo de codificação e decodificação tanto de áudio como de vídeo segue padrões bem definidos da indústria (ORZESSEK, SOMMER, 1998), sendo que o objetivo principal destes padrões é o de atingir um máximo grau de compressão com uma qualidade de entrega aceitável, já que é sabido que conversões analógico-digitais trazem perdas à fonte codificada (codificação com perdas) (EVANS, 1995). Nos sistemas de TV digital hoje existentes, são utilizados os padrões de codificação de áudio e vídeo definidos nas especificações MPEG.

2.2.3 Canal de *broadcast* e canal de retorno

Um ponto em comum entre todos os sistemas de TV Digital é a possibilidade de se realizar a comunicação através de 2 canais (ver figura 2.2), que são utilizados em conjunto e muitas vezes simultaneamente. Estes canais são:

- *Broadcast* (que pode utilizar uma rede terrestre, de satélite, a cabo ou mesmo microondas em alguns casos. Também já existem padrões para a recepção de TV digital em dispositivos móveis (ORZESSEK, SOMMER, 1998)).
- Retorno (que normalmente utiliza a rede telefônica ou algum padrão *wireless* como transporte).

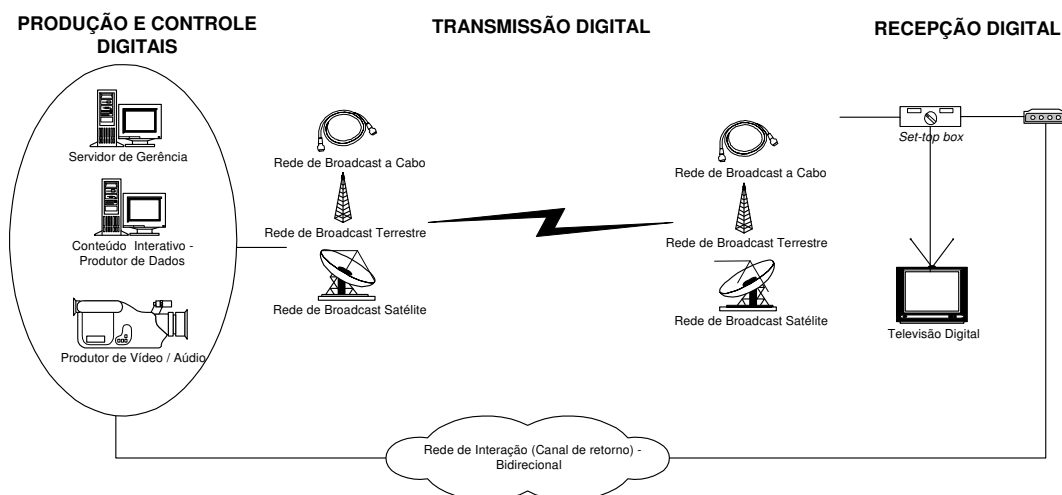


FIGURA 2.2 – ARQUITETURA DE REDE PADRÃO PARA TV DIGITAL

No caso específico de transmissão de **dados** pelo canal de *broadcast*, 4 métodos são empregados atualmente (definidos no (ITU-R, 1996)):

1. **Data Piping:** dados transmitidos de forma não sincronizada com nenhuma outra mídia ou mecanismo de repetição. Utiliza tecnologias proprietárias para interpretação e codificação do seu conteúdo.
2. **Data Streaming:** envio de dados sincronizados com eventos de outras aplicações.
3. **Data / Object Carousel:** envio do mesmo dado periodicamente, como um “carrossel” de dados.

4. Multiprotocol Encapsulation: que fornece um mecanismo para o transporte de dados sobre MPEG-2-TS. Estes dados transportados podem estar encapsulados no próprio protocolo IP. Deste modo pacotes IP direcionados a algum elemento específico na rede podem ser enviados pelo canal de *broadcast*.

Nota-se que o canal de *broadcast*, em um sistema de TV Digital, possui a mesma função que em um sistema analógico, ou seja, o de transmitir sinais que todos os elementos terminais da rede possam captar e processar. A diferença está justamente na multiplexação de mídias e na possibilidade do envio de dados por este canal.

2.2.4 MPEG2-Transport Stream (TS)

Conforme citado anteriormente, as especificações MPEG definem os padrões de codificação de áudio e vídeo a serem utilizados em sistemas de TV digital. Além destes importantes padrões, o MPEG-2 define também os meios pelos quais estes produtos codificados devem ser entregues ao usuário final. Estas definições são encontradas na especificação MPEG-2 TS.

O MPEG-2 TS é formado de um elemento principal, chamado de **transport stream (TS)**. O transport stream é na verdade a seqüência de bits entregue ao usuário final, com informações suficientes para (ORZESSEK, SOMMER, 1998):

1. Extrair o programa desejado dentro de um conjunto de programas enviados simultaneamente pelo transport stream.
2. Extrair um conjunto de programas inter-relacionados dentro de um conjunto de programas enviados simultaneamente pelo transport stream.
3. Manter a sincronização entre os diversos programas e os diversos elementos de vídeo, áudio e dados que formam estes programas.

Estes elementos de áudio, vídeo e dados que formam um (ou mais) programa(s) são chamados na terminologia MPEG-2 TS de **elementary streams**. Cada elementary stream possui um

identificador único dentro do transport stream, chamado de **program identifier (PID)**. A coleção de PIDs que forma um programa é indicada em uma **program map table (PMT)**. (ORZESSEK, SOMMER, 1998). Finalmente, existe uma outra tabela, também enviada no MPEG-2 TS, chamada de **program association table (PAT)**, que lista todos os PMTs presentes em um Transport Stream, ou seja, todos os programas disponíveis ao usuário final (ORZESSEK, SOMMER, 1998).

Como pode ser notado, o sistema MPEG-2 TS é complexo porém lida satisfatoriamente com os problemas de identificação e sincronismo de programas e elementary streams. Além disto, os sistemas em uso atualmente utilizam outras tabelas de dados além das aqui apresentadas, sendo estas extensões ao padrão MPEG-2 TS (por exemplo (ETSI, 1998)).

2.2.5 Middleware

Em um sistema de TV digital, as aplicações devem ser escritas de modo que detalhes do hardware, ou mesmo do sistema operacional dos terminais de acesso, sejam transparentes ao desenvolvedor.

Para que esta portabilidade seja uma realidade, é utilizado nos terminais de acesso o chamado **middleware**. O middleware é na verdade uma camada de software entre as aplicações e o sistema operacional e tem como funções principais fornecer serviços básicos para um determinado domínio de aplicações e também ocultar a complexidade do Sistema Operacional (O'DRISCOLL, 1999). Existem middlewares proprietários (Microsoft TV, Power TV, OpenTV Core, Liberty, etc) e também abertos (Multimedia Home Platform (MHP, utilizado no sistema DVB), Digital Television Application Software Environment (DASE, utilizado no sistema ATSC), Application Engine Platform for Digital Broadcasting (ARIB, utilizado no sistema ISDB), Open Cable Application Platform (OCAP), etc). Os middlewares disponibilizam suas interfaces (funções) através de uma Application Programming Interface (API), fazendo assim com que as aplicações tirem proveito desta abstração do hardware. No caso dos middlewares abertos, estas APIs são públicas (O'DRISCOLL, 1999).

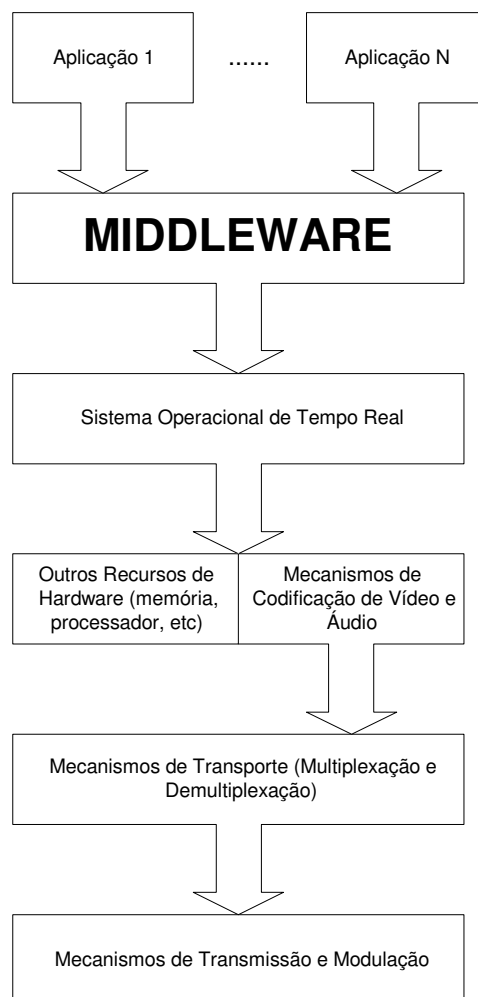


FIGURA 2.3 – O MIDDLEWARE DENTRO DA ARQUITETURA DE UMA *SET-TOP BOX*

O entendimento dos diversos padrões de middleware faz-se requisito essencial para os desenvolvedores de aplicações para TV digital.

Dentro deste contexto, destaca-se a iniciativa do chamado Global Executable MHP (GEM), padrão de middleware ainda em desenvolvimento porém que almeja a unificação dos 3 middlewares mais utilizados (MHP, DASE e ARIB - apesar do nome se referir explicitamente ao MHP). Lembra-se que esta é uma iniciativa muito nova porém alguns projetos já se baseiam na utilização do GEM.

Convém também notar que um sistema de gerenciamento de *set-top boxes* e serviços deve estar totalmente integrado ao middleware adotado. Apesar deste ponto importante, nenhum destes 3 middlewares mais utilizados atualmente (nem o GEM) especificam, mesmo que minimamente, uma arquitetura de gerenciamento das *set-top boxes* e serviços de TV digital.

2.2.6 MPEG-21

Apesar do MPEG-21 não ter sido desenvolvido especificamente para o ambiente de TV digital, ele pode ser adotado como solução de metadados para este ambiente, ou seja, como descritor de conteúdo dos serviços multimídia oferecidos, sejam eles de vídeo ou não.

A especificação MPEG-21 (ISO, 2002) detalha e padroniza o uso de tecnologias e metadados específicos tanto para a resolução do problema de direitos autorais (embora este problema ainda esteja longe de ser resolvido) como do problema de quais plataformas de software são necessárias para se executar uma determinada mídia em um determinado terminal de acesso (por exemplo qual adaptador é necessário para a execução de determinada mídia) (BURNETT et al, 2003). Entretanto, o que interessa na especificação MPEG-21 para este trabalho são 2 itens: o Digital Item Declaration (DID) e o Digital Item Identification (DII) (ISO, 2002).

Ambos são metadados, porém um é responsável por descrever o conteúdo do serviço multimídia (DID) e o outro (DII) por identificar univocamente o serviço.

O DID descreve a estrutura do serviço, seu conteúdo e como este conteúdo está interligado. Ele faz isto listando todas as mídias que compõem o mesmo e apontando as referências para estas. Ele é formado basicamente por **descritores** e pelos chamados **recursos**, que são na verdade a mídia propriamente dita (por exemplo uma referência a um arquivo musical). Um conjunto destes define um **componente**, sendo que a descrição completa de um serviço multimídia é formada por um conjunto destes componentes (ISO, 2002). Já o DII identifica estes serviços e as mídias que os compõem, através do uso de Universal Resource Identifiers (URIs).

Com estes metadados, é possível então listar completamente as características e os conteúdos dos serviços multimídia que serão oferecidos aos usuários finais.

Neste ponto, após os principais conceitos envolvidos em sistemas de TV digital terem sido esclarecidos, são apresentadas as características particulares de cada um dos 3 sistemas de TV digital atualmente em uso.

2.3 Sistema Americano - ATSC

O ATSC é o padrão utilizado nos Estados Unidos, tendo sido desenvolvido em 1996. Ele tem como principais características de transmissão (WHITAKER, 1999):

1. Codificação de vídeo MPEG-2 (como todos os outros padrões).
2. Codificação de áudio Dolby Digital Audio Coding-3.
3. Transporte de mídias com multiplexação MPEG-2 TS (como todos os outros padrões).
4. Banda de 6 Mhz.
5. Capacidade de transmissão de 19 Mbits/s (terrestre) e 38 Mbits/s (cabo).

O sistema americano foi um trabalho pioneiro, que porém acabou pecando pela grande necessidade de compatibilidade com o sistema analógico vigente no país (National Television Systems Committee (NTSC)). Na verdade, praticamente todos os esforços na definição do padrão foram relativos aos modos de codificação de vídeo e áudio a serem utilizados, tentando manter sempre esta compatibilidade, sendo que outros fatores importantes (como interatividade) foram esquecidos em um primeiro momento. Muita discussão também foi realizada com relação ao modo de transmissão de dados de *broadcast* mais adequado,

respeitando-se as taxas de erro dos canais de comunicação e das modulações empregadas neste sistema.

A pilha de protocolos do sistema ATSC suporta desde o envio de dados não sincronizados (mecanismo de Data Piping) até o uso de IP pelo mecanismo de Multiprotocol Encapsulation (CRINON; HEREDIA, 2000). Também existe a possibilidade do uso de Data Carousel (envio periódico de dados), através da tecnologia Digital Storage Media – Command and Control (DSM-CC), que fornece mecanismos para o encapsulamento e envio de dados periódicos sobre o MPEG-2 TS (CRINON; HEREDIA, 2000).

Com a pilha de protocolo IP presente na *set-top box*, pode ser alocado um endereço IP para o canal de *broadcast* e outro para o canal de interatividade, sendo que o primeiro poderia participar de grupos de IP *multicast* (ou não, o importante é haver 2 canais físicos de comunicação sem haver problemas de roteamento IP na rede). Na verdade, já existem pesquisas que envolvem o uso de IP *multicast* no sistema ATSC (CRINON; HEREDIA, 2000). Mais adiante é detalhado nesta pesquisa qual o melhor modo de se trabalhar com estes 2 tipos de tráfego de dados para o caso específico do gerenciamento de rede.

O padrão oficial não oferece interatividade nem há um modo de realização de recepção móvel (no entanto há aplicações comerciais que desenvolveram alguns mecanismos de interatividade e também de recepção móvel).

Hoje o sistema ATSC adota o Java como a plataforma padrão para execução de aplicações nas *set-top boxes*. Também são suportadas aplicações escritas em XHTML (eXtensible HyperText Markup Language) e JavaScript, entre outras.

2.4 Sistema europeu - DVB

O sistema europeu – DVB– teve seu início oficial em 1993, apesar de que muitas pesquisas já tinham sido iniciadas na década de 80. Ele foi o resultado da união de grandes empresas européias, principalmente alemãs, francesas e britânicas, que resultaram na especificação de

várias padrões para TV Digital, seja de transmissão terrestre (DVB-T), via satélite (DVB-S), a cabo (DVB-C) ou mesmo via rádio microondas (DVB-MS, DVB-MC), embora a prioridade para desenvolvimento tenha sido dada primeiramente à transmissão via satélite e a cabo, por serem mais simples que a terrestre (WOOD, 1997).

A filosofia do desenvolvimento do sistema DVB seguiu 2 princípios: o sistema deveria ser desenvolvido de acordo com as necessidades do mercado e deveria manter a interoperabilidade através da adoção de padrões abertos (WOOD, 1997). Hoje o sistema DVB é utilizado em outras regiões do mundo além da Europa, como por exemplo a Austrália.

O padrão MPEG-2 foi o escolhido para a codificação de áudio e vídeo, bem como para a multiplexação das diferentes mídias. A banda requerida para a transmissão destas mídias é de 19 Mbits/s a até 68 Mbits/s, porém o mais comum é se utilizar canais de 39 Mbits/s. No caso do DVB-T a transmissão pode atingir até 26 Mbits/s (WHITAKER, 1999). Desta banda, normalmente 110Kbits/s são alocados para tráfego de dados. Desde o início deste projeto o uso de um canal de interatividade foi considerado (REMERS, 1995). Mais especificamente são definidos 3 modos de interatividade com o usuário (que podem ser utilizadas por diferentes arquiteturas de gerenciamento dependendo da situação):

- Enhanced Broadcast: onde aplicações interativas são armazenadas na *set-top box* localmente. Não há uso do canal de retorno.
- Interactive Broadcast: onde um canal de retorno é utilizado para interagir com aplicações enviadas pelo canal de *broadcast*.
- Internet Access: onde um canal de retorno permanente é utilizado para interação

O middleware utilizado no sistema DVB é o MHP, cuja especificação está na versão 1.1.1 (ETSI, 2003). No MHP, as aplicações são escritas em Java ou Hyper-Text Markup Language (HTML), XHTML, ou European Computer Manufacturers Association Script (ECMAScript) e são transportadas por um *stream* DVB ou por uma conexão IP. No caso das aplicações em Java, elas são chamadas de Xlets (JONES, 2002). Do mesmo modo que no sistema ATSC, as aplicações são identificadas e descritas através do envio de metadados sobre a mesma. No

caso de uma aplicação de gerenciamento, os metadados também seriam enviados identificando esta aplicação.

Hoje amplamente utilizado, o sistema DVB é considerado o mais completo em termos de especificações e documentação, e estão em desenvolvimento vários trabalhos comerciais e também de pesquisa que tratam de problemas como transmissão de dados, gerenciamento, interatividade, entre outros. Foram apresentados aqui apenas os seus conceitos principais, pertinentes para o entendimento deste trabalho.

2.5 Sistema japonês - ISDB

O sistema japonês de TV Digital – ISDB – foi o primeiro padrão desenvolvido fora do âmbito dos Estados Unidos e Europa. Este sistema levou em conta as principais características dos sistemas em uso na época no Japão, e foi responsável por um grande avanço da indústria local. Talvez por ter desenvolvido este sistema praticamente apenas dentro de seu país, a documentação sobre o mesmo não é tão comum e alguns aspectos permanecem como caixas-pretas para pesquisadores de outros países.

Baseado na geração, transmissão e recepção digitais, o sistema japonês também utiliza o MPEG-2 como codificação de vídeo e o MPEG-2 Advanced Audio Coding (AAC) para áudio, sendo utilizado o MPEG-2 TS para transporte de sinais multiplexados pelo canal de *broadcast*. Neste canal de *broadcast* cada serviço (ou mídia) é identificado por um código de identificação de serviços (YOSHIMURA et al, 1992), sendo este na verdade um metadado (semelhante ao dos modelos ATSC e DVB). O projeto do ISDB também sempre levou em consideração questões de segurança no acesso a aplicações, sendo muitas destas restrições de acesso aplicadas na própria *set-top box*. Como o sistema DVB, o ISDB também permite a recepção móvel de sinais, sendo também compatível com o sinal de telefonia 3G utilizado no país. Em alguns testes realizados em laboratórios brasileiros, o sistema ISDB apresentou a melhor qualidade de transmissão e recepção de sinais, em comparação com os sistemas DVB e ATSC (ABERT, SET; 2000).

Inicialmente desenvolvido para utilização de um canal de satélite, hoje o ISDB também possui padrões para a transmissão terrestre, microondas e a cabo. A máxima taxa de transmissão é de 40 Mbits/s. Em um primeiro momento, a taxa dedicada para aplicações de dados alocada foi de 100 kbits/s, sendo que atualmente as aplicações de dados demandam mais banda do que este valor inicial. O tamanho do pacote utilizado é de 188 bytes. No início de seu desenvolvimento, a interatividade não foi considerada como parte do sistema, sendo que depois de alguns anos vários trabalhos apareceram sobre este tema e hoje a mesma é uma realidade (TADASHI, 1996).

O sistema ISDB se encontra hoje em uso em larga escala no Japão, onde várias aplicações e novos serviços foram incorporados ao mesmo, proporcionando novas dimensões ao trabalho inicial de digitalização de vídeo e áudio.

2.6 Sistema brasileiro de TV digital

As atividades do Brasil em TV Digital foram primeiramente alavancadas por pesquisas relacionadas a padrões de HDTV. Em 1998 a ANATEL (Agência Nacional de Telecomunicações), deu início ao processo de avaliação dos padrões existentes no mercado (ATSC, DVB e ISDB), tendo culminado com uma série de testes realizados em conjunto, nos anos de 1999 e 2000, pela própria Anatel, algumas universidades e 17 emissoras de Televisão.

Em 2003, foi decretado oficialmente pelo Governo Brasileiro o início dos trabalhos para a construção de um **Sistema Brasileiro de TV Digital** (decreto presidencial número 4901 de 26/11), que possui um apoio financeiro baseado no FUNTTEL (Fundo para o Desenvolvimento Tecnológico das Telecomunicações). Este Sistema Brasileiro de TV Digital tem como objetivo a construção de um **sistema terrestre de transmissão digital apoiado em padrões já utilizados em outros sistemas**. Deve ser lembrado que já existem sistemas comerciais de TV Digital no Brasil, via satélite (baseado em DVB-S) e sobre redes de TV a cabo (oferecidos a partir de 2005). Entretanto, estas duas iniciativas não estão disponíveis na TV aberta, mas sim na TV por assinatura. Por este motivo, os mesmos têm custo alto para os padrões brasileiros e atingem uma minoria da população.

Em 2004, foi definido um Comitê de Desenvolvimento, Consultor e Grupo Gestor, coordenados pela Financiadora de Estudos e Projetos (FINEP) e pelo CPqD, que seriam os responsáveis pelo andamento dos trabalhos. Este comitê definiu 4 linhas de trabalho principais:

1. Modelo e sistema de regulação.
2. Conteúdo e programação.
3. Inclusão social e inclusão digital.
4. Tecnologia e desenvolvimento (onde este trabalho está inserido).

Os principais objetivos do comitê são:

1. Realizar a integração da TV Digital com a Internet, permitindo assim uma verdadeira inclusão digital.
2. Preservar a compatibilidade com os padrões existentes de TV analógica e realizar uma migração gradual.
3. Projetar e construir um terminal de acesso (*set-top box*) de baixo custo e totalmente nacional.
4. Permitir que haja interatividade do usuário com o sistema, via canal de retorno.
5. Respeitar o verdadeiro perfil do brasileiro usuário da TV analógica hoje existente, desenvolvendo um sistema intuitivo, motivador e, principalmente, de baixa complexidade para o usuário final.

Em 2004, a Financiadora de Estudos e Projetos (FINEP) emitiu várias Requisições Formais de Proposta (RFPs) às universidades, instituições e empresas privadas já envolvidas com algum tipo de pesquisa sobre a TV Digital no Brasil, endereçando os principais módulos do

sistema a ser construído, desde os métodos de codificação adotados até a definição do middleware a ser utilizado. Uma das preocupações também foi o módulo de **gerência de serviços**, que trata da gerência de segurança e autenticação no acesso a serviços do SBTVD.

Como é possível verificar, o Sistema Brasileiro de TV Digital está em fase inicial de definições e desenvolvimento, e esta pesquisa vem dar uma contribuição no sentido de propor uma arquitetura fim a fim de gerenciamento. Caso esta arquitetura de gerência seja incorporada desde o início, esta já será uma vantagem em relação aos outros sistemas existentes, que tiveram que incorporar módulos de gerência em ambiente de produção.

Entretanto, convém citar que esta pesquisa não possui vínculos com o SBTVD, deste modo determinando prioridades distintas deste onde necessário.

2.7 Convergência entre TV digital e outras redes

Os sistemas de TV Digital em uso atualmente têm como objetivo comum (porém não final) a convergência de redes. Não só em termos técnicos, mas principalmente em termos econômicos e operacionais esta convergência é bem vinda, pois preserva os investimentos anteriores e faz com que a utilização de uma rede a princípio projetada para uma finalidade específica extrapole estes limites iniciais, trazendo benefícios a fornecedores e usuários. Esta convergência de redes é também um dos motivadores principais para que a inclusão digital, realizada por meio do acesso à Internet através da TV digital, possa vir a ser uma realidade em países como o Brasil.

A convergência de redes também é fator essencial na construção de um sistema de gerência de redes de TV digital satisfatório, pois apenas deste modo interações poderão ser realizadas com os usuários e suas *set-top boxes*, seja esta interação completa ou apenas temporária.

3. GERENCIAMENTO DE REDES

3.1 Introdução ao gerenciamento de redes

Segundo (CARVALHO; GABOS, 2003), “gerenciamento é o processo de se controlar uma rede de modo a maximizar sua eficiência e produtividade”. Nenhuma rede hoje em produção pode viver sem um sistema adequado de gerência. A complexidade e o próprio custo de operação e manutenção das redes atuais fazem com que o investimento em gerenciamento seja prioritário para o oferecimento adequado de serviços de rede aos usuários finais. Atualmente, em operadoras de telecomunicações, é comum se ter uma rede dedicada para gerência dos equipamentos e serviços, independente da rede de produção (AIDAROUS; PLEVYAK, 2004). O caso de uma rede de TV Digital não é diferente, e as características desta rede e sua possível convergência com outras redes nos levam a crer que muitos desafios devam ser superados para que o gerenciamento da mesma seja realizado a contento.

Tradicionalmente, o gerenciamento de redes é dividido em 5 áreas principais, chamadas de **Áreas Funcionais de Gerenciamento** (STALLINGS, 1997). Estas áreas funcionais e seus objetivos são apresentados abaixo:

1. Gerenciamento de Falhas – Objetivos:

- detectar, reportar, isolar, corrigir e até mesmo antecipar falhas.
- manter logs de eventos de falha significativos.

2. Gerenciamento de Desempenho – Objetivos:

- monitorar e reportar métricas de desempenho da rede.

- alterar condições de desempenho que estejam prejudicando o bom funcionamento da rede, de seus elementos ou mesmo de serviços.

3. Gerenciamento de Contabilização – Objetivos:

- controlar a utilização dos recursos de rede.
- atribuir custos relativos à utilização de recursos da rede, informando aos usuários os custos resultantes de sua utilização.
- limitar o consumo de alguns usuários ou aplicações.
- delimitar escalas de tarifação.
- emitir relatórios sobre a utilização da rede.

4. Gerenciamento de Configuração – Objetivos:

- gerenciar o ciclo de vida do sistema e sua configuração associada.
- permitir que a troca de configuração de recursos de rede possa ser realizada para determinado fim (em caso de falha, melhoria de desempenho, atualização obrigatória, etc).
- coletar informações sobre a configuração da rede.

5. Gerenciamento da Segurança – Objetivos:

- gerenciar os mecanismos e procedimentos que garantem proteção aos recursos de rede.
- manter e manipular registros de segurança.

- realizar controle e armazenamento de chaves privadas e públicas utilizadas por seus usuários.

A primeira iniciativa a aplicar os princípios acima no gerenciamento de redes foi o desenvolvimento do **Modelo de Gerenciamento OSI** (STALLINGS, 1997). Este modelo define e conceitua o papel de gerentes e agentes, um banco de dados de informações de gerenciamento (orientado a objetos), um protocolo de comunicação de gerência e quais as interfaces que devem ser utilizadas por outras aplicações para o acesso às informações de gerenciamento. Todos estes conceitos são aplicados hoje em outras arquiteturas de gerenciamento.

Outra iniciativa pioneira é o Telecommunications Management Network (TMN), uma arquitetura de gerenciamento desenvolvida para atender aos complexos requisitos do gerenciamento de redes de telecomunicações (ALENCAR, 1999). Na verdade o TMN é uma rede à parte utilizada apenas para fins de gerenciamento. O TMN segue os mesmos princípios de gerenciamento citados acima porém propõe uma divisão em camadas funcionais do escopo do gerenciamento (ver figura 3.1):

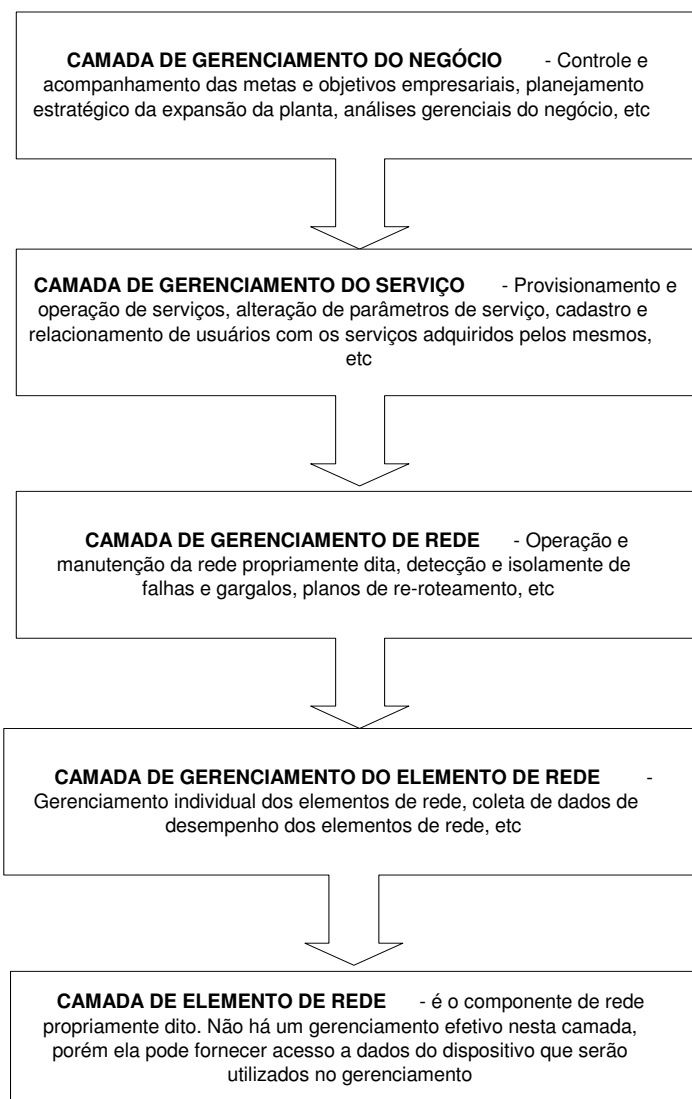


FIGURA 3.1 – CAMADAS FUNCIONAIS DE GERENCIAMENTO TMN (ALENCAR, 1999)

A definição destas camadas é importante pois isola a funcionalidade de cada módulo de gerenciamento, fazendo assim com que haja independência entre estes módulos.

Atualmente, várias arquiteturas de gerenciamento estão em uso ou em desenvolvimento. Muitas delas seguem uma tendência de descentralizar cada vez mais a função do gerente de redes, eliminando assim pontos de falha, melhorando o desempenho geral do sistema e transferindo inteligência aos agentes finais, que hoje contam cada vez com maior poder de

processamento e memória. Uma destas arquiteturas distribuídas, DME (Distributed Management Environment) (AUTRATA, 1992), é analisada mais adiante neste trabalho.

A seguir, arquiteturas importantes de gerenciamento são detalhadas. As arquiteturas aqui propostas foram utilizadas, em maior ou menor escala, como base para a definição da arquitetura proposta neste trabalho de pesquisa.

3.2 Arquiteturas de gerenciamento

3.2.1 Simple network management protocol - SNMP

A arquitetura SNMP (Simple Network Management Protocol) (CASE et al., 1988), (CASE et al., 1989), (CASE et al., 1990), (CASE et al., 1996), (LEVI, MEYER, STEWART; 1998), (BLUMENTHAL, WIJNEN; 1998), (WIJNEN, PRESUHN, McCLOGHRIE; 1998) é a mais utilizada atualmente para o gerenciamento de redes IP. Ela foi desenvolvida a partir de conceitos e princípios simples e também opera de modo simplificado. Este foi um dos principais motivos da sua rápida expansão. Aliás, esta expansão ocorreu não apenas pela sua simplicidade, pois o SNMP se difundiu acompanhando a expansão da própria Internet por ser o protocolo de gerenciamento proposto para esta rede e por pressões de mercado (como o próprio governo americano que exigiu das empresas as soluções de Internet).

O SNMP está em sua versão 3 e teve sua versão 1 aceita como padrão oficial do Internet Engineering Task Force (IETF) em Maio de 1990 (STD-15) (CASE et al., 1990), sendo que sua versão 2 foi publicada em Abril de 1993. Entretanto esta versão 2 não é utilizada, e foi substituída rapidamente pela versão 3. A versão atual (3) foi especificada em Abril 1999 e se tornou um padrão oficial do IETF (STD-62) em Dezembro 2002. Na verdade, o STD-62 engloba várias RFCs, todas relacionadas à versão 3 do SNMP.

O SNMP é uma arquitetura, porém também este é o nome dado para o **protocolo de gerência** utilizado nesta arquitetura, o que pode gerar confusão. Na verdade, sua arquitetura é formada pelos seguintes componentes:

- Gerente: controla centralmente as operações de gerenciamento e comunicação com os agentes. A partir do SNMPv2, é possível se ter comunicação entre gerentes, descentralizando um pouco a arquitetura.
- Agentes: entidades de software que rodam nos recursos a serem gerenciados e respondem ao Gerente a eventos de gerenciamento. Os Agentes também enviam comandos aos Gerentes indicando que uma situação limite foi atingida em determinado recurso.
- Base de Informação – MIB (Management Information Base): não orientada a objetos, é na verdade uma coleção de variáveis que descrevem o estado de recursos ou serviços da rede. As MIBs são definidas pelo Internet Engineering Task Force (IETF) e hoje existem mais de 100 delas (sendo que cada uma delas contém centenas de variáveis); as mesmas são globais, isto é, existe uma hierarquia global que deve ser respeitada onde fica determinada uma posição exata e um número de identificação para que cada nova variável projetada seja inserida. Existem MIBs para controles de dispositivos de Domain Name Servers (DNS), servidores de Hyper Text Transfer Protocol (HTTP), impressoras e até mesmo já existem MIBs para o gerenciamento de *set-top boxes* para TV Digital. (CABLEHOME QOS, 2003; CABLEHOME SECURITY, 2003; CABLEHOME ADDRESS, 2003; CABLEHOME CONFIGURATION, 2003; CABLEHOME DEVICE, 2003; CABLEHOME DIAGNOSTICS, 2003).
- Protocolo de Comunicação: é o SNMP propriamente dito. Este protocolo define primitivas de comunicação entre gerente e agente e transporta as informações de gerenciamento sobre User Datagram Protocol (UDP)/IP. O SNMP possui operações tanto de pesquisa de problemas como de requisição/resposta

Sendo a simplicidade sua principal vantagem, a arquitetura SNMP tem sua maior limitação na **segurança**. No SNMPv1, o acesso a MIB de um elemento é feito por uma senha simples, sendo que isto não foi alterado na versão 2. Já a versão 3 teve uma preocupação grande com o

fator segurança, e a mesma apresenta técnicas de criptografia de mensagens e um esquema de autenticação mais elaborado.

Por ser a mais difundida das arquiteturas de gerenciamento de redes, a arquitetura SNMP, apesar de não ser a mais adequada para alguns ambientes, não deve ser ignorada em nenhuma projeto de gerenciamento, para que, no mínimo, a compatibilidade com sistemas e produtos existentes seja preservada.

3.2.2 Web-based enterprise management - WBEM

O desenvolvimento da Internet foi um dos motivos, entre outros, que levaram ao desenvolvimento da arquitetura Web-Based Enterprise Management (WBEM), baseada em vários padrões utilizados na Internet. O Distributed Management Task-Force (DMTF) (<http://www.dmtf.org>), órgão responsável pelas especificações WBEM, primeiro desenvolveu esta arquitetura para gerenciar, via Web, elementos de uma rede corporativa (desktops, servidores, etc). Com o grande crescimento da Internet e conseqüentemente de sistemas Web, o WBEM, após as primeiras especificações datadas de 1997, cresceu e agora abrange uma solução completa de gerenciamento (THOMPSON, 1998). Hoje o DMTF é um órgão cujos participantes são as grandes empresas de informática e redes.

O objetivo principal do WBEM é realizar o gerenciamento de elementos via browser. Mais do que isto, tem como ponto forte a utilização de protocolos já utilizados na Internet (portanto bastante conhecidos) e também a alta integração de plataformas por conta do uso destes protocolos. Os principais pontos da filosofia WBEM são (THOMPSON, 1998):

- O uso de um modelo de dados completo e extensível, que permita conhecer o estado de um recurso e também manipulá-lo.
- Uma arquitetura do lado do cliente que permita que o mesmo possua alguma independência e inteligência sobre o processo de gerenciamento.

- A utilização de um protocolo de comunicação amplamente adotado, com possibilidade de transporte tanto de objetos como de operações (comandos) de gerenciamento.

Com estes princípios em mente, a arquitetura WBEM foi desenvolvida mantendo os conceitos de agente e gerente apresentados no SNMP, porém a mesma utiliza o HTTP como protocolo de comunicação (DMTF, 2004) e sua base de informação (MIB) (DMTF, 2005) é escrita em eXtensible Markup Language (XML), sendo que as operações WBEM também são escritas em XML e transportadas por HTTP (apesar de que pode haver outro mecanismo de transporte além do HTTP). Esta base de informação, em sistemas WBEM, é chamada de Common Information Model (CIM). A mesma, diferentemente das MIBs SNMP, é orientada a objetos, ou seja, possui mecanismos como herança, relacionamentos, definição de métodos, etc, o que representa um enorme ganho de inteligência no lado do agente, descentralizando deste modo a arquitetura. Para se ter acesso ao CIM e a outras operações de gerenciamento, o agente gerenciado incorpora uma entidade, chamada de Common Information Model Object Manager (CIMOM), que realiza a interface com o mundo externo. (THOMPSON, 1998). O CIM, apesar de apresentar várias inovações com relação às MIBs tradicionais, ainda é um dos pontos de resistência para a entrada de sistema WBEM no mercado, pois com o uso do CIM existe a necessidade de se reescrever as MIBs de vários equipamentos para CIM. Atualmente, conversores MIB de/para CIM existem para realizar este trabalho automaticamente, facilitando a transição.

Os objetos CIM são organizados através de uma estrutura chamada de CIM Schema. Este na verdade define uma hierarquia para a definição de objetos, e é dividido em 3 partes principais (THOMPSON, 1998):

1. Core schema: classes de mais alto nível com suas propriedades e relacionamentos.
2. Common schema: classes genéricas de elementos específicos (sistemas, dispositivos, redes, aplicações, etc). Não representam informações particulares a nenhum fabricante.
3. Extension schemas: classes que representam informações específicas de fabricantes

Caso o WBEM fosse usado para o gerenciamento de uma rede de TV Digital, deveriam ser desenvolvidos tanto um common schema genérico para os dispositivos que formam esta rede

(ex: *set-top box*) como também vários extension schemas, um de cada fabricante. Entretanto, uma grande parte dos CIM schemas já desenvolvidos poderiam ser aproveitados (DMTF, 2005).

Com relação aos mecanismos de segurança, o WBEM utiliza protocolos padrões da WEB, como Secure Socket Layer (SSL) e pode também apresentar mecanismos de autenticação e autorização com os utilizados na Internet; RADIUS por exemplo (RIGNEY et al, 1997).

Hoje existem várias implementações WBEM disponíveis, muitas delas de domínio público. Entre estas implementações, destacam-se :

- Pegasus (The Open Group) (<http://www.openpegasus.org>)
- WBEMServices (Sun) (<http://wbemservices.sourceforge.net>)
- IBM's SBLIM (<http://www.ibm.com>)
- Microsoft WMI (<http://www.microsoft.com>)
- Cisco's CiscoWorks2000 (<http://www.cisco.com>)

3.2.2.1 Uso do XML na definição de objetos de gerenciamento (CIM) em WBEM

O padrão WBEM também define que os objetos de gerenciamento (CIM) serão construídos através da linguagem XML. Isto representa uma vantagem em relação ao SNMP, pois até hoje não existe uma forma padrão no mesmo para armazenamento destes objetos, por isto este esforço de uniformização do DMTF. Na verdade, cada objeto de gerenciamento CIM é primeiramente definido por um XML Schema, e com este esquema em mãos, é possível descrever os parâmetros reais de gerenciamento propriamente ditos através da instanciação de documentos XML (ver figura 3.2).

```

<?xml version="1.0" encoding="UTF-8"?>
<snmp-data xmlns="http://www.ibr.cs.tu-bs.de/projects/libsmi/xsd/snmp-data"
  xmlns:IF-MIB="http://www.ibr.cs.tu-bs.de/projects/libsmi/xsd/IF-MIB"
  [...] >
  <context ipaddr="134.169.246.1" hostname="ciscobs.rz.tu-bs.de" port="161"
    community="public" time="2003-11-10T16:57:31Z">
    <IF-MIB:interfaces>
      <IF-MIB:ifNumber>10</IF-MIB:ifNumber>
    </IF-MIB:interfaces>
  [...]
    <IF-MIB:ifEntry ifIndex="2">
      <IF-MIB:ifDescr>FastEthernet0/0</IF-MIB:ifDescr>
      <IF-MIB:ifType>ethernetCsmacd</IF-MIB:ifType>
      <IF-MIB:ifMtu>1500</IF-MIB:ifMtu>
      <IF-MIB:ifSpeed>100000000</IF-MIB:ifSpeed>
      <IF-MIB:ifPhysAddress>00:03:fd:32:e4:00</IF-MIB:ifPhysAddress>
      <IF-MIB:ifAdminStatus>down</IF-MIB:ifAdminStatus>
    [...]
      <IF-MIB:ifName>Fa0/0</IF-MIB:ifName>
      <IF-MIB:ifLinkUpDownTrapEnable>enabled</IF-MIB:ifLinkUpDownTrapEnable>
    [...]
      <IF-MIB:ifRcvAddressEntry ifRcvAddressAddress="00:00:00:00:03:00">
        <IF-MIB:ifRcvAddressStatus>active</IF-MIB:ifRcvAddressStatus>
        <IF-MIB:ifRcvAddressType>other</IF-MIB:ifRcvAddressType>
      </IF-MIB:ifRcvAddressEntry>
    [...]
      </IF-MIB:ifEntry>
    [...]
      <IF-MIB:ifStackEntry ifStackHigherLayer="2" ifStackLowerLayer="0">
        <IF-MIB:ifStackStatus>active</IF-MIB:ifStackStatus>
      </IF-MIB:ifStackEntry>
    </context>
  </snmp-data>

```

FIGURA 3.2 – DOCUMENTO XML REPRESENTANDO UM OBJETO CIM (KLIE; STRAUB, 2004)

Além desta vantagem do WBEM, destacam-se ainda algumas das questões de gerenciamento que não foram resolvidas pela arquitetura SNMP como (KLIE; STRAUB, 2004):

1. Não existe um padrão aceito para o armazenamento otimizado dos objetos MIB SNMP. Com o WBEM utiliza XML, este armazenamento já é padronizado.
2. Não existe um padrão aceito para o processamento otimizado dos objetos MIB SNMP. Do mesmo modo como no problema do armazenamento, o XML (WBEM) possui padrões bem definidos para processamento.

3. A transferência de uma grande quantidade de dados é mais lenta que em outras arquiteturas devido às poucas opções de operações de coleta de dados disponíveis na arquitetura, sendo que o SNMP apresenta outros problemas de escalabilidade (YOON; JUN; HONG, 2003). No WBEM, mais operações de gerenciamento estão disponíveis que no SNMP.
4. O *overhead* das informações de gerenciamento é grande já que o SNMP não possui um mecanismo eficiente de descobrimento de informações dentro de uma MIB (muitas vezes uma MIB inteira deve ser lida para que um problema seja encontrado).
5. A informação contida em uma MIB apenas pode ser mostrada ao usuário por meio de uma aplicação proprietária. Já o XML pode lido por qualquer browser, e ainda com formatos de apresentação distintos, definidos em um documento eXtensible Stylesheet Language (XSL).
6. Não há uma solução amplamente aceita para a área funcional do gerenciamento de configuração, questão aliás ainda sendo discutida atualmente no IETF. No WBEM esta questão ainda não está resolvida, apesar de que os CIM Schema mais recentes (DMTF, 2005) tratem deste problema, ao declararem variáveis de gerenciamento de configuração.

Em muitos destes casos, a tecnologia XML e seus complementos oferecem uma solução satisfatória. Pela sua própria natureza, a mesma possui uma sintaxe e codificação bem estruturada e definida, o que facilita o desenvolvimento de soluções de armazenamento e processamento. Aplicações que realizam correlação de eventos de gerenciamento (através da análise de *traps* SNMP), cada vez mais comuns hoje em dia, poderiam se beneficiar do formato XML para auxiliar no descobrimento de problemas na rede.

Hoje, como foi citado anteriormente, o protocolo padrão para transporte de arquivos XML é o Hyper Text Transfer Protocol (HTTP). Para o caso do gerenciamento, têm-se as seguintes operações HTTP sendo utilizadas:

- HTTP GET – para leitura de dados de gerenciamento.

- HTTP POST – para envio de comandos aos agentes sendo gerenciados.
- O servidor HTTP recebe informações de *traps*.

Atualmente as aplicações de gerenciamento, sejam baseadas completamente em WBEM ou não, estão cada vez mais adotando o XML como padrão para armazenamento de objetos de gerenciamento (KLIE; STRAUB, 2004). Sem ignorar o grande legado deixado pelo SNMP, existem hoje várias implementações disponíveis de tradutores XML-SNMP, para que haja uma transição tranquila de um padrão para o outro, quando necessário. (KLIE; STRAUB, 2004; YOON; JUN; HONG, 2003).

Existem também várias soluções de tradução de documentos XML em objetos Java. Inicialmente não construídas especificamente para o uso em soluções de gerenciamento, as mesmas podem ser utilizadas sem problemas por implementações baseadas em Java. Atualmente a Java Architecture for XML Binding (JAXB) é a API mais indicada para realizar estas traduções (JSR 222, 2005).

3.2.3 Distributed Management Environment - DME

O DME (Distributed Management Environment) foi uma das iniciativas pioneiras em propor uma arquitetura de gerenciamento descentralizada (AUTRATA, 1992). Apesar da mesma não ter sido bem sucedida e não ser utilizada em larga escala hoje em dia, seus pontos principais serão analisados aqui pois serão úteis no desenvolvimento deste trabalho.

O DME foi desenvolvido pela Open Software Foundation (OSF), sendo um complemento à arquitetura Distributed Computing Environment (DCE) . Seu foco foi no desenvolvimento de uma plataforma padrão para a construção de soluções completas de gerenciamento. Um dos fundamentos da arquitetura DME é o fato da mesma ser orientada a objetos, porém, ao contrário da arquitetura OSI, estes objetos gerenciados possuem inteligência, deste modo otimizando o gerenciamento (AUTRATA, 1992). Esta utilização da tecnologia orientada a objetos, sendo estes objetos providos de **inteligência**, traz dois grandes benefícios aos usuários da tecnologia DME, na visão de seus autores (AUTRATA, 1992):

- Problemas menores podem ser resolvidos de forma específica e controlada, sem que estes se espalhem por outras ramificações do sistema.
- Objetos e seus respectivos métodos podem ser reutilizados. Após um longo trabalho inicial de montagem destes objetos, a reutilização dos mesmos reduz bastante o tempo de desenvolvimento de novos sistemas.

O DME fornece 4 serviços principais para o **desenvolvimento** de sistemas de gerência (AUTRATA, 1992):

1. uma interface com usuário – GUI, que fornece não apenas ferramentas para desenvolvimento, mas também uma visão integrada da rede sendo gerenciada com a utilização de visões distintas, domínios e construção de políticas.
2. serviços de objetos, para organizar cada aplicação em um conjunto consistente e bem definido de objetos distribuídos. Inclui-se aqui um Management Request Broker, utilizado para localizar objetos no sistema.
3. acesso a protocolos de comunicação de gerência como SNMP e CMIP (deste modo mantendo a arquitetura compatível com protocolos já em uso na época).
4. serviços de gerenciamento de eventos, para controlar os eventos e notificações de gerenciamento de uma forma segura e distribuída. Nestes serviços está previsto o controle do envio de eventos imediatamente ou no modo batch (após uma coleta ter sido realizada), o que pode ser extremamente útil no caso de redes de TV Digital, onde o canal de retorno não está sempre presente.

Para que se pudesse desenvolver aplicações de gerenciamento sobre a arquitetura DME, a OSF também disponibilizou, na época, um DME Development Toolkit, onde as APIs DME podem ser utilizadas através das linguagens C ou C++.

Apesar de todos estes pontos positivos, poucas ou quase nenhuma implementação da arquitetura DME foram desenvolvidas (SLOMAN et. al, 1993).

Além do DME, outras arquiteturas distribuídas foram propostas ao longo dos anos, entre elas (KANAHI, BEADLE):

- Distributed Big Brother – utiliza técnicas de inteligência artificial nos agentes para descentralizar a tarefa de gerência.
- Hierarchical Network Management – divide a tarefa de gerência em Gerentes, Agentes e Sub-Gerentes (estes controlando apenas um grupo pequeno de agentes).
- Management by Delegation (MbD) – onde tarefas de gerenciamento são **delegadas** a agentes da rede dinamicamente.

3.2.4 Java Management Extensions - JMX

A linguagem Java possui uma API específica para auxílio no desenvolvimento de aplicações de gerenciamento de redes. Chamada de JMX, a mesma fornece a capacidade de gerenciar qualquer aplicação Java, através da disponibilização de vários serviços exclusivos para este fim, tentando ao máximo manter a compatibilidade com as arquiteturas de gerenciamento existentes atualmente. Hoje a arquitetura JMX é compatível com SNMP, WBEM, Common Request Broker Architecture (CORBA) e Lightweight Directory Active Protocol (LDAP). Para fazer com que uma aplicação Java passe a ser gerenciável, a arquitetura JMX exige apenas a adição de 3 a 5 linhas de código por aplicação (JAVA MANAGEMENT EXTENSIONS, 2005). Obviamente, a arquitetura JMX exige o uso de uma máquina virtual Java (JVM) rodando no agente e também no gerente.

A arquitetura JMX é separada em 3 níveis (JSR3, 2005) (ver figura 3.3):

1. Nível de Instrumentação – realiza a comunicação com o recurso ou o objeto Java a ser gerenciado. Este recurso ou objeto (aplicação) Java a ser gerenciado é chamado de Managed Bean (Mbean), e é uma espécie de embrulho JMX que é aplicado sobre estes elementos. O interessante é que este embrulho JMX pode ser aplicado sobre aplicações

que não foram escritas em Java, preservando assim a compatibilidade com aplicações legadas ou de outra natureza.

2. Nível de Agente – um agente Java desenhado para rodar em um recurso gerenciável e fornecer serviços de gerenciamento. Um agente JMX possui um servidor Mbean (responsável pelos registros de Mbeans pertencentes a um recurso ou a uma aplicação), vários adaptadores de protocolo (responsáveis pela tradução de várias arquiteturas de gerenciamento em JMX (como SNMP ou XML/HTTP)) e vários conectores, que fornecem uma comunicação JMX (sobre HTTP/HTTPS) fim a fim.
3. Nível de Gerente – realiza o controle das operações de gerenciamento e agrupa os agentes de modo que o gerenciamento possa ser organizado do modo correto.

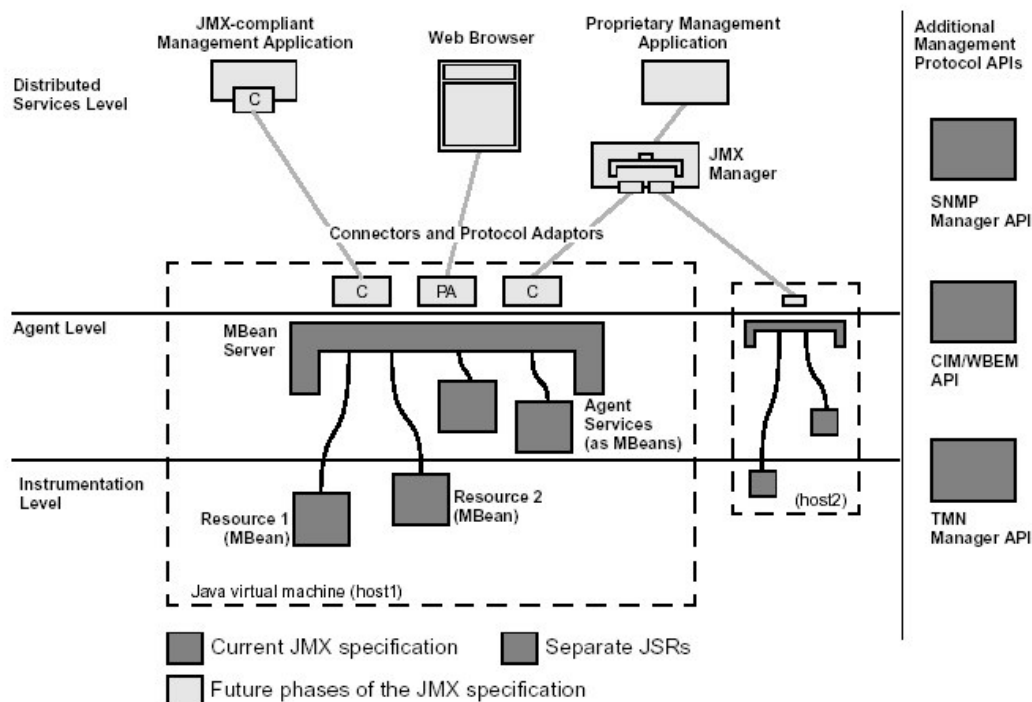


FIGURA 3.3 – ARQUITETURA JMX (JSR3, 2005)

O JMX ainda fornece alguns serviços básicos para serem utilizados por aplicações de gerenciamento, entre estes (lembrando que uma Mbean pode ser um recurso ou uma aplicação Java) (JAVA MANAGEMENT EXTENSIONS, 2005):

1. Registro de Mbeans.
2. Descobrimto de Mbeans.
3. Criação de relacionamentos e dependências entre Mbeans.
4. Funções de temporização.
5. Monitoração de atributos de Mbeans.
6. Segurança.
7. Descobrimto de agentes e gerentes.
8. Gerenciamento de políticas de rede.

Para auxiliar no processo de construção de aplicações de gerenciamento baseadas em JMX, e para manter a compatibilidade com a especificação, JMX fornece também ferramentas para testes de conformidade e uma implementação de referência, para que o desenvolvedor possa escrever a aplicação corretamente e para que ela possa ser portada sem problemas para outros ambientes (JAVA MANAGEMENT EXTENSIONS, 2005).

Atualmente a tecnologia JMX é utilizada por algumas aplicações de gerenciamento, sendo que existem também algumas poucas ferramentas para desenvolvimento (TELEFONICA, 2004). Apesar destas facilidades, da compatibilidade com arquiteturas existentes e da vantagem da mesma ser escrita em Java, a mesma ainda não apresenta um grande número de serviços de gerenciamento prontos para uso, o que dificulta bastante e aumenta o tempo de desenvolvimento de aplicações JMX.

3.2.5 Open services gateway initiative - OSGi

Uma das mais recentes iniciativas relacionada a gerenciamento de redes, voltada e criada para (não apenas) o gerenciamento de serviços, foi a criação da OSGi, uma entidade formada por vários fabricantes e centros de pesquisa que tem como missão “capacitar o desenvolvimento e a entrega de serviços de redes de longa distância até redes locais e dispositivos específicos” (MARPLES; KRIENS, 2001) (ver figura 3.4). Primeiramente focada na entrega de serviços aos chamados **gateways residenciais** (que também podem ser entendidos como terminais de acesso com uma razoável capacidade de processamento e memória), hoje a plataforma OSGi possui aplicações embarcadas tanto em automóveis como em celulares, desktops, entre outros (OSGI, 2004). Tanto isto é verdade que empresas como IBM, Siemens, Philips e Samsung já apresentaram propostas de produtos que utilizam a plataforma OSGi (OSGI, 2004).

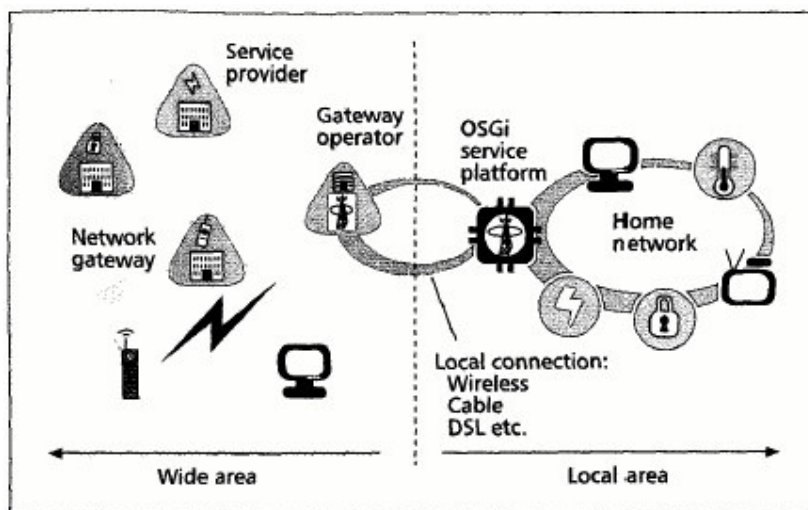


FIGURA 3.4 – GATEWAY DE SERVIÇOS OSGi (MARPLES; KRIENS, 2001)

Também baseada em Java, esta arquitetura necessita de uma JVM para rodar, sendo que o OSGi disponibiliza um controle bem definido do tempo de vida de cada aplicação, essencial em um ambiente com poucos recursos de hardware e que esteja executando aplicações sobre uma JVM.

As principais funcionalidades disponibilizadas pela plataforma OSGi são:

1. Gerenciamento dos componentes de software: realizado através do agrupamento lógico de várias aplicações que formam um único **serviço**. Este agrupamento é construído a partir de arquivos padrão Java Archive (JAR), que são chamados nesta plataforma de **Bundles**. Um *Bundle* é capaz de utilizar uma mesma JVM e pode ser iniciado, instalado, parado, atualizado ou destruído.
2. Gerenciamento remoto de componentes: realizado através da disponibilização de APIs compatíveis com as principais arquiteturas de gerenciamento, como SNMP e CIM/WBEM.
3. Ambiente de execução segura: auxiliado pelo uso dos mecanismos de segurança da própria JVM, da linguagem Java e das extensões de segurança do Java 2. Além disto, a plataforma OSGi consegue delimitar o uso de recursos por um determinado *Bundle*.
4. Interoperabilidade: pelo próprio uso da linguagem Java e de uma JVM.
5. Certificação de compatibilidade: realizado por um programa de certificação de aplicações OSGi.

A arquitetura OSGi possui elementos clássicos de aplicações Java (como a JVM), bem como elementos como carregadores de classes, controladores de classes e registradores de serviços (ver figura 3.5).

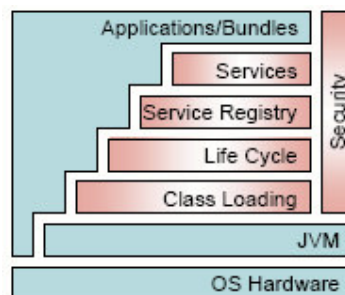


FIGURA 3.5 – ARQUITETURA OSGi (OSGI, 2004)

Outra vantagem desta plataforma é que a mesma já disponibiliza alguns serviços de gerenciamento prontos para uso, entre eles:

- Serviços de log.
- Serviços de gerenciamento de configurações.
- Administração de usuários.
- Serviço de HTTP.
- Compatibilidade com Universal Plug and Play (UPnP) (OSGI, 2004).
- XML Parser baseado no Java Application for XML Parsing (JAXP) (OSGI, 2004).

Alguns outros serviços ainda estão em desenvolvimento, sendo que a comunidade OSGi permanece muito ativa. Estes novos serviços oferecem soluções para as seguintes áreas funcionais de gerenciamento tradicionais (TELEFONICA, 2004): gerenciamento de configuração, falhas, segurança e contabilização.

Atualmente, a plataforma OSGi vem sendo muito apoiada e defendida por diversas vertentes de pesquisa, inclusive o Institution of Electrical and Electronical Enginners (IEEE) já publicou vários artigos sobre o tema (COCHINWALA, M; MOYER, S; SHIM, H.; 2005), (ISHIKAWA, H. et al.; 2004), (KANG, KYU-CHANG; LEE, JEON-WOO; 2004), (MARPLES, D; KRIENS, P.; 2001). Acredita-se que o grande número de serviços disponibilizados por esta arquitetura lhe rende uma grande vantagem, sendo que também o uso otimizado de recursos é outro fator positivo frente a outras soluções em meio a um cenário de gerenciamento de *set-top boxes* de baixo poder de processamento. Um porém fica por conta da compatibilidade com outras arquiteturas de gerenciamento, já que a própria especificação OSGi demonstra que este desenvolvimento ainda necessita de futuros ajustes.

3.2.6 Gerenciamento de serviços

Uma das principais tendências no âmbito das arquiteturas de gerenciamento é a monitoração e controle de aplicações, também chamadas de **serviços**. Atualmente, já não é mais possível propor uma solução que contemple apenas a gerência dos recursos de rede (hardware); é necessário também gerenciar (no sentido clássico da palavra) as aplicações que são consumidas pelos usuários.

Para (LEWIS; 2000), uma arquitetura de gerenciamento de serviços deve ter a mesma preocupação que uma arquitetura tradicional (voltada apenas a recursos), ou seja, o objetivo primário da gerência de serviços deve ser “assegurar que as aplicações sejam entregues aos clientes com custo reduzido e com o nível de qualidade esperado, deste modo também capacitando o provedor a introduzir novos serviços rapidamente na rede” (LEWIS; 2000). Qualquer arquitetura de gerenciamento de serviços também deve apresentar uma grande flexibilidade para inclusões, modificações e exclusões (LEWIS; 2000), devido à própria natureza dinâmica dos serviços atualmente oferecidos.

No que tange ao gerenciamento propriamente dito, é consenso entre a maioria dos autores afirmar que um serviço, para ser gerenciável, necessita que o mesmo já possua funções de gerenciamento implícitas (ou seja, código de gerenciamento, mesmo que mínimo, deve ser inserido nas aplicações). As arquiteturas JMX e OSGi foram construídas seguindo este princípio.

Para a modelagem de uma arquitetura que gerencie efetivamente serviços, é interessante atentar para o ciclo de vida de um serviço (GARSCHHAMER et. al; 2001) (ver figura 3.6); com ele, temos mais clareza para a definição das funções de gerenciamento necessárias para um determinado serviço.



FIGURA 3.6 – CICLO DE VIDA DE UM SERVIÇO (GARSCHHAMER et. al; 2001)

Portanto neste ciclo de vida as seguintes etapas são definidas:

1. **Projeto:** nesta fase o serviço é construído (pela primeira vez), sendo que o mesmo é alocado em alguma “prateleira” para que os clientes possam adquiri-lo. Para o sistema de gerência, é necessário saber qual perfil de usuário que pode acessá-lo, quais os parâmetros de sua configuração, quanto o mesmo consumirá de recursos de rede, qual seu custo, etc.
2. **Negociação:** no momento em que o usuário deseja obter um serviço, o mesmo, que está na “prateleira” acima definida, deve ser instanciado. No entanto, nesta fase de negociação, vários parâmetros devem ser definidos e armazenados no sistema de gerência (para efeitos de análise de desempenho ou mesmo para contabilização): tarifas, valores de variáveis de QoS, limites para o uso, responsabilidade em caso de falhas, etc. Ou seja, está sendo definida aqui a **configuração** de um serviço de acordo com o negociado com um cliente.
3. **Instalação:** a entrega e instalação propriamente dita do serviço ao usuário. Para o sistema de gerência, interessam aspectos de desempenho, falhas e segurança, sendo que os

recursos necessários para a entrega devem apenas ser alocados de acordo com os requisitos do usuário final.

4. Utilização: onde são definidos dois estados, de operação, na qual o sistema de gerência deve ser responsável por manter o serviço rodando de acordo com os parâmetros definidos; e também o estado de atualização, onde o sistema de gerência identifica que mudanças devem ser realizadas no serviço para que, por exemplo, haja melhoria do desempenho.
5. Desinstalação: momento no qual os sistemas de contabilização da gerência são atualizados.

Portanto, um sistema de gerência de serviços deve acompanhar todos os serviços, um a um, desde sua construção até sua desinstalação, passando pela instanciação e entrega ao cliente. Como era de se esperar, a tarefa é complexa, e atualmente não há uma solução aceita por todos os pesquisadores ou segmentos de mercado (como o SNMP por exemplo). Ainda existem muitos problemas, entre eles cita-se (GARSCHHAMER et. al; 2001):

- Definição de um modelo de informação comum a ser entendido por provedor e cliente – o MPEG-21 pode ser uma solução, ao menos para a descrição de conteúdo dos serviços de vídeo (BURNETT, 2003).
- Mapeamento e definição de parâmetros de QoS para melhoria do desempenho – esse problema aparece pelo próprio número de soluções de QoS hoje existentes.

Atualmente, existem algumas propostas de arquiteturas de gerenciamento de serviços (LEWIS; 2000), como a do Telecommunications Information Networking Architecture (TINA) Consortium. Baseado na arquitetura CORBA, este consórcio formado por operadoras de telecomunicações e fabricantes de equipamentos tem por objetivo especificar uma arquitetura aberta que possa gerenciar serviços de telecomunicações em ambientes que comportam vários fabricantes e várias operadoras. Nesta, os serviços são descritos através de objetos, e são definidos os métodos para projeto, provisionamento e utilização dos mesmos. Como a maioria das propostas, a arquitetura TINA baseia-se na orientação a objetos e no

reuso de componentes de software (algo ainda não realizado do modo automático como se realmente esperava anteriormente).

Este trabalho considera esta complexidade da gerência de serviços porém tenta fornecer soluções simples utilizando protocolos existentes para resolver este problema no ambiente específico de TV digital.

4. GERENCIAMENTO DE REDES DE TV DIGITAL

Nesta seção são apresentadas algumas soluções de gerenciamento para redes de TV Digital, sejam elas arquiteturas totalmente fim a fim, direcionadas às *set-top boxes*, serviços ou que enderecem apenas alguns pontos de gerenciamento.

São analisadas arquiteturas proprietárias e abertas, que atendem redes de TV digital a cabo, terrestres ou via satélite.

No final do capítulo, uma análise crítica destas soluções é apresentada.

4.1 Gerenciamento da transmissão de TV digital no sistema EUTELSAT

A EUTELSAT é uma das maiores empresas operadoras de redes de satélite do mundo. Operando hoje na Europa e Ásia, transmite mais de 500 canais de TV digital simultaneamente através de sua rede (BARDELLI et. al, 2000).

Antes apenas uma rede utilizada para transmissão de sinais de TV analógicos, em 2000 a EUTELSAT começou a operar com sinais digitais (utilizando MPEG-2 DVB). Deste modo um novo desafio estava sendo iniciado, e o gerenciamento antes utilizado teria que ser revisado para se adequar a este novo sistema.

Dois pontos importantes foram inicialmente observados como as principais diferenças entre o antigo sistema, analógico, e o novo, digital: um aumento significativo no número de canais transmitidos e a necessidade de se monitorar novos e diferentes parâmetros de qualidade da transmissão (BARDELLI et. al, 2000). Estes parâmetros foram inicialmente definidos pelo documento da European Telecommunications Standard Institute (ETSI) ETR290, intitulado “Measurement Guidelines for DVB Systems” (ETSI, 1997). Utilizados em produção pela EUTELSAT, as medições sugeridas pela ETSI não trouxeram a melhoria na qualidade da

transmissão esperada, fazendo com que a mesma desenvolvesse seus próprios parâmetros de qualidade, o que evidencia a dificuldade em lidar, monitorar e garantir níveis de qualidade da transmissão digital.

O primeiro ponto que a EUTELSAT considerou é que as monitorações para a qualidade do sinal digital transmitido deveriam ser realizadas portadora por portadora, e não seguir um padrão comum definido para todo o sistema, como proposto pelo documento da ETSI. Os parâmetros de interesse para monitoração foram definidos como os seguintes, entre outros:

- Perda de pacotes.
- Erros de informação de Packet Identifier (PID).
- Erros nas tabelas Program Specific Information (PSI/SI).
- Erros de Program Clock Reference (PCR) e Presentation Time Stamps (PTS), que revelam problemas de sincronismo.

Os limites de cada um destes parâmetros seriam definidos manualmente pelo operador, portadora a portadora, com os valores que mais se adequassem a cada uma delas; valores estes verificados empiricamente pelo operador. Alarmes seriam disparados para o sistema de gerência se algum destes valores limite fosse ultrapassado.

Após estas definições e da constatação de que a solução poderia ser aplicada em um ambiente real de produção, a EUTELSAT decidiu unificar todos os seus sistemas de gerência e monitoração, fossem eles legados de sistemas analógicos ou novas soluções aplicadas a sistemas digitais. Esta unificação gerou o chamado Television Surveillance System for EUTELSAT Satellites (TESEUS) (ver figura 4.1). Este sistema foi desenvolvido pela Siemens da Áustria e entrou em operação no final do ano 2000 (BARDELLI et. al, 2000).

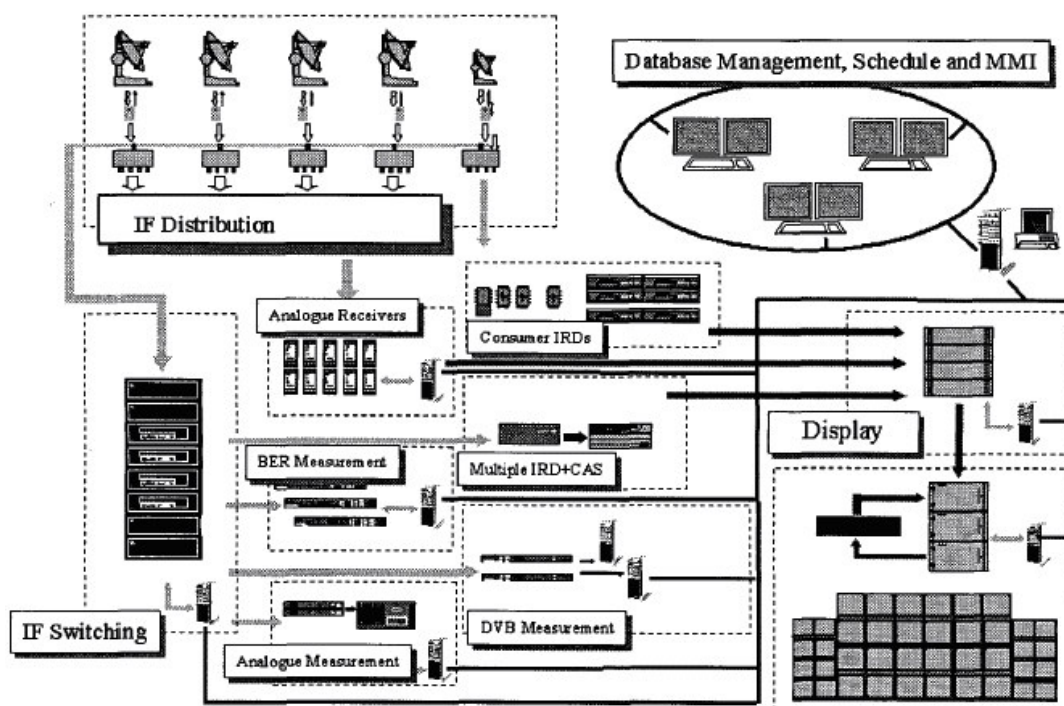


FIGURA 4.1 – ARQUITETURA DO SISTEMA DE MONITORAÇÃO TESEUS (BARDELLI et. al, 2000)

Como evidenciado na arquitetura acima, neste sistema os sinais tanto digitais como analógicos (acima à esquerda na figura 4.1) são recuperados por diversas antenas que atuam como corpos de prova. Com a recuperação destes sinais, vários parâmetros são medidos (indicados como *measurements* na figura 4.1), e após isto é realizada uma análise dos valores encontrados nestas medições, análise esta que é enviada ao centro de gerência da EUTELSAT (indicado do lado direito da figura 4.1). O sistema de gerência é responsável por manter uma base de dados atualizada indicando as condições de operação de cada portadora e também dispara alarmes quando um limiar é atingido. Este sistema também demonstra ser flexível pois permite a inserção de novos limites e mesmo de novos parâmetros de monitoração para cada portadora. A criação de relatórios e uma interface gráfica poderosa e de fácil utilização são outros pontos fortes do mesmo. (BARDELLI et. al, 2000).

Como pode ser notado, a arquitetura proposta pela EUTELSAT é voltada à manutenção da qualidade da transmissão dos sinais digitais e à monitoração dos elementos que transportam estes sinais, sendo que a mesma monitora itens específicos do padrão MPEG-2 TS, não tendo

como objetivo a gerência de elementos de mais alto nível (como por exemplo serviços), ou mesmo da *set-top box* presente na casa do usuário.

4.2 Gerenciamento de *set-top boxes* baseadas em Java no sistema ATSC

O sistema americano ATSC possui como tecnologia de middleware o DASE (ATSC, 1999), conforme já citado neste trabalho. Este permite que aplicações Java possam ser “baixadas” da operadora (ou provedor) e possam ser executadas deste modo na *set-top box*. Mais do que isto, este middleware (baseado em uma Java Virtual Machine) pode ser utilizado para rodar alguma aplicação que gerencie a própria *set-top box* e as aplicações que rodam na mesma (MEANDZIJA, 2000).

Esta aplicação de gerenciamento foi então realmente conceituada, definida e implantada por um grupo especial de pesquisa do sistema ATSC (ATSC Technical Committee T3 Specialist Group S17). Este grupo definiu quais as finalidades e os requisitos para o desenvolvimento de uma arquitetura que contemplasse o gerenciamento de recursos e aplicações no ATSC.

Foi então deste modo definido um modelo de estado para cada aplicação rodando na *set-top box*, e também para cada recurso de hardware. Neste modelo, que é baseado na especificação (ITU-T, 1992), o estado de uma aplicação e/ou um recurso é coletado e apresentado pelo uso de 3 variáveis de estado e várias variáveis de status, que são apresentadas na tabela abaixo (ver tabela 4.1- onde são apresentadas todas as variáveis de estado e algumas variáveis de status):

TABELA 4.1 – VARIÁVEIS DE ESTADO E STATUS DA ESPECIFICAÇÃO ITU-T X.731

| Estado Administrativo | Estado Operacional | Estado de Utilização | Status de Alarme | Status de Disponibilidade | Status da Aplicação |
|-----------------------|--------------------|----------------------|---------------------|---------------------------|--------------------------------|
| <i>locked</i> | <i>enabled</i> | <i>idle</i> | <i>under repair</i> | <i>in test</i> | <i>initialization required</i> |
| <i>unlocked</i> | <i>disabled</i> | <i>active</i> | <i>critical</i> | <i>failed</i> | <i>not initialized</i> |
| <i>shutting down</i> | | <i>busy</i> | <i>major</i> | <i>power off</i> | <i>initializing</i> |
| | | | <i>minor</i> | <i>offline</i> | <i>reporting</i> |
| | | | <i>outstanding</i> | <i>off duty</i> | <i>terminating</i> |
| | | | | <i>dependency</i> | |
| | | | | <i>degraded</i> | |
| | | | | <i>no installed</i> | |
| | | | | <i>log full</i> | |

As variáveis de estado indicam informações “macro” do recurso sendo gerenciado, e as variáveis de status apresentam informações mais detalhadas. Esta é a única diferença entre estas.

Estas informações representam o esboço de um modelo de informações mais completo que poderia ser usado para viabilizar a gerência de recursos e aplicações em qualquer sistema de TV digital. A grande vantagem deste modelo é manter aplicações e recursos utilizando um mesmo conjunto de variáveis, o que facilita muito o processamento e armazenamento de informações.

Este modelo de informação pode ser muito útil na construção de uma arquitetura de gerenciamento de TV digital que contemple as 5 áreas funcionais de gerenciamento (configuração, desempenho, falhas, segurança e contabilização), englobando tanto a monitoração de recursos da *set-top box* como também das aplicações que executam sobre a mesma.

Esta aplicação de gerenciamento foi implantada na arquitetura ATSC/DASE em pequena escala, utilizando o canal de retorno como canal de comunicação entre gerente e agente (MEANDZIJA, 2000).

4.3 Arquitetura de gerenciamento do sistema DAVIC

A organização DAVIC (digital audio visual council) é uma entidade sem fins lucrativos, baseada na Suíça, e que possui hoje mais de 200 membros (<http://www.davic.org>).

O sistema DAVIC é um conjunto de especificações e padrões destinados a alcançar a interoperabilidade fim a fim de plataformas que executem serviços de áudio e vídeo transmitidos via *broadcast*. Deste modo, estes padrões podem ser utilizados em sistemas de TV digital para a entrega de conteúdo ao usuário final e também para possibilitar a interatividade com este mesmo usuário. Estes padrões definem, entre outros pontos (YASUDA; RYAN, 1998):

1. Quais as funcionalidades que um sistema digital de áudio e vídeo deve fornecer.
2. Arquiteturas de referência do provedor, da rede de distribuição e também do sistema presente na casa do usuário.
- 3. Arquitetura de gerenciamento.**
4. Protocolos de camadas baixas e definição de interfaces físicas.
5. Protocolos de mais alto nível.
6. Representação de informação.
7. Ferramentas básicas de segurança.
8. Ferramentas para verificação de conformidade e interoperabilidade.

A pilha de protocolos DAVIC pode realizar encapsulamento de áudio/vídeo/dados diretamente sobre IP ou também sobre MPEG-2 TS. Entretanto, na mesma é sugerido o transporte pela rede Asynchronous Transfer Mode (ATM) (ver figura 4.2).

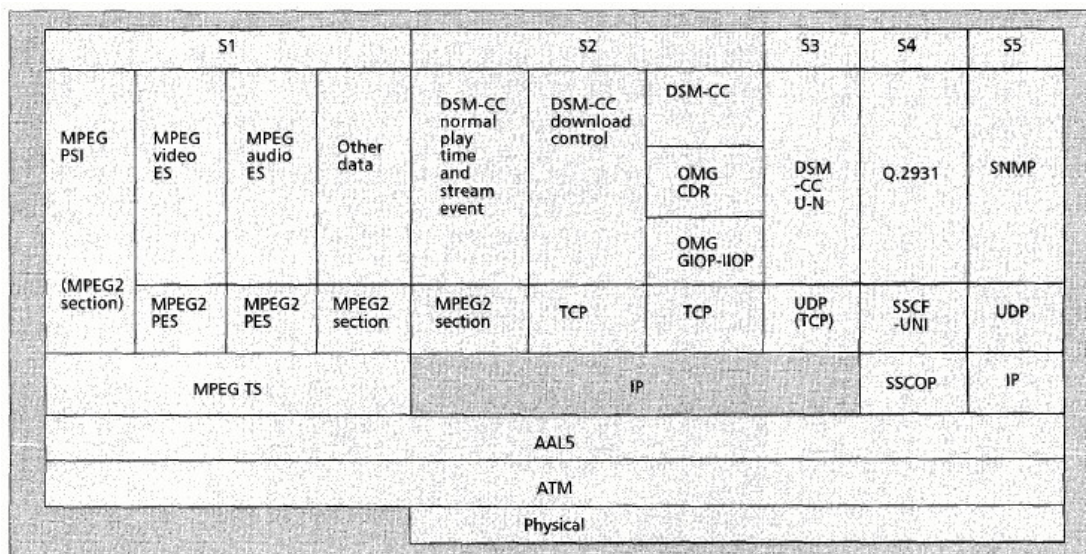


FIGURA 4.2 – PILHA DE PROTOCOLOS DAVIC (YASUDA; RYAN, 1998)

Atualmente existem várias implementações do sistema DAVIC, entre elas destacam-se a Time Warner's cable-based interactive TV (Pegasus), o NEC France CSI VOD system, o NEC Hyper-MS VOD systems e o Panasonic's Vídeo Server (PanaViSS).

Com relação à arquitetura de gerenciamento propriamente dita (DAVIC, 1998), a mesma encerra as seguintes funções:

1. Administração automatizada
2. Monitoração de falhas
3. Contabilização
4. Manutenção

Por ser uma arquitetura fim a fim, estas funções são direcionadas a 3 entidades: ao gerenciamento da rede do provedor, do distribuidor (aqui se utiliza a nomenclatura DAVIC; na verdade esta pode ser vista como a rede de acesso) e do sistema presente na casa do usuário. Segundo a especificação, os protocolos de gerenciamento que podem ser utilizados

são SNMP ou CMIP (DAVIC, 1998). A especificação de 1998 considera também que a tendência é que sejam desenvolvidas também soluções baseadas em WBEM e mesmo CORBA.

Para o desenvolvimento das funções de gerenciamento, foram considerados os conceitos clássicos OSI e TMN, sendo que cada módulo de gerenciamento definido na especificação é direcionado a resolver um problema específico de uma área funcional de gerenciamento em uma determinada camada funcional TMN (mesma filosofia adotada nesta pesquisa).

A especificação DAVIC propõe vários cenários de rede onde sua solução de gerenciamento poderia ser aplicada. Na verdade, são definidos cenários distintos onde a rede do provedor pode pertencer a uma entidade, a rede do distribuidor à outra entidade e o sistema presente na casa do cliente à também uma outra entidade. Outros cenários descrevem a utilização da arquitetura de gerenciamento onde todos os elementos pertencem à mesma entidade. O proposto é, portanto, a possibilidade da divisão das tarefas de gerenciamento.

É definido ainda nesta especificação o modelo de informação de gerenciamento (MIB), apesar do mesmo ainda estar incompleto.

Como pode ser observado, a arquitetura DAVIC é realmente uma arquitetura fim a fim, pois propõe o gerenciamento do provedor de serviços multimídia, da rede de acesso (distribuidor na nomenclatura DAVIC) e da *set-top box* presente na casa do usuário final (DAVIC, 1998). Bem definidos, o sistema contempla vários cenários de utilização, porém peca ao não englobar ainda uma solução de WBEM e outra solução que gerencie mais especificamente os serviços que rodam na *set-top box* do usuário final. Tomando como base o item 2.2.2.6 deste trabalho (que tange ao gerenciamento de serviços), nota-se que a especificação DAVIC foca apenas no gerenciamento das fases de projeto e negociação de serviços, não contemplando as outras fases da entrega de serviços ao usuário final. Além disto, o sistema não prevê uma possibilidade de expansão que englobe o gerenciamento de uma rede local residencial via *set-top box*, função esta que pode ser necessária daqui a algum tempo.

4.4 Gerenciamento de redes européias de TV digital a cabo

Como já citado neste trabalho, a rede européia EUTELSAT sofreu uma grande mudança na sua estrutura de gerenciamento ao passar do sistema analógico para o digital. Não diferente, as redes a cabo da Europa também começaram a conviver com estas mudanças há aproximadamente 5 anos. Este fato levou a uma remodelagem dos sistemas de gerenciamento destas redes, afinal, além das novas possibilidades de entrega de conteúdo, serviços digitais interativos começaram a ser desenvolvidos, e isto tudo precisava ser monitorado.

Uma das principais iniciativas neste sentido foi a do projeto Integrated Broadband Communication over Broadcast Networks (IBCoBN). Este projeto tem como integrantes as principais operadoras de cabo da Europa e visa oferecer soluções a esta mudança do mundo analógico para o digital. Na verdade, este projeto foi uma reação européia à criação do americano CableLabs (detalhado adiante neste trabalho), sendo chamado até mesmo de EuroCableLabs no início de sua existência. Este EuroCableLabs foi formado por empresas de telecomunicações de Portugal, Suécia, Bélgica, Inglaterra, Alemanha, Rússia entre outros.

Esta iniciativa tenta tirar proveito principalmente das novas funcionalidades oferecidas pelo sistema digital, em especial do uso do canal de retorno. Além disto, foi necessária a ampliação da visão de gerenciamento para considerar não apenas os elementos de rede ou a rede propriamente dita, mas também os serviços oferecidos por esta rede (BHATTI; JNIGHT, 1998). Os aspectos de segurança e de gerenciamento da segurança (na autenticação do usuário, controle de acesso, pagamentos, etc) também foram pontos importantes para o projeto desta arquitetura.

Na iniciativa IBCoBN, a pilha de protocolos utilizada é a IP (ver figura 4.3). Como possibilidade de protocolos de gerenciamento, é proposto o uso de SNMP, TINA-C ou TMN, sendo que é considerado o uso destas 3 tecnologias simultaneamente, deste modo fazendo com que cada uma possa trabalhar onde melhor se encaixe.

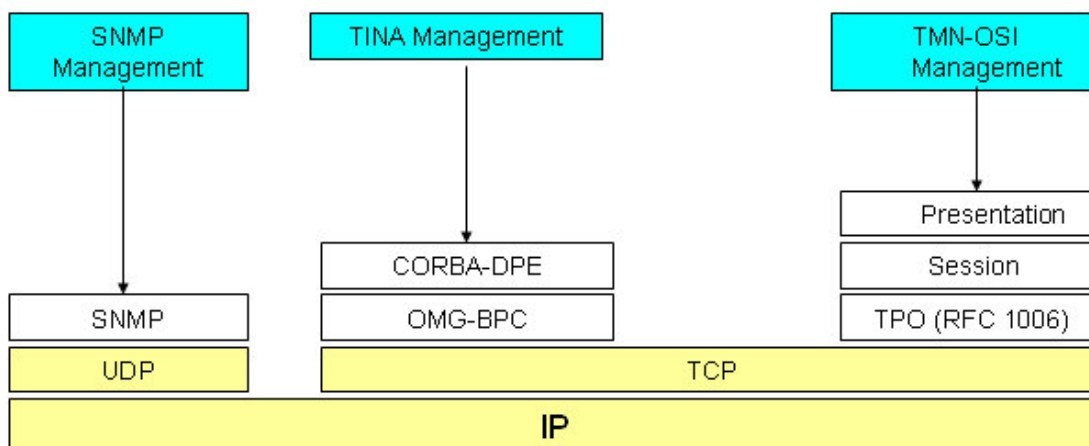


FIGURA 4.3 – PILHA DE PROTOCOLOS DE GERENCIAMENTO PROPOSTA POR IBCoBN (BHATTI; JNIGHT, 1998)

A iniciativa do IBCoBN não gerou efetivamente especificações ou padrões para a definição do gerenciamento em redes de TV a cabo européias, mas é apresentada neste trabalho por considerar tendências como o uso de IP, da necessidade de um canal de retorno (neste caso permanente), da obrigatoriedade do gerenciamento de serviços e da coexistência de várias soluções de gerenciamento hoje existentes para que o controle da rede de TV a cabo, que passa do mundo analógico ao mundo digital, possa ser realizado de maneira satisfatória.

4.5 Arquitetura de gerenciamento do projeto CableHome da CableLabs

O projeto CableHome, desenvolvido pela CableLabs (Cable Television Laboratories, Inc. – uma entidade americana dedicada à pesquisa de novas tecnologias para redes de TV a cabo), tem como objetivos principais:

1. Definir mecanismos que permitam a interoperabilidade entre diversas *set-top boxes*.
2. Propor padrões de qualidade de serviço (QoS) que garantam a entrega do nível de serviço desejado pelo cliente.

3. Propor uma arquitetura de gerenciamento para redes de TV a cabo e para redes residenciais suportadas pela *set-top box*.

Na verdade, todos estes objetivos visam, em última instância, à possibilidade do oferecimento de novos serviços aos usuários finais.

Na arquitetura CableHome, a *set-top box* recebe o nome de Home Access device (HA). Na visão do CableLabs, ela seria uma futura gerenciadora de serviços dentro de uma *home network* (CABLELABS, 2005). Apesar de ser uma arquitetura não voltada exclusivamente a redes de TV Digital, é apresentada aqui pois fornece uma boa visão dos requisitos necessários para o gerenciamento de *set-top boxes*.

As funções de gerenciamento identificadas pelo grupo como essenciais ao suporte para oferecimento destes novos serviços são:

1. Gerenciamento de servidor DHCP que distribui IPs para toda a rede (na arquitetura CableHome toda a comunicação é realizada via IP).
2. Gerenciamento e análise de eventos disparados pelas *set-top boxes* aos sistemas gerenciadores.
3. Gerenciamento dos serviços de resolução de nomes.
4. Gerenciamento do horário configurado em cada *set-top box*.
5. Gerenciamento das funções de Universal Plug and Play (UPnP) em cada *set-top box*, e sua respectiva rede residencial, para descobrimento automático de novos elementos que se conectem a esta rede residencial.
6. Gerenciamento de falhas que possam ocorrer com os recursos da *set-top box*.
7. Gerenciamento do desempenho de recursos da *set-top box*.

O protocolo de gerenciamento utilizado é o SNMP, e há uma extensa definição de MIBs realizada pelo CableLabs para definir o modelo de informação deste gerenciamento. Ferramentas de diagnóstico e controle remoto também são especificadas.

A arquitetura de gerenciamento CableHome possui 4 elementos principais (ver figura 4.4):

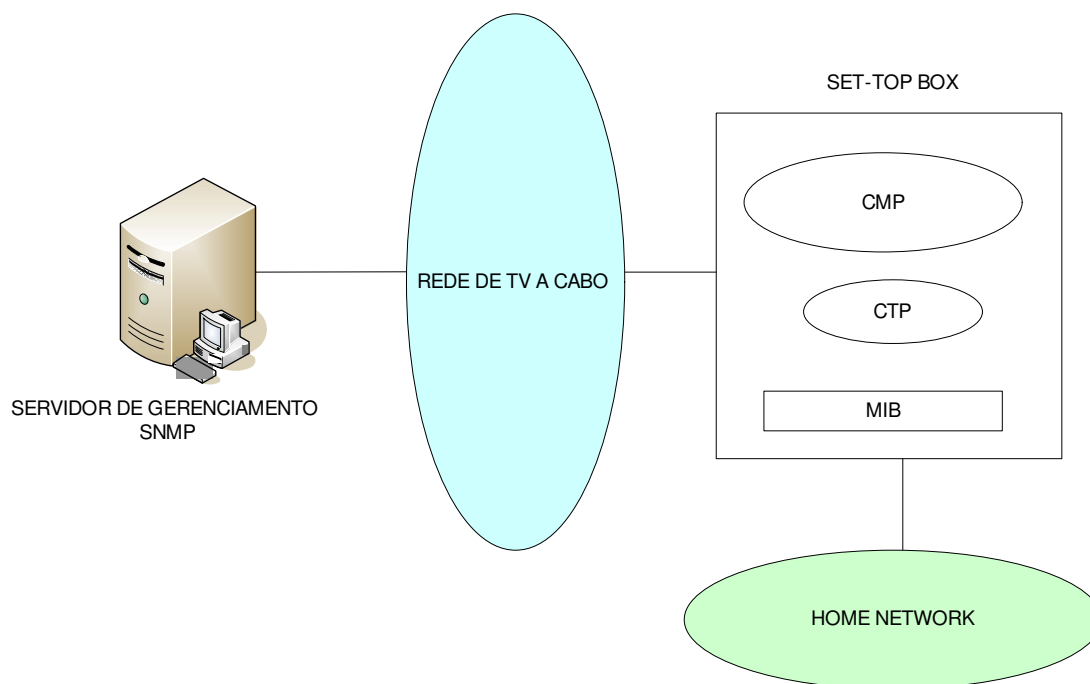


FIGURA 4.4 – ARQUITETURA DE GERENCIAMENTO CABLEHOME

1. O chamado CableHome Management Portal (CMP), residente na *set-top box*, que é responsável por realizar a comunicação com o gerente, permitindo assim o envio de eventos assíncronos ao mesmo e também possibilitando a configuração remota da *set-top box*. Este é o elemento que também realiza a comunicação com os elementos da home network, caso esta esteja presente. O CMP é na verdade um agente SNMP, sendo portanto o elemento de maior importância nesta arquitetura.

2. O chamado CableHome Test Portal (CTP), também residente na *set-top box*, e que é um módulo específico para gerenciamento de falhas e desempenho de recursos desta *set-top box*.
3. Um sistema gerente baseado em SNMP.
4. As MIBs SNMP, que foram projetadas para fornecer informações sobre parâmetros de qualidade de serviço, de segurança e de configuração da *set-top box*, bem como do estado dos próprios módulos CTP e CMP. Além disto, outras MIBs padrão, como por exemplo a MIB IP (RFC 2011) são utilizadas nesta arquitetura.

Como pode ser notado, a arquitetura de gerenciamento CableHome é voltada ao gerenciamento tradicional de recursos de rede, utilizando a tecnologia IP e também o SNMP. Não apresentando uma solução completa de gerenciamento de serviços/aplicações e dependente da infra-estrutura de IP, a arquitetura, não escalável, limita-se a definir MIBs e mecanismos de controle e diagnóstico remoto, que são úteis porém que não atendem totalmente aos requisitos de gerenciamento de redes de TV digital.

4.6 Gerenciamento proposto pelo Governo Brasileiro para o sistema brasileiro de TV digital

Como já citado anteriormente neste trabalho, o Governo Brasileiro emitiu, em 2004, Requisições Formais de Proposta (RFPs) onde são definidos o escopo de atuação e as várias linhas de pesquisa para o desenvolvimento e implantação da TV digital no Brasil (FINEP, 2004a). Na verdade, são definidos nestes documentos apenas os requisitos, sendo que as respectivas arquiteturas e especificações serão desenvolvidas baseadas no atendimento ou não destes requisitos.

Mais especificamente, uma das RFPs é direcionada diretamente ao gerenciamento da rede de TV digital, mais precisamente ao desenvolvimento de uma arquitetura de gerenciamento de serviços para o sistema de TV digital modalidade terrestre.

Como parte dos requisitos citados para gerenciamento de serviços, citam-se a necessidade de um controle de registro de uso (contabilização) e a indicação de quais métodos serão utilizados para garantir a segurança tanto das informações armazenadas na *set-top box* (para que não sejam utilizadas por quem não adquiriu o serviço) como as transportadas pela rede.

O controle de acesso é baseado na construção de perfis, onde cada usuário estaria mapeado, deste modo determinando suas permissões para a obtenção de serviços dentro do sistema. É opcional o uso de várias técnicas de autenticação de usuários (como por exemplo o uso de *smart cards*), sendo que a forma de autenticação utilizada não deve ser um limitante para o sistema e o usuário deve ter autonomia para selecionar o tipo de autenticação que mais lhe é adequada (lembrando que o hardware da *set-top box* deve estar preparado para o suporte destas técnicas). O uso do canal de retorno também é permitido, seja ele permanente ou intermitente. Um outro requisito deste projeto é a possibilidade de bloqueio de acesso a algum programa (serviço), devido à classificação indicativa (acesso condicional) do mesmo, bloqueio este configurado pelo próprio usuário final localmente em seu terminal de acesso.

Um outro ponto importante é a geração de relatórios de utilização de serviços para cada usuário, sendo que é solicitado um padrão único a ser utilizado para os registros de utilização. Não é considerado o uso da pilha de protocolos IP na *set-top box*, seja no canal de *broadcast* ou no canal de interatividade (FINEP, 2004a).

Nesta RFP, o Governo Brasileiro preocupa-se com pontos importantes de uma arquitetura de gerenciamento, como sistema de autenticação e segurança, voltando-se também para o âmbito do gerenciamento de serviços. Entretanto, para que uma proposta de gerenciamento possa ser levada realmente a cabo, acredita-se que outros aspectos deveriam ser pesquisados, como por exemplo controles de falha e desempenho de toda a rede e também do terminal de acesso presente na casa do usuário final. Devido ao grande número esperado de terminais de acesso, uma arquitetura escalável também é obrigatória, requisito este não citado nas especificações.

4.7 Panorama geral dos sistemas de gerenciamento para TV digital

Com o estado da arte apresentado, fica clara a direção que as pesquisas e as soluções estão tomando para resolver o problema do gerenciamento de TV digital. Também é notada a falta de especificações padronizadas nos 3 sistemas atualmente em uso (DVB, ATSC, e ISDB), tendo como exceção a tentativa do ATSC de gerenciar as *set-top boxes* via Java.

A tabela abaixo apresenta, resumidamente, os pontos discutidos aqui, com a indicação do atendimento ou não de determinado item de gerenciamento por cada solução apresentada:

| Item/solução | EUTELSAT | ATSC/JAVA | DAVIC | IBCoBN | CableHome | SBTVD |
|---|----------|-----------|-------|--------|-----------|-------|
| Uso do canal de retorno permanente | | X | X | X | X | X |
| Uso do canal de retorno temporário | | X | | | | X |
| Obrigatoriedade do uso de IP | | X | X | X | X | |
| Opção do uso de IP | | | | | | X |
| Obrigatoriedade do uso de ATM | | | X | | | |
| Monitoração do sinal transmitido | X | | X | | | |
| Gerência de serviços | | X | | | X | X |
| Gerenciamento de configuração | | X | | | X | |
| Gerenciamento de segurança | | | X | | X | X |
| Gerenciamento dos elementos da rede de transporte | X | | X | | | |
| Gerenciamento das set-top boxes | | X | X | X | X | X |
| Inteligência nas set-top boxes | | X | | X | | X |
| Possibilidade de controle de uma home network | | X | | | X | |

TABELA 4.2 – RESUMO COMPARATIVO DE SOLUÇÕES DE GERENCIAMENTO PARA TV DIGITAL

Nota-se que uma das preocupações recorrentes na grande maioria das arquiteturas é a presença de um canal de retorno, sendo que muitas delas consideram o mesmo como permanente, o que é realidade em uma rede de TV a cabo, porém não em uma rede de TV digital pública e terrestre. Existem até alguns trabalhos que não consideram o uso do canal de retorno, porém não se acredita ser este um caminho promissor. Nota-se também que um canal de retorno intermitente, (que é mais comum) altera em muito o funcionamento de qualquer arquitetura de gerenciamento.

A tecnologia IP é também considerada na maioria das arquiteturas, apesar de atualmente ainda ser incipiente a utilização de IP em redes de TV digital, pelo menos no canal de *broadcast*. Algumas arquiteturas ficam limitadas à tecnologia Asynchronous Transfer Mode (ATM).

Muitas das arquiteturas também tratam do problema do gerenciamento de serviços, porém com uma visão ainda de facilitar a entrega dos mesmos, e não de gerenciá-los efetivamente.

Do mesmo modo, o uso de arquiteturas que contemplem o gerenciamento da configuração e segurança das *set-top boxes* e serviços também ainda não é considerado nas pesquisas e soluções apresentadas, apesar de este ser um aspecto clássico muito importante na definição de uma arquitetura de gerenciamento.

Nota-se também que a maioria das arquiteturas não leva em conta o gerenciamento dos elementos da rede de transporte (exceto a arquitetura DAVIC e a EUTELSAT). Este aspecto, também extremamente importante, não faz parte do escopo desta pesquisa, porém se trata de assunto amplo e ainda não resolvido, afinal os elementos da rede de transporte de TV digital são novos e extremamente distintos dos utilizados em outras redes.

Outro ponto é que, apesar da falta de recursos nas *set-top boxes*, inteligência não pode faltar neste elemento, pois esta é a única saída viável para o problema da escalabilidade, que é extremamente crítico em redes gigantescas e complexas como as de televisão.

É importante citar que as soluções acima são utilizadas em ambientes bem peculiares, podendo ser adequadas, até certo nível, para a resolução dos problemas de gerenciamento nestes ambientes. Entretanto, nenhuma delas consegue reunir funcionalidades suficientes para resolver o complexo problema do gerenciamento de *set-top boxes* e serviços de redes de TV digital públicas e terrestres levando em conta todos os requisitos que as mesmas exigem.

Estes aspectos foram levados em consideração neste trabalho, e na construção do sistema aqui proposto tentou-se levar em conta o atendimento a estes requisitos, do modo mais completo possível.

5. ANÁLISE DOS REQUISITOS DE UM SISTEMA DE GERENCIAMENTO DE *SET-TOP BOXES* E SERVIÇOS EM REDES DE TV DIGITAL

Esta etapa apresenta quais os requisitos a serem atendidos para a construção de um sistema de gerenciamento de *set-top boxes* e serviços em redes de TV digital.

Foram levados em conta itens importantes considerados em pesquisas que abordam de modo tradicional o problema de gerenciamento e também outros estudos que apontam características que são bem peculiares para a resolução de problemas de gerenciamento de redes TV digital.

Uma proposta completa e inovadora dentro deste âmbito deveria considerar, no mínimo:

1. Gerenciamento de serviços, que é hoje uma exigência de qualquer arquitetura de gerenciamento.
2. Utilização de um protocolo de comunicação independente de IP e ATM no canal de *broadcast*. Muitas das redes de TV digital não suportarão estes protocolos na rede *broadcast*, pelo menos em um primeiro momento (o uso de IP *unicast* e IP *multicast* no canal de *broadcast* ainda é limitado).
3. Um gerenciamento projetado especificamente para o elemento de rede *set-top box*, que apresenta restrições claras de desempenho mas que deve possuir um mínimo de inteligência, sendo portanto um elemento que deve ser modelado de modo distinto aos outros elementos de rede hoje existentes.
4. Definição de uma arquitetura de gerenciamento da segurança da *set-top box* e também das aplicações (serviços).

5. Definição de uma arquitetura de gerenciamento da configuração da *set-top box* e também das aplicações (serviços).
6. Presença de um canal de retorno, no mínimo intermitente. A maioria das propostas considera canais de retorno permanentes, o que não ocorre para um sistema de TV digital terrestre. O sistema gerente também deveria estar ciente de que, devido à presença do canal de retorno não ser permanente, a gerência deve tolerar atrasos no envio de respostas da *set-top box*. Por outro lado, o terminal de acesso deve estar preparado para administrar o envio das mensagens de gerenciamento apenas quando este canal de retorno estiver disponível, armazenando as mesmas para envio posterior caso contrário. **Isto implica também na consideração de que algumas ações de gerenciamento podem não requerer confirmação.** Para os casos onde o canal de retorno não estivesse disponível, as operações de gerenciamento seriam realizadas porém não haveria nenhum tipo de confirmação. Para estes casos pode-se adotar, por exemplo, uma estratégia onde, dada uma ação de gerenciamento, os efeitos da aplicação da mesma serão avaliados primeiramente em *set-top boxes* que possuem canal de retorno, para que apenas depois (caso não existam efeitos negativos nas *set-top boxes* com canal de retorno), a mesma ação seja aplicada em *set-top boxes* sem canal de retorno.
7. Compatibilidade com uma arquitetura de gerenciamento mais flexível que o SNMP, como por exemplo WBEM, que facilita muito o processamento inteligente de informações na *set-top box*, através do uso de XML.
8. Real integração com um middleware aberto e de mercado, aproveitando recursos e APIs oferecidas pelo mesmo para o bom desempenho da arquitetura.
9. Integração com um sistema de autenticação e autorização de usuários (externo ao sistema de gerenciamento). Esta integração deveria permitir ao usuário selecionar o tipo de autenticação a ser usada por ele, sendo função do sistema de gerenciamento controlar estas informações e do sistema de autenticação e autorização armazenar e verificar senhas.
10. Utilização de plataformas realmente abertas e componentizadas, de modo que os módulos das mesmas pudessem ser utilizados em outras arquiteturas.

Com estas características definidas, foi possível projetar a especificação de requisitos de um sistema genérico de gerenciamento de TV digital. Nota-se que esta especificação é independente de arquitetura, podendo ser aproveitada na implementação em vários cenários e plataformas distintas.

5.1 Quality function deployment (QFD) / Especificação de requisitos funcionais

Para auxiliar no desenvolvimento da especificação de requisitos propriamente dita, foi adotada a técnica do QFD. O QFD é um método muito utilizado em projetos para garantir a qualidade do produto final baseando-se nos requisitos exigidos pelos usuários do sistema. Ou seja, ele é um método de mapeamento de requisitos direcionado completamente à satisfação total do usuário, que deve estar plenamente atendida e muitas vezes deve ser superada. O método QFD preocupa-se tanto com requisitos funcionais como não funcionais, porém neste trabalho o foco foi na identificação dos requisitos funcionais.

Uma vez definidos estes requisitos, os mesmos devem ser analisados e classificados, qualitativa e quantitativamente, de modo a estabelecer uma prioridade de implementação dos mesmos.

A partir daí, uma especificação de requisitos mais completa pode ser escrita, desde modo já se sabendo que o cliente final estará sendo atendido do modo como se espera. E isto desde o início, salvando tempo e recursos durante o decorrer do projeto. Obviamente sempre haverá desvios, mas os mesmos são minimizados com o uso do método QFD.

Já com a especificação de requisitos construída a partir do QFD mapeado, é possível definir quais os casos de uso que irão descrever a operacionalidade do sistema em desenvolvimento.

Primeiramente então foram definidos todos os requisitos de gerenciamento que seriam desejados pelo usuário final do sistema de TV digital. Os mesmos foram classificados pelo seu tipo em:

1. F – S – requisito funcional geral do sistema.
2. F – U – requisito funcional direcionado exclusivamente ao usuário final.
3. F –G – C - requisito funcional de gerência de configuração do sistema.
4. F –G – S - requisito funcional de gerência de segurança do sistema.
5. F –G – B - requisito funcional de gerência de contabilização do sistema.
6. F –G – F - requisito funcional de gerência de falhas do sistema.
7. F –G – P - requisito funcional de gerência de desempenho do sistema.

Após esta classificação, estes requisitos foram ordenados segundo a expectativa que o cliente teria sobre os mesmos. Esta classificação, segundo o método QFD, pode assumir 3 valores:

- Implícita – o cliente não enxerga explicitamente este requisito, porém se o mesmo não estiver presente gerará descontentamento.
- Normal – o cliente enxerga este requisito e considera o mesmo essencial para o produto.
- Surpreendente – algo que o cliente não enxerga à primeira vista porém que pode gerar um valor muito grande quando presente.

Foram determinadas também as prioridades de cada um destes requisitos, classificadas em: Alta, Média e Pequena. Estas prioridades foram definidas de acordo com o escopo desta pesquisa e com o tempo disponível para o desenvolvimento da mesma.

Na tabela abaixo (ver tabela 5.1) estão listados todos os requisitos mapeados e são indicados em cinza os que serão detalhados, pois estão de acordo com os objetivos desta pesquisa. Acredita-se que os outros requisitos formam um conjunto de funcionalidades a serem vislumbradas em trabalhos futuros, não fazendo parte do escopo deste trabalho.

TABELA 5.1 – QUALITY FUNCTION DEPLOYMENT INICIAL

| ID | Requisito | Tipo | Expectativa do Cliente | Prioridade | Esforço |
|----|--|--------|------------------------|------------|---------|
| 1 | Contabilizar o uso da rede | F-G-B- | Normal | Alta | Baixo |
| 2 | Contabilizar o uso de serviços | F-G-B | Normal | Alta | Médio |
| 3 | Sistema pré-pago | F-G-B | Surpreendente | Pequena | Alto |
| 4 | Configurar e armazenar a configuração das set-top boxes | F-G-C | Implícita | Alta | Médio |
| 5 | Configurar e armazenar a configuração de serviços | F-G-C | Surpreendente | Alta | Médio |
| 6 | Coletar e atuar sobre as falhas das set-top boxes | F-G-F | Implícita | Alta | Médio |
| 7 | Coletar e atuar sobre as falhas de serviços | F-G-F | Implícita | Alta | Alto |
| 8 | Controlar e atuar sobre as falhas de rede | F-G-F | Implícita | Média | Alto |
| 9 | Analisar e atuar sobre o desempenho dos serviços | F-G-P | Implícita | Alta | Alto |
| 10 | Analisar e atuar sobre o desempenho das set-top boxes | F-G-P | Implícita | Média | Médio |
| 11 | Analisar e atuar sobre o desempenho geral da rede | F-G-P | Implícita | Média | Médio |
| 12 | Gerenciar os aspectos de segurança no acesso às set-top boxes | F-G-S | Normal | Alta | Médio |
| 13 | Gerenciar os aspectos de segurança no acesso aos serviços | F-G-S | Normal | Alta | Médio |
| 14 | Garantir a privacidade e não-repudição dos dados trafegados na rede | F-G-S | Implícita | Pequena | Alto |
| 15 | Independência no método de acesso para autenticação no sistema | F-G-S | Surpreendente | Média | Médio |
| 16 | Atualizar automaticamente as novas funcionalidades de software na set-top box | F-S | Implícita | Pequena | Alto |
| 17 | Localizar automaticamente novos elementos que sejam conectados a uma rede residencial controlada pela rede residencial | F-S | Surpreendente | Pequena | Alto |
| 18 | Acesso possível a partir de vários pontos distintos | F-S | Surpreendente | Alta | Médio |
| 19 | Ter acesso a uma ajuda on-line interativa | F-S | Surpreendente | Pequena | Médio |
| 20 | Ter acesso a um Call-Center para ajuda on-line via TV | F-S | Surpreendente | Pequena | Alto |
| 21 | Acesso do usuário ao sistema com disponibilização de uma visão particular | F-U | Surpreendente | Média | Alto |
| 22 | Gerenciamento do compartilhamento de conteúdo com outros peers na rede de TV digital | F-U | Surpreendente | Pequena | Alto |

Com base nestes requisitos escolhidos para implementação (novamente, indicados em cinza na tabela acima e focados no gerenciamento de serviços e da *set-top box*), foi possível detalhar o sistema um pouco mais, também se adotando ainda o método QFD. Foram definidos portanto, para cada um destes requisitos, quais as condições nas quais os mesmos deveriam ser atendidos, bem como as restrições identificadas. Também foram definidos em quais módulos do sistema estes requisitos seriam implementados e quais as características técnicas que os mesmos deveriam respeitar. (ver tabela 5.2).

TABELA 5.2 – QUALITY FUNCTION DEPLOYMENT REFINADO

| ID | Requisito | Condições | Restrições | Características | Subsistemas |
|----|--|--|--|---------------------------------|---------------------------|
| 1 | Contabilizar o uso da rede | Não há | Dependente de interfaces adequadas com os sistemas de autenticação | Seguro, transparente | Contabilização, segurança |
| 2 | Contabilizar o uso de serviços | Não há | Dependente do modelo de serviços | Seguro, transparente | Contabilização, segurança |
| 4 | Configurar as set-top boxes | Armazenar sempre a última configuração disponível | Problemas de comunicação com o gerente podem gerar configurações erradas | Transparente, leve | Configuração |
| 5 | Configurar serviços | Não há | Dependente do modelo de serviços | Transparente, leve | Configuração |
| 6 | Coletar e atuar sobre as falhas das set-top boxes | Sistema atualizado de 10 em 10 minutos | Intermitência do canal de retorno | Rápida, transparente, confiável | Falhas, Configuração |
| 7 | Coletar e atuar sobre as falhas de serviços | Sistema atualizado de 10 em 10 minutos | Dependente do modelo de serviços | Rápida, transparente, confiável | Falhas, Configuração |
| 9 | Analisar e atuar sobre o desempenho dos serviços | Não há | Dependente do modelo de serviços | Confiável, rápido | Performance, Configuração |
| 10 | Analisar e atuar sobre o desempenho das set-top boxes | Não há | Intermitência do canal de retorno | Confiável, rápido | Performance, Configuração |
| 12 | Gerenciar os aspectos de segurança no acesso às set-top boxes | Não permitir que informações de autenticação trafeguem pela rede | Tipo de criptografia e outros mecanismos de segurança utilizados | Confiável | Segurança |
| 13 | Gerenciar os aspectos de segurança no acesso aos serviços | Não há | Dependente do modelo de serviços | Confiável | Segurança |
| 15 | Independência no método de acesso para autenticação no sistema | Mudança podese feita pelo usuário ou via call-cente | Não se aplica a todos os tipos de autenticação | Confiável | Segurança |
| 18 | Acesso possível a partir de vários | Não há | Restrito primeiramente à | Confiável | Segurança |

| | pontos distintos | | Internet | responsiv o | |
|----|---|----------------------------|---|----------------------------------|--------------------|
| 21 | Acesso do usuário ao sistema com disponibilização de uma visão particular | Tempo de resposta adequado | Criação de relatórios pelo usuário limitada | Confiável, responsiv o, claro | Contabiliz ação |

Nota-se neste QFD uma alocação clara dos requisitos em módulos bem particulares (Configuração, Segurança, etc). Esta alocação será útil para o desenho da estrutura geral do sistema sendo construído. Dependências relativas à intermitência do canal de retorno e ao modelo de serviços utilizado também são restrições importantes.

5.2 Requisitos não funcionais

Com o QFD definido, foi possível identificar também os requisitos não funcionais para o problema em questão. Estes estão listados abaixo e foram agrupados de maneira lógica para um melhor entendimento.

5.2.1 Requisitos de hardware da *set-top box*

1. Memória não volátil disponível: a aplicação de gerenciamento deve guardar as variáveis e objetos de gerenciamento do terminal de acesso em memória não volátil. Estima-se que 10Kbytes desta memória do terminal de acesso sejam suficientes e que nem todas variáveis e objetos precisem ser guardados nesta memória não volátil.
2. Memória volátil disponível: muitas *set-top boxes* possuem uma memória volátil total de 32Mbytes. Este valor é suficiente para o carregamento de várias aplicações, sendo que se estima que a aplicação de gerenciamento aqui proposta aloque menos de 500kbytes de memória volátil.
3. Processamento: estima-se que a arquitetura de gerenciamento aqui proposta necessite de um terminal de acesso de 233MHz ou menos, já que funções de gerenciamento não exigem tanto processamento.

5.2.2 Requisitos da rede de televisão digital

4. Presença de um canal de retorno que esteja disponível para uso da aplicação de gerenciamento por pelo menos parte do tempo, essencial para que haja comunicação no sentido agente – servidor de gerência. Caso o canal de retorno não esteja presente, o gerenciamento terá que ser efetuado de um modo mais restrito e “cauteloso”. O gerenciamento destas *set-top boxes* que não possuem canal de retorno seria realizado baseado nos resultados das ações de gerenciamento aplicadas nas *set-top boxes* que possuem canal de retorno, mesmo que intermitente.
5. Banda disponível para dados no canal de *broadcast*: estima-se que, inicialmente, 16kb/s sejam necessários para o tráfego de gerência no canal de *broadcast*.

5.2.3 Requisitos do servidor de gerência

6. Capacidade de armazenamento do servidor de gerência: são necessários uma alta capacidade de armazenamento e um gerenciador de banco de dados poderoso para que todas as informações de gerenciamento da arquitetura aqui proposta sejam corretamente armazenadas e possam ser retornadas com desempenho satisfatório.
7. Quantidade de *set-top boxes* gerenciáveis pelo servidor de gerência: relacionada ao número de *set-top boxes* que possam enviar respostas e informações a um único servidor de gerência. Este número deve ser dimensionado de acordo com o hardware do(s) servidor(es) de gerência disponível(is) bem como de acordo com a banda disponível para comunicação com estes servidores. Pode-se ter vários servidores espalhados por determinadas regiões, porém todas as informações de gerenciamento devem estar centralizadas em um único ponto da rede.

5.2.4 Requisitos do middleware da *set-top box*

8. Disponibilização de API de persistência (acesso à memória não volátil) por parte do middleware do terminal de acesso: necessário para que a aplicação de gerenciamento tenha acesso à memória não volátil do terminal de acesso.
9. Disponibilização de API de tratamento de eventos externos (controle remoto, etc) por parte do middleware do terminal de acesso: necessário para que a aplicação de gerenciamento possa receber os dados inseridos pelo usuário final, sempre que necessário.
10. Disponibilização de API de controle da interface entre usuário e televisão digital (apresentação): necessária pois algumas funções de gerenciamento podem exigir que o usuário seja alertado que as mesmas estejam ocorrendo, ou mesmo para que haja uma resposta adequada e em tempo real a algum evento realizado pelo usuário relacionado a funções de gerenciamento.
11. Disponibilização de API de acesso ao canal de retorno: essencial para que haja comunicação agente – servidor de gerência.
12. Disponibilização de uma API do middleware que identifique e separe os dados de gerenciamento que venham pelo canal de *broadcast*, de modo que os mesmos sejam entregues corretamente à aplicação de gerenciamento.

5.2.5 Requisitos de interatividade com o usuário final

13. Alfabetização do usuário final: o gerenciamento, como toda aplicação desenvolvida para a televisão digital, terá como alvo um público com variados níveis de instrução e experiência com dispositivos eletrônicos, inclusive analfabetos. Para isso, deverão ser consideradas algumas recomendações para a linguagem e modelos de interface gráfica empregados de forma a garantir a utilização da TV Digital do mesmo modo que o sistema de televisão já disponível no país.

14. Utilização do controle remoto: está sendo considerado como principal dispositivo de entrada o controle remoto, que possui recursos menores de interação quando comparado com um teclado e mouse. Dessa forma, a utilização do controle remoto pressupõe: (i) facilidade de manuseio com apenas uma das mãos, (ii) possuir elementos que correspondam com os da tela, (iii) botões com as mesmas cores e (iv) mesmos símbolos. Também deverão ser observados padrões já existentes no mercado, como as 4 (quatro) teclas coloridas presentes no padrão DVB. Deve haver sempre a preocupação com os usuários que possuem baixa alfabetização (digital ou não) mantendo, quando possível, um baixo nível de complexidade.
15. Feedback das ações de usuário: apresentar feedback para todas as ações do usuário indicando a realização de uma operação e mesmo o acionamento de uma tecla sem ação em um dado instante para não haver a impressão de falha.
16. Usabilidade: para a elaboração da interface gráfica da aplicação devemos considerar que os meios de interação da TV Digital são fundamentalmente distintos dos meios de interação de um computador pessoal e da Internet: (i) tamanho, formato, resolução de tela e elementos gráficos maiores; (ii) distância do usuário e da tela maiores, e; (iii) controle remoto como principal dispositivo de interação entre usuário e sistema. Dessa forma, as interfaces da aplicação deverão ser geradas segundo os seguintes princípios da usabilidade: (i) simples, para a total compreensão por parte das pessoas não acostumadas aos layouts da TV Digital, tornando extremamente fácil o acesso às informações procuradas pelos telespectadores, (ii) fácil compreensão, para evitar que as pessoas percam tempo procurando o que precisam. Esses objetivos deverão ser alcançados através de ícones (gráficos ou em forma de texto) auto-explicativos e de fácil memorização, a serem padronizados. Também deve ser levado em conta que os textos devem ser breves, as fontes utilizadas grandes e em negrito, a interface na tela deve conter instruções resumidas explicando aos usuários o que pode ser feito e não se deve utilizar ícones abstratos e relacionados com as interfaces do computador, considerando que a maioria dos telespectadores não tem um vocabulário de Internet.
17. Navegabilidade: outro critério considerado na elaboração das interfaces gráficas da aplicação é a navegabilidade que, além dos aspectos intrínsecos de usabilidade, engloba as características dos usuários, particularmente as estratégias que estes adotam no processo

de resolução de problemas e de como seu processo decisório é constituído. Dessa forma, a navegação deve considerar principalmente: (i) utilizar os números ou as setas no controle remoto para navegação; (ii) elementos selecionados devem ter suas cores e tamanhos alterados; (iii) o número de botões a ser apertado deve ser mínimo assim como as informações na tela.

5.2.6 Requisitos do projeto de software da aplicação

18. Tolerância à falhas: abordagens de construção de software robusto deverão ser usadas no desenvolvimento do sistema de gerenciamento, de forma a construir a aplicação com identificação de trechos críticos (trechos onde podem ocorrer erros durante a sua execução) e seus respectivos tratamentos.

5.2.7 Requisitos de consistência de dados

19. Consistência dos dados armazenados no servidor de gerência e nos agentes: este é um ponto muito importante, pois o canal de retorno não estará sempre disponível para a comunicação no sentido agente – gerente. Deste modo, devem existir os tratamentos adequados para que a inconsistência de dados (principalmente na gerência de configuração) sejam minimizadas e tratadas de forma correta quando ocorrerem.

5.2.8 Requisitos de integrabilidade

20. Integrabilidade com outras aplicações de TV digital presentes no terminal de acesso: como qualquer outra aplicação de TV digital, é necessário que a arquitetura de gerenciamento disponibilize interfaces para que suas funções sejam utilizadas por estes outros módulos. Do mesmo modo, a aplicação de gerenciamento deve utilizar as interfaces disponibilizadas por outras aplicações, sejam elas residentes ou não.

21. Real integração com um middleware aberto e de mercado, aproveitando recursos e APIs oferecidas pelo mesmo para o bom desempenho da arquitetura.
22. Integração com um sistema de autenticação e autorização de usuários (externo ao sistema de gerenciamento).
23. Integração com um sistema de controle e descrição de serviços, para que o gerenciamento de serviços possa ser efetivamente realizado.

Todos estes requisitos – funcionais e não funcionais - traçam um cenário bem particular para redes de TV Digital, que devem ser respeitados no desenvolvimento de qualquer solução de gerenciamento para este ambiente.

6. DETALHAMENTO OPERACIONAL DE UM SISTEMA DE GERENCIAMENTO DE *SET-TOP BOXES* E SERVIÇOS EM REDES DE TV DIGITAL

Para se detalhar a operacionalidade do sistema sendo construído, foi utilizado o método de casos de uso, recomendado para qualquer projeto que se apóie na tecnologia de orientação a objetos (caso desta pesquisa).

Como na especificação de requisitos apresentada no capítulo 5, o detalhamento operacional aqui listado é independente de plataforma, podendo ser utilizado em cenários com tecnologias diversas.

6.1 Diagrama de casos de uso

Na arquitetura aqui proposta, foram identificados os seguintes atores:

- Administrador do sistema.
- Usuário final.
- Sistema de Autenticação e Autorização de usuários (que não faz parte do sistema de gerenciamento em si, apenas interage com ele).
- Sistema de controle de serviços (que não faz parte do sistema de gerenciamento em si, apenas interage com ele).

Com estes atores definidos, os casos de uso macro, intimamente relacionados com as 5 áreas funcionais clássicas de gerenciamento (configuração, segurança, contabilização, desempenho

e falhas) puderam ser projetados (lembrando que neste trabalho as 5 áreas funcionais atuam sobre os elementos de rede, a rede e também sobre os serviços conforme definido anteriormente no escopo da pesquisa). Estes casos de uso macro são portanto:

- Realizar a gerência de desempenho.
- Realizar a gerência de falhas.
- Realizar a gerência da segurança.
- Realizar a gerência da contabilização.
- Realizar a gerência de configuração.

Com estes elementos foi possível então definir o seguinte diagrama de casos de uso (ver figura 6.1):

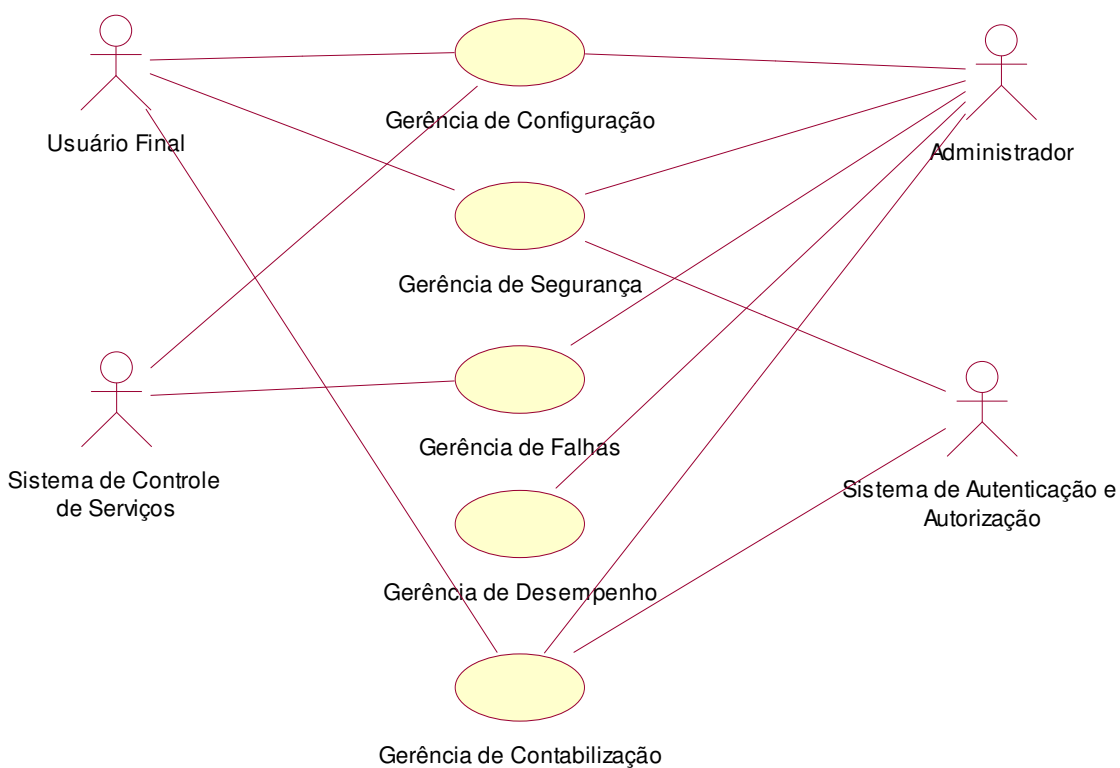


FIGURA 6.1 – DIAGRAMA DE CASOS DE USO

Neste diagrama, atenta-se para o fato do administrador do sistema participar ativamente dos 5 casos de uso macro, enquanto o usuário final não participa das ações relacionadas às falhas e desempenho, que devem ocorrer de modo transparentes para ele.

Um detalhamento de todos os casos de uso propostos para a operação deste sistema é realizado no Apêndice A – Detalhamento dos Casos de Uso.

7. ANÁLISE E ESCOLHA DA SOLUÇÃO TÉCNICA

Para que um sistema de gerenciamento de TV digital seja construído, é necessário não apenas a especificação de requisitos e a definição dos casos de uso, mas também uma análise das tecnologias disponíveis para a implementação e uma justificativa do uso das mesmas, já que atualmente há várias opções possíveis para uma mesma solução.

Neste capítulo são apresentadas estas soluções tecnológicas, bem como um cenário factível de utilização de um sistema de gerenciamento de *set-top boxes* e serviços em redes de TV digital para o qual são selecionadas, dentro das possibilidades apresentadas, as tecnologias mais indicadas a serem adotadas.

7.1 Tecnologias disponíveis

Como já citado, o problema de gerenciamento de TV digital é complexo, e só pode ser resolvido de modo satisfatório pela combinação do uso de várias tecnologias trabalhando em conjunto para atingir um mesmo objetivo.

A seguir são definidas quais as áreas tecnológicas de estudo para a solução deste problema e quais as opções disponíveis. Muitas delas já foram apresentadas em detalhe nesta pesquisa nos itens anteriores. Nota-se que a lista aqui é não exaustiva, podendo ser complementada por outros estudos semelhantes e por tecnologias inovadoras que passem a serem utilizadas ao longo do tempo.

7.1.1 Comunicação física do canal de *broadcast*

O canal de *broadcast* transporta os dados, imagens e sons multiplexados a partir da operadora (ou gerador de conteúdo) para a casa do usuário. Existem 4 opções para a implantação da camada física desta comunicação:

- Terrestre: é o sistema mais difundido para a transmissão de TV analógica aberta, porém também um dos menos utilizados para a transmissão de TV digital. Tem como vantagem a possibilidade da transmissão simultânea de sinais analógicos e digitais (separados por faixas de frequências distintas), recurso que é muito útil durante um período de migração da tecnologia analógica para a digital.
- Microondas: possui grande banda disponível porém um alto custo de implantação. Sujeito a interferências climáticas.
- Satélite: uma das primeiras redes a realizar transmissão digital, é hoje muito utilizada para sistemas de TV digital, mesmo porque seu alcance geográfico é praticamente ilimitado. Entretanto, possui uma alta latência de transmissão e um custo alto de banda. Também sujeito a interferências climáticas.
- Cabo: pouco difundida no Brasil (a não ser nas grandes capitais), possui um alto custo de implantação. Pode fornecer uma grande banda de transmissão, porém esta banda terá que ser compartilhada pelos usuários até mesmo na última milha. Tem como vantagem a facilidade da migração de sinais analógicos para digitais e a possibilidade da transmissão simultânea destes 2 sinais em um mesmo cabo (como a terrestre).

7.1.2 Mecanismo de multiplexação do canal de *broadcast*

O mecanismo de multiplexação do canal de *broadcast* é responsável por enviar por um mesmo canal físico programas e mídias distintos, que são demultiplexados na *set-top box* de acordo com a sintonização que o usuário final realiza. Atualmente existem 2 opções viáveis (ORZESSEK; SOMMER, 1998):

- MPEG-2 TS, utilizado em todas os padrões de TV digital hoje existentes, possui mecanismos padronizados e amplamente difundidos para multiplexação e demultiplexação de mídias. Disponibiliza mecanismos de sincronismo de mídias e é independente de protocolo de transporte.
- Real-Time Protocol (RTP) e RealTime Control Protocol (RTCP), utilizado em redes IP multiserviço, isto é, redes IP que podem transmitir aplicações de tempo real com QoS. Como o MPEG-2 TS, fornece mecanismos de sincronismo de mídias porém é dependente ao IP e UDP. É também uma tecnologia menos madura e menos utilizada que o MPEG-2 TS (no ambiente de TV digital).

7.1.3 Rede de transporte do canal de *broadcast*

Para a rede de transporte do canal de *broadcast* várias soluções estão disponíveis, sendo muitas delas na verdade um agrupamento de várias opções. Considera-se neste estudo as seguintes (ORZESSEK; SOMMER, 1998):

- Transporte MPEG-2 TS puro: é a solução utilizada em todos os sistemas de TV digital e possui padrões para sincronismo de mídias e QoS (embora os mecanismos de QoS não sejam tão amplamente utilizados). Pode ser encapsulado sobre IP ou ATM e não insere grande overhead na rede.
- Transporte IP: é uma opção que vem sendo bastante pesquisada e começa a ser utilizada comercialmente para transporte de serviços de tempo real com QoS (apesar de não oferecer QoS nativamente). Deve trabalhar em conjunto com o RTP/RTCP e UDP, embora esta solução introduza grande overhead na rede para transmissão de pacotes pequenos (como na solução de voz sobre IP), desperdiçando assim uma banda valiosa na rede de transporte. Pode ainda ser encapsulado sobre ATM ou mesmo MPEG-2 TS (embora o encapsulamento sobre MPEG-2 TS não seja tão comum).

- Transporte ATM: suporta QoS nativamente porém é uma rede que vem caindo em desuso. Por oferecer QoS na camada 2, tem ainda o problema de utilizar um circuito virtual dedicado para cada grupo de serviços (que tenham requisitos de QoS semelhantes), comprometendo assim a escalabilidade da rede.

7.1.4 Identificação da aplicação de gerenciamento nas tabelas MPEG-2

Os programas transmitidos em uma rede de TV digital são na verdade formados por várias mídias que devem atuar em conjunto e em sincronismo para que o usuário possa experimentar o resultado final desejado.

Cada canal de transmissão digital possui uma ampla gama de programas sendo transmitidos, sendo que estes programas são formados por várias mídias elementares.

Na nomenclatura MPEG, identifica-se em cada canal uma Program Association Table (PAT), que possui ponteiros para as tabelas de cada programa. Estas, por sua vez, chamadas de Program Mapping Table (PMT), possuem ponteiros para cada mídia elementar. Estes ponteiros, tanto da PAT como da PMT, são chamados de Programa Identifier (PID), e identificam univocamente uma mídia sendo transmitida em um canal MPEG-2 TS (ORZESSEK, SOMMER, 1998).

Caso esteja sendo utilizada a tecnologia de Multiprotocol Encapsulation, é possível que os dados de gerenciamento sejam encapsulados em TCP/UDP e IP.

Isto posto, têm-se as seguintes possibilidades para a identificação desta aplicação de gerenciamento:

- Alocação de um PID único para a aplicação de gerenciamento
- Alocação de uma porta UDP específica para identificação da aplicação de gerenciamento, no caso em que a tecnologia de Multiprotocol Encapsulation esteja sendo adotada.

7.1.5 Modo de transmissão de dados no canal de *broadcast*

O canal de *broadcast* transmite vários programas e mídias simultaneamente. Uma solução de gerenciamento deverá ocupar por um tempo determinado este canal para transmissão de comandos de gerenciamento. Quem define como ocorrerá esta alocação do canal de *broadcast* para a transmissão de dados é a tecnologia do modo de transmissão de dados no canal de *broadcast*, para a qual existem as seguintes opções de implantação (já apresentadas anteriormente nesta pesquisa), todas baseadas no uso da tecnologia Digital Storage Media – Command and Control (DSM-CC), que faz parte do padrão MPEG-2:

- Data Piping: dados transmitidos de forma não sincronizada com nenhuma outra mídia ou mecanismo de repetição.
- Data Streaming: envio de dados sincronizados com eventos de outras aplicações.
- Data / Object Carousel: envio do mesmo dado periodicamente, como um “carrossel” de dados.
- Multiprotocol Encapsulation: envio de dados sobre outro protocolo de transporte que não MPEG-2 TS.

7.1.6 Canal de retorno

Para o canal de retorno existe uma ampla gama de possibilidades para o transporte dos dados, sendo que para cada uma delas pode-se considerar o uso ou não de IP e também se o canal é permanente ou intermitente. Opções mais utilizadas atualmente:

- Canal de retorno via Sistema Telefônico Fixo Comutado (STFC): tem a desvantagem de requerer uma chamada explícita por parte do usuário para acesso à rede. Além disto tem

baixa banda de transmissão e costuma apresentar instabilidade na conexão. Entretanto, é uma solução barata e amplamente difundida (principalmente no Brasil).

- Canal de retorno *wireless*: pode disponibilizar um canal de retorno permanente com boa banda de transmissão. Entretanto, tem custo elevado atualmente (que pode vir a decair ao longo do tempo).
- Canal de retorno RF: também disponibiliza um canal permanente de retorno, porém com baixa banda de transmissão. Utiliza a tecnologia analógica.
- Canal de retorno pela própria rede a cabo (para redes com canal de *broadcast* também a cabo): apesar da banda ser compartilhada até a última milha em uma rede a cabo, é a solução mais interessante para usuários que já tenham a rede a cabo instalada em sua residência, pois fornece uma alta banda de transmissão. Tem como desvantagem o alto custo quando comparado com a solução STFC.
- Canal de retorno via satélite: pode ser interessante em regiões remotas, porém apresenta os problemas de alta latência e alto custo de banda.

7.1.7 Protocolo de comunicação de gerência

O protocolo de comunicação de gerência é responsável por enviar os comandos de gerência à *set-top box* e vice-versa. Aqui aparecem uma série de soluções possíveis, sendo muitas delas dependentes da rede de transporte do canal de *broadcast* bem como das tecnologias utilizadas no canal de retorno. Observa-se também que este protocolo é essencialmente dependente da arquitetura de gerenciamento utilizada e não necessariamente são utilizadas exatamente as mesmas soluções no canal de *broadcast* e retorno. Seguem as opções mais utilizadas atualmente:

- XML (arquitetura WBEM) diretamente sobre MPEG-2 TS ou sobre HTTP/IP. É uma solução que provê escalabilidade sendo aderente a padrões amplamente difundidos na

Internet. Entretanto, requer o desenvolvimento de uma nova base de dados de gerenciamento (CIM). Pode ser compatível com o SNMP através do uso de tradutores na rede. Sozinho, não resolve o problema do gerenciamento de serviços.

- SNMP sobre IP: solução simples e amplamente utilizada, que porém não garante a segurança nem a escalabilidade necessárias em uma rede de TV digital. Apresenta também o problema de possuir agentes (*set-top boxes*) não-inteligentes, pois segue um paradigma simples de leitura e escrita de variáveis de gerenciamento. Obviamente, também não resolve o problema do gerenciamento de serviços.
- JMX/XML (utilizados em conjunto, principalmente para solução do problema de gerenciamento de serviços/aplicações) diretamente sobre MPEG-2 TS ou sobre HTTP/IP: solução que roda sobre padrões da Internet e é integrável com XML. Trata do gerenciamento de serviços porém não existem ferramentas disponíveis para implementações mais complexas. Também pode comprometer a utilização de recursos na *set-top box*.
- OSGi / XML (utilizados em conjunto, principalmente para solução do problema de gerenciamento de serviços/aplicações) diretamente sobre MPEG-2 TS ou sobre HTTP/IP: trata do problema do gerenciamento de serviços aplicando padrões da Internet, sendo que o OSGi é integrável com o XML e disponibiliza uma série de ferramentas prontas para customização das funções de gerenciamento. Entretanto, o OSGi não é uma ferramenta ainda amplamente difundida.

7.1.8 Modo de armazenagem das informações de gerenciamento

Existem 2 modos principais de armazenagem das informações de gerenciamento, tanto do lado gerente como agente:

- MIB SNMP: amplamente utilizado, de fácil implementação porém que trata os objetos gerenciados como simples variáveis (passivas), não permitindo assim que os agentes gerenciados possuam inteligência.
- Modelo de informação CIM utilizando XML (arquitetura WBEM): requer que uma nova base de informação de gerenciamento seja desenvolvida (apesar de que, atualmente, o modelo CIM já contempla uma grande gama de informações de gerenciamento). Apresenta facilidades de armazenamento e processamento, além de ser orientada a objetos, o que permite que haja inteligência nos agentes.

Atualmente o uso de tradutores entre estes 2 mundos é amplo (KLIE; STRAUB, 2004).

7.1.9 Linguagem de programação utilizada no lado gerente

Para o desenvolvimento da aplicação de gerenciamento propriamente dita, 4 direções costumam ser tomadas:

- Uso de orientação a objetos com adoção da linguagem Java e seus respectivos ambientes de execução (fornecidos ou não pela Sun, como a Java Virtual Machine). O uso da linguagem Java e suas plataformas permitem uma ampla portabilidade porém isto implica em uma maior utilização de recursos do que outras iniciativas. Lembra-se que a maioria das aplicações para TV digital em sistemas atualmente em uso é escrita em Java.
- Uso de orientação a objetos com adoção da linguagem C++, hoje também utilizada em alguns sistemas de TV digital. A linguagem C++, por não estar baseada na utilização de uma máquina virtual, é menos portátil que a linguagem Java, porém mais eficiente.
- Uso de Visual Basic e da plataforma .NET. Permite maior integração com sistemas baseados em Windows e também um bom desempenho. Porém, é dependente de um único fabricante

- Uso da linguagem C, com programação procedural: pouco utilizada porém que ainda traz ganhos de desempenho. Tem como grande desvantagem não ser compatível com orientação a objetos.

7.1.10 Linguagem de programação utilizada no lado agente

Lembrando que a aplicação de gerenciamento do lado agente deve rodar sem tomar muitos recursos da *set-top box* porém com um mínimo de inteligência, têm-se as seguintes opções:

- Uso de orientação a objetos com adoção da linguagem Java e seus respectivos ambientes de execução para elementos com poucos recursos computacionais (Java 2 Platform, Micro Edition (J2ME) por exemplo). Permite portabilidade porém implica em uma alta utilização de recursos (mesmo com J2ME). Lembra-se que a maioria das aplicações para TV digital em sistemas atualmente em uso é escrita em Java, através por exemplo da API JavaTV.
- Uso de orientação a objetos com uso da linguagem C++. Permite melhor desempenho que as aplicações escritas em Java, porém não permite portabilidade, o que restringe sua utilização.
- Uso de métodos procedurais e utilização da linguagem C e até mesmo de linguagem Assembler, para preservação do desempenho através de uma utilização otimizada e inteligente de recursos. Por não ser compatível com orientação a objetos, praticamente inviabiliza seu uso em sistemas de TV digital, limitando-se a ser utilizada em rotinas de baixo nível de interação com hardware.

7.1.11 Plataforma de middleware

A plataforma de middleware deve ser escolhida de modo que atenda os requisitos traçados para a aplicação de gerenciamento de TV digital e também de modo que seja integrável às outras tecnologias que compõem todo o sistema. As opções disponíveis são aquelas utilizadas nos 3 sistemas padrão de TV digital:

- MHP, utilizada no sistema DVB: mais amplamente utilizada e conhecida. O middleware GEM (que tem como objetivo ser um padrão definitivo) é baseado fortemente no MHP .
- DASE, utilizada no sistema ATSC: adota de certa forma soluções semelhantes ao MHP, apesar de pender muito estas soluções ao mercado americano, o que pode inviabilizar o uso do mesmo no Brasil.
- ARIB, utilizada no sistema ISDB: pouco utilizado a não ser em seu país de origem (Japão).

7.1.12 Arquitetura de controle de serviços

Em uma arquitetura onde os serviços/aplicações oferecidos aos usuários finais devam ser gerenciados de forma completa e correta, é necessária a escolha de uma arquitetura adequada para o controle e entrega destes serviços. Esta arquitetura não fará parte efetivamente do sistema de gerenciamento, mas será responsável por interagir com este, ao descrever corretamente os serviços, seus parâmetros e características para que os mesmos possam ser entregues devidamente ao usuário final.

Muitas vezes esta arquitetura de serviços não é considerada nos sistemas de entrega de conteúdo multimídia, mesmo porque esta tecnologia ainda está em desenvolvimento. Entretanto, a mesma se mostra interessante considerando-se que ela pode melhorar a experiência do usuário final, ao informá-lo corretamente, através de seus descritores, dos conteúdos e aplicações disponíveis para seleção e entrega. Do mesmo modo, estes descritores facilitam a comunicação entre aplicações distintas que lidam com conteúdo multimídia, pois descrevem este conteúdo de uma forma padronizada.

As 2 opções mais comuns para estas tecnologias são:

- Solução proprietária desenvolvida pelo próprio produtor do conteúdo, sendo que esta traz problemas óbvios de interoperabilidade.
- MPEG-21, que define um modelo para a descrição de conteúdo, controle e entrega de serviços multimídia, apesar deste ainda estar em fase de definição.

7.1.13 Localização física da aplicação a ser executada no lado agente

Em um ambiente de TV digital, existem basicamente 3 possibilidades para a localização física de uma aplicação que pode ser executada pela *set-top box* (O'DRISCOLL, 1999):

- Residente na *set-top box* e parte integrante do middleware. Consome recursos da *set-top box* porém provê bom desempenho. Neste item encaixam-se normalmente aplicações que disponibilizam interfaces de manipulação do hardware e/ou Sistema operacional.
- Residente na *set-top box*, porém não integrante ao middleware. Possui bom desempenho e não requer carregamento dinâmico. Apesar disto, deve ser cuidadosamente dimensionada para não prejudicar o desempenho da *set-top box* como um todo.
- Remota, que deve ser baixada para a *set-top box* para ser executada, automaticamente ou mesmo manualmente segundo comando do usuário. Tem as desvantagens de requerer um download, o que pode não ser viável para uma aplicação de gerenciamento que deve estar rodando desde o momento em que a *set-top box* é iniciada.

7.2 Cenário de utilização

Um cenário de utilização factível para a implantação da arquitetura de gerenciamento de TV digital a ser proposta neste trabalho é um semelhante ao DVB-T (sistema DVB com canal de *broadcast* terrestre). Este sistema possui as seguintes características:

1. Canal de broadcast com banda disponível de normalmente 19 Mbits/s.
2. Multiplexação de vários programas e mídias em um único canal de broadcast.
3. Possibilidade de interatividade com o usuário final, podendo ser esta uma interatividade local, por um canal de retorno temporário ou por um canal de retorno permanente.
4. Possibilidade de envio de aplicações via canal de *broadcast* ou via canal de retorno, que serão executadas caso sejam sintonizadas pelos usuários e tenham as permissões necessárias para esta execução.
5. Identificação de aplicações enviadas no canal de *broadcast*.
6. Utilização de um middleware padrão, amplamente adotado e pesquisado, que possui também um módulo com suporte a IP.
7. Possibilidade de execução de aplicações Java, HTML ou XML nas *set-top boxes*.

Com estas funcionalidades disponíveis, o sistema DVB-T é hoje um dos sistemas de TV digital mais populares em uso e se beneficiaria de uma arquitetura de gerenciamento que contemplasse todos os requisitos apresentados no item 5 deste trabalho.

Outro ponto importante é que, por ser bastante difundido, várias sistemas em uso ou em desenvolvimento (por exemplo, sistema brasileiro), possuem (ou possuirão) características bem semelhantes, alargando assim a abrangência deste trabalho.

7.3 Tecnologias utilizadas

Apresentadas todas as tecnologias disponíveis e um cenário de utilização factível no qual esta pesquisa estará se baseando para o desenho da arquitetura de gerenciamento, é possível justificar a seleção das tecnologias mais adequadas para este trabalho (baseando-se nas tecnologias listadas anteriormente no item 7.1), a seguir:

- Comunicação física do canal de *broadcast*: terrestre, pois é o mais comum dos modos de comunicação física do canal de *broadcast* (principalmente no Brasil) e possui banda disponível suficiente para a entrega de dados da aplicação de gerenciamento.
- Mecanismo de multiplexação do canal de *broadcast*: MPEG-2 TS, pois é o utilizado nos sistemas de TV digital hoje existentes e atende completamente o requisito de multiplexação de vários sinais no canal de *broadcast*.
- Rede de transporte do canal de *broadcast*: MPEG-2 TS puro, pois deste modo não é necessário atrelar uma comunicação IP mandatária no canal de *broadcast*, que poderia restringir o escopo da arquitetura aqui proposta.
- Identificação do programa nas tabelas MPEG-2: caso a tecnologia de Multiprotocol Encapsulation esteja disponível, deve ser selecionada uma determinada porta UDP para identificação da aplicação de gerenciamento. Esta opção é mais simples e mais viável de ser implementada do que a alocação de um PID único de gerenciamento. Caso a mesma não esteja disponível, deve-se adotar o uso do PID.
- Modo de transmissão de dados no canal de *broadcast*: data carousel DSM-CC, pois o gerente deve enviar comandos às *set-top boxes* periodicamente. Estes comandos estariam portanto encapsulados em mensagens de data carousel DSM-CC. É necessário também que a aplicação de gerenciamento possua uma banda disponível para envio destes comandos no canal de *broadcast*. Não é recomendável o uso do object carousel DSM-CC pois arquivos não serão transmitidos pela aplicação de gerenciamento, pelo menos em um primeiro momento. Apesar de se adotar este método de carrossel de dados, na maioria das vezes a aplicação de gerenciamento não estaria enviando nenhum comando às *set-top boxes*, então poderia ser enviado apenas um comando como NULO. Quando algum

comando tivesse que ser realmente enviado, a aplicação de gerenciamento, no lado agente, capturaria este comando e o interpretaria da forma adequada.

- Canal de retorno: será utilizado o canal de retorno STFC, com uso de IP, por ser uma solução bastante difundida e de baixo custo.
- Protocolo de comunicação de gerência: OSGi/XML sobre MPEG-2 TS no canal de *broadcast* e sobre HTTP/IP o canal de retorno. Deste modo, o gerenciamento de serviços está coberto pela tecnologia OSGi, que também é voltada a rodar em hardwares de baixo desempenho, que é o caso das *set-top boxes*. Além disto, a tecnologia OSGi fornece uma série de serviços de gerenciamento e controle de aplicações, baseados em Java, já prontos para uso (e sem custo). O uso do XML (arquitetura WBEM) é interessante pois não estará sendo considerado obrigatoriamente o uso de IP no canal de *broadcast*. Entretanto, caso a tecnologia de Multiprotocol Encapsulation esteja disponível, fica mais simples o envio no canal de *broadcast* das informações de gerenciamento OSGi/XML encapsuladas em TCP e IP. Além destes pontos, acredita-se que a arquitetura WBEM fornece soluções mais inteligentes para as questões de gerenciamento do que o SNMP, diminuindo assim o tráfego na rede e também o consumo de recursos na *set-top box*. Também é importante notar que a arquitetura OSGi é compatível com XML.
- Modo de armazenagem das informações de gerenciamento: modelo de informação CIM utilizando XML (arquitetura WBEM), pois a mesma, de forma semelhante ao protocolo de comunicação de gerência, pode realizar um processamento inteligente de informações no lado da *set-top box*. Também é interessante manter o uso do XML nas MIBs já que o mesmo será adotado como protocolo de comunicação de gerência. Além disto, o modelo de informações de gerenciamento CIM já é bastante amplo, sendo necessárias apenas algumas extensões a este para cobrir o ambiente de TV digital. Lembra-se novamente que a arquitetura WBEM é compatível com SNMP, através do uso de tradutores na rede, fazendo com que deste modo a arquitetura proposta possa ser integrada também com SNMP.

- Linguagem de programação utilizada no lado gerente: Java, pelas ferramentas e plataformas amplamente disponíveis e pela compatibilidade com orientação e objetos, método que será utilizado no desenvolvimento deste sistema.
- Linguagem de programação utilizada no lado agente: como no lado gerente, uso da linguagem Java porém através da plataforma J2ME. O Java também deve ser selecionado pois a arquitetura OSGi roda sobre uma máquina virtual Java.
- Plataforma de middleware: MHP, por ser aberta, amplamente documentada e utilizada na maioria das pesquisas sobre TV digital.
- Arquitetura de controle de serviços: MPEG-21, por ser um padrão aberto e uma das primeiras iniciativas a definir soluções para os problemas de descrição de conteúdo, controle e entrega de serviços.
- Localização física da aplicação a ser executada no lado agente: residente, porém não integrada ao middleware, pois isto levaria a um enrijecimento das soluções de gerenciamento que, na verdade, apenas se utilizam das APIs disponibilizada pelo middleware (ex: módulos de persistência, interface gráfica com usuário, canal de interatividade, entre outros) para serem executadas. Acredita-se também que a aplicação não deve ser baixada dinamicamente pois, mesmo sem canal de retorno, é necessário que a aplicação de gerenciamento esteja rodando a todo o momento. O que pode (e deve) ocorrer dinamicamente é a atualização da aplicação de gerenciamento, com novos módulos que possam ser integrados à aplicação quando necessário. Como a aplicação de gerenciamento terá acesso de leitura e escrita a informações de sistema, a mesma deve estar protegida de modo que outras aplicações não usem estas informações. Caso estas questões de segurança não possam ser resolvidas de modo satisfatório, então a aplicação de gerenciamento deve ser executada “ao lado” do middleware. Deste modo, várias funções de sistema acessadas por esta aplicação em específico não serão acessadas por mais nenhuma aplicação, protegendo deste modo a *set-top box* e o próprio usuário final.

Concluindo, todas as tecnologias justificadas neste capítulo foram selecionadas por serem as mais adequadas ao cenário definido no item anterior, que se acredita ser um dos cenários mais comuns em sistemas de TV digital.

A arquitetura proposta levará em conta estas tecnologias, porém é importante citar que a implementação realizada adiante não necessariamente irá adotar a solução completa, pelas próprias restrições que normalmente são impostas a um ambiente de teste.

8. PROPOSTA PARA UMA ARQUITETURA ABERTA DE UM SISTEMA DE GERENCIAMENTO DE *SET-TOP BOXES* E SERVIÇOS EM REDES DE TV DIGITAL

Apresentados todos os requisitos e também o detalhamento operacional de um sistema genérico de gerenciamento de TV digital, juntamente com a definição de um cenário plausível de aplicação e a justificativa da utilização das tecnologias mais indicadas para este cenário, é possível definir neste momento a arquitetura da solução de gerenciamento desta pesquisa.

Deste modo, são apresentadas a seguir a arquitetura de rede utilizada bem como a modelagem UML projetada para a aplicação de software. A arquitetura proposta é também posicionada em relação ao middleware do terminal de acesso.

Por fim, é definida uma estratégia para integração dos módulos de software da aplicação e quais os testes de verificação/validação que devem ser realizados durante o desenvolvimento da mesma.

8.1 Arquitetura de rede

A arquitetura de rede necessária para a implantação da arquitetura de gerenciamento proposta contempla os seguintes itens principais (ver figura 8.1.1):

1. Multiplexação de outras mídias com os dados da aplicação de gerenciamento no canal de *broadcast*, através do MPEG-2 TS, sem uso de IP, e utilizando-se do método de Data Carousel do padrão DSM-CC para envio periódico de dados XML de gerenciamento.

2. Canal de retorno via STFC utilizando IP através da Internet para atingir o servidor de gerência (poderia ser utilizada uma conexão segura do Servidor de Acesso Remoto à rede do servidor de gerência, caso necessário). Neste sentido da comunicação os dados de gerenciamento XML são transportados por HTTP.
3. Acesso de clientes da gerência ao sistema de gerência via HTTP.

Além destes elementos principais, são indicadas também na topologia as principais plataformas de software necessárias (ver figura 8.1):

- Java Virtual Machine (JVM) rodando tanto na *set-top box* como no gerente.
- Plataforma OSGi (inclusive OSGi XML Parser) rodando na *set-top box*.
- Middleware MHP executando na *set-top box*.
- HTTP servidor e cliente executando na gerência e na *set-top box* (neste caso para a comunicação via canal de retorno).
- Banco de dados com os descritores de conteúdo MPEG-21, localizado no provedor de conteúdo (externo ao sistema de gerenciamento de TV digital).

Também é indicada na topologia a base de dados de gerenciamento, particionada nos seguintes bancos (ver figura 8.1):

- Configuração
- Gerência
- Segurança
- Desempenho

- Falhas

É também importante citar que sempre devem ser adotados módulos de software mais leves para o lado agente (por exemplo, existem várias opções disponíveis para a execução de um HTTP Server que consuma poucos recursos de processamento e memória).

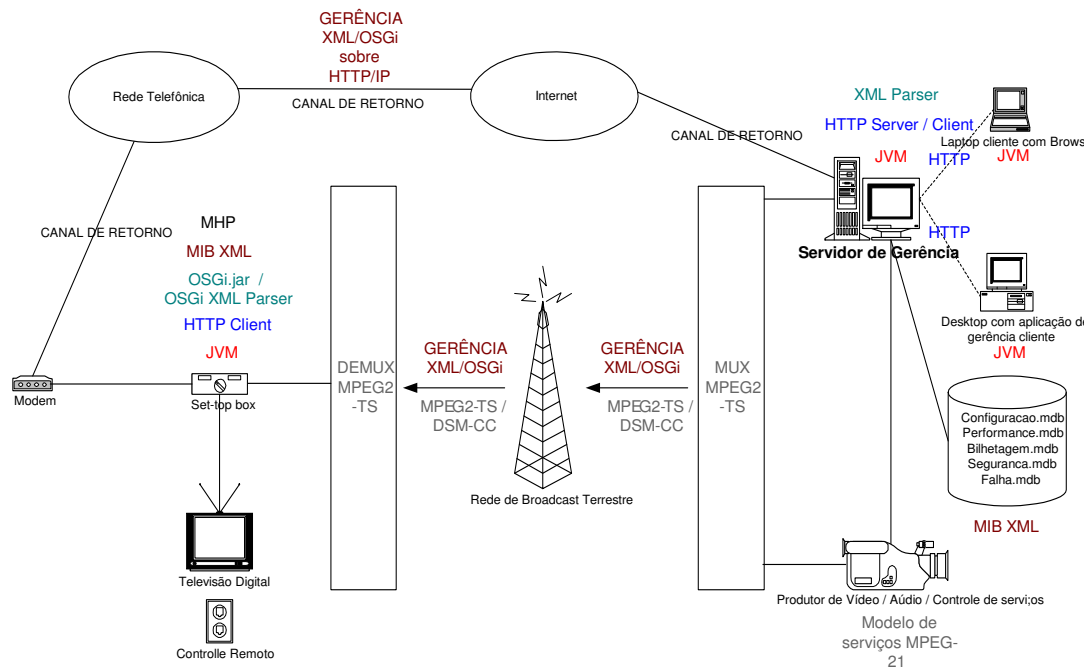


FIGURA 8.1 – ARQUITETURA DE REDE

8.1.1 Escalabilidade

Dentro da arquitetura de rede selecionada, bem como da arquitetura de software apresentada adiante, é importante citar como esta solução de gerenciamento trata o problema da escalabilidade.

Como é sabido, redes de TV digital são extremamente complexas, ramificadas e possuem um enorme número de elementos conectados à mesma.

Entretanto, nota-se que no canal de *broadcast* não há problema de escalabilidade, pois os comandos de gerenciamento serão enviados para todos os elementos da rede (ou para apenas um grupo) da mesma maneira que programas são enviados. A propósito, o tráfego de gerenciamento demanda bem menos banda do que outros serviços, o que facilita o envio destes dados no *broadcast*. Note que até este ponto o sistema de gerenciamento envia dados comuns a todos os elementos da rede. Caso o canal de *broadcast* venha a ser utilizado também para o envio de mensagens *unicast*, por exemplo pelo uso de IP neste canal, começa-se a ter problemas de falta de banda para o envio das mesmas. Conclui-se, portanto, que caso o canal *broadcast* seja usado para transporte de tráfego *unicast*, mesmo que de gerenciamento (que não consome tanta banda), é necessário utilizar este recurso com cautela para que dados não sejam descartados na rede por falta de banda. Apesar desta pesquisa não tratar diretamente sobre o uso de *unicast* no *broadcast*, tampouco sobre o uso de IP neste canal, é recomendável que esta modalidade de envio de dados *unicast* no canal de *broadcast* contemple apenas o envio de mensagens bem específicas, sob pena de congestionamento da rede e perda de pacotes.

Já para o canal de retorno, faz-se necessário a divisão de tarefas de recebimento e processamento das mensagens de gerenciamento entre vários servidores. Como esta comunicação será em IP, e o endereço do servidor de gerenciamento deverá ser descoberto através da adoção do formato Universal Resource Locator (URL), é relativamente conhecida e simples a tarefa de direcionar o cliente ao servidor de gerenciamento mais próximo, a partir do resultado da resolução de nome realizada.

Sugere-se o uso de vários servidores de gerência espalhados por uma mesma cidade e que cada um teria a responsabilidade sobre uma região, por exemplo.

Estes servidores poderiam concentrar os dados de gerenciamento coletados em uma base de dados única e comum a todos (espelhada e redundante). Como cada *set-top box* tem uma identificação única na rede, seria possível identificar univocamente cada uma delas mesmo nesta base comum. Esta base de dados também disponibilizaria o acesso às partes interessadas e autorizadas através de uma interface padrão de acesso a estes dados.

8.1.2 Mobilidade

Um outro aspecto a ser considerado dentro da arquitetura de rede apresentada é a questão da mobilidade. Por mobilidade entende-se que um usuário tenha a mesma experiência multimídia não importa onde nem em qual *set-top box* seu acesso seja feito.

Com relação ao gerenciamento propriamente dito, a questão da mobilidade não deve ser vista como essencial quando considerado apenas o gerenciamento da *set-top box* (afinal, o que é gerenciado neste escopo é o próprio elemento de rede *set-top box*).

Por outro lado, com relação ao gerenciamento de serviços, a questão da mobilidade passa a ser importante, pois é interessante que a gerência continue atuando efetivamente sobre os serviços que estão disponíveis para um determinado usuário, não importa em qual *set-top box* o mesmo esteja conectado. Para tanto, é imprescindível que o usuário esteja *logado* na rede de TV digital (ou seja, esta solução é referente aos casos onde o canal de retorno está disponível). Entretanto, o sistema aqui presente não contempla esta solução pois considera que o gerenciamento de serviços está amarrado a uma identificação física da *set-top box*, que é única na rede, porém que está relacionada com o hardware.

8.1.3 Modelo de negócios

Dada a arquitetura de rede apresentada, pode-se listar algumas opções de modelos de negócios que podem ser adotadas para uma possível comercialização da aplicação aqui definida. Primeiramente definem-se as partes interessadas:

1. Fabricantes de *set-top boxes*
2. Operadoras de redes de televisão aberta
3. Operadoras de redes de televisão por assinatura

Cada uma destas partes interessadas irá atuar do lado gerente (sempre como o Ator Administrador, conforme definido no item 6 deste trabalho)

Por exemplo, no caso do fabricante de *set-top box* estar atuando como Administrador, o mesmo gerenciaria as *set-top boxes* de **sua** fabricação. Para isto, deveria atuar apenas no canal de retorno ou acordar o envio de comandos através do canal de *broadcast* de alguma operadora de televisão (de preferência aberta neste caso). Neste cenário, o fabricante de *set-top box* é o único que realiza gerência, sendo que deste modo suas *set-top boxes* possuirão uma vantagem competitiva com relação a de outros fabricantes, podendo até mesmo oferecer serviços gerenciáveis a seus clientes.

Este é apenas um exemplo da utilização comercial desta aplicação, sendo que outros cenários podem ser projetados de acordo com os interesses das partes envolvidas.

8.2 Modelagem

Na modelagem UML aqui realizada, são definidos os subsistemas de software da arquitetura em questão bem como os diagramas de classe de cada um destes pacotes. Porém, primeiramente é importante posicionar a arquitetura proposta com relação ao middleware presente na *set-top box*.

Com os diagramas a seguir e a arquitetura de rede apresentada no item anterior acredita-se que o sistema aqui proposto possa ser construído de forma completa e correta.

8.2.1 Posicionamento da arquitetura em relação ao middleware

Para que haja um melhor entendimento do posicionamento da arquitetura de gerenciamento em relação ao middleware da *set-top box*, lembra-se que este último é composto de alguns módulos essenciais e que os mesmos atuam como interfaces para que outras aplicações (residentes ou não) possam utilizar o hardware e os recursos da *set-top box*.

Listam-se portanto os módulos principais do MHP, adotado como middleware nesta pesquisa.

1. Módulos de processamento gráfico

- Elementos gráficos: auxiliam na construção das telas de interface com o usuário.
- Tratamento de eventos: recebem e tratam os eventos (estímulos) externos, como por exemplo o acionamento do controle remoto.
- Apresentação de mídias: apresentam as mídias para o usuário.

2. Módulos de Processamento

- Processamento de mídias: reproduzem os diferentes formatos de mídias adequadamente.
- Processamento de dados: processam os dados advindos do canal de *broadcast* ou do canal de retorno.

3. Módulos de comunicação

- Tratamento dos dados do canal de *broadcast*: entende e separa os dados do canal de *broadcast* enviando-os posteriormente ao módulo de processamento de dados.
- Canal de retorno: fornece interfaces para a comunicação via canal de retorno.
- Comunicação entre aplicações: realiza a comunicação entre as diversas aplicações da *set-top box*.

4. Módulos de captura de conteúdo

- Sintonizador: sintoniza o canal e o programa a ser assistido, de acordo com a seleção do usuário.

- Tratamento do elementary stream: fornece informações sobre os elementary streams sendo entregues via canal de *broadcast*, inclusive sobre as tabelas sendo transportadas.
- Seleção do elementary stream: seleciona o elementary stream.

5. Módulos adicionais:

- Persistência: fornece interfaces para que dados sejam gravados e lidos da memória não-volátil do terminal de acesso.
- Acesso condicional: responsável pelo acesso a conteúdos criptografados.
- Gerenciamento de aplicações: controla o ciclo de vida das aplicações que executam em determinado instante no terminal de acesso.

A partir do oferecimento destes módulos, várias aplicações independentes do middleware (porém que vão utilizar as interfaces do mesmo) poderão rodar no terminal de acesso, sejam elas residentes ou não. Podemos citar entre estas uma aplicação de Correio Eletrônico e outra de um Navegador HTML.

A aplicação que rodará no terminal de acesso da arquitetura de gerenciamento proposta nesta pesquisa também se encaixa neste grupo de aplicações residentes que utilizam as interfaces disponibilizadas pelo middleware para realizar suas funções adequadamente.

Para ilustrar melhor estes conceitos e a proposta aqui realizada, apresentamos o diagrama a seguir que posiciona a arquitetura aqui proposta em relação ao middleware do terminal de acesso (ver figura 8.2) lembrando que comumente os middlewares rodam sobre uma máquina virtual Java:

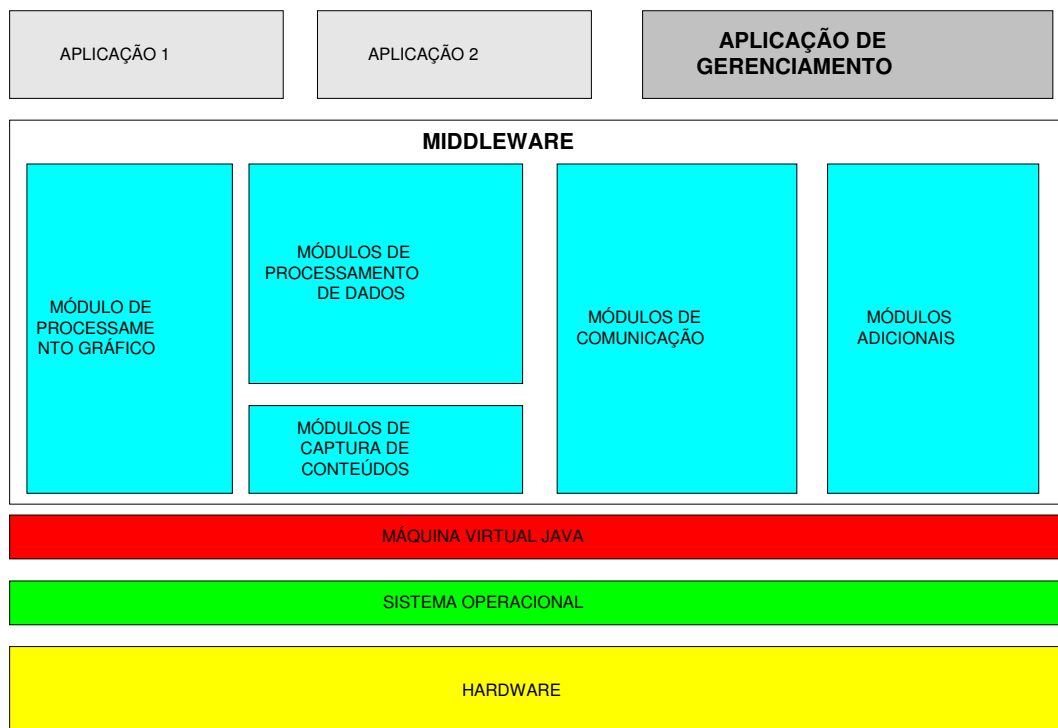


FIGURA 8.2 – APLICAÇÃO DE GERENCIAMENTO RELATIVA AO MIDDLEWARE

A figura 8.2 não tenta indicar quais módulos do middleware são utilizados por quais aplicações. Muitas vezes uma aplicação utilizará as interfaces disponibilizadas por todos os módulos do middleware. Lembra-se também que, conforme comentado no item 7, é possível que, por questões de segurança, seja recomendável executar esta aplicação de gerenciamento “ao lado” do middleware no lado agente.

Adiante, definimos quais as interfaces usadas pela aplicação de gerenciamento aqui proposta e também quais os módulos componentes desta aplicação.

8.2.2 Arquitetura interna

São apresentados a seguir os componentes propostos para esta arquitetura, primeiramente do lado servidor e após do lado agente.

8.2.2.1 Arquitetura interna - gerente

No diagrama de componentes a seguir são apresentados os 6 módulos principais desta arquitetura (ver figura 8.3, componentes em amarelo claro), suas associações e a nomenclatura sugerida para os pacotes associados aos mesmos (gtd foi a sigla adotada para gerenciador de TV digital):

- Módulo Gerenciador da Aplicação - Gerente, que possui o controle de execução sobre os outros módulos e faz a interface com os elementos gráficos do sistema – `gtd.gerente*`
- Módulo (de gerência) de desempenho - `gtd.gerente.desempenho*`
- Módulo (de gerência) de contabilização - `gtd.gerente.contabilizacao*`
- Módulo (de gerência) de segurança - `gtd.gerente.seguranca*`
- Módulo (de gerência) de falhas - `gtd.gerente.falhas*`
- Módulo (de gerência) de configuração - `gtd.gerente.configuracao*`

Além destes, seguindo a idéia do desacoplamento de componentes proposta no paradigma de programação orientada a objetos, são apresentados mais 6 componentes responsáveis pela interface com o mundo externo (ver figura 8.3, componentes em amarelo escuro), além de um sétimo módulo responsável pela interpretação e construção de mensagens de gerenciamento (acessado por todos os 5 módulos principais). Estes componentes, **ainda pertencentes à aplicação de gerenciamento**, isolam as interfaces do sistema com o mundo externo, de modo que as alterações em algum destes elementos externos não afetem o sistema de gerenciamento como um todo, facilitando assim a manutenção e atualização da arquitetura aqui proposta. Para estes componentes serão definidas então suas funções principais desejadas, deixando os detalhes de implementação dos mesmos em aberto. Estes componentes são:

- Módulo Gerenciador Eventos Usuário – trata os eventos externos recebidos - `gtd.gerente.eventos*`. Funções:
 - Recebe e trata os eventos externos recebidos, como por exemplo o acionamento do teclado ou mouse pelo usuário ou a entrada de dados.
 - Envia tais dados ao controlador de eventos do estado corrente.
- Módulo Gerenciador Interface Usuário – apresenta as telas ao usuário - `gtd.gerente.interfaces*`. Funções:
 - Apresenta as telas ao usuário final, de acordo com o estado corrente.
- Módulo Gerenciador OSGi – interface com o módulo OSGi, utilizado para gerência de serviços - `gtd.gerente.osgi*`. Funções:
 - Faz o acesso OSGi aos agentes
- Módulo Gerenciador Banco de Dados (BD) – interface com o BD de gerenciamento - `gtd.gerente.bd*`. Funções:
 - Inserção de dados no banco de dados de gerenciamento.
 - Exclusão de dados do banco de dados de gerenciamento.
 - Atualização de dados no banco de dados de gerenciamento.
 - Retorno de dados do banco de dados de gerenciamento.
- Módulo Gerenciador MPEG-21 – interface com sistema descritor de conteúdo de serviços (neste caso MPEG-21) - `gtd.gerente.mpeg21*`. Funções:
 - Inserção de dados no depósito de descritores de conteúdo MPEG-21.

- Exclusão de dados do depósito de descritores de conteúdo MPEG-21.
 - Atualização de dados no depósito de descritores de conteúdo MPEG-21.
 - Retorno de dados do depósito de descritores de conteúdo MPEG-21.
- Módulo Gerenciador Autenticação Usuários – interface com o sistema de autenticação - `gtd.gerente.aaa*`. Funções:
 - Coleta e recebe dados de autenticação e autorização de usuários, interpretando mensagens do servidor de autenticação e mapeando esta informação para um formato inteligível para a aplicação de gerenciamento.
 - Módulo Mensagens de Gerenciamento – `gtd.gerente.mensagens*`. Funções:
 - Interpreta mensagens XML de gerenciamento.
 - Constrói mensagens XML de gerenciamento.

Ainda estão indicados no diagrama os componentes externos ao sistema que são tratados pelos módulos citados anteriormente. Estes componentes estão indicados em laranja (ver figura 8.3) e são:

- Eventos Usuário.
- Interface Usuário.
- OSGi.jar – que auxilia no gerenciamento de serviços.
- BD Gerenciamento – banco de dados de informação de gerenciamento, localizado no lado gerente.

- BD Descritor de Serviços MPEG-21 – armazena os descritores de conteúdo MPEG-21.
- Autenticação de Usuários – sistema responsável pela autenticação, autorização e bilhetagem dos usuários.

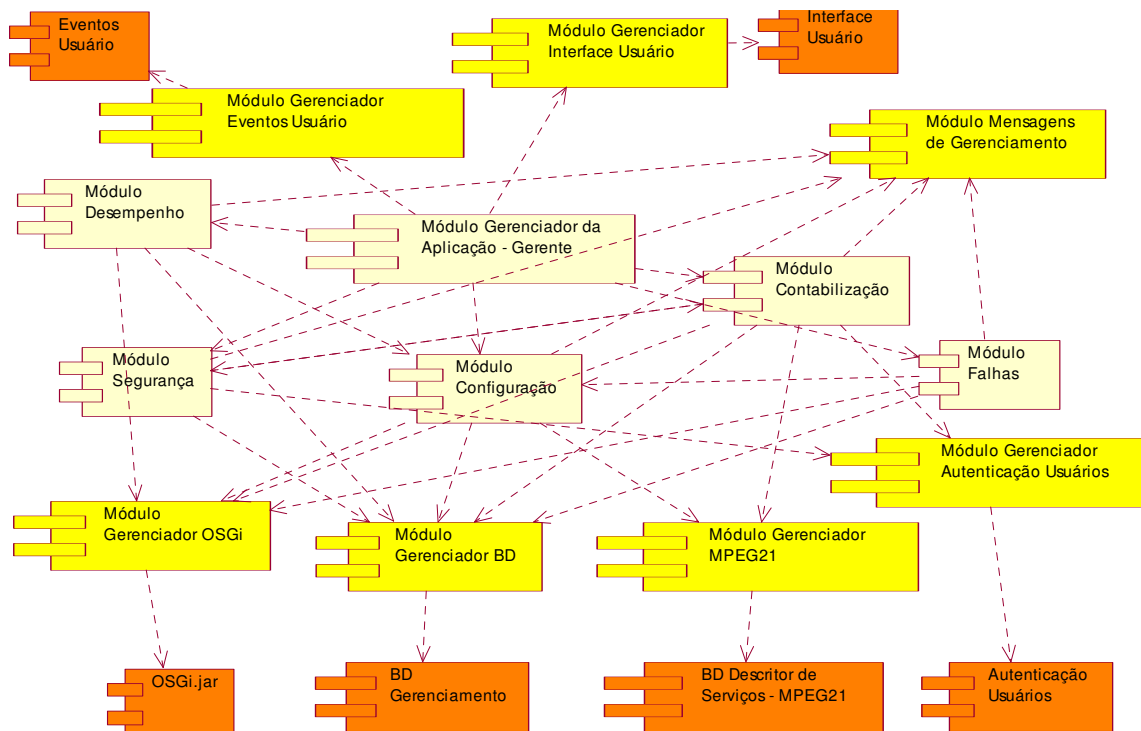


FIGURA 8.3 – DIAGRAMA DE COMPONENTES - GERENTE

8.2.2.2 Arquitetura interna - agente

Já do lado agente preocupações com a interação com o middleware, sistema operacional e até mesmo recursos de hardware da *set-top box* devem ser levados em conta.

Do mesmo modo como na arquitetura interna do gerente, identificamos os seguintes componentes principais (ver figura 8.4, componentes em amarelo claro):

- Módulo Gerenciador da Aplicação - Agente, que possui o controle de execução sobre os outros módulos e faz a interface com os elementos gráficos do sistema - `gtd.agente*`
- Módulo (de gerência) de desempenho - `gtd.agente.desempenho*`
- Módulo (de gerência) de segurança - `gtd.agente.seguranca*`
- Módulo (de gerência) de falhas - `gtd.agente.falhas*`
- Módulo (de gerência) de configuração - `gtd.agente.configuracao*`
- Módulo (de gerência) de contabilização - `gtd.agente.contabilizacao*`

Um ponto importante no lado agente são as poucas associações de dependência entre estes módulos principais, já que muitas decisões são tomadas apenas do lado gerente, para que o processamento seja poupado no agente.

Também do mesmo modo como no gerente, são identificados os componentes responsáveis pelas interfaces com o mundo externo (seguindo também o princípio de desacoplamento) além do módulo de tratamento de mensagens de gerenciamento. Para estes componentes serão definidas aqui suas funções principais desejadas, deixando os detalhes de implementação dos mesmos em aberto. Estes componentes são (ver figura 8.4, componentes em amarelo escuro):

- Módulo Gerenciador Eventos Usuário – trata os eventos externos recebidos - `gtd.agente.eventos*`. Funções:
 - Recebe e trata os eventos externos recebidos, como por exemplo o acionamento do controle remoto da televisão digital pelo usuário ou a entrada de dados.
 - Envia tais dados ao controlador de eventos do estado corrente.

- Módulo Gerenciador Interface Usuário – apresenta as telas ao usuário - `gtd.agente.interfaces*`. Funções:
 - Apresenta as telas ao usuário final, de acordo com o estado corrente e com a interface adotada.
- Módulo Gerenciador OSGi – interface com o módulo OSGi, utilizado para gerência de serviços - `gtd.agente.osgi*`. Funções:
 - Faz a coleta de dados de gerenciamento de serviços, através da camada OSGi presente em todos os serviços que foram carregados na *set-top box*. Como se está adotando a arquitetura WBEM, este módulo OSGi comporta-se como um **provedor** WBEM para a coleta de dados de gerenciamento de serviços (sendo que o modelo CIM já contempla informações de gerenciamento de serviços).
 - Gerencia os serviços (também através desta camada OSGi), realizando funções de inicialização, pausa, atualização e destruição dos mesmos.
- Módulo Gerenciador Persistência – interface com a memória persistente da *set-top box* (leitura e escrita) - `gtd.agente.persistencia*`. Cabe aqui um comentário relativo à forma de armazenamento implementada pela memória persistente. O sistema será desenvolvido baseando-se em um armazenamento orientado a objetos, método bastante razoável para utilização em *set-top boxes*. Deste modo, cada objeto (que será instanciado a partir de uma classe específica), será armazenado na memória com os mesmos atributos deste respectivo objeto. É necessário portanto o uso de uma chave única de identificação para cada classe e também o uso de uma outra chave para a distinção de cada objeto instância dentro de uma classe específica. Estas chaves podem ser escolhidas no momento da implementação. Abaixo as funções relativas ao módulo de interface com a memória persistente:
 - Inserção de dados na memória persistente da *set-top box*.

- Exclusão de dados da memória persistente da *set-top box*.
- Atualização de dados na memória persistente da *set-top box*.
- Retorno de dados da memória persistente da *set-top box*.

- Módulo Gerenciador Comunicação – interage com o canal de retorno e também recebe os dados do canal de dados do *broadcast* - `gtd.agente.comunicacao*`. Funções:
 - Envio de dados ao canal de retorno.
 - Recebimento de mensagens de gerenciamento pelo canal de *broadcast*.
 - Recebimento de mensagens de gerenciamento pelo canal de retorno.

- Módulo Gerenciador de Configuração do Sistema Operacional – interage com algumas variáveis de sistema que poderão ser controladas pela aplicação de gerenciamento - `gtd.agente.os*`. Funções:
 - Coleta de parâmetros de “baixo nível” da *set-top box* através de comunicação com o sistema operacional da mesma.
 - Alteração de parâmetros de “baixo nível” da *set-top box* através de comunicação com o sistema operacional da mesma.

- Módulo Mensagens de Gerenciamento – `gtd.agente.mensagens*`. Funções:
 - Interpreta mensagens de gerenciamento.
 - Constrói mensagens de gerenciamento.

Já os módulos externos utilizados pela aplicação no lado agente são (ver figura 8.4, componentes em laranja):

- Eventos Usuário.
- Interface Usuário.
- OSGi.jar – que auxilia no gerenciamento de serviços.
- Persistência – memória persistente do terminal de acesso.
- Canal de *broadcast* (dados) – recebe, trata e envia à aplicação correta os dados oriundos do canal de *broadcast*.
- Canal de retorno – envia dados no canal de retorno.
- Sistema Operacional – controle de variáveis de ambiente e sistema.

Neste ponto é interessante citar a preocupação com a segurança desta aplicação de gerenciamento no lado agente. Deve haver um mecanismo que controle o acesso, tanto de leitura como de escrita, a variáveis de “baixo nível”, ou seja, variáveis de sistema. Caso este mecanismo seja disponibilizado pelo middleware, deve haver um controle para que apenas a aplicação de gerenciamento acesse estas APIs de alteração de variáveis de sistema. Caso este mecanismo não exista, uma sugestão é que a aplicação aqui proposta esteja sendo executada “ao lado” do middleware, ou seja, a mesma terá acesso a APIs de alterações de variáveis de sistema, porém outras aplicações residentes comuns não terão acesso às mesmas. Deste modo a segurança da *set-top box* e do próprio usuário estão garantidas.

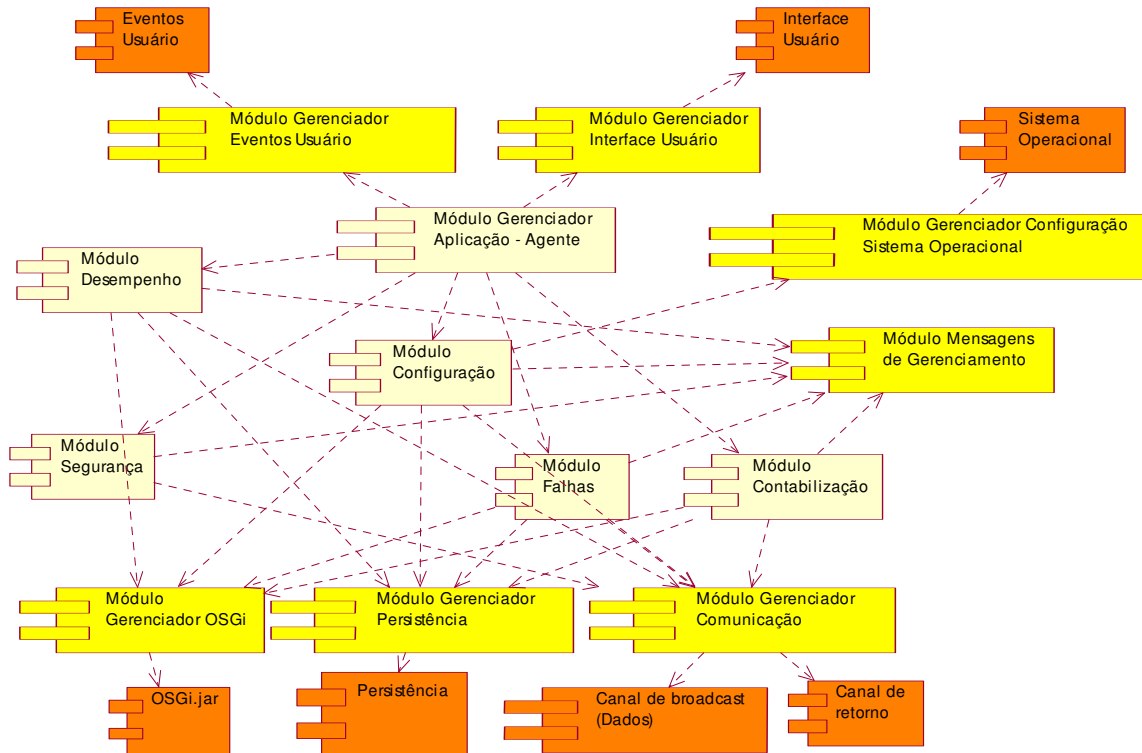


FIGURA 8.4 – DIAGRAMA DE COMPONENTES - AGENTE

Nos itens a seguir é apresentado o detalhamento de cada um dos componentes aqui citados.

8.2.3 Modelagem dos estados do sistema

Como proposto em (GAMMA et al., 1997), um sistema pode ser modelado através de seus estados, que são ativados a partir de eventos internos ou externos e que podem por sua vez também disparar algumas ações.

A modelagem por estados mostra-se bastante útil, tanto para o fácil entendimento, manutenção e atualização do sistema como também para a melhoria do desempenho do mesmo (READ; TICHHOSKY; 2003), principalmente quando este é executado sobre plataformas com poucos recursos computacionais, que é o caso das *set-top boxes* de TV digital.

Estaremos adotando portanto esta metodologia no desenvolvimento desta arquitetura. Para tanto, são apresentados primeiramente o diagrama de estados do agente (mais simples) e após do gerente. Com estes definidos, serão apresentados os diagramas de classes correspondentes derivados destes.

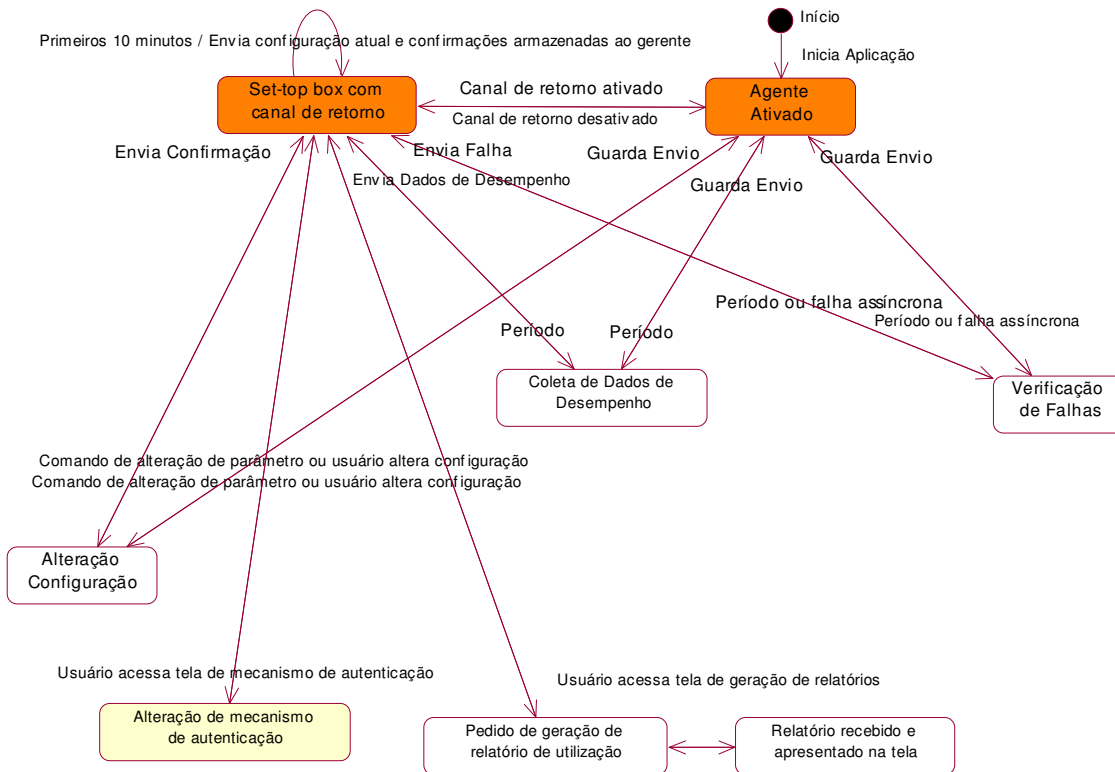


FIGURA 8.5 – DIAGRAMA DE ESTADOS - AGENTE

Nestes diagramas (figuras 8.5 e 8.6), foi adotada a seguinte convenção:

- Estados em laranja – estados principais do sistema.

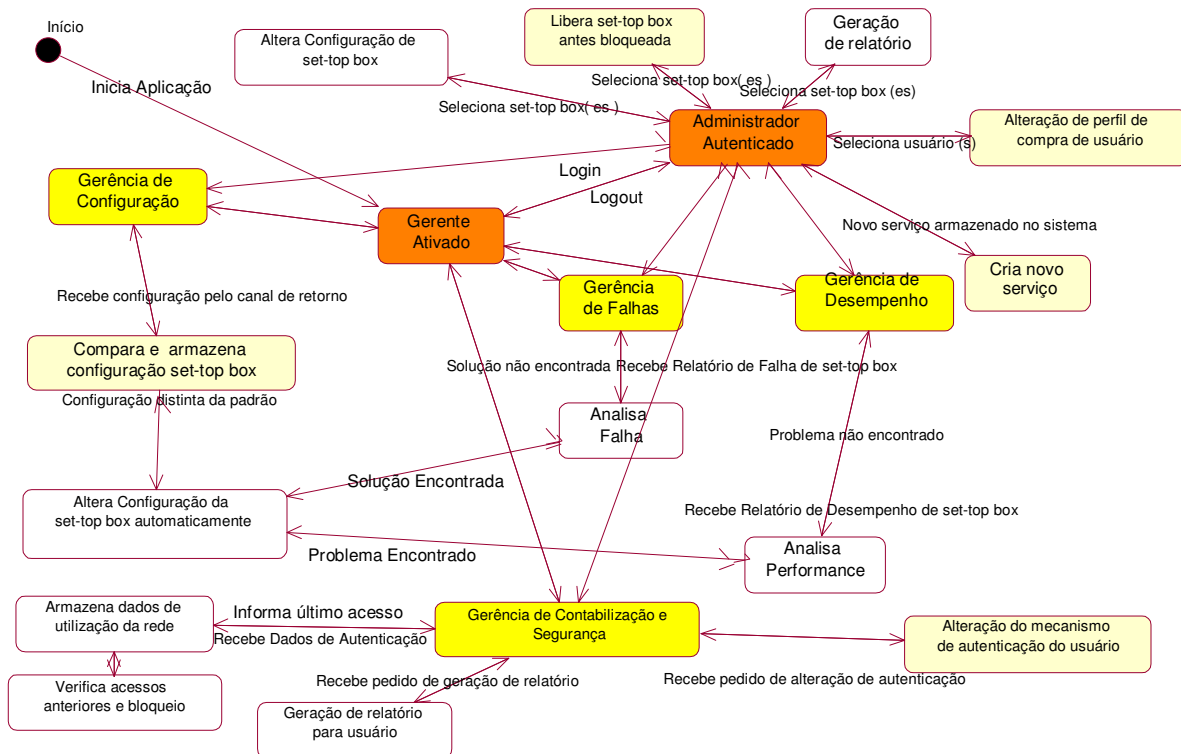


FIGURA 8.6 – DIAGRAMA DE ESTADOS – GERENTE

- Estados em amarelo escuro - estados simbólicos dos vários módulos de gerência do sistema. Efetivamente, o sistema não estaciona em nenhum destes estados, os mesmos estão indicados apenas para facilitar o entendimento.
- Estados em amarelo claro – estados de atuação exclusiva sobre as *set-top boxes* ou sobre serviços (exclusivamente).

- Estados em branco – estados que podem estar relacionados à atuações tanto sobre *set-top boxes* como sobre serviços, dependendo do contexto. Estes estados serão desmembrados em 2 classes, uma representando a atuação sobre *set-top boxes* e outra representando a atuação sobre serviços.

Com relação ao gerente, nota-se que o mesmo possui ações que são realizadas automaticamente e outras que são realizadas apenas com o administrador autenticado no sistema (tarefas de operação do sistema).

Já com relação ao agente, o mesmo possui 2 estados principais bem distintos, um com canal de retorno disponível e outro sem. A principal diferença entre eles, que representa um dos maiores desafios na construção de sistemas de gerência de TV digital, é que o primeiro envia ao gerente imediatamente o resultado das alterações realizadas em algum estado, e o segundo armazena tais informações localmente para depois enviá-las ao gerente.

Abaixo são definidos os diagramas de classe correspondentes derivados destes diagramas.

8.2.3 Pacotes – Gerente

Do lado gerente, podemos identificar uma classe principal, chamada de *GerenteAplicaçãoGTD*, que controla o tempo de vida da aplicação de gerenciamento, que é efetivamente uma instância da classe *GerenteGTD*. Como método público, será disponibilizado o construtor da classe *GerenteAplicaçãoGTD*, que poderá ser acionado por outras aplicações (ou apenas pelo próprio administrador) para a execução da aplicação de gerenciamento do lado gerente.

Já a classe *GerenteGTD*, por sua vez, é formada por 3 classes principais: uma relacionada à lógica de negócio envolvida em um estado (*GerenteGTDState*), outra relacionada ao controle de eventos (*GerenteGTDController*), e uma última que controla a apresentação de dados ao

usuário (GerenteGTDView). Este nada mais é que o modelo clássico Model-View-Controller (MVC) apresentado em (GAMMA et. al, 1997).

Para cada estado, estas 3 classes devem ser instanciadas e os respectivos objetos do estado anterior devem ser destruídos. **O controle da lógica de mudanças de estado pode ser distribuído entre as próprias classes concretas dos estados ou ser centralizado na classe GerenteGTD, sendo esta uma decisão de implementação. Esta lógica deve ser implementada de acordo com os diagramas de estado apresentados anteriormente neste trabalho (item 8.2.3).**

Nota-se também que as classes que representam os estados GerenteAtivadoState e AdministradorAutenticadoState estão presentes em todos os diagramas para que sua importância seja destacada, já que estes são os 2 estados principais da aplicação do lado gerente.

Nos diagramas adiante foram adotadas as seguintes legendas:

1. Elementos em amarelo claro: classes principais do sistema gerenciador – pacote `gtd.gerente*`.
2. Elementos em amarelo escuro: classes que representam os estados propriamente ditos – pacote `gtd.gerente.area_funcional_especifica*`.
3. Elementos em verde: classes que representam os elementos de gerência WBEM - pacote `gtd.gerente*`. É importante notar que estes elementos, que armazenam as classes e instâncias WBEM, são na verdade os descritores do estado de uma *set-top box* ou de um serviço em um determinado instante. Adotando-se o CIM Core Schema padrão (DMTF, 2005), já é possível descrever com acuidade o estado destes 2 elementos. Do lado gerente, apenas as classes CIM são de interesse, pois as instâncias estarão presentes nos agentes.
4. Elementos em laranja: interfaces.

Nos diagramas a seguir não estão indicados todas as especializações das classes GTDController e GTDView para facilidade de entendimento dos diagramas. Aliás, todos

os diagramas fazem parte do mesmo sistema, e poderiam ser apresentados em um único diagrama, o que não é realizado por motivos óbvios de espaço.

8.2.3.1 Pacote Configuração – Gerente – `gtd.gerente.configuracao*`

O pacote `gtd.gerente.configuracao*` é um dos mais importantes deste trabalho, pois indica qual a modelagem adotada para a estrutura tanto da *set-top box* como dos serviços, modelagem esta necessária para se realizar a gerência de configuração.

Este pacote possui 6 estados principais, sendo que 4 deles são especializações de 2 estados abstratos. Com relação à *set-top box* e aos serviços, eles são modelados pelas Classes CIM adequadas. Nota-se também que as interfaces OSGi e MPEG-21 são acessadas apenas pelos estados que envolvem a gerência de serviços.

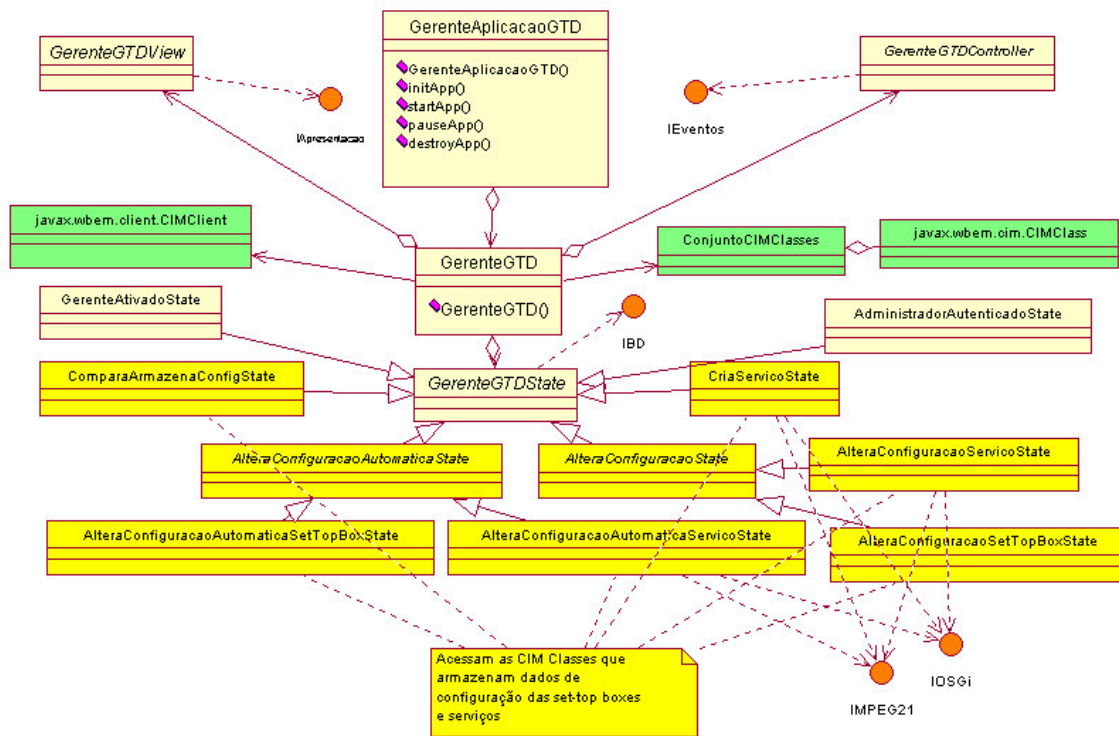


FIGURA 8.7 – DIAGRAMA DE CLASSES PACOTE CONFIGURAÇÃO - GERENTE

8.2.3.2 - Pacote Contabilização – Gerente - `gtd.gerente.contabilizacao*`

O pacote `gtd.gerente.contabilizacao*` possui 8 estados, sendo que este pacote também possui classes abstratas como o pacote configuração. Os elementos de dados de contabilização são representados pelas classes CIM adequadas.

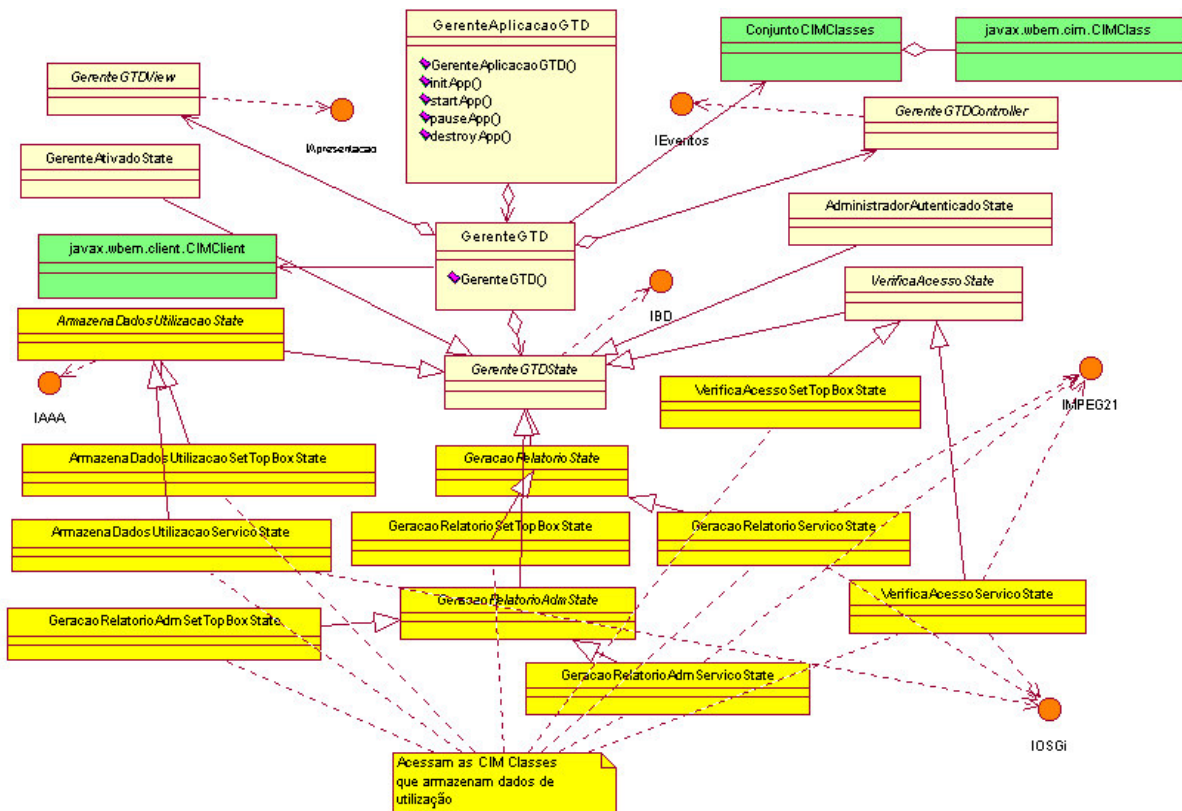


FIGURA 8.8 – DIAGRAMA DE CLASSES PACOTE CONTABILIZAÇÃO - GERENTE

8.2.3.3 - Pacote Falhas – Gerente - gtd.gerente.falhas*

O pacote `gtd.gerente.falhas*` modela os estados de análise de falhas, tanto do terminal de acesso como de serviços (através do OSGi), lembrando que o lado gerente da arquitetura é responsável por analisar e corrigir as falhas apresentadas pelos elementos da rede de TV digital.

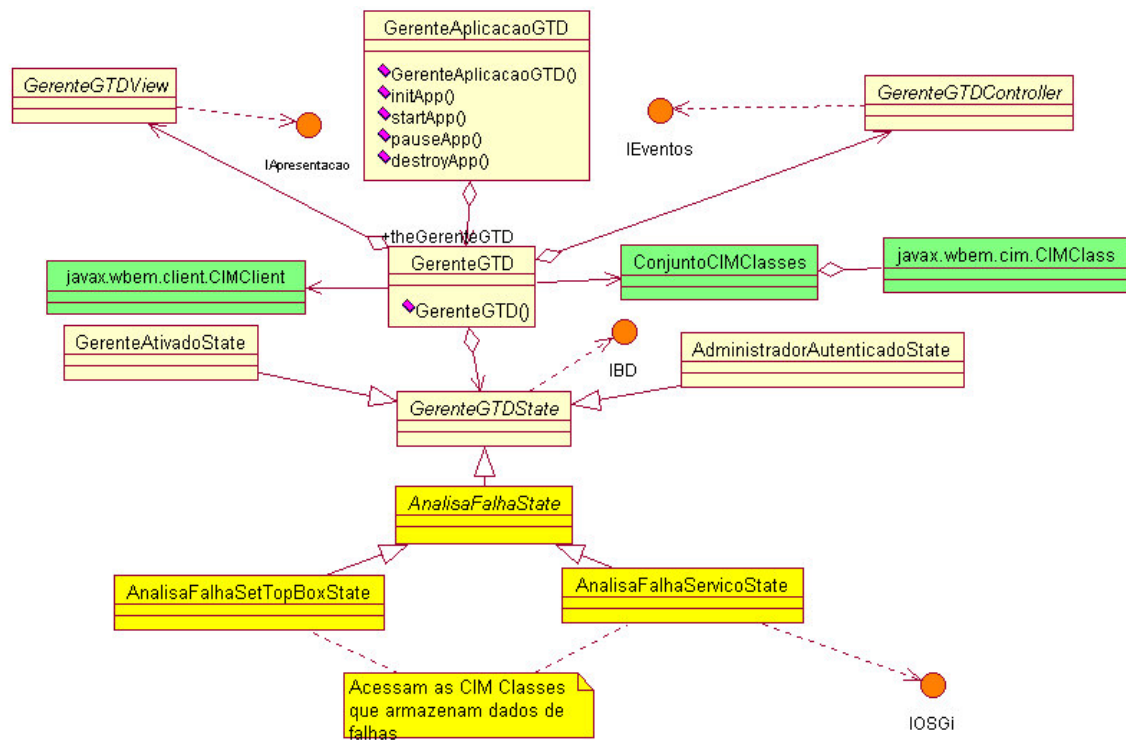


FIGURA 8.9 – DIAGRAMA DE CLASSES PACOTE FALHA - GERENTE

8.2.3.4 Pacote Desempenho – Gerente - `gtd.gerente.desempenho*`

O pacote `gtd.gerente.desempenho*` segue a mesma filosofia do pacote de falhas, sendo modelado por 2 estados principais e sendo responsável por analisar situações onde haja falta de desempenho de uma *set-top box* ou serviço.

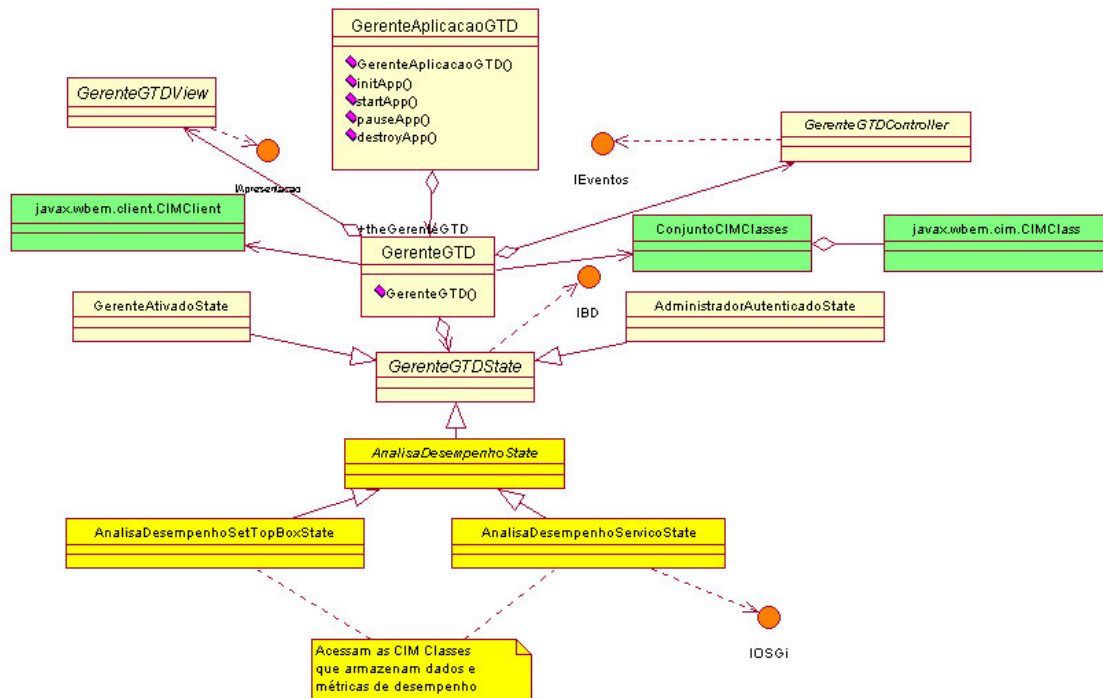


FIGURA 8.10 – DIAGRAMA DE CLASSES PACOTE DESEMPENHO - GERENTE

8.2.3.5 Pacote segurança – Gerente - `gtd.gerente.seguranca*`

O pacote `gtd.gerente.seguranca*`, através das classes CIM adequadas, modela o método de autenticação utilizado pelo usuário, bem como seu perfil de compra, suas informações para cadastro e também o bloqueio ou não da *set-top box* do usuário. São definidos 3 estados principais neste pacote.

inicialização da *set-top box*. A aplicação do lado agente também poderá ser parada e/ou reiciniada, porém deve haver um mecanismo de controle de acesso adequado para tanto (não contemplado nesta pesquisa).

Já a classe *AgenteGTD*, do mesmo modo que do lado gerente, é formada por 3 classes principais: uma relacionada à lógica de negócio envolvida em um estado (*AgenteGTDState*), outra relacionada ao controle de eventos (*AgenteGTDController*), e uma última que controla a apresentação de dados ao usuário (*AgenteGTDView*).

Para cada estado, estas 3 classes devem ser instanciadas (as classes do estado anterior devem ser destruídas). **O controle da lógica de mudanças de estado pode ser realizado nos próprios estados concretos ou centralizado na classe *AgenteGTD*, sendo esta uma decisão de implementação. Esta lógica deve ser implementada de acordo com os diagramas de estado apresentados anteriormente neste trabalho (item 8.2.3).**

Nota-se também que as classes que representam os estados *AgenteAtivadoState* e *CanalRetornoState* estão presentes em todos os diagramas para que sua importância seja destacada, pois são estes os 2 estados principais do sistema do lado agente.

Nos diagramas adiante foram adotadas as seguintes legendas:

1. Elementos em amarelo claro: classes principais do sistema gerenciador – *gtd.agente**
2. Elementos em amarelo escuro: classes que representam os estados propriamente ditos - *gtd.agente.area_funcional_especifica**
3. Elementos em verde: classes que representam os elementos de gerência WBEM - pacote *gtd.gerente**. Ao contrário do lado gerente, no agente tanto as classes como as **instâncias** CIM devem ser modeladas.
4. Elementos em laranja: interfaces.

8.2.4.2 Pacote contabilização – agente - gtd.agente.contabilizacao*

O pacote `gtd.agente.contabilizacao*` possui 4 estados, que são classes concretas de 2 estados abstratos. Como elementos descritores de estados, adotam-se as classes e instâncias CIM adequadas.

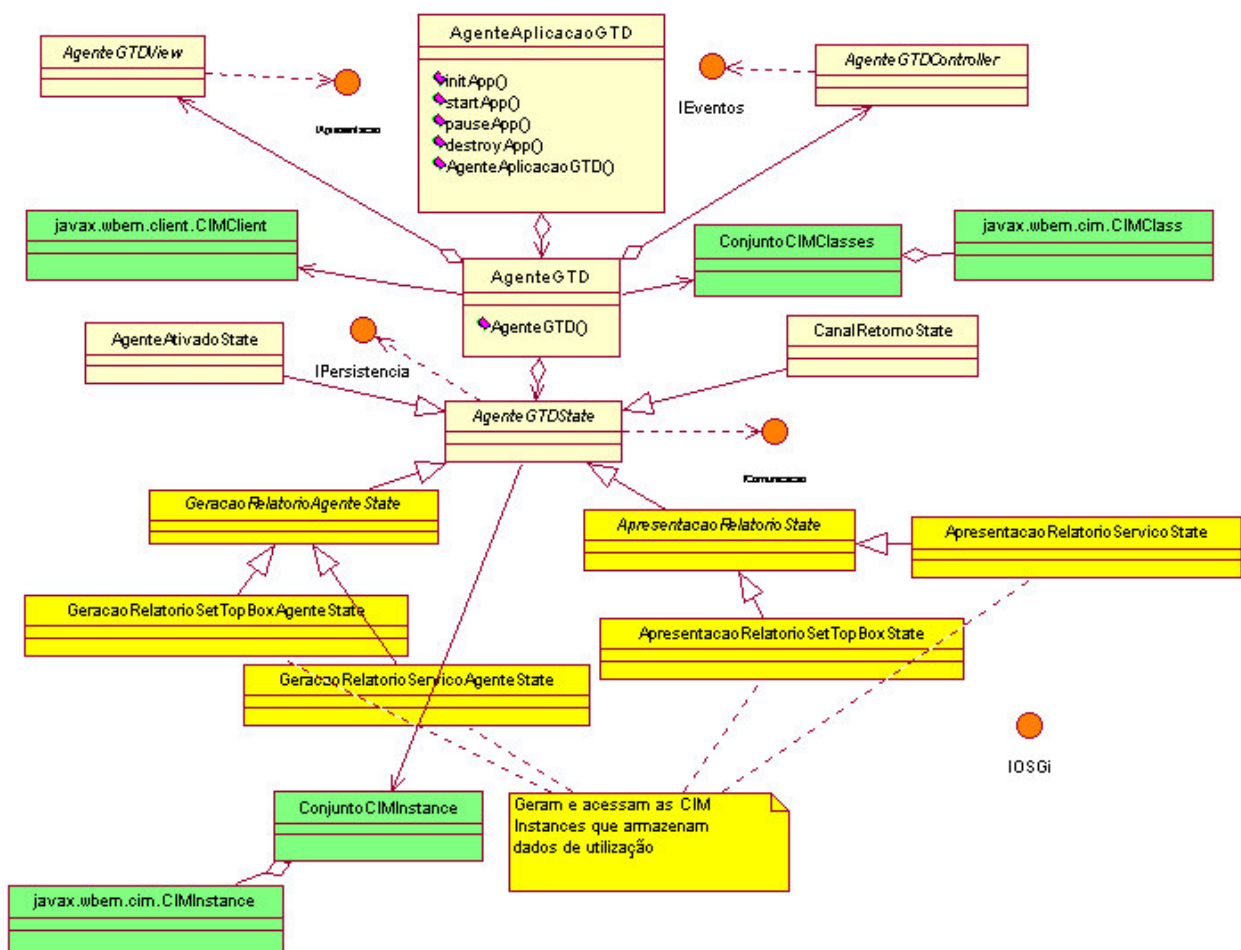


FIGURA 8.12 – DIAGRAMA DE CLASSES PACOTE CONTABILIZAÇÃO - AGENTE

8.2.4.3 Pacote falhas – agente - `gtd.agente.falhas*`

O pacote `gtd.agente.falhas*` possui também a mesma filosofia do gerente, porém aqui o foco é na verificação de falhas, e não na solução das mesmas. Nota-se também que as falhas de serviços são coletadas pela interface OSGi, e devem ser enviadas aos elementos WBEM adequados (CIM Client). Deste modo, o OSGi atua também como **provedor** WBEM.

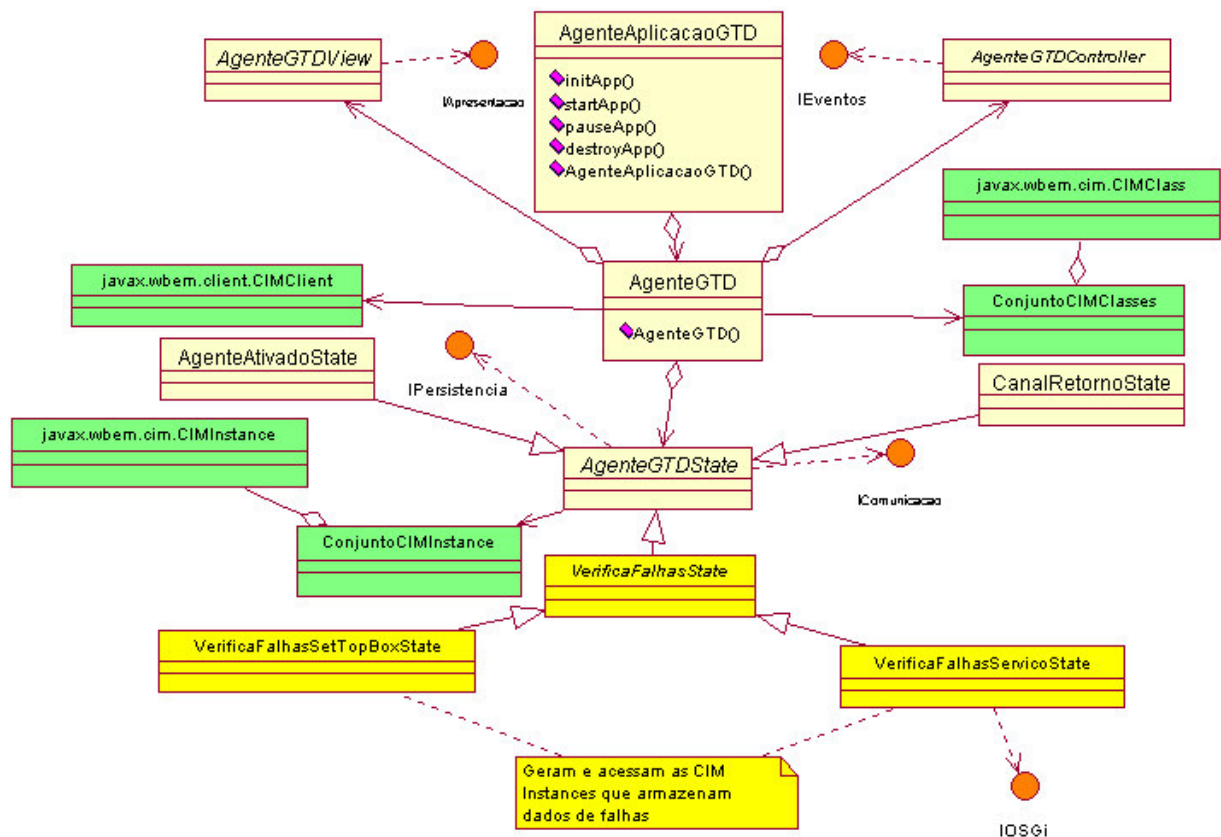


FIGURA 8.13 – DIAGRAMA DE CLASSES PACOTE FALHAS – AGENTE

8.2.4.4 Pacote desempenho – agente - gtd.agente.desempenho*

O pacote gtd.agente.desempenho* possui também a mesma filosofia do pacote de falhas, porém com 2 estados que modelam a coleta de dados de desempenho; dados estes que são posteriormente armazenados nas instâncias CIM adequadas.

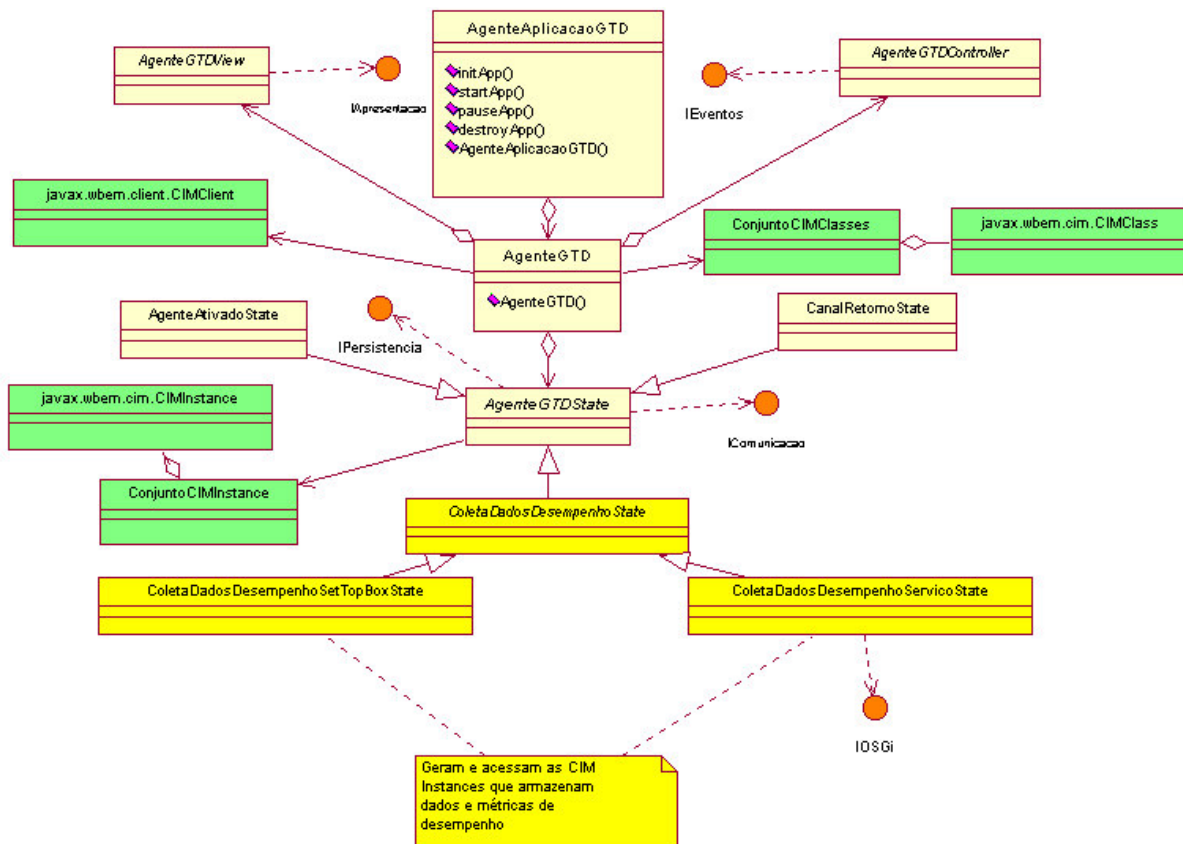


FIGURA 8.14 – DIAGRAMA DE CLASSES PACOTE DESEMPENHO – AGENTE

8.2.4.5 Pacote segurança – agente - gtd.agente.seguranca*

Mais simples que do lado gerente, o pacote `gtd.agente.seguranca*` modela apenas o estado de alteração de autenticação, utilizando-se para isto da classe `AlteraAutenticacaoAgenteState`.

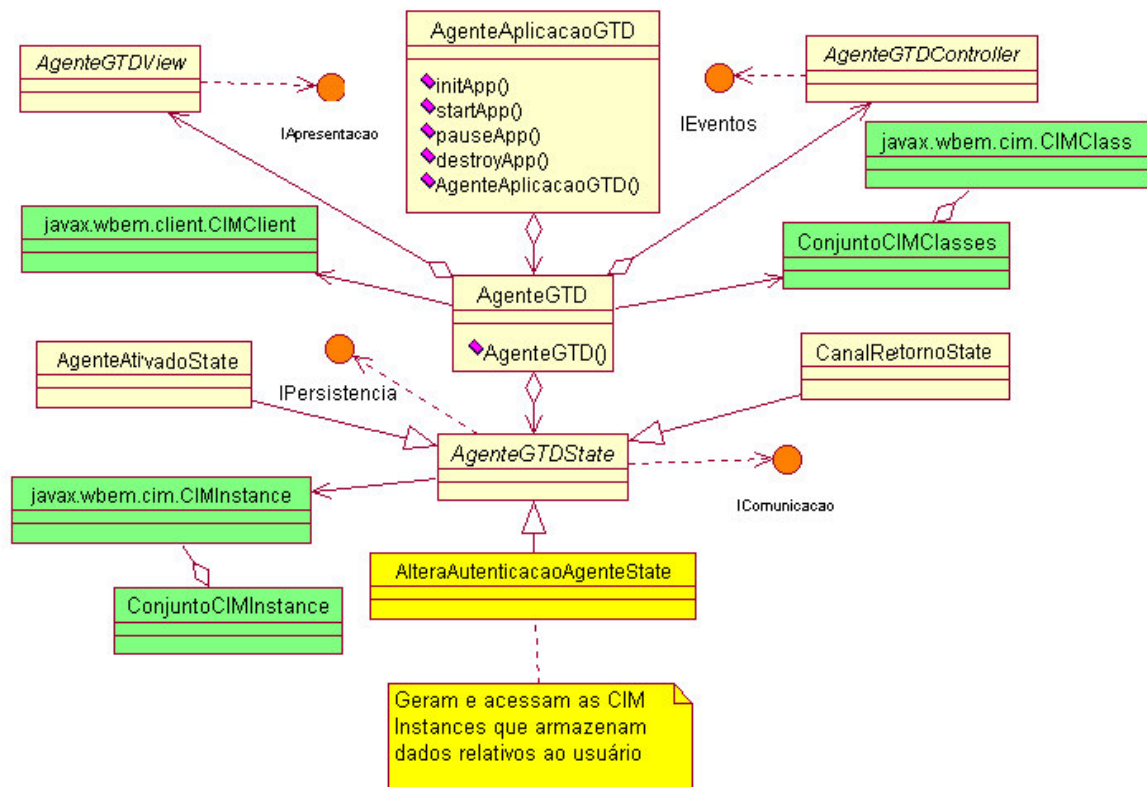


FIGURA 8.15 – DIAGRAMA DE CLASSES PACOTE SEGURANÇA – AGENTE

8.3 Testes de verificação

Os testes de verificação são necessários ao longo da construção do sistema, para confirmar se o sistema está sendo construído corretamente. Os testes indicados neste item (ver tabela 8.1) devem ser realizados ao longo da construção desta arquitetura. Nota-se na tabela que, além dos testes de verificação de rede, são recomendados testes para as funções básicas de cada um dos subsistemas de gerência (os testes para a integração entre os subsistemas são tratados adiante).

TABELA 8.1 – TESTES DE VERIFICAÇÃO

| ID | Item Testado | Teste | Saída Esperada |
|----|----------------------------|---|--|
| 1 | Rede | Gerente realiza comunicação com equipamento de rede via canal de broadcast e canal de retorno | Sistema gerente troca dados de gerência com set-top box |
| 2 | Gerência de Contabilização | Comunicação com sistema externo de autenticação e autorização | Comunicação com sistema externo de autenticação e autorização ocorre sem problemas |
| 3 | Gerência de Contabilização | Coleta de dados de contabilização para geração de relatórios | Relatórios são gerados de acordo |
| 4 | Gerência de Falhas | Deteção de falhas na set-top box | Gerente recebe comunicação de falhas da set-top box e procura solução |
| 5 | Gerência de Falhas | Deteção de falhas de serviço | Gerente recebe comunicação de falhas de serviço e procura solução |
| 6 | Gerência de Desempenho | Coleta e envio de dados de desempenho da set-top box | Gerente recebe os dados de performance da set-top box |
| 7 | Gerência de Desempenho | Envio de dados de desempenho de serviço | Gerente recebe os dados de performance da serviços |
| 8 | Gerência de Configuração | Alteração de uma configuração de uma set-top box | Configuração da set-top box é alterada de acordo |
| 9 | Gerência de Configuração | Alteração de uma configuração de um serviço | Configuração de serviço é alterada de acordo |
| 10 | Gerência de Segurança | Configuração do perfil de um usuário (inclusive método de autenticação) | Perfil configurado de acordo |
| 11 | Gerência de Segurança | Acesso remoto via web (administrador) | Administrador acessa aplicação |
| 12 | Gerência de Segurança | Comunicação com subsistema de contabilização | Comunicação com subsistema de contabilização ocorre sem problemas |

8.4 Estratégia de integração

Neste item é apresentada uma estratégia para a integração de todos os subsistemas (pacotes) que formam esta arquitetura (ver tabela 8.2). A estratégia aqui apresentada é a mais indicada por apresentar a integração de pacotes importantes desde o início (Configuração e Falhas), de modo que o sistema já possa apresentar funcionalidades interessantes ao usuário final sem ainda estar completamente desenvolvido.

TABELA 8.2 – ESTRATÉGIA DE INTEGRAÇÃO

| Etapa | Atividade |
|--------------|---|
| 0 | Definição do ambiente e ferramentas de programação |
| 1 | Construção do módulo gerenciador gerente e agente |
| 2 | Construção do subsistema de configuração (gerente e agente) |
| 3 | Construção do subsistema de falhas (gerente e agente) |
| 4 | Construção das interfaces (agente e gerente) com o mundo externo |
| 5 | Integração do subsistema de falhas com subsistema de configuração |
| 6 | Construção do subsistema de contabilização (gerente e agente) |
| 7 | Construção do subsistema de segurança (gerente e agente) |
| 8 | Integração do subsistema de contabilização com subsistema de segurança |
| 9 | Construção do subsistema de desempenho (gerente e agente) |
| 10 | Integração do subsistema de desempenho com subsistema de configuração |
| 11 | Realização dos testes de validação (casos de uso) |

8.5 Testes de integração

De acordo com a estratégia de integração definida anteriormente, devem ser realizados testes de integração (ver tabela 8.3) a cada etapa de integração completada, de modo que os subsistemas interajam entre si conforme esperado.

TABELA 8.3 – TESTES DE INTEGRAÇÃO

| ID | Teste | Sistema 1 | Sistema 2 | Saída Esperada |
|-----------|--|------------------|------------------|---|
| 1 | Alteração de configuração de um serviço na verificação de uma falha | Configuração | Falhas | Configuração do serviço é realmente alterada com a verificação de uma falha |
| 2 | Alteração de configuração de uma set-top box na verificação de uma falha | Configuração | Falhas | Configuração da set-top box é realmente alterada com a verificação de uma falha |
| 3 | Alteração de configuração de um serviço na verificação de um problema de desempenho | Configuração | Desempenho | Configuração do serviço é realmente alterada com a verificação de um problema de desempenho |
| 4 | Alteração de configuração de uma set-top box na verificação de um problema de desempenho | Configuração | Desempenho | Configuração da set-top box é realmente alterada com a verificação de um problema de desempenho |
| 5 | Usuário se autentica com erro na rede | Segurança | Contabilização | Subsistema de Segurança é acionado pelo Subsistema de Contabilização |
| 6 | Usuário solicita o uso de um serviço | Segurança | Contabilização | Subsistema de Segurança é acionado pelo Subsistema de Contabilização |

8.6 Testes de validação

Os testes de validação devem ser realizados no término da construção do sistema, para que as funcionalidades propostas sejam validadas perante o usuário final. O melhor modo de se realizar estes testes de validação é simular cada um dos casos de uso definidos no Apêndice A deste trabalho.

Se possível, dentro da bateria de testes, também deve ser observado se os requisitos não-funcionais de cada um dos subsistemas, conforme definidos no item 5.2 deste trabalho, estão sendo atendidos.

9. ESTUDO DE CASO DE GERENCIAMENTO DE TV DIGITAL: UM CENÁRIO E SUA IMPLEMENTAÇÃO

Neste capítulo é definido um estudo de caso de gerenciamento de TV digital apoiado na arquitetura definida nesta pesquisa. Para tanto, é apresentado um cenário e as respectivas plataformas de hardware e software a serem adotadas tanto do lado gerente como do lado agente. Os resultados desta implementação são analisados no fim deste capítulo.

9.1 Objetivo do estudo de caso e testes propostos

No estudo de caso aqui apresentado, serão implantados 2 módulos da arquitetura desenhada anteriormente, tanto do lado gerente como do lado agente:

- Módulo de gerenciamento de configuração
- Módulo de gerenciamento de falhas

Para tanto, é necessário construir também o módulo principal de controle da aplicação (tanto do lado gerente como do lado agente). A comunicação WBEM entre gerente e agente e a correta execução da aplicação de gerenciamento, principalmente do lado agente, são outros pontos de atenção deste estudo de caso.

Seguindo as definições do capítulo 8 deste trabalho, será necessário também construir os módulos que realizam a interface com o mundo externo, tanto do lado gerente como agente. A implementação destes elementos para este estudo de caso pode ser realizada de forma que os mesmos não tenham todas as funções (descritas anteriormente) disponíveis, porém apenas aquelas que são necessárias para a realização do estudo. Futuramente, o sistema pode ser

estendido do modo desejado a outras situações. O importante é que estas interfaces com o mundo externo foram projetadas de modo independente do resto do sistema, e podem ser alteradas posteriormente sem grandes problemas.

Para que este estudo de caso seja levado a bom termo, será seguida a seqüência abaixo (resumida) de desenvolvimento dos itens envolvidos, conforme já detalhado na estratégia de integração apresentada neste trabalho (a proposta abaixo é um subconjunto da estratégia de integração apresentada anteriormente, vide item 8.4):

1. Definição do ambiente e ferramentas de programação.
 2. Construção do módulo gerenciador de gerente e agente.
 3. Construção do módulo de configuração (gerente e agente).
 4. Construção do módulo de falhas (gerente e agente).
 5. Construção dos módulos de interface com o mundo externo (gerente e agente) inclusive interface com o usuário.
-
1. Integração do módulo de falhas com configuração (gerente e agente).

9.2 Ferramentas utilizadas

Seguem abaixo as plataformas de hardware e software utilizadas para este estudo de caso.

9.2.1 Agente

TABELA 9.1 – CONFIGURAÇÃO DO AGENTE NO ESTUDO DE CASO

| Item | Plataforma |
|---------------------|--|
| Java | Java 2 SDK 1.4.2_10 |
| Plataforma OSGi | Knopflerfish OSGi 1.3.4 |
| XML Parser | OSGi bundle |
| HTTP Server/Client | OSGi bundle |
| Plataforma WBEM | Wbemservices 1.0.2 |
| CIM Schema | Versão 2.9 |
| Sistema Operacional | Linux |
| Hardware | Set-Top Box ou Hardware Plataforma Intel com processador de baixo poder e memória de até 64 Mbytes |
| Middleware | MHP 1.1.1 |
| Aplicação | agente.jar |

9.2.2 Gerente

TABELA 9.2 – CONFIGURAÇÃO DO GERENTE NO ESTUDO DE CASO

| Item | Plataforma |
|---------------------|------------------------------|
| Java | Java 2 SDK 1.4.2_10 |
| Plataforma OSGi | Knopflerfish OSGi 1.3.4 |
| XML Parser | OSGi bundle |
| HTTP Server | Apache 2.0.53 |
| Plataforma WBEM | Wbemservices 1.0.2 |
| CIM Schema | Versão 2.9 |
| Banco de Dados | MySQL 4.1 |
| Sistema Operacional | Linux |
| Hardware | Pentium 3 com 256 Mbytes RAM |
| Aplicação | gerente.jar |

9.3 Topologia

A topologia adotada para este estudo de caso é a apresentada na figura 9.1. Nela, tanto o canal de broadcast como o canal de retorno foram emulados por uma rede Ethernet.

Esta topologia foi adotada pela falta de ambiente disponível de TV digital via *broadcast*. Entretanto, existe o interesse de se utilizar o ambiente FlexTV (FINEP, 2004b) para implementação quando o mesmo estiver disponível para testes (no capítulo posterior são listadas algumas possibilidades de integração deste trabalho com o FlexTV).

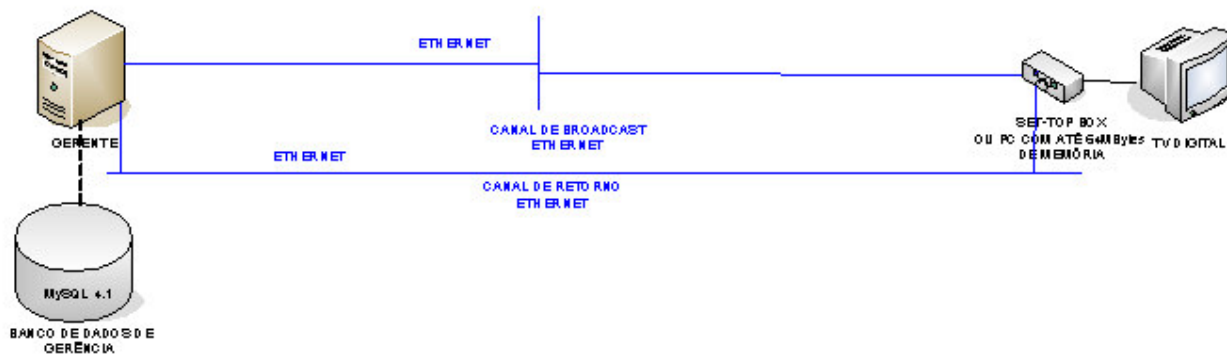


FIGURA 9.1 – TOPOLOGIA

9.4 Resultados da Implementação

Neste ponto são apresentados os resultados da implementação realizada.

9.4.1 *Framework* baseado em estados

O *framework* baseado em estados foi aplicado na construção dos módulos internos do gerente e agente. A utilização do mesmo permitiu que houvesse uma boa independência e também uma grande flexibilidade por parte dos vários pacotes e classes que compõem este sistema.

Tanto do lado gerente como agente, a inteligência da troca de estados foi implementada em uma única classe, `gerenteGTD` e `agenteGTD`. Deste modo pôde-se ter um maior controle da máquina de estados.

Alguns eventos podem não gerar nenhuma alteração de estado. Estes também foram tratados, de modo que quando um deles ocorre, as classes `gerenteGTD` e `agenteGTD` apenas mantêm o estado atual, sem realizar nenhuma nova instanciação.

Para o estudo de caso específico de configuração e falhas, fez-se necessária a implementação dos seguintes estados abaixo (nota-se o maior número de estados no lado do gerente). Estes estados foram selecionados por terem alguma relação com o módulo de gerenciamento e/ou falhas:

- Gerente:

```
final int ESTADO_INICIO=-1;
final int GERENTE_ATIVADO=1;
final int ADMINISTRADOR_AUTENTICADO=2;
final int COMPARA_ARMAZENA_CONFIG=3;
final int ALTERA_CONFIGURACAO_AUTOMATICA_SET_TOP_BOX=4;
final int ALTERA_CONFIGURACAO_AUTOMATICA_SERVICO=5;
final int ALTERA_CONFIGURACAO_SET_TOP_BOX=6;
final int ALTERA_CONFIGURACAO_SERVICO=7;
final int CRIA_SERVICO=8;
final int ANALISA_FALHA_SET_TOP_BOX=9;
final int ANALISA_FALHA_SERVICO=0;
```

- Agente:

```
final int ESTADO_INICIO=-1;
final int AGENTE_ATIVADO=1;
final int CANAL_RETORNO=2;
final int ALTERA_CONFIGURACAO_SET_TOP_BOX_AGENTE=3;
final int ALTERA_CONFIGURACAO_SERVICO_AGENTE=4;
final int VERIFICA_FALHA_SET_TOP_BOX=5;
final int VERIFICA_FALHA_SERVICO=6;
```

Estes estados seguem a máquina de estados anteriormente definida no item 8.2.3.

Segue abaixo um exemplo da execução do gerente (ver figura 9.2), onde o primeiro estado, Gerente Ativado (no qual o administrador ainda não está autenticado) é instanciado. O que caracteriza um estado é a instanciação das 3 classes do modelo MVC, neste caso `AgenteAtivadoState`, `AgenteAtivadoView` e `AgenteAtivadoController`. Este modelo MVC será detalhado mais adiante.



FIGURA 9.2 – ESTADO INICIAL - GERENTE

Neste ponto o sistema fica no aguardo de um evento para que possa realizar alguma função de gerenciamento. Depois de processada esta função, será analisada a troca ou não de estado de acordo com o evento gerado. Caso o administrador se autentique em algum momento, o sistema passa ao estado 2 – ADMINISTRADOR AUTENTICADO (ver figura 9.3).

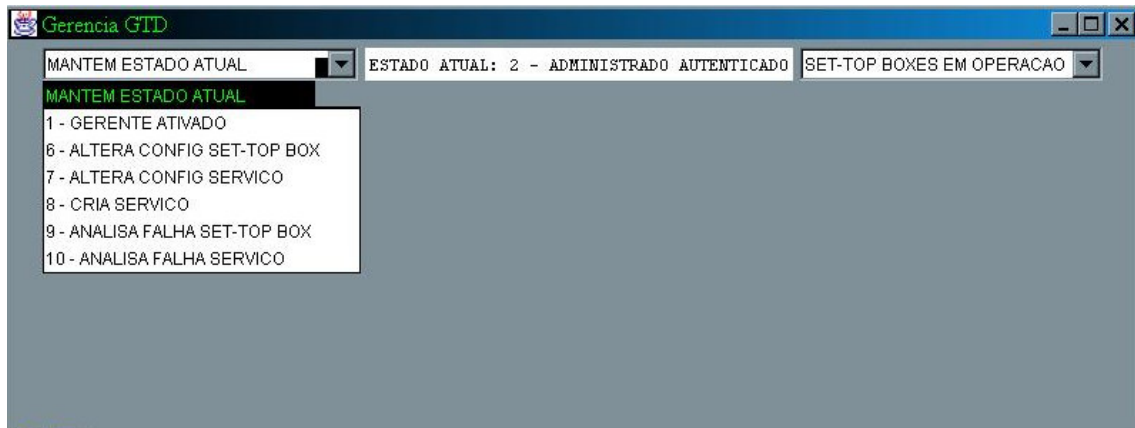


FIGURA 9.3 – ESTADO ADMINISTRADOR AUTENTICADO - GERENTE

Como os estados 1- GERENTE ATIVADO e 2 – ADMINISTRADOR AUTENTICADO são os estados principais do lado gerente, pode-se, dependendo do evento gerado, haver uma transição para qualquer outro estado possível do lado gerente. Por outro lado, caso este sistema esteja no estado 6 –ALTERA CONFIGURAÇÃO SET-TOP BOX, apenas os estados 1 e 2 (e o próprio estado 6) poderiam ser os próximos estados. Isto segue a máquina de estados anteriormente definida neste trabalho. Neste estado 6, também é possível verificar quais são as *set-top boxes* armazenadas no sistema de gerência (ver figura 9.4, lado direito), identificadas aqui pelo seu endereço físico.

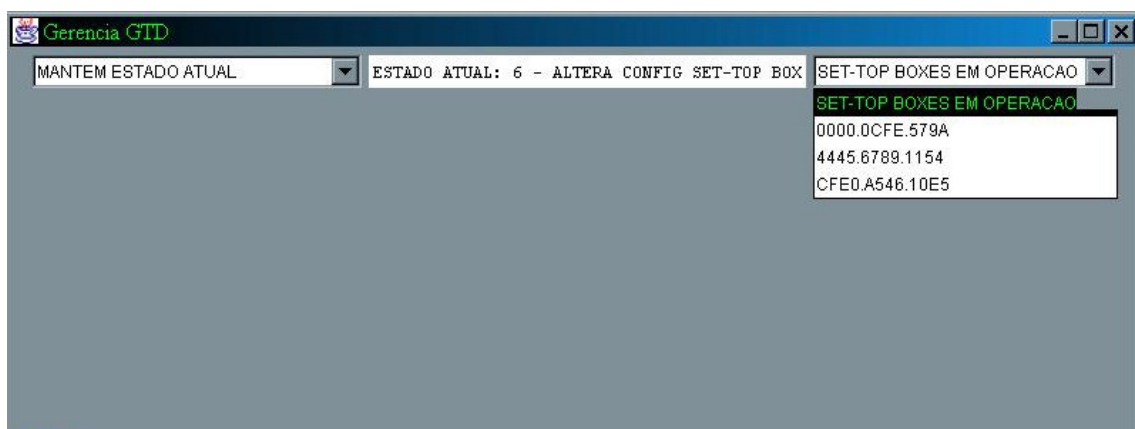


FIGURA 9.4 – ESTADO ALTERA CONFIGURAÇÃO SET-TOP BOX - GERENTE

É importante citar que estes exemplos mostram o **funcionamento interno** da aplicação aqui desenvolvida, sendo que tais alterações de estados são transparentes ao usuário da aplicação, tanto do lado gerente (administrador) como do lado agente (usuário final).

Este *framework* de estados adotado, embora bastante eficiente, tem o inconveniente de gerar um grande número de classes (no mínimo três por estado). Em sistemas com máquinas de estado muito complexas, este fato pode levar a um altíssimo crescimento no número de classes a serem desenvolvidas, além de que o controle da troca de estados ficará bem mais complexo.

9.4.2 MVC

Como dito anteriormente, para todos os estados, tanto do lado gerente como do lado agente, devem ser instanciadas as respectivas classes de State, Controllers e Views. Utilizando-se esta metodologia, é possível que se defina uma classe principal View e outra classe principal Controller, de modo que todas as classes que estão associadas a um estado **herdem** destas 2 classes principais a maioria dos métodos de entrada e saída. Deste modo, facilita-se a manutenção e independência deste sistema.

Apesar de ter sido implementado somente parcialmente, o desenvolvimento do modelo MVC foi bastante facilitado pela definição das 2 classes principais View e Controller, pois deste modo as funcionalidades de captura de eventos (controller) e alteração de tela (view) puderam ser escritas em um único ponto (nestas superclasses).

Para simplificação, foram implementadas apenas entradas e saídas simples diretamente das telas.

9.4.3 OSGi

Para a plataforma OSGi, adotou-se o Knopflerfish 1.3.4, que é uma ferramenta aberta para o controle e desenvolvimento de aplicações via OSGi (<http://www.knopflerfish.org>).

Com ele, é possível controlar, via gerente, uma série de serviços dentro do agente. Também é possível retirar estatísticas dos mesmos para que estas sejam tratadas por alguma aplicação.

O Knopflerfish oferece um ambiente específico para o controle de serviços, a partir de onde é possível verificar quais serviços estão executando em um agente, quais estão pausados, quais necessitam atualização, etc. No exemplo da figura abaixo, nota-se por exemplo os serviços de HTTP e de Log (que será utilizado para retirada de estatísticas e verificação de falhas de serviço) – do lado esquerdo da figura -, entre outros.

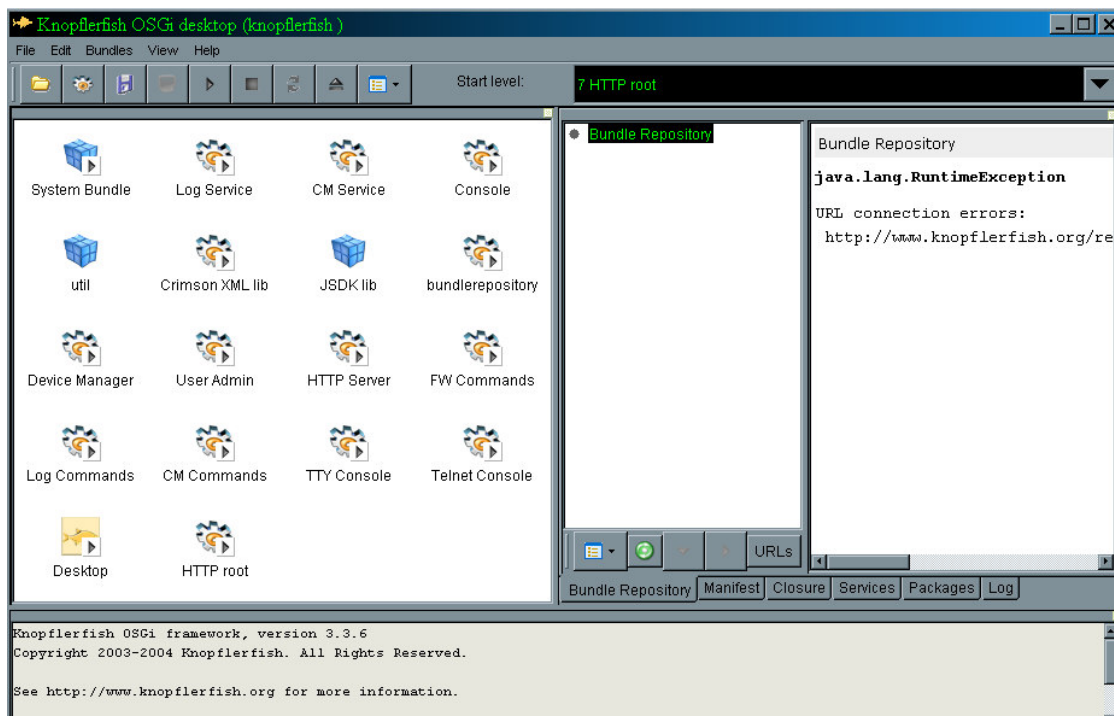


FIGURA 9.5 – KNOPFLERFISH - OSGi

Outros serviços interessantes que podem ser observados na figura 9.5 são o Console e o Telnet Console. Com estes é possível realizar qualquer operação sobre serviços via linha de

comando. Os comandos OSGi são bem específicos, e podem ser enviados do gerente ao agente via Console (com possibilidade de encapsulamento pelo MPEG-2 TS) e no sentido contrário via HTTP. A figura abaixo mostra um exemplo de um Console OSGi.

```

install [-help] [-nodeps] <name;version> ... - Install bundle
start [-help] [-nodeps] <name;version> ... - Install and start bundle
update [-help] [-nodeps] [-check] <name;version> ... - Update bundle
> help
Available command groups (type 'enter' to enter a group):
session - Session commands built into the console
obr - OBR commands
logconfig - Configuration commands for the log.
log - Log commands
framework - Framework commands
configuration - Configuration commands
>

```

FIGURA 9.6 – CONSOLE KNOPLERFISH - OSGi

Via este console, é possível realizar, de modo simples, operações de:

- inclusão de serviços.
- exclusão de serviços.
- atualização de serviços.
- inicialização de serviços.
- colocar serviços que já estiverem rodando em pausa ou mesmo abortar totalmente a execução dos mesmos.

Ou seja, todas as funções necessárias ao sistema de gerência de serviços aqui proposto.

Em cada *set-top box*, deve ser instalada tanto a aplicação de gerenciamento aqui proposta como a plataforma OSGi aqui utilizada (sem a presença do desktop que consumiria muitos

recursos da *set-top box*). Do lado gerente, cada *set-top box* pode ser acessada individualmente (ou em grupo) para que a(s) mesma(s) tenha(m) seus serviços gerenciados.

Também é importante citar que, para que qualquer serviço (aplicação) seja controlado via OSGi, este deve seguir algumas regras de desenvolvimento de forma que o mesmo torne-se um *bundle*, compatível com OSGi (OSGi ALLIANCE, 2004) (lembrando que *bundle* é uma nomenclatura OSGi, e significa na verdade um conjunto, um agregado de serviços). Para tanto, a adição de algumas poucas linhas de código Java e a implementação de uma interface específica são necessárias. Para facilitar este trabalho, já existe um *plug-in* do IBM Eclipse que gera os *bundles* OSGi automaticamente a partir de qualquer aplicação Java (<http://www.knopflerfish.org/eclipse-update>).

Não foi verificado aqui se é possível gerenciar serviços que tenham sido escritos em outra linguagem que não Java apesar do consórcio OSGi afirmar que isto é possível (OSGi ALLIANCE, 2000).

9.4.5 WBEM

A implementação de WBEM utilizada, *Wbemservices* (da Sun Microsystems), também foi construída em Java e a mesma disponibiliza uma API para a utilização das funções WBEM (<http://wbemservices.sourceforge.net>). Esta solução foi construída a partir da Java Specification Request (JSR) 48 (JSR48, 2005), que na verdade é uma especificação Java que foi desenvolvida a partir dos padrões WBEM.

Além disto, o *Wbemservices* já oferece uma CIMOM pronto para uso, e que aceita comandos via HTTP (ver figura 9.7) e RMI. Como no canal de *broadcast* foi adotado o uso de XML puro, fez-se necessário o desenvolvimento de um novo CIMOM que fosse compatível com esta tecnologia. O CIMOM pode ser ativado em qualquer máquina que possua uma JVM, deste modo as mesmas podem rodar sem maiores problemas nas *set-top boxes*. A figura abaixo apresenta um CIMOM HTTP ativado.

```

C:\WINDOWS\system32\cmd.exe - start_cimom

C:\TempEI4\wbemservices>cd cimom/bin

C:\TempEI4\wbemservices\cimom\bin>start_cimom

The JAVA_HOME environment variable must be set to an
existing copy of the Java SDK.

Example: set JAVA_HOME=c:\jdk1.2.2

C:\TempEI4\wbemservices\cimom\bin>set JAVA_HOME=c:\jdk1.2.2

C:\TempEI4\wbemservices\cimom\bin>start_cimom
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Starting up the CIMOM
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

C:\TempEI4\WBEMSE~1\cimom\bin>c:\jdk1.2.2\bin\java -classpath C:\TEMP EI4\WB
EMSE~1\cimom\lib\wbemstartup.jar -Xmx128m -Djava.security.manager -Djava.securit
y.policy=C:\TEMP EI4\WBEMSE~1\cimom\bin\cimom.policy -DBaseDir=C:\TEMP EI4\WBEMSE~
1 -Dpropdir=C:\TEMP EI4\WBEMSE~1\cimom\bin -Dlogdir=C:\TEMP EI4\WBEMSE~1\cimom\log
r org.wbemservices.wbem.bootstrap.StartWBEMServices C:\TEMP EI4\WBEMSE~1\cimom\bi
n\WBEMServices_Win32.properties

```

FIGURA 9.7 – CIMOM HTTP

No momento da ativação do CIMOM HTTP, o mesmo abre 2 portas TCP para requisições (isto apenas no caso do CIMOM HTTP), 5987 e 5988 (ver figura 9.8).

```

U:\>netstat -an

Conexões ativas

Proto Endereço local      Endereço externo    Estado
TCP    0.0.0.0:135          0.0.0.0:0           LISTENING
TCP    0.0.0.0:445          0.0.0.0:0           LISTENING
TCP    0.0.0.0:1103         0.0.0.0:0           LISTENING
TCP    0.0.0.0:3389         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5987         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5988         0.0.0.0:0           LISTENING
TCP    127.0.0.1:1033       0.0.0.0:0           LISTENING
TCP    143.107.111.96:139  0.0.0.0:0           LISTENING
TCP    143.107.111.96:1034 143.107.111.22:445  ESTABLISHED
TCP    143.107.111.96:1099 143.107.111.96:5987 TIME_WAIT
TCP    143.107.111.96:1100 143.107.111.96:1098 TIME_WAIT
TCP    143.107.111.96:1102 143.107.111.96:1098 TIME_WAIT
TCP    143.107.111.96:1103 143.107.111.96:1105 ESTABLISHED
TCP    143.107.111.96:1104 143.107.111.96:5987 ESTABLISHED
TCP    143.107.111.96:1105 143.107.111.96:1103 ESTABLISHED
TCP    143.107.111.96:5987 143.107.111.96:1104 ESTABLISHED
UDP    0.0.0.0:445          *: *
UDP    0.0.0.0:500          *: *
UDP    0.0.0.0:1027         *: *

```

FIGURA 9.8 – PORTAS TCP - CIMOM HTTP

Além do CIMOM, que é o elemento central em uma arquitetura WBEM, a Wbemservices também oferece algumas ferramentas de apoio, por exemplo uma aplicação que permite que sejam visualizadas todas as Classes e Instâncias CIM (ver figura 9.9) a partir de arquivos .mof (detalhados mais adiante).

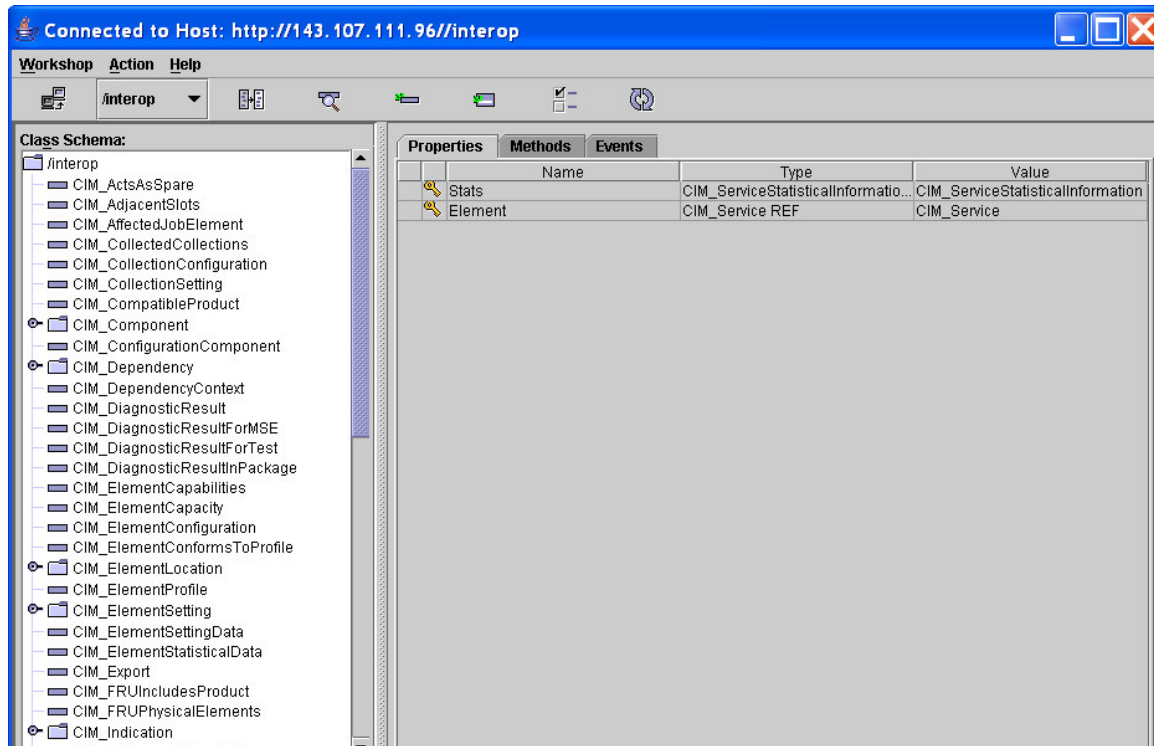


FIGURA 9.9 – VISUALIZAÇÃO DAS CLASSES E INSTÂNCIAS CIM

A API Wbemservices da Sun é bem e completa e segue fielmente a JSR48 (com a definição de Classes, Métodos e Instâncias CIM), entretanto é de difícil desenvolvimento e extensão, o que fez com que boa parte do tempo desta implementação fosse gasta com a mesma.

9.4.6 CIM Schema

Foi adotado o uso do CIM Schema versão 2.9 (DMTF, 2005), mesmo porque esta é a versão que roda com o Wbemservices mais recente da Sun.

Com relação a este CIM Schema, notou-se, ao longo da pesquisa, que o mesmo é até certo ponto bem completo, e que em um primeiro momento extensões a este CIM Schema não são necessárias mesmo em um ambiente de produção de gerenciamento de TV digital.

Algumas classes (e instâncias) CIM definidas neste CIM Schema e de interesse para este projeto (pois simplificaram muito o desenvolvimento desta aplicação de gerenciamento) são por exemplo:

- Device_memory.mof, que descreve os estados da memória de um elemento de hardware (no caso a *set-top box*).
- Device_ports.mof, que descreve as interfaces físicas de um elemento de hardware (no caso a *set-top box*).
- Network_Routes.mof, que descreve as rotas IP presentes em um recurso (no caso a *set-top box*).
- Core_Service.mof, que descreve um serviço. Neste caso o OSGi deve retornar para o WBEM os estados de um serviço, de acordo com este CIM Schema. Deste modo, o OSGi atua como um **provedor** de serviços WBEM, conforme proposto anteriormente.
- Core_Statistics.mof, que descreve as estatísticas, que podem ser utilizadas tanto para *set-top boxes* como para serviços.
- vários outros exemplos.

Notem que todos os CIM Schemas são definidos em arquivos do formato .mof, que na verdade é uma linguagem padrão CIM. No momento em que o CIMOM do Wbemservices é ativado (em uma *set-top box* por exemplo), ele irá ler todos os arquivos .mof necessários para criar as classes CIM e também irá criar as instâncias CIM (que possuem efetivamente os valores que descrevem o estado da *set-top box* ou dos serviços) adequadas, do mesmo modo como na aplicação Wbemservices da figura 9.9. Estas instâncias CIM variam conforme o estado da aplicação do lado agente. Para um estado, alguns objetos CIM serão instanciados;

para outro estado outros objetos CIM o serão. Já do lado gerente, são apenas criadas as classes CIM, e as mesmas são portanto fixas.

Como já citado, o nível de detalhe das especificações do CIM Schema 2.9 já podem ser suficientes para realizar o gerenciamento de *set-top boxes* e serviços.

9.4.7 Persistência

Apesar do MHP definir algumas APIs para o tratamento da persistência do lado agente (ETSI, 2003), foi também desenvolvido aqui um módulo de interface com a memória persistente da *set-top box*. Neste caso, a persistência foi simulada por um arquivo a partir de onde se pode escrever e ler os dados de interesse para a persistência (inclusive o último estado no qual a aplicação agente se encontrava antes de ser destruída).

Orientado a objetos, o módulo de persistência aqui desenvolvido definiu chaves para a escrita e (re) leitura de objetos, de acordo com cada classe instanciada.

Com relação aos objetos CIM, os mesmos foram armazenados em arquivos separados, no formato .mof, e foram carregados (instanciados) sempre que necessário a partir destes arquivos.

9.4.8 MPEG-21

Foram definidos alguns exemplos de metadados descritores de conteúdo de serviços multimídia MPEG-21, através da ferramenta DI Creator (<http://www.enikos.com>) versão 1.0.0-b05 (ao longo da pesquisa também notou-se as poucas opções disponíveis de ferramentas para desenvolvimento de metadados MPEG-21).

Como exemplo, foram criados 2 descritores do mesmo conteúdo, porém um destinado a usuários de um determinado perfil (ouro por exemplo) e outros de um perfil mais simples.

O primeiro descritor apresentado abaixo define um recurso multimídia (no caso uma figura) em um formato específico, no caso **.bmp**.

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS">
  <!--Created by DICreator 1.0.0-b05 licenced to eu - eu - visit us at
http://www.enikos.com-->
  <Container>
    <Descriptor>
      <Statement mimeType="text/plain">Imagens com identificaçãõ do LARC no
formato .bmp</Statement>
    </Descriptor>
    <Item>
      <Descriptor>
        <Statement mimeType="text/plain">Logo LARC com compressao
.bmp</Statement>
      </Descriptor>
      <Component>
        <Resource mimeType="application/octet-stream" ref="file:/U:/MPEG-
21/Larc-bmp.bmp"/>
      </Component>
    </Item>
  </Container>
</DIDL>
```

Já um segundo descritor (abaixo) define o **mesmo** recurso multimídia (no caso uma figura) porém em um outro formato (**.gif**).

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS">
  <!--Created by DICreator 1.0.0-b05 licenced to eu - eu - visit us at
http://www.enikos.com-->
  <Container>
    <Descriptor>
      <Statement mimeType="text/plain">Imagens com identificacao do LARC no
formato .gif</Statement>
    </Descriptor>
    <Item>
      <Descriptor>
        <Statement mimeType="text/plain">Logo LARC com compressao
.gif</Statement>
      </Descriptor>
```

```

    <Component>
      <Resource mimeType="image/gif" ref="file:/U:/MPEG-21/Larc-
gif.gif"/>
    </Component>
  </Item>
</Container>
</DIDL>

```

Ambos os exemplos acima descrevem um mesmo serviço (que na verdade aqui se trata apenas de uma simples figura, uma em um formato com maior qualidade e a outra com um formato em menor qualidade). A mídia que compõe este serviço é o *Resource* MPEG-21, indicado em ambos os metadados (.bmp ou .gif). Com relação à identificação dos mesmos, esta não está indicada aqui, porém o MPEG-21 utiliza como DII o formato Universal Resource Identifier (URI) (ISO, 2002).

Deste modo, dependendo do perfil do usuário (ou até mesmo de outros fatores, como qualidade da rede em um determinado instante de tempo), o mesmo receberia um recurso de maior qualidade ou de menor qualidade, sendo a identificação e descrição destes 2 formatos distintos realizados pelo MPEG-21.

O gerente da rede seria responsável então por apresentar todas as opções de serviço (ou todas as configurações de um mesmo serviço) disponíveis ao usuário (através dos metadados MPEG-21), captar a seleção do usuário, e verificar se ele está habilitado para o consumo daquele serviço. Finalmente, o gerente entregaria este serviço ao usuário através do URI que o identifica.

9.5 Visão geral da implementação

A implementação apresentada neste capítulo mostrou-se bastante completa no sentido de seguir a especificação definida neste trabalho.

A maioria das APIs e ferramentas prontas demonstraram um grande número de funcionalidades interessantes no ambiente de gerenciamento, principalmente o OSGi, que

atendeu muito bem as especificações relativas ao gerenciamento de serviços. Outro ponto positivo é o extensivo CIM Schema já padronizado que foi pesquisado e utilizado. Conforme já dito anteriormente, acredita-se que o mesmo é suficiente para atender aos requisitos de gerenciamento aqui propostos.

Como ponto de atenção destaca-se principalmente a implementação WBEM (Wbemservices), que apresenta, além de alguns problemas de integração, questões de performance (em parte por ter sido escrita em Java). O desenvolvimento de um CIMOM não dependente do HTTP é um ponto crítico porém plenamente factível de ser realizado. Apesar disto, a adoção do WBEM mostrou-se uma solução criativa e que dá maiores “poderes” ao operador desta rede de gerência do que o tradicional SNMP.

Acredita-se portanto que a arquitetura aqui proposta atende os requisitos traçados anteriormente, até mesmo superando as expectativas com relação a este gerenciamento, principalmente no que tange ao controle de serviços.

10. CONCLUSÃO E TRABALHOS FUTUROS

10.1 Conclusão

A arquitetura de gerenciamento aqui proposta mostrou-se, na implementação do estudo de caso anteriormente apresentado, bastante eficiente no que tange ao gerenciamento de serviços e também muito flexível e extensível, devido à adoção da plataforma WBEM.

Nota-se que esta pesquisa apresenta as seguintes inovações e contribuições em relação às arquiteturas listadas no item 4 deste trabalho:

1. O gerenciamento de serviços foi explorado e uma solução factível e eficiente foi apresentada. Este ponto por si só já representa um grande avanço em relação às propostas anteriores, proporcionando uma grande flexibilidade em operações de gerenciamento do dia a dia.
2. O gerenciamento da *set-top box* propriamente dito também é um dos pontos centrais da arquitetura, e a adoção do WBEM traz inteligência às ações da *set-top box* (lembrando que uma Classe ou Instância CIM pode possuir vários métodos que podem executar pequenas ações de reparo, por exemplo). Um dos problemas geralmente apontados pelos críticos do WBEM, de que o CIM Schema do mesmo é restrito, também não procede, pois o CIM Schema utilizado (última versão) (DMTF, 2005) mostrou-se completo o suficiente para a implantação até mesmo em ambientes de produção. Mesmo assim, caso necessário, há sempre a opção de se estender estes objetos de gerenciamento, através da criação de novas classes CIM compatíveis com o formato adotado pelo WBEM. Lembra-se ainda que, caso necessário e para manutenção da compatibilidade, pode-se adotar tradutores SNMP – WBEM na rede, a fim de que investimentos já realizados sejam preservados. Outro ponto a favor do WBEM é que o mesmo se utiliza muito do XML, padrão amplamente utilizado e de fácil processamento e armazenamento.

3. A não obrigatoriedade do uso de IP no canal de *broadcast* possibilita que esta arquitetura seja utilizada em sistemas de TV digital que já estão em uso, não tendo assim que aguardar o advento da IPTV (televisão sobre IP) para que esta arquitetura seja utilizada. Além disto, é possível adaptar facilmente esta arquitetura a uma rede que está evoluindo do MPEG-2 TS para o IPTV preservando assim novamente investimentos e recursos.
4. Pouco explorado em outros sistemas, o gerenciamento da configuração também teve papel de destaque nesta arquitetura, tanto no que tange à *set-top box* como aos serviços. Em uma rede complexa e gigantesca como as de TV digital, é imprescindível que o gerenciamento da configuração esteja contemplado, de modo satisfatório e eficiente.
5. Do mesmo modo que o gerenciamento da configuração, o gerenciamento da segurança também teve uma solução interessante apresentada, embora a mesma, por se tratar de assunto extremamente amplo, necessite de complementações, principalmente no que tange ao controle de acesso de leitura e escrita a informações de sistema do lado agente. Conforme já apresentado, uma das possibilidades seria rodar a aplicação de gerenciamento “ao lado” do middleware, preservando assim o acesso a estas informações, que somente seria realizado pela aplicação de gerenciamento. Apesar disto, um dos pontos fortes desta arquitetura é a independência do método de acesso do usuário (senha, cartão inteligente, etc), contemplando assim a possibilidade de extensão dos métodos de acesso e também deste modo apresentando grande flexibilidade. A definição de perfis de usuários também não foi deixada de lado, sendo este outro fator importante tanto para o usuário final como para a operadora ou produtora de conteúdo.
6. O tratamento dado ao problema do canal de retorno intermitente, que é um dos principais pontos de pesquisa e debate sobre arquiteturas de gerenciamento de TV digital, ganhou uma solução simples porém eficiente, que é o armazenamento das mensagens de gerenciamento no agente até o momento que este canal esteja disponível. Obviamente, este armazenamento depende dos recursos de hardware da *set-top box*, e dados podem ser descartados. Porém, caso isto venha a acontecer, é possível priorizar alguns dados de gerenciamento para que os mesmos tenham prioridade alta ou baixa de descarte. Ainda com relação ao canal de retorno, o tratamento deste problema no que tange ao gerente também apresentou uma solução satisfatória e até certo ponto bem distinta dos sistemas

- atuais, onde o gerente não aguarda a todo o momento mensagens de confirmação, e trata o não envio destas mensagens de modo diferente ao tradicional “*time-out*”. Ou seja, um canal de retorno intermitente não é mais uma experiência “traumática” ao gerente. Para aqueles casos em que o canal de retorno não está disponível em nenhum momento, uma das opções seria agrupar todas as *set-top boxes* que possuem esta característica e enviar apenas comandos que já foram testados em outras *set-top boxes* (com canal de retorno) e não geraram problemas para as mesmas, já que nesta situação, caso ocorra uma falha, o gerente nunca ficará sabendo pois não haverá canal de retorno para que as respostas sejam enviadas ao mesmo.
7. Real integração com um middleware aberto e de mercado (MHP), aproveitando os recursos e APIs oferecidas pelo mesmo, facilitando assim o desenvolvimento da arquitetura. Além disto, como o GEM, proposta ainda bem recente de um middleware universal, baseia-se fortemente no MHP, esta arquitetura prova ser compatível a longo termo no que tange à plataforma de middleware.
 8. A integração com um sistema de autenticação e autorização, apesar de hoje já amplamente difundida, principalmente por questões de bilhetagem, também não foi deixada de lado.
 9. A utilização do MPEG-21 como descritor de conteúdo de serviços multimídia representa mais um item de inovação desta pesquisa. Embora recente, a tecnologia MPEG-21 apresenta-se bem avançada no que se refere à descrição de serviços multimídia. Apesar disto, existem várias outras opções de metadados descritores de serviços que podem ser utilizadas. Para tanto, algumas poucas alterações são necessárias no sistema aqui proposto, pois o mesmo foi desenhado do modo mais independente possível. Um outro item importante é que, embora não contemplado neste trabalho, o MPEG-21 poderia também ser utilizado para o controle e entrega de conteúdo multimídia, permitindo assim que uma gerência fim-a-fim de serviços fosse oferecida. Neste âmbito, o MPEG-21 seria capaz de: indicar quais os serviços estão disponíveis a partir de uma determinada classificação, agrupar serviços similares para facilidade de entrega e maior personalização, controlar a assinatura a serviços, além de listar (em um arquivo *log* por exemplo) todas as operações realizadas em um determinado serviço.

Com esta lista, não exaustiva, e lembrando que se trabalhou o tempo todo com padrões e tecnologias abertas (e de custo zero), pode-se afirmar que a arquitetura de gerenciamento aqui proposta resolve os problemas acima citados, que hoje ainda são questões em aberto em ambientes de gerenciamento de redes de TV digital.

Com um modelo de negócios de gerenciamento bem definido, é possível, com esta arquitetura, agregar valor não só às operadoras transmissoras mas também aos geradores de conteúdo, sem deixar de lado o usuário final.

Entretanto, é importante citar que esta pesquisa utilizou algumas tecnologias bem recentes e, sem dúvida, agregou em uma mesma proposta plataformas bem distintas. Portanto, é necessário que a própria arquitetura amadureça (mesmo porque no estudo de caso apresentado apenas uma parte da mesma foi implementada) para que esta possa ser aplicada em ambientes mais amplos, até mesmo ambientes de produção.

10.2 Trabalhos Futuros

Apesar da pesquisa aqui apresentada ter alcançado seus objetivos em vários aspectos, alguns outros pontos futuros de desenvolvimento foram deixados em aberto.

A percepção da necessidade do desenvolvimento destes novos aspectos foi notada tanto no início dos trabalhos como também, e principalmente, durante o decorrer desta pesquisa.

Abaixo são listados os aspectos mais interessantes para uma futura extensão deste trabalho:

1. Definição de uma arquitetura de gerenciamento da segurança mais completa, que contemple também o uso de chaves criptográficas e aspectos de segurança no que tange ao acesso aos recursos da *set-top box*, até mesmo detalhando permissões para a execução de aplicações baixadas dinamicamente, via canal de retorno ou mesmo canal de *broadcast*. A integração com sistemas locais de autenticação na *set-top box* (por exemplo sistemas criadores de contas e filtros de conteúdo) também são de interesse.

2. Extensão da arquitetura para que contemple também o gerenciamento de uma rede residencial controlada pela *set-top box*. Este ponto é muito interessante pois permite o reuso desta mesma arquitetura em novas pesquisas sobre redes residenciais, assunto hoje também muito debatido e de interesse tanto de universidades como empresas. A vantagem do trabalho aqui apresentado é o de ter definido o uso da tecnologia OSGi, que é facilmente extensível para efetuar o gerenciamento de recursos e serviços dentro de uma rede residencial, através do fornecimento de *bundles* específicos (compatíveis com Universal Plug and Play, por exemplo) para tanto (OSGi Alliance 2004).
3. Ainda na área de gerenciamento de home networks via *set-top box*, também seria possível o gerenciamento do consumo de conteúdos multimídia dentro dos elementos da home network (por exemplo aparelhos de som). Como já foi adotado o padrão MPEG-21 para descrição de conteúdo de serviços multimídia, este poderia ser reaproveitado utilizando-se novamente o OSGi para o gerenciamento do ciclo de vida destes serviços. Além disto, parâmetros de qualidade de serviço (QoS) poderiam ser medidos e alterados dentro deste contexto de *home network*, melhorando assim a experiência do usuário final. Este controle também poderia ser implementado na troca de conteúdos multimídia via TV digital entre usuários finais, análogo ao que ocorre hoje na Internet.
4. Extensão da arquitetura aqui apresentada para uma rede IPTV. Como já citado, acredita-se que esta evolução possa ser executada de modo relativamente simples, pois a arquitetura já foi preparada para tanto. Este é um ponto extremamente atraente pois preserva os investimentos realizados em redes MPEG-2 TS puras.
5. Complementação da arquitetura aqui apresentada para que a mesma contemple o gerenciamento dos elementos de transporte da rede de TV digital. Em particular, poderia ser desenvolvida uma integração com algum sistema já existente.
6. Verificação e alteração em tempo real de parâmetros de qualidade de serviço da rede de TV digital, através dos dados de desempenho coletados periodicamente pelas *set-top boxes* e serviços. Neste caso, faz-se necessário uma integração com as técnicas de QoS empregadas na rede, tanto no canal de *broadcast* como no canal de retorno.

7. Desenvolvimento de um call-center via TV digital, onde o atendente apareceria na tela da TV e estabeleceria deste modo um contato visual com o usuário final. Operações de gerenciamento poderiam ser realizadas em tempo real com a aprovação e análise do usuário.

Além disto, destaca-se também que este trabalho poderia ser integrado com o desenvolvimento do middleware FlexTV do Sistema Brasileiro de TV Digital (SBTVD) (FINEP, 2004b). Neste, um dos módulos desenvolvidos foi o de gerência de perfil de usuário (chamado de GPU) (OS, 2005) onde o usuário tem a possibilidade de configurar, na *set-top box*, perfis de utilização, para fins de personalização e segurança. Ou seja, o GPU é uma espécie de gerenciamento local da *set-top box*. Existe o interesse de se integrar este sistema GPU com a arquitetura de gerência aqui definida, para que deste modo se tenha uma solução de gerência mais completa e descentralizada, dando assim liberdade para que o usuário final possa ter um papel mais ativo neste gerenciamento. Neste caso, a questão da mobilidade (citada no item 8.1.2) também estaria endereçada, pois o gerenciamento de serviços (em específico) poderia ser realizado efetivamente por usuário (lembrando que uma mesma *set-top box* pode ser utilizada por vários usuários (OS, 2005)), e não mais por hardware, como proposto hoje.

Um outro módulo do FlexTV que poderia ser integrável ao sistema aqui desenvolvido é o de gerenciador de aplicações (FINEP, 2004b), que controla o ciclo de vida das aplicações dentro da *set-top box*. Como a aplicação de gerenciamento aqui apresentada controla a instalação, bloqueio e execução de aplicações do lado agente (através do gerente), seria interessante que as mesmas fossem integradas, ou no mínimo, sincronizadas, para que não houvesse situações de conflito entre as mesmas.

REFERÊNCIAS BIBLIOGRÁFICAS

ABERT, SET. **Digital television systems: brazilian tests**, final report. São Paulo, maio 2000.

ADVANCED TELEVISION SYSTEMS COMMITTEE. **Digital television receiver software environment for terrestrial and cable broadcast of data and interactive services**. T3/S17 Specialist Group, Doc.075, fev. 1999.

ALENCAR, MARCELO SAMPAIO DE. **Telefonia digital**. Segunda Edição. São Paulo: Editora Érica, 1999. 312 p.

AIDAROUS, S.; PLEVYAK T. **Telecommunications network management into the 21st century: techniques, standards, technologies and applications**. Primeira Edição. John Wiley & Sons, Inc., 2004. 448p.

ATZORI, L. et al. Multimedia information broadcasting using digital TV channels. **IEEE Transactions on Broadcasting**, Piscataway, vol.43, issue 3, p.242-251, set.. 1997.

_____. Private data broadcasting using digital TV and MPEG-2 transport stream. **Conference Publication 443**. p. 453 – 457. 1997. Apresentado na Conferência 443, IEE, IPA97, 15-17 julho 1997..

AUTRATA, M. DME Overview. In: Network Operations and Management Symposium, 1992. **Networks without bounds**. IEEE: 1992. Vol.2. p. 78-89.

BALABANIAN et al. An Introduction to Digital Storage Media: Command and Control. **IEEE Communications Magazine**. Piscataway, vol.34, issue 11, p.122-127, nov. 1996.

BANDYOPADHYAY, D et al. A video conference management system. **International Journal of Network Management**. v.18, p.182-190, 1998.

BARDELLI et al. Monitoring the Transmission Quality of Digital Television: EUTELSAT's experience and the TESEUS System (Television Surveillance System for EUTELSAT Satellites) In: GLOBAL TELECOMMUNICATIONS CONFERENCE. **IEEE CNF**. Globecom '00, IEEE, vol. 3, dez. 2000, p.1443-1447.

BHATTI, SALEEM; KNIGHT, GRAHAM. On management of CATV full service networks: an European perspective. **IEEE Network**. Piscataway, p.27-39, out. 1998.

BLUMENTHAL, WIJNEN. **User-based security model (USM) for version 3 of the Simple Management Protocol (SNMPv3)**: RFC 2264. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. jan., 1998.

BNDES. **Relatório TV digital: desafio ou oportunidade?**. Estudos setoriais, BNDES, nov.2000.

_____. **Relatório TV por assinatura: panoramas e possibilidades de investimento**. Estudos setoriais, BNDES, 2002.

BOOCH, GRADY; RUMBAUGH, JAMES; JACOBSON, IVAR. **The Unified Modeling Language User Guide**. 5ª Edição. Reading, Massachusetts, EUA: Addison Wesley Longman, Inc, 1999. 483 p.

BOUGANT, F; DELMOND, F; PAGEOT-MILLET, C. The user profile for the virtual home environment. **IEEE Communications Magazine**. Piscataway, vol.41, issue 1, p.93-98, jan.2003

BRANCHEAU, JAMES C.; BROWN, CAROL V. The Management of end-user computing: status and directions. **ACM Computing Surveys**. v.25, n.4, dez. 1993.

BURNETT et al. MPEG21: goals and achievements. **IEEE Multimedia**. Piscataway, p.60-70, dez. 2003.

CABLE TELEVISION LABORATORIES Inc. **CableHome 1.1 specification**. CH-SP-CH1.1-I07-050211, 2005.

CABLEHOME ADDRESS MAPPING MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CABLEHOME CONFIGURATION MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CABLEHOME DEVICE SECURITY MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CABLEHOME GATEWAY DEVICE MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CABLEHOME QOS MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CABLEHOME REMOTE DIAGNOSTICS TOOLS MIB SPECIFICATION. Produzido por CableLabs. Disponível em <http://www.cablelabs.com>. Acesso em 04 abril 2005.

CARVALHO, TEREZA CRISTINA MELO DE BRITO; GABOS, DENIS. **Gerenciamento de redes**. Primeira Edição. São Paulo: Universidade de São Paulo, 2003.

CASE et al. **Simple network management protocol**: RFC 1067. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. ago, 1988.

CASE et al. **Simple network management protocol**: RFC 1098. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. abr, 1989.

CASE et al. **Simple network management protocol**: RFC 1157. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. maio, 1990.

CASE et al. **Structure of management information for version 2 of the simple network management protocol (SNMPv2)**: RFC 1902. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. jan., 1996.

COCHINWALA, M; MOYER, S; SHIM, H. An integrated service management approach using OSGI technology and ACAP. **Second Consumer Communications and Networking Conference, 2005**. CCNC. 2005. p.144 – 149. 3-6 Jan. 2005.

CRINON, REGIS J.; HEREDIA, EDWIN A. The ATSC data *broadcast* standard. **Proceedings of the 2000 ACM workshops on multimedia**. Los Angeles: ACM Press, p.59-62, 2000.

DEMOS, GARY. Advanced television for the United States: status and issues. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES. **Proceedings of the 23rd annual conference on computer graphics and interactive techniques**. Nova York: ACM Press, 1996, p.501-502.

DIGITAL AUDIO-VIDEO COUNCIL. **DAVIC specification part 6:** management architecture and protocols. DAVIC 1.4 part 6, 1998.

DMTF. CIM Schema 2.9. Produzido pelo Distributed Management Task-Force (DMTF).Jan, 2005. Disponível em <http://www.dmtf.org>. Acesso em 3 outubro 2005

DMTF. Specification for CIM Schema over HTTP (DSP 0200). Produzido pelo Distributed Management Task-Force (DMTF). Dez, 2004. Disponível em <http://www.dmtf.org>. Acesso em 3 outubro 2005

EUROPEAN TELECOMMUNICATIONS STANDARD INSTITUTE. **Digital video broadcasting (DVB):** measurement guidelines for DVB systems. ETSI ETR290, maio 1997.

_____.**Digital video broadcasting (DVB):** multimedia home platform (MHP) specification 1.1.1. ETSI TS 102, v1.2.1, jun. 2003.

_____.**Digital video broadcasting (DVB):** specification for service information (SI) in DVB systems. EN 300 468, v1.3.1, fev. 1998.

EVANS, BRIAN. **Understanding Digital TV: The Route to HDTV.** Primeira Edição. Piscataway: New Jersey, 1995. 242 p.

FINANCIADORA DE ESTUDOS E PROJETOS; MINISTÉRIO DA CIÊNCIA E TECNOLOGIA a. **Sistema brasileiro de televisão digital:** formulário de descrição de atendimento a requisitos – gerência de serviços. Primeira Edição. Brasília, nov. 2004.

FINANCIADORA DE ESTUDOS E PROJETOS; MINISTÉRIO DA CIÊNCIA E TECNOLOGIA b. **Sistema brasileiro de televisão digital:** formulário de descrição de atendimento a requisitos – middleware. Primeira Edição. Brasília, nov. 2004.

GAMMA, E. et al. **Design Patterns:** Elements of Reusable Object-Oriented Software. Primeira Edição. Addison Wesley Longman, Inc., 1997. 398p.

GARSCHHAMMER M. et al. Towards generic service management concepts: a service model based approach. **Integrated Network Management Proceedings.** p. 719 – 732. 2001. Apresentado no IEEE/IFIP International Symposium em 14-18 maio 2001.

HARTOG et al. Convergence of residential gateway technology. **IEEE Communications Magazine**. Piscataway, vol.42, issue 5, p. 138-143, maio 2004.

HEGERING, H.; ABECK, S; WIES, R. A corporate operation *framework* for network service management. **IEEE Communications Magazine**. Piscataway, vol.34, issue 1, p.62-68, jan. 1996.

HEMMEN, L. J. G. T. Van. Models Supporting the Network Management Organization. **International Journal of Network Management**. v.10, p.299-314.

HONG, TAEK JU; MI-JOUNG, CHOI; HONG, JAMES W. An efficient and lightweight embedded web server for web-based network management - **International Journal of Network Management**, v.10, p.261-275, 2000.

HOROWITZ, B; MAGNUNSSON, N., NICLAS, K. Telias's service delivery for the home. **IEEE Communications Magazine**. Piscataway, p.120-125, Abril 2002.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, Inc. **IEEE Guide to Software Requirements Specification**: ANSI/IEEE Standard 830-1984. Piscataway, 10 fev. 1984.

_____. **IEEE Guide to Systems Requirements Specification**: ANSI/IEEE Standard 1233-1998. Piscataway, dez. 1998.

INTERNATIONAL TELECOMMUNICATIONS UNION. **A guide to digital terrestrial television broadcasting in the VHF/UHF bands**. Document 11-3/3-ITU, jan. 1996.

_____. **Data communications networks**: information technology, open systems interconnections, systems management, state management function, X.731, ITU-T, 1992.

ISHIKAWA, H. et al. Building smart appliance integration middleware on the OSGi *framework*. **Proceedings of the Seventh IEEE International Symposium on Object-Oriented Real-Time Distributed Computing**. p. 1-8. 2004. Apresentado no IEEE ISORC'04.

ISO/IEC. **MPEG-21 Overview v.5**. Document ISO/IEC JTC1/SC29/WG11/N5231, oct. 2002.

JAVA MANAGEMENT EXTENSIONS WHITE PAPER. Produzido por Sun Microsystems. Disponível em <http://www.sun.com>. Acesso em 02 maio 2005.

JENKINS, RON. Why web-based network monitoring?: leveraging the platform. **International Journal of Network Management**. v.19, p.175-183, 1999.

JONES, J. DVB-MHP/Java TV data transport mechanisms. In: 40th INTERNATIONAL CONFERENCE ON TECHNOLOGY OF OBJECT-ORIENTED LANGUAGES AND SYSTEMS, 2002, Sydney, Australia. **Conferences in Research and Practice Information Technology**. Sydney, v.10, 2002.

JSR 3. Produzido por SUN MICROSYSTEMS. Disponível em <http://java.sun.com>. Acesso em 22 abril 2005.

JSR 48. Produzido por SUN MICROSYSTEMS. Disponível em <http://java.sun.com>. Acesso em 2 novembro 2005.

JSR 222. Produzido por SUN MICROSYSTEMS. Disponível em <http://java.sun.com>. Acesso em 2 novembro 2005.

KANAHI, M.; BEADLE, H. W. PETER. Decentralized approaches for network management. **Computer Communication Review**. v.27, issue 3.

KANG, KYU-CHANG; LEE, JEON-WOO. Implementation of management agents for an OSGi-based residential gateway. **The 6th International Conference on Advanced Communication Technology**. p. 1103-1007. Vol.2, 2004.

KIM, YONGSUK et al. Interactive *broadcast* terminal system using MPEG-2 and MPEG-4. **Proceedings of the IEEE International Symposium on Circuits and Systems**. p. III-682 – III-685. 2000. Apresentado no ISCAS 2000 em 28-31 maio 2001, Geneva, Suíça.

KLIE, T; STRAUB, F. Integrating SNMP agents with XML-based management systems. **IEEE Communications Magazine**. Piscataway, p.76-83. jul. 2004.

LEE, NAM-KYUNG et al. Cooperation system of DSM-CC Data Carousel and MPEG-4 system via satellite. **Information Technology: Coding and Computing, 2002**. Proceedings on International Conference. p.421-424. 8-10 April 2002.

LEVI, MEYER, STEWART. **SNMPv3 applications**: RFC 2263. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. jan., 1998.

LEWIS, D. A review of approaches to developing service management systems. **Journal of Network and Systems Management**. v.8, n.2.

MADUREIRA, OMAR MOORE. **Curso de especialização em gestão de projetos: a formulação do produto dirigida ao cliente**. Primeira Edição. São Paulo: Universidade de São Paulo, 2001.

MARPLES, D; KRIENS, P. The open services gateway initiative: an introductory overview. **IEEE Communications Magazine**. Piscataway, p.110-114, dez. 2001.

MEANDZIJA, B. Manageability of java-based digital TV receivers. **IEEE Journal on Selected Areas in Communication**. Piscataway, v.18, n.5, maio 2000.

MULLER, NATAHN J. Improving network operations with intelligent agents. **International Journal of Network Management**. v.7, p.116-126, 1997.

ORZESSEK, MICHAEL; SOMMER, PETER. **ATM & MPEG2**: Integrating digital video into broadband networks. Primeira Edição. Upper Saddle River: Prentice Hall, 1998, 326p.

OSGi ALLIANCE. **About the OSGi service platform white paper**. San Ramon: revisão 3.0. 12 jul. 2004

_____. **Open services gateway specification overview**. San Ramon: Versão 1.0, jan. 2000.

O'DRISCOLL, GERARD. **The essential guide to digital set-top boxes**. Primeira Edição. Upper Saddle River: Prentice Hall, nov.1999, 295p.

OS, MARCELO DUTRA. **Sistema brasileiro de televisão digital**: especificação técnica de objeto: software – Gerenciador de Perfil de Usuário. Primeira Edição. São Paulo, set. 2005, 71p.

READ, KRIS; TICHKOSKY LANCE. Refactoring J2ME midlets to use the state pattern. **University of Calgary, Department of Computer Science**. Fevereiro, 2003.

REIMERS, U. The DVB project: digital television for Europe. **IEEE Coloquium**. Piscataway, vol.1995, issue 142, p.1-7, jun. 1995.

RIGNEY et al. **Remote Authentication Dial-in User Service (RADIUS): RFC 2058**. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. Jan, 1997..

ROSE, BILL. Home networks: a standard perspective. **IEEE Communications Magazine**. Piscataway, vol.39, issue 12, p.78-85, dez. 2001.

ROSE, M. T. **The simple book**: an introduction to management of TCP-IP internets. Segunda Edição. Upper Saddle River: Prentice-Hall, 1991, 347p.

SCHULZRINNE, H. et al. Ubiquitous computing in home networks. **IEEE Communications Magazine**. Piscataway, vol.41, issue 11, p.128-135, nov. 2003.

SCHWALB, EDWARD M. **iTV handbook**: standards and specifications. Upper Saddle River: Prentice Hall, 2003.

SILVEIRA, REGINA MELO. **Tecnologias para aplicações de banda larga**. Primeira Edição. São Paulo: Universidade de São Paulo, 2002.

SLOMAN et al. An architecture for managing distributed systems. In: DISTRIBUTED COMPUTING SYSTEMS, 1993. **Proceedings of the Fourth Workshop on Future Trends of Distributed Computing Systems**. IEEE Press, set. 1993. p.40-46.

SOFTWARE ENGINEERING INSTITUTE. **A systems Engineering Capability Maturity Model**. Hanscom: versão 1.1, Novembro 1995, 297 p.

SOMMERVILLE, IAN. **Software Engineering**. Sétima edição. Addison-Wesley, 2004, 784p.

STALLINGS, WILLIAM. **Local and metropolitan area networks**. Fifth Edition. Upper Saddle River: Prentice Hall, 1997. 605 p.

_____. **Network Management**. Primeira Edição. Piscataway: IEEE Computer Society Press, 1993. 254p.

_____. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**. Terceira Edição. Addison-Wesley Professional, dez. 1998. 640p.

SYSTEM CONCEPT FOR INTERACTIVE BROADCASTING CONSUMER TERMINALS. Produzido por SAMBITS. Disponível em <http://www.irt.de/sambits>. Acesso em 17 maio 2005.

TADASHI, I. Interactivity in Broadcasting and its Application to ISDB Services. **IEEE Transactions on Broadcasting**. Piscataway, v.42, n.3, set. 1996.

TEGER, S.; WALKS, D.J. End-user perspectives on home networking. **IEEE Communications Magazine**. Piscataway, vol.40, issue 4, p.114-119, abril 2002.

TELEFONICA. **OSGi management: a JMX approach**. Espanha: draft, versão 0.1, WP3-Task-0403-1, 2004.

TERPLAN K. **Communications Network Management**. Englewood Cliff: Prentice-Hall, 1992.

THOMPSON, PATRICK J. Web-based enterprise management architecture. **IEEE Communications Magazine**. Piscataway, p.80-86, março 1998.

VANG, S; ZUNDEL, J-P.L. The emergence of middleware in home telecommunication equipment. **IEEE Communications Magazine**. Piscataway, vol.39, issue 9, p.106-110, jun. 2001.

YASUDA, H.; RYAN, HENRY J. F.; DAVIC and Interactive Multimedia Services. **IEEE Communications Magazine**. Piscataway, p.137-143, set. 1998.

YOON, JEONG-HYUK; HONG, JU-TAEK; HONG, JAMES H. Development of SNMP-XML translator and gateway for XML-based integrated network management. **International Journal of Network Management**. v.13, p.259-276.

YOSHIMURA et al. Integrated services digital broadcasting. In: BROADCASTING CONVENTION, 1992. **IBC international**. jul. 1992. p.350-354.

WHITAKER, JERRY. **DTV: the revolution in digital video**. Second Edition. McGraw-Hill, 1999. 619 p.

WIJNEN, PRESUHN, McCLOGHRIE. **View-based access control model (VACM) for the simple network management protocol (SNMP): RFC 2265**. Produzido pelo Internet Engineering Task Force (IETF). Disponível em <http://www.ietf.org>. jan., 1998.

WOOD, D. The DVB project: philosophy and core system. **Electronics & Communication Engineering Journal**. Piscataway, vol. 9, issue 1, p.5-10, fev. 1997.

ZHANG, HONGGUANG et al. Design and implementation of *broadcast* file system based on DSM-CC data carousel protocol. **IEEE Transactions on Consumer Electronics**. p.929-933. Vol. 50, Issue 3, Ago. 2004.

ZUFFO, MARCELO KNORICH. **A convergência da realidade virtual e internet avançada em novos paradigmas de TV digital interativa**. Primeira Edição. São Paulo: Universidade de São Paulo, 2001.

APÊNDICE A – Detalhamento dos Casos de Uso

Neste apêndice os casos de uso macro definidos no item 6 acima são particionados em casos mais específicos. Os mesmos são detalhados e agrupados de acordo. Algumas premissas são assumidas para os casos de uso macro de modo que a especificação não apresente ambigüidade. Todos os casos de uso aqui apresentados devem ser implementados respeitando a especificação de requisitos definida no item anterior deste trabalho.

A.1 Caso de uso macro 1 – realizar a gerência de configuração

A.1.1 Caso de Uso 1.1 – Configurar as *set-top boxes*

Premissas:

A configuração das *set-top boxes* também contempla a configuração de rede das mesmas.

Este caso de uso pode ser extrapolado para a configuração de um grupo de *set-top boxes*.

Set-top box possui algum identificador único dentro do sistema.

Atores:

Administrador

Relação TMN x OSI:

Elemento x Configuração

Curso normal dos eventos:

1. Administrador seleciona a *set-top box* (conectada à rede) que ele deseja configurar.
2. É apresentada ao administrador a configuração atual da *set-top box*.
3. Administrador seleciona um componente da *set-top box* para configurar e indica qual parâmetro deve ser alterado.
4. Subsistema de configuração envia a alteração de configuração selecionada a *set-top box* (pode ser também enviado um módulo de software neste comando).
5. *Set-top box* informa ao sistema de gerência a conclusão da operação caso o canal de retorno esteja disponível.
6. Subsistema de configuração atualiza a configuração da *set-top box* em sua base de dados.

Exceções / Tratamento das exceções:

- *Set-top box* não confirma a conclusão ou não da operação / Subsistema de configuração não atualiza a configuração da *set-top box* em sua base de dados (caso o canal de retorno não esteja disponível no momento, a *set-top box* deve armazenar a mudança realizada e depois enviar a confirmação ao subsistema de configuração).

A.1.2 Caso de Uso 1.2 – Armazenar/atualizar a configuração de uma *set-top box* que acabou de se autenticar na rede

Premissas:

Apenas serão atualizadas as configurações de *set-top boxes* que efetivamente realizaram a autenticação na rede (via canal de retorno).

Set-top box possui algum identificador único dentro do sistema.

Neste caso de uso a *set-top box* já é conhecida do sistema, ou seja, está presente em sua base de dados.

Atores:

Administrador, usuário final

Relação TMN x OSI:

Elemento x Configuração

Curso normal dos eventos:

1. Com o canal de retorno disponível, usuário se autentica na rede.
2. Configuração da *set-top box* é enviada pela mesma ao subsistema de configuração da gerência.
3. Subsistema de configuração identifica a *set-top box* e compara a configuração enviada com a configuração armazenada.
4. Caso a mesma seja diferente, o administrador é avisado (opcional) e a nova configuração é armazenada. Caso contrário, nada ocorre.

Exceções / Tratamento das exceções:

- Usuário se autentica porém *set-top box* não envia configuração ao subsistema de configuração / Nada ocorre.

A.1.3 Caso de Uso 1.3 – Armazenar a configuração de uma *set-top box* que se autenticou pela primeira vez na rede

Premissas:

Apenas serão atualizadas as configurações de *set-top boxes* que efetivamente realizaram a autenticação na rede (via canal de retorno).

Set-top box possui algum identificador único dentro do sistema.

Atores:

Administrador, usuário final

Relação TMN x OSI:

Elemento x Configuração

Curso normal dos eventos:

1. Com o canal de retorno disponível, usuário se autentica na rede com uma “nova” *set-top box*.
2. Configuração da *set-top box* é enviada pela mesma ao subsistema de configuração.
3. Subsistema de configuração identifica que se trata de uma nova *set-top box*.
4. Subsistema de configuração cria uma nova entrada na base de dados para armazenar a configuração desta nova *set-top box*.
5. Subsistema de configuração verifica qual o padrão de configuração para o modelo de *set-top box* utilizado pelo usuário final.

6. Subsistema de configuração envia as alterações necessárias de configuração a *set-top box* para que a mesma permaneça no padrão da rede (podem também ser enviados módulos de software nesta etapa).
7. *Set-top box* informa ao sistema de gerência a conclusão da operação caso o canal de retorno esteja disponível.
8. Subsistema de configuração atualiza a configuração da *set-top box* em sua base de dados.

Exceções / Tratamento das exceções:

- *Set-top box* não confirma a conclusão ou não da operação de atualização da configuração / Subsistema de configuração não atualiza a configuração da *set-top box* em sua base de dados (caso o canal de retorno não esteja disponível no momento, a *set-top box* deve armazenar a mudança realizada e depois enviar a confirmação ao subsistema de configuração).

A.1.4 Caso de Uso 1.4 – Armazenar a configuração de uma *set-top box* que teve sua configuração alterada por um usuário

Premissa:

Set-top box possui algum identificador único dentro do sistema.

Atores:

Administrador, usuário final

Relação TMN x OSI:

Elemento x Configuração

Curso normal dos eventos:

1. Usuário altera localmente algum parâmetro da configuração da *set-top box* (via hardware ou software).
2. Caso o canal de retorno esteja disponível, a alteração realizada é enviada imediatamente ao subsistema de configuração.
3. Subsistema de configuração atualiza a configuração da *set-top box* em sua base de dados.

Exceções / Tratamento das exceções:

- *Set-top box* não informa ao subsistema de gerência a alteração em sua configuração / Subsistema de configuração não atualiza a configuração da *set-top box* em sua base de dados (caso o canal de retorno não esteja disponível no momento, a *set-top box* deve armazenar a mudança realizada e depois enviar a confirmação ao subsistema de configuração).

A.1.5 Caso de Uso 1.5 – Configurar os serviços do lado gerente

Premissas:

Este caso de uso pode ser extrapolado para a configuração de um grupo de serviços.

É utilizado um sistema de controle de serviços que possua descritores bem definidos de parâmetros destes serviços; MPEG-21 por exemplo (BURNETT et al, 2003).

É possível ter um mesmo serviço configurado de formas distintas dependendo do usuário que irá consumi-lo.

Atores:

Administrador, sistema de controle de serviços

Relação TMN x OSI:

Serviço x Configuração

Curso normal dos eventos:

1. Administrador seleciona o serviço que deseja configurar.
2. É apresentada ao administrador a configuração atual do serviço.
3. Administrador seleciona um parâmetro do serviço para configurar e indica qual variável deve ser alterada.
4. Subsistema de configuração envia o comando selecionado ao serviço.
5. Sistema de controle de serviços confirma a alteração realizada.
6. Subsistema de configuração atualiza a configuração do serviço em sua base de dados.

Exceções / Tratamento das exceções:

- Sistema de controle de serviços não confirma a alteração de configuração / Volta ao passo 4.

A.1.6 Caso de Uso 1.6 – Configurar os serviços do lado agente

Premissas:

Este caso de uso pode ser extrapolado para a configuração de um grupo de serviços.

Os serviços podem ser reconfigurados do lado agente mesmo depois de já estarem rodando, mesmo que isto implique em uma reinicialização do mesmo.

Atores:

Administrador

Relação TMN x OSI:

Serviço x Configuração

Curso normal dos eventos:

1. Administrador seleciona o serviço que deseja configurar (e também a *set-top box* ou grupo de *set-top boxes* que serão afetadas).
2. É apresentada ao administrador a configuração atual do serviço.
3. Administrador seleciona um parâmetro do serviço para configurar e indica qual variável deve ser alterada.
4. Subsistema de configuração envia o comando selecionado ao serviço do lado agente.
5. Agente confirma alteração de configuração de serviço ao gerente.
6. Subsistema de configuração atualiza a configuração do serviço em sua base de dados.

Exceções / Tratamento das exceções:

- *Set-top box* não informa ao subsistema de gerência a alteração na configuração do serviço / Subsistema de configuração não atualiza a configuração do serviço em sua base de dados (caso o canal de retorno não esteja disponível no momento, a *set-top box* deve armazenar a mudança realizada e depois enviar a confirmação ao subsistema de configuração).

A.1.7 Caso de Uso 1.7 – Armazenar a configuração de um novo serviço

Premissas:

É utilizado um sistema de controle de serviços que possua descritores bem definidos de parâmetros destes serviços; MPEG-21 por exemplo.

É possível ter um mesmo serviço configurado de formas distintas dependendo do usuário que irá consumi-lo.

Atores:

Administrador, Sistema de controle de serviços

Relação TMN x OSI:

Serviço x Configuração

Curso normal dos eventos:

1. Novo serviço é disponibilizado no sistema de controle de serviços.
2. Sistema de controle de serviços envia configuração do novo Serviço ao Subsistema de Configuração.
3. Subsistema de configuração verifica que o serviço é realmente novo.
4. Subsistema de configuração armazena a configuração atual do novo Serviço.

Exceções / Tratamento das exceções:

- Serviço já existe / Subsistema de configuração envia um código de erro ao sistema de controle de serviços e avisa o administrador do erro.

A.2 Caso de uso macro 2- realizar a gerência de contabilização

A.2.1 Caso de Uso 2.1 – Contabilizar o uso da rede pelo usuário

Premissas:

Sistema de autenticação e autorização (de usuários), externo ao sistema aqui descrito, realiza todo o controle da autenticação e autorização.

Atores:

Sistema de autenticação e autorização

Relação TMN x OSI:

Rede x Contabilização

Curso normal dos eventos:

1. Sistema de autenticação e autorização avisa ao subsistema de contabilização que usuário se autenticou na rede.
2. Subsistema de contabilização armazena registro de início de contabilização em sua base de dados.
3. Sistema de autenticação e autorização avisa ao subsistema de contabilização que usuário se desligou da rede.
4. Subsistema de Contabilização armazena registro de fim de contabilização em sua base de dados.

Exceções / Tratamento das exceções:

- Não há.

A.2.2 Caso de Uso 2.2 – Usuário final (ou administrador) solicita relatório de utilização da rede

Premissas:

Sistema de autenticação e autorização (de usuários), externo ao sistema aqui descrito, realiza todo o controle da autenticação e autorização.

O canal de retorno deve estar disponível se este caso de uso for realizado pelo usuário final.

Atores:

Usuário final (ou administrador)

Relação TMN x OSI:

Rede x Contabilização

Curso normal dos eventos:

1. Usuário (ou administrador), já autenticado na rede, solicita, via interface na sua TV digital, relatório de utilização de rede.
2. Tipo de relatório requisitado é enviado ao subsistema de contabilização.
3. Subsistema de contabilização cria relatório em algum formato específico (por exemplo XML ou HTML).

4. Subsistema de contabilização envia relatório a *set-top box*.
5. *Set-top box* envia relatório ao interpretador local que o mostra para o usuário na tela de sua TV digital.

Exceções / Tratamento das exceções:

- Não há

A.2.3 Caso de Uso 2.3 – Contabilizar o uso de um serviço pelo usuário

Premissas:

O sistema de autenticação e autorização é responsável pela autenticação de serviços para a entrega dos mesmos aos usuários finais.

Utilização de serviço implica em canal de retorno disponível, pelo menos até o início de uso do mesmo.

Atores:

Sistema de autenticação e autorização

Relação TMN x OSI:

Serviço x Contabilização

Curso normal dos eventos:

1. Sistema de autenticação e autorização avisa ao subsistema de contabilização que usuário iniciou o uso de um serviço.

2. Subsistema de contabilização armazena registro de início de contabilização de serviço.
3. Sistema de autenticação e autorização avisa ao subsistema de contabilização que usuário finalizou o uso de um serviço.
4. Subsistema de Contabilização armazena registro de fim de contabilização de serviço.

Exceções / Tratamento das exceções:

- Não há.

A.2.4 Caso de Uso 2.4 – Usuário final (ou administrador) solicita relatório de utilização de serviços

Premissas:

O sistema de autenticação e autorização é responsável pela autenticação de serviços para a entrega dos mesmos aos usuários finais.

O canal de retorno deve estar disponível caso este caso de uso seja realizado pelo usuário final.

Atores:

Usuário final (ou administrador)

Relação TMN x OSI:

Serviço x Contabilização

Curso normal dos eventos:

1. Usuário (ou administrador), já autenticado na rede, solicita, via interface na sua TV digital, relatório de utilização de serviços.
2. Tipo de relatório requisitado é enviado ao subsistema de contabilização.
3. Subsistema de contabilização cria relatório em algum formato específico (por exemplo XML ou HTML).
4. Subsistema de contabilização envia relatório a *set-top box*.
5. *Set-top box* envia relatório ao interpretador local que o mostra para o usuário na tela de sua TV digital.

Exceções / Tratamento das exceções:

- Não há

A.3 Caso de uso macro 3 – realizar a gerência de falhas

A.3.1 Caso de Uso 3.1 – *Set-top box* apresenta falha e comunica ao sistema de gerência assincronamente

Premissas:

As falhas são identificadas apenas pela *set-top box* e enviadas ao sistema de gerência assincronamente.

Algumas falhas de rede (como queda de link) são identificadas também neste caso de uso.

Atores:

Administrador

Relação TMN x OSI:

Elemento de Rede x Falha

Curso normal dos eventos:

1. *Set-top box*, que possui um agente de gerenciamento e identificação de falhas, verifica que há uma falha em algum módulo.
2. *Set-top box* informa assincronamente ao subsistema de falhas da gerência a falha ocorrida.
3. Subsistema de falhas procura em seu banco de dados uma solução para o problema.
4. Subsistema de falhas, com a solução encontrada (entende-se aqui alteração de configuração), envia mensagem ao subsistema de configuração para reconfigurar a *set-top box* (referência caso de uso 1.1).
5. Subsistema de falhas armazena problema encontrado e solução adotada em sua base de dados.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível no momento da identificação da falha / *Set-top box* armazena falha para depois enviá-la ao subsistema de falhas quando o canal de retorno estiver disponível.
- Subsistema de falhas não encontra solução para o problema / Administrador é avisado da falha encontrada sem solução.

A.3.2 Caso de Uso 3.2 – Serviço apresenta falha e comunica ao sistema de gerência assincronamente

Premissas:

As falhas são identificadas pelo próprio serviço, que está sendo executado em uma *set-top box*.

As falhas são enviadas ao sistema de gerência assincronamente.

Este caso pode ser extrapolado para a situação na qual o mesmo serviço apresenta a mesma falha em diversas *set-top boxes*. Neste caso pode ocorrer uma reconfiguração do serviço do lado gerente, para que novos downloads deste serviço não venham a ter o mesmo problema.

Atores:

Sistema de controle de serviços, administrador

Relação TMN x OSI:

Serviço x Falha

Curso normal dos eventos:

1. Serviço, que está rodando em algum terminal e possui um agente de gerenciamento e identificação de falhas própria, verifica que há falha em algum parâmetro.
2. Serviço informa, via *set-top box*, ao subsistema de falhas da gerência a falha ocorrida.
3. Subsistema de falhas procura em seu banco de dados uma solução para o problema.

4. Subsistema de falhas, com a solução encontrada (entende-se aqui alteração de configuração) envia mensagem ao subsistema de configuração para reconfigurar o serviço no lado agente (referência caso de uso 1.6).
5. Subsistema de falhas armazena problema encontrado e solução adotada em sua base de dados.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível no momento da identificação da falha / *Set-top box* armazena falha para depois enviá-la ao subsistema de falhas quando o canal de retorno estiver disponível.
- Subsistema de falhas não encontra solução para o problema / Administrador é avisado da falha encontrada sem solução.

A.4 Caso de uso macro 4 – realizar a gerência de desempenho

A.4.1 Caso de Uso 4.1 – Desempenho da *set-top box* é analisada e melhorada, caso necessário

Premissas:

Dados de desempenho de rede (TTL, perda de pacotes, etc) são tratados também neste caso de uso.

Atores:

Administrador

Relação TMN x OSI:

Elemento de Rede x Desempenho e Rede X Desempenho

Curso normal dos eventos:

1. *Set-top box* envia ao subsistema de desempenho da gerência um relatório contendo os dados medidos de desempenho da rede e da *set-top box* (Este envio pode ser periódico, a cada 2 horas, por exemplo).
2. Subsistema de desempenho armazena os dados enviados pela *set-top box* em sua base de dados.
3. Subsistema de desempenho verifica se os dados enviados estão dentro dos limites previamente estabelecidos para a boa operação da rede e da *set-top box*.
4. Caso estejam, caso de uso encerra-se.
5. Caso não estejam, subsistema de desempenho procura em seu banco de dados qual parâmetro da *set-top box* rede deve ser alterado para melhorar a desempenho do mesmo.
6. Subsistema de desempenho, com a solução encontrada (entende-se aqui alteração de configuração) envia mensagem ao subsistema de configuração para reconfigurar o terminal de acesso (referência caso de uso 1.1).
7. Subsistema de desempenho armazena problema encontrado e solução adotada em sua base de dados.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível no momento do envio das medições de desempenho / *Set-top box* armazena dados para depois enviá-la ao subsistema de desempenho quando o canal de retorno estiver disponível.

- Subsistema de Desempenho não encontra solução para o problema / Administrador é avisado de problema encontrado sem solução e nenhum comando é enviado a *set-top box*.

A.4.2 Caso de Uso 4.2 – *Set-top box* coleta dados para análise de desempenho

Premissas:

Dados de desempenho de rede (TTL, perda de pacotes, etc) são tratados também neste caso de uso.

Atores:

Não há (caso de uso interno ao sistema)

Relação TMN x OSI:

Elemento de Rede x Desempenho e Rede X Desempenho

Curso normal dos eventos:

1. *Set-top box*, em um intervalo de tempo determinado (por exemplo, de 1 em 1 hora), realiza testes e medições de desempenho.
2. Com os dados coletados, *set-top box* organiza os mesmos em um formato padrão e o armazena para envio posterior ao subsistema de desempenho.

Exceções / Tratamento das exceções:

Set-top box não consegue coletar todos os dados necessários / Dados são coletados no próximo intervalo de tempo.

A.4.3 Caso de Uso 4.3 – Desempenho do serviço é analisado e melhorado, caso necessário

Premissas:

Serviço sendo executado possui um módulo específico para medidas de desempenho

Este caso pode ser extrapolado para a situação na qual o mesmo serviço apresenta o mesmo problema de desempenho em diversas *set-top boxes*. Neste caso pode ocorrer uma reconfiguração do serviço do lado gerente, para que novos downloads deste serviço não venham a ter o mesmo problema.

Atores:

Sistema de controle de serviços, administrador (opcional)

Relação TMN x OSI:

Serviço x Desempenho

Curso normal dos eventos:

1. Serviço, que está rodando em alguma *set-top box* e possui um módulo de controle de desempenho, envia ao subsistema de desempenho, periodicamente, uma lista das medições efetuadas durante sua execução.
2. Subsistema de desempenho armazena os dados enviados pelo serviço em sua base de dados.
3. Subsistema de desempenho verifica se os dados enviados estão dentro dos limites previamente estabelecidos para a boa operação deste serviço específico.
4. Caso estejam, caso de uso encerra-se.

5. Caso não estejam, subsistema de desempenho procura em seu banco de dados qual parâmetro do serviço deve ser alterado para a melhoria de desempenho do mesmo.
6. Subsistema de desempenho, com a solução encontrada (entende-se aqui alteração de configuração) envia mensagem ao subsistema de configuração para reconfigurar o serviço no lado agente (referência caso de uso 1.6).
7. Subsistema de desempenho armazena problema encontrado e solução adotada em sua base de dados.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível no momento do envio das medições de desempenho / *Set-top box* armazena dados para depois enviá-la ao subsistema de desempenho quando o canal de retorno estiver disponível.
- Subsistema de Desempenho não encontra solução para o problema / Administrador é avisado de problema encontrado sem solução e nenhum comando é enviado.

A.5 Caso de uso macro 5 – realizar a gerência de segurança

A.5.1 Caso de Uso 5.1 – Sistema de gerência informa ao usuário quando foi realizado seu último acesso à rede

Premissas:

A autenticação propriamente dita na rede e na compra de serviços é realizada por um sistema externo de autenticação de usuários, sendo escopo do subsistema de segurança informar o tipo de autenticação utilizada pelo usuário.

Considera-se aqui que a autenticação é realizada na rede; muitas vezes uma autenticação local na *set-top box* pode estar disponível.

Atores:

Sistema de autenticação e autorização, usuário final

Relação TMN x OSI:

Rede x Segurança

Curso normal dos eventos:

1. Usuário se autentica na rede.
2. Sistema de autenticação e autorização envia dados de autenticação ao subsistema de contabilização (referência caso de uso 2.1).
3. Subsistema de contabilização informa subsistema de segurança da autenticação.
4. Subsistema de segurança envia à *set-top box* qual o horário e data da última autenticação do usuário (anterior a esta).
5. Interface com usuário mostra na tela a data da última autenticação do usuário.

Exceções / Tratamento das exceções:

Primeiro acesso do usuário / Mensagem de “boas-vindas” ao usuário é enviada à *set-top box*.

A.5.2 Caso de Uso 5.2 – Usuário se autentica com erro n vezes

Premissas:

A autenticação propriamente dita na rede e na compra de serviços é realizada por um sistema externo de autenticação de usuários, sendo escopo do subsistema de segurança informar o tipo de autenticação utilizada pelo usuário.

Considera-se aqui que a autenticação é realizada na rede; muitas vezes uma autenticação local na *set-top box* pode estar disponível.

O número *n* de vezes que o usuário se autenticou com erro pode ser alterado pelo administrador do sistema.

Atores:

Sistema de autenticação e autorização, usuário final

Relação TMN x OSI:

Rede x Segurança

Curso normal dos eventos:

1. Usuário tenta se autenticar na rede e erra a senha (ou outro modo de autenticação) por *n* vezes - (Subsistema de segurança é avisado via sistema de autenticação e autorização).
2. Subsistema de segurança bloqueia a *set-top box* via envio de comando à mesma.
3. *Set-top box* confirma a execução do comando.
4. Interface com usuário mostra na tela a mensagem de bloqueio do mesmo.
5. Base de dados do subsistema de segurança que contém todas as *set-top boxes* bloqueadas é atualizada.

Exceções / Tratamento das exceções:

- Elemento de rede não confirma o bloqueio do mesmo / Nada ocorre.

A.5.3 Caso de Uso 5.3 – *Set-top box* antes bloqueada é liberada

Premissas:

Não há.

Atores:

Usuário final, administrador

Relação TMN x OSI:

Rede x Segurança

Curso normal dos eventos:

1. Usuário pede a liberação da *set-top box* via administrador (call center).
2. Subsistema de segurança desbloqueia o elemento de rede via envio de comando.
3. Elemento de rede confirma a execução do comando.
4. Interface com usuário mostra na tela a mensagem de desbloqueio do mesmo.
5. Subsistema de configuração é atualizado.

Exceções / Tratamento das exceções:

- Elemento de rede não confirma o desbloqueio do mesmo / Quando o canal de retorno estiver disponível, mensagem é enviada ao subsistema de segurança.

A.5.4 Caso de Uso 5.4 – Usuário requisita a utilização de um serviço

Premissas:

A autenticação propriamente dita na rede e na compra de serviços é realizada por um sistema externo de autenticação de usuários, sendo escopo do subsistema de segurança informar o tipo de autenticação utilizada pelo usuário.

Considera-se aqui que a autenticação é realizada na rede; muitas vezes uma autenticação local na *set-top box* pode estar disponível.

Serviços são cobrados pelo número de utilizações, não pelo tempo de utilização (este modelo de serviços pode ser alterado; é adotado aqui por motivos de simplicidade, sem perda de generalidade).

Atores:

Sistema de autenticação e autorização, usuário final

Relação TMN x OSI:

Serviço x Segurança

Curso normal dos eventos:

1. Usuário, já autenticado na rede, requisita a utilização de um serviço.

2. Sistema de autenticação e autorização verifica a senha do usuário (ou outra forma de autenticação) e informa a requisição do usuário ao subsistema de segurança.
3. Subsistema de segurança verifica se perfil do usuário é compatível com o serviço especificado.
4. Subsistema de segurança envia mensagem positiva ao sistema de autenticação e autorização caso haja compatibilidade entre perfil de usuário e serviço solicitado.
5. Base de dados de segurança é atualizada com dados de compra de serviço pelo usuário.
6. Subsistema de segurança envia mensagem para envio de serviço ao usuário final ao sistema de controle de serviços.
7. Serviço é entregue ao cliente.

Exceções / Tratamento das exceções:

- Subsistema de segurança verifica que perfil de usuário não é compatível com requisição de serviço realizada / Subsistema de Segurança envia mensagem de não compatibilidade ao sistema de autenticação e autorização.

A.5.5 Caso de Uso 5.5 – Usuário altera seu perfil (plano de compra)

Premissas:

A autenticação propriamente dita na rede e na compra de serviços é realizada por um sistema externo de autenticação de usuários, sendo escopo do subsistema de segurança informar o tipo de autenticação utilizada pelo usuário.

Considera-se aqui que a autenticação é realizada na rede; muitas vezes uma autenticação local na *set-top box* pode estar disponível.

Atores:

Sistema de autenticação e autorização, usuário final, administrador (opcional)

Relação TMN x OSI:

Rede x Segurança

Curso normal dos eventos:

1. Usuário, a partir de interface de sua TV digital (ou administrador via interface de gerência), solicita a alteração de seu perfil (plano de compra).
2. *Set-top box* envia solicitação ao subsistema de segurança da gerência.
3. Subsistema de segurança faz as verificações necessárias e armazena a mudança solicitada.
4. Subsistema de segurança informa a alteração do plano ao Sistema de autenticação e autorização (externo).
5. Subsistema de segurança envia mensagem à *set-top box* confirmando a operação.
6. *Set-top box* envia mensagem à tela de navegação do usuário confirmando a operação.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível quando do envio da solicitação de alteração ao subsistema de segurança / Operação é cancelada

A.5.6 Caso de Uso 5.6 – Usuário altera seu mecanismo de autenticação

Premissas:

A autenticação propriamente dita na rede e na compra de serviços é realizada por um sistema externo de autenticação de usuários, sendo escopo do subsistema de segurança informar o tipo de autenticação utilizada pelo usuário.

Atores:

Sistema de autenticação e autorização, usuário final, administrador (opcional)

Relação TMN x OSI:

Rede x Segurança

Curso normal dos eventos:

1. Usuário, a partir de interface de sua TV digital (ou administrador via interface de gerência), solicita a alteração de seu mecanismo de autenticação.
2. *Set-top box* envia solicitação ao subsistema de segurança da gerência.
3. Subsistema de segurança faz as verificações necessárias e armazena a mudança solicitada.
4. Subsistema de segurança informa a alteração ao Sistema de autenticação e autorização (externo).
5. Subsistema de segurança envia mensagem à *set-top box* confirmando a operação.
6. *Set-top box* envia mensagem à tela de navegação do usuário confirmando a operação.

Exceções / Tratamento das exceções:

- Canal de retorno não disponível quando do envio da solicitação de alteração ao subsistema de segurança / Operação é cancelada.

A.5.7 Caso de Uso 5.7 (opcional) – Administrador acessa sistema de gerência a partir da Internet

Premissas:

Interface web está disponível no sistema de gerência aqui apresentado.

Atores:

Administrador, sistema de autenticação e autorização

Relação TMN x OSI:

N/A

Curso normal dos eventos:

8. Administrador, a partir da Internet, se autentica no sistema via sistema de autenticação e autorização
9. Subsistema de segurança apresenta sistema de gerência com as mesmas funções do sistema que opera em modo nativo.

Exceções / Tratamento das exceções:

- Não há