

Richard Augusto Schafer De Martini

Um Modelo de Gerenciamento de Rede PACS
para o Sistema Catarinense de Telemedicina

Florianópolis, Santa Catarina

Out. 2008

Richard Augusto Schafer De Martini

Um Modelo de Gerenciamento de Rede PACS para o Sistema Catarinense de Telemedicina

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação

Orientador

Aldo Von Wangenheim

Universidade Federal de Santa Catarina

Florianópolis, Santa Catarina

Out. 2008

Trabalho de Conclusão de Curso sob o título “Um Modelo de Gerenciamento de Rede PACS para o Sistema Catarinense de Telemedicina”, apresentado por Richard Augusto Schafer De Martini sob aprovação, em Florianópolis, Santa Catarina, pela banca examinadora constituída:

Prof. Dr. rer.nat. Aldo von Wangenheim
Departamento de Informática e Estatística - INE
Orientador

Rafael Andrade
Departamento de Informática e Estatística – INE

Harley Miguel Wagner
Departamento de Informática e Estatística - INE

Rafael Simon Maia
Departamento de Informática e Estatística - INE

Douglas Dyllon Jeronimo de Macedo
Departamento de Informática e Estatística - INE

À minha família, em especial meus pais.

Agradecimentos

Ao Grupo Cyclops pela oportunidade de aprendizado, trabalho e pelos bons momentos que aqui passei. Aos colegas de Cyclops, com os quais tive grande aprendizado. Em especial a “*família DCMServer*” e ao Thiago Coelho, sem os quais nada disso seria possível. Agradeço também ao orientador Aldo Von Wangenheim e aos membros da banca, Rafael Andrade, Harley Wagner, Rafael Simon e Diego Dias pelo apoio, críticas e sugestões.

A minha namorada Kalynka pelo apoio, compreensão, carinho e paciência durante a realização desse trabalho. Finalmente, agradeço a minha família pela ajuda, pela educação, oportunidade e confiança que me foram dadas.

Resumo

A evolução dos ambientes de rede e dos servidores, bem como a utilização e difusão do padrão DICOM permitiram a evolução e a popularização de ambientes PACS. A Rede Catarinense de Telemedicina amplia a utilização dessa tecnologia em hospitais públicos interconectando-os e criando uma base integrada de pacientes e exames por meio do Portal de Telemedicina. Essa arquitetura constitui-se como um ambiente PACS geograficamente distribuído, a qual é caracterizada pela utilização de um servidor de ponte entre os equipamentos que adquirem as imagens e o Portal de Telemedicina.

No entanto, a configuração desta arquitetura traz problemas de gerenciamento e administração de cada um destes servidores. Operações como limpeza de disco, monitoramento de status de sistema e de serviços são usuais e possuem um custo elevado. Em função disso, é necessária uma solução que incorpore tais necessidades e permita a descentralização destes processos.

A ampliação da RCTM e o desenvolvimento de novas ferramentas de interoperabilidade entre sistemas exigem uma solução capaz de integrá-las. Para isso, é importante a criação de um conjunto de regras e mecanismos que constituam um meio de associatividade entre os aplicativos e o serviço de gerenciamento. Este conjunto de regras e mecanismos constitui um modelo, que pode ser estendido e utilizado para a administração de quaisquer serviços geograficamente distribuídos.

Palavras Chaves: Telemedicina, PACS, administração, gerenciamento, serviços.

Abstract

The network environment and servers evolution as well the DICOM pattern dissemination enables the development and popularization of PACS environments. The Santa Catarina State Telemedicina Network expands the use of this technology in public hospitals interconnecting them and creating a integrate patient and examination database through the Telemedicine Portal. This architecture represents and geographically distributed PACS environment, which is has like a characteristic the use of as server bridge connecting equipments and de Telemedicine Portal.

However, this configuration has problems like the administration and management of these bridge servers. Cleaning disk operations, system and services status tracking are usual and take a lot of time. Examining these facts, a solution is necessary to cover these necessities and enables the process decentralization.

Considering the extension of the Telemedicine Network and the development of new features to the system, the management solution needs to be sufficient abstract to integrate these news applications. As way to permit this process, a set of rules and mechanism establishment be like a integration way between services and the management system. This set of rules and mechanism seems like a model, what can be extended and used in others geographically distributed systems.

Key Words: Telemedicine, PACS, service, management, administration.

Sumário

<i>Agradecimentos</i>	<i>v</i>
<i>Resumo</i>	<i>vi</i>
<i>Abstract</i>	<i>vii</i>
<i>Sumário</i>	<i>viii</i>
<i>Lista de Figuras</i>	<i>xi</i>
<i>Lista de Abreviaturas e Acrônimos</i>	<i>xiii</i>
1. Introdução	15
1.1. Definição do Problema.....	15
1.2. Justificativa.....	18
1.3. Objetivos.....	19
1.3.1. Objetivo Geral	19
1.3.2. Objetivos Específicos.....	20
1.4. Cenário de Aplicação.....	20
1.5. Estrutura do trabalho.....	22
2. Fundamentação Teórica	23
2.1. Sistemas Similares	23
2.2. Telemedicina	25
2.3. A Rede Catarinense de Telemedicina	26
2.4. DICOM.....	27

2.5.	PACS.....	27
2.6.	PACS Geograficamente Distribuído.....	28
3.	Metodologia	31
3.1.	Análise do cenário	32
3.2.	A Proposta.....	33
3.3.	O modelo proposto.....	35
3.3.1.	Organização do Sistema	35
3.3.2.	O Servidor	37
3.3.3.	Cliente.....	38
3.3.3.1.	Sistema de autogerenciamento	40
3.3.3.2.	O Processo de Manutenção do Sistema.....	41
3.3.3.3.	Cópias de Segurança	41
3.3.4.	Visão do sistema.....	42
3.3.5.	Comunicação	43
3.3.5.1.	O Processo de Transmissão de Dados	45
3.3.5.2.	Definição de mensagens	48
3.3.6.	Segurança	50
4.	Cyclops Healthcare Integration System	52
4.1.	Características importantes.....	52
4.2.	Arquitetura do Sistema	53
4.3.	Integração com o Portal de Telemedicina e níveis de acesso	55
4.4.	Integração com o Cyclops DCMServer.....	58
4.4.1.	Comunicação e processamento de mensagens	59
4.4.2.	Alterações nas camadas de aplicativo existente.....	59
4.4.3.	Alterações nas configurações	61
4.5.	Integração com os Outros Aplicativos.....	62
5.	Conclusão.....	64
5.1.	Trabalhos Futuros	65
6.	Referências Bibliográficas.....	66

7.	<i>Apêndice A – Requisitos do sistema</i>	69
8.	<i>Anexos</i>	79
8.1.	Unix Signals	79
8.2.	Arquivo de configuração do cycClient	81

Lista de Figuras

Figura 1 – Rede PACS Atual.....	16
Figura 2 - Rede PACS Prevista.....	16
Figura 3 – Visão Esquemática do Sistema Atual.....	17
Figura 4 – Abstração do Sistema.....	21
Figura 5 – PACS geograficamente distribuído na RCTM.....	29
Figura 6 – Abstração do problema ao modelo computacional Cliente-Servidor.....	36
Figura 7 – Estrutura de uma mensagem WebService.....	44
Figura 8 - Processo de Associação entre cliente e servidor.....	46
Figura 9 – O processo de Transmissão de dados.....	47
Figura 10 – Estrutura das Mensagens de Associação.....	49
Figura 11 – Estrutura das Mensagens de Transmissão.....	50
Figura 12 – Arquitetura do Sistema.....	54
Figura 13 – Visão do Portal de Telemedicina.....	55
Figura 14 – Níveis de Acesso.....	56
Figura 15 – Diagrama de camadas do Cyclops DCMServer.....	60
Figura 16 – Requisitos Funcionais básicos.....	69
Figura 17 – Requisitos Funcionais de configuração de serviços.....	70
Figura 18 – Requisitos Funcionais de manutenção.....	70
Figura 19 – Requisitos Funcionais de visualização de status de exames.....	71
Figura 20 – Requisitos Funcionais de atualizações automáticas.....	71
Figura 21 – Requisito Funcional de Comunicação entre aplicações.....	71
Figura 22 – Requisitos Funcionais de cópias de segurança.....	72
Figura 23 – Requisitos Funcionais de Comunicação.....	72
Figura 24 – Requisitos Módulo de Regulação de Exames.....	73

Lista de Abreviaturas e Acrônimos

AETitle	Application Entity Title
CHIS	Cyclops Healthcare Integration System
CT	Computed Tomography
CycClient	Cyclops Dicom/HL7 Client
DCMServer	Cyclops Dicom Server
DICOM	Digital Imaging Communications in Medicine
ECG	Eletrocardiograma
HL7	Health Level 7
HL7Server	Cyclops HL7 Server
JPEG	Joint Photographic Experts Group
MR	Magnetic Resonance
OSCAR	Open source cluster application resources
php	Hypertext Preprocessor
RCTM	Rede Catarinense de Telemedicina
REMS	Remote Exam Management System
SOAP	Simple Object Access Protocol
ssh	Secure Shell
SSL	Secure Sockets Layer
SC	Secondary Capture
SES/SC	Secretária de Estado da Saúde de Santa Catarina

TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VNC	Virtual Network Computing
XA	X-Ray Angiography

1. Introdução

Os avanços nas telecomunicações têm proporcionado a quebra de barreiras geográficas. Atualmente é possível comunicar-se e interagir com pessoas em outras cidades, países e continentes. Esse avanço possibilita o desenvolvimento de diversas áreas de conhecimento, como a medicina. Hoje já é possível realizar cirurgias e acessar exames e informações de um paciente remotamente. Esses avanços auxiliaram o processo de crescimento e popularização da telemedicina. Entende-se por telemedicina, a utilização de qualquer meio que auxilie no diagnóstico ou tratamento de um paciente à distância (Bashshur 2002).

Pensando nisso, a Universidade Federal de Santa Catarina, (UFSC) em conjunto com a Secretaria de Estado da Saúde de Santa Catarina desenvolvem a Rede Catarinense de Telemedicina. Atualmente essa rede possui mais de 70 municípios integrados e, até julho de 2008, já permitiu a inclusão de mais de 120 mil exames de diversas modalidades realizados pelo estado (Wallauer, et al. March-April 2008.). Entre essas modalidades pode-se destacar: Eletrocardiograma, Tomografia Computadorizada, Cintilografia e Hemodinâmica.

O Portal de Telemedicina é o ponto central de todo o sistema. A partir dele é possível acessar todos os exames armazenados, executar tarefas administrativas, como a emissão de laudos, e manipular as imagens associadas. O processo de administração, manipulação, visualização de imagens médicas em um ambiente hospitalar configura um ambiente PACS (Picture Archiving and Communication System) (Cao, Huang e Zhou 2003).

1.1. Definição do Problema

Os exames armazenados no banco da RCTM podem ser divididos em duas classes – DICOM e não DICOM. A primeira, não pode ser enviada diretamente ao servidor do Portal de Telemedicina por questões como o grande fluxo de dados e sua segurança. Como forma de resolver essa questão, em cada instituição integrada a RCTM e que realize esses procedimentos é instala-

do um servidor conhecido como *bridge* que serve como ponte entre o aparelho e o Portal de Telemédicina. As Figuras 1 e 2 apresentam o cenário atual da rede PACS e o previsto, respectivamente.

Atual



Figura 1 – Rede PACS Atual

A Figura 1 apresenta os pontos de comunicação DICOM atualmente implantados. A Figura 2 mostra também os pontos já previstos para implantação de tomografia computadorizada, ressonância magnética, hemodinâmica e ultra-som em todo o estado. Cada ponto representa uma instituição, a qual receberá um servidor que atuará como ponte para o Portal de Telemédicina.

Previsto

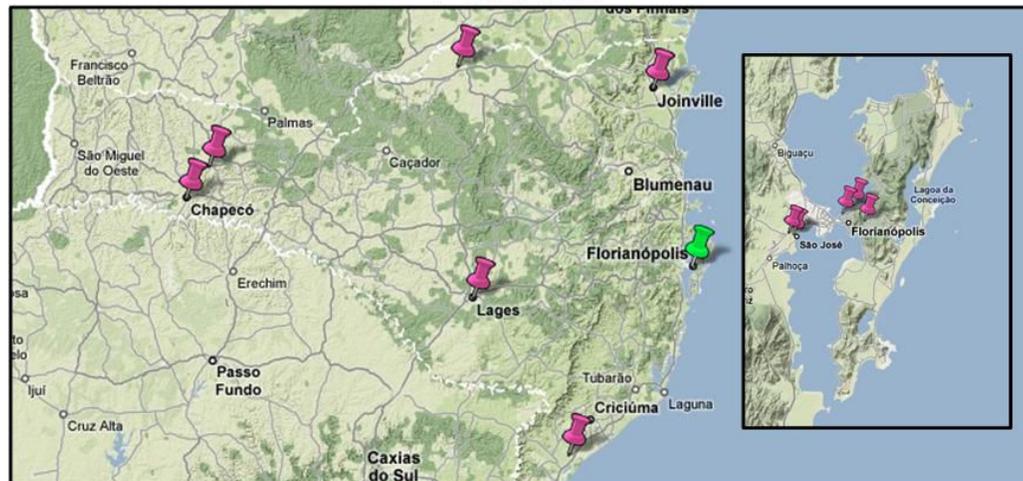


Figura 2 - Rede PACS Prevista

Cada uma dessas *bridges* é responsável por receber, armazenar e encaminhar para o servidor DICOM as imagens adquiridas pelos equipamentos que realizam exames. Isso permite que os equipamentos não fiquem conectados diretamente a Internet, proporciona segurança ao equipamento e permite a utilização de mecanismos de proteção, como cifragem de dados. Outra vantagem da utilização dessa arquitetura é tornar os equipamentos independentes da rede. Assim, a execução de procedimento médico e seu envio ao servidor de ponte mesmo que exista algum problema com a conexão da *bridge* a Internet, funcionará corretamente.

No entanto, é necessária a manutenção e a administração permanente dessas *bridges*. Entre as ações normalmente executadas, pode citar a exclusão de exames já encaminhados para liberação de espaço em disco, a inclusão/alteração de AETitles cadastrados e a visualização de *status* e registros do sistema. Esses processos são custosos e centralizam as tarefas em poucos administradores. Outra desvantagem da utilização dessas *bridges* é a dessincronização dos dados. A inexistência de um mecanismo que realize esse processo aumenta a utilização de disco com informações replicadas, utilização da rede e impossibilita acesso do usuário aos dados armazenados na *bridge*.

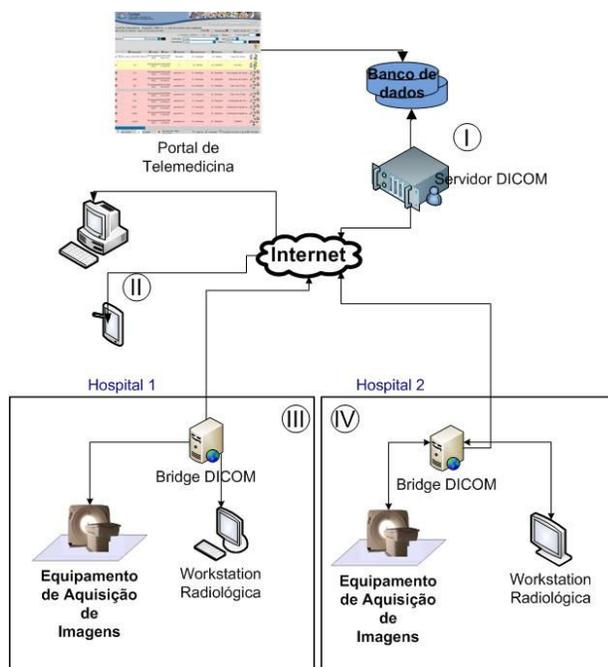


Figura 3 – Visão Esquemática do Sistema Atual.

A Figura 3 exibe a arquitetura atual de todo o sistema. As instituições integradas à rede são representadas por III e IV. Em cada uma existe um servidor que serve como ponte entre a aquisição do exame e seu armazenamento no sistema. Essa *bridge* também é o delimitador entre o ambiente interno da instituição e a Internet.

Após passar por esse primeiro processo, as imagens referentes a um exame são enviadas pela Internet ao Servidor DICOM, representado por I. Esse realiza o armazenamento das informações e envia uma cópia dos dados para o Portal de Telemedicina. Com a utilização desse sistema o usuário tem acesso as informações e as imagens de cada exame. Outros meios de acesso, como visualizadores DICOM ou *workstations* portáteis são representadas por II.

1.2. Justificativa

Os avanços em telecomunicações, na informática e nos sistemas computacionais têm permitido a integração de aplicativos e o desenvolvimento de soluções distribuídas. Esse processo trouxe consigo um paradoxo, a necessidade de outros sistemas para realizar o gerenciamento remoto dos serviços e dos próprios servidores. No entanto, esse paradoxo não é exclusivo da área computacional, podem-se encontrar semelhanças entre essa adversidade e problemas em muitas outras áreas, como na empresarial - administração de matrizes e filiais - e no governo com o gerenciamento dos recursos entre estados, municípios e a união.

Analogamente na área empresarial, uma indústria qualquer possui uma matriz e suas diversas filiais geograficamente distribuídas. Cada filial pode ter autonomia e liberdade para tomar certas decisões, como a contratação e demissão de funcionários. No entanto, precisa enviar relatórios periodicamente e manter a matriz informada das decisões tomadas. Já a empresa mãe, normalmente envia novas diretrizes às filiais, assim como pode exigir a tomada de decisões e determinar ações a serem realizadas pela empresa filho.

O processo de gerenciamento remoto em sistemas computacionais assemelha-se ao processo encontrado no mundo empresarial. Geralmente é encontrado um servidor central que realiza o papel da empresa matriz e diversos clientes, que fazem o papel das filiais. Esse processo é reali-

zado, freqüentemente, por meio de camadas extras a aplicação ou de uma ou mais ferramentas que realizem o processo ou parte dele.

Diversas soluções computacionais já foram desenvolvidas para sistemas assim. Mais especificamente, pode-se citar desde o SSH (*Secure Shell*) para sistemas Unix, até soluções como extensões ao padrão DICOM (Luz Jr., et al. 2006) para gerenciamento de servidores DICOM remotos. Cada solução procura resolver um conjunto de problemas específicos, sendo que utilizando diversas ferramentas é possível realizar manutenção e gerenciamento total do sistema.

Porém, cada solução tem suas peculiaridades e restrições. Como exemplo, a utilização de *ssh* permite acesso a todas as configurações do servidor, mas exige a criação ou compartilhamento de uma conta de usuário diretamente no sistema acessado. Já a extensão DICOM, apenas permite a visualização de informações do seu sistema, impossibilitando a execução de chamadas e a execução de serviços.

A utilização de vários serviços polui a arquitetura e torna árduo o processo de aprendizado e manutenção de um sistema. O estágio atual da RCTM e os novos pontos já previstos para integração exigem a utilização de mecanismos e técnicas claras que facilitem o gerenciamento e, ao mesmo tempo, descentralize esse processo. Por isso, uma nova ferramenta integrada a sistemas já existentes é necessária para a realização das tarefas administrativas. Esse aplicativo precisa ser robusto o suficiente para realizar o gerenciamento e manutenções do sistema, mas flexível o suficiente para que cada aplicativo não perca suas características e permita a integração de novos serviços.

1.3. Objetivos

1.3.1. Objetivo Geral

Desenvolver uma metodologia para gerenciamento de rede, serviços e servidores PACS geograficamente distribuídos na Rede Catarinense de Telemedicina.

1.3.2. *Objetivos Específicos*

- Elaborar um conjunto de regras para gerenciamento de servidores e serviços geograficamente distribuídos;
- Estabelecer uma metodologia que permita a integração de serviços gerenciáveis ao modelo;
- Propor mecanismos de integração entre o Portal de Telemedicina, seus perfis de acesso e serviços gerenciados pelo sistema;
- Desenvolver mecanismos de integração entre o Cyclops DCMServer, o Cyclops HL7 Server e a ferramenta proposta.

1.4. **Cenário de Aplicação**

O processo de gerenciamento remoto de sistemas distantes pode ser facilitado com a utilização de ferramentas automatizadas de execução de tarefas e a partir de mecanismos que permitam a chamada remota de procedimentos por usuários do sistema. Esses são os dois principais pontos a serem abordados e permitem, respectivamente, a redução do custo gasto por tarefa e aumento da produtividade do sistema. Para ampliar os ganhos de desempenho no segundo caso, entretanto, é importante que as chamadas possam ser executadas remotamente por meio de uma visão simplificada por um conjunto de usuários cadastrados.

Pensando nisso, o desenvolvimento de um ambiente que permita executar operações e enviar os comandos para a *bridge* é fundamental para facilitar o processo encontrado na RCTM. Nesse ambiente, o usuário do sistema deve informar quais ações devem ser tomadas em quais servidores. Após a confirmação da operação que deve ser realizada, o sistema envia uma mensagem ao sistema afetado, informando-o qual ação e quais os parâmetros necessários para sua execução. Depois de efetivado o comando, o cliente (*bridge*) responde o status da ação, que posteriormente é exibida ao usuário pelo sistema.

O controle de acesso e os registros do sistema precisam ser mantidos para segurança de cada *bridge*. A utilização de níveis de acesso e o armazenamento de registros sobre as operações executadas são suficientes para isto. A descentralização das operações possibilita que algumas operações sejam executadas diretamente por usuários do sistema e permite que alguns erros sejam contornados de forma rápida.

A Figura 4 mostra uma abstração do sistema, de onde é possível invocar a execução remota de ações em cada servidor. Essas operações evitam acessos em cada servidor e reduzem o tempo gasto com esse processo.

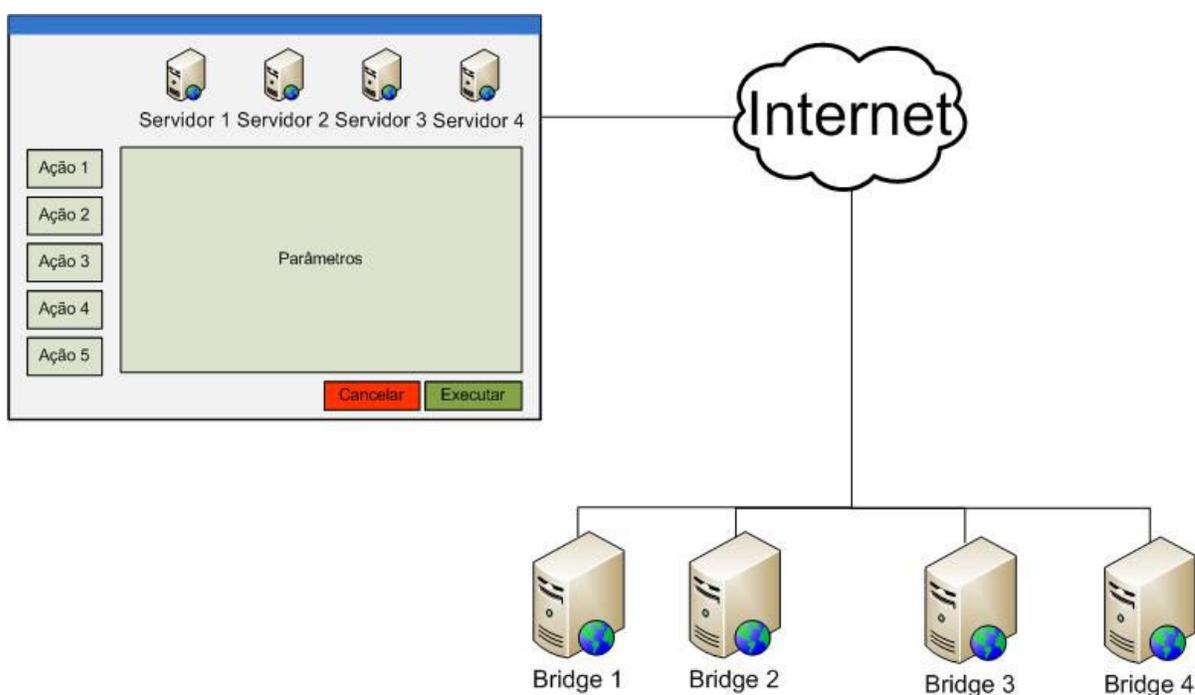


Figura 4 – Abstração do Sistema

As operações realizadas em cada servidor podem ser divididas em dois grupos: programadas e não programadas. As primeiras referem-se a operações que periodicamente são executadas para manter o sistema em funcionamento. Essa característica as torna críticas e, por serem periódicas, permite que sejam executadas automaticamente. Pensando nisso, é importante que cada *bridge* realize essas operações pré-programadas como as de limpeza de disco e manutenção de banco de dados sem a necessidade de uma solicitação.

1.5. Estrutura do trabalho

Neste primeiro capítulo foram descritos o problema de gerenciamento remoto de servidores e serviços, seu contexto e seu cenário. Além desses dados, também foi exposta a necessidade de uma ferramenta para resolvê-lo, assim as vantagens e desvantagens da utilização de outras soluções existentes. Já o segundo capítulo apresenta maiores detalhes sobre o funcionamento da Rede Catarinense de Telemedicina e os conceitos básicos de algumas partes envolvidas.

A abstração do problema, assim como um conjunto de regras e uma metodologia genérica é apresentada no capítulo 3. Essa parte contém, mais especificamente, um modelo contendo uma solução para o gerenciamento remoto de serviços. O capítulo 4 apresenta uma aplicação da metodologia proposta no contexto da RCTM. Nesse capítulo estão descritos os mecanismos utilizados na integração da ferramenta proposta com os serviços já existentes, assim como especifica algumas alterações necessárias em cada *software*. Já no capítulo 5 são descritos as conclusões retiradas deste trabalho e expostos os resultados esperados com o desenvolvimento de uma ferramenta baseada no modelo proposto. Além disso, propostas de extensão desse modelo ou especificações estão descritas na parte relacionada a Trabalhos Futuros.

2. *Fundamentação Teórica*

Neste capítulo será descrito o estado da arte em sistemas de administração e será contextualizado o problema. Primeiramente o problema de gerenciamento remoto será estudado e uma breve descrição será realizada. Após isso, serão explicados os conceitos básicos sobre a Rede Catarinense de Telemedicina (RCTM) e suas características. Finalmente, será contextualizado o gerenciamento remoto de servidores junto a RCTM.

2.1. **Sistemas Similares**

O problema de gerenciamento remoto de serviços é típico de sistemas distribuídos. A partir dos avanços dos servidores e da rede foram sendo desenvolvidas soluções em diversos níveis para o problema. A primeira solução foi o DEC VT100, que não mais era que um “terminal burro” (teclado e monitor) que acessava remotamente os servidores por meio de um modem e permitia o envio de comandos ao sistema. Esse sistema foi inovador para a época e pode ser considerado um precursor para os meios modernos de acesso remoto.

Posteriormente, com os avanços na área de redes e nos sistemas operacionais, outras soluções foram surgindo, como o *ssh*(*Secure Shell*) e o *VNC*(*Virtual Network Computing*) – duas das soluções mais conhecidas atualmente. Ambas operam de modo parecido e permitem o envio de comandos e parâmetros a serviços do sistema remotamente. Atualmente esses aplicativos são considerados referências em administração e acesso remoto de servidores e garantem a execução e instalação de qualquer serviço na máquina, desde que o usuário possua permissão para tal.

Soluções como o *ssh* e o *vnc* permitem acesso remoto ao servidor e execução de comandos com seus parâmetros. No entanto, exigem um usuário e senha na máquina para cada usuário que necessite executar alguma operação. Essa exigência abre uma brecha na segurança da máquina, o que inviabiliza a utilização de soluções como essa. Outro ponto desfavorável na utilização

dessas ferramentas é a utilização do sistema operacional Unix na arquitetura do sistema atual, o qual ainda não é muito conhecido por usuários fora da área da informática.

Na área de gerenciamento de serviços existem diversos aplicativos que realizam tarefas semelhantes em sistemas específicos. Na área de *clusters*, a ferramenta Torque auxilia no processo de gerenciamento de cada nodo. Essa ferramenta, desenvolvida por um conjunto de empresas e universidades, permite gerenciamento de *Unix process* com ênfase em escalabilidade e tolerância a falhas. Outras ferramentas, como o OSCAR (*Open Source Cluster Application Resources*) também são destinadas a essa área e auxiliam no processo de gerenciamento de clusters.

Na área de PACS há diversos gerenciadores em nível de um servidor, como o *Conquest PACS*. Aplicativos como esse, geralmente possuem uma interface gráfica de onde é possível alterar configurações e executar algumas operações básicas, como inserção de clientes e remoção de exames antigos. No entanto, pensando-se em sistemas geograficamente distribuídos com múltiplos servidores e clientes não existe nenhuma solução aberta disponível.

Em função disso, é necessária a utilização de outras ferramentas, como o *ssh*. No entanto, esse também possui suas desvantagens, por isso uma outra solução precisa ser desenvolvida. Para o desenvolvimento de um aplicativo como esse, a utilização de algumas ferramentas e bibliotecas o torna mais robusto e evita gastos com o desenvolvimento de sub-partes já existentes. Áreas como comunicação e o armazenamento em banco de dados possuem *frameworks* estáveis desenvolvidos e podem ser facilmente re-utilizados.

Em função da comunicação entre *bridges* e Telemedicina, é necessário o desenvolvimento de um meio de comunicação entre as pontas. Essa tarefa pode ser realizada com a utilização da biblioteca *Berkeley Sockets* (Desenvolvida pela Universidade de Berkeley a partir de 1983) e de algumas partes do padrão *WebService*. A biblioteca de comunicação é referência para desenvolvimento de aplicativos que necessitem de comunicação entre aplicativos por meio de redes de computadores.

Já o padrão *WebService* foi desenvolvido como forma de padronizar as comunicações entre diferentes aplicativos. Esse padrão permite a interoperabilidade entre sistemas diversos e as mensagens trocadas seguem o padrão XML. No contexto desse projeto, é interessante utilizar o padrão como forma de facilitar o processo de integração de novas mensagens e transparecer todo o processo (Alonso, et al. 2003).

O armazenamento das configurações e dos históricos de transações pode ser realizado com um banco de dados relacional. Entre as diversas opções grátis disponíveis, o *SQLite* é uma que se encaixa bem no problema. Sua facilidade de instalação e baixa utilização de recursos do sistema são duas vantagens que a tornam vantajosa.

2.2. Telemedicina

O termo telemedicina refere-se a qualquer meio que auxilie o diagnóstico ou tratamento médicos realizados a distância (Bashshur 2002), mais especificamente será tratada por meio do uso de sistemas computacionais. Como exemplo, pode-se citar uma prescrição realizada por ligação telefônica, troca de emails entre médico e paciente, e, mais recentemente, sistemas web de acesso a resultados de exame. Os avanços em telecomunicações e na informática têm permitido um rápido desenvolvimento desta área. Esse processo tem sido bastante heterogêneo, sendo que algumas áreas como a telerradiologia e a telepatologia se encontram mais avançadas e mais utilizadas que outras áreas como a cirurgia a distância, que já possui seus primeiros produtos e prontuário eletrônico de paciente.

Atualmente a telemedicina vem sendo desenvolvida como um mecanismo de melhorar o atendimento médico em diversas regiões como, por exemplo, na Inglaterra (Tackley 2004), em uma experiência no Equador (Mijares 2004), ou mesmo em Santa Catarina (Wallauer, et al. March-April 2008.). Por meio dessa tecnologia é possível melhorar o atendimento médico em áreas com deficiência de profissionais capacitados e equipamentos, reduzindo custos com transportes de pacientes e melhorando o atendimento em grandes centros (Maia, Wangenheim e Nobre 2006).

2.3. A Rede Catarinense de Telemedicina

O Portal de Telemedicina é uma ferramenta desenvolvida pelo Grupo Cyclops em parceria com a Secretaria de Estado da Saúde de Santa Catarina (SES). Ele se caracteriza por ser um sistema *web* que armazena e permite visualização de exames de imagens realizados em diversos equipamentos distribuídos pelo estado. Atualmente ele armazena exames de Eletrocardiograma, Tomografia Computadorizada, Ultra-som, Nutrição Parenteral e Medicina Nuclear realizados em mais de 75 instituições em aproximadamente 70 cidades (Wallauer, et al. March-April 2008.).

Essa ferramenta é o ponto centralizador da RCTM, uma rede assíncrona de telemedicina assistencial. Essa rede tem permitido a melhoria do atendimento médico em localidades carentes de infra-estrutura ou equipe capacitada, assim como também auxilia na redução de filas em hospitais e centros especializados. Em funcionamento desde 2005, tem reduzido custos com transportes de pacientes e tem proporcionado um aumento no número de exames realizados no estado.

O envio dos exames ao aplicativo é de responsabilidade de um conjunto de ferramentas que recebem ou adquirem as imagens diretamente dos equipamentos que o realizaram. Exames exportados no formato DICOM, como Tomografias Computadorizadas, Ressonâncias Magnéticas e Hemodinâmicas, são de responsabilidade do Cyclops Dicom Server, o qual armazena e copia os exames para o Portal de Telemedicina.

A RCTM também integra outras soluções já desenvolvidas como o *HL7 Server* – servidor que permite a interoperabilidade entre sistemas médicos e comunicação de equipamento com essa característica de exportação – e o Dicomizer, o qual *DICOMiza* (transformação de imagens comuns em arquivos no formato DICOM) os exames realizados por equipamentos que não possuem suporte a esse padrão. Todas essas soluções permitem a integração de equipamentos independentemente de fabricante e garantem uma solução simples, reduzindo custos com deslocamento de pacientes e armazenamento em papel dos resultados.

2.4. DICOM

O padrão DICOM (*Digital Imaging Communications in Medicine*) foi desenvolvido em conjunto pela comunidade científica e os fabricantes de aparelhos de imagem como forma de padronizar os arquivos digitais que representam as imagens de exames deste tipo (Dellani 2001). Inicialmente, cada fabricante desenvolvia sua solução para seu equipamento. Isso impossibilitava a integração de diversos equipamentos e obrigava clínicas e hospitais adquirirem equipamentos de apenas um fabricante ou diversas *workstations* radiológicas específicas. O alto custo dessas soluções e a inércia inicial do projeto estagnaram o desenvolvimento da área, que ficou restrita a soluções e *workstations* proprietárias.

Como forma de solucionar esses problemas, o DICOM surgiu em 1993. O padrão foi desenvolvido praticamente do zero e substituiu as tentativas de padronização anteriores, como o padrão NEMA. O novo padrão permite a integração de diversos aparelhos de vários fabricantes e possibilita o compartilhamento de soluções para equipamentos diversos. Isso permitiu uma rápida popularização da nova solução que logo foi adotada por praticamente todos os fabricantes. Este fato, aliado aos avanços nos ambientes de rede, impulsionaram o desenvolvimento de ambientes PACS (*Picture Archiving and Communication System*).

2.5. PACS

O conceito PACS abrange ambientes onde há comunicação, visualização, manipulação e armazenamento de imagens médicas em ambientes médico-hospitalares (Cao, Huang e Zhou 2003). Esse conceito permite encaixar diversos ambientes clínico-hospitalares. Considerando-se a RCTM como referência, caracteriza-se como PACS ambientes que possuem pelo menos as seguintes características:

- **Aquisição de Imagens.** Equipamentos realizam a aquisição das imagens utilizando técnicas e exames de imagem. Normalmente os equipamentos possuem suporte ao padrão DICOM. Para os aparelhos que não exportam seus exames nesse padrão, existem ferramentas que realizam a “*DICOMização*” das imagens, como o *Cyclops Dicomizer* e posteriormente realizam a transmissão desses exames.
- **Comunicação.** Após a realização de um exame, as imagens, ou um subconjunto delas, são transmitidas em formato DICOM, HL7 ou através de outro formato/protocolo para um servidor de imagens correspondente.
- **Armazenamento.** As informações dos exames recebidos pelos servidores são armazenadas em um banco de dados ou em uma estrutura de dados específica.
- **Visualização.** Por meio de visualizadores DICOM, estações de trabalho ou de sistemas web, como o Portal de Telemedicina, os médicos responsáveis e o paciente têm acesso a esses dados.
- **Aquisição de Dados.** Através de algum mecanismo informações sobre o pacientes são obtidas e armazenadas, em conjunto ou não com o banco de imagens.

Ambientes como esse permitem o armazenamento de um histórico médico-hospitalar de pacientes. Dados como esse garantem maior controle sobre enfermidades, histórico clínico e facilitam o processo de diagnósticos. Em nível institucional, ambientes como esse permitem a diminuição de custos em longo prazo por meio da redução de gastos com filmes radiológicos, facilitação do processo de recuperação de exames e facilidade e qualidade no seu armazenamento. (Becker e Arenson 1994)

2.6. PACS Geograficamente Distribuído

A evolução da Internet e a disponibilização de banda larga a custo baixo permitiram a integração de sistemas geograficamente distantes. Um PACS geograficamente distribuído consiste na integração de um ou mais ambientes PACS distantes fisicamente através de um ponto centrali-

zador. Esse ambiente também é caracterizado pelo compartilhamento de algumas informações assim como pela individualidade de outros dados.

O processo de compartilhamento de dados normalmente é realizado, comumente por mecanismos como servidores DICOM distribuídos, *webservices*, o padrão HL7, CORBA (Ribeiro, et al. 2002) e outros *middlewares*. Normalmente dados como prontuário de paciente e acesso as informações são compartilhados, enquanto serviços, configurações e exames pendentes são mantidos distribuídos em cada ambiente PACS.

Na RCTM, o Portal de Telemedicina é o ponto central dos ambientes PACS. Cada instituição que realize exames de alta complexidade possui um ambiente PACS interno, no qual existe o armazenamento temporário das imagens, o que permite um acesso rápido aos exames utilizando própria rede interna da instituição. Cada ambiente possui uma *bridge*, a qual é o ponto central dos seus exames e a ponte entre os equipamentos que realizam os exames e o Portal de Telemedicina. Após a realização de um procedimento e a emissão de seu laudo, essas informações ficam disponíveis e acessíveis por um repositório *web* de exames e dados de paciente.

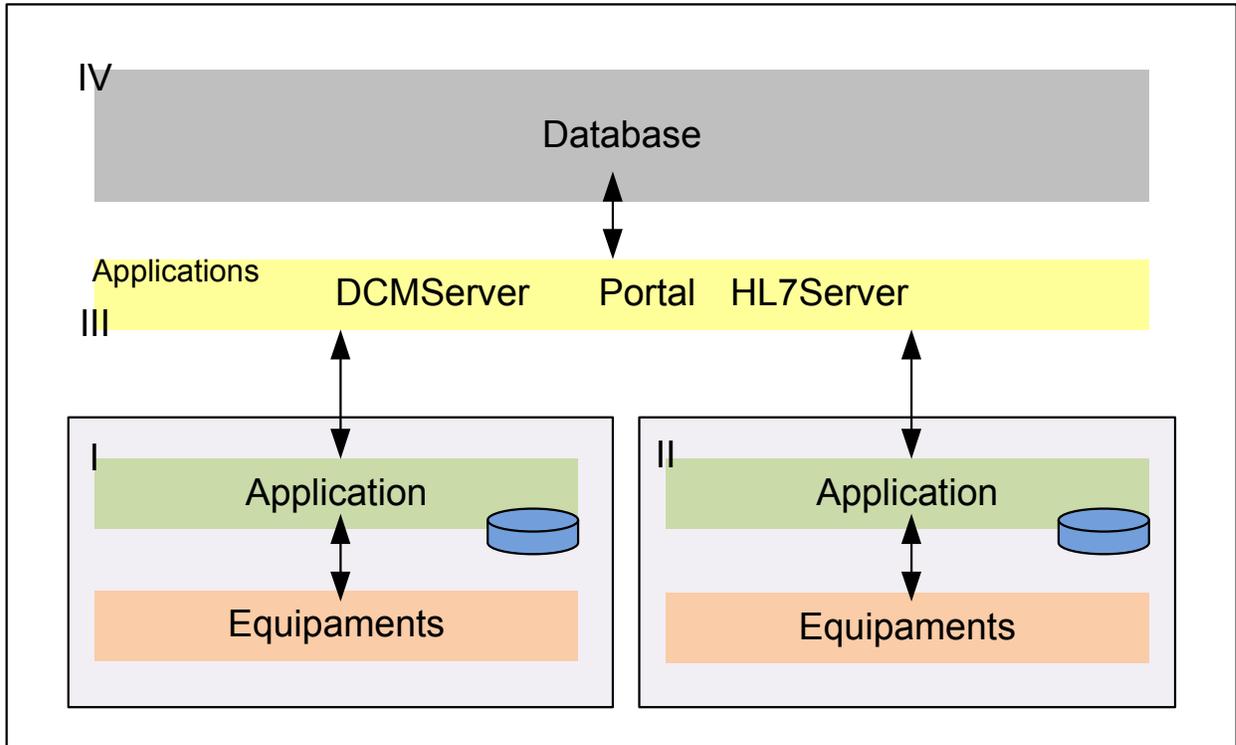


Figura 5 – PACS geograficamente distribuído na RCTM.

A Figura 5 exibe uma abstração modelo de PACS geograficamente distribuído presente na RCTM. Os quadros I e II representam duas instituições conectadas à rede. Em cada uma existe um conjunto de equipamentos que se comunicam com aplicativos, localizados em um servidor de ponte, que realizam o armazenamento dos exames adquiridos. Como exemplo, pode-se citar equipamentos de Tomografia Computadorizada enviando exames ao DCMServer em uma *bridge*.

A utilização de um servidor de ponte permite um acesso rápido as suas informações armazenadas, como exames em formato DICOM. Entretanto, periodicamente esses dados são transmitidos para os serviços presentes no quadro III. Este quadro representa o nível de aplicações que trocam informações com o banco de dados unificado (IV). Ferramentas como o DCMServer e o HL7Server são responsáveis pelo envio de exames ao banco, enquanto sua visualização e processos administrativos são realizados pelo Portal de Telemedicina.

3. *Metodologia*

O processo de desenvolvimento de um sistema possui basicamente cinco fases: estudo preliminar, análise do sistema atual, projeto lógico, projeto físico e projeto de implantação (Pressman 2006). Na primeira etapa são definidos os parâmetros iniciais do projeto – cenário, requisitos, limitações, impactos, etc. A segunda etapa caracteriza-se pela análise dos requisitos levantados e pelo estudo aprofundado do cenário do projeto. Na terceira e quarta etapa são desenvolvidos diagramas e o código fonte do aplicativo, respectivamente. Na implantação são realizados treinamentos e capacitação técnica.

O processo de análise e levantamento de requisitos de um aplicativo inicia-se pela descrição e estudo do cenário do problema encontrado. A partir desses dados é possível iniciar o processo de análise e gerência de requisitos. Por isso, esse processo é muito importante para diminuir os problemas encontrados na implantação do aplicativo e aumentar a aceitação do produto.

Por ser, tipicamente, um processo operacional interno, o problema de manutenção e administração remota das *bridges* na telemedicina pode ser estudado de forma mais ampla. Questões como a possibilidade de inclusão de outros serviços prontos ou em processo de desenvolvimento, a experiência adquirida no processo de implantação desse serviço e a análise de padrões, como o DICOM, permitem conclusões que podem alterar o foco da solução.

Atualmente a manutenção e administração das *bridges* têm sido feito com a utilização do protocolo *ssh* e de comandos do próprio Unix. Isso permite que todas as operações necessárias sejam executadas e que o sistema tenha um bom funcionamento. Entretanto, questões como a segurança de acesso restringem essa solução a poucas pessoas, o que torna o processo ineficaz e improdutivo.

Em função disso, é necessária uma solução que permita uma administração e manutenção ágil, fácil e segura tanto do sistema, quanto dos serviços instalados. Questões como a utilização de espaço em disco, frequência de utilização da rede e da memória são importantes para manter o bom funcionamento de todo o sistema. Portanto, operações de administração dessas características também precisam estar inseridas em qualquer mecanismo remoto de manutenção.

Além dos cuidados que o sistema operacional exige, existem também os específicos de cada serviço. Esses novos requisitos podem ser divididos em 2 grupos: geral e específico. Os primeiros referem-se às necessidades de qualquer serviço que possa ser instalado e que podem ser compartilhadas. Entre essas características estão as alterações de configuração, atualização, cópia de segurança e troca de mensagens. Especificidades desse tipo devem estar disponíveis para qualquer serviço que seja instalado. No entanto, é necessária uma padronização dessas operações para facilitar a integração de novos serviços e evitar o uso de outras ferramentas para esse propósito.

Porém, cada serviço instalado possui características próprias, como a sintaxe de suas configurações, estrutura de armazenamento de dados ou outra especificidade. Como exemplo, o *DCMServer* armazena as imagens recebidas em um banco de dados e necessita ler suas configurações de um arquivo no formato XML. Características como essas são únicas e podem mudar em cada aplicativo.

Além disso, outro problema como a deficiência na sincronia de informações dos bancos de dados locais com as informações disponíveis no Portal de Telemedicina, impossibilita uma melhor utilização de algumas características do sistema. Entre as falhas, pode-se citar a falta de controle de acesso as informações de paciente em Workstations radiológicas e a ineficaz transferência de dados entre servidores em um mesmo nível.

Como forma de viabilizar esse processo existe diversas soluções, como *middlewares* que realizam sincronização de banco de dados. Entretanto, a necessidade de um sistema de manutenção dos servidores que armazenam esses bancos de dados permite a utilização de conceitos semelhantes para esse problema. O modelo de gerenciamento também deve ser compatível o suficiente para permitir alterações de dados, que podem ser considerados com configurações, em banco de informações.

3.1. Análise do cenário

Diversas soluções para área médica estão sendo desenvolvidos para serem integradas ao sistema existente na RCTM. Essas soluções futuramente necessitarão ser integradas ao Portal de Telemedicina. Sistemas web possuem mecanismos de integração mais simples. Entretanto siste-

mas do tipo *standalone* podem exigir mecanismos complexos e camadas intermediárias de integração. Cada uma dessas soluções tem suas particularidades, sendo que algumas precisam de uma ponte para o Portal, enquanto outras trabalham diretamente com o aplicativo e não necessitam de cuidados especiais.

A ponte entre aplicações, como o *DCMServer*, que executam em servidores distribuídos e o Portal de Telemedicina é chamada de *bridge*. Esse servidor, normalmente, fica localizado no hospital em que se encontra o aparelho que realiza os exames, e fica sob responsabilidade conjunta entre o serviço de informática da instituição e da equipe de telemedicina. Especificamente na região metropolitana de Florianópolis, esses servidores encontram-se na rede do Governo do Estado. Tal fato o condiciona a mecanismos rígidos de segurança e tornam o processo de comunicação extremamente burocrático.

3.2. A Proposta

A partir da análise dos requisitos levantados (Apêndice A) e de um estudo aprofundado do problema e seu cenário, alguns pontos pertinentes podem ser levantados. Entre eles, pode-se citar:

- **Número crescente de equipamentos e hospitais a integrar.** A popularização do Portal de Telemedicina e a demanda de exames permitem o aumento do número de hospitais integrados. Esses podem possuir equipamentos de imagem com suporte aos padrões de exportação já existentes ou outras soluções. Tal fato, se ligado a necessidade de um servidor de ponte nessa situação, permite um número ilimitado de *bridges*;
- **Desenvolvimento de outras soluções.** Equipamentos antigos, normalmente, não possuem suporte a exportação de seus exames em formato DICOM. Outras soluções, como o HL7 já estão sendo utilizados em casos que o padrão DICOM não consegue suprir todas as necessidades ou que apenas necessite de uma solução mais simples;

- **Status do sistema.** A visualização da situação atual do sistema é importante tanto ao usuário que acessa o sistema, quanto ao seu administrador. Por isso, é importante a visualização do status atual de uma *bridge* e seus serviços de modo rápido;
- **Análise dos problemas encontrados.** Frequentemente os problemas que acontecem na troca de informações com os serviços instalados nos hospitais são de responsabilidade da rede interna da instituição ou de outro problema local. A partir disso, é preciso diferenciar e pré-identificar os motivos que originaram a falha de transferência e exibir o procedimento adequado que deve ser seguido e, posteriormente, um histórico de todas as ações realizadas;
- **Natureza da proposta.** Um sistema deve ser projetado e desenvolvido em função dos objetivos que pretende atender (Zachman 1987).

Levando em consideração as questões citadas optou-se pela criação de um modelo de gerenciamento de serviços geograficamente distantes. Esse modelo é o ponto de partida para serviços gerenciáveis e especifica regras formais e mecanismos de integração. Essa escolha é embasada por motivos como:

- **Facilidade de integração de outros serviços.** A criação de um conjunto de regras permite o desenvolvimento de módulos de integração para outras ferramentas que precisem ser integradas;
- **Transparência.** O estabelecimento de regras e normas permite o desenvolvimento de documentação clara e legível e torna seu entendimento mais fácil e rápido. Essas características viabilizam a construção de aplicativos de maneira mais eficiente;
- **Centralização.** Com a utilização de um modelo único, o processo de manutenção e gerenciamento de aplicativos torna-se mais rápido, fácil e acessível. O que permite a descentralização do processo de administração de recursos mais eficiente;
- **Descentralização.** Níveis de segurança e perfis de acesso permitem que diversas pessoas possam realizar a manutenção integralmente ou apenas de uma parte específica, sem abrir mão de segurança;
- **Redução de custos.** O custo com o desenvolvimento de soluções para gerenciamento de serviços é reduzido.

3.3. O modelo proposto

A partir dos dados levantados com funcionamentos dos sistemas atuais e dos estudos realizados anteriormente, é possível desenvolver um modelo que permita o gerenciamento remoto de sistemas de informação. Esse modelo consiste na definição computacional de regras e especificações técnicas sobre o seu funcionamento e métodos que permitam integração futura de outros aplicativos.

A utilização dessa metodologia permite facilidade na integração que aplicativos sejam mais facilmente integrados ao sistema, assim como garante integridade e transparência no fluxo de dados. Em função disso, é importante que o modelo seja bem planejado e especificado. O conjunto de regras propostas pode ser dividido em 6 grupos, os quais garantirão o bom funcionamento de todo o sistema.

3.3.1. *Organização do Sistema*

A arquitetura e a disposição funcional dos componentes de um sistema são importantes para o seu funcionamento, para o entendimento de suas rotinas e para sua manutenção. Em função disso, é importante que o processo de escolha de um modelo computacional e sua modelagem seja cuidadosamente realizado de forma coerente em função dos objetivos que pretendem ser alcançados. Entre os modelos existentes, o mais utilizado e mais recomendado para esse tipo de problema é a arquitetura cliente-servidor. Esse modelo computacional pode ser utilizado facilmente e permite o desenvolvimento de diversos tipos de soluções, entre eles: utilização de clientes gordos, servidor gordo, lógica distribuída, apresentação remota, entre outras.

Esse modelo computacional é facilmente encaixado e abstraído para o problema encontrado. Características como o número ilimitado de clientes e a centralização das informações troca-

das através de um servidor tornam esse modelo favorável para utilização. Sua abstração é simples, sendo que pode ser entendido da seguinte forma: Cada *bridge* caracteriza-se como um cliente gordo e comunica-se com um servidor central, localizado nos servidores da Telemedicina e responsável pela comunicação com todos os outros nodos.

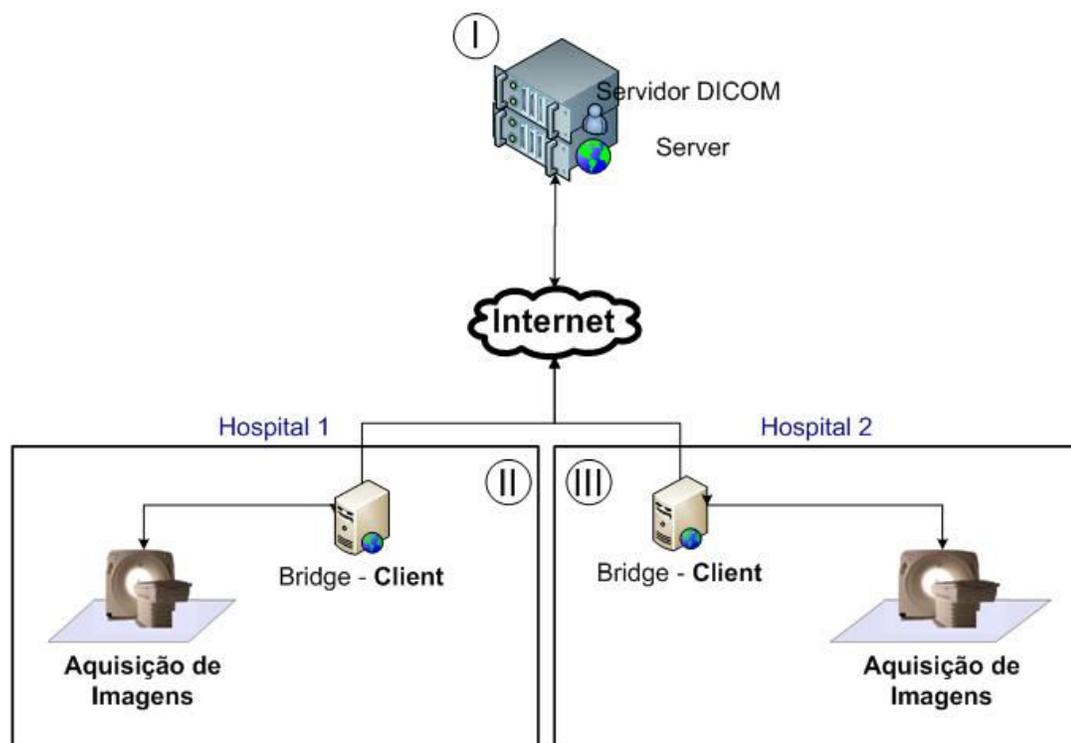


Figura 6 – Abstração do problema ao modelo computacional Cliente-Servidor

A Figura 6, mostra uma abstração da arquitetura do sistema aplicada ao gerenciamento de servidores geograficamente distribuídos. Nos quadros II e III estão representadas duas instituições distintas que realizam procedimentos e armazenam esses dados, em formato DICOM, em suas, respectivas, *bridges*. Além desse armazenamento, cada servidor de ponte também possui um cliente que realiza seu gerenciamento utilizando as chamadas requisitadas pelos servidores representados por I. Esses servidores, além de centralizar as informações dos clientes que realiza a administração das *bridges* também realizam o armazenamento dos dados gerados por todos os pontos integrados a rede.

3.3.2. *O Servidor*

Em um sistema com clientes e servidores, cabe ao servidor realizar a centralização das informações e da comunicação. No gerenciamento remoto dos serviços, além da centralização dos dados, são funções do servidor:

- Trocar informações com os clientes. O gerenciamento das mensagens trocadas deve ser realizado pelo servidor. Essa tarefa consiste em transmitir, re-transmitir e verificar a integridade das mensagens trocadas. Uma mensagem íntegra não deve possuir erros de sintaxe e necessita de uma semântica adequada. Alguns tópicos semânticos necessitam ser verificados, como forma de evitar trocas desnecessárias de mensagens inválidas.
- Armazenamento de dados. Dados freqüentemente utilizados ou que possuem um grau de relevância elevado devem estar armazenados de modo que permitam acesso local pelo servidor. Essa *cache* permite que dados como o histórico do status de cada cliente, de transmissão de dados e as mensagens trocadas fiquem armazenados de maneira que reduzam tempo gasto com acesso aos clientes e facilitem ações como administração e auditorias.
- Servir como ponto intermediário na comunicação entre 2 clientes. Garantir que a transmissão de informações entre 2 clientes passe pelo servidor permite maior integridade do sistema. Sendo assim, cabe ao servidor intermediar a troca de informações e armazenar o histórico desse evento.
- Atuar como camada intermediária entre o usuário e os servidores. O servidor deve ser um ponto intermediário na comunicação com os clientes, sendo que não é permitida a troca de mensagens diretamente entre um usuário e clientes. Isso implica na necessidade de outra camada que realize a interação do usuário com o servidor.
- Sincronização de dados. Dados que necessitem ser verificados e sincronizados devem ser comparados pelo servidor. Sendo que esse pode enviar mensagens com dados a serem alterados ou apenas realizar conferência de informações a partir de requisições efetuadas.

- Controle de Tarefas. Cabe ao servidor realizar o controle de fluxo de informações a nível de troca de mensagens com o cliente, ou seja: Verificar se existem mensagens pendentes com o cliente que impossibilitem a troca de informações.

3.3.3. *Cliente*

Normalmente, em um sistema que possui clientes e servidores, cabe aos servidores a maioria das operações. Na maioria dos sistemas, os clientes apenas são mecanismos que adquirem dados de maneira específica e os transmitem para que o servidor realize suas operações, como processamento e armazenamento de dados. No entanto, em um sistema de gerenciamento remoto de sistemas de informações é interessante que exista uma inversão nesses papéis, ou seja: a utilização de clientes gordos. Clientes gordos são aqueles que executam a maior parte do processamento do sistema, como em sistemas distribuídos. Esse processo permite maior liberdade na relação cliente/servidor e reduz riscos de panes e falhas sistêmicas em função de problemas pontuais no servidor.

Portanto, pode-se afirmar que o servidor atua basicamente como um mecanismo transmissor de mensagens, enquanto os clientes são responsáveis pelos serviços gerenciáveis e alguns pontos do sistema operacional. Para realizar esse processo, cada nodo folha necessita de um conjunto de ferramentas que permitam a execução de tarefas pertinentes. Essas tarefas podem ser divididas em alguns conjuntos, sendo que cada é importante para o funcionamento do sistema como um todo, o qual possui uma ou mais dessas partes. Abaixo estão listadas e descritas cada uma dessas funções.

- **Comunicação com os serviços gerenciados.** Sinais de sistema, conhecidos como *Unix Signals*, consistem em um meio de passar sinais a um aplicativo em execução, que pode os interpretar e realizar algum procedimento em função de seu estado atual. Esses sinais são o principal meio de comunicação entre o sistema de gerenciamento e seus serviços gerenciáveis. No entanto, nem todos os aplicativos possuem suporte a esses sinais ou a sua adaptação pode se tornar muito complicada. Como forma contornar esse problema, o cliente também pode comunicar-se com os serviços por meio de mensagens XML no padrão WebService, com sintaxe

xe definida caso-a-caso. Esse padrão serve como ferramenta que possibilita a interoperabilidade entre sistemas e garante comunicação entre serviços.

- **Alteração de configuração dos serviços.** Cada cliente também é responsável pelas configurações de cada sistema gerenciado e de suas configurações também. Isso facilita o processo de alteração de configurações porque o centraliza, além disso aumenta a segurança e confiabilidade dos sistemas. Essas configurações, cifradas ou não, podem estar armazenadas em um banco de dados, arquivos de configurações ou, em alguns casos, em arquivos XML.
- **Agendamento de operações sobre serviços ou sistema.** Algumas operações sobre os serviços ou ao sistema operacional não podem ser executadas a qualquer momento. Isso ocorre pelo fluxo de dados nos serviços ou mesmo pela necessidade de manter o sistema em funcionamento. É necessário, então, que o sistema permita o agendamento de operações em horários e dias específicos.
- **Manutenção de serviços do sistema.** Alguns aplicativos instalados no sistema operacional são importantes para o correto funcionamento dos serviços gerenciados. Portanto, é importante que também sejam observados e administrados. Sistemas Gerenciadores de Banco de Dados são um exemplo disso e, como a maioria dos sistemas, precisam de manutenção constante para garantir desempenho adequado.
- **Administração das informações geradas pelos serviços.** Serviços do sistema geram informações armazenadas em disco, banco de dados ou em outra estrutura de dados e muitas vezes informações desnecessárias - lixo. Esses dados também precisam ser administrados para evitar problemas como a redução do espaço disponível em disco e tempo de resposta. Dados de sistemas apenas devem ser excluídos quando excederem um período de tempo pré-determinado ou quando algum ponto do sistema chegar a um estado crítico.
- **Controle de fluxo de processos.** Usualmente, existem processos que necessitam ser executados exclusivamente. Pode-se afirmar que isto caracteriza processos críticos. Em função disso, é importante que o cliente consiga manipular esses processos de modo que apenas sejam executados um a um. Isso deve ser realizado por meio de semáforos sobre processos e do armazenamento do contexto de cada um desses aplicativos.

A partir dessas funções é possível efetuar a maioria das operações rotineiras necessárias para manter um sistema em funcionamento. No entanto, cada processo necessita ser chamado a par-

tir de uma visão humana de sistema. Cada processo assim possui um custo e o tempo gasto pelos mecanismos de segurança o torna mais elevado ainda. Além disso, diversas dessas ações são efetuadas a partir de um processo de rotina, como a exclusão de exames quando o espaço disponível em disco atinge determinado estado. A partir desses dados é possível mapear essas ações e automatizar processos como forma de reduzir o tempo gasto com essas operações. Com esses dados em mãos, é possível desenvolver mecanismos que realizem ações pré-programadas em função do estado atual do sistema e paradigmas de inteligência artificial.

3.3.3.1. *Sistema de autogerenciamento*

A automatização de algumas tarefas do sistema aumenta sua confiabilidade e reduz o tempo gasto com ações facilmente rastreáveis. Tarefas auto-executáveis, porém, necessitam de características que garantam uma execução independente, inteligente e possibilitem ganho real de tempo e desempenho. Entre as características que marcam essas tarefas pode-se destacar a independência de parâmetros ambientais (dados que possuem grande dependência do contexto e do ambiente em que esta inserida) e execução independente do sistema, ou seja: não seja diretamente dependente da execução de outras ações pelo sistema operacional ou outros serviços.

Sistemas críticos que podem ser utilizados a qualquer momento precisam, entretanto, de outros cuidados especiais. O fluxo de informações precisa ser constantemente controlado e analisado de forma que tarefas pré-programadas apenas sejam executadas em momentos adequados. Durante a execução desta ação, o sistema deve ser capaz de aferir se existe necessidade ou não de redirecionamento de fluxo de dados como forma de manter um funcionamento adequado dos serviços utilizados.

Indiretamente, o controle de fluxo de dados possibilita aumento na eficiência do sistema por meio da melhoria de sua entrada e saída. No entanto, é importante que exista uma metodologia clara de controle de fluxo baseada em sistemas de escalonamento ou inteligência artificial para evitar *deadlocks*. Somente com um sistema como esse é possível automatizar a tomada de decisão de tarefas. Entre as tarefas mais usuais e que possuem, inerentemente, as características necessárias para isso pode-se citar:

- Manutenções de sistema;
- Controle de estado do sistema (status);
- Controle de históricos;
- Cópias de segurança;
- Atualizações Automáticas.

3.3.3.2. *O Processo de Manutenção do Sistema*

As tarefas de manutenção de sistemas são fundamentais para garantir o correto funcionamento de todos os serviços. Entre essas tarefas, provavelmente a mais usual é o controle de espaço livre em disco e sua limpeza. Somente por meio de controles como esse, é possível permitir o funcionamento de serviços em sua plenitude. No entanto, mesmo processos aparentemente simples necessitam ser desenvolvidos de forma a atender as necessidades do sistema. Como exemplo, a limpeza de disco precisa ser realizada em 2 níveis: Rotina e Emergência. Cabe ao sistema descobrir o momento correto de realização de tarefas como essa para evitar riscos com panes e falhas.

Entre outras operações que atendem esse requisito pode-se destacar ações de rotina na manutenção de banco de dados, como otimizações e re-indexações. Em nível de *software*, normalmente poucas alterações são realizadas. Entretanto, sistemas sempre estão suscetíveis a atualizações e correções de *bugs*. Em função disso, cabe ao sistema gerenciar esse processo e realizá-lo de forma clara e simples o suficiente para o retorno a versão anterior em caso de falhas.

3.3.3.3. *Cópias de Segurança*

Rotinas pré-programadas de manutenção de sistema garantem seu funcionamento e evitam paradas por falta de recursos, entre outros. No entanto, alguns erros não são recuperáveis ou podem causar grandes estragos ao serviço ou ao sistema computacional. A recuperação do siste-

ma em casos como este, normalmente, obriga a restauração de cópias de segurança anteriormente armazenadas.

Essas cópias são uma garantia que permite agilizar o processo de recuperação de falhas ou ataques ao sistema e evita maiores perdas de informações. Em um sistema que possui diversos servidores, a automatização desse processo é importante e garante cópias freqüentes o suficiente para esses processos. Por isso, cada cliente deve realizar uma cópia de segurança, em tempo pré-determinado ou quando necessário, dos seus dados de configuração, dos serviços e dos dados freqüentemente mais acessados. Esse processo é realizado localmente e depois de concluído é automaticamente enviado ao servidor que retransmite uma instancia a um local devidamente organizado para isso.

Ao passo que necessário, o próprio cliente, desde que em situações fora de controle, pode requerer uma instância dessas cópias para o servidor. Quando isso acontece, um administrador do sistema é automaticamente avisado através de um meio eletrônico e pode dar seguimento às ações necessárias. A interação humana nesse processo é fundamental e garante uma comprovação do estado e da real necessidade de uma atitude drástica como essa.

3.3.4. *Visão do sistema*

Uma das características de aplicativos em geral é possuir ou não uma visão gráfica que possibilite interação com seu usuário. Esse processo, normalmente, é realizado por meio de visões gráficas utilizando janelas ou por meio do uso de terminais. A necessidade da utilização desta *feature* deve ser observada pelo aplicativo, sendo características como a necessidade de parâmetros em tempo de execução, os usuários padrões e sua ergonomia como fatores importantes para isso.

Abstraindo a utilização e necessidade de interface gráfica no problema de gerenciamento remoto, a primeira conclusão que pode ser tomada é a desnecessariedade de uma visão gráfica em cada cliente. A centralização das informações de cada cliente e do servidor em um único ponto facilita o processo de gerenciamento em relação ao conjunto de servidores gerenciáveis e isso pode ser estendido a uma visão única de sistema. No entanto, a conclusão de utilização de uma

interface web ou de clientes gráficos é inteiramente ligada às características do sistema em que se encaixam os servidores.

Essa interface gráfica é totalmente independente do funcionamento do sistema, funcionando exclusivamente como um mecanismo de visão de dados. Apenas deve existir uma ponte de comunicação entre o servidor e as informações disponibilizadas. Isto deve ser realizado por meio da troca de mensagens XML, utilizando o mesmo padrão e as mensagens trocadas entre os clientes e os servidores. No entanto, em cada mensagem deve ser acrescentado dados referentes à identificação do cliente ou do servidor relacionado à mensagem.

3.3.5. *Comunicação*

Em sistemas que utilizam clientes e servidores remotamente localizados em computadores diferentes ou mesmo para sistemas distintos, a comunicação é um ponto fundamental para garantir o correto funcionamento desses ambos os sistemas. Por meio da comunicação entre esses pontos é possível realizar trocas de informações e manter sincronizadas diferentes instâncias de softwares. Para o correto funcionamento da comunicação entre os sistemas, entretanto, alguns pontos precisam ser levados em consideração: protocolo, padrão e segurança.

Atualmente, os protocolos TCP (acrônimo do inglês *Transmission Control Protocol*) e UDP (*User Datagram Protocol*) são os mais utilizados para transporte de informações utilizando *sockets*. As principais diferenças entre ambos estão na segurança do transporte de dados e mecanismos de segurança. O protocolo TCP/IP possui mecanismos de detecção de erros e segurança para o envio/recepção de dados. Em função disso e da necessidade de segurança de transporte, o protocolo TCP é o mais indicado para troca de informações em aplicativos de gerenciamento remoto ou em outros aplicativos que necessitem de um grau mínimo de confiança de tráfego. Já o protocolo UDP é mais indicado para aplicativos que necessitem de maior velocidade no processo de transmissão.

Utilizando um protocolo, é possível padronizar e facilitar a comunicação entre dois aplicativos. No entanto, seu processo de desenvolvimento é muito mais amplo e complexo. Pontos como a tipagem das mensagens, cifragem dos dados e rotinas de transmissão são fundamentais para a comunicação entre clientes e servidores.

A mensagem é ponto fundamental para comunicação entre clientes e servidores. É nela que tramitam as informações e seu processo de reconhecimento é fundamental para que um sistema, analogamente, “entenda” o que outro quer dizer. Em função disso, o padrão de mensagem é importante para facilitar o processo de reconhecimento e permitir a inclusão de novas mensagens (especificamente, necessárias na adição de novos serviços a serem gerenciados). Pensando nesses aspectos, a mensagem precisa ser simples o suficiente para permitir acréscimo de informações e novos tipos e robusta o suficiente para garantir sua integridade e de seus dados. Levando-se em consideração esses pontos, o padrão *Webservice* encaixa-se como solução viável, o qual utiliza o padrão XML para o armazenamento das informações. Esse padrão foi desenvolvido para servir de modelo de integração entre sistemas diversos (Alonso, et al. 2003).

Normalmente utiliza-se como meio de integração de sistemas web, utilizando o protocolo HTTP para comunicação. No entanto, nada o impede de ser utilizado com outros protocolos, o que o torna flexível o suficiente para ser utilizado.

Sua utilização é feita com a utilização do protocolo SOAP (acrônimo de *Simple Object Access Protocol*), o qual permite invocações remotas de procedimentos e trocas de mensagem independente de ambientes. Cada mensagem armazena um objeto, o qual é identificado e processado pelo receptor da mensagem. As mensagens possuem envelope e corpo, entre os itens obrigatórios, um cabeçalho e um retorno entre os opcionais de sua sintaxe.



Figura 7 – Estrutura de uma mensagem WebService

A Figura 7 mostra a estrutura de uma mensagem. No “Envelope” estão contidos o estilo de codificação, declarações e atributos adicionais. No cabeçalho, que é opcional, estão definidos atributos adicionais ao primeiro subgrupo. Informações como características de processamento e tipo de mensagem estão definidos nessa parte. O corpo da mensagem é onde estão os dados a serem transportados, como parâmetros de execução ou informações. Já o retorno, opcional, é onde são carregados *status* de erro e valores de retorno.

3.3.5.1. *O Processo de Transmissão de Dados*

Pode-se dividir o processo de transmissão de dados em duas partes: Associação e Transferência. Ambas são importantes para o processo de transmissão de dados e são interligadas de forma que uma Transferência só ocorre após o sucesso do processo de associação. Cada etapa dos dois processos é caracterizada por um tipo de mensagem. Essas mensagens possuem identificação única e necessitam ser transmitidas de forma a manter a integridade do sistema. Abaixo são descritos os dois processos:

Associação. Essa etapa é onde ocorre a primeira troca de dados entre clientes e servidores e onde se define os mecanismos e padrões utilizados na Transferência. Uma mensagem é enviada contendo informações como a identificação de clientes e servidores, chaves de acesso e a cifra-gem utilizada na transferência dos dados.

A Figura 8 demonstra como o processo de associação funciona. Inicialmente o requisitante envia uma mensagem contendo sua identificação, sua chave de segurança e seus parâmetros de execução que serão utilizados posteriormente no processo de Transmissão. A partir desses dados, o receptor verifica se o requisitante é válido, se possui suporte ao tipo de Transmissão de dados exigida e se suporta a cifra-gem de dados solicitada. Caso todos os parâmetros sejam atendidos, o receptor responde o cliente com status de sucesso, caso contrario retorna uma mensagem de erro. Essa mensagem possui o código do erro encontrado.

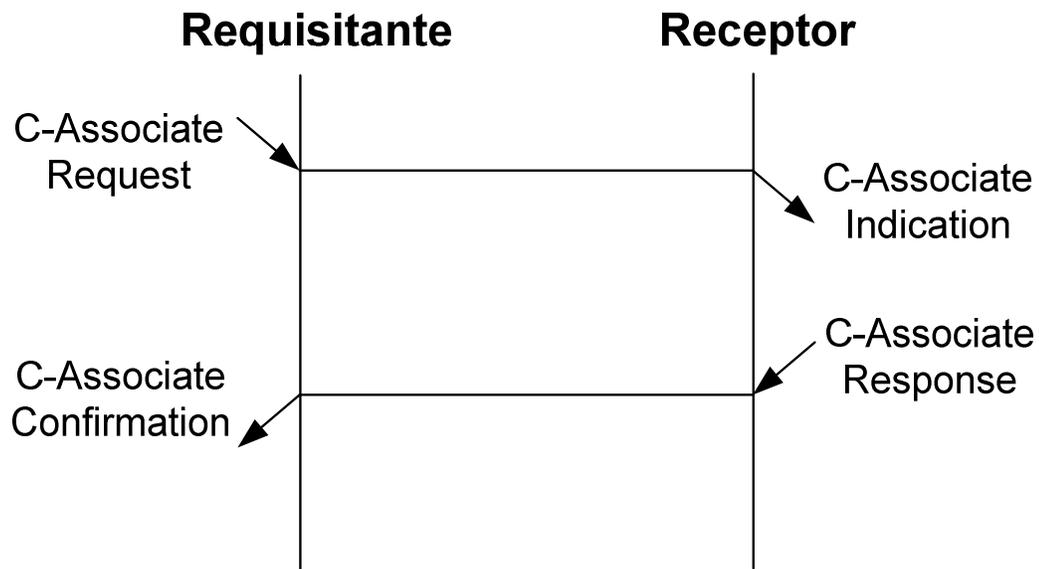


Figura 8 - Processo de Associação entre cliente e servidor

Transmissão. Após o sucesso no processo de negociação, é possível iniciar o processo de transmissão, no qual ocorre a troca de informações. Abaixo estão listadas cada uma dessas partes, assim como estão descritos os principais passos de cada uma das partes.

- *Data Transfer Request:* Mensagem enviada pelo Requisiteante avisando o Receptor que iniciará o processo de transmissão de dados.
- *Data Transfer Confirmation.* Com essa mensagem o Receptor informa o Requisiteante que pode iniciar a troca de dados. Caso a troca de informações não possa ser iniciada, por motivos como número máximo de clientes conectados ou tarefas que exigem exclusividade, a mensagem de retorno contém um aviso que impede o início do próximo passo.
- *Data Send.* Um conjunto de dados, de tamanho máximo negociado no processo de associação é enviado.
- *Data Received.* O receptor confirma o recebimento dos dados ou pede o re-envio, em caso de erro.
- *Release Indication.* Indicação de que terminou o processo de transmissão.

- *Release Confirmation*. Mensagem que indica o término de todo o processo. O envio dessa mensagem depende diretamente dos parâmetros definidos no processo de associação. Essa mensagem pode ser enviada logo após o término da transmissão, ou após o término da execução da operação associada. Os erros associados a essa mensagem podem invalidar todo o processo de Transmissão ou apenas uma parte específica, dependendo do tipo de falha.

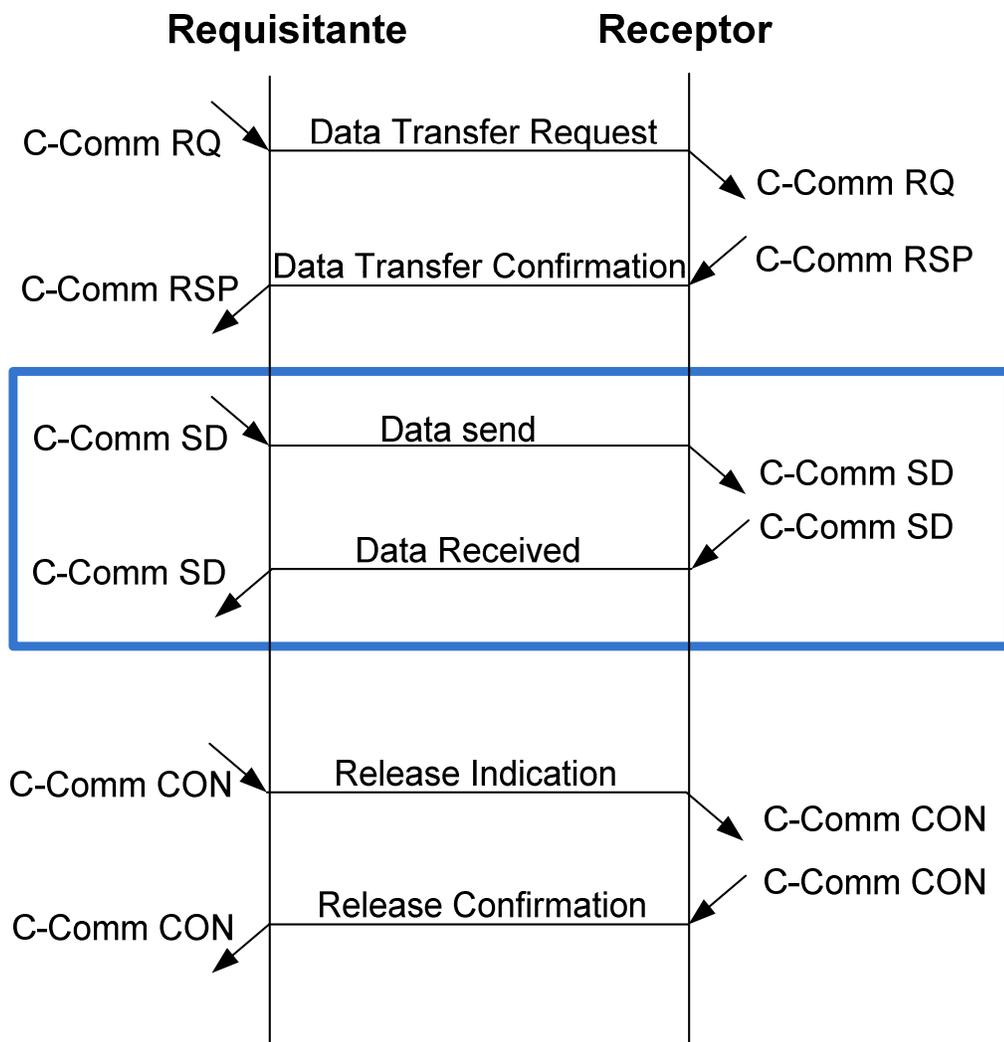


Figura 9 – O processo de Transmissão de dados

A Figura 9 exibe os principais eventos que ocorrem no processo de transmissão de dados. O sinal de “C-Comm RQ” refere-se a *Data Transfer Request*. Essa mensagem de requisição de transferência de dados é respondida pelo sinal de *Data Transfer Confirmation*, o qual pode autorizar ou não o processo de troca de mensagem.

Caso a transferência de dados seja autorizada, os sinais de Data Sent e Data Received (C-Comm SD) são trocados enquanto existirem pacotes a serem transmitidos. Ao término desse processo, o Requisitante emite um sinal de *Release Indication* informando o Receptor que todos os dados foram transmitidos. Após o recebimento deste sinal, o Receptor realiza os procedimentos de reconhecimento e os indicados pela mensagem e retorna ao Requisitante uma mensagem de falha ou sucesso na mensagem de *Release Confirmation*.

3.3.5.2. Definição de mensagens

Os tipos de mensagens definem diretamente o conteúdo e as informações válidas em seu corpo. No padrão Webservice, a mensagem pode ser tipada diretamente em seu cabeçalho. Isso facilita e torna claro o seu processo de reconhecimento. Em função disso, aliado aos tipos de transmissão de dados, as mensagens podem utilizar a mesma subdivisão:

Mensagens de Associação.

No processo de associação entre o requisitante e o receptor é onde se realiza toda a negociação de como será efetuada o processo de Transmissão. Portanto, apenas tramitam dados referentes ao processo em si, deixando-se para depois a troca de mensagens com informações relevantes. Entre os principais atributos negociados nessa fase, pode-se destacar:

Encrypto: Tipo de cifragem utilizada nos dados.

Type: Tipo da mensagem. Assume valor para mensagem de Associação.

Id: Identificação do requisitante.

Key: Chave de segurança do requisitante.

Messages Type: Tipos de mensagens de Transmissão que serão trocadas. (relativas a aplicação, modularização)

Max Package Size: Tamanho máximo de pacote a ser transmitido.

Release Processor Type: Tipo de *Release Confirmation* exigido: após o término da transmissão ou do processamento.

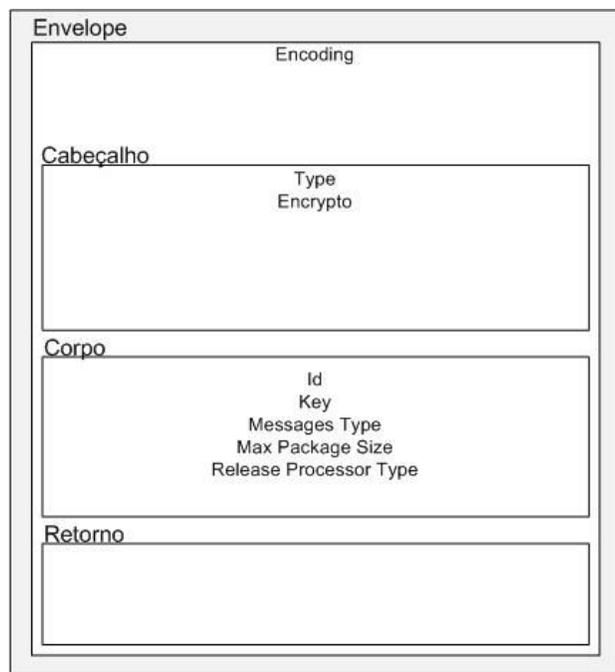


Figura 10 – Estrutura das Mensagens de Associação

A Figura 10 mostra como estão distribuídos os principais atributos nas mensagens de associação. Nessa fase são definidos os parâmetros utilizados no processo de troca de informações.

Mensagens de Transmissão.

O processo de Transmissão é onde ocorre a troca de informações em si baseados diretamente nos parâmetros definidos no processo de Associação. As mensagens trocadas neste processo são dependentes diretamente da configuração do cliente, servidor e dos serviços gerenciados. Em função disso, o número de mensagens válidas e de atributos relevantes é elevado. Abaixo pode-se observar os atributos genéricos mais relevantes.

Type: Tipo da mensagem trocada. (Transmissão)

Encrypto: Tipo de cifragem utilizada nos dados.

Message Number: Número da mensagem. Refere-se a qual parte esta sendo transmitida, caso o tamanho da mensagem exceda o tamanho máximo de mensagem.

Message Type: Tipo de mensagem. Referente diretamente ao tipo de aplicação utilizada.

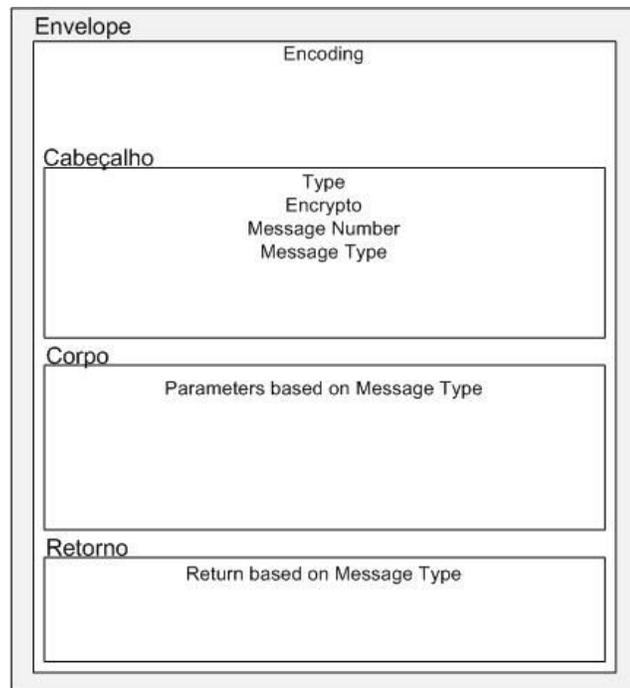


Figura 11 – Estrutura das Mensagens de Transmissão

A Figura 11 exibe a distribuição dos atributos em mensagens de Transmissão. Por ser um tipo amplo de mensagem, diversos atributos são definidos como locais, como pode-se observar no corpo e no retorno da mensagem.

3.3.6. *Segurança*

A segurança em sistemas cliente-servidor é fundamental para o funcionamento de todo o sistema. Em função disso, é importante garantir a segurança de comunicação, acesso e processamento das informações que tramitam pelo sistema. Podem-se considerar esses pontos como um

gargalo que exigem mecanismos distintos de tratamento. Abaixo estão descritos os principais meios utilizados para assegurar isto.

Comunicação. A segurança no processo de comunicação pode ser assegurada, basicamente, por meio de dois pontos: autenticação de acesso e cifragem de dados. Mecanismos baseados em identificação de usuário e chaves de segurança são capazes de realizar a identificação do requisitante de um processo de comunicação. Por meio dessas chaves que o cliente pode identificar o cliente que requisita transmissão de dados e verificar se não houve interceptação dos dados. Essas chaves devem possuir um mínimo de 16 caracteres para garantir um mínimo de segurança e dificultar a utilização de algoritmos de quebra de chaves baseados em força bruta.

Estas características são suficientes para reduzir potencialmente riscos de autenticação de usuários não autorizados. No entanto, a confidencialidade dos dados transmitidos exige que as mensagens trocadas sejam cifradas. Entre os mecanismos que solucionam esse problema pode-se citar a utilização de chaves públicas e privadas para criptografia de dados. Esse método consiste na cifragem, por uma chave pública, dos dados pelo requisitante e na leitura e no reconhecimento dos dados pelo receptor com a utilização de uma chave privada única.

A utilização de mecanismos de segurança deve ser definida no processo de associação no atributo *Encrypto*. Esse campo define qual o tipo de segurança deve ser utilizada e seu valor deve ser aceito pelo receptor das mensagens. Usualmente o mecanismo SSL é suficientemente seguro para garantir a segurança. No entanto, a utilização, ou não, de outros padrões deve ser definida no mesmo campo referente a segurança e precisa ser negociado do mesmo modo entre clientes e servidor.

Acesso. A segurança de acesso ao sistema deve ser realizada em nível de usuário. Ou seja, a aplicação que realiza a visualização e permite a execução de comandos deve possuir mecanismos que garantam acesso seguro as configurações. Níveis de acesso são importantes e permitem descentralizar a manutenção e administração dos sub-sistemas.

4. *Cyclops Healthcare Integration System*

A utilização de um modelo com servidores de ponte entre os equipamentos que adquirem imagens e Portal de Telemedicina aumenta o número de servidores a serem gerenciados e, conseqüentemente, os problemas relativos a esse processo em toda a RCTM. Como forma de solucionar esse problema, a ferramenta *Cyclops Healthcare Integration System (CHIS)* será responsável pela administração e manutenção de serviços e de algumas partes do sistema operacional. Esse aplicativo é específico para as características encontradas em toda a rede e deve gerenciar os serviços de transmissão de exames disponíveis em todo o sistema, como o HL7 Server.

Outros aplicativos, como o *Cyclops PingPong*, que realiza o monitoramento do status dos servidores e pelo funcionamento dos serviços DICOM instalados, devem ser integrados ao sistema. A confiabilidade de todo o sistema passa diretamente pelo processo de integração desses serviços e a segurança garantida pelo sistema. Por isso, é importante que se tome cuidados inerentes a cada aplicativo e que se mantenham suas características principais inalteradas.

4.1. Características importantes

A integração entre dois aplicativos é, normalmente, facilitada quando se mantém suas características. Como forma de reduzir os problemas encontrados por falta de coerência neste sentido, o CHIS precisa ser desenvolvido de modo a manter as principais características de cada aplicativo. Inicialmente pode-se considerar 5 softwares que serão integrados:

- **DCMServer.** Serviço que lê suas configurações de um arquivo XM, recepta imagens no formato DICOM 3.0 e as copia ao Portal de Telemedicina.
- **CycClient.** Aplicativo que percorre um diretório e realiza o envio de todos os arquivos DICOM encontrados com as características específicas ou arquivos XML que podem ser enviados ao servidor HL7.

- **HL7Server.** Servidor que implementa o protocolo HL7 e permite acesso a banco de dados e, futuramente, realizará o envio de exames de não-DICOM como Eletrocardiograma e alguns Ultra-sons ao Portal de Telemedicina.
- **Portal de Telemedicina.** Visão de sistema. A partir do Portal o usuário final de todo o sistema tem acesso aos exames realizados por um sistema web.
- **PingPong.** Sistema de monitoramento de status de servidores de Ponte.

Os pontos fundamentais de todos esses aplicativos são sua natureza e seu comportamento. Cada um possui pontos específicos que necessitam serem preservados ou características que impossibilitam, inviabilizam ou dificultam o desenvolvimento de soluções paralelas para sua administração ou a visualização de suas informações. Em função disso, é importante que esse processo seja realizado de forma conjunta, utilizando páginas *web* integradas ao Portal de Telemedicina, com informações compartilhadas entre visualização, alteração e acesso aos seus dados.

4.2. Arquitetura do Sistema

A partir dos sistemas já existentes e de suas configurações atuais é possível elaborar uma solução que permita o correto gerenciamento das tecnologias já existentes e proporcione a integração de novos serviços. A partir desta premissa e dos problemas ocasionados por alterações drásticas na arquitetura existe é importante que a ferramenta CHIS tenha apenas papel auxiliar as soluções já existentes.

A Figura 12 exibe a abstração da arquitetura de todo o sistema com a integração do sistema CHIS. Este é o centralizador do processo de administração e manutenção dos servidores em que serviços responsáveis pela transmissão de exames executam. Em função disso e das características da arquitetura, pode-se considerar que o CHIS e o Servidor DICOM estão em uma mesma camada de sistema e possuem restrições de acesso e funcionamento semelhantes.

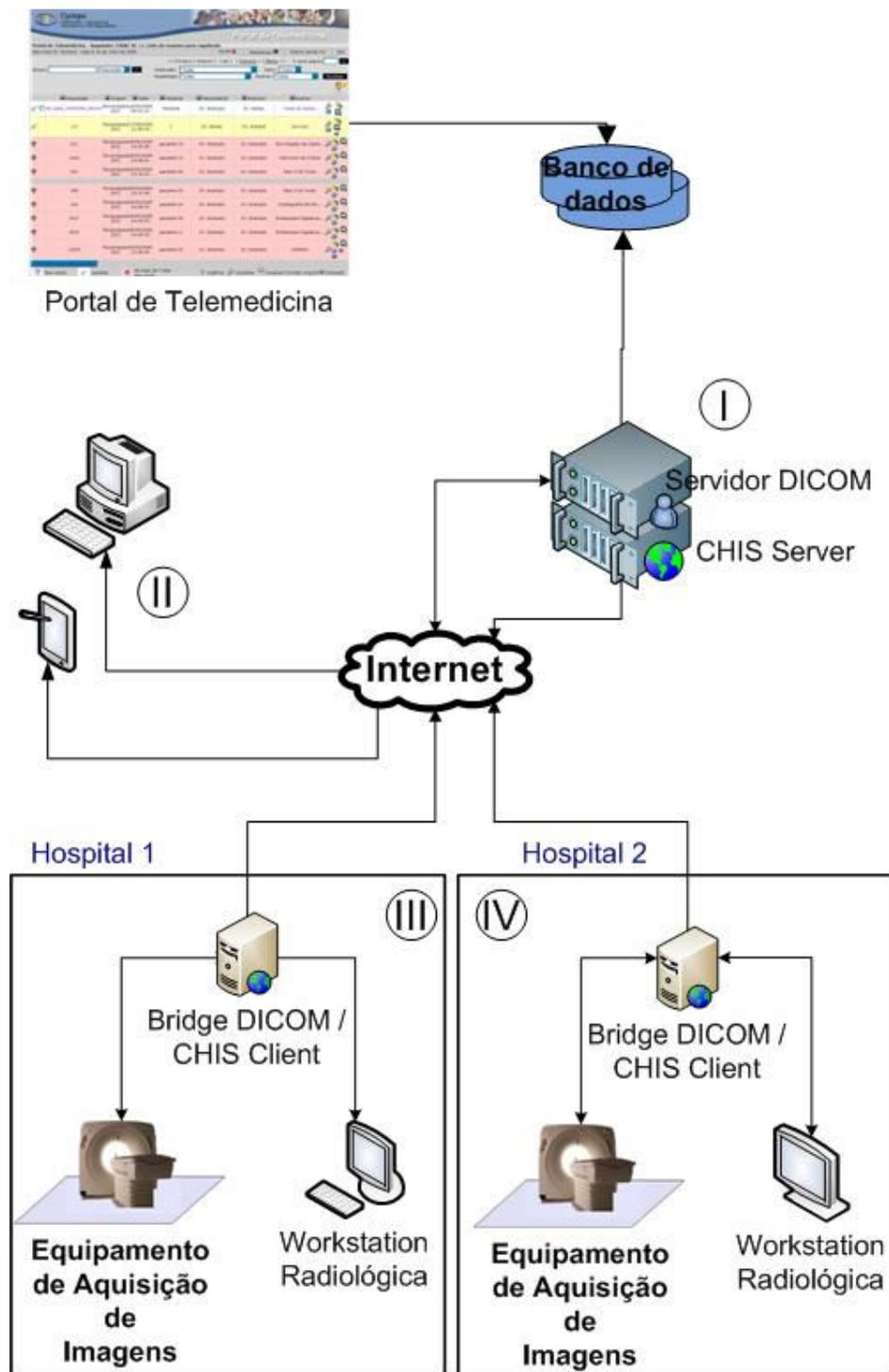


Figura 12 – Arquitetura do Sistema

A comunicação entre *CHIS Server* e seus clientes é realizada por meio de mensagens XML no padrão WebService. Cada cliente é responsável pelos serviços que executam em seu

A Figura 13 mostra as duas áreas de acesso do Portal de Telemedicina. À esquerda está a visualização dos exames, em forma de lista. Já à direita está uma visão do módulo de administração, ao qual podem ser adicionados usuários, instituições, municípios e modalidades de exames, entre outros.

A divisão realizada nos perfis pode ser ampliada e utilizada para delimitação de níveis de acesso de usuário. Para isso, é necessária a criação de grupos funcionais para as alterações que podem ser realizadas. A partir desses agrupamentos o usuário do sistema pode visualizar e realizar chamadas específicas nos clientes que possui permissão.

É importante salientar que o Portal de Telemedicina adquire a responsabilidade pela gerência de acesso as informações do sistema. Isso também acontece pela utilização dos perfis de acesso e pela autenticação já exigida para entrar no sistema. Esta divisão, no entanto, é extremamente funcional e retira a responsabilidade de autenticação e validação de usuários do sistema do CHIS Server, facilitando e reduzindo o tempo gasto com mensagens de erros e validação de usuários.

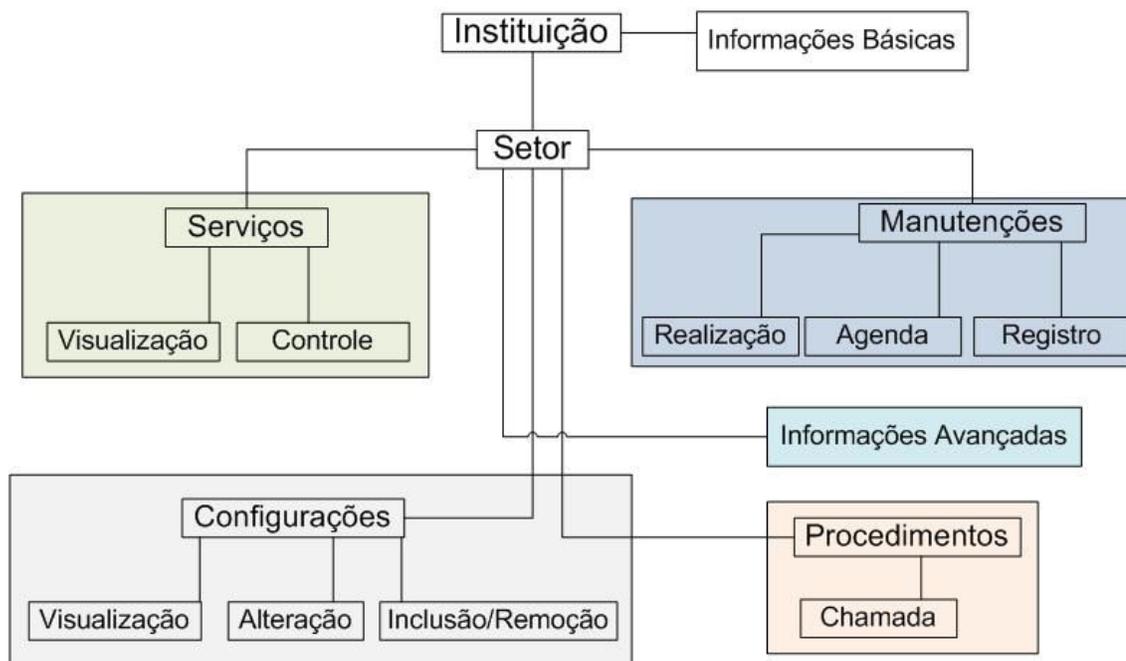


Figura 14 – Níveis de Acesso

A Figura 14 exhibe um diagrama com os níveis de acesso e suas ações básicas de operação para cada grupo funcional. Cada nodo do gráfico representa um grupo e cada módulo representa uma ação que pode ser realizada. A raiz de todo o sistema é a instituição operante, sendo que

abaixo estão delimitados os setores cadastrados e suas respectivas informações. A listagem, as permissões de acesso estão descritas abaixo.

- **Instituição:** Dados básicos referentes a Instituição onde localizam-se os equipamentos. Todos os usuários associados têm acesso a esses dados.
 - **Informações Básicas:** Dados referentes ao funcionamento e aos status funcionais da *bridge*. Nesse ponto relacionam-se informações como o histórico de desligamentos e status atual do sistema (em manutenção, ok, etc.).
- **Setor:** Informações referentes aos seus exames realizados e ao seu fluxo de dados.
 - **Informações Avançadas:** Dados referentes as estatísticas e informações referentes ao setor. Em função disso, podem ser visualizados por todos os usuários associados.
 - **Serviços:** Esse módulo possui as informações referentes ao funcionamento do aplicativo que realiza a transmissão dos exames. A partir dele é possível visualizar seu status atual, o histórico de transações e transmissões. Esses dados estão disponíveis para todos os usuários do setor, no entanto as informações de controle apenas estão visíveis aos administradores do sistema. A partir desses dados é possível realizar paradas de sistema, enviar sinais de controle ou mensagens ao serviço.
 - **Manutenções:** As informações referentes a manutenções e paradas programadas do sistema podem ser visualizadas por meio desse módulo. Esses dados são compartilhados por todos os usuários associados ao setor e permitem e exibem os dados cadastrados diretamente pelo administrador do sistema. Essas informações referem-se diretamente ao registro de novas paradas e a agenda programada.
 - **Configurações:** Esse grupo reúne informações referentes as configurações dos serviços executados. A visualização dos dados associados ao seu cliente pessoal podem ser visualizados pelo usuário cadastrado. Já as outras ações, apenas estão disponíveis para acesso pelo administrador do sistema. Essas referem-se a inclusão, remoção e alteração de clientes, servidores e dados cadastrados nas base de informações dos serviços.
 - **Procedimentos:** Esses dados, visíveis apenas ao administrador do sistema, permitem o envio de procedimentos a serem executados pelo cliente. Entre essas chamadas pode-se citar a remoção de exames para limpeza de disco, manutenções e configurações em serviços do sistema operacional.

4.4. Integração com o *Cyclops DCMServer*

A ferramenta *Cyclops DCMServer* é responsável pelo recebimento, armazenamento e comunicação de arquivos no padrão DICOM 3.0. Esse aplicativo trabalha como um *middleware* entre os equipamentos que realizam os exames e o Portal de Telemedicina. A utilização dessa ferramenta também permite o acesso ao banco de exames e requisições de transferência dessas informações para outros computadores através de comunicação em formato DICOM. Analisando esses dados, sua natureza de funcionamento – apenas um processo de sistema, usuário final não tem acesso direto, entre outras características – as definições do padrão DICOM e suas configurações, chega-se as seguintes conclusões:

- É importante que exista integração entre as configurações do servidor e as informações do Portal. Referências entre os equipamentos, os setores em que está implantado e seus usuários associados garantem maior controle sobre seus dados.
- Os dados armazenados necessitam de maior transparência de acesso. A nível de acesso, a criação de uma identificação (*AETitle*) para cada usuário é uma alternativa ponderável. Pensando-se em informações duplicadas e dessincronizadas, é necessário um mecanismo que gereencie esses dados. Esse processo precisa ponderar a *cache* criada para reduzir tempos de acesso a exames e a replicação desnecessária de configurações e suas cópias.
- A sincronização dos dados nas *bridges* pode ser realizada por meio de técnicas baseadas em eventos. Para isso, é necessário abstrair os conjuntos de informações referentes a um exame em um estudo. Já as configurações não precisam ser instantaneamente atualizadas, adequando-se também a essa técnica. Isso evita o uso de técnicas de força bruta, como o *polling*, o que aumenta a ociosidade da rede.

Com essas informações obtidas em mãos, é possível imaginar um mecanismo de integração e aumento de desempenho do aplicativo. Isto pode ser realizado com a utilização de meios que evitam a duplicação desnecessária de dados, sincronizam as informações entre os clientes e o servidor centralizador e a maior transparência do sistema ao seu usuário. No entanto, objetivos como esses não são fáceis de atingir, sendo necessárias algumas pequenas alterações no aplicativo para permitir essas melhorias.

4.4.1. *Comunicação e processamento de mensagens*

A comunicação entre a ferramenta CHIS e o Cyclops DCMServer pode ser realizada de duas formas: sinais de sistema e mensagens XML. Ambas as formas são mecanismos que permitem a mudança de parâmetros de *software* em tempo de execução. No entanto, cada uma dessas formas deve ser utilizada em casos específicos, como forma de manter coerência e um grau de padronização.

As mensagens XML devem ser utilizadas quando for necessário que uma alteração seja realizada imediatamente. Ou seja, quando a manipulação do processo atual não pode ser ignorada, como no processamento de arquivos grandes (ex: exames com modalidade XA). A partir da mensagem, o sistema então realiza a alteração necessária e re-inicializa o processo atual. Esse mecanismo também deve ser utilizado quando for esperada uma resposta do sistema ou em casos que os sinais de sistema não possam agir satisfatoriamente.

A utilização de sinais de sistema deve ser realizada quando o próprio CHIS possa realizar a alteração e apenas informar o aplicativo de que deve realizar determinada seqüência de operações. Esses passos são determinados pelo sinal de sistema escolhido, o qual deve ser interpretado para que o aplicativo saiba qual seqüência de operações a realizar. Um exemplo típico dessa alternativa são as mudanças de configurações. O aplicativo CHIS realiza as alterações necessárias e envia um sinal para que o DCMServer leia novamente suas configurações e mantenha-se em funcionamento. Outro exemplo trivial é a necessidade de envio de um sinal de *kill* para o sistema, o qual o avisa para seguir os passos anteriores a parada do processo.

4.4.2. *Alterações nas camadas de aplicativo existente*

O suporte do aplicativo a troca de mensagem em tempo de execução passa pelo seu recebimento, reconhecimento e tratamento. Para que esses processos possam ser corretamente executados e a fim de manter organizado o sistema, é necessária a acomodação de mais camadas com

esses fins. Isso só acontece quando o sistema estiver dividido em camadas funcionais. Com essa divisão lógica é possível reduzir o número de mudanças a serem realizadas e aumentar a reutilização desse código. A Figura 15 exibe a organização dos pacotes do Cyclops DCMServer. Na figura é possível visualizar as camadas adicionadas (em vermelho) e suas outras camadas funcionais. As camadas na horizontal relacionam-se diretamente com suas camadas inferiores e com as verticais. Já essas, apenas participam de relações, mas não possuem relação direta com outras camadas.

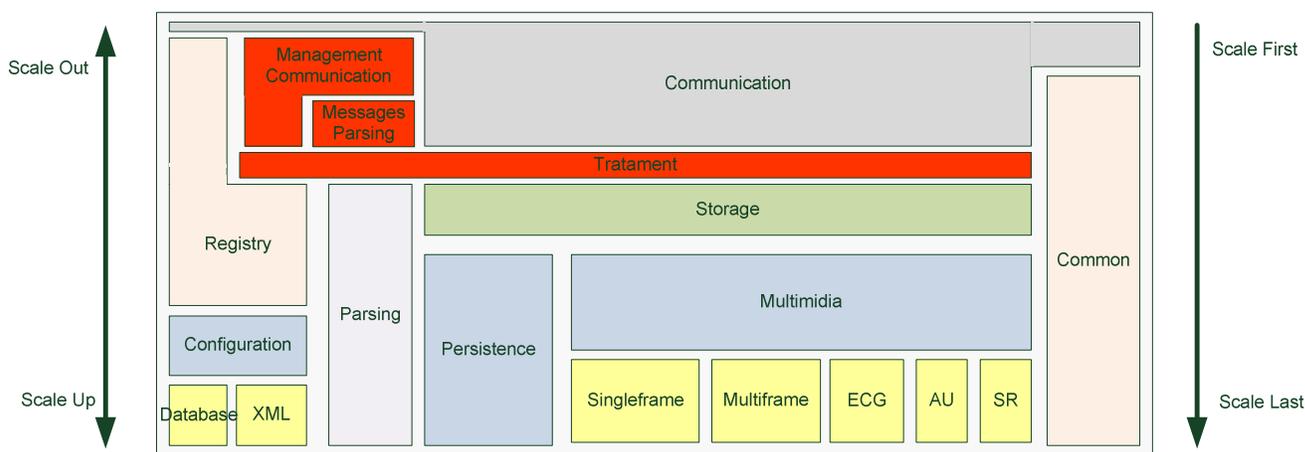


Figura 15 – Diagrama de camadas do Cyclops DCMServer

As camadas “*Management Communication*”, “*Messages Parsing*” e “*Tratament*” necessitam ser adicionadas para que o aplicativo suporte a troca de mensagens em tempo de execução. Respectivamente, elas são responsáveis pelo recebimento das mensagens, pelo seu reconhecimento e pela rotina de passos a serem executados. Essas características possibilitam que apenas a camada de tratamento seja modificada caso algum processo necessite ser modificado ou adicionado.

A adição destas camadas implica na adição de novas funcionalidades ao aplicativo. A nova camada de comunicação necessita de outra porta para comunicação com o *CHIS Client*, assim como exige a adição de novos atributos – como identificação de clientes e servidores – as configurações do serviço. Após o recebimento dessa mensagem, a camada de reconhecimento de

mensagens a interpreta e apenas passa sua identificação a camada de tratamento. Essa última é responsável pela execução dos procedimentos necessários e por isso possui associação com as demais camadas do sistema.

4.4.3. *Alterações nas configurações*

Além da adição de novas camadas ao sistema, algumas necessitam de modificações para que o sistema possa ser integrado. Entre essas alterações, as principais, são as alterações na leitura e no armazenamento das configurações. Essa alteração necessita ser feita para facilitar o processo de alteração e leitura de dados. Em função disso, as configurações passam de um arquivo XML para informações contidas em um banco de dados relacional.

No entanto, a utilização de um banco de dados traz consigo a dependência de mais um serviço de sistema, que possui períodos de manutenção. Essas desvantagens, porém, são compensadas pelas vantagens que um banco de dados proporciona. Entre essas vantagens estão: facilidade de escrita e leitura em comparação com arquivos em disco, integração simplificada com outros sistemas e a possibilidade de utilização de mecanismos de segurança com um custo menor.

A possibilidade de armazenamento remoto (centralização de todas as configurações) das bases de dados deve ser ignorada. Isso por que a necessidade de sincronização dos dados e seu grau de importância nos clientes exigem que essas configurações estejam sempre acessíveis de forma rápida. O serviço depende exclusivamente desses dados e por isso, é importante que exista um mecanismo de escape, para o caso de ocorrer algum erro de conexão com o banco de dados ou algum problema de integridade. Em função disso, um arquivo XML de backup deve ser mantido para reduzir esse problema, mesmo que necessite ser manualmente configurado. Em função disso, essa alternativa não precisa estar integrada ao serviço de gerenciamento remoto e deve conter, pelo menos, as configurações de acesso dos equipamentos e os dados para re-envio entre o servidores de ponte e o Portal de Telemedicina.

4.5. Integração com os Outros Aplicativos

Os primeiros passos para a integração de uma ferramenta ao CHIS são a análise das necessidades e o levantamento de requisitos. Isso deve ser realizado levando-se em consideração a natureza do aplicativo, frequência de utilização e a importância dos serviços realizados. Abaixo estão descritos os principais serviços, além do Cyclops DCMServer e do Portal de Telemedicina a serem integrados ao CHIS, assim como também estão especificados os principais requisitos de integração de cada sistema.

- **HL7Server.** O servidor de mensagens HL7 consiste em uma ferramenta que realiza troca de mensagens em formato XML. Devido a flexibilidade de suas mensagens, é possível realizar desde troca de informações de pacientes até envio de exames ao Portal de Telemedicina. Em função disso, esse sistema pode ser considerado crítico e deve manter-se operante 24hrs por dia. Além de um mecanismo de monitoramento de status, sua integração com o CHIS também deve ser realizada para permitir alteração em suas configurações.

Basicamente são necessárias duas mudanças para a integração da ferramenta ao CHIS. A primeira consiste em desenvolver o tratamento de sinais e transformar o aplicativo em um serviço de sistema. A partir disso é possível enviar sinais de sistema ao sistema. Além disso, é importante que as configurações do sistema também sejam armazenadas em um banco de dados relacional, nos mesmos moldes da ferramenta DCMServer. A modelagem do banco de dados deve ser independente do servidor DICOM e um arquivo de segurança também deve ser mantido para garantir maior integridade ao sistema.

- **CycClient.** A ferramenta cycClient consiste em um cliente que realiza o envio de arquivos DICOM ou XML, respectivamente, a um servidor DICOM ou ao HL7Server. Seu modo de funcionamento consiste na leitura de um arquivo de configuração em formato XML que contém informações a respeito do diretório em que estão armazenados os arquivos, o caminho do diretório em caso de sucesso na transmissão, falha ou no caso de erros no processo de reconhecimento e validação do arquivo. O Anexo 8.2 exibe o arquivo de configuração da ferramenta.

Atualmente a ferramenta já é executada como um serviço de sistema, com os principais *Unix signals* já sendo reconhecidos. Em função disso, a única grande alteração necessária é a necessidade de leitura das configurações de um banco de dados. Nesse aplicativo, esse processo é mais custoso e necessita de maiores cuidados. Isso devido a natureza de operação da ferramenta e de seu modo de operação atual. No momento, apenas é possível monitorar os arquivos em um único diretório, sendo que é necessária uma instância do aplicativo para cada diretório a ser monitorado. Já o modo de funcionamento da ferramenta é definido antes de sua execução, sendo passado como um parâmetro de funcionamento.

Como forma de resolver isso, pode-se utilizar um banco de dados relacional representando objetos com atributos referentes ao diretório a ser percorrido, o tipo de arquivos presentes e o modo de operação do sistema. Por meio dessa alteração, é possível realizar o monitoramento de diversos diretórios com uma mesma instância em execução. Assim, com a execução dessa modificação a ferramenta pode ser integrada ao CHIS, recebendo sinais de sistema para gerenciar seu fluxo de informações.

- **PingPong.** A ferramenta PingPong consiste em um aplicativo que realiza o monitoramento de cada *bridge* por meio da utilização de comunicação DICOM. Seu funcionamento consiste em enviar, periodicamente, mensagens do tipo C-ECHO para o servidor DICOM em execução em cada bridge. A lista de clientes está armazenado no mesmo banco de dados do Portal de Telemedicina, sendo os dados e os registros do sistema integrados a ferramenta web.

Em função destas características, esse é o aplicativo mais fácil de ser integrado ao CHIS. Esse processo apenas necessita do desenvolvimento do reconhecimento dos sinais de sistema. Esses serão utilizados para administrar o fluxo de informações da ferramenta.

5. *Conclusão*

A popularização da Rede Catarinense de Telemedicina e o número de hospitais e equipamentos a serem integrados ao sistema trazem um aumento significativo no fluxo de informações do sistema. Conseqüentemente, o crescimento no número de servidores de ponte torna a manutenção da rede custosa e impossibilita a descentralização do processo de administração de falhas. Esses fatos tornam clara a necessidade de uma nova solução que facilite a administração e o gerenciamento da rede e que permita a automatização de muitas dessas tarefas.

O estado atual da RCTM e as funcionalidades já desenvolvidas tornam claro que qualquer solução deve ser encaixada no sistema sem alterar as características de cada serviço. Em função disso, o conjunto de regras e mecanismo proposto segue as convenções já estabelecidas e evita ao máximo alterar as ferramentas existentes. A utilização de um *WebService*, mesmo que utilizando *sockets* para comunicação, mantém o sistema atualizado com as técnicas modernas de integração de sistemas e facilita sua integração com sistemas *web*.

A partir desses dados e dos problemas atuais, é possível afirmar que a utilização de uma ferramenta de gerenciamento facilitará o processo de administração do sistema e permita a expansão do número de pontos integrados. Além desse fato, a monitoração do sistema garante maior integridade e segurança às informações armazenadas. Conseqüentemente, a preocupação sobre a confiabilidade dos dados e do sistema de médicos, pacientes e do próprio Governo do Estado é reduzida, o que facilita o processo de conscientização e aumenta a utilização do sistema.

O encaixe dessa ferramenta a arquitetura atual cobre uma lacuna existente e torna o Sistema Catarinense de Telemedicina robusto e seguro a transmissão e armazenamento de dados e os servidores geograficamente distribuídos. Essa segurança permite a ampliação do sistema a outras áreas e, conseqüentemente, auxilia na melhoria do sistema público de saúde do estado.

5.1. Trabalhos Futuros

A melhoria do conjunto de regras e mecanismos propostos passa pelo desenvolvimento de alguns tópicos pontuais. Entre eles, pode-se citar:

Desenvolvimento do sistema de autogerenciamento dos clientes baseados em técnicas de inteligência artificial. Por meio de técnicas como Redes Neurais é possível melhorar a tomada de decisão do momento de execução de uma manutenção crítica de sistema. Dados como o fluxo diário e momentâneo de informações, ociosidade do sistema e grau de utilização do servidor podem ser utilizados para garantir maior integridade do sistema de autogerenciamento.

Pesquisa de meios de integração de sistemas a gerenciar em tempo de execução. Para redução de custos e para garantir maior facilidade no processo de integração de novos serviços pode-se buscar soluções que realizem esse processo em tempo de execução. Tal fato reduz o tempo gasto com o desenvolvimento de código-fonte para integração do novo aplicativo.

6. *Referências Bibliográficas*

Alonso, G., F. Casati, H Kuno, e V. Machiraju. *Web Service - Concepts, Architectures and Applications*. Berlin: Springer Verlag, 2003.

Balota, M. R. “Estudo da Rede de Computadores e Garantia de Qualidade de Serviços para Aplicações em Telemedicina.” Trabalho de Conclusão de Curso, Jundiaí, 2005.

Bashshur, R. L. “Telemedicine and Health Care.” *Telemedicina J. and E-Health*, 2002: 5-12.

Becker, Shawn H., e Ronald L. Arenson. “Cost and Benefits of Picture Archiving and Communication Systems.” *J Am Med Informatics Assoc.*, 1994: 1:361-371.

Cao, F, H. K. Huang, e X. Q. Zhou. “Medical image security in a HIPAA mandated PACS environment.” *Computerized Medical Imaging and Graphics*, 2003: 185-196.

Cartagena, Miguel. “Um Sistema de Monitoração e Gerência da Rede Catarinense de Telemedicina.” *Monografia*. Florianópolis: Universidade Federal de Santa Catarina, 2007.

Commitee, ACR-NEMA. “Digital Imaging and communication in medicine (DICOM).” v. 3.0. 1993.

Dellani, Paulo. “Desenvolvimento de um servidor de imagens médicas digitais no formato DICOM.” *Dissertação de Mestrado*. Florianópolis: Universidade Federal de Santa Catarina, 2001.

Hasselbring, Wilhelm. “Information System Integration.” *Communications of the ACM*, 2000: 33-38.

Luz Jr., Antonio da, Elisa Manfrin de Araújo, Eros Comunello, e Aldo von Wangenheim. “Gerenciamento Remoto de PACS: Uma Proposta de Extensão ao DICOM.” *Congresso Brasileiro de Informática na Saúde*. 2006.

Macedo, D. D. J. de, H. G. W. Perantunes, Rafael Andrade, A. v. Wangenheim, e M. A. R. Dantas. “Asynchronous Data Replication: A National Integration Strategy for Databases on Telemedicine Network.” *21st IEEE International Symposium on Computer-Based Medical Systems CBMS '08*, 17-19 de Junho de 2008: 638-643.

Maia, R. S., A. v. Wangenheim, e L. F. Nobre. “A Statewide Telemedicine Network for Public Health in Brazil.” *19th IEEE International Symposium on Computer-Based Medical Systems CBMS '06*, 2006: 495-500.

Maia, Rafael Simon. “Um sistema de telemedicina de baixo custo em larga escala.” *Dissertação de Mestrado*. Florianópolis: Universidade Federal de Santa Catarina, 2006.

Martini, Richard Augusto Schafer De, Douglas Dylon Macedo, Rafael Andrade, e Aldo von Wangenheim. “Um Sistema de Gerenciamento Remoto de Rede PACS.” *Congresso Brasileiro de Informática em Saúde*. 2008.

Mijares, M. T. “Telemedicina in Ecuador: Failure or a Learning Experience.” *Proc. 6th International Workshop Enterprise Networking and Computing in Healthcare Industry, VECPAR*. 2004. 41-43.

Pressman, Roger S. *Engenharia de Software*. Sexta. McGraw-Hill, 2006.

Ribeiro, Leonardo Andrade, Paulo Roberto Dellani, Aldo von Wangenheim, Michael M. Richter, Kerstin Maximini, e Eros Comunello. “CyclopsDistMedDB. - A Transparent Gateway for Distributed Medical Data Access in DICOM Format.” *15th IEEE International Symposium on Computer-Based Medical Systems CBMS 2002*, 2002.

Tackley, Roger M. “The National Programme for Information Technology in the NHS, Anaesthesia & intensive care medicine.” *Intensive Care / Informatics*, 1 de Dezembro de 2004: 400-401.

Urtiga, Keylla Sá, Luiz A. C. Louzada, e Carmen Lúcia B. Costa. “Telemedicina: uma visão geral do estado da arte.” *Anais do CBIS*. 2004.

Wallauer, Jader, Douglas D. J. de Macedo, Rafael Andrade, and Aldo Wangenheim. "Building a National Telemedicine Network." *IT Professional*, March-April 2008.: vol.10, no.2, pp.12-17.

Zachman, John. “A framework for information systems architecture.” *IBM Systems Journal* , 1987: 276-292.

7. *Apêndice A – Requisitos do sistema*

Após a análise dos dados obtidos com o estudo do cenário e juntamente ao levantamento de soluções existentes é possível realizar o levantamento dos requisitos do sistema. Com essa etapa é possível validar e documentar o processo de desenvolvimento de software. Abaixo seguem figuras com representações dos principais requisitos levantados.

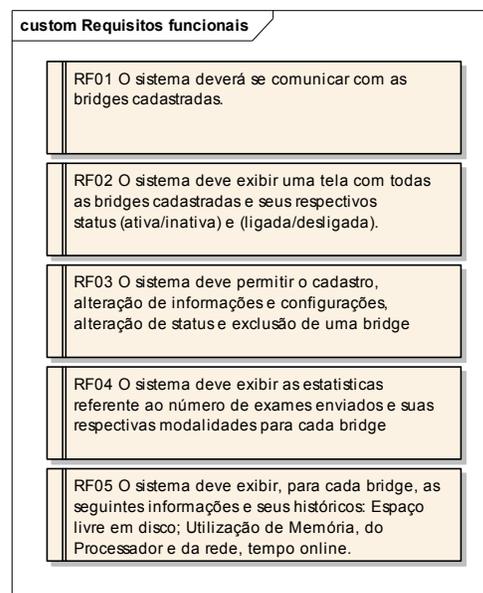


Figura 16 – Requisitos Funcionais básicos

Os requisitos representados pela Figura 16 referem-se às funcionalidades mais básicas do aplicativo. Eles contêm apenas informações referentes ao cadastro das máquinas clientes e informações triviais referentes a esse processo.

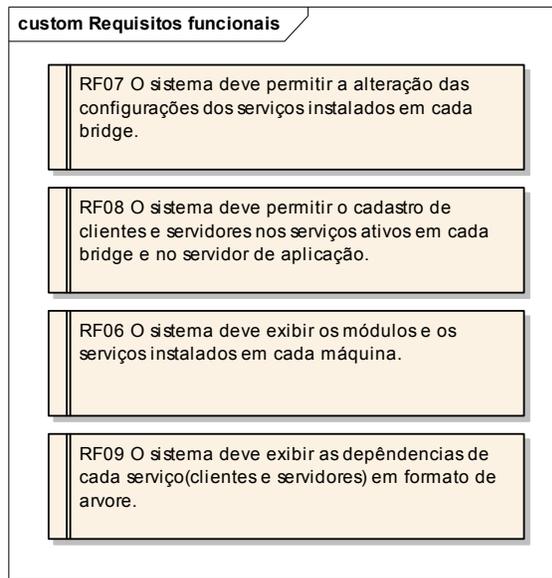


Figura 17 – Requisitos Funcionais de configuração de serviços

A Figura 17 representa os requisitos de configuração de serviços em cada cliente. Esses requisitos representam a inclusão e alteração dos aplicativos instalados no servidor, bem como permite visualização de módulos e serviços configurados na máquina. Já a Figura 18 refere-se aos requisitos que representam a execução de manutenções no sistema.

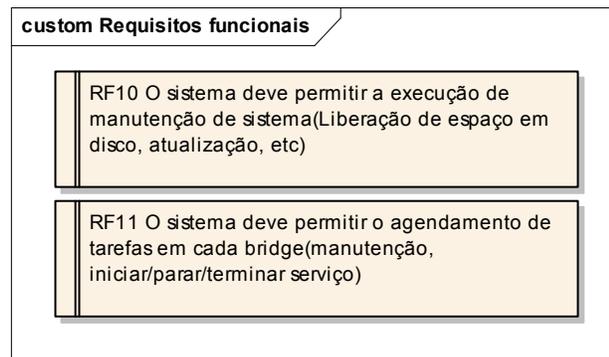


Figura 18 – Requisitos Funcionais de manutenção

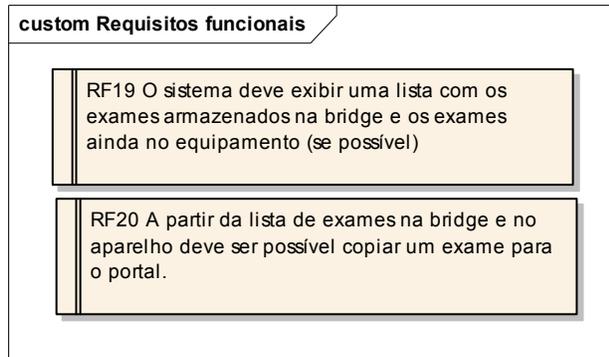


Figura 19 – Requisitos Funcionais de visualização de status de exames

A Figura 19 representa os requisitos referentes a visualização dos exames armazenados em cada servidor. Por meio do cliente deve ser possível realizar cópias de exame do cliente para o Portal de Telemedicina e, quando possível, do equipamento para o cliente. Já os requisitos de atualização e manutenção do sistema estão representados na Figura 20.

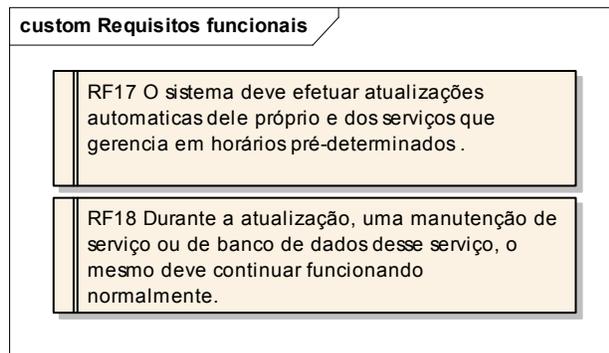


Figura 20 – Requisitos Funcionais de atualizações automáticas

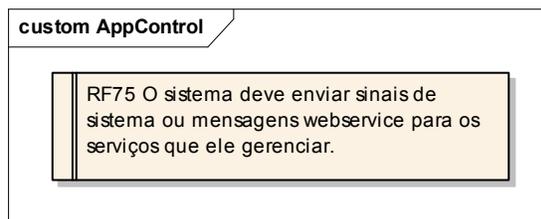


Figura 21 – Requisito Funcional de Comunicação entre aplicações

A Figura 21 representa os mecanismos de comunicação entre o aplicativo de gerenciamento e cada serviço. Já na Figura 22, são representados os requisitos referentes às cópias de segurança das informações do sistema e dos dados persistentes a cada serviço. Os requisitos referentes à comunicação entre clientes e servidores e clientes e serviços são representados na Figura 23.

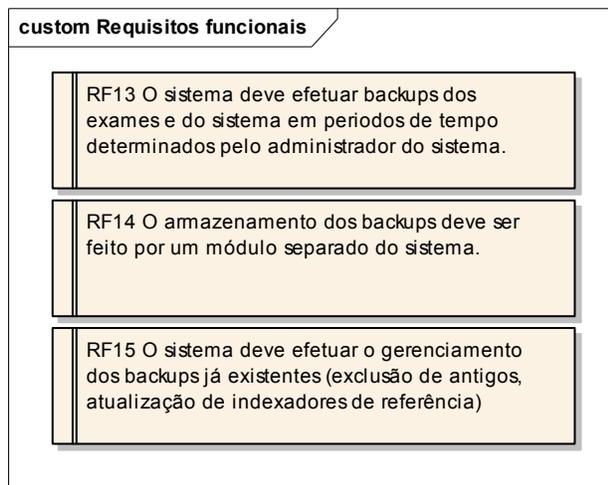


Figura 22 – Requisitos Funcionais de cópias de segurança

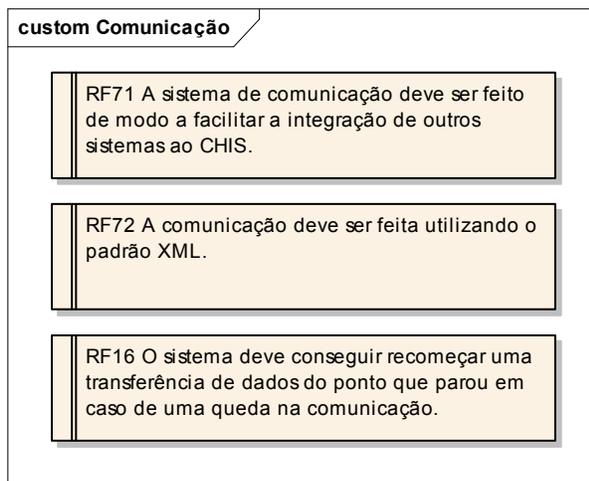


Figura 23 – Requisitos Funcionais de Comunicação

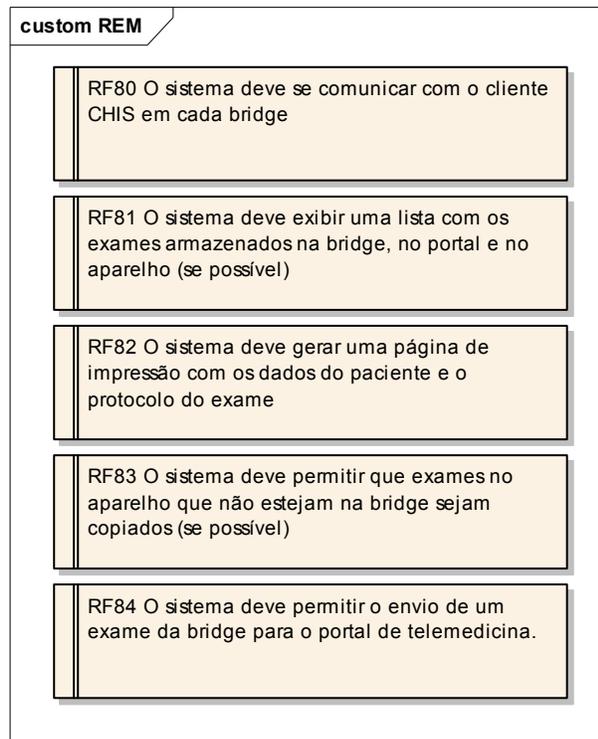


Figura 24 – Requisitos Módulo de Regulação de Exames

Na Figura 24 são descritos os requisitos da aplicação REMS (Remote Exam Management System). Por meio desse aplicativo é possível visualizar o status de cada exame, antes mesmo de ser publicado no Portal de Telemedicina, no setor em que foi realizado o exame. Além da visualização dos exames também é possível realizar a geração e impressão do protocolo referente a cada exame.

Apêndice B – Artigo Relacionado
Um Modelo de Gerenciamento Remoto de PACS Geograficamente Distribuído

Richard A. S. de Martini¹, Douglas D. J. de Macedo²,
Rafael Andrade³, Aldo von Wangenheim⁴

^{1,2,4} Departamento de Informática e Estatística

^{3,4} Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento

Universidade Federal de Santa Catarina (UFSC), Brasil

Introdução

A evolução dos ambientes de rede, servidores e o desenvolvimento do padrão DICOM possibilitaram nos últimos anos a difusão de ambientes PACS (*Picture Archiving and Communication System*). Estes são definidos como sistemas integrados de comunicação e armazenamento de imagens médicas, geralmente em formato DICOM [1].

O constante desenvolvimento dos sistemas computacionais e dos equipamentos médico-hospitalares culminou em um novo paradigma, os PACS distribuídos [2]. Pensando neste tipo de arquitetura, a Universidade Federal de Santa Catarina (UFSC), em conjunto com a Secretaria de Estado da Saúde de Santa Catarina desenvolveu a Rede Catarinense de Telemedicina (RCTM). Esta rede permitiu até julho de 2008, a inclusão

de mais de 120 mil exames de diversas modalidades realizados pelo estado, armazenados em um único servidor central [3].

A utilização de PACS geograficamente distribuídos potencializa os problemas de administração encontrados em sistemas simples. A distância geográfica entre a aquisição das imagens e seu armazenamento, muitas vezes, obriga que as informações sejam transmitidas pela Internet. Isso torna o sistema vulnerável a oscilações, o que pode deixar pontos inacessíveis por tempo indeterminado. Essas características, aliadas ao número de pacientes e as equipes médicas envolvidas necessitam de tratamentos diferenciados para que as informações sejam corretamente manipuladas.

A utilização da Internet para comunicação entre os pontos geograficamente distribuídos traz problemas típicos de outros sistemas ao PACS, como por exemplo, fraca

tolerância a falhas e vários problemas com interceptação de dados. Nestes casos, a utilização de criptografia e de outros mecanismos de segurança torna-se extremamente necessário, devido ao risco de ataques e de interceptação de dados.

Metodologia

Os exames armazenados no banco da RCTM podem ser divididos em duas classes – DICOM e não DICOM. Os exames DICOM estão concentrados, em sua grande maioria, na região metropolitana de Florianópolis. Essa área concentra quatro hospitais SUS, responsáveis por quase 1.400 exames por mês. A ferramenta CHIS (Cyclops Healthcare Integration System) é responsável pela administração e manutenção de cada *bridge*. O software caracteriza-se por utilizar a arquitetura cliente-servidor - um servidor central e clientes em cada *bridge*. A utilização da tecnologia *web-service* facilita a integração de novos serviços e torna as mensagens trocadas simples e robustas.

O software pode ser dividido em duas metades. O CHIS *Server* é responsável pelo envio das mensagens a cada cliente. O CHIS *Client* recebe do servidor as operações que devem ser realizadas nos softwares ou no sistema e envia sinais ao DICOM *Server* na *bridge*. As mensagens trocadas entre o clien-

te e o servidor seguem o padrão XML [4]. Elas possuem a identificação do cliente, do servidor, o tipo e os parâmetros necessários para a operação. Os tipos de mensagens são operações que devem ser executadas pelo cliente ou estatísticas do sistema.

apresenta os pontos de comunicação geograficamente distribuídos. Os pontos em roxo representam a rede pública e os em amarelo, a rede privada.



Figura 1 – Rede PACS na grande Florianópolis.

Em cada um desses hospitais existe um servidor, denominado *bridge*. Esse é responsável pelo armazenamento temporário e o envio das imagens recebidas para o DICOM *Server*. Esse é o encarregado pela centralização e armazenamento dos exames realizados em todas as instituições interligadas a rede. A ferramenta CHIS (Cyclops Healthcare Integration System) é responsável pela administração e manutenção de cada *bridge*. O

software caracteriza-se por utilizar a arquitetura cliente-servidor - um servidor central e clientes em cada *bridge*. A utilização da tecnologia *web-service* facilita a integração de novos serviços e torna as mensagens trocadas simples e robustas.

O software pode ser dividido em duas metades. O *CHIS Server* é responsável pelo envio das mensagens a cada cliente. O *CHIS Client* recebe do servidor as operações que devem ser realizadas nos softwares ou no sistema e envia sinais ao *DICOM Server* na *bridge*. As mensagens trocadas entre o cliente e o servidor seguem o padrão XML [4]. Elas possuem a identificação do cliente, do servidor, o tipo e os parâmetros necessários para a operação. Os tipos de mensagens são operações que devem ser executadas pelo cliente ou estatísticas do sistema.

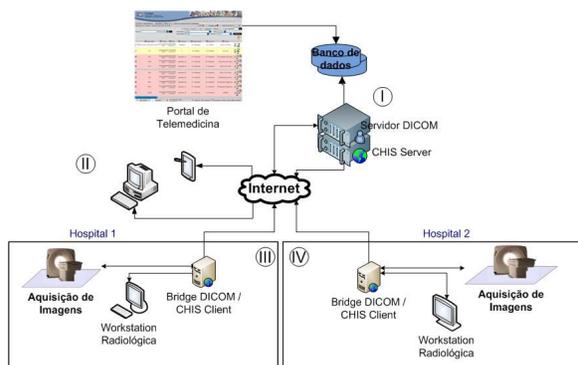


Figura 2 – Modelo de funcionamento do PACS na RCTM.

A Figura 2 descreve o funcionamento da rede PACS integrada a RCTM. Em I estão definidos os servidores de armazenamento e comunicação de dados. II são os visualizadores DICOM pessoais e *workstations* portáteis, que acessam os servidores definidos em I. A aquisição das imagens é feita nos hospitais representados por III e IV.

As interfaces *web* são responsáveis por criar as mensagens que o servidor envia aos clientes. O acesso a este conjunto de páginas está disponível no Portal de Telemedicina, seguindo todos os seus mecanismos e padrões de acesso. Após a solicitação de uma operação, uma mensagem é gerada e enviada ao *CHIS Server* que repassa a mensagem ao cliente determinado. O conjunto de ferramentas disponível é suficiente para resolver os problemas mais usuais encontrados. Entre essas ferramentas, pode-se citar:

- alterações de configuração dos DICOM Servers. Possibilidade de visualizar, alterar ou adicionar clientes e servidores. Opções avançadas como tipo de compressão de imagens, configurações de porta e tipo de armazenamento também podem ser alteradas por esse meio e
- visualização de status e agendamento de manutenções de sistema. Essa ferramenta permite verificar a utilização de memória, espaço em disco e históricos relacionados

em cada *bridge*. Também é possível agendar manutenções ou desligar o sistema.

Resultados Esperados

Com a realização desta proposta a administração e a manutenção das *bridges* são feitas pela ferramenta CHIS. Essa característica permitirá descentralizar e adequar a manutenção de cada um desses servidores ao Portal de Telemedicina. A arquitetura utilizada permite o desenvolvimento de ferramentas de manutenção automática, o que reduz o tempo gasto com operações simples. Através do desenvolvimento de *features* desse tipo será possível automatizar manutenções de sistema, atualizações e backups.

A automatização dos clientes também permite o desenvolvimento de um sistema inteligente de envio de arquivos. Ferramentas assim reduzem problemas encontrados no envio das imagens com velocidade e instabilidade de rede. Isso proporciona um aumento na produtividade do sistema e diminui a possibilidade de gargalos de comunicação.

Discussão e Conclusões

Uma das premissas que foram levadas em consideração no desenvolvimento desse

modelo é a possibilidade de se adicionar novas features ao software ou novos serviços a serem gerenciados. Partindo desse ponto, visualiza-se a extensão do modelo para a administração e gerenciamento de alguns requisitos de sistema, como backups e atualizações.

A comunicação entre a aplicação e os serviços gerenciados pode utilizar o padrão webservice ou Unix Signals. Isso facilita a integração e permite a utilização de modelos híbridos.

Referências

[1] Cao, F.; Huang, H. K.; Zhou, X. Q., "Medical image security in a HIPAA mandated PACS environment" *Computerized Medical Imaging and Graphics*, pp 185-196, 2003, Elsevier.

[2] Macedo, D. D. J. de; Perantunes, H. G. W.; Andrade, R.; Wangenheim, A. v.; Dantas, M. A. R. Dantas, "Asynchronous Data Replication: A National Integration Strategy for Databases on Telemedicine Network," *21st IEEE International Symposium on Computer-Based Medical Systems, 2008. CBMS '08*. pp.638-643, 17-19 June 2008.

[3] Wallauer, J.; Macedo, Douglas D. J. de;

Andrade, R.; von Wangenheim, A., "Building a National Telemedicine Network," *IT Professional* , vol.10, no.2, pp.12-17, March-April 2008.

[4] Alonso G, Casati F, Kuno H, Machiraju V. *Web Service – Concepts, Architectures and Applications*, Berlin: Springer Verlag; 2003.

Contato

Richard Augusto Schafer De Martini (rmartini@inf.ufsc.br) – Universidade Federal de Santa Catarina – Campus Universitário, s/n - Florianópolis/SC – Brasil - Tel.: (48) 3721-9166 - <http://www.telemedicina.ufsc.br>

8. *Anexos*

8.1. Unix Signals

SIGHUP	1	Exit	Hangup
SIGINT	2	Exit	Interrupt
SIGQUIT	3	Core	Quit
SIGILL	4	Core	Illegal Instruction
SIGTRAP	5	Core	Trace/Breakpoint Trap
SIGABRT	6	Core	Abort
SIGEMT	7	Core	Emulation Trap
SIGFPE	8	Core	Arithmetic Exception
SIGKILL	9	Exit	Killed
SIGBUS	10	Core	Bus Error
SIGSEGV	11	Core	Segmentation Fault
SIGSYS	12	Core	Bad System Call
SIGPIPE	13	Exit	Broken Pipe
SIGALRM	14	Exit	Alarm Clock
SIGTERM	15	Exit	Terminated
SIGUSR1	16	Exit	User Signal 1
SIGUSR2	17	Exit	User Signal 2
SIGCHLD	18	Ignore	Child Status
SIGPWR	19	Ignore	Power Fail/Restart
SIGWINCH	20	Ignore	Window Size Change
SIGURG	21	Ignore	Urgent Socket Condition
SIGPOLL	22	Ignore	Socket I/O Possible
SIGSTOP	23	Stop	Stopped (signal)
SIGTSTP	24	Stop	Stopped (user)
SIGCONT	25	Ignore	Continued

SIGTTIN	26	Stop	Stopped (tty input)
SIGTTOU	27	Stop	Stopped (tty output)
SIGVTALRM	28	Exit	Virtual Timer Expired
SIGPROF	29	Exit	Profiling Timer Expired
SIGXCPU	30	Core	CPU time limit exceeded
SIGXFSZ	31	Core	File size limit exceeded
SIGWAITING	32	Ignore	All LWPs blocked
SIGLWP	33	Ignore	Virtual Interprocessor Interrupt for Threads Library
SIGAIO	34	Ignore	Asynchronous I/O

* Fonte: <http://www.tech-faq.com/lang/pt/unix-signals.shtml>

8.2. Arquivo de configuração do cycClient

```
<parameters>
```

```
<dirMode
```

```
  rootDir="/diretorioRaiz"
```

```
  watchedDir="watched"
```

```
  failedDir="failed"
```

```
  successDir="success"
```

```
  invalidDir="invalid"
```

```
>
```

```
<AETitles
```

```
  localAE="Identificação do cliente">
```

```
<remoteAE
```

```
  AETitle="Identificação do servidor"
```

```
  address="192.168.0.1"
```

```
  port="80"
```

```
>
```

```
</AETitles>
```

```
<server host="192.168.0.1" port="22"/>
```

```
<ssl enabled="TRUE" certificate="☺" key_file="" dh_file="" />
```

```
</parameters>
```