UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Leonardo Andrade Ribeiro

UM PORTAL DE BANCO DE IMAGENS MÉDICAS DISTRIBUÍDO USANDO CORBA PARA INTEGRAÇÃO DE SERVIÇOS DE TELERADIOLOGIA

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr.rer.nat Aldo von Wangenheim

Florianópolis, fevereiro de 2002

UM PORTAL DE BANCO DE IMAGENS MÉDICAS DISTRIBUÍDO USANDO CORBA PARA INTEGRAÇÃO DE SERVIÇOS DE TELERADIOLOGIA

Leonardo Andrade Ribeiro

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciências da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Raul S. Wazlawick (coordenador)

Banca Examinadora

_____

Aldo von Wangenheim

_____

Antônio Augusto Medeiros Fröhlich

_____

Marino Bianchin

À minha família.

AGRADECIMENTOS

À minha família.

Aos colegas e amigos que contribuíram ao seu modo para a realização deste trabalho.

Sempre serei grato!

SUMÁRIO

RESUMO

A área do diagnóstico por imagem é um dos campos da medicina mais propensos a uso da Telemedicina, porque normalmente não existe a obrigação do contato direto com o paciente pelo radiologista responsável durante a elaboração do diagnóstico. A carência de especialistas em locais distantes de grandes centros urbanos faz da Telemedicina uma importante ferramenta para melhorar os serviços de atenção à saúde. Neste trabalho é apresentado um modelo baseado em uma abordagem federada chamo Portal de Teleradiologia, para a integração de bancos de imagens médicas DICOM distribuído geograficamente. O objetivo é prover uma visão única e transparente dos dados compartilhados sem sacrificar a autonomia dos sistemas integrantes ou interferir com operações locais. Um requisito obrigatório para sistemas desta natureza é a garantia de um contexto seguro para a execução das operações. O modelo proposto prevê também a identificação única de pacientes entre as diversas bases de dados através de técnicas de relacionamento de registros. Para a implementação do protótipo do sistema foi utilizada a tecnologia de objetos distribuídos através da arquitetura CORBA.

ABSTRACT

The image diagnosis area is the most propense medical field for Telemedicine applications, because it does not require a direct contact between the patient and the responsible radiologist during the report build. The lack of specialists in places away from urban centers makes the Telemedicine an important tool for improvement of healthcare services. In this work is presented a model, based in a federated approach called Teleradiology Gateway, designed to promote the integration of distributed DICOM medical record databases over wide areas. The main objective is to provide a single and transparent view over the shared data while preserving the autonomy of component systems and continued of existing applications. A mandatory requirement of this kind of systems is the assurance of a secure context. The proposed model provides the patient unique identification though record linkage methods. In the prototype developed is used distributed object technology though CORBA architecture.

LISTA DE FIGURAS

1. INTRODUÇÃO

A necessidade de armazenar e dispor exames médicos contendo imagens e sinais biológicos, tomografia computadorizada, ressonância magnética e eletrocardiografia, está cada vez mais presente nos ambientes radiológicos. A chamada "radiologia sem filme" representa uma solução para melhorar a acessibilidade e qualidade das informações radiológicas, ao mesmo tempo em que propicia uma sensível redução nos custos.

Neste cenário o padrão DICOM (*Digital Imaging and Communication in Medicine*), encontra-se consolidado como padrão de fato mundial para a transmissão, arquivamento e formatação de imagens radiológicas digitais. A possibilidade de utilizar aparelhos de imagem, impressoras, *scanners*, câmeras digitais, bem como uma grande variedade de softwares de diversos propósitos e fornecedores, conectados por uma rede de baixo custo, impulsionou o crescimento de sistemas PACS (*Picture Archiving and Comunnication Systems*). Além disso, o padrão DICOM facilita a interface entre o sistema de informação do ambiente radiológico, RIS (*Radiological Information System*) e o restante do sistema hospitalar, HIS (*Hospital Information System*).

O advento da Internet como um meio de abrangência global para o intercâmbio de dados, permite a hospitais e clínicas expandir seu escopo de atuação para além de suas fronteiras físicas através da Telemedicina. A área do diagnóstico por imagem é um dos campos da medicina mais propensos a uso da Telemedicina, porque normalmente não existe a obrigação do contato direto com o paciente pelo radiologista responsável durante a elaboração do diagnóstico. Clínicos podem acessar exames de diferentes bancos de imagens a partir de uma estação de visualização radiológica em seu consultório ou até no computador de sua residência.

1.1 O Projeto Cyclops

O projeto Cyclops é um projeto binacional de pesquisa de longo prazo iniciado pelos professores Dr.rer. nat. Aldo von Wangenheim e Dr.Michael M.Richter na universidade de Kaiserslautern, Alemanha, em 1992. Tem como objetivo o desenvolvimento e a transferência de novos métodos, técnicas e ferramentas no campo da Análise de Imagens Médicas através da utilização de técnicas de Inteligência Artificial, Visão Computacional e Telemedicina.

Neste contexto, a cooperação com parceiros médicos e industriais foi iniciada em 1993. Hoje o Projeto se encontra em sua Fase II, estando focado na cooperação para o desenvolvimento de aplicações que possam ser de utilidade prática clínica e hospitais.

Este consórcio pretende alcançar as metas do Projeto através da cooperação entre os parceiros do Brasil e da Alemanha, com desempenhando tarefas em áreas complementares. O referido consórcio internacional de pesquisa e desenvolvimento é composto por Universidades, parceiros industriais da área de softwares, parceiros médicos e empresas produtoras de equipamentos médicos radiológicos de ambos os países.

1.2 Motivação

O padrão DICOM é um padrão mundial de fato para formatação e transmissão de imagens médicas digitais. O suporte à transmissão de imagens sobre a pilha de protocolos TCP/IP foi sem dúvida um dos grandes responsáveis pela disseminação do DICOM e sua conseqüente presença ubíqua em ambientes PACS (Picture Archiving and Comunnications Systems). Enquanto que, dentro do ambiente de uma instituição de saúde, o padrão DICOM tem como principal vantagem a eliminação da necessidade de impressão de exames em filme, seus benefícios podem ser ainda maiores quando seu escopo é expandido para o contexto da Internet. Entre as possíveis vantagens tem-se o melhor acompanhamento do histórico de saúde de um paciente, redução da duplicação de exames, suporte para segunda opinião além da agilização do acesso a dados críticos durante emergências.

Entretanto, apesar da uniformidade de modelo de dados proporcionada pelo DICOM e seu suporte à rede de comunicações, a utilização da infra-estrutura da Internet para compartilhamento de imagens médicas ainda é bastante aquém de seu potencial. Diversos fatores contribuem para o confinamento do DICOM no ambiente Intranet de hospitais e clínicas. Dentre estes motivos pode-se destacar três como preponderantes: a ausência de diretrizes claras que regulem a transmissão de exames pela Internet para que preceitos básicos da ética médica e privacidade individual sejam resguardados; características do padrão DICOM que dificultam o desenvolvimento de um ambiente colaborativo; dificuldades intrínsecas à tarefa de integrar bases de dados distribuídas e autônomas. Enquanto que o primeiro motivo vem sendo minimizado com o crescente posicionamento favorável à prática da Telemedicina por parte das Associações Médicas

em diversos países do mundo, os dois últimos ainda possuem diversas questões em aberto a serem tratadas.

1.3 Objetivos

O objetivo deste trabalho é desenvolver um modelo de uma arquitetura federada, para a integração de bancos de imagens médicas digitais em conformidade com o padrão DICOM 3.0. O sistema proposto deve prover uma visão única e transparente para os clientes e ao mesmo tempo manter a autonomia e independência dos sistemas integrantes.

1.4 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Avaliação do padrão DICOM em um sistema federado e propor extensões, caso necessário;
- Criar um modelo para integração transparente de servidores DICOM pré-existentes à arquitetura federada;
- Definição de um modelo de segurança adequado à arquitetura proposta baseado em critérios de avaliação rígidos;
- Definição de protocolos de comunicação flexíveis para busca e entrega de imagens médicas entre clientes e a arquitetura federada;
- Pesquisa e avaliação de métodos de avaliação de equivalência semântica de registros;
- Implementar um protótipo da arquitetura federada para ser usado foi servir como plataforma básica para testes, validações e refinamentos sobre a pesquisa realizada.

2. CONCEITOS EM SISTEMAS DE BANCO DE DADOS FEDERADOS

Um banco de dados é uma coleção de dados relacionados representando aspectos do mundo real aspecto e refletindo as mudanças sobre o mesmo. Esta coleção de dados é logicamente coerente com um significado e um propósito, definidos durante seu projeto e seguidos durante sua construção e utilização. Para manter esta uniformidade estrutural e semântica em um banco de dados utilizado por aplicações independentes e que não obedecem a um padrão pré-estabelecido, é necessária uma aplicação que controle a utilização deste banco de dados.

De acordo com [LM90], sistemas de banco de dados representaram uma solução para o problema causado pelo acesso compartilhado a arquivos heterogêneos e criados por múltiplas aplicações autônomas em um ambiente centralizado. Em outras palavras, um sistema gerenciador de banco de dados (SGBD – acrônimo do termo em inglês *database management system*) é uma coleção de programas que facilita o processo de definição, construção e manipulação de banco de dados utilizados por várias aplicações.

Seguindo ainda o raciocínio [LM90], como conseqüência natural do êxito da idéia de um sistema gerenciando uma base de dados em comum e sua disseminação entre organizações, surgiu a necessidade de acesso a múltiplos SGBDs autônomos. As técnicas e métodos desenvolvidos para sistemas de banco de dados distribuídos, porém gerenciados por um mesmo SGBD, não tratavam a questão crucial neste cenário que é a autonomia entre os SGBDs a serem integrados. Com isso houve a necessidade de uma nova abordagem para tratar o desafio de integrar sistemas de banco de dados autônomos e possivelmente heterogêneos chamados sistemas de banco de dados federados (SBDF). Os métodos empregados nesta abordagem podem ser naturalmente utilizados, com um nível variável de adaptação, para integrar servidores DICOM autônomos, que é o objeto de estudo deste trabalho.

O objetivo deste capítulo é fornecer conceitos básicos a respeito da teoria sobre sistemas de banco de dados federados. A contextualização das técnicas descritas neste capítulo com o problema da integração de banco de imagens DICOM será feita no capítulo 6. A estrutura do restante deste capítulo é a seguinte: Na seção 2.1 serão apresentados alguns conceitos básicos sobre banco de dados que são pertinentes para as discussões sobre banco de dados federados. A seção 2.2 apresentará as três dimensões

ortogonais - distribuição, heterogeneidade e autonomia – que norteiam a classificação de sistemas de banco de dados distribuídos. Na seção 2.3 é apresentada a taxonomia utilizada na literatura na área de banco de dados distribuídos seguindo as dimensões apresentadas na seção 2.2. Por último a seção 2.4 apresenta a arquitetura de referência para banco de dados federados que será adotada na modelagem do *framework* para integração de bancos de imagens DICOM.

2.1 Conceitos gerais sobre sistemas de banco de dados

Alguns conceitos básicos sobre sistemas de banco de dados são extremamente pertinentes para a discussão de sistemas federados. A seguir apresentaremos cada um destes conceitos.

2.1.1 Modelo de dados

Uma característica fundamental da teoria de banco de dados é a abstração de dados, onde detalhes do armazenamento de dados, que não são necessários para a maioria dos usuários, são mantidos ocultos. Um modelo de dados é a principal ferramenta para prover esta abstração. Um modelo de dados é um conjunto de conceitos que podem ser usados para descrever a estrutura de um banco de dados. A estrutura de um banco de dados é definida pelo tipo de dados, relacionamentos, e restrições que devem existir para os dados. A maioria dos modelos de dados também inclui um conjunto de operações básicas para especificar consultas e atualizações no banco de dados. É comum também incluir conceitos no modelo de dados para especificar um determinado comportamento; isto se refere a um conjunto de válidas operações definidas pelo usuário que são permitidas no banco de dados em adição às operações básicas do modelo de dados. Por exemplo podemos ter a operação CAL_ORCAMENTO que pode ser aplicada ao objeto PROJETO.

Vários modelos de dados têm sido propostos e utilizados ao longo dos anos. Eles podem classificados de acordo com o nível de abstração dos detalhes de armazenamento dos dados. Modelos de dados de alto nível são mais próximos da maneira como os usuários finais percebem os dados, enquanto que os de baixo nível descrevem como os dados são organizados, gravados e acessados no dispositivo de armazenamento. Entre os exemplos de modelos de dados de alto nível estão o popular modelo Entidade-

Relacionamento e o modelo orientado a objetos, enquanto que a estrutura de dados como árvores B é bastante utilizada em modelo de dados de baixo nível.

Alguns modelos de dados adotam uma abordagem híbrida. Eles provêem conceitos inteligíveis para usuários finais ao mesmo tempo em que evidenciam alguns aspectos dos métodos de armazenamento, podendo ser implementados de uma maneira mais direta. O modelo relacional, o mais comum atualmente, é um exemplo desta classe de modelos de dados, juntamente com o hierárquico e o de rede.

São cada vez mais comuns sistemas de banco de dados que utilizam o chamado modelo de dados objeto-relacional. Este modelo híbrido visa beneficiar-se do estabelecimento do paradigma da orientação a objetos no desenvolvimento de aplicações e do modelo de dados relacional em sistemas de banco de dados. Praticamente todos principais sistemas de banco de dados (*Oracle*, DB2, *SQL-Server*) possuem algumas funcionalidades do modelo objeto-relacional.

Alguns modelos de dados são projetados especialmente para serem utilizados como modelo dados comum durante a integração de fontes de dados de dados heterogêneas. O modelo de dados orientado a objeto EXPRESS, que faz parte do padrão ISO *Stardard for Exchange of Product Model Data* (STEP), é um exemplo desta classe de modelos dados.

2.1.2 Esquemas e instâncias

Em qualquer modelo de dados é importante distinguir entre a descrição do banco de dados e o banco de dados em si. A descrição do banco de dados é chamada de esquema do banco de dados ou meta-dados. O esquema de uma banco de dados é especificado durante o projeto do banco de dados e é esperado que ele não mude freqüentemente. Ele descreve aspectos estruturais do banco de dados como nomes de tipos de registros e itens de dado e restrições de integridade referencial. O esquema pode também descrever restrições semânticas como: “Estudantes a partir da terceira fase do curso de ciências da computação podem fazer a disciplina de sistemas operacionais desde que já tenham feito a disciplina estrutura de dados”.

Os dados armazenados em um banco de dados podem mudar com freqüência cada vez que operações como inserção, remoção ou atualização, são realizadas e registros são inseridos, apagados ou modificados. O conjunto de dados em um banco de dados em um particular momento é chamado de estado do banco de dados. A distinção entre o

esquema do banco de dados e seu estado é pertinente. Quando um banco de dados é definido, apenas seu esquema é especificado para o SGBD. Neste ponto, o banco de dados correspondente está no "estado vazio", sem dados armazenados. A partir deste ponto, a cada atualização nos dados o banco de dados muda de estado. O SGBD é parcialmente responsável por assegurar que todo estado do banco de dados é um estado válido, ou seja, que este estado mantenha a estrutura e atenda às restrições especificadas no esquema. O SGBD armazena o esquema em uma estrutura chamada catálogo para referenciá-lo facilmente sempre que necessário.

2.1.3 Arquitetura de SGBD

A arquitetura de SGBD mais utilizada atualmente é a produzida pelo comitê formado pelos grupos ANSI (*American National Standard Institute*) e SPARC (*Standards Planning and Requirements Committe*). Esta arquitetura utiliza um modelo de referência baseado na organização de dados, onde são definidas as unidades funcionais que terão acessos a diferentes tipos de dados de acordo com diferentes visões sobre o sistema.

A arquitetura ANSI/SPARC, ilustrada na figura 1, define uma arquitetura em três níveis, cada um com seu corresponde esquema descritivo e visão sobre os dados. São eles:

- O nível interno. Este é o nível da arquitetura ANSI/SPARC mais próximo ao dispositivo de armazenamento. Neste nível são realizadas a definição física e a organização dos dados no meio de armazenamento. A localização dos dados e os mecanismos de acesso usados para obter e manipular dados são questões tratadas neste nível.
- O nível conceitual. Este nível representa a visão abstrata de um banco de dados. Ele reproduz a visão de "mundo real" do ambiente modelado no banco de dados. Desta forma, o esquema conceitual representa os dados e o relacionamento sem considerar requisições de aplicações individuais ou restrições da meio físico utilizado para armazenamento.
- O nível externo. Este nível é responsável pela interface com o usuário final, definindo a maneira como usuários "enxergam" o banco de dados. Uma visão de usuário individual representa a porção do banco de dados que será acessada e manipulada por este usuário. Uma visão pode ser compartilhada por um certo

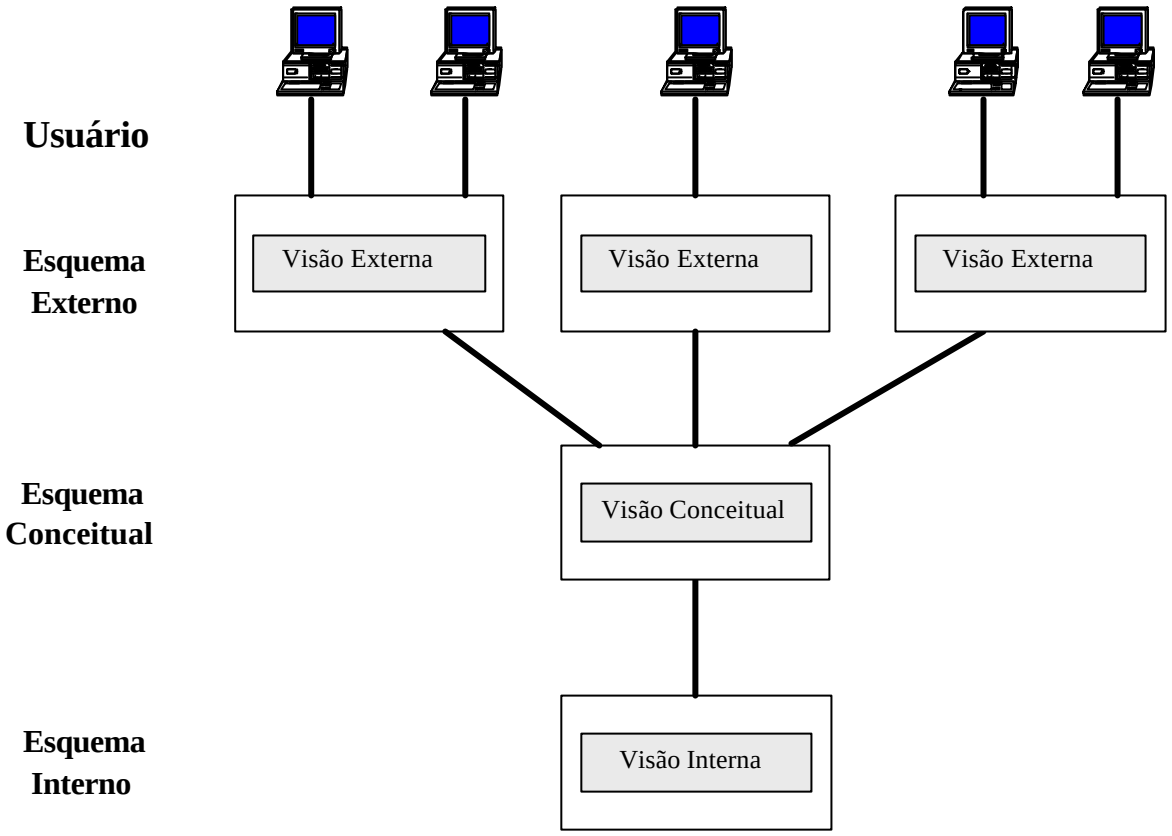

Figura 1: Arquitetura ANSI/SPARC

número de usuários , sendo que a união das visões de usuários constitui a visão externa.

Na prática, normalmente não é possível atender a todas estas requisições completamente, devido a questões de performance. A maioria dos SGBDs não separa os três níveis completamente. Alguns SGBDs incluem detalhes do nível físico no esquema conceitual. Na maioria dos SGBDs que suporta visões de usuário, os esquemas externos são especificados com o mesmo modelo de dados que descreve a informação do nível conceitual.

É importante ressaltar que, os três esquemas são apenas descrições dos dados; os dados encontram-se realmente apenas no nível físico. Em um SGBD baseado em uma arquitetura de três níveis como a ANSI/SPARC, cada grupo de usuários refere-se apenas para seu próprio esquema externo. Conseqüentemente, o SGBD deve transformar uma requisição especificada no esquema externo em uma requisição no esquema conceitual e depois em uma requisição no esquema interno para finalmente

processar os dados armazenados no banco de dados. Caso a requisição seja uma recuperação de dados, o dado extraído do banco de dados deve ser formatado para casar com a visão externa do usuário. O procedimento de transformar requisições e resultados entre níveis é chamado mapeamento.

2.1.4 Linguagens do SGBD

Uma vez que o projeto de um banco de dados está completo e um SGBD é escolhido para gerenciar o banco de dados, o próximo passo é especificar os esquemas em todos os níveis da arquitetura e qualquer mapeamento entre eles. Uma linguagem é utilizada então para definir cada esquema, que será processada por um compilador específico e posteriormente, o resultado será armazenado no catálogo do SGBD.

A terminologia utilizada para estas linguagens é relacionada ao esquema que elas irão descrever. Para descrição do esquema conceitual é utilizado a *Data Definition Language* (DDL). O esquema interno é definido pela *Storage Definition Language* (SDL). Para especificação do esquema externo através de visões de usuários é utilizada a *View Definition Language*. Adicionalmente, para permitir aos usuários manipular o banco de dados, como realizar operações de atualização de dados, é utilizada a *Data Manipulation Language* (DML).

É comum, nos SGBDs atuais, não fazer uma separação clara entre os tipos de linguagens explanados acima. O mais usual é a utilização de uma linguagem integrada que inclua construtores para definição do esquema conceitual, visões, manipulação de dados e definição de armazenamento. Um exemplo típico é a linguagem para banco de dados relacional SQL, que representa a combinação entre DDL, VDL, DML e SDL.

Com relação ao DML é importante destacar o padrão desenvolvido pela Microsoft *Open Database Connectivity* (ODBC) baseado no *Call Level Interface* (CLI) do SQL *Access Group*. O padrão representa uma camada de *middleware* entre a aplicação e os SBDBs a serem acessados. Esta é constituída pelo cliente ODBC, utilizado pela aplicação e o *driver* ODBC, com função de receber os comandos ODBC da aplicação e traduzi-los para a linguagem do SGBD de destino.

Outras linguagens também são utilizadas como *wrapper* para acesso a diferentes fontes de dados como o OLE DB e *XPath*. A primeira, também desenvolvida pela *Microsoft* permite o acesso a diversos arquivos gerados por aplicações como *Excel*, *Outlook* e *Project*. A segunda especificada pelo consórcio W3C permite o acesso a

documentos XML, inclusive a partes específicas do mesmo. Várias outras linguagens vem sendo utilizadas para acesso aos mais diversos tipos de fontes de dados. Normalmente o *wrapper* que utiliza estas linguagens faz parte do próprio conjunto de aplicações do banco de dados.

Algumas linguagens podem ser utilizadas para a integração entre esquemas descritos com diferentes modelos de dados. Estas linguagens são utilizadas na tradução dos esquemas dos SBDs locais para um modelo de dados comum através da especificação declarativa de mapeamentos entre as estruturas de cada modelo. A linguaguem BRIITY [HST97] é um exemplo deste tipo linguagem, desenvolvida para mapeamento entre visões definidas em EXPRESS e esquemas alvos.

2.2 Características de Sistemas Bancos de Dados Distribuídos

Há diversas maneiras como banco de dados podem ser colocados juntos para compartilharem múltiplos SGBDs. A taxonomia mais utilizada na literatura baseia-se em três dimensões ortogonais: autonomia, distribuição, e heterogeneidade. A seguir serão descritos os aspectos relevantes de cada dimensão para a área de sistemas federados.

2.2.1 Distribuição

Dados relacionados podem ser distribuídos entre múltiplos bancos de dados. A localização física pode ser tanto em um mesmo computador como em vários sistemas distribuídos geograficamente e interconectados por sistemas de comunicação de dados. Os dados podem ser distribuídos de diferentes maneiras, com múltiplas cópias de partes ou do todo mantidas e estruturadas de diferentes maneiras. Como exemplo tem-se, utilizando termos do modelo de dados relacional, a distribuição horizontal ou vertical de tabelas.

Em SGBDs distribuídos, dados podem ser espalhados de maneira induzida para obter certas vantagens como balanceamento de carga entre servidores e localidade para operações críticas. No caso de SBDFs, a maior parte da distribuição dos dados se deve a existência de múltiplos bancos de dados antes de o SFDB ter sido projetado.

2.2.2 Heterogeneidade

Heterogeneidade pode ocorrer de diversas maneiras em sistemas distribuídos. Algumas não estão relacionadas diretamente ao SGBDs, como heterogeneidade de hardware e sistema operacional e diferenças de protocolos de rede. Nos dias atuais, este tipo de heterogeneidade tem sido bastante minimizado e a maioria do SGBDs possuem versões disponíveis para diferentes arquiteturas e sistemas operacionais.

Por outro lado, a heterogeneidade causada por sistemas de banco de dados diferentes ainda apresenta questões que comumente tem que ser resolvidas caso a caso. É possível classificar dois tipos de heterogeneidades: estrutural e semântica. A primeira é decorrente de diferenças nos SGBDs ou nos meios utilizados para manipulação do banco de dados, enquanto que a segunda é resultado de diferenças na semântica dos dados.

2.2.2.1 Heterogeneidade estrutural

A heterogeneidade devido a diferenças entre os SGBDs participantes resulta em diferenças no modelo de dados. Com isso podem ocorrer diferenças na estrutura dos dados, nas restrições do modelo e nas linguagens de consultas. Detalhando melhor cada um destes aspectos tem-se:

- Diferenças na estrutura: Como conseqüência direta do uso de diferentes modelos de dados tem a heterogeneidade estrutural. A representação de dados com diferentes ferramentas de modelagens cria heterogeneidade porque herda o poder de expressão e limitações de modelos de dados individuais. Por exemplo, alguns modelos de dados como os modelos orientados a objetos suportam generalização e herança enquanto outros não. A tarefa de lidar com este tipo de diferença é bastante facilitada se duas representações compartilham o mesmo conteúdo de informação.
- Diferenças em restrições: Algumas restrições impostas sobre o banco de dados por um modelo podem não ser capturada por outro modelo. Mecanismos alternativos como *triggers* podem ser usados para lidar com estas diferenças e garantir a unidade comportamental entre os bancos de dados.
- Diferenças em linguagens de consultas: Linguagens de consulta com diferentes paradigmas são utilizadas para manipular dados representados por diferentes modelagens (por exemplo, acesso sobre conjuntos em sistemas relacionais e acesso sobre registros individuais em sistemas hierárquicos). Mesmo que dois SGBDs

suportem o mesmo modelo de dados, diferenças em suas linguagens de consulta podem levar à heterogeneidade (por exemplo, as linguagens QUEL e SQL ambas destinadas ao modelo relacional). Linguagens de consultas diferentes e que usam o mesmo modelo de dados freqüentemente selecionam métodos bastante diferentes para expressar requisições idênticas.

Outros tipos de heterogeneidades ocorrem devido a outros aspectos dos SGBDs. Por exemplo, técnicas para gerenciamento de transações como controle de concorrência, protocolos para *commit* e *recovery*, podem produzir dificuldades para integração de diferentes sistemas.

A integração entre sistemas de banco de dados e os chamados sistemas legados também apresentam vários problemas relacionados à heterogeneidade estrutural. Uma das principais características dos sistemas legados é a inexistência de um esquema conceitual ou a forte similaridade do mesmo com o esquema interno. A maioria das aplicações que utilizam estes sistemas diretamente requer grande rapidez para acesso aos dados. Esta requisição conduz implicitamente a esquemas não-normalizados e otimizados para perfis de acesso específicos. Como conseqüência, a estrutura dos esquemas difere com as aplicações e seus perfis de acesso.

Sistemas legados são normalmente intercalados em infra-estruturas de processamento de informações que são consultadas via interfaces codificadas *hard-coded* em várias aplicações e sistemas relacionados. A migração de um sistema legado para uma nova geração, que reimplemente aplicações de uma maneira uniforme e que abstraia ao máximo os detalhes da representação física dos dados, não é, na maioria das vezes, uma solução viável por razões econômicas.

2.2.2.2 Heterogeneidade Semântica

Heterogeneidade semântica ocorre quando não existe concordância entre o significado, interpretação ou finalidade de uso de um mesmo dado. Como exemplo considere o SUP_PRIM da relação ECONOMIA no banco de dados BR, que descreve quanto uma determinada organização (ou país) está economizando para honrar os juros de uma dívida. O cálculo do valor deste atributo é feito sem considerar os gastos com infra-estrutura básica. Em outro banco de dados existe um outro atributo de mesmo nome na relação DEVEDORES no banco de dados FMI, onde o valor deste atributo é calculado considerando os gastos com infra-estrutura básica. Considerando que estes

atributos possuem as mesmas propriedades sintáticas, uma tentativa de comparar BR.ECONOMIA.SUP_PRIM e FMI.DEVEDORES.SUP_PRIM é errônea porque eles são semanticamente heterogêneos.

A tarefa de lidar com a integridade semântica talvez seja a mais árdua durante a integração de bancos de dados. O projeto de um banco de dados é influenciado, na maioria das vezes, pelas necessidades de uma aplicação em particular para otimizar a performance. De modo análogo, restrições de integridade são freqüentemente embutidas, distribuídas e replicadas em aplicações, impossibilitando desta forma um controle da semântica dos dados de maneira uniforme. Como resultado, no nível de esquema conceitual, apenas um mapeamento parcial da semântica de uma aplicação é factível. Desta forma, capturar todos estes aspectos semânticos dificilmente poderão ser conduzidos de uma maneira totalmente automática.

2.2.3 Autonomia

Autonomia refere-se à distribuição de controle, não de dados. Ela indica o grau em que sistemas de banco de dados (SBD) individuais podem operar de maneira independente. Um SBD componente pode exibir vários tipos de autonomia e as requisições para classificar um sistema como autônomo podem ser especificadas de diversas maneiras. A classificação feita por [PZ88] é bastante utilizada na literatura que inclui três tipos de autonomia: projeto, comunicação e execução. Além destas três a autonomia de associação [SL90] também é importante para a classificação de sistemas federados. A seguir será discutida cada uma destes tipos de autonomias de SBDs.

Autonomia de projeto: Refere-se à possibilidade de um SBD componente de realizar suas próprias decisões de projeto. Esta é a principal causa da heterogeneidade em um SBDF. Entre as questões sobre o projeto de BDS incluem-se:

a) dados a serem gerenciados, ou seja, o subconjunto do mundo real a ser representado,
b) representação (modelo de dados e linguagem de consulta) e o nome dos elementos de dados,
c) conceituação ou interpretação semântica dos dados,
d) restrições usadas para gerenciar os dados,
e) funcionalidades do sistema,
f) associação ou intercâmbio com outros sistemas e
g) implementação.

Autonomia de comunicação: Refere-se à habilidade de cada SBD componente de decidir como e qual tipo de informação eles querem fornecer para os demais componentes ou para os programas que controlam a execução global do sistema.

Autonomia de execução: Refere-se à capacidade de um SBD componente de executar operações locais (comandos ou transações submetidas por usuários locais) sem interferência por parte das operações externas (operações submetidas através do SGBDF). Um SBD que possua autonomia de execução possui total controle sobre a ordem de execução das operações externas. Desta forma, um SBD componente pode abortar qualquer operação que não atenda à suas restrições locais e suas operações locais não são afetadas por sua participação no sistema federado. Desta forma a consistência do sistema e suas operações não devem ser comprometidas quando outros SGBDs juntam-se ou deixam o sistema de banco de dados distribuído.

Autonomia de associação: Refere-se à capacidade de um SBD componente de decidir quais funcionalidades (operações que ele suporta) e quais recursos (dados que ele gerencia) estarão disponíveis para acesso externo. Isto inclui ainda a habilidade de associar-se ou desassociar-se da federação e a habilidade de participar de outras federações.

Os requisitos para manter a autonomia dos SBDs e compartilhar seus dados são conflitantes. Em um ambiente real, normalmente não é possível suportar a autonomia dos componentes completamente. É comum que sistemas relaxem alguns aspectos da autonomia dos SBDs componentes. Por exemplo, a entrada ou saída de um SDB em um sistema federado pode ser implementada através de um acordo entre os administradores do banco de dados local e do sistema federado.

2.3 Taxonomia de sistema de banco dados federados

Um SBD pode ser centralizado ou distribuído. Um SBD centralizado consiste de um único SGBD centralizado gerenciando um único banco de dados em um mesmo computador. Um SBD distribuído consiste de um único SGBD distribuído gerenciando múltiplos banco de dados. Nestes sistemas existe uma distribuição física dos dados enquanto que o controle lógico/conceitual é centralizado e desempenhado por um único SGBD. Desta forma, tanto banco de dados integrados quanto distribuídos, utilizam uma abordagem logicamente centralizada. Eles provêem um único esquema conceitual para

usuários e programas de aplicação. Múltiplos esquemas externos (visões) podem ser fornecidos em nestes sistemas, mas um único esquema conceitual central é utilizado.

Alguns autores não diferenciam se os bancos de dados residem em um mesmo computador ou se estão separados geograficamente em diferentes arquiteturas de hardware e software. Outros autores entretanto, como [OV91], destacam a importância da distribuição física dos dados, pois ela adiciona características que não são encontrados quando os banco de dados residem em um mesmo computador. Por exemplo, diferentes estratégias para otimização de consultas distribuídas podem ser utilizadas baseadas em questões como taxa de transferência e latência de acesso dos DBSs componentes. Além disso, técnicas adicionais são necessárias para prover transparência sobre a localização dos dados quando os bancos de dados estão distribuídos fisicamente. De fato, apesar de que, pela classificação mais utilizada na literatura, não ser feita distinção se múltiplos banco de dados estão residentes em um mesmo computador ou não, na prática, esta questão é determinante em um projeto de banco de dados distribuídos.

Um sistema *multidatabase* (SMDB) suporta operações em múltiplos SDBs componentes. Cada SDB componente é gerenciado por um SGDB componente. Um SDB componente em um SMDB pode ser centralizado ou distribuído. Um SMDB é dito homogêneo se o SGBD de todos os SDBs componentes são idênticos; caso contrário ele dito heterogêneo. Sistemas que apenas permite apenas o intercâmbio de dados periódico entre múltiplos SGBDs e que não seja baseado em transações ou que apenas provejam acesso a um SGBD por vez não podem ser considerados um SMDB. O primeiro é referenciado na literatura como sistema de intercâmbio de dados; o último como interface para SGBD remoto.

As diferenças no nível de autonomia entre *multidatabases* e SGBDs distribuídos são também são refletidas em modelos arquiteturais. A diferença fundamental é o esquema conceitual global. No caso de SGBDs distribuídos, o esquema conceitual global define a visão conceitual de todo o banco de dados, enquanto que no caso de *multidatabases*, ela representa apenas a coleção de dados que cada SGDB local deseje compartilhar. No último, o banco de dados global é igual à união de todos os banco de dados locais enquanto que no primeiro apenas um subconjunto dos dados fazem parte do sistema distribuído.

A taxonomia seguida neste trabalho focaliza na dimensão de autonomia para classificação de SBDs. Desta forma SMDB podem ser classificados em dois tipos baseados na autonomia dos DBSs componentes: sistemas de banco de dados não-federados e sistemas de banco de dados federados. Um sistema de banco de dados não-federado é a integração de SGBDs componentes que não são autônomos. Ele possui apenas um nível de gerenciamento, e todas operações são executadas de maneira uniforme. Em contraste com um sistema federado, um sistema não-federado não faz qualquer distinção entre usuários locais e externos. Um tipo particular de sistema não-federado no qual todos bancos de dados são completamente integrados para fornecer um único esquema global pode ser chamado de SMDB unificado. Ele aparece logicamente para seus usuários como um DBS distribuído.

Um sistema de banco de dados federado é composto de DBSs componentes que são autônomos ainda que, façam parte de uma federação e permitam parcial e controlado compartilhamento de seus dados. Autonomia de associação implica que DBSs componentes possuem controle sobre os dados que eles gerenciam. Devido a esta requisição, não existe controle centralizado em uma arquitetura federada.

Um SBDF representa um compromisso entre nenhuma integração (onde os usuários devem explicitamente configurar suas interfaces com múltiplos banco de dados autônomos) e integração total (onde a autonomia de SBD componente é parcialmente sacrificada para que usuários possam acessar dados através de uma única interface global do sistema ainda que sem acessar diretamente um SGBDs externo como um usuário local). A arquitetura federada é bastante adequada para migrar um conjunto de autônomos e independentes SBDs para um sistema que permita parcial e controlado compartilhamento de dados sem afetar as aplicações existentes.

Para permitir compartilhamento controlado de dados ao mesmo tempo em que preserva a autonomia dos SBDs componentes e manter inalterada a execução das aplicações pré-existentes, um SBDF suporta dois tipos de operações: local e global (ou federativa). Esta dicotomia de operações local e global é uma característica essencial de um SBDF. Operações globais envolvem acesso a dados utilizando o SGBDF e podem abranger dados gerenciados por múltiplos SBDs. SBDs componentes devem controlar o acesso os dados que eles gerenciam. Operações locais são submetidas para um SBD componente diretamente.

SBDFs podem ser categorizados como levemente ou fortemente acoplados observando-se quem gerencia a federação e como componentes são integrados. Um SBDF é levemente acoplado se a responsabilidade por criar ou manter a federação é do usuário e não existe qualquer controle aplicado pelo sistema federativo e seus administradores. Uma federação é dita fortemente acoplada se a federação e seus administradores têm a responsabilidade de criar e manter a federação e ativamente controlar o acesso aos SBDs componentes.

Uma federação é constituída por uma seletiva e controlada integração de seus componentes. A atividade de desenvolvimento de um SBDF resulta na criação de um esquema federado onde as operações globais são executadas. Um SBDF levemente acoplado sempre suporta múltiplos esquemas federados. Um SBDF fortemente acoplado pode ter um ou mais esquemas federados.

Os termos *sistemas de bancos de dados federados* e *arquitetura de banco de dados federados* foram introduzidos por [HM85] com o significado de "uma coleção de componentes que unem federações levemente acopladas para compartilhar e intercambiar informações" e "um modelo de organização baseado em banco de dados autônomos, com o compartilhamento de dados entre si controlado através interfaces explícitas". Neste modelo pioneiro, a arquitetura federada consiste especificamente de componentes, que podem ser usuários individuais, aplicações, estações de trabalho e outros bancos de dados e um único dicionário federado. Este dicionário federado é um componente especializado que mantém a topologia da federação e controla a adição de novos componentes.

A arquitetura *multidatabase* de [LM90] compartilha muitas funcionalidades da arquitetura de [HM85]. As definições desta arquitetura incluem o que foi definido anteriormente como SBDFs levemente acoplados. A conceito chave de um SBDF, entretanto, são a autonomia dos componentes, e o compartilhamento parcial e controlado dos dados. Estes conceitos podem ser suportados também quando os componentes são fortemente acoplados.

2.4 Arquitetura de referência

Uma arquitetura de referência é necessária para ressaltar várias questões e decisões de projeto em um sistema de banco de dados. No contexto deste trabalho, uma arquitetura de referência é indispensável para confrontar um sistema de servidores

DICOM integrados, que é o objetivo deste trabalho, com um sistema de banco de dados federados. Através da definição de uma arquitetura de referência e de cada um de seus componentes é possível situar todos os requisitos derivados do objetivo de integrar servidores de imagens médicas autônomos com a teoria de sistemas de banco de dados federados. Com isso é possível identificar até que ponto é possível aplicar modelos já desenvolvidos na literatura para o desenvolvimento deste trabalho e em quais partes será necessário especializar outros modelos.

A arquitetura de referência seguida neste trabalho é de [SL90]. Esta arquitetura possui como tipos básicos comandos, esquemas, bancos de dados, processadores, mapeamentos e os dados em si. Dentre estes componentes, os processadores possuem um papel especialmente importante em um sistema federado. Os processadores são módulos de software independentes de aplicação de um SGBD. A seguir serão detalhados os tipos possíveis de processadores da arquitetura de referência.

2.4.1 Tipos de processadores em uma arquitetura de referência

Arquiteturas de gerenciamento de dados diferem nos tipos de processadores presentes e no relacionamento entre estes processadores. Existem quatro tipos de processadores, cada um realizando diferentes funções na manipulação de comandos e dados acessados: processadores de transformação, processadores de filtragem, processadores de construção e processadores de construção. A seguir, serão apresentados os tipos de processadores em uma arquitetura federada.

2.4.1.1 Processador de transformação

Processadores de transformação traduzem comandos de uma linguagem, chamada linguagem fonte, para outra linguagem, chamada linguagem alvo, ou transformam dados de um formato (formato fonte) para outro formato (formato alvo). Processadores de transformação provêem um tipo de independência de dados chamado transparência de modelos de dados, onde as estruturas de dados e comandos usados por um processador são escondidas dos demais processadores. A transparência de modelo de dados esconde as diferenças em linguagens de consultas e formatos de dados. Por exemplo, as estruturas de dados usadas por um processador podem ser modificadas para aumentar a eficiência sem obrigar mudanças no outros processadores.

Processadores de transformação normalmente realizam a tradução dos dados resultantes juntamente com a tradução dos comandos de entrada. Por exemplo, considere um processador de transformação que realiza a transformação de comandos SQL para comandos CODASYL, permitindo que dados em sistema CODASYL possam ser acessados através de comandos SQL. Será necessário também traduzir os registros resultantes da consulta ao banco CODASYL em tabelas para apresentação ao usuário SQL.

Para realizar estas transformações, um processador de transformação precisa de mapeamentos entre os objetos de cada esquema. A tarefa de tradução de um esquema envolve transformar um esquema (esquema A) que descreve dados usando um determinado modelo de dados em um esquema equivalente (esquema B) descrevendo os mesmos dados com um modelo de dados diferente. Esta tarefa também gera os mapeamentos que correlacionam os objetos de um esquema (esquema B) com objetos de outro esquema (esquema A). A tarefa de transformar comandos envolve utilizar estes mapeamentos para traduzir comandos relacionados a objetos de um esquema (esquema B) em comandos relacionados a objetos de outro esquema (esquema A).

Mapeamentos são associados com processadores de transformação em uma de duas maneiras. No primeiro caso, os mapeamentos são codificados na lógica do processador de transformação, conduzindo a processadores de transformação específicos para determinados esquemas. Alternativamente, os mapeamentos podem ser armazenados em uma estrutura de dados separada e acessados pelo processador de transformação durante a conversão de comandos e dados. Esta última é uma abordagem mais flexível.

2.4.1.2 Processador de filtragem

O processador de filtragem restringe os dados e comandos que podem ser passados para outros processadores. Associados com cada processador de filtragem estão os mapeamentos que descrevem as restrições sobre os comandos e dados. Estas restrições, da mesma forma que os mapeamentos do processador de transformação, podem estar tanto embutidas no código do processador de filtragem como especificadas em uma estrutura de dados separada. Exemplos de processadores de filtragem seguem:

- Corretor sintático, o qual checa comandos para verificar se os mesmos estão sintaticamente corretos antes de passar para camadas inferiores da arquitetura.

- Corretor de integridade semântica, o qual checa comandos de entrada e/ou dados de saída para verificar se os mesmos atendem a todas restrições de integridade semântica, podem modificá-los para adequá-los automaticamente.
- Controlador de acesso, o qual verifica se apenas usuários com permissões suficientes podem executar determinados comandos sobre determinados dados ou verifica se o usuário pode usar os dados produzidos por um determinado processador.

Uma tarefa importante que pode ser resolvida pelo processador de filtragem é a atualizações de visões. Esta tarefa ocorre quando as diferenças ente as estruturas de dados da visão e o esquema existem de tal forma que permite a existência de mais de uma maneira de traduzir ou atualizar um comando.

2.4.1.3 Processador de construção

Processadores de construção dividem e/ou replicam uma operação submetida por um único processador em operações que são aceitas por dois ou mais processadores. De maneira simétrica, processadores de construção podem combinar dados produzidos por vários processadores em um único conjunto de dados para consumo por outro único processador. Eles podem suportar transparência de localização, distribuição e replicação de maneira que um processador não precise conhecer nenhuma destas informações ao submeter comandos para outro processador.

Entre as tarefas que podem ser gerenciadas pelos processadores de construção estão:

- Integração de esquemas: Integrar múltiplos esquemas em um único esquema. O processo de integração de esquemas não pode ser realizado de maneira automatizada. Isto porque seria necessário que toda a semântica dos esquemas estivessem completamente especificadas, o que, atualmente, nenhum modelo de dados é capaz de fazer. O processador de construção é utilizado nas etapas posteriores do processo de integração, após a comparação e conformação dos objetos dos esquemas. Estas etapas podem envolver a tradução dos esquemas para um modelo de dados comum (tratamento de heterogeneidade estrutural) e na manutenção de um dicionário usado para identificação de conflitos, homônimos e sinônimos, etc (tratamento de heterogeneidade semântica).
- Negociação: Determinar que protocolo deve ser usado entre os administradores dos vários esquemas locais a serem integrados e determinar o conteúdo do esquema

integrado resultante. Para definir um esquema federado, os administradores dos bancos de dados locais e federados devem realizar um acordo sobre o conteúdo dos esquemas de exportação e as operações permitidas. O processador de construção é responsável por prover mecanismos para tratar diversos aspectos dessa negociação. Por exemplo, quando um administrador de banco de dados componente decide retirar o direito de acesso dos usuários federados sobre um objeto do esquema local ou modificar atributos deste objeto. Outro exemplo, seria quando um usuário federado decide acessar um objeto de um esquema que não está mais disponível.

- Decomposição de consultas e otimização: Decomposição de consultas realizadas em um esquema integrado. A decomposição de consultas consiste em converter uma consulta aplicada ao esquema federado em várias consultas aplicadas ao BDs componentes e executar estas consultas. As otimizações sobre estas consultas podem ser realizadas principalmente em um SBDF fortemente acoplado. O processamento de consultas em um SGBDF é bastante similar a um SGBD distribuído. Entretanto um SGBDF tem que lidar com complexidades adicionais devido à heterogeneidade e autonomia de seus componentes. Por exemplo, o custo de executar uma operação pode ser diferente entre os BDs componentes devido a diferenças entre a habilidade de cada SGBD local em realizar otimizações locais. Devido à autonomia dos BDs componentes, o custo de uma operação pode variar até mesmo quando aplicada a um único BD devido a questões de gerenciamento local. Todas estas questões são tratadas pelo processador de construção no SGBDF (onde é chamado de Gerenciador de Dados Globais). Quando tratadas no BD componente é tratada pelo processador de transformação (onde é chamado de Interface de Banco de dados Local).
- Gerenciamento global de transações: Controle de concorrência e atomicidade para manter o sistema federado consistente entre múltiplas atualizações simultâneas. O processador de construção neste caso funciona como um gerenciador global de transações. Entretanto, o suporte a este tipo de tarefa em um sistema federado seguindo o modelo do controle de transações aplicado em um sistema local, ou mesmo distribuído, é praticamente impossível devido à autonomia dos BDs componentes. O problema principal é que o SBGDF não tem conhecimento sobre as transações locais desde que os SGBDs componentes são autônomos. Desta forma, todas as possíveis soluções para prover um gerenciamento global de transações em

sistema federado tem que sacrificar a autonomia dos BDs locais ou a *seriabilidade* das transações globais.

2.4.1.4 Processador de acesso

Um processador de acesso aceita comandos e produz dados aplicando estes comandos ao banco de dados. Ele pode aceitar comandos de vários processadores e intercalar o processamento destes comandos. Exemplos de processadores de acesso seguem:

- Um sistema de gerenciamento de arquivos que executa rotinas de acesso sobre um arquivo armazenado.
- Um programa de aplicação que aceita comando e gera dados para serem retornados ao processador que gerou estes comandos.
- Um gerenciador de dados de um SGBD contendo os métodos de acesso a dados.
- Um gerenciador de dicionários que gerencia acesso ao dicionário de dados do sistema.

Outras questões que são tratadas pelo processador de acesso incluem controle de concorrência local, e operações de gerenciamento de dados como *commitment*, *backup*, e *recovery*.

2.4.2 Arquitetura de esquemas em cinco níveis para banco de dados federados

A arquitetura de três níveis ANSI/SPARC descrita na seção 2.1.3 é adequada para descrever a arquitetura de um SGBD centralizado. Entretanto, ela é inadequada para descrever a arquitetura de um SBDF. O esquema de três níveis deve ser estendido para suportar os três níveis de dimensões de um sistema de banco de dados federado – distribuição, heterogeneidade e autonomia. Na Figura 2 é representada a arquitetura de esquema de cinco níveis definida por [SL90] para sistemas federados e utilizada a longo deste trabalho. A figura 3 representa a arquitetura de sistema consistindo de esquemas e processadores de um SBDF. A seguir será descrito cada esquema desta arquitetura:

Esquema Local: Um esquema local é o esquema conceitual de um BDS componente. Um esquema local é representado no modelo de dados nativo do SGBD componente, e conseqüentemente, diferentes esquemas locais podem ser representados em diferentes modelos de dados.

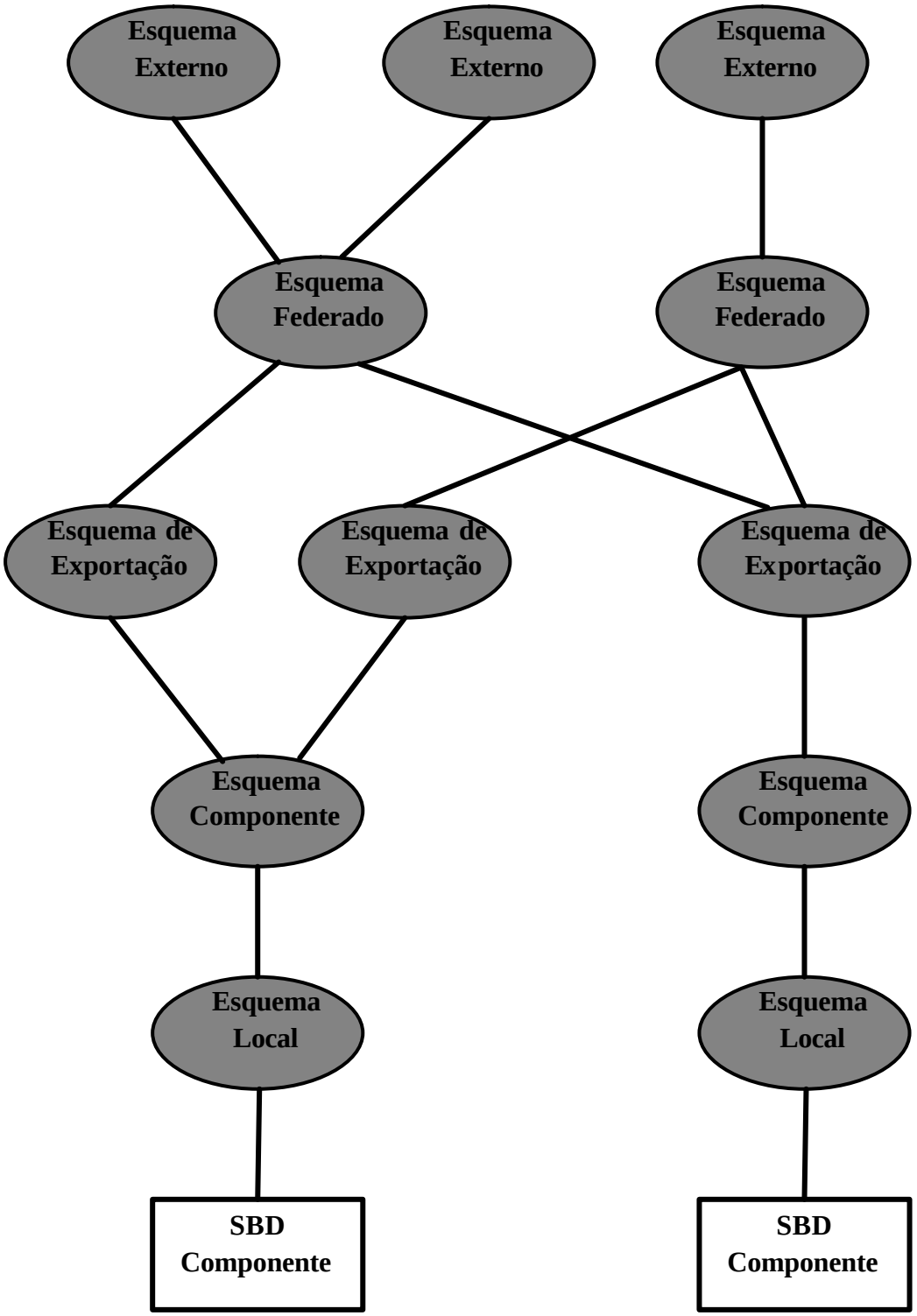

Figura 2: Arquitetura de cinco níveis de um SBDF

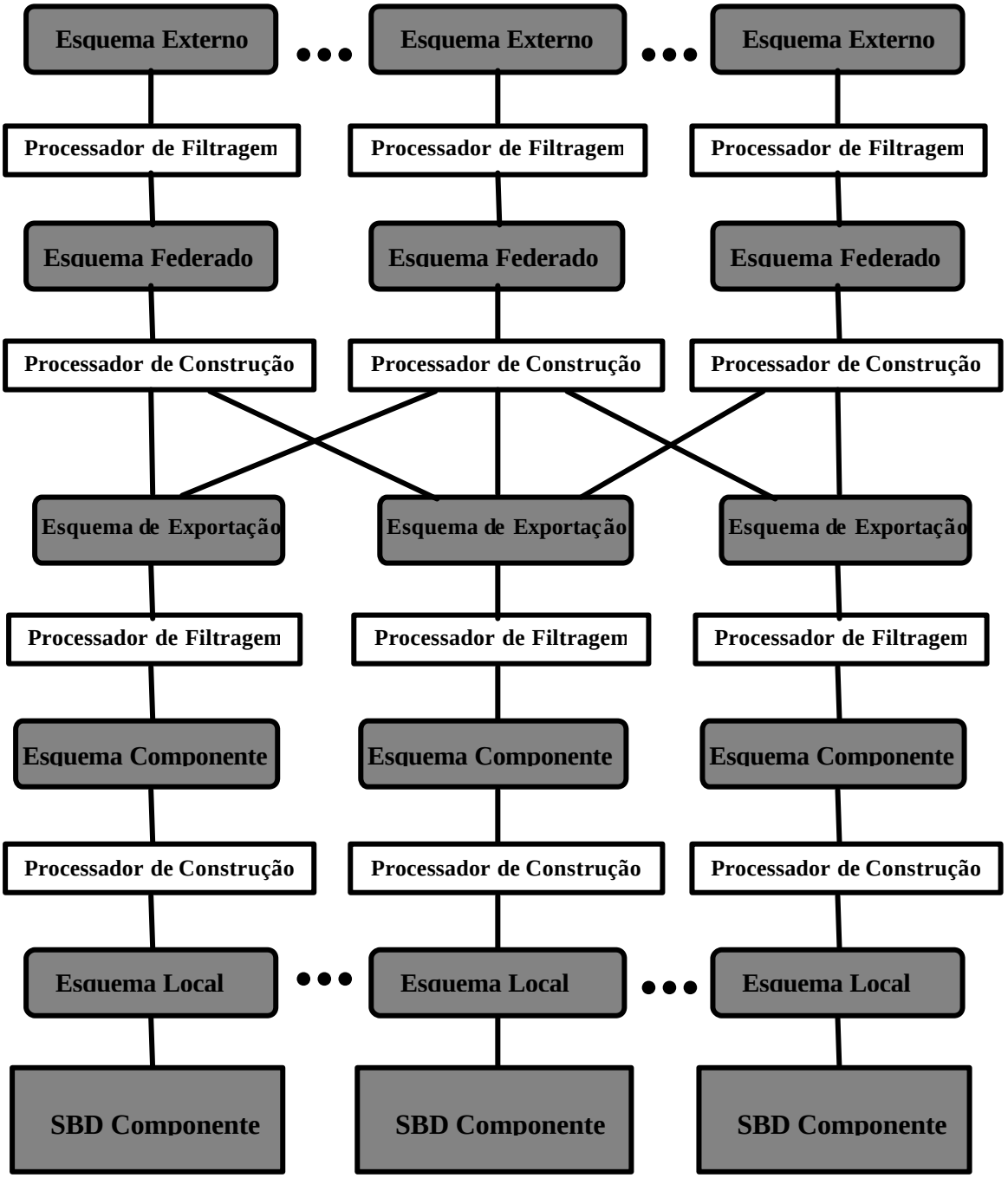

Figura 3: Arquitetura de sistema de SBDF

Esquema Componente: Um esquema componente é derivado pela tradução dos esquemas locais em um modelo de dados chamado de modelo de dados comum do SBDF. As duas razões para se definir os esquemas componentes em um MDC são: 1) eles descrevem os esquemas locais divergentes usando uma única representação e 2)

informações semânticas ausentes em um esquema local podem ser adicionadas para seu esquema componente correspondente. Desta forma, as tarefas de negociação e integração realizadas durante o desenvolvimento de um SBDF fortemente acoplado são facilitadas. De maneira análoga, os esquemas componentes facilitam a especificação de visões e consultas de *multidatabases* em um SBDF levemente acoplado.

O processo de tradução de um esquema local para um esquema componente gera mapeamentos entre os objetos do esquema componente e os objetos do esquema local. Processadores de transformação utilizam-se destes mapeamentos para transformar comandos aplicados aos esquemas componentes em comandos correspondentes aos esquemas locais.

Esquema de Exportação: Nem todos os dados de um SBD componente poderão estar disponíveis para a federação e seus usuários. Um esquema de exportação representa um subconjunto de um esquema componente que é disponível para o SBDF. Ele pode incluir informação de controle de acesso para usuários da federação. O propósito de definir esquemas de exportação é facilitar o gerenciamento para os administradores dos SBDs componentes sobre as informações disponibilizadas para o sistema federado, assegurando a autonomia de associação. Um processador de filtragem pode ser usado para prover o controle de acesso definido pelo esquema de exportação, limitando o conjunto de operações que podem ser submetidas ao esquema componente correspondente. Processadores de filtragem e os esquemas suportam a autonomia de um SBDF.

Esquema Federado: Um esquema federado é uma integração de múltiplos esquemas de exportação. Um esquema federado também inclui a informação sobre a distribuição que é gerada quando os esquemas de exportação são integrados. Alguns sistemas usam um esquema separado chamado *esquema de distribuição* ou *esquema de alocação* para conter estas informações. Um processador de construção transforma comandos aplicados ao esquema federado em comandos correspondentes a um ou mais esquema de exportação. Processadores de construção e os esquemas federados suportam a distribuição de um SBDF.

Podem existir múltiplos esquemas federados em SBDF, um para cada classe de usuários da federação. Uma classe de usuários da federação é um grupo de usuários e/ou

aplicações que executam um conjunto relacionado de atividades sobre o esquema federado.

Esquema Externo: Um esquema externo define o esquema que será usado diretamente por usuários e aplicações. As razões para o uso de um esquema externo são as seguintes:

- Customização: Um esquema federado pode ser bastante grande, complexo e difícil de aplicar modificações. Um esquema externo pode ser usado para especificar um subconjunto das informações do esquema federado que é relevante para os usuários do esquema externo. Eles podem ser modificados mais rapidamente para atender a modificações das necessidades dos usuários. O modelo de dados para um esquema externo pode ser diferente do adotado pelo esquema federado.
- Restrições de integridade adicionais: Restrições de integridade adicionais podem ser especificadas no esquema externo, como restrições direcionadas para um grupo específico de aplicações ou usuários.
- **Controle de acesso:** Esquemas de exportação provêem controle de acesso com respeito aos dados gerenciados pelos bancos de dados componentes. Similarmente, esquemas externos fornecem controle de acesso relacionado aos dados gerenciados pelo SBDF.

Um processador de filtragem analisa os comandos de um esquema externo para assegurar sua conformidade com o controle de acesso e restrições de integridade do esquema federado. Se um esquema externo está representado em um modelo de dados diferente do esquema federado, um processador de transformação será necessário para transforma comandos aplicados ao esquema externo em comandos para o esquema federado.

2.4.3 Modelo de dados comum e mapeamentos

Além de adicionar níveis na arquitetura de esquemas, os requisitos de heterogeneidade e autonomia podem também ditar mudanças no conteúdo de um esquema. Por exemplo, se um SBDF possui múltiplos SGBDs heterogêneos, provendo diferentes funcionalidades de gerenciamento de dados, um esquema componente deverá conter informações sobre as operações suportadas pelo seu SGDB componente.

Um SBDF pode ter que suportar esquemas locais e externos descritos em modelos de dados diferentes. Para facilitar as atividades de projeto, integração e manutenção, entretanto, os esquemas federados deverão estar representados em um mesmo modelo

de dados. Este modelo de dados é chamado de modelo de dados comum (do inglês *common data model,* CDM). Uma linguagem associada com o CDM é chamada de *linguagem de comandos interna*. Todos comandos aplicados aos esquemas federado, de exportação e componente são representados usando estar linguagem de comandos interna.

Projeto e integração de uma banco de dados federado é um processo complexo envolvendo não apenas a estrutura dos dados armazenados nos bancos de dados mas também a semântica dos dados. Assim, é desejável utilizar um modelo de dados que possibilite construções semânticas mais detalhadas que o modelo dados usado para descrever os esquemas locais. Desta forma, o esquema componente poderá conter mais informações semânticas que o esquema local correspondente. Informações adicionais podem ser fornecidas pelo desenvolvedor do SBDF durante os processos de projeto, integração e transformação do esquema.

Um importante tipo de informação associada com todos esquemas de um SBDF são os mapeamentos. Eles correlacionam objetos de esquemas de um nível com os objetos de esquema no nível mais baixo na arquitetura. Os mapeamentos podem ser armazenados como parte da informação do esquema ou como objetos distintos do dicionário de dados do SBDF. A quantidade de informações necessárias para descrever um objeto pode ser diferente de um esquema para outro. Por exemplo, a descrição de um tipo de entidade em um esquema federado pode incluir os nomes dos usuários que podem acessá-lo, ao passo que esta informação não é armazenada para outro tipo de entidade em um esquema componente. Os tipos de objetos entre diferentes esquemas podem variar. Por exemplo, um esquema federado pode ter objetos descrevendo as capacidades de vários SGBDs componentes no sistema, enquanto que estes tipos de objetos não existem em esquemas locais.

3. DICOM 3.0 (DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE VERSION 3.0)

Durante a década de 1970, com surgimento da tomografia computadorizada e outras modalidades de diagnóstico digital, intensificou-se o uso de computadores em aplicações clínicas. As diversas soluções proprietárias que surgiram em resposta a esta demanda produziram um cenário onde uma variedade de protocolos de transmissão e formatos de imagens digitais coexistiam sem qualquer interoperabilidade entre si.

Este panorama fez com que em 1983, o *American College of Radiology* (ACR) e o *National Electrical Manufacturers Association* (NEMA) formassem um comitê no intuito de desenvolver um padrão que possibilitasse a comunicação entre equipamentos de imagens médicas digitais de diferentes fabricantes. O primeiro produto deste comitê foi publicado em 1985 sob o nome de ACR-NEMA *Standards Publication* N° 300-1985, sob a versão 1.0. Após duas revisões, em 1988 foi publicada a segunda versão do padrão ACR-NEMA. Ambas publicações abordavam a interface de hardware, um conjunto mínimo de comandos de softwares e um dicionário dos elementos necessários para codificação, interpretação e exibição das imagens.

Finalmente em 1989 foi publicada a versão 3.0 do padrão ACR-NEMA, agora sob o nome de *Digital Imaging and Communications in Medicine* (DICOM). Esta versão trouxe um grande número de melhorias das versões anteriores. Entre as principais estavam:

- Suporte para um ambiente de rede. O padrão ACR-NEMA era aplicável apenas em um ambiente ponto-a-ponto. O padrão DICOM suporta a pilha de protocolos rede TCP/IP.
- Suporte para o ambiente de intercâmbio de mídia *off-line*. O padrão ACR-NEMA não especificava um formato de arquivos ou sistema de arquivos lógico. DICOM suporta operação em um ambiente de intercâmbio de mídias *off-line* usando meios como CD-R e MOD e sistema de arquivos lógicos como ISSO 9660 e FAT16.
- Especificação sobre como dispositivos acordantes com o padrão devem reagir a comandos e dados recebidos. Através do conceito de classes de serviços, DICOM especifica a semântica dos comandos e dados associados.
- Especificação dos níveis de conformidade. DICOM explicitamente descreve como um desenvolvedor deve estruturar um documento de conformidade de seu produto.

- Estruturação da documentação em diversas partes. Com isto a adição de novas funcionalidades ao padrão é simplificada. Esta estruturação segue as diretivas definidas pela organização ISO.
- Introdução do conceito de objetos de informação. As entidades do mundo real que são de interesse do padrão DICOM são modelados em definições explícitas, chamadas objetos de informação. Estas definições existem não apenas para imagens e gráficos mas também para *waveforms*, laudos, elementos de impressão, etc.
- Especificação de técnicas para unicamente identificar um objeto de informação. Com isto é possível definir o relacionamento entre objetos de informação em um ambiente de rede de maneira inequívoca.

Atualmente, o padrão DICOM encontra-se consolidado como padrão de fato mundial para a transmissão, arquivamento e formatação de imagens radiológicas digitais. A possibilidade de utilizar aparelhos de imagem, impressoras, *scanners*, câmeras digitais, bem como uma grande variedade de softwares de diversos propósitos e fornecedores, conectados por uma rede de baixo custo, impulsionou o crescimento de sistemas PACS (*Picture Archiving and Comunnication Systems*). Além disso, o padrão DICOM facilita a interface entre o sistema de informação do ambiente radiológico, RIS (*Radiological Information System*) e o restante do sistema hospitalar, HIS (*Hospital Information System*).

3.1 Descrição teórica do padrão DICOM

Para estabelecer a integração entre servidores DICOM de imagens é necessário seguir o conjunto de protocolos, serviços e interfaces de comunicação em rede definidas pelo padrão. Também é necessário interpretar corretamente a sintaxe e semântica dos comandos e objetos de informação transportados durante uma interação entre entidades de aplicação. A seguir apresentaremos as principais partes do padrão DICOM que são relevantes para este trabalho.

3.1.1 Modelo de aplicação DICOM

A partir de usa última versão, o DICOM 3.0 baseado em modelos explícitos e detalhados de como as entidades de informação (paciente, imagem, relatório, etc) que compõem a área de interesse do padrão DICOM devem ser descritas e como elas estão relacionadas entre si. A descrição deste modelo, que corresponde ao Modelo de

Aplicação DICOM, é feita utilizando os diagramas Entidade-Relacionamento, que é bastante útil por fornecer aa abstração necessária para garantir o entendimento comum tanto para usuários quanto para fabricantes de equipamentos e desenvolvedores de softwares que utilizam a estruturação de dados descritos no DICOM.

O padrão DICOM define o modelo de aplicação DICOM, um modelo abstrato de dados, orientado a objetos, usado para especificar classes de objetos do mundo real que fazem parte da área de interesse do padrão DICOM. Neste modelo, que também é referenciado como modelo DICOM do mundo real, é definido também o relacionamento entre estas classes de objetos e o escopo do padrão DICOM.

3.1.2 Definição de objetos de informação

A representação das classes de objetos do mundo real que compõem o escopo do padrão DICOM recebe o nome de IOD (*Information Object Definition*). Cada IOD define a natureza e atributos relevantes da classe de objetos do mundo real que é representada. Um IOD provê às aplicações DICOM uma visão comum sobre o conjunto de informações a serem trocadas. O diagrama entidade-relacionamento usado para modelar os relacionamentos entre os IODs é chamado de Modelo de Informação DICOM. Este diagrama é diretamente derivado do modelo de aplicação DICOM onde as entidades do mundo real são representadas por IODs. Não é necessário entretanto, uma relação de um para um entre IOD e entidades do mundo real, ou um IOD pode representar um conjunto de entidades relacionadas. A figura 4 apresenta o diagrama dos principais componentes do modelo de informações DICOM.

IODs podem ser classificados de duas formas baseando-se em sua representação de objetos do mundo real: normalizados e compostos. IOD normalizados representam geralmente uma única entidade do modelo DICOM do mundo real [DM03] e são utilizados principalmente para representação de elementos de gerenciamento de atividades. O IOD *Print Queue* é um exemplo de um IOD Normalizado. IODs normalizados não fazem parte do escopo deste trabalho, desde que o objetivo do mesmo é o compartilhamento de modalidades de imagem, laudos estruturados e sinais biológicos, o que não é representado por IODs normalizados.

Para intercâmbio de modalidades contendo estruturas complexas, como seqüências de dados de imagens, laudos estruturados ou *waveforms* é necessário que o contexto completo seja transmitido entre as Entidades de Aplicação comunicantes. Para

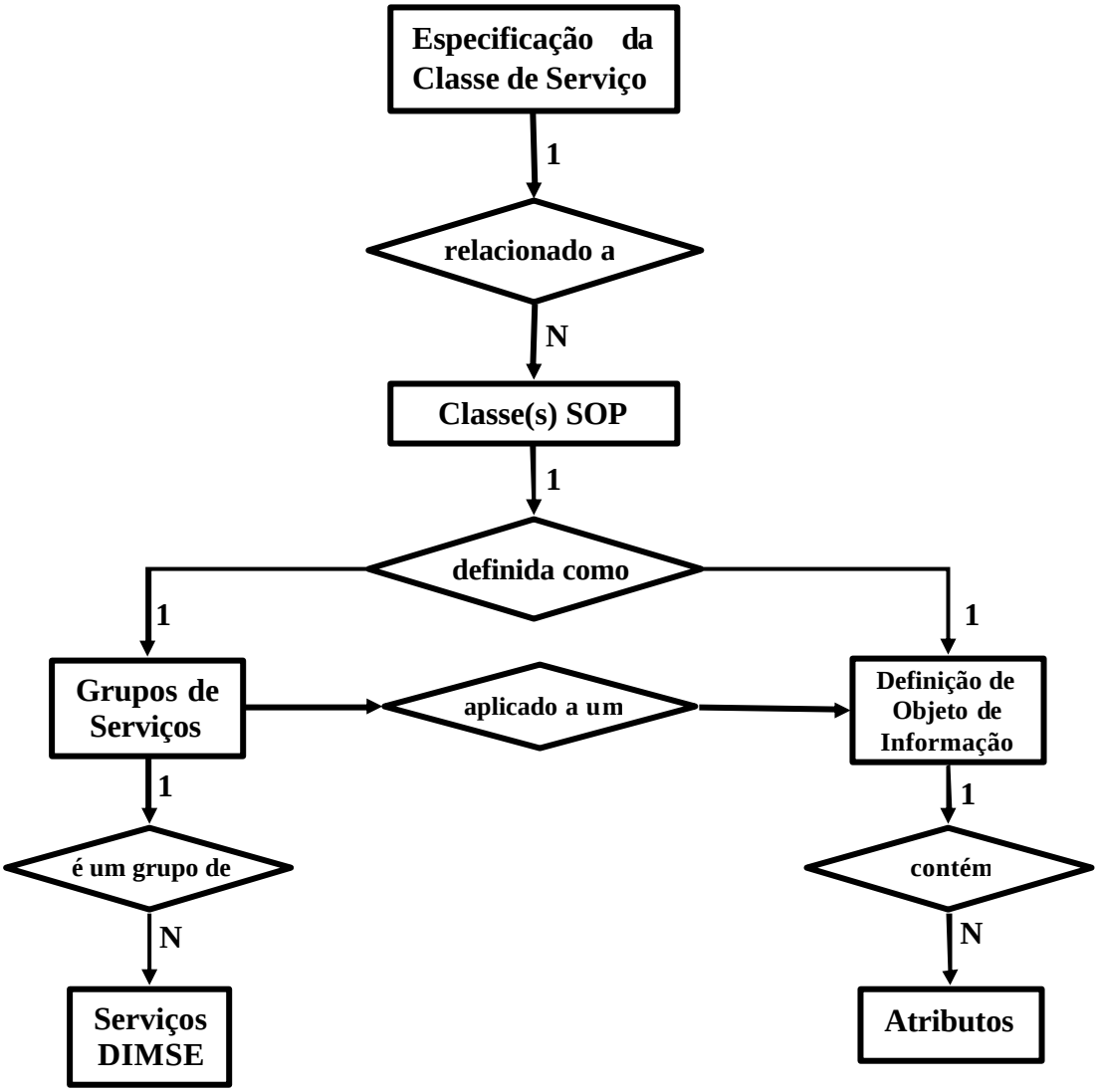

Figura 4: Modelo de informações DICOM

estas classes de modalidades, diversas Entidades de Informação presentes no Modelo de Aplicação DICOM precisam ser transmitidas para assegurar a correta interpretação da informação. Por exemplo, durante a transmissão de uma imagem de Tomografia Computadorizada, é necessário não apenas as informações inerentes para apresentação da imagem, como dados de *pixel* e transformações a serem aplicadas sobre os mesmos, mas também dados do paciente, estudo, relação temporal com outras imagens, etc. Portanto, os atributos presentes em um IOD Composto não são derivados apenas do objeto do mundo real que ele representa, por exemplo uma determinada modalidade de imagem radiológica, mas sim de um conjunto relacionado de Entidade de Informação

que, juntas, compõe o completo contexto onde este objeto está inserido. A estrutura que representa este conjunto de Entidades de Informação relacionadas em um contexto é o IOD Composto.

O padrão DICOM define ainda o conceito de IOM (*Information Object Modules*), que consiste basicamente em um conjunto de atributos relacionados dentro de uma Entidade de Informação ou IOD Normalizado. Um IOM representa um alto nível semântico da documentação do padrão DICOM. Esta estruturação lógica de atributos relacionados facilita a definição e implementação dos objetos de informação permitindo que um mesmo módulo esteja presente em vários IODs. Existem também módulos específicos para uma determinada modalidade possuindo, opcionalmente, especializações de determinados atributos presentes em outros módulos. Dentro do escopo de um IOD, um módulo pode ter a seguinte classificação:

- Obrigatório (M), o módulo deverá ser suportado.
- Condicional (C), o módulo será obrigatório em determinadas condições. Caso estas condições não estejam presentes o módulo não deverá estar presente.
- Opcional (U), a presença ou não do módulo é opcional.

3.1.3 Estrutura e codificação dos dados

Diversas questões precisam tratadas para que um conjunto de valores transmitidos como um fluxo de dados sejam corretamente interpretados pela entidade par de uma comunicação. O padrão DICOM define as informações trocadas durante uma operação como um conjunto de valores de atributos relacionados diretamente ou indiretamente com objetos de informação. Deste grupo de dados fazem parte os comandos utilizados pelo protocolo de serviços DICOM. A estrutura de dados que contém o valor de cada atributo é chamada elemento de dados. Um elemento de dados é a menor unidade de informação definida pelo padrão DICOM, representada por apenas uma entrada no dicionário de dados [DM05].

Existem quatro informações que caracterizam um elemento de dados. São elas:

- *Tag*: Identifica unicamente o elemento de dados dentro do dicionário de dados. É composta por um par de números, o primeiro identificando o grupo e o segundo o elemento.
- VR (acrônimo do inglês *Value Representation)*: Descreve o tipo dos dados e formato dos valores codificados nos elementos de dados. A tabela 6.2-1, PS 4, lista e

descreve a semântica todos os VRs definidos no DICOM. Dentre os VRs suportados merece destaque o SQ (Sequence of Items). Este VR permite o aninhamento de conjuntos de dados. Elementos de dados com este VR possuem como valor um conjunto de itens, onde cada itém contém outro conjunto de elementos de dados.

- VM (Value of Multiplicity): Especifica o número de valores que podem ser codificados em um mesmo elemento de dados.
- Tipo: Define se o elemento de dados é obrigatório, obrigatório sob certas condições ou opcional dentro de um conjunto de dados. Define ainda se o mesmo pode ter valor vazio ou não.

Destas informações, as três primeiras são únicas para todas instâncias de um elementos e são especificadas no dicionário de dados. O tipo de cada elemento de dados é definido pela semântica do objeto de informação representado ou pelo serviço DICOM utilizado.

Existem três tipos estruturação para um elemento de dados. Duas destas estruturas contém o VR do elemento de dados (VR explícito) representado explicitamente em um campo da estrutura mas diferem na maneira em que o tamanho de seus campos é expresso. A outra estruturação não contém o VR do elemento de dados (VR implícito). Todas as três estruturações contêm campos para a *tag* do elemento de dados, tamanho em bytes do valor armazenado e o próprio valor armazenado. A figura 5 exemplifica a estrutura de um elemento de dados DICOM

A presença ou não do campo VR no elemento de dados é dependente da sintaxe de transferência negociada durante o estabelecimento da conexão. A sintaxe de transferência define um conjunto de regras que permitem que entidades de aplicação DICOM negociem de maneira inequívoca os métodos de codificação aplicados no conjunto de dados transferidos. Entre os elementos de codificação definidos pela sintaxe de transferência está a estrutura dos elementos de dados, a ordenação de bytes e a informação se dados de *pixel* da imagem estarão comprimidos ou não. Desta forma, duas entidades de aplicação comunicantes podem acertar um conjunto de métodos de codificação suportados por ambas. DICOM define a sintaxe de transferência *Implicit VR Little Endian*, como a sintaxe padrão, que deve ser suportada e presente em toda requisição de associação por qualquer aplicação concordante com a especificação. A codificação definida pela sintaxe de transferência aplica-se ao grupo de dados que

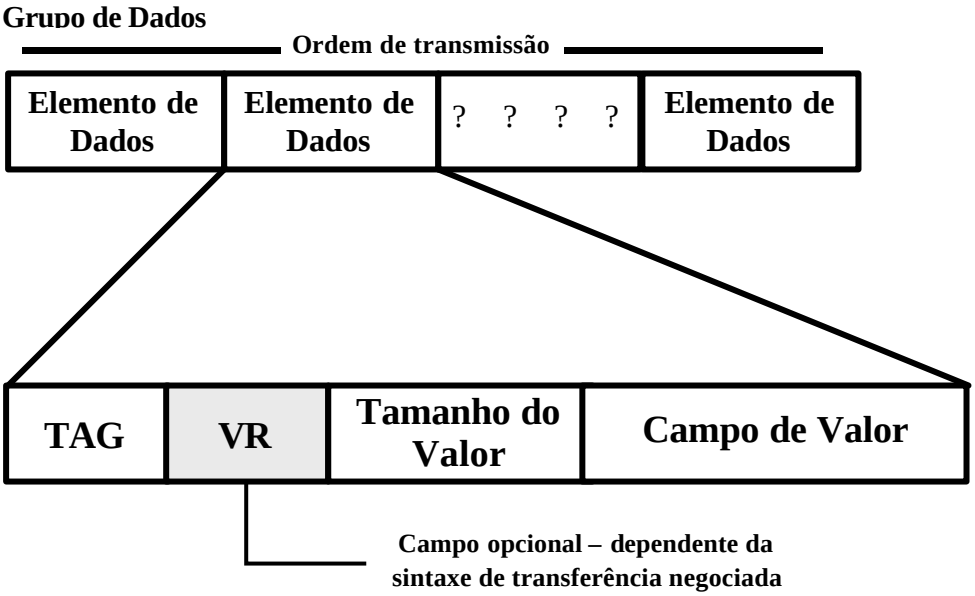

Figura 5: Estruturas do elemento de dados

representam os objetos de informação e não aos comandos usados pelos protocolos dos serviços. A codificação dos comandos possui uma estrutura fixa, definida em [DM07].

O conjunto de caracteres codificados padrão do DICOM é o ISO 8859, acrescido dos caracteres de controle LF,FF,CR e ESC. entidades de aplicações podem estender ou substituir o conjunto de caracteres padrão usando o atributo *"Specifc Character Set"*. Determinados VRs restringem ainda os caracteres válidos para subconjuntos do conjunto de caracteres padrão.

3.1.4 Classes de serviços

As especificações das classes de serviços DICOM fornecem uma definição abstrata de atividades do mundo real aplicáveis para intercâmbio e manipulação de informações médicas digitais. Atividades como armazenar, recuperar e verificar, que são aplicadas sobre um objeto do mundo real, como uma imagem de tomografia computadorizada, são representadas e semanticamente descritas nas definições das classes de serviços. Desta forma, existem definições de classes de serviço para verificação, armazenamento, consulta/recuperação, entre outras.

Uma entidade de aplicação DICOM pode desempenhar dois papéis ao executar uma atividade definida em uma classe de serviço durante uma associação: SCU (*Service*

Class User) ou SCP (*Service Class Provider*). No papel de SCU, a entidade de aplicação invoca operações na entidade par e recebe notificações da mesma. No papel de SCP a entidade de aplicação executa as operações requisitadas pela entidade par e invoca notificações para a mesma. Esta definição é similar ao conceito de comunicação cliente/servidor.

3.1.4.1 Serviços DIMSE

Para uma invocar ou executar operações ou notificações, uma entidade de aplicação DICOM utiliza um serviço ou grupo de serviços DIMSE (*DICOM Message Service Element*). Desta forma, tanto o SCU quanto o SCP durante uma comunicação são usuários dos serviços de mensagens DIMSE [DM07].

Os serviços DIMSE são especificados através do conceito de primitivas de serviços. Estas primitivas são divididas em quatro classes:

- *Request*. Uma entidade solicita alguma ação.
- *Indication.* Uma entidade deve ser informada sobre algum evento.
- *Response.* Uma entidade quer responder a um evento.
- *Confirm.* A resposta a uma solução anterior é enviada.

Desta forma, para uma entidade de aplicação no papel de SCU solicitar a execução de alguma operação para outra entidade no papel de SCP, ela emite uma primitiva *request* para seu provedor dos serviços DIMSE. Esta primitiva resultará em uma mensagem enviada para o provedor DIMSE do SCP. A receber a mensagem, o provedor DIMSE emite uma primitiva *indication* para o SCP. Após executar a operação solicitada, o SCP emite uma primitiva *response* para seu provedor DIMSE, que resultará em outra mensagem enviada para provedor DIMSE do SCU. Ao recebê-la, é emitida a primitiva *confirm* para o SCU, completando a seqüência.

Os procedimentos e regras de codificação para a construção das mensagens usadas para transportar requisições e respostas de operações são definidos pela máquina de protocolos DIMSE. A máquina de protocolos DIMSE recebe as primitivas de serviço *request* e *response* e constrói mensagens. Ela também recebe mensagens e repassa seu conteúdo para o usuário de seus serviços utilizando as primitivas *indication* e *confirmation*. Diferentes mensagens são construídas por diferentes serviços DIMSE e para diferentes primitivas.

Uma mensagem DIMSE é fragmentada em comandos e dados e encapsulados em um PDV (*Presentation Data Value*). É função da máquina de protocolos DIMSE tratar a fragmentação de cada mensagem DIMSE. Ela deve ainda gerenciar corretamente mensagens de diferentes associações e contextos de apresentação. Maiores detalhes sobre associação e contexto de apresentação serão apresentados na seção 3.2.5.

Devido a grande diferença entre a maneira com que operações sobre IOD compostos e normalizados são realizadas, os serviços DIMSE são divididos em duas classes. DIMSE-N é a classe de serviços aplicados sobre IODs normalizados e DIMSE-C é a classe de serviços aplicados sobre IODs compostos.

3.1.4.2 Classes SOP

A união de um IOD ou outra informação relacionada com a classe de serviço com um grupo de serviços DIMSE, criam o que é chamado de classe SOP (*Service Object Pair*). Uma classe de serviços pode ser definida como uma coleção de classes SOP relacionadas a uma mesma atividade.

A seleção de classes SOP é utilizada por entidades de aplicação para estabelecer um conjunto de capacidades acordantes. O IOD define a estrutura de dados e grupo de serviços DIMSE define um contexto de comunicação preciso para prover um serviço em nível de aplicação. A definição de uma classe SOP contém as regras e semântica que podem restringir o uso de serviços em um grupo DIMSE ou os atributos de IOD.

Duas classes SOP definidas por uma única classe de serviços podem diferir quanto ao IOD, grupo DIMSE ou ambos. Entretanto, dois IODs diferentes não podem estar presentes na mesma classe SOP. Cada classe SOP possui um identificador único chamado SOP UID. Outro identificador é criado sempre que uma instância de classe SOP é criada e armazenada como atributo desta instância. A identificação é própria mais para o uso interno do sistema do que para identificação humana e possui dois aspectos: a identificação da classe e a identificação do fornecedor.

3.1.5 Negociação de associação

Durante o estabelecimento de uma associação são negociados o tipo dos dados a serem transmitidos e como os dados serão codificados. A associação é sempre a primeira fase em uma interação DICOM e é realizada apenas uma vez. Três parâmetros

chaves negociados durante uma associação são o contexto de aplicação, o contexto de apresentação e itens de informação.

3.1.5.1 Contexto de aplicação

Um contexto de aplicação define explicitamente um conjunto de elementos de serviço de aplicação, opções relacionadas e qualquer outra informação necessária para cooperação entre duas entidades de aplicação DICOM . Em particular ele especifica o protocolo DIMSE utilizado pela camada de aplicação. É o nível mais alto de negociação. Apenas um contexto de aplicação deve ser oferecido por associação. Para esta versão do padrão DICOM, a 3.0, existe apenas um contexto de aplicação definido que é o “1.2.840.10008.3.1.1.1”.

3.1.5.2 Contexto de apresentação

Um contexto de apresentação define a estrutura e codificação dos dados durante uma associação. Ele representa um nível mais baixo de negociação e um ou mais contextos de apresentação podem oferecidos e aceitos durante uma associação. Uma entidade de aplicação pode aceitar ou rejeitar cada contexto de apresentação individualmente. Um contexto de apresentação é composto por três componentes: um contexto de apresentação ID, uma sintaxe abstrata e uma lista de um ou mais sintaxes de transferência.

A sintaxe abstrata é utilizada para identificar quais classes SOP e opções relacionadas serão suportadas durante uma associação. O nome de uma sintaxe abstrata é definido pela classe de serviço e consiste no UID da classe SOP. Opcionalmente é possível especificar para uma sintaxe abstrata um grupo meta classes SOP, que representa a união de um conjunto de classes SOP pertencentes a mesma classe de serviço. Com isso é possível negociar diversas classes SOP em uma mesma sintaxe abstrata.

Apenas uma sintaxe abstrata deve ser oferecida por contexto de apresentação. Entretanto, múltiplas sintaxes de transferências podem ser oferecidas por contexto de apresentação, mas apenas uma poderá ser aceita. Para classe ou meta classe SOP, um contexto de apresentação deve ser negociado de maneira que este contexto de apresentação suporte a sintaxe abstrata associada e uma sintaxe de transferência

adequada. Contextos de apresentação são identificados dentro do escopo de uma associação por um contexto de apresentação ID.

3.1.5.3 Itens de informação

Duas entidades de aplicação podem negociar, durante o estabelecimento de uma associação, várias funcionalidades relacionadas ao protocolo DIMSE. Entre elas estão:

- o tamanho máximo de dados contendo fragmentos de mensagem DIMSE que serão enviados de cada vez.
- identificação da implementação e nome de versão
- questões para operações assíncronas.

3.2 O suporte à segurança no padrão DICOM

Um dos principais objetivos que motivaram a criação do padrão DICOM ainda não foi completamente alcançado. Juntamente com a interoperabilidade entre aparelhos de diversos fabricantes e suporte à implantação de PACS em ambientes radiológicos o padrão DICOM durante seu surgimento visava "permitir a criação de banco de dados informações para diagnósticos distribuídos geograficamente que possam ser examinado por uma ampla variedade de dispositivos" [DM01]. Apesar das amplas possibilidades pelo de fato do DICOM ser um padrão de fato em nível mundial, o crescente interesse pela Telemedicina por parte do mercado e organizações de saúde pública e a difusão das redes de alta velocidade, a utilização de bancos de dados geográficos de imagens no padrão DICOM ainda é bastante incipiente, praticamente inexistente.

A especificação tardia sobre a utilização de mecanismos segurança de maneira efetiva por parte do padrão DICOM foi um dos fatores que inibiram sua utilização fora contexto das Intranets. O tratamento às diversas questões que envolvem uma transação segura entre duas entidades foi adicionado apenas em 2000, através dos suplementos 31, 41 e 51, e incorporados ao padrão base em 2001.

3.2.1 A abordagem sobre segurança

O padrão DICOM aborda as questões relativas a segurança de dados através da especificação de perfis de seguranças, aos quais aplicações podem declarar conformidade. Diversas questões específicas são deixadas para decisões específicas de implementação. O padrão apenas provê mecanismos que poderão ser utilizados para a

implantação da segurança em diferentes níveis. Por exemplo, não é definida qualquer espécie de política de controle de acesso pelo padrão DICOM, mas são especificados os meios tecnológicos para que as entidades de aplicação envolvidas em um processo de comunicação possam trocar informação suficiente para implementação de políticas restritas para acesso a recursos do sistema.

Quando duas entidades de aplicação concordam em trocar informações através do padrão DICOM através de negociação durante a etapa de associação, elas estão essencialmente concordando em algum nível de confiabilidade entre si. Primariamente, entidades de aplicação confiam que seus pares durante uma comunicação irão manter a confidencialidade e integridade dos dados sob seu controle. Entidades de aplicação podem não confiar no canal de comunicação através do qual elas irão comunicar com outras entidades de aplicação. Desta maneira, o padrão provê mecanismos para que entidades de aplicação autentiquem de maneira secura ser par na comunicação, detectem qualquer mudança não-autorizada nas mensagens trocadas, e protejam a confidencialidade das mensagens enquanto elas atravessam canais de comunicação.

Diversas premissas para o estabelecimento de um contexto seguro são assumidas pelo padrão como implementados pelas entidades de aplicação participantes. Estas questões são de responsabilidade da implementação de uma entidade de aplicação e do ambiente onde a mesma está inserido. São elas:

- Proteção dos dados. É assumido pelo padrão que durante a comunicação DICOM entre duas entidades de aplicações, que as mesmas implementam apropriadas políticas de segurança, suporte à auditoria, proteção física, manutenção da integridade e confidencialidade dos dados.
- Controle de acesso. O padrão DICOM assume que Entidades de Aplicação podem seguramente identificar usuários locais, bem como seus respectivos papéis representados no sistema. É importante ressaltar que o termo usuários pode se referir também a entidades abstratas como organizações e partes de uma equipamento. Quando Entidades de Aplicação concordam em intercambiar informações via DICOM elas também poderão intercambiar informações sobre usuários da entidade de aplicação através de certificados digitais enviados durante a criação do canal seguro.

- Identificação de usuários. O padrão DICOM assume que entidades de aplicação possuem meios de determinar se os proprietários (por exemplo, paciente, instituição) da informação têm usuários particulares autorizados, ou classes de usuários para acessar a informação. Além disso é assumido que cada autorização poderá ser considerada no processo de controle de acesso empregado pela entidade de aplicação. Neste ponto, este padrão não considera a maneira que uma autorização poderá ser trocada entre entidades de aplicação, apesar deste ser considerado um tópico futuro.
- Controle de certificados. É assumido que entidade de aplicação usando TLS (*Transport Layer Security*) pode obter e validar seguramente certificados X.509 para os usuários da entidade de aplicação.

3.2.2 Perfis de Segurança

O padrão DICOM especifica um conjunto de perfis de segurança relacionados a diversos aspectos que compõem um contexto seguro para utilização e intercâmbio de informações médicas. Implementações que almejem agregar segurança à suas transações DICOM podem escolher seguir um ou mais destes perfis de acordo com o nível de segurança desejado. Existem quatro grupos de perfis de segurança distintos, embora exista uma interdependência em alguns aspectos entre eles. Cada grupo pode compreender outros perfis que abordam questões específicas. São eles:

- Perfil de uso seguro
- Perfil de conexão de transporte seguro
- Perfil de assinatura digital
- Perfil de segurança em armazenamento em mídia

A seguir serão descritas as principais questões dos três primeiros perfis, que são pertinentes ao escopo deste trabalho.

3.2.2.1 Perfil para uso seguro

Este perfil de segurança aborda três aspectos para possibilitar a aplicação de políticas de segurança sobre a utilização de instâncias SOP, cada um definido em um perfil específico. São eles o armazenamento eletrônico via rede de comunicação, o uso de assinatura digital e o uso de assinatura digital para preservação de bits.

O perfil para armazenamento eletrônico é designado para que entidades de aplicação possam rastrear e verificar o *status* de instâncias SOP, tanto o conjunto de dados original quanto as cópias subseqüentes. As regras são requeridas durante a utilização da classe de serviço de armazenamento. Elas regem principalmente a utilização do atributo *SOP Instance Status*. Este atributo pode possuir seguintes valores em uma instância SOP mantida por entidade de aplicação :

- OR (Original): Este valor é atribuído quando uma instância é criada, mas antes dela ser certificada para uso em diagnósticos. Apenas uma entidade de aplicação pode manter uma instância SOP com este *status*. Caso uma entidade de aplicação envie uma instância com este *status* para outra entidade, ela deve apagar tua cópia local após a transmissão ou mudar seu *status* para NS. A transmissão deve ser realizada usando o perfil para conexão de transporte seguro, através da classe de serviço *Storage Commitment* (esta classe de serviço requer que o SCP explicitamente confirme o armazenamento da instância SOP recebida).
- AO (*Authorized Original*): Este valor é atribuído após a entidade de aplicação certificar-se que a instância SOP é adequada para diagnóstico, quando o *status* é modificado de OR para AO. Assim como OR, apenas uma entidade de aplicação pode manter uma instância SOP com este *status*. Para transmissão de uma instância SOP com este *status*, a entidade de aplicação deve ou atribuir o *status* para AC ou NS na cópia enviada ou modificar o *status* de sua cópia para AC.
- AC (*Authorized Copy*): Este valor define que uma instância SOP foi obtida de uma origem segura. Uma entidade pode de aplicação pode transmitir uma cópia de instância para outra entidade mantendo este *status* ou modificando-o para NS.
- NS (*Not Specified*): A transmissão de instâncias SOP com *status* OR, AO ou AC requer a comunicação seja feita entre duas entidades acordantes com o perfil para armazenamento eletrônico seguro e a utilização de conexão de transporte seguro. Uma entidade deve modificar o *status* de uma instância SOP para NS sempre que realizar uma transmissão não atender estas especificações. O *status* NS indica que não existe garantia quanto a autenticidade da instância SOP.

Uma entidade de aplicação em conformidade com este perfil pode modificar apenas os atributos *SOP Instance Status, SOP Authorization Data and Time,*

Authorization Equipment Certification Number, e *SOP Authorization Comment Atributes* em uma instância SOP desde sua criação.

O perfil básico para uso de assinatura digital apenas define regras básicas para a correta utilização de assinatura digital. Entre elas estão as validações de assinaturas digitais de outras entidades de aplicação e a remoção da assinatura digital caso a mesma torne-se inválido devido a alguma modificação nos dados da instância SOP.

O perfil para uso de assinatura digital com preservação de *bits* determina as regras para a manutenção da integridade de dados de uma instância SOP. Implementações que seguem este perfil devem transmitir uma cópia exatamente igual, *bit-a-bit* da instância SOP armazenada. Isto inclui manter a ordem em itens de seqüência, não remover ou modificar qualquer elemento de dados ou assinatura digital. A sintaxe de transferência utilizada deve utilizar VR explícito.

3.2.2.2 Perfil para conexão de transporte segura

Este perfil é aborda dois protocolos de segurança para conexão de transporte, o ISCL (*Integrated Secure Communication Layer*) e o TLS (*Transporte Layer Security*). O ISCL é um protocolo para comunicação de dados segura desenvolvido pela *Medis-DC* (*Medical Information System Development System*), uma organização de pesquisa e desenvolvimento de sistemas de informação médica sediada no Japão. O ISCL é direcionado principalmente para prover serviços de comunicação entre *smartcards* e sistemas de gerenciamento de chaves. Provê interfaces para acesso a dispositivos leitores e escritores em *smartcards* e também para serviços da camada de transporte da pilha de protocolos TCP/IP. Não será detalhado mais sobre o perfil para o ISCL porque o mesmo foge ao escopo deste trabalho.

O protocolo TLS corresponde a versão 3.0 do SSL (*Secure Socket Layer*). O TLS é um serviço localizado acima da camada de transporte responsável por gerenciar um canal de comunicação segura, com mecanismo de encriptação de dados em uma comunicação cliente/servidor.

É fortemente recomendado pelo padrão DICOM que sistemas que suportem que suportem este perfil usem a porta TCP 2762 registrada como “dicom-tls” para implementar o *upper layer* sobre TLS.

A declaração de conformidade deverá indicar quais mecanismos a implementação suporta para gerenciamento de chaves criptográficas. O perfil não especifica como uma

conexão TLS é estabelecida, ou o significado de qualquer certificado trocado durante a autenticação. Uma vez que a Entidade de Aplicação tenha estabelecido uma conexão de transporte seguro, então uma associação em nível de *upper layer* poderá utilizar um canal seguro de dados.

Pode existir relação entre o tamanho de PDU e o tamanho dos registros TLS que poderá influenciar na eficiência do transporte de dados. O tamanho máximo de um registro TLS e ISCL é menor que o tamanho máximo permitido para um PDU.

A tabela 1 mostra a configuração mínima para cada mecanismo:

Configuração Suportada	Mecanismo TLS
Autenticação de Entidade	Certificados RSA
Troca de chaves	RSA
Integridade de Dados	SHA
Privacidade	DES Triplo EDE,CBC

Tabela 1 Características mínimas para os mecanismos TLS

3.2.2.3 Perfil para assinatura digital

Este perfil aborda a questões para a implementação e utilização da assinatura digital. É dividido em diretivas base para a implementação da assinatura digital, e para a criação e autorização de uma assinatura digital.

O perfil base para assinatura define linhas gerais para o uso da encriptação RSA (*Rivest-Shamir-Adleman*) de um MAC (*Message Authentication Code*) para gerar uma assinatura digital. Não é especificado por este perfil qualquer conjunto de elemento de dados em particular que devam ter a assinatura digital aplicada aos mesmos.

Para a geração do MAC, o criador de uma assinatura digital deve usar uma das seguintes funções *hash*: RIPEMD-160, MD5 ou SHA-1. O elemento de dados MAC *Algorithm* (0400,0015) deve portanto possuir como valor um dos nomes das funções *hash* acima. Todas as entidades que validem assinaturas digitais devem ser capazes de usar um MAC gerado por uma das funções *hash* especificadas.

A chave pública associada com a chave privada RSA, seja de uma entidade de aplicação ou produtor do aparelho, devem ser transmitidas em um certificado X.509. O valor do atributo *Certificate Type* deve ser “X509_1993_SIG”.

Para geração de uma assinatura digital em uma instância SOP recém-criada é definido um perfil específico. De acordo com este perfil os seguintes atributos deverão estar inclusos na instância SOP:

- o UID da Classe SOP e da instância.
- a data e horário da classe SOP se estiverem presentes.
- o UID da instância de Estudo e da Série.
- os atributos do módulo *General Equipment* se presentes.
- os atributos dos módulos *Overlay Plane*, *Curve* e *Graphic Annotation* que estiverem presentes.
- os atributos dos módulos *General Image* e *Image Pixel* que estiverem presentes.
- os atributos dos módulos *SR Document* e *SR Document Content* que estiverem presentes.
- os atributos dos módulos *Waveform* e *Waveform Annotation* que estiverem presentes.

Outro perfil é destinado para técnico ou clínico que aprova uma instância SOP e pretende criar uma assinatura digital para a mesma. A assinatura digital produzida em conformidade com este perfil garante que, desde que o técnico ou clínico fez sua aprovação, o conjunto de dados de imagem de uma instância da classe SOP não sofreram alteração.

Como requisito mínimo, os seguintes atributos deverão estar inclusos em instância SOP para geração de uma assinatura digital em conformidade com este perfil:

- UID da Classe SOP e da instância.
- UID da instância de Estudo e da Série.
- quaisquer atributos cujos valores possam ser verificados pelo técnico ou clínico, ou seja, valores que possam ser visualizados pelo mesmo.
- qualquer atributo dos módulos Overlay Plane, *Curve* ou *Graphic Annotation* que estiverem presentes.
- qualquer atributo dos módulos *General Image* e *Image Pixel* que estiverem presentes.
- quaisquer atributos dos módulos *SR Document General* e *SR Document Content* que estiverem presentes.
- quaisquer atributos dos módulos *Waveform* e *Waveform Annotation* que estiverem presentes.

A assinatura digital poderá ser criada usando a metodologia descrita em no perfil base para assinatura digital RSA. A entidade de aplicação deverá determinar a identidade do técnico ou clínico e obter seu certificado através de mecanismo á critério da organização, como *login* ou *smartcard*.

Este perfil requer a posse de um par de chaves pública por parte de seus usuários, aplicando portanto uma política de segurança mais restrita que o perfil para criação de assinatura digital. Embora, a própria assinatura digital baste para autenticação, o perfil anterior permite a utilização de um mesmo par de chaves por diversos usuários, podendo ser utilizado para autenticação de uma entidade de aplicação ou até mesmo de uma organização.

4. A ARQUITETURA CORBA (*COMMON OBJECT REQUEST BROKER ARCHITECTURE*)

A arquitetura CORBA é um *framework* popular para o desenvolvimento de aplicações baseadas em objetos distribuídos. É apoiada desde sua criação por importantes segmentos da área de informática e empresas interessadas em sistemas de informação. Apesar de ainda não estar amplamente difundida como *middleware* de transmissão de dados e enfrentar concorrência de outras tecnologias como SOAP (*Simple Object Access Protocol*), CORBA é uma tecnologia madura, e suportada pelos principais ambientes de desenvolvimento. Os motivos para a escolha da arquitetura CORBA como *middleware* de comunicação dos clientes com o sistema federados são apresentados na seção 8.1.

O objetivo deste capítulo é apresentar os principais conceitos da tecnologia CORBA. Não será realizada entretanto, uma descrição abrangente e aprofundada de toda arquitetura CORBA. O intuito é apresentar a arquitetura de uma maneira geral, focando em alguns pontos pertinentes ao escopo deste trabalho. Informações detalhadas sobre a arquitetura CORBA podem ser encontradas em [HV99], [OM03] e [SJ00]. A estrutura deste capítulo é a que segue. Na seção 4.1 será apresentados o consórcio OMG e um resumo da evolução da concepção da especificação CORBA dentro de seu âmbito. Na seção 4.2 é apresentada a arquitetura OMA, uma arquitetura conceitual que engloba a especificação CORBA. Na seção 4.3 são apresentados os diversos componentes e conceitos que juntos compõem CORBA. A seção 4.4 apresenta os serviços de objetos, facilidades da arquitetura OMA que são independentes de domínio de aplicação. Finalmente na seção 4.5 são apresentados os domínios OMA, conjuntos de interfaces padronizadas direcionadas a um segmento de mercado específico, com destaque especial ao domínio destinado à área médica.

4.1 O consórcio OMG (*Object Management Group*)

A OMG foi criada em abril de 1989 por onze grandes companhias relacionadas à área de tecnologia, incluindo *3Com*, *American Airlines*, *Canon*, *Hewlett-Packard*, *Philips* e *Sun Microsystems.* Atualmente, este consórcio possui com algo em torno de 800 empresas associadas, sendo o maior do mundo.

A principal finalidade da criação da OMG foi estabelecer uma padronização para o desenvolvimento de softwares, seguindo o paradigma da orientação a objetos. As atividades concentram-se na elaboração de especificações detalhadas no intuito de fornecer um ambiente de trabalho comum para o desenvolvimento de aplicações independentes de plataformas de hardware e sistemas operacionais. A OMG não produz *softwares*, apenas especificações. Outra especificação criada pela OMG bastante conhecida e utilizada é a *Unified Modeling Language* (UML) uma linguagem formal para análise, modelagem e documentação de sistemas de informação.

A primeira especificação chave da OMG foi a OMA (*Object Management Architecture*), em 1990. A OMA é uma arquitetura orientada a objetos para prover interações entre aplicações independentes de plataforma. Desde então a OMA tem função de servir como uma arquitetura "guarda-chuva" para todas as demais especificações da OMG.

A primeira versão de CORBA foi publicada pela OMG em 1991, era dirigida unicamente para programas escritos em C, que era a linguagem de programação dominante para sistemas distribuídos na época. Porém, as dificuldades em utilizar um sistema designado para objetos distribuídos com uma linguagem sem facilidades para aplicar a orientação a objetos tornaram-se evidentes. Logo diversas companhias começaram a desenvolver adaptações próprias de CORBA para C++, o que conduziu inexoravelmente à incompatibilidade entre as diferentes versões.

Finalmente a OMG emitiu um RFP (*Request for Proposal*) para padronizar o mapeamento para C++ e outras linguagens orientadas a objeto como *Smalltalk*, ou que unificou, após uma etapa de adaptação, a direção e controle sobre o CORBA. A versão 2.0 do CORBA, além do mapeamento para linguagens orientadas a objeto , o protocolo IIOP (*Internet Inter-ORB Protocol*). Com ele, sistemas desenvolvidos utilizando diferentes implementações CORBA podiam finalmente interoperar entre si. As versões 2.2 e 2.3 trouxeram o POA (*Portable Object Adapter*), que solucionou problemas de compatibilidade do lado do servidor e a possibilidade de passar objetos por valor, respectivamente. A última especificação CORBA lançada pela OMG é a 3.0. Ela traz o suporte a controle de qualidade de serviço e novos modelos de comunicação para suporte à transmissão através de *firewalls*.

4.2 A arquitetura OMA

A OMA foi a primeira especificação produzida pela OMG para acomodar uma extensa variedade de sistemas distribuídos. Pode-se considerar a OMA como um conceito arquitetural de alto-nível enquanto que CORBA seria sua principal aplicação. A OMA baseia-se em dois modelos relacionados para descrever como objetos distribuídos e sua iteração entre si podem ser especificados de diferentes maneiras e de modo independente de plataforma. São eles o Modelo de Objetos e o Modelo de Referência.

O Modelo de Objetos, que define como as interfaces entre objetos distribuídos em um ambiente heterogêneo são descritas. Ele segue o padrão da orientação a objetos, usando como conceitos chave a encapsulação e o envio de mensagens.

O Modelo de Referência caracteriza interações entre estes objetos. Ele fornece categorias de interfaces que são agrupamentos gerais de interfaces de objetos. Todas estas categorias são ligadas conceitualmente por um ORB (*Object Request Broker*). Geralmente um ORB habilita a comunicação entre clientes e objetos, ativando de maneira transparente estes objetos que não estão sendo executados quando as requisições são enviadas para eles. O ORB também provê uma interface que pode ser usada diretamente por clientes assim como objetos. A figura 1 descreve este Modelo de Referência.

Serviços de objetos são interfaces independentes do domínio, ou orientadas horizontalmente, usadas por várias aplicações de objetos distribuídos. Por exemplo, todas aplicações devem obter referências para objetos que eles pretendem usar. O Serviço de Nomes OMG é um objeto de serviço que permite que as aplicações consultem e descubram estas interfaces. Objetos de serviços são normalmente considerados parte do núcleo da infra-estrutura de computação distribuída.

Interfaces de domínio possuem um papel similar à categoria de serviços de objeto, com a diferença que é dependente de domínio, ou orientadas verticalmente. Por exemplo, existem interfaces usadas em aplicações de atenção à saúde que exclusivas para este segmento como o *Person Identification Service*. Outras interfaces são específicas para finanças, telecomunicações, entre outras áreas.

Interfaces de aplicação são desenvolvidas especificamente para uma determinada aplicação. Elas não são padronizadas para OMG. Entretanto, caso certas interfaces de

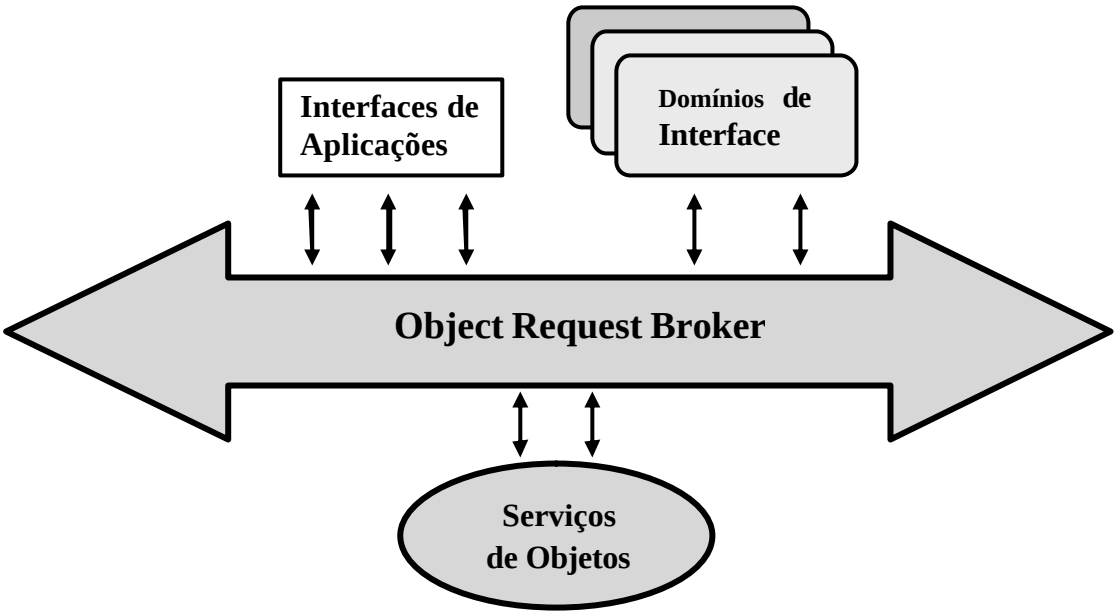

Figura 6: Categorias de interfaces OMA

aplicação se torne comum em muitas aplicações, eles podem tornar-se candidata à padronização em uma das categorias de interfaces.

4.3 A plataforma CORBA

A plataforma CORBA representa o *middleware* para que aplicações distribuídas possam interajam entre si de maneira transparente quanto aos agrupamentos possíveis de sistemas operacionais, hardware, protocolos de comunicação e linguagens de programação. Especificamente, CORBA é um conjunto de especificações que detalham sobre as interfaces entre o ORB e os demais conjuntos de objetos componentes da arquitetura OMA. Em outras palavras, CORBA seria um *zoom* sobre as categorias de interfaces e o *broker*.

O *framework* CORBA é formada pelos seguintes componentes:

- ORB
- OMG IDL
- Interfaces para invocação de procedimentos estáticos: *Static Stub Interface* e S*keleton Static Interface*.
- Interfaces para invocação de procedimentos dinâmicos: DII (*Dinamic Invocation Intefarce*) e DSI (*Dinamic Skeleton Interface*).

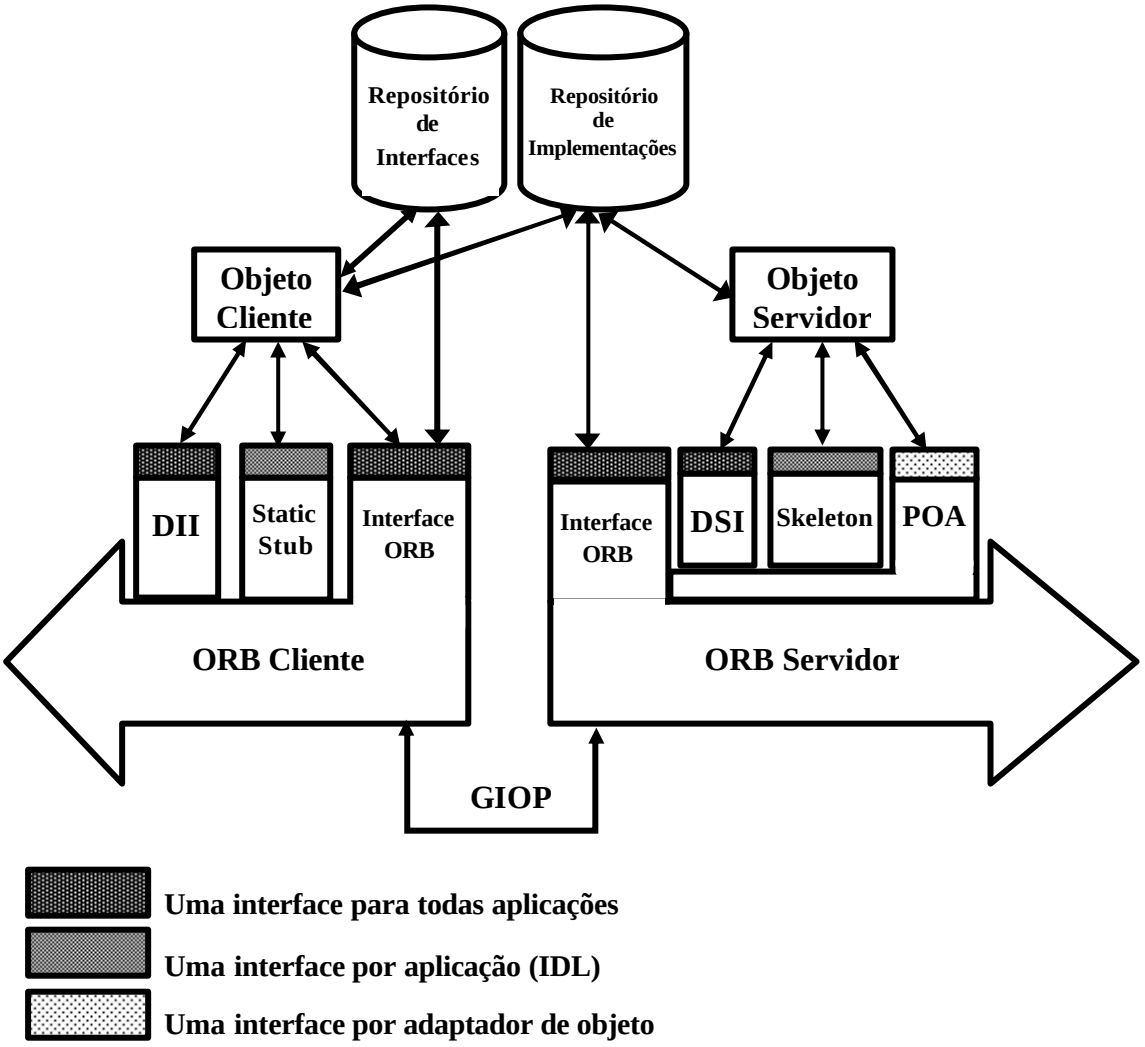

Figura 7: Arquitetura CORBA

- Repositórios de interface e implementação
- Adaptadores de objetos: POA (*Portable Objetc Adapter*)
- Protocolos inter-ORB: IIOP

A figura 5 apresenta estes componentes e a integração entre eles. Um conceito importante da arquitetura CORBA que não está representado na figura é a referência para objetos remotos . A primeira etapa em interação de objetos via CORBA consiste em obter esta referência.

A seguir cada um destes componentes será detalhado apropriadamente.

4.3.1 ORB (Object Request Broker)

O ORB é o componente central da arquitetura OMA, funcionando como um barramento de comunicação entre os demais componentes. Como já mencionado anteriormente, CORBA especifica as interfaces de programação para que aplicações possam utilizar os serviços de um ORB.

O ORB é responsável pelas seguintes funções dentro da arquitetura OMA/CORBA:

- Identificação do método a ser chamado no objeto servidor.
- Codificação dos valores da representação local dos valores para uma representação em comum.
- Entrega da requisição e mensagens resultantes para os objetos comunicantes.
- Sincronização entre requisições e repostas
- Ativação e desativação de objetos persistentes
- Controle de exceção para reportar diversas falhas para clientes e servidores.

Todas estas funcionalidades visam promover a transparência para a comunicação entre objetos. O ORB esconde os seguintes detalhes de comunicação durante a interação entre dois objetos:

- Localização: O cliente não precisar saber onde o objeto servidor reside.
- Implementação: O cliente não precisar saber como o objeto servidor é implementado, qual linguagem de programação ou *script* utilizada ou detalhes do sistema operacional ou hardware onde ele reside.
- Estado de execução do objeto: O cliente não precisa saber se o objeto servidor encontra-se ativado, desativado ou pronto para aceitar requisições. O ORB servidor transparentemente inicia o objeto se necessário antes de enviar requisições para ele.
- Mecanismo de comunicação: O cliente não precisa saber quais os mecanismos de comunicação que o ORB utiliza.

A concentração de todas as questões de comunicação em único objeto proporciona um grande aumento de flexibilidade e na facilidade para a realização de operações remotas. Com isso é possível uma centralização de diversas etapas de uma interação entre cliente e servidor em um único objeto ou programa. Em vez de utilizarem-se clientes específicos para utilizarem serviços específicos, tem-se que um cliente poderá utilizar diferentes serviços fornecidos por diferentes servidores. O antigo

cenário um-para-um entre cliente-servidor transforma-se em muitos-para-muitos com a arquitetura CORBA. Desta forma, é possível construir-se estruturas bem mais complexas, mas ao mesmo tempo mais simples de implementar e administrar.

As especificações da OMG definem os objetos apenas em termos de interfaces IDL e sua semântica e não em termos de implementação. Esta abordagem permite grande flexibilidade no projeto de uma particular implementação de CORBA, o que é importante porque diferentes ambientes freqüentemente apresentam diferentes restrições e requisitos do ORB. Por exemplo a implementação de um ORB a ser executado em um servidor será bastante diferente da implementação de um ORB designado para um sistema embarcado. Mas apesar da diferenças na implementação, as especificações CORBA asseguram que todos ORBs em conformidade com as mesmas poderão se comunicar.

4.3.2 OMG IDL (Interface Definition Language)

Em CORBA, qualquer interface de objeto distribuído é definida em IDL. A definição de interface especifica as operações que um objeto esta preparado para executar, dos parâmetros de entrada e saída requeridos, e qualquer exceção que possam ser geradas durante a execução. Esta interface constitui um contrato entre o objeto, que teve suas interfaces definidas em IDL, e seus clientes, que irão utilizar esta interface para criar e executar invocações.

O seguinte exemplo ilustra a sintaxe da linguagem IDL:

```
interface Conta {
        void deposit ( in unsigned long amount );
        void withdraw (in unsigned long amount );
        long balance ( );
}
```

Na arquitetura CORBA, o ORB cliente intercepta qualquer invocação de método a partir de uma referência remota de um objeto e a repassa para o ORB que reside no computador onde se encontra o objeto remoto. O ORB então utiliza a interface pré-compilada definida em IDL referente ao método invocado para transmissão dos argumentos. No computador remoto, o ORB residente realiza as conversões que se fizerem necessárias para a arquitetura e linguagem de programação utilizada e repassa os argumentos no formato adequado para o objeto, causando a execução do método

relacionado. O processo no objeto servidor será transparente para o objeto cliente, que apenas receberá o resultado retornado pelo objeto remoto invocado. No ambiente do objeto cliente o ORB local irá então, também baseado na interface IDL sobre o objeto remoto realizar as conversões necessárias sobre o resultado retornado.

O procedimento de conversão dos dados entre a representação local e a representação externa dos dados através da definição IDL é chamada *marshalling*. A representação externa do CORBA é chamada de *Common Data Representation* (CDR).

A concepção da arquitetura CORBA assegura a encapsulação de um objeto e também uma segurança inerente: clientes apenas poderão acessar um objeto da maneira como ele é definido por sua interface em IDL; não há na arquitetura CORBA qualquer meio de passar pela interface IDL e ganhar acesso à implementação do objeto.

A arquitetura CORBA separa a interface, escrita em OMG IDL, da implementação, à qual será escrita em alguma linguagem de programação. Criar a interface de um objeto seguindo a especificação CORBA e implementar o código deste objeto representa etapas separadas na criação de uma aplicação CORBA, embora em alguns ambientes o ORB crie uma interface IDL automaticamente a partir do código em linguagem de programação nativo.

IDL é uma linguagem fortemente "tipada" ou seja, cada variável deve ter definida seu formato previamente. Isto permite que o ORB trate de questões como diversidade de formato de dados e ordenação de bytes em uma rede heterogênea, comuns em aplicações distribuídas. O OMG IDL já um padrão ISO e ITU-T; de fato a maneira mais completa para referi-se à interface padrão CORBA é OMG/ISO IDL. O número do padrão ISO é 14750, ele corresponde ao IDL especificado em CORBA 2.2.

4.3.3 Invocação Estática: *Stubs* e *Skeletons*

O mecanismo utilizado para chamada de procedimento remoto de CORBA baseia-se no RPC *(Remote Procedure Call).* No modelo RPC a comunicação entre o cliente do procedimento e servidor deste procedimento possui dois atores: *stub,* que é porção de software responsável por encaminhar a requisição de procedimento remoto do cliente, e o *skeleton* que é a porção de software do servidor responsável por gerenciar as invocações de procedimentos que chegam, e entregá-los aos objetos que os implementam. A grande diferença entre abordagem utilizada em RPC e CORBA é

enquanto na primeira cada *stub* encontra individualmente o *skeleton* correspondente na segunda toda esta tarefa é delegada ao ORB.

O despacho de invocações de procedimentos remotos através de *stubs* e *skeletons* é chamado de invocação estática dentro da terminologia CORBA. Um *stub* é essencialmente um *proxy* para o objeto alvo. IDL *stubs* e *skeletons* são construídos estaticamente na aplicação cliente e no objeto servidor. Eles são criados pelo compilador IDL

A principal função dos *stubs* e *skeletons* é o mapeamento entre a definição de interface em OMG IDL e linguagem fonte. O OMG IDL é uma linguagem declarativa, e não pode ser usada diretamente para implementar aplicações distribuídas. Portanto, mapeamentos de linguagem determinam como uma declaração em IDL é convertida para as facilidades específicas de uma linguagem de programação. Aspectos importantes deste mapeamento é a conversão das interfaces (e outros pseudo-objetos), tipos, e objetos, para as estruturas correspondentes da linguagem de programação fonte. Desta forma, *stubs* e *skeletons* formam a base da implementação real dos objetos em sua respectiva linguagem de programação.

4.3.4 O Repositório de Interfaces

Juntamente com a criação do *stub* cliente e o *skeleton* no servidor, CORBA possibilita que o ORB armazene as definições de interface em IDL em uma estrutura chamada Repositório de Interfaces. Este banco de interfaces é um conceito bastante importante na modelagem distribuída CORBA. Este repositório poderá estar disponível não apenas para o ORB local, mas também para clientes, implementações de objetos, e utilitários como *browsers* hierárquicos e ferramentas de depuração de código. Desta forma, as definições de interface em IDL poderão ser adicionadas ao IR, apagadas, modificadas ou recuperadas; e árvores de heranças poderão ser construídas, combinando-se diversos componentes básicos para formas estruturas mais complexas. Esta árvore poderá posteriormente ser percorrida para determinar o tipo exato de cada objeto.

O Repositório de Interfaces fornece um mecanismo de persistência para objetos, permitindo a consulta e recuperação de interfaces de objetos remotos em tempo de execução. Com clientes possuem uma alternativa aos *stubs* estáticos utilizando a

interface para a invocação dinâmica de objetos, o DII. [SJ00] cita três maneiras que um ORB pode usar Repositório de Interfaces diretamente:

- Para fornecer interoperabilidade entre diferentes implementações de ORBs.
- Para checagem de assinaturas de requisições em tempo de execução, a fim de verificar se uma requisição foi emitida através de um DII ou através de um *stub*.
- Para a correção de grafos de herança.

Para clientes e usuários [SJ00] cita os seguintes exemplos:

- Controlar instalações e distribuições de definições de interfaces em um ambiente de rede.
- Implementação de componentes em uma ambiente CASE (por exemplo, um browser de interfaces).
- Para apresentar e modificar definições de interfaces ou outras informações armazenadas em IDL durante o processo de desenvolvimento.
- Para compilar *stubs* e *skeletons* diretamente do Repositório de Interfaces ao invés de arquivos IDLs, desde que toda a informação requerida para compilação está contida em ambos formatos.

4.3.5 Invocações e despachos dinâmicos

CORBA suporta duas interfaces para invocação dinâmica de procedimentos: a d*inamic invocation interface*, DII (), que provê suporte para requisições de clientes e a *dinamic skeleton interface* (DSI), que provê suporte para o despacho das requisições do lado do servidor. O DII e o DSI podem ser considerados como um *stub* genérico e um *skeleton* genérico, respectivamente. Cada um é uma interface mantida diretamente pelo ORB, e não são dependentes de qualquer interface IDL em particular dos objetos sendo invocados.

O DII oferece bastante flexibilidade para o programador em relação à utilização de *stubs* estáticos apesar de ter uma performance inferior pois o repositório de interfaces tem que ser consultado a cada invocação remota[1]. De maneira análoga, o DSI permite

[1] O ambiente de desenvolvimento *VisualWorks* que utiliza a linguagem *Smalltalk* em sua implementação CORBA, o *Distributed Smalltalk*, utiliza os métodos de invocação dinâmica como padrão, com o repositório de interfaces sendo utilizado a cada invocação tanto do lado do cliente quando do servidor.

que servidores sejam escritos sem que os *skeletons* para os objetos sejam compilados estaticamente no programa.

4.3.6 Adaptadores de objetos

Adaptadores de objetos agem como a ligação entre a implementação dos objetos CORBA e o ORB. Eles são responsáveis por criar as referências para os objetos e assegurar que cada objeto servidor é "encarnado" em uma real implementação do objeto. Na terminologia CORBA a implementação real do serviço oferecido pelo objeto CORBA é chamada de *servant*. Pode-se imaginar um objeto CORBA como um *wrapper* para o *servant* que irá executar a operação e o adaptador de objeto como o meio que o *wrapper* irá utilizar para acionar o *servant*.

A funcionalidade fornecida pelo ORB através de um adaptador de objetos freqüentemente também inclui a interpretação de referências para objetos, invocação de métodos, interações de segurança, ativação e desativação de objetos, mapeamento de referências para implementações e registro de implementações. Sem os adaptadores de objetos, as implementações dos objetos precisariam conectar-se diretamente ao ORB para receber as requisições, o que aumentaria bastante a complexidade da interface OBR e dificultaria a padronização.

CORBA permite múltiplos adaptadores de objetos em um ORB, tratando uma grande variedade de características de objetos, como granularidade, tempo de vida, políticas e estilos de implementação. Até a versão 2.1, CORBA especificava apenas um adaptador de objetos, o *basic object adapter* (BOA). Entretanto, no intuito de manter a generalidade para suportar diversas linguagens de programação, a especificação BOA acabou sendo vaga ou incompleta em alguns pontos. Isto resultou em problemas de portabilidade entre diferentes implementações.

Desde a versão 2.2, CORBA especifica o *portable object adapter* (POA) como substituto ao BOA. O POA suporta uma grande variedade de interações entre objetos CORBA e linguagens de programações ao mesmo tempo em que mantém a portabilidade da aplicação. Desta forma, a especificação BOA foi removida de CORBA e POA é novo padrão para adaptador de objetos.

4.3.7 Referência para objetos remotos

Alguns objetos possuem tempo de vida bastante longo, e tem que manter seu estado persistente durante todo este tempo. Como exemplo temos uma tabela de configuração sobre bancos de imagens remotos. Outros objetos poderão um tempo de vida intermediário, como imagens médicas armazenadas temporariamente em algum mecanismo de *cache* e serão destruídos após um tempo determinado. Pequenos objetos podem ter um tempo de vida curto e transitório, como um botão em uma caixa de diálogo que aparece apenas como confirmação de alguma ação.

Todo objeto CORBA em um sistema, sem importar seu tempo de vida, tem sua própria referência de objeto. Ela é atribuída por seu ORB na criação do objeto e permanece válida até o momento em que o objeto é explicitamente apagado. Clientes obtêm referências para objetos de diversa maneiras, por exemplo através de repositórios locais ou remotos e arquivos. Com a referencia remota para o objeto é realizada a associação com a de acordo com o mapeamento da linguagem utilizada. Esta associação permite que o ORB dirija a invocação para o específico objeto alvo.

A OMG define certos pré-requisitos para validar uma referência de objeto. Por exemplo, o cliente pode armazenar a referência de um particular objeto em um arquivo ou banco de dados. Quando o cliente recupera a referência de um objeto posteriormente ao armazenamento, a especificação requer que sua invocação execute com sucesso mesmo que o objeto tenha sido movido para outro lugar durante o intervalo em que a referência esteve armazenada (esta requisição não se aplica caso o objeto tenha sido apagado).

A OMG permite que qualquer ORB decodifique uma referência de objeto. Referências de objetos possuem um papel chave no processo de fornecer um recurso para um usuário no âmbito de CORBA. Referências para objetos podem passados entre aplicações de qualquer lugar do mundo utilizando para isso banco de dados, serviços de nomes, comércio, localização pública de arquivos ou qualquer outro meio. Qualquer aplicação utilizando qualquer ORB em sua rede local poderá recuperar referências de objetos e repassá-las para o ORB para invocar o objeto.

Para um programador não há qualquer necessidade de entender a forma e como uma referência para objeto funciona, tendo em vista que de acordo com a especificação da OMG apenas um ORB poderá interpretar referências de objeto. Um programador

deve apenas obter a referência de um objeto e passá-la para o ORB, e o ORB levará sua invocação ao objeto alvo.

4.3.7.1 IOR (*Interoperable Object Reference*)

A referência de objeto interoperável é formato padrão adotado pelo OMG para referência de objetos entre diferentes implementações de ORBs. Este formato é utilizado para comunicação entre ORBs e não é o mesmo utilizado entre comunicações entre objetos gerenciadas por um único ORB. O IOR é o meio universal para se identificar um objeto. Ele é opaco para a aplicação cliente e encapsula completamente todas as informações necessárias para o envio da requisição. Entre as informações contidas em um IOR estão detalhes a respeito do ambiente do objeto servidor, da arquitetura de hardware, componentes de rede e sobre os próprios componentes CORBA presentes na entidade par da comunicação. Mais especificamente, as informações abordam as seguintes questões:

- Qual o tipo do objeto? ORBs precisam saber tipo do objeto para preservar a integridade do sistema.
- Quais protocolos o ORB que foi invocado poderá usar? Em uma comunicação direta ORB-para-ORB é necessário que a IOR liste os protocolos aceitos pelo ORB que realizou a invocação.
- Quais serviços do ORB estão disponíveis. A invocação poderá envolver serviços ORBs estendidos; as especificações da OMG já permitem estas invocações entre transações, invocações seguras, e outros casos. Colocando-se esta informação no IOR, é possível eliminar a negociação ORB-para-ORB do contexto da informação.

A figura 7 mostra uma referência para um serviço de nomes remoto ligado ao ORB *ORBacus* inserido em um ambiente Java. A referência obtida no ambiente *VisualWorks* DST.

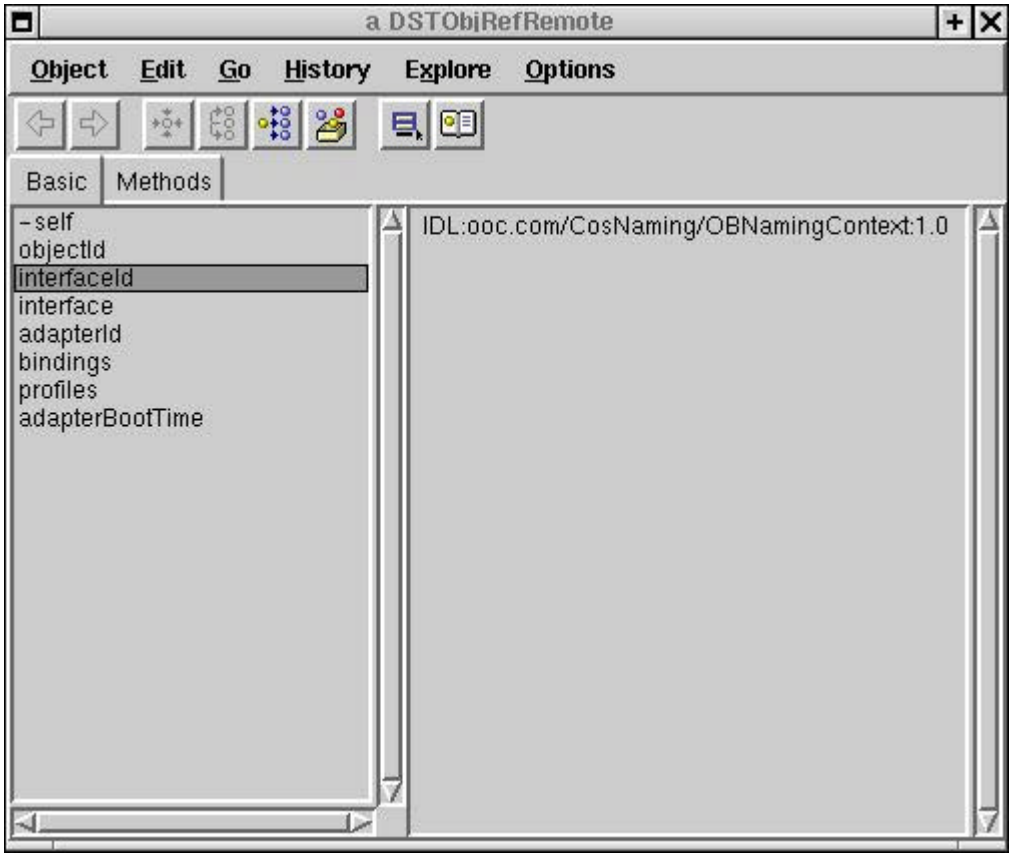

Figura 8: Referência para um objeto remoto no ambiente *VisualWorks*/DST.

Adicionalmente ao formato IOR, um ORB pode fornecer codificações de referências proprietárias. Esta capacidade pode ser útil se um ORB é projetado para um determinado ambiente, como um banco de dados orientados a objetos. Entretanto, referências proprietárias não podem ser utilizadas com ORBs de diferentes fornecedores.

4.3.8 Protocolos de comunicação entre ORBs

A especificação CORBA é neutra com relação com relação com respeito à protocolos de rede; para atingir esta neutralidade foi criado o *General Inter-ORB Protocol* (GIOP), o qual especifica, em alto nível, um padrão para comunicação entre diverso ORBs. GIOP como o nome sugere é apenas um protocolo geral; o padrão CORBA especifica protocolos adicionais que especializam GIOP para ser usado em um particular protocolo de transporte. Desta forma, o GIOP jamais será usado diretamente.

A especificação GIOP aborda os seguintes aspectos:

- Detalhes da camada de transporte: GIOP formaliza uma série de requisições sobre a camada de transporte que carrega as implementações do protocolo GIOP. Entre elas, tem-se que a camada de transporte deve ser orientada a conexão, *full-duplex*, simétrica e confiável.
- Estruturação do *Common Data* Representaion (CDR): determina o formato binário dos tipos IDL durante a transmissão. Entre as características do CDR estão o suporte à ordenação de *bytes big-endian* e *little-endian* através da utilização de rótulos, permitindo que máquinas enviem os dados em seu formato nativo e o alinhamento dos tipos de dados entre as palavras de bytes (por exemplo, um valor *double* é alinhado para oito *bytes*).
- Formatos de mensagem: GIOP define oito tipos de mensagens que são usadas por clientes e servidores para ser comunicarem. Apenas duas dessas mensagens são necessárias para a semântica básica de invocação remota de procedimentos do CORBA. O restante são mensagens de controle ou mensagens que provêem suporte a certas otimizações.

A especialização mais utilizada de GIOP é o Internet Inter-ORB *Protocol* (IIOP). Ele que é direcionado para a pilha de protocolos TCP/IP, o qual faz parte da requisição de compatibilidade determinada pela OMG para todas implementações de ORBs. Outro protocolo suportado é o DCE, *Distributed Computing Enviroment*, designado para ambientes distribuídos criado pelo *Open Share Foundation*.

4.3.9 Repositório de implementações

O processo em que um cliente estabelece uma conexão com o servidor correto e como este servidor associa as requisições que chegam com o *servant* é chamada de *binding*. ORBs tipicamente suportam dois tipos distintos de *binding*: direto e indireto. Sempre que uma aplicação servidora cria uma referência para objeto, é embutida nesta referência informação para possibilitar o *binding*. Especificamente, um IOR contém um endereço IP ou registro de nome DNS, número de porta TCP, e uma chave de objeto. Se o servidor insere seu próprio endereço IP e porta TCP, a referência usa *binding* direto. Se o servidor insere o dados de conexão de outro componente externo CORBA, a referência usa *binding* indireto. Este componente externo é chamado de repositório de implementações.

O repositório de implementações mantém um registro de servidores conhecidos, quais deles estão ativos, em qual computador e usando qual porta TCP. Ele pode prover funcionalidades adicionais como facilidades para migração de servidores e objetos, balanceamento de carga e *bootstrap* automático do servidor. Para esta última, o repositório de implementações deve possuir informações sobre os comandos e parâmetros corretos para iniciar o servidor sob demanda, caso ele não esteja em execução durante a chegada da requisição.

4.3.10 Desenvolvimento de um sistema distribuído CORBA

Esta seção visa colocar todos os componentes CORBA descritos anteriormente dentro do contexto de uma operação, enfatizando a atuação de cada componente. O desenvolvimento de aplicações CORBA envolve um conjunto de etapas onde, em cada uma delas, diferentes componentes são acionados. Com exceção do compilador IDL e do ORB, que são normalmente fornecidos em único produto, todos os demais componentes são opcionais em um sistema CORBA. Esta flexibilidade permite configurações e ao mesmo tempo provê um mínimo de interoperabilidade. A seguir serão apresentados os passos para a criação e execução de uma aplicação CORBA, nos lados do cliente e do servidor.

4.3.10.1 Procedimentos no sistema cliente

Para que uma aplicação cliente possa utilizar os serviços de um objeto CORBA é necessária a definição da interface do método a ser invocado e a referência para o objeto que serve este método. A definição da interface é gerada através da compilação da descrição em IDL para a linguagem de programação do cliente. O cliente possui duas maneiras de utilizar esta definição de interface: estaticamente ou dinamicamente. Na primeira, como resultado da compilação da descrição IDL é gerado um *stub*, que é posteriormente ligado estaticamente a aplicação cliente. Durante a invocação do procedimento remoto, o cliente utilizará a interface para invocações estáticas do ORB através deste *stub*. Para invocações dinâmicas o cliente obtém a definição de interface através de um repositório de interfaces e utiliza a API do ORB para invocação de procedimentos dinâmicos, o DII.

Para que o ORB encontrar o objeto servidor do método invocado é necessária a referência remota para o objeto servidor, o IOR. Esta referência pode ser obtida através

de um serviço CORBA externo como o serviço de nomes ou transação ou através de um arquivo texto, se a referência estiver no formato *stringified*. Com esta referência, o ORB cliente pode então enviar a requisição do procedimento ao servidor remoto.

Normalmente, o ORB é uma biblioteca que é ligada ao código da aplicação, agindo como *proxy* para as invocações de métodos remotos. Sob ponto de vista lógico, o ORB é um único componente, mas possui algumas funções específicas para o cliente e outras específicas para o servidor. O ORB do lado do cliente gerencia as invocações do cliente e seleciona os servidores e métodos relacionados. Ele realiza a validação e *marshaling* dos argumentos usando a definição de interface IDL. *Marshaling* significa converter as estruturas e tipos de dados que correspondem aos argumentos do método remoto para o formato comum da especificação CORBA, o CDR. Posteriormente, ao receber os dados de retorno do método invocado, o ORB os converte do formato CDR para o formato nativo da aplicação cliente. O ORB cliente utiliza os formatos de mensagens requisição especificados no GIOP. O protocolo específico utilizado é a especialização do GIOP para a camada de transporte e rede utilizado, como o IIOP para o TCP/IP.

4.3.10.2 Execução no sistema servidor

De maneira análoga ao cliente, para que uma aplicação servidora disponibilize seus métodos para atender a invocações remotas é necessário a definição da interface em IDL e a criação da referência para o objeto que implementa estes métodos. A compilação da descrição IDL gera um *skeleton* no lado do servidor, para ser ligado estaticamente na aplicação. Assim como no cliente, o servidor pode opcionalmente obter dinamicamente a definição da interface através de um repositório de interfaces. Neste caso, a API do ORB utilizada pela aplicação servidora é o DSI.

O adaptador de objeto é utilizado para criar o objeto servidor CORBA e referência para o mesmo. O POA é a atual especificação CORBA para adaptadores de objetos. Para publicação desta referência a aplicação pode utilizar um serviço de nomes ou gerar a referência em formato *stringified*. Esta última é representada em formato ASCII e pode ser enviado ao cliente por qualquer método, um e-mail por exemplo. A referência de objetos pode referenciar diretamente o ORB ligado à aplicação servidora como a um repositório de implementações. Neste repositório de informações estão publicadas todas

informações para que o cliente encontre e acione o servidor, como parâmetros de rede e iniciação.

O ORB do lado do servidor, que é ligado à aplicação servidora, recebe a requisição para despacho de um método. Esta requisição chega encapsulada em uma mensagem GIOP. O ORB servidor utilizará as mensagens de resposta e controle GIOP para realizar a interação com o cliente. Em seguida será realizado o *unmarshalling* dos dados em formato CDR para os tipos e estruturas de dados nativos da aplicação servidora, usando as definições contidas no *skeleton* correspondente.

Após a validação e decodificação dos dados, o ORB invoca o POA para ativar o objeto CORBA, se necessário. O POA fará a encarnação do objeto CORBA em um *servant*, a real implementação do objeto que irá realizar a operação. Finalmente, é feito o *marshalling* do resultado da operação do *servant* e enviado para o cliente da requisição.

4.4 Serviços de Objetos

Serviços de objetos são facilidades da arquitetura OMA independentes de domínio de aplicação. São chamados de facilidades horizontais, sendo importantes para o desenvolvimento de aplicações distribuídas por prover uma base de interoperabilidade comum. Desta forma, desenvolvedores dispõem de um conjunto de funcionalidades comuns à aplicação distribuição sem a necessidade de implementar sua própria solução.

Diversos serviços de objetos foram especificados pela OMG em [OS01]. A seguir serão descritos brevemente os principais:

- Serviço de nomes: Este serviço provê a funcionalidade de associar um nome a uma referência para objeto. A mesma referência de um objeto pode ser associada com diferentes nomes. Estes associamentos (normalmente referenciado como *name binding*) são mantidos em tabelas dentro de um objeto chamado contexto de nomes. Uma entrada em uma tabela de um contexto de nomes pode referenciar a outro contexto de nomes. Desta maneira é possível a construção de árvores hierárquicas dentro do serviço de nomes, similar às estruturas de diretórios de um sistema de arquivos. A funcionalidade oferecida pelo serviço de nomes é bastante similar ao DNS (*Domain Name Service*). Enquanto que o DNS mapeia nomes de domínio para endereços IP, o serviço de nomes CORBA mapeia nomes para referências para objetos.

- Serviço de eventos: Este provê suporte para interações assíncronas entre objetos. O modelo deste serviço é baseado no conceito de fornecedores, objetos que produzem eventos, e consumidores, os objetos que recebem estes eventos. Um canal de eventos transporta eventos dos fornecedores para os consumidores. Ele é o elemento central em um serviço de eventos, sendo responsável pelo registro de fornecedores e consumidores, entrega dos eventos para todos consumidores e controle de tempo e erros associados com o serviço. Fornecedores podem gerar eventos sem conhecer a identidade dos consumidores, e vice-versa.
- Serviço de ciclo de vida: Este serviço lida com a criação, destruição e relocação de objetos. Ele define operações para copiar, mover e remover grafos de objetos relacionados.
- Serviço de persistência de objetos (p*ersistent object service*, POS): Este serviço permite que objetos "persistam" além do tempo que a aplicação que os criou ou clientes que utilizem seus serviços estejam ativos. POS salva o estado de um objeto em um meio de armazenamento e recupera este estado quando necessário.
- Serviço de transação de objetos (*object transaction service*, OTS): Este serviço define interfaces que permite que múltiplos objetos distribuídos cooperem entre si para garantir a atomicidade de uma operação. Estas interfaces capacitam objetos a executar o *commit* ou *rollback* de todas mudanças efetuadas de maneira similar à utilizada em bancos de dados.
- Serviço de tempo: Este serviço permite que usuários obtenham o horário atual juntamente com uma estimativa de erro associada com ele. Ele três tipos de objetos: os *universal time objects* (UTO), os *time interval objects* (TIO) e objetos para controle de eventos de tempo. Manter a sincronização entre horários é fundamental para a ordenação dos diversos eventos que ocorrem em um sistema distribuído.
- Serviço de comércio (*trader service*): Este serviço permite que usuários descubram objetos baseados nos serviços que eles oferecem, de maneira similar às páginas amarelas de um catálogo telefônico. Exportadores divulgam seus serviços para o este objeto e importadores usam este objeto para descobrir os serviços que procuram.
- Serviço de segurança: Este serviço provê as seguintes funcionalidades: identificação e autenticação, autorização e controle de acesso, auditoria, segurança de comunicação, não-repudiação e administração.

4.5 Domínios de interfaces OMA

Os domínios OMA são facilidades verticais direcionadas a um segmento de mercado específico. Cada domínio CORBA fornece interfaces padronizadas que abordam questões pertinentes de um modelo de negócios em particular.

Atualmente, existem nove forças tarefas atuando em domínios específicos de mercado. Existem forças tarefas especializas em domínios de negócio, finanças, comércio eletrônico, manufaturamento, medicina, telecomunicações, transportes, e utilidades são alguns exemplos.

4.5.1 *HealhCare* DTF (*HealthCare Domain Task Force*)

O domínio vertical pertinente a este trabalho é o *HealhCare* DTF. Inicialmente chamado CORBAMed, o HealhCare DTF compreende diversas especificações relacionadas a indústria médica e representa vendedores, fornecedores de planos de saúde e usuários finais. Ele define interfaces padronizadas entre serviços médicos relacionados. Estas interfaces fornecem uma interoperabilidade adicional para que aplicações que utilizem o padrão CORBA de maneira transparente.

A primeira especificação produzida pelo *HealhCare* DTF foi direcionada para a padronização de interfaces e integração de sistemas identificação única de pacientes. A seguir será apresentada uma visão geral sobre a mesma.

4.5.1.1 PIDS (*Person Identification Service*)

A especificação PIDS define um modelo de referência baseado em interfaces CORBA para o gerenciamento de identificadores de pacientes. Especificamente o PIDS visa prover as seguintes funcionalidades:

- Suporte de IDs em um particular domínio e a correlação de IDs entre múltiplos domínios.
- Suporte à busca e pareamento de informações de pacientes de maneira interativa e também automatizada, independente dos mecanismos de relacionamento utilizados.
- Suporte para criação de federações de PIDS em uma topologia independente.
- Prover suporte para que implementações de PIDS apliquem políticas de confidencialidade e mecanismo de segurança.

- Definir níveis de conformidade abrangendo diferentes graus de sofisticações, de sistemas de identificação somente para consulta a federações de sistema de identificação.

A figura 8 descreve os elementos estruturais básicos do modelo de identificação do PIDS.

O Domínio de ID é o bloco básico do modelo PIDS. Um Domínio ID mantém um identificador único (ID) para identidade de pessoas representadas no Domínio ID. Idealmente, existe um e apenas um ID por pessoa, mas na realidade podem existir IDs duplicados onde uma pessoa pode mais de um ID em um mesmo domínio. Para consistência interna, o Domínio ID não pode atribuir duas pessoas para o mesmo ID, caso contrário o sistema não poderá distinguir entre duas entradas de duas pessoas diferentes. O ID é um mecanismo de controle interno e poderá ou não ser utilizado externamente. Desta forma, o ID e seu domínio, juntos, constituem um ID único para uma pessoa.

Na especificação dos PIDS, várias interfaces são detalhadas. As duas mais diretamente utilizadas são *IdentifyPerson* e *ProfileAcess*. A interface *IdentifyPerson* é

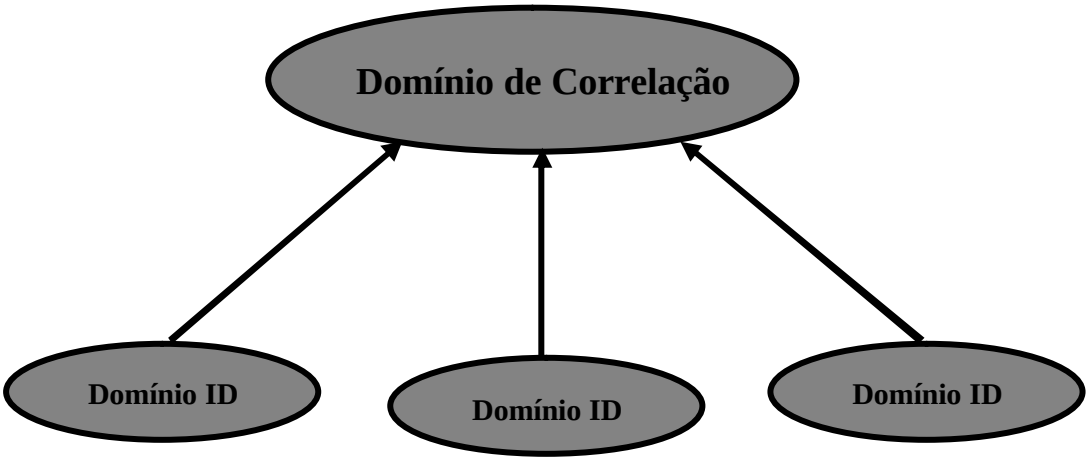

Figura 9: Elementos estruturais básicos do modelo de identificação PIDS

provê basicamente um meio de consulta onde se envia características para serem correlacionadas no Domínio ID e recebe-se uma lista de candidatos como resultado. A interface *ProfileAccess* pode ter a funcionalidade de uma consulta ou atualização usada

para enviar um ID para uma pessoa específica e para receber o perfil desta pessoa. Um perfil é um conjunto de características contendo valores da pessoa para suas respectivas características.

A unidade estrutural coordenadora do modelo PIDS é o Domínio de Correlação. O Domínio de Correlação permite acesso aos perfis correlacionados dos IDs em todos os Domínios ID participantes. O Domínio de Correlação será portanto um bloco de ligação que complementará os existentes Domínios ID, servindo como meio para que um Domínio de ID aumente seu escopo, interagindo com outros Domínio de ID. O Domínio de Correlação define duas interfaces: *load_profiles* e *get_corresponding_id*. A primeira é utilizada para adicionar e correlacionar perfis de um paciente. A segunda é utiliza para obter um conjunto de IDs relacionados a um mesmo paciente.

O Domínio de Correlação pode correlacionar tanto um Domínio ID quanto outros Domínios de Correlação, o que permite a construção de estruturas hierárquicas. Como um Domínio ID pode participar de mais de um Domínio de Correlação é também possível a construção de estruturas pares.

4.5.1.2 Aplicação do PIDS para integração de servidores de imagens

O modelo PIDS provê um *framework* para a manutenção de identificadores únicos de pacientes entre diversas organizações. Entretanto, é importante ressaltar que, o modelo consiste basicamente em um conjunto de interfaces descritas em IDL, que definem *wrappers* para sistemas de identificação já existentes. Ou seja, não é abordada a resolução do problema em si, que é correlacionar identificadores de diferentes organizações, um problema de relacionamento de registros.

O modelo proposto para o Domínio ID foge do escopo deste trabalho. Como detalhado no capítulo sobre relacionamento de registros, para alimentar o sistema de identificação do Portal de Teleradiologia é que desejável que os participantes possuam os sistemas HIS e RIS integrados. O desenvolvimento de um gerenciador de identificadores para atender as interfaces definidas no Domínio ID representa uma tarefa além da simples integração HIS/RIS.

Por outro lado, os requisitos para a implementação das interfaces do Domínio de Correlação vão de encontro ao modelo proposto neste trabalho. As interfaces *load_profiles* e *get_corresponding_id* podem ser mapeadas facilmente para funções já requeridas no sistema MPI do Portal de Teleradiologia. Desta forma, tem-se que a

conformidade com o Domínio de Correlação requer pouca modificação no modelo proposto neste trabalho. Além disso, a especificação PIDS é o modelo referência para o sistema de identificação do cartão SUS. Desta forma, a implantação das interfaces para o Domínio de Correlação garantirão a interoperabilidade futura com este sistema.

5. RELACIONAMENTO DE REGISTROS

É razoável cogitar que o interesse em relacionar conjuntos de registros surgiu praticamente junto com a própria prática de registrar informações. Combinar registros de fontes de dados distintas é uma prática comum para aumentar, comparar, integrar e relacionar informações de um determinado objeto do mundo real.

O termo relacionamento de registros (do inglês *record linkage*) tem sido usado para identificar a metodologia de buscar registros correspondentes em dois ou mais arquivos ou encontrar registros duplicados dentro de um mesmo arquivo utilizando recursos computacionais. Esta definição tem origem na área de saúde pública onde arquivos de pacientes individuais eram relacionados usando informações como nome, data de nascimento e outras informações. O processo de relacionar registros possui ainda diferentes nomes em diferentes comunidades de usuários e de pesquisa [CC02]. Enquanto que termo em inglês r*ecord linkage* é normalmente utilizado por epidemiologistas e estatísticos, o mesmo, ou bastante similar, processo pode ser denominado *entity heterogeneity, entity identification, object isomerism, instance identification, merge/pung, entity reconciliation, list washing* e *data cleaning* por cientistas da computação, entre outros. Neste trabalho utilizaremos o termo relacionamento de registros, tradução de r*ecord linkage,* adotado em trabalhos nacionais anteriores como [CC00].

A área médica foi pioneira em utilizar recursos computacionais para relacionamento de registros [NK59] e é, ainda hoje, a maior geradora de estudos nesta área. Existem várias motivações para o relacionamento de registros na medicina. A mais citada talvez seja a criação e manutenção de MPIs (acrônimo do inglês *Master Patient Index*), que permite manter uma abrangente documentação do histórico de vida um indivíduo ligando seus registros clínicos de diferentes bases de dados. Estudos epidemiológicos e sobre saúde são outra área de aplicação das técnicas de relacionamento de registros. Neste tipo de pesquisa existe a necessidade de relacionar registros de nascimento com registros de óbito ou de incidência de doenças para inferir fatores de risco ou causas de determinados fatos clínicos. Por exemplo, tem-se o estudo conduzido na Escócia, onde registros de maternidade foram relacionados com registros de nascimento para verificar se problemas de hipertensão durante a maternidade altera a associação entre o retardo do crescimento intra-uterino e mortalidade de recém

nascidos. Trabalhos recentes têm aplicado as técnicas de relacionamento de registros em áreas como recuperação de informações em mecanismos de busca e mineração de dados.

Dentro do escopo deste trabalho, as técnicas relacionamento de registros têm importância fundamental para a integração de servidores de imagens DICOM. A capacidade de identificar um mesmo paciente em diferentes bases de imagens é uma funcionalidade indispensável. Sem ela, o sistema não proverá uma integração real dos bancos de imagens, limitando-se a prover uma interface para acesso remoto, e possivelmente simultâneo, aos mesmos. Para a construção um sistema federado de servidores DICOM é necessário realizar a integração de seus dados e para isso as técnicas de relacionamento de registros são fundamentais.

A estrutura deste capítulo é a que segue. Na seção 5.1 são apresentados os principais problemas para o relacionamento de registros dentro do escopo deste trabalho. Na seção 5.2 são apresentados os conceitos básicos fundamentais sobre relacionamento de registros, a formalização do problema e as duas principais estratégias empregadas: a abordagem determinística e a abordagem probabilística. Na seção 5.3 são apresentados os principais componentes que compõem um sistema para relacionamento de registros. Finalmente na Seção 5.4 é discutida aplicação das técnicas de relacionamento de registros dentro do escopo do presente trabalho.

5.1 Descrição do Problema

A grande dificuldade em identificar precisamente registros produzidos por diferentes sistemas como relacionados entre si é um fator extremamente limitante para a criação de bancos de dados distribuídos de informações médicas de qualquer natureza. A heterogeneidade entre os sistemas informação de organizações de saúde e a conseqüente incompatibilidade em seus métodos de identificação de pacientes impossibilitam a correlação direta entre registros originados de locais distintos. A única alternativa é a comparação direta de características pessoais (por exemplo nome, idade, sexo, estado civil, data e local de nascimento), que em conjunto podem ser utilizadas para identificar satisfatoriamente um indivíduo. A similaridade de entradas em campos de atributos é portanto, a base para a decisão de correlacionar ou incorporar entre si dois registros.

O auxilio de computador torna possível a correlação de grandes bancos de dados de clínicas e hospitais para criação de índices únicos para cada paciente e associá-los a *links* que conduzem para os locais de origem da informação. Estes sistemas recebem normalmente o nome de MPI (acrônimo do inglês Master Patient Index). Nestes sistemas, o relacionamento de registros deverá ser realizado em dois momentos. O primeiro é quando um novo registro é adicionado ao sistema. Um algoritmo de correlação efetivo deve ser capaz realizar comparações com milhares de registros, preferencialmente em paralelo. O segundo momento é quando consultas são feitas ao sistema fornecendo um conjunto de características de um paciente para busca de seus dados.

Registros médicos de pacientes possuem muitas características que podem conduzir a erros durante a correlação de registros. A heterogeneidade, novamente, é fator que adiciona complexidade à correlação de registros, pois atributos com mesmo valor semântico podem ser representados de maneira diferente em dois sistemas. Por exemplo nomes e sobrenomes podem estar representados inversamente entre dois registros. Apelidos, abreviações ou informação codificada (por exemplo 1 para masculino e 2 para feminino) e outras políticas específicas para representação de informações dificultam bastante à busca por similaridade entre registros. Apesar de alvo de organizações internacionais que promovem a interoperabilidade entre sistemas hospitalares como o HL-7 (acrônimo do inglês *Health Level Seven*), ainda é incipiente no contexto mundial, e praticamente inexistente no Brasil, a adoção de métodos padronizados para representação de informações pessoais. Além disso, somado à necessidade de utilização de sistemas legados tem-se a certeza de que estes problemas de heterogeneidade irão perdurar por muitos anos.

Outro problema é a variedade de erros que podem ocorrer durante a inclusão de dados em sistema, por exemplo, erros fonéticos e erros de digitação (adição ou remoção de espaços, caracteres inválidos). Alguns campos de atributos podem ser utilizados de maneiras errada, possuindo dados de semântica diferente do contexto ou que deveriam estar em outro campo. Podem ocorrer também mudanças válidas nos dados de uma paciente, como mudança de sobrenome após casamento.

Um algoritmo efetivo para correlação de registros deve ser capaz tomar decisões quanto à correlação ou não de dois registros que possuem um determinado grau de

similaridade. A determinação do grau de similaridade necessário para que dois registros sejam relacionados é de difícil formalização e baseia-se principalmente em análise probabilística. O conhecimento prévio ou adquirido de características sobre o contexto de origem dados (por exemplo, métodos comuns de abreviação de campos e características específicas relacionadas à origem étnica) é de grande valia para o refinamento de regras de correlação. A qualidade do conjunto de regras de correlação determinará o sucesso de um sistema MPI. É bastante indesejável que um sistema MPI identifique informações de dois pacientes como pertencentes a um só, o que conduz a regras restritas para a decisão a favor da correlação. Por outro, uma restrição severa conduz a não-identificação de registros relacionados a um mesmo paciente ou a uma constante necessidade de interação humana , o que reduz a utilidade de sistema.

O principal desafio da correlação de registros por auxílio de computador é reproduzir o mais próximo possível a capacidade de análise e inferência do ser humano para lidar com problemas como os citados acima. As diversas questões que envolvem a correlação de registros dificultam a criação de uma abstração matemática que compreenda todo o domínio do problema e que possibilite a elaboração de métodos exatos para abordagem ao mesmo. Desta forma, a criação e emprego de técnicas e métodos formais adequados para a implementação de sistemas MPI geograficamente distribuídos ainda é fonte de ampla pesquisa mundial.

5.2 Conceitos principais sobre relacionamento de registros

O relacionamento de registros pode ser apresentado de maneira mais formal da seguinte maneira:

Supondo um arquivo A com n_a registros e um arquivo B com n_b registros, então o arquivo A×B contém $n_a \times n_b$ pares de registros. Cada um dos n_b registros no arquivo B é um par em potencial para cada um dos n_a registros do arquivo A. Desta forma, existem $n_a \times n_b$ pares de registros em potencial cujo *status* par ou não-par está para ser definido. Dentre estes, pelo menos $\min(n_a, n_b)$ podem ser pares. Quando $n_a = n_b = n$, por exemplo, existem no máximo n pares (podem existir registros iguais nos arquivos), e pelo menos $n(n-1)$ pares falsos, e a busca será feita por $O(n)$ pares entre $O(n^2)$ pares

falsos. O processo de relacionamento utiliza a informação em comum nos arquivos A e B para classificar cada par de registro em $n_a \times n_b$ como um par ou não-par.

O relacionamento registros em diferentes bases de dados é relativamente trivial nos casos em que os registros de cada base incluam campo comum que permita a identificação de cada registro de forma inequívoca, como, por exemplo, CPF. Uma simples operação de *join* em SQL ou equivalente em outros sistemas de gerenciamento de dados seria suficiente. Entretanto, na ausência de uma chave única e compartilhada por todos conjuntos de dados várias técnicas de relacionamento de registros são necessárias.

Os métodos utilizados para lidar com os problemas de relacionamento de registros podem ser categorizados em dois grupos com relação a abordem utilizada: determinísticos e probabilísticos. A seguir será detalhada cada uma destas abordagens.

5.2.1 Abordagem Determinística

As estratégias determinísticas para relacionamento de registros, apesar de ainda ser, pelo menos em parte, utilizada por diversos softwares para relacionamento de registros, dispõem de pouca literatura disponível. Na estratégia mais simples e restritiva do relacionamento de registros "determinístico", pares são determinados por comparações do tipo "tudo ou nada". Neste tipo de pareamento, quando comparando dois registros através dos campos nome e sobrenome, os registros são considerados pares verdadeiros apenas se o texto nos dois campos são iguais em todos os caracteres. Regras mais flexíveis podem ser usadas que permitam que sub-conjuntos pré-definidos de identificadores determinem um relacionamento.

Enquanto que, estratégias determinísticas possuem a desvantagem de que não tratarem valores ausentes ou combinações parciais, elas tem a vantagem de que o especialista pode informalmente usar seu conhecimento sobre as fontes de dados para decidir sobre a estratégia de relacionamento. Por exemplo, é possível variar taxas de erro através de uma metódica seleção de identificadores para pareamento. Limitando apropriadamente o número de passos usados no processo de pareamento, o especialista consegue algum controle sobre as taxas de pares falsos e verdadeiros. Entretanto, estas taxas de erros são amarradas à confiabilidade dos identificadores disponíveis e controle preciso não é possível.

Outra argumentação a favor da abordagem determinística é a facilidade de implementação de seus métodos com a relação à abordagem probabilística. O *framework* matemático denso utilizado pelas técnicas probabilísticas dificulta sua utilização por especialistas não relacionados com a área da matemática e estatística, como médicos por exemplo. A estratégia determinística também pode ser opção razoável para situações onde exista apenas uma necessidade eventual de relacionar registros.

Métodos como a medida de entropia de Shannon foi sugerida por [RW91] para estimar a quantidade informação para relacionamento disponível em cada arquivo de dados. [RW91] também compara as estratégias determinísticas mais simples com as estratégias probabilísticas em relacionamentos de dados entre as organizações *Canadian Vital Statistics* e *Manitoba Health Services Comission*, e faz sugestões sobre quando seu uso é apropriado. [GW03] apresenta uma comparação empírica entre métodos determinísticos e probabilísticos e [GW04] apresenta uma estratégia determinística utilizando o método de codificação fonética NWSIIS (*New York State Intelligence Information System*). Entretanto de maneira geral, pareamentos determinísticos são realizados de maneira *ad hoc* e não são publicados.

Existe um certo consenso na literatura de que a abordagem determinística não é adequada em situações onde existem diversos campos com valores imprecisos [GR03], [MH03], [RW91]. Entretanto, como notado em [GC04] não existe nenhuma evidência factual ou pesquisa direcionada que comprove esta característica na literatura. É importante que mais estudos como o apresentado em [GC04], que ainda está inconcluso, que comparem de maneira apropriada características e resultados das estratégias determinísticas e probabilísticas em situações onde a verdade sobre pares verdadeiros e falsos de registros seja conhecida.

5.2.2 Abordagem probabilística

O trabalho considerado pioneiro sobre relacionamento de registros com abordagem probabilística foi realizado em Howard Newcombe em 1959 [NK59]. Este trabalho foi motivado pela produção de dados estatísticos para pesquisa genética e biomédica. Mais especificamente o trabalho buscava desenvolver um *framework* para criação e manutenção de um histórico de grupos de pessoas que haviam sido expostas ao contato com baixos níveis de radiação para determinar alguma relação com a causa

de uma eventual morte. Newcombe reconheceu o relacionamento de registros como um problema estatístico, caracterizado pela presença de erros de identificação da informação e a decisão de qual par de registros deverão ser correlacionados. Suas descobertas foram cruciais e conduziram às abordagens computadorizadas seguintes para tratamento do relacionamento de registros.

Newcombe percebeu que a freqüência relativa da ocorrência do valor de uma *string* como sobrenome entre pares verdadeiros e falsos poderia ser usada no cálculo de um peso associado com o pareamento de dois registros. Este peso foi denominado *binit* e algo como um escore para cada campo de comparação, Outra constatação foi de que escores sobre diferentes campos como nome, sobrenome e idade, poderiam ser somados para obter-se uma pontuação de pareamento geral. De modo mais específico, considerando as seguintes *odds ratio*

$$\log_2(P_L) - \log_2(P_F) \qquad (1)$$

onde P_L é a freqüência entre pares verdadeiros e P_F é a freqüência relativa entre pares falsos. Desde que a real situação de pareamento não é conhecida freqüentemente conhecida, é sugerida uma aproximação em (1) com a seguinte relação

$$\log_2(P_R) - \log_2(P_R)^2 \qquad (2)$$

onde P_R é a freqüência de uma *string* (nome, sobrenome, lugar de nascimento,etc). Se casarmos um arquivo representando uma grande quantidade de dados com ele mesmo, então a segunda relação é uma boa aproximação da primeira relação. Estas idéias de Newcombe nortearam praticamente todas as pesquisas posteriores em relacionamento probabilístico de registros.

Em 1969, Fellegi e Sunter formalizaram seu reconhecimento intuitivo e rigorosamente descreveram o espaço de pares de registros consistindo de todas comparações possíveis [FS69]. Neste trabalho foi introduzida a fundação matemática formal para o relacionamento de registros probabilístico. Seguindo [GR03], a descrição do trabalho de Fellegi e Sunter é a que segue:

Pares de registros são modelados como:

- par, A1.
- possível par, A2.
- não-par, A3.

Para um registro a fonte de dados A e um registro b da fonte de dados B, a informação disponível nos registros é denotada por a (a) e b (b) respectivamente. Um vetor de comparação ou concordância, g, para um par de registros (a (a), b (b)) representa o nível de concordância entre os registros a e b.

Quando pares de registros são comparados baseado-se em k campos de identificação, o vetor g possui k componentes. g = (g^1(a (a), b (b)),..., g^k(a (a), b (b))) é uma função sobre o conjunto de todos pares de registro $n_a \times n_b$.

Para um vetor de concordância g em Γ, o espaço de todos possíveis vetores de comparação, $m(g)$ é definido como uma probabilidade condicional de que, dado g, dois registros formam um par verdadeiro. Ou seja:

$$m(g) = P(g \mid (a,b) \in M) \quad (1)$$

Similarmente,

$$u(g) = P(g \mid (a,b) \in U) \quad (2)$$

denota a probabilidade condicional de, dado g, dois registros formam não-par verdadeiro.

Existem dois tipos de possíveis erros de classificação: falsos pares (erros Tipo I, quando dois registros classificados como pares, na verdade são não-pares) e falsos não-pares (erros Tipo II, quando dois registros classificados como não-pares, na verdade são pares). A probabilidade de um falso par é:

$$P(A_1|U) = \sum_{g \in \Gamma} u(g) P(A_1 \mid g) \quad (3)$$

e a probabilidade de um não-par falso é:

$$P(A_3|M) = \sum_{g \in \Gamma} m(g) P(A_3 \mid g) \quad (4)$$

Para valores fixos da média de falsos pares (m) e da média de falsos não-pares (l), Fellegi e Sunter define a regra para relacionamento ótimo sobre Γ em níveis m e l, denotados por $L(m, l, \Gamma)$ como a regra para o qual $P(A_1|U) = m$, $P(A_3|M) = l$, e $P(A_2| L) = P(A_2| L')$ para todas outras regras L'. A regra ótima é definida aqui como a regra que minimiza a probabilidade de classificar um par como pertencente a A_2, o subconjunto de duplas de registros que requerem revisão manual.

Deixe $\frac{m(g)}{u(g)}$ ser ordenada monotonamente decrescente (com valores iguais ordenados arbitrariamente) e o g associado ser indexado 1, 2, ..., $N\Gamma$. Se $m = \sum_{i=1}^{n} u(gi)$ e $l = \sum_{i=n'}^{N\Gamma} m(gi)$, $n <$ n', então a regra ótima é uma função da relação de vizinhança $\frac{m(g)}{u(g)}$ e é dada pelas seguintes equações:

$$(a,b) \in A_1 \text{ se } \prod_m = \frac{m(g)}{u(g)} \tag{5}$$

$$\in A_2 \text{ se } \prod_l < \frac{m(g)}{u(g)} < \prod_m \tag{6}$$

$$\in A_3 \text{ se } \frac{m(g)}{u(g)} = \prod_l \tag{7}$$

onde $\prod_m = \frac{m(g)}{u(g)}$ e $\prod_l = \frac{m(g_{n'})}{u(g_{n'})}$.

Com a presunção de independência condicional dos componentes do vetor g, a regra de decisão acima pode ser escrita como uma função de $\log\left(\frac{m(g)}{u(g)}\right) = \sum_{j=1}^{k} w_j$, onde o peso $w_j = \left(\frac{m(g^j)}{u(g^j)}\right)$.

Intuitivamente, existirá um número bem maior de duplas de registros classificadas como não pares que duplas classificadas como pares. O grau de separação entre os modos é uma indicação do nível de dificuldade da tarefa de relacionamento e a quantidade de erros Tipo I e Tipo II que poderão resultar.

5.3 Componentes de um sistema para relacionamento de registros

A abordagem prática a ser utilizada por um sistema de relacionamento de registros compreende diversas etapas e agrega diferentes técnicas de manipulação dos dados. Entre as etapas mais comuns durante fluxo dos dados em um sistema de relacionamento de registros estão:

- Padronização: Sua utilização depende da qualidade dos dados a serem comparados.

- Blocagem: A funcionalidade desta etapa é reduzir o número de comparações de duplas de registros.
- Comparação: Nesta etapa é feita a comparação entre as duplas de registros.
- Classificação: Nesta etapa é realizada a decisão se a dupla de registros é um par, não-par ou possível par, utilizando as técnicas descritas em 5.3, embora modelos de decisão alternativos possam ser utilizados
- Avaliação: Nesta etapa é realizada a avaliação do sistema de relacionamento de registros baseado em diferentes critérios de performance.

Existe ainda uma pré-etapa antes do processamento dos dados onde são selecionados os atributos para comparação. A seguir, os principais métodos empregados em cada uma destas etapas serão detalhados apropriadamente.

5.3.1 Seleção de atributos para comparação

Atributos comuns podem ser selecionados para uso em uma função de comparação. As questões envolvendo o processo de seleção de atributos incluem:

- Identificar os atributos em comum.
- Assegurar se os atributos em comum possuem informação suficiente para garantir a qualidade de relacionamento requerida pelo projeto.
- Selecionar o subconjunto ótimo de atributos em comum.

Características de atributos que afetam a decisão de seleção incluem nível de erros nos valores e o número (e distribuição) destes valores. Por exemplo, um campo como sexo possui apenas dois valores e conseqüentemente não adiciona informação suficiente para identificação de um par verdadeiro. Um campo como sobrenome adiciona bastante informação, mas é mais passível de ser registrado erroneamente.

Uma regra de decisão para seleção de atributos é selecionar todos os atributos em comum. A redundância provida por atributos relacionados pode ser útil para reduzir erros de pareamento. Entretanto, redundância de atributos é inútil se seus erros são bastante correlacionados ou dependentes funcionalmente.

Adicionalmente, o peso de identificadores pessoais com baixa qualidade pode ser melhorado considerando a semântica dos campos como causa de morte e co-morbidades conhecidas.

5.3.2 Métodos de Padronização

Os métodos de padronização são também chamados de *data cleaning* ou reconciliação em nível de atributos. Sem a padronização, vários pares verdadeiros poderão ser relacionados erroneamente como não-pares porque os atributos de identificação em comum não possuem similaridade suficiente.

As idéias básicas para padronização são:

- Substituir variações de grafia de palavras que ocorrem comumente por grafias padrões.
- Padronizar a representação de vários atributos para o mesmo sistema de unidades ou codificação. Por exemplo, 0/1 no lugar de M/F para o atributo sexo.
- Executar checagens de integridade em valores de atributos ou combinações de valores de atributos.

Métodos de padronização precisam ser específicos para as bases de dados a serem relacionadas e para o processo utilizado durante a extração de informações. Por exemplo, os erros mais comuns de grafias de nomes diferem bastante baseado na origem étnica do nome. Desta forma, o processo de padronização para nomes italianos são otimizados para tal e diferem do processo utilizado para nomes de origem latina.

5.3.3 Busca e blocagem

O método de busca e blocagem é usado para reduzir o número de comparações entre duplas de registros agrupando os registros com maior potencial de relacionamento entre si. Uma boa variável para blocagem deve conter um grande número de valores de atributos distribuídos uniformemente e cada atributo tendo uma baixa probabilidade de reportar erro.

Erros nos atributos usados para blocagem podem resultar em falhas para agrupar pares de registros relacionáveis. Para atributos de texto, diversos códigos fonéticos têm sido derivados para evitar efeitos de pronúncia e erros aurais ocorridos durante o registro dos nomes. Códigos fonéticos comuns incluem *Russel-Soundex* e NYSIIS. Estes códigos foram otimizados para populações específicas de nomes e tipos específicos de pronúncias de origem inglesa. Alguns sistemas comerciais provêem ferramentas para derivar códigos fonéticos para populações específicas.

Um bom algoritmo de blocagem deve apresentar uma boa negociação entre custo computacional e taxas de falsos não-pares, dois requisitos normalmente conflitantes. A seguir serão apresentadas algumas técnicas utilizadas nos esquemas de blocagem.

5.3.3.1 Método de vizinhança ordenada

O método de vizinhança ordenada (*sorted neighbourhood method* – SNM), envolve a varredura de N registros ordenados dos arquivos A e B usando uma janela de tamanho fixo, w. Todas duplas de registros contidos no intervalo desta janela são comparadas. SNM requer $w \times N$ comparações de registros. A relação de erro induzida pelo SNM é criticamente dependente da escolha das chaves de ordenação.

Múltiplos passos com chaves de ordenação independentes podem ser realizados para minimizar o número de erros. Um fechamento transitivo sobre pares de registros pareados pode ser computado para combinar os resultados de passos independentes. Um exemplo das circunstâncias onde isto é útil é o que segue:

A casa com B $\rightarrow$ Descarte B

C casa com D $\rightarrow$ Descarte C

E não casa com A ou D mas pode casar com B e C

Resultado: A,D,E são mantidos separados quando na realidade os mesmos representam a mesma entidade.

Um problema com a abordagem de múltiplos passos é que o número de falsos positivos aumenta à medida que os mesmos são propagados por cada passo.

5.3.3.2 Método de fila de prioridade

Este método é relacionado ao SNM, mas conjuntos de registros representativos pertencentes a clusters recentes na lista de registros são armazenados na lista de prioridade. Heurísticas são necessárias para selecionar estes registros representativos de um *cluster*. A vantagem é que evita a necessidade de ordenar as fontes de dados em cada passo de blocagem, o qual pode economizar um tempo computacional significante para grandes fontes de dados.

5.3.3.3 Blocagem como pré-seleção

A idéia da pré-seleção é baseada em “regras de rejeição” calculadas rapidamente porque quase todas duplas de registros podem ser classificadas como não-par após um simples processamento. Pré-seleção é a aplicação de regras de rejeição adequadas para

reduzir o número de comparações. Regras de rejeição podem ser derivadas de funções de comparações usadas e de uma amostra de treinamento. O tipo de comparação da pré-seleção realizada pode ser ajustado para controlar a negociação entre complexidade da blocagem e a taxa requerida de pares desclassificados.

5.3.4 Comparação

O *framework* probabilístico de Fellegi-Sunter requer o cálculo do vetor de comparação *g* para cada dupla de registros. Atributos de tipo texto são comumente utilizados como campos de comparação entre registros durante o preenchimento do vetor comparação.

A comparação de textos pode ser difícil porque registros lexicamente similares aparentam caracterizar um par quando podem não ser de fato. Por exemplo considere o seguinte campo do tipo texto contendo endereços:

“Apt 11, Bloco B, num. 2601, Centro”

“Apt 12, Bloco B, num. 2601, Centro”

“Apt 13, Bloco B, num. 2601, Centro”

Apesar de bastante similares lexicamente, cada valor representa informações completamente distintas. O uso de funções de comparação que explorem a semântica do texto de maneira dependente do domínio ajuda a resolver este problema. Neste exemplo, *Apt 11* e *Apt 12* poderiam ser separados do restante do texto e rotulados como número de apartamento e uma função de comparação que capture qualquer diferença de caractere como significante seria empregada.

[JM89] introduziu um comparador de *strings* que registra inserções, *deleções* e transposições entre duas *strings*. O algoritmo básico formulado por Jaro possui três etapas principais:

1. calcule do tamanho das *strings*

2. encontre o número de caracteres em comum entre duas *strings*

3. encontre o número de transposições

A definição geral é que o número de caracteres iguais deve ser pelo menos metade do tamanho da *string* menor. A definição de transposição é que o caractere de uma string está fora de ordem com o caractere similar de outra *string*. O resultado final método é:

$$C(s1,s2) = 1/3 * \left(\frac{Ncommon}{Ls1} + \frac{Ncommon}{Ls2} + 0.5 * \frac{Ntranspositions}{Ncommon} \right) \quad (8)$$

Onde *s1* e *s2* são duas *strings* a serem comparadas, com tamanhos *Ls1* e *Ls2* respectivamente. N_{common} e $N_{transpostions}$ são os números de caracteres comuns e transposições.

[PW88] realizou modificações no comparador de *strings* de [JM89] em três pontos:

- Um peso de 0.3 é atribuído para um caractere similar durante a contagem de caracteres similares. O modelo de Winkler de caracteres similares inclui aqueles que podem ocorrer devido a erros de varredura (“1” versus “1”) ou erros de digitação (“V” versus “B”).
- Mais pesos são dados para concordância no começo de uma *string*. Isto é baseado na observação que poucos erros tipográficos ocorrem no início de uma *string* e a taxa de erro aumenta monotonamente nos caracteres seguintes da *string*.
- A comparação dos valores é ajustada se as *strings* são maiores que seis caracteres e mais da metade dos caracteres estão além da primeira quadrupla de concordância.

5.3.5 Modelos de Decisão

Uma vez que pesos de pareamento de atributos individuais são calculados, o próximo passo é combiná-los para compor um peso composto ou escore e então decidir se a dupla de registros constitui um par, não-par ou possível par.

O método mais simples de calcular o peso composto é usar a média de todos os pesos de pareamento, assumindo que cada atributo contribui uniformemente. Se algum conhecimento da importância de atributos individuais está disponível, a média ponderada pode ser usada.

Se a distribuição dos valores de um campo não é uniforme, um peso específico de valor (baseado em freqüência), ou específico de resultados pode ser introduzido. Por exemplo, o sobrenome “Silva” ocorre com uma freqüência bem maior que “Zabrinsky” e desta forma, um pareamento de “Silva” deverá ter um peso menor que um pareamento de “Zabrinsky”. A idéia básica é que uma concordância em valores raros de um atributo é mais determinante para caracterização de um par do que uma concordância sobre um valor comum.

5.3.6 Avaliação dos resultados

A qualidade do resultado de um procedimento de relacionamento de registros pode ser avaliada baseando-se nas seguintes dimensões:

- O número de pares relacionados corretamente (pares verdadeiros) n_m.
- O número de pares relacionados incorretamente (pares falsos, erros de Tipo I) n_{fp}.
- O número de pares não relacionados corretamente (não-pares verdadeiros) n_u.
- O número de pares não relacionados incorretamente (não-pares falsos, Tipo II) n_{fn}.

Conhecendo de antemão o número total de pares verdadeiros, N_m, e número total de não-pares verdadeiros, N_u, (por exemplo utilizando bases de dados geradas artificialmente) várias medidas de diferentes perspectivas podem ser definidas destas dimensões, a saber:

- Sensitividade: n_m / N_m, o número de registros relacionados corretamente pelo sistema dividido pelo número total de registros relacionados.
- Especificidade: n_u / N_u, o número de registros não relacionados corretamente dividido pelo número total de registros não-relacionados.
- Taxa de pareamento: $(n_m + n_{fp}) / N_m$, o número total de pares relacionados dividido pelo número total de registros relacionados.
- Valor previsto de acertos: $n_m / (n_m + n_{fp})$, o número de pares relacionados corretamente dividido pelo número de pares relacionados.

Pode ser observado que a sensitividade mede a porcentagem de registros relacionados corretamente enquanto que a especificidade mede a porcentagem de registros não relacionados corretamente.

Critérios adicionais de performance para o relacionamento de registros são em termos de tempo de processamento e número de registros que requerem revisão manual.

5.4 Sistema MPI no Portal de Teleradiologia

O sistema MPI para relacionamento de registros possui um papel fundamental no Portal de Teleradiologia. Como já referido anteriormente, a maior parte da funcionalidade do sistema depende da capacidade de identificar exames de diferentes servidores como relacionados a um mesmo paciente. No Brasil, apesar de iniciativas como o Cartão SUS para o sistema de saúde público, que ainda está incipiente, não

existe um meio de identificação amplamente utilizado. Identificadores como CPF ou RG não possuem presença garantida em registros clínicos e podem ser inexistentes no caso de menores de idade. Com isso, todo sistema que vise a integração de sistemas de saúde terá que utilizar os métodos de relacionamento de registros.

Este aspecto de um sistema de banco de dados federados, em que o enfoque é dado à integração de dados entre os componentes, e não a interoperabilidade, ainda é um tema pouco explorado, principalmente com aplicação de técnicas de relacionamento de registros. Este termo é referenciado de diferentes maneiras na literatura. [WM89] usa o termo identificação de instâncias entre banco de dados; [BH94] refere-se como identificação de semântica e [ZH00] refere-se como equivalência semântica.

5.4.1 Integração HIS/RIS

Para o emprego de um sistema MPI para a integração de bancos de imagens DICOM o primeiro problema a ser tratado é a ausência dos dados para o relacionamento. Embora o padrão DICOM especifique informações demográficas através dos módulos *PatientModule* e *PatientStudyModule*, o que se tem observado em diversos hospitais e clínicas é que estes atributos não são normalmente preenchidos durante a criação da imagem. Além disso alguns identificadores poderosos como CPF não são cobertos pelo padrão DICOM.

Desta forma, tem-se a existe a necessidade de buscar estas informações ausentes no sistema administrativo do hospital. Isto conduz a outro problema se o sistema RIS do hospital ou clínica não for integrado com o sistema HIS. Pode-se concluir que “bons” participantes do Portal de Teleradiologia possuem uma completa integração HIS/RIS. Estes são capazes de alimentar o esquema federado do sistema com maiores quantidades de informações, possibilitando que seus pacientes sejam identificados em outras bases de dados.

A integração HIS/RIS é uma tarefa bastante dependente do ambiente onde será empregada e foge do escopo deste trabalho um estudo mais detalhado sobre o tema, entretanto algumas considerações podem ser feitas. Mesmo que completamente isolados entre si, é bastante comum que sistemas HIS e RIS compartilhem o mesmo identificador para pacientes.. Com isso é possível aplicar os métodos de relacionamento de registros determinísticos, que são relativamente de fácil implementação. Para a comparação dos

registros pode ser necessário ainda aplicar algum tipo de racionalização estrutural ou até mesmo semântica. [1]

Outro ponto a ser considerado é quanto ao conteúdo e estruturação dos campos de identificação que serão enviados para o sistema MPI. Nos subconjunto dos campos de identificação que forem cobertos pelo padrão DICOM, basta seguir o mesmo para garantir uniformidade semântica e estrutural entre os dados. O problema estaria nos atributos não presentes no padrão DICOM como CPF. Desta forma, tem-se a necessidade da definição de um modelo de dados comum entre o Portal de Teleradiologia e seus integrantes. Também será necessário converter as informações do subsistema de integração HIS/RIS para um formato comum antes de enviá-las para o sistema MPI do Portal de Teleradiologia. Este assunto será abordado novamente na seção 6.3.3 quando for discutida a estrutura do nível auxiliar para a arquitetura do Portal de Teleradiologia.

5.4.2 Sistema MPI no esquema federado

As informações que alimentarão o sistema MPI do Portal de Teleradiologia terão uma natureza heterogênea quanto ao seu conteúdo. Apesar da padronização com a utilização de um CDM, não existe garantia de todos os integrantes do sistema federado enviarão todos os valores para cada atributo. Por exemplo, um campo de identificação como o número de identificação SUS estará em branco para pacientes que não utilizarem serviços públicos.

Além disso, é esperada uma grande quantidade de erros tipográficos nos dados enviados ao sistema MPI. O Brasil possui uma grande diversidade étnica e variações em nomes e sobrenomes são bastante comuns. Na região Sul do país, onde existe predominância de nomes e sobrenomes de origens alemã e italiana é previsto que erros de digitação ocorram com mais freqüência.

Todas estas considerações acima conduzem à preferência pela abordagem probabilística para o sistema MPI do Portal de Radiologia. A estratégia probabilística é considerada mais adequada para bases de dados de baixa qualidade [GR03], [MH03],

---

[1] É fácil observar que a integração de dois sistemas autônomos com o HIS e RIS situa-se na área de banco de dados federados e portanto, pode empregar sua metodologia. Com isso o esquema federado do Portal de Teleradiologia terá como sistemas componentes outros esquemas federados.

[RW91], apesar de que, como observado por [GC04], ainda exista uma carência de estudos que comprovem isto. Outro argumento a favor da abordagem probabilística é a grande freqüência de utilização do sistema relacionamento de registros. Como já explanado anteriormente, em um sistema MPI o relacionamento de registro ocorre sempre durante a inserção e consulta de registros.

Diversas técnicas para suporte ou em paralelo aos métodos de decisão da abordagem probabilística poderão ser usadas. Entre elas estão redes neurais, processamento de sinais, lógico *fuzzy*, *clustering* e raciocínio baseado em casos. Como um sistema MPI requer tempo de resposta pequeno, por exemplo no caso de consultas, é necessária a alocação de um grande poder de processamento. Apesar de fazer parte logicamente do processador de construção do esquema federado, o sistema MPI poderá estar fisicamente localizado em outro computador ou até mesmo em *cluster* e utilizando técnicas de processamento paralelo. [MS02] descreve um *framework* para MPI que utiliza um *cluster* de baixo custo formado por computadores pessoais e sistema operacional Linux.

6. PORTAL DE TELERADIOLOGIA

Neste capítulo será apresentado o Portal de Teleradiologia, um modelo de sistema federado, para a integração de bancos de imagens médicas digitais em conformidade com o padrão DICOM 3.0. O objetivo do Portal de Teleradiologia é prover uma visão única e transparente dos dados armazenados em servidores distribuídos. Apesar de ser direcionado à integração de bancos de imagens, o sistema prevê a extensão de seu escopo para integração de serviços de prontuário eletrônico.

A tarefa de integrar e prover um acesso único para bases de dados distribuídas e autônomas representa um grande desafio. A primeira dificuldade é heterogeneidade estrutural e semântica dos dados entre sistemas projetados e implementados de maneira independente. As diferenças estruturais impossibilitam a comunicação e a correta formatação de um fluxo de dados entre sistemas distintos. Diferenças semânticas dificultam a concordância entre dois sistemas entre o significado, interpretação, finalidade e identificação de similaridades entre os objetos de informação. Somado a isto ainda existem as diferenças nas funcionalidades de cada sistema, com definições de serviços e protocolos de aplicação distintos.

Além destes problemas inerentes à construção de um sistema federado, a integração de sistema de ambientes médicos apresenta outros fatores que aumentam a complexidade da tarefa. O primeiro ponto é a relativa imaturidade do mercado de informática médica. Apesar do grande volume de dados produzido e dos anos de experiência na aplicação bem-sucedida das tecnologias de informação em outras indústrias não é raro encontrar hospitais com boa parte de seus dados processados manualmente. A informatização em um hospital ou clínica é feita muitas vezes de maneira *ad-hoc* entre seus departamentos, fazendo com que estes subsistemas coexistam sem qualquer interoperabilidade entre si.

Existem várias razões para este déficit de informatização no ambiente médico, incluindo baixos investimentos, mercado fragmentado, falta de regulamentação, principalmente com relação a telemedicina, carência de padrões ou lenta adoção aos existentes. Outras razões são específicas ao ambiente médico, como a complexidade dos dados médicos, questões de segurança e confidencialidade e a falta de um identificador único de paciente em muitos países.

Diversas iniciativas para padronização de sistemas informações médicas têm sido feitas. Na parte de mensagens e registros clínicos destaca-se o HL-7, que apresenta um detalhado modelo de dados e codificação de mensagens baseado em XML. No Brasil uma iniciativa de padronização de registros clínicos baseada no HL-7 foi realizada pelo DataSUS [DS00] em 2000. Ainda na padronização de modelos de dados, destacam-se as atividades do CEN TC251 (*Comitié Européen de Normalisation Technical Commitee 251*) em especial o projeto *Synapses* [CT25]. Na parte da padronização de serviços e *middleware* destacam-se o *Healthcare* DTF (antigo CORBAMed) [OH03] e o HISA (*Healthcare Information Systems Architecture*) [HI00]. Todos estes padrões entretanto, tem experimentado uma adoção bastante lenta no mercado de informática médica, mesmo em regiões com maior nível de desenvolvimento tecnológico como a América do Norte e a Europa. No Brasil em especial a adoção tem sido praticamente inexistente. Até mesmo iniciativas embasadas por órgão governamentais como a padronização proposta pelo DataSUS tem recebido pouca atenção até o momento.

Uma exceção a este cenário é o padrão DICOM, um padrão *de-facto* com ampla aceitação mundial. Inicialmente direcionado apenas aos sistemas RIS e PACS, o padrão vem aumentando seu escopo, principalmente após a adição da modalidade *Structure Report*, para sistemas de prontuário eletrônico. Esta uniformidade proporcionada pela utilização de um padrão como o DICOM poderia minimizar bastante a tarefa de construir um sistema federado, principalmente se as bases de dados integrantes restringir-se a bancos de imagens médicas digitais. Entretanto, alguns problemas relacionados ao padrão DICOM e outras questões específicas do domínio de aplicação impedem que se utilize apenas o modelo de dados, de serviços e semântico do DICOM na arquitetura federada. Este assunto é detalhado nas seções 6.3.3.

Apesar de remover grande parte dos obstáculos do caminho, a padronização de informações ainda não provê uma completa integração e interoperabilidade entre bases de dados clínicos ou radiológicos. Ainda é necessário identificar os registros oriundos de bases distintas e independentes que sejam relacionados a um mesmo paciente. Em outras palavras, é preciso ter meios para relacionar os registros entre as organizações participantes da federação e determinar sua equivalência semântica. A inexistência de um identificador único de paciente, como é o caso do Brasil, criam a necessidade de empregar um sistema de MPI dentro do sistema federado. Uma questão que surge desta

necessidade é como alimentar o sistema MPI com informações de cada integrante do sistema federado.

A estrutura deste capítulo é a que segue. A seção 6.1 discute os protocolos utilizados no Portal de Teleradiologia. A seção 6.2 apresenta os domínios de operação que compõem o sistema federado. A seção 6.3 apresenta em detalhes a arquitetura de do Portal de Teleradiologia, seus principais componentes e domínios. A seção 6.4 discute as questões relacionadas aos mecanismos e políticas de segurança. O subsistema para relacionamento de registros é discutido na seção 6.4. Finalmente, na seção 6.5 é discutido o protótipo implementado.

6.1 Protocolos do Portal de Teleradiologia

O modelo de informações de interesse do Portal de Teleradiologia, no caso objetos de informação DICOM, possuem características que facilitam a tarefa de integração com relação a outros modelos de informações em alguns aspectos mas também impõem dificuldades adicionais em outros. Parte destas características é intrínseca ao contexto de informações médicas e suas políticas de operação. Outra parte deriva-se do próprio padrão DICOM e de sua abordagem para modelar e manipular este contexto de informação. Todos estes aspectos guiaram a definição dos protocolos utilizados para implementação dos serviços do Portal de Teleradiologia.

Primeiro será abordado os elementos que facilitam a implementação de um sistema federado de servidores DICOM. A ausência operações de atualização é o principal deles. As naturezas sensitivas dos dados clínicos coíbem a utilização deste tipo de operação em um contexto mais amplo. O padrão DICOM prevê operações de atualizações em objetos DICOM através de classes de serviços para gerenciamento de informações de paciente e estudo. Até mesmo modalidades como *Structered Reporting* possuem especificações que abordam operações de atualização através de classes SOP específicas de armazenamento [DC04]. Entretanto, durante a etapa de especificação de requisitos deste trabalho considerou-se este tipo de operação como extremamente dependente do ambiente local e de suas políticas de operação. Além disso, operações desta natureza em um ambiente federativo poderiam ultrapassar os limites da ética médica, pelo menos no atual estágio em que se encontra o debate sobre a prática da Telemedicina. Portanto, o atual modelo de operação do Portal de Teleradiologia não

contempla operações de atualização. Isto elimina um dos principais problemas em sistemas federados que é o gerenciamento de transações.

Outro aspecto importante é harmonização estrutural proporcionada pelas especificações de elementos de dados e modelagem do mundo-real do padrão DICOM. A homogeneidade estrutural dispensa a tradução dos modelos dados locais para uma representação em comum o que a princípio dispensaria a definição de um esquema componente. A ausência de informações suficientes nos objetos de informação DICOM para alimentar um sistema MPI conduz a necessidade de um esquema auxiliar representado por um módulo de integração HIS/RIS. Este assunto será tratado com mais detalhes na seção 6.3.3.

A modelagem fortemente orientada a objeto adotada pelo padrão DICOM após a versão 3.0 facilita a aplicação de técnicas de sistemas federados destinadas ao modelo de dados objetos-relacionais. A utilização de *wrappers* e uniformização sobre a visão dos dados e operações, principalmente para extensão do escopo do sistema para cobrir outros tipos de registros clínicos, também são facilitadas.

Com relação aos aspectos negativos, o DICOM possui características em seu protocolo de comunicação que limitam tua aplicabilidade em um ambiente de cooperação entre vários componentes. Para o estabelecimento de uma associação entre duas entidades de aplicação DICOM, é utilizado o serviço ACSE (*Association Control Service Element*) A-ASSOCIATE. Os parâmetros de uma mensagem A-ASSOCIATE incluem o endereço de rede e título da entidade de aplicação das duas entidades comunicantes. Estes parâmetros são utilizados pela entidade de aplicação que recebe a requisição para aceitar ou rejeitar uma associação. Desta forma, é requerido um conhecimento mútuo destes parâmetros entre duas entidades de aplicação para que as mesmas estabeleçam uma associação. O padrão DICOM não suporta meios para configuração dinâmica destes parâmetros.

Outro fato é que o elemento de serviço mais utilizado para transferência de imagens pelo DICOM, o C-MOVE, possui um modo de operação similar ao protocolo FTP em seu modo de operação normal, ou seja, durante a transferência das imagens médicas, o servidor estabelece uma conexão com o cliente como suboperação, no papel de cliente. Estas características dos serviços de comunicação do padrão DICOM dificultam, podendo até mesmo inviabilizar, a conexão de clientes que não possuam

endereço IP fixo ou que utilizam a Internet através de um serviço NAT (*Network Address Translator*). O suporte à rede do DICOM é, portanto, inadequado como canal de comunicação em um sistema onde clientes possam acessar os serviços do Portal de Teleradiologia, independentemente de sua localidade.

Com relação à comunicação entre os servidores de imagens e o sistema, é aceitável manter os serviços de comunicação DICOM, pois é necessário possuir um IP fixo ou meios de fazer com os segmentos de transporte cheguem ao servidor como o DNAT (*Destination NAT)* de qualquer maneira. A configuração dos servidores de imagens não é problema pois apenas o Portal de Teleradiologia estabelecerá uma associação diretamente com os servidores.

Outra questão importante sobre os mecanismos de comunicação usados para a comunicação entre os servidores DICOM e Portal de Teleradiologia é a definição das classes de serviços usadas nesta comunicação. O padrão DICOM especifica atualmente quinze classes de serviços para diferentes tipos de operação (e.g. verificação, armazenamento, gerenciamento de impressão) e diferentes tipos de objetos de informação (e.g. modalidades de imagens, laudos estruturados, fila de impressão). Para o escopo deste trabalho são pertinentes apenas as classes de serviços relacionadas a objetos de informação compostos e aos serviços de consulta, armazenamento e recuperação. Dentro deste subconjunto estão as classes de serviços básicas *Storage* e *Query/Retrieve*. Outras classes de serviço são especializadas para determinadas modalidades, como a *Storage Structure Reporting*, entidades de informação, como a *Study Management* e serviços específicos, como a *Storage Commitment*. Este grupo de classes de serviços especializadas apesar de prover operações mais sofisticadas, ainda são pouco suportadas pela maioria das implementações DICOM. Desta maneira, entre as informações no catálogo do sistema federado é necessário manter dados sobre as classes de serviço e classes SOP suportadas por cada servidor DICOM componente. Estas informações podem ser obtidas estaticamente durante a fase de integração ou dinamicamente através dos serviços de associação. No protótipo desenvolvido, para assegurar a compatibilidade com o maior número de implementações existentes e simplificar o gerenciamento da comunicação com os servidores DICOM optou-se por utilizar apenas as classes de serviços básicas.

Para a comunicação de dados entre os clientes e o Portal de Teleradiologia foi escolhida a arquitetura CORBA. Os motivos para a escolha de CORBA são vários. A independência de plataforma de hardware e software são os motivos mais óbvios. A grande flexibilidade de implementação de serviços, através dos métodos de invocação dinâmica e de componentes como os repositórios de interfaces e implementação facilitam a manutenção e extensão da arquitetura federada. Além disso, nos testes realizados durante o desenvolvimento do protótipo, CORBA apresentou uma excelente performance para transmissão de grandes volumes de dados, no caso imagens DICOM[1].

É prevista uma futura extensão do Portal de Teleradiologia para prover acesso a prontuários eletrônicos de paciente, além de imagens radiológicas digitais. Nesta nova arquitetura seria necessário encapsular todo o ambiente de informação de um hospital ou clínica para manter um modelo de dados comum entre a arquitetura ocultar a heterogeneidade oriunda dos sistemas HIS, onde normalmente encontram-se as informações de prontuário eletrônico. Novamente, uma solução baseada em CORBA é adequada. O paradigma da orientação a objetos é reconhecido por diversos autores [LM84], [LM91], [RE96] com adequada para tratar o problema da integração de bases de dados autônomas. As características inerentes do modelo objetos mais atrativas para a implementação de um sistema federado são o encapsulamento e a herança/especialização. A primeira permite ocultar a heterogeneidade de interfaces e implementação entre os componentes do sistema enquanto que a segunda é útil para o gerenciamento da interoperabilidade permitindo a criação de tipos que abstraem a similaridade das entidades armazenadas em diferentes bancos de dados. [FH93], apresenta um modelo para sistemas federados apoiado na utilização de agentes e objetos distribuídos e [WX01], [RK00], [OV91] apresentam soluções usando CORBA em específico.

6.2 Ambientes de operação

Com relação aos ambientes de operação, a arquitetura do Portal de Teleradiologia é dividida claramente em três componentes principais, a saber:

---

[1] Implementação CORBA *Distributed Smalltalk*, presente no ambiente de programação *Cincom Visualworks*.

- Ambiente de integração. Neste ambiente onde é concentrado todo gerenciamento da arquitetura federada. Módulos funcionais da federação como o módulo de controle de acesso, sistema MPI, processador construtor de consultas distribuídas e gerenciador de meta-dados são todos executados neste ambiente. Sob o aspecto da infra-estrutura de *hardware* este ambiente de operação deve possuir grande poder de processamento. Apesar de constituir logicamente apenas um ambiente de operação, fisicamente o processamento dos módulos funcionais pode estar distribuídos em diferentes computadores. É de extrema importância que este ambiente possua uma relação privilegiada com a rede, ou seja possua conexão direta com um *backbone* ou rede de comunicação de alta velocidade. Arquitetura redundante e tolerante a falhas também são requisitos fundamentais. Quanto ao aspecto de *software*, é imprescindível que o ambiente de integração implemente um *Trusted Computing Base* (TCB) que atenda ao critério de classificação B, para proteção obrigatória sobre informações classificadas, apresentado em [TCa85][1].
- Ambiente de usuário. Este ambiente é formado pelas diversas classes de usuários da federação. A principal característica deste ambiente é a ampla heterogeneidade de software e hardware. A escolha de CORBA como *middleware* entre clientes e o sistema federado permite uma extensa flexibilidade quanto à configuração do ambiente usuário. Desta forma, pode-se ter desde o cenário de um radiologista usando um sistema a partir de uma poderosa *workstation* para laudos, com um canal de comunicação de grande vazão a um médico utilizando o sistema a partir de sua residência através de um computador pessoal e uma conexão por discagem via *modem* e canal físico PSTN (*Public Switched Telephone Network*).
- Ambiente provedor de dados. Este domínio compreende todos os bancos de imagens no padrão DICOM 3.0 aos quais o sistema federado possua acesso. Sob o aspecto de hardware este ambiente é bastante heterogêneo. Um servidor DICOM pode estar hospedado tanto no conjunto de equipamentos que compõem um aparelho de

[1] Entre o amplo conjunto de procedimentos especificados nas subclasses deste critério de avaliação estão a auditoria, identificação e autenticação, aplicação de correções de segurança em softwares, utilização de senhas fortes, métodos para segurança física, *backups*, entre vários outros. Em suma, o ambiente deve ser protegido contra a inserção de lógica maliciosa e caracterizado como um sistema de segurança fechado de acordo com a classificação de [TCb85].

tomografia quanto em um computador pessoal. A conexão de rede existente neste ambiente também é bastante heterogênea, sendo bastante dependente dos recursos e necessidades de uma organização em específico. Sob o aspecto de software o padrão DICOM provê uma grande uniformidade quanto ao modelo de dados transferidos entre o Portal de Teleradiologia e os provedores de dados quando atendendo a requisições dos clientes, independente de características específicas de implementação, como linguagem de programação. Entretanto a necessidade de um esquema auxiliar para alimentar o sistema MPI do Portal e conseqüente necessidade de interface com sistema HIS adiciona um amplo espectro de heterogeneidade, pois os sistemas HIS normalmente encontrados nos hospitais e clínicas não seguem qualquer tipo de padronização.

A figura 9 representa estes três ambientes de operação que juntos compõe o escopo do Portal de Teleradiologia. As abordagens para tratar a integração destes ambientes baseiam-se na arquitetura de cinco níveis apresentada em [SL90] e são apresentadas na seção seguinte.

6.3 Arquitetura do Portal de Teleradiologia

Esta seção provê uma visão geral sobre a arquitetura do Portal de Teleradiologia e de seus principais componentes. A abordagem utilizada neste trabalho define que cada componente do sistema seja autônomo e independente, ou seja, um hospital ou clínica não terá que modificar a maneira de lidar com seus dados e nem sofrerá qualquer espécie de controle externo para participar do modelo proposto. Com isso, a tarefa de integrar servidores DICOM autônomos pode facilmente ser classificada no campo de pesquisa de Banco de Dados Federados. Os principais conceitos que caracterizam um sistema federado estão presentes como a autonomia de componentes, compartilhamento controlado e parcial dos dados, e dicotomia entre operações locais e externas. Mais especificamente, a arquitetura proposta é categorizada como um sistema federado fortemente acoplado. Ou seja, toda a administração da federação e o controle de acesso aos dados dos sistemas locais são realizados pelo servidor de integração que representa a federação. Os usuários do sistema federado não possuem qualquer conhecimento sobre a distribuição dos dados, "enxergando" a federação como um único banco de dados.

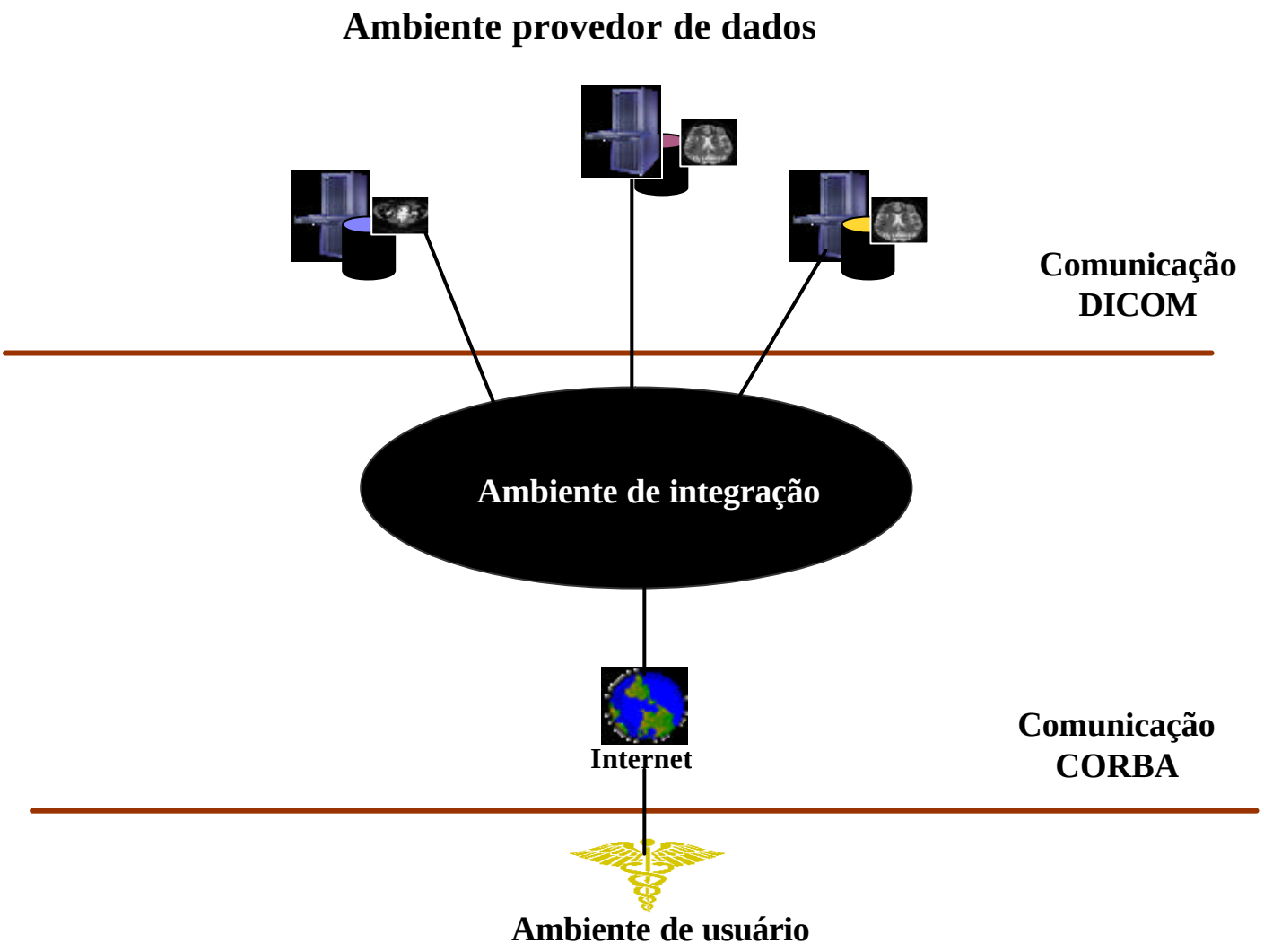

Figura 10: Ambientes de operação do Portal de Teleradiologia

A arquitetura utilizada para descrever o Portal de Teleradiologia é baseada na arquitetura de referência apresentada em [SL90], e representada nas figuras 3 e 4. Componentes adicionais são necessários para atender a requisitos específicos do sistema. Características intrínsecas do padrão DICOM aplicam ainda algumas variações a arquitetura básica de [SL90]. A seguir será descrito cada um dos níveis desta arquitetura. Em cada nível, serão ressaltados os esquemas e processadores associados, e as questões pertinentes identificadas juntamente com a abordagem proposta para tratá-las. A seqüência adotada será a *bottom-up*, ou seja partindo do nível local em direção ao

nível externo da federação, que representa as visões do Portal de Teleradiologia para seus usuários.

6.3.1 Nível local

O nível local do Portal de Teleradiologia é caracterizado pela homogeneidade estrutural e semântica provida pelo padrão DICOM. O modelo de informações descrito através do modelo entidade-relacionamento apresentado na figura 4 representa o modelo de dados uniformizado entre os esquemas locais. A manipulação dos dados é realizada através dos elementos de serviços presentes nas classes de serviço *Storage* e *Query/Retrieve* definidas em [DM04]. Em específico, são utilizados os elementos de serviço C-STORE DIMSE-C , C-FIND DIMSE-C e C-MOVE DIMSE-C. O esquema local também é composto pelas informações necessárias para estabelecimento da associação como o título da entidade de aplicação e o endereço de rede que recebe as requisições de associação.

O padrão DICOM não especifica nada a respeito de como as informações administradas por uma entidade de aplicação devam ser armazenadas. Esta decisão é deixada a critério da implementação. Desta forma, a camada de armazenamento pode variar de uma simples estrutura de diretórios a um poderoso SGBD relacional. Neste último caso, o modelo de dados utilizado por DICOM facilita bastante sua conversão para um esquema conceitual descrito através de modelo lógico baseado em registros. Desta forma, a entidade de aplicação DICOM representa o nível externo do DBS, definindo a visão dos dados para o usuário.

Um exemplo de uma adaptação do modelo de dados DICOM para um modelo relacional está em [DP01], trabalho que resultou no desenvolvimento de um servidor DICOM, o CyclopsDICOMServer. Na modelagem utilizada, cada entidade de aplicação DICOM associada ao CyclopsDICOMServer é mapeada em um banco de dados diferente. O diagrama do esquema descritivo adotado é baseado no modelo de informação *Patient Root*[1] da classe de serviços *Query/Retrieve* [DC04]. Este modelo organiza a informação de um objeto DICOM em quatro níveis hierárquicos. Em ordem

[1] O CyclopsDicomServer também suporta o modelo de informações *Study Root*, embora este modelo não esteja representado no esquema descritivo relacional do DBS de armazenamento. Os mapeamentos entre uma requisição com uma classe SOP *Study Root* e esquema do DBS são realizados intemamente.

decrescente, são eles *Patient*, *Study*, *Series* e *Image*. Cada um destes níveis possui um relacionamento de cardinalidade 1:n com o nível inferior. O padrão DICOM especifica para cada nível os atributos requeridos, únicos e um conjunto de atributos opcionais que são dependentes do nível de detalhamento desejado para as operações de consulta e recuperação. Esta modelagem é espelhada pelo CyclopsDicomServer no esquema do DBS de armazenamento, com cada nível representado por uma tabela. Os atributos destas tabelas são utilizados apenas para o tratamento das operações de consulta e recuperação. A instância do objeto de informação, juntamente com dados de *pixel* quando presentes, são mantidos em outra tabela, onde todo o conjunto de dados do objeto DICOM é armazenado em atributo do tipo BLOB (*Binary Large Object*).

Abaixo está a descrição do esquema relacional usado pelo servidor DICOM para armazenar instâncias de objetos DICOM no DBS *PostgresSQL*. A notação utilizada é retirada dos comandos para criação das tabelas a partir do ambiente de programação *smalltalk VisualWorks*. É importante notar que por decisão de implementação o controle de integridade referencial é realizado na aplicação, e não são utilizados os mecanismos do DBS.

```
create table patients (
    "patientsName"        varchar(64),
    "patientID"           varchar(64),
    "patientsBirthDate"   date,
    "patientsBirthTime"   time,
    "patientsSex"         varchar(16),
    PRIMARY KEY           ("patientID"),
    UNIQUE                ("patientID"))

'create table studies (
    "studyDate"             date,
    "studyTime"         time,
    "accessionNumber"   varchar(16),
    "patientsName"      varchar(64),
    "patientID"         varchar(64),
    "studyID"           varchar(16),
    "studyInstanceUID" varchar(64),
    PRIMARY KEY        ("studyInstanceUID"),
    UNIQUE                 ("studyInstanceUID"))'.

'create table series (
    "modality"                          varchar(16),
    "seriesNumber"                      varchar(12),
    "seriesInstanceUID"                 varchar(64),
    "numberOfSeriesRelatedInstances"    integer,
    "seriesDate"                        date,
```

```
    "seriesTime"                    time,
    "bodyPartExamined"              varchar(16),
    "studyInstanceUID"              varchar(64),
    PRIMARY KEY                    ("seriesInstanceUID"),
    UNIQUE                        ("seriesInstanceUID"))'

'create table images (
    "instanceNumber"          varchar(12),
    "overlayNumber"           varchar(12),
    "curveNumber"             varchar(12),
    "lutNumber"               varchar(12),
    "sopInstanceUID"          varchar(64),
    "seriesInstanceUID"       varchar(64),
    "studyInstanceUID"        varchar(64),
    "patientID"               varchar(64),
    "instanceCreationDate"    date,
    "instanceCreationTime"    time,
    PRIMARY KEY               ("sopInstanceUID"),
    UNIQUE                    ("sopInstanceUID"))'

'create table imageData (
    "sopInstanceUID"       varchar(64),
    "seriesInstanceUID"    varchar(64),
    "studyInstanceUID"     varchar(64),
    "patientID"            varchar(64),
    imageBlobOID           oid,
    PRIMARY KEY            ("sopInstanceUID"),
    UNIQUE                 ("sopInstanceUID"))'
```

O CyclopsDICOMServer foi desenvolvido no mesmo âmbito deste trabalho, o projeto Cyclops e foi o principal servidor DICOM utilizado durante o desenvolvimento do protótipo deste trabalho. Desta forma, ele representa o principal modelo de referência para servidores DICOM. Entretanto durante a realização deste trabalho evitou-se "amarrar" qualquer decisão de projeto ao CyclopsDicomServer, mantendo o requisito inicial de independência quanto a implementações DICOM específicas.

6.3.2 Nível componente

Como já explanado na seção 2.4.2, a função do esquema componente é descrever esquemas locais divergentes através de uma representação única e capturar informações semânticas que não estejam descritas nos esquemas locais. Estas funções não são necessárias no modelo do Portal de Teleradiologia. O padrão DICOM confere uma uniformidade estrutural e semântica aos esquemas locais e dispensa a necessidade de um esquema componente e o processador de transformação correspondente. O sistema

federado utiliza os elementos de serviço e associação DICOM para comunicar-se com os servidores DICOM no papel de SCU. A estruturação do conjunto de dados transmitidos durante a associação seguem a mesma estruturação definida pela sintaxe de transferência negociada. Desta forma, na arquitetura do Portal de Teleradiologia não está definido um esquema componente.

6.3.3 Nível auxiliar

Com já abordado na seção 5.5.1, apenas o conjunto de dados DICOM é insuficiente para alimentar um sistema MPI. Isto se deve a dois fatores. O primeiro é representado por um obstáculo cultural, pois grande parte dos campos interessantes para aplicação de algoritmos e técnicas de relacionamento de registros não são preenchidos durante a produção da imagem, pelo menos no âmbito de hospitais e clínicas brasileiros. Por exemplo, os valores dos atributos DICOM *Patient's Address*, *Patient's Birth Time*, *Patient's Birth Date* não estão, na grande maioria dos casos, presentes em conjunto de dados DICOM[1]. O segundo fator é a ausência na modelagem do padrão DICOM de alguns identificadores poderosos como CPF e RG.

A solução é buscar estas informações no sistema de informação hospitalar, promovendo a chamada integração HIS/RIS. Este módulo de integração é construído através da integração entre os esquemas locais do sistema RIS, formado pelos bancos de imagens e o do sistema HIS. Esta tarefa em si, representa a criação de um sistema federado, através da integração de duas bases de dados autônomas formadas pelos bancos DICOM e os sistemas de gestão hospitalar.

As informações provenientes do módulo de integração HIS/RIS são complementares ao modelo de informação alvo, provendo aspectos semânticos que não estão presentes nas fontes dos dados. Estas informações entretanto, são apenas auxiliares para o modelo de integração proposto e não fazem parte do domínio de informação principal. Desta forma, na modelagem proposta neste trabalho o esquema resultante desta integração HIS/RIS é classificado como um esquema auxiliar e não como esquema componente. Entretanto, é prevista a expansão do escopo do Portal de

[1] Inferência a partir de anos de experiência do Projeto Cyclops com imagens oriundas de diversas clínicas e hospitais parceiros do projeto. Não representa entretanto um dado científico, pois não foram utilizadas técnicas necessárias para produção estatística.

Teleradiologia em trabalhos futuros para promoção da integração de prontuário de pacientes. Neste novo contexto, o conteúdo da integração HIS/RIS torna-se o principal campo de informações do sistema federado e seu esquema descritivo representará um esquema componente da federação.

A principal dificuldade da tarefa de integração HIS/RIS tem sua origem na heterogeneidade estrutural dos sistemas HIS, que pode ser resultante tanto da modelagem quanto do conteúdo dos esquemas fontes. Como já explanado anteriormente, a inexistência de um padrão mundial *de-facto* para este modelo de informações conduz inevitavelmente a um ambiente de grande diversidade de soluções. Outro problema é a existência ainda bastante comum dos chamados sistemas legados nestes ambientes, onde é comum a inexistência de um esquema conceitual definido e boa parte da estruturação dos dados encontra-se *hard-coded* nas aplicações. Estas características impedem uma solução universal, obrigando que o problema seja tratado caso a caso e com necessidade de intervenção manual. Por outro lado, o modelo de informações de interesse é reduzido, resumindo-se a informações demográficas do paciente, de maneira que o problema de harmonização destas bases de dados restringe-se a poucas entidades de informação. A integração semântica é facilitada pelo fato de normalmente os sistemas radiológicos e hospitalares em uma organização compartilharem o mesmo identificador.

Quanto à definição do modelo de informação para o esquema auxiliar, a abordagem proposta neste trabalho é estender os objetos de informação DICOM através de atributos privados adicionais Tipo 3. Estes atributos não modificarão a semântica da classe SOP associada, mantendo o UID original. A classe SOP estendida será divulgada no Conformance *Statement* do módulo de integração [DM02]. Desta forma, limita-se a incongruência vertical, que é a diversidade estrutural entre o esquema alvo e os esquema fontes, a apenas um esquema fonte, no caso o esquema proveniente do sistema HIS. A padronização proporcionada pelo DICOM permite aumentar o nível de automatismo durante a etapa de integração e definição de mapeamentos.

A definição dos atributos específicos que estarão presentes na modelagem do sistema de integração HIS/RIS deverá ser feita após um levantamento cuidadoso das modelagens de diferentes sistemas hospitalares. Os critérios para inclusão de um determinado atributo são o peso de comparação para o sistema MPI, constância do

atributo entre as modelagens dos sistemas fonte. A idéia é uniformizar ao máximo o conjunto de dados que irão alimentar o sistema MPI, incluindo apenas atributos que estejam presentes na maioria das modelagens dos sistemas hospitalares. Esta tarefa está prevista nos trabalhos futuros deste trabalho, quando for desenvolvido o sistema MPI.

A comunicação entre o módulo de integração HIS/RIS e o sistema federado será realizada através da classe SOP *Detached Patient Management*. Esta classe SOP permite que entidades de aplicação requisitem transferência de informação sobre pacientes para outra entidade de aplicação. Ela pode ser usada, por exemplo para promover o intercâmbio e harmonização entre atributos de pacientes entre os sistemas RIS e HIS. O padrão DICOM não especifica detalhes sobre como será realizada a consulta um sistema HIS pré-existente, o que é deixado a critério específico de implementação [DM04].

No nível auxiliar do Portal de Teleradiologia, os serviços da classe SOP *Detached Patient Management* são utilizados para que a entidade de aplicação do nível de exportação busque informações adicionais sobre um paciente através do módulo de integração RIS/HIS. Desta a forma, a comunicação entre o sistema federado e módulo de integração HIS/RIS é toda realizada através de serviços de aplicação padronizados pelo DICOM. A entidade de aplicação do nível de exportação desempenha o papel de SCU e o módulo de integração HIS/RIS desempenha o papel de SCP. O elemento de serviço utilizado é o DIMSE N-GET e os atributos do IOD normalizado transmitidos com este serviço estão definidos na tabela E.3-2 em [DM04]. No conjunto de dados transmitidos na primitiva de requisição do nível federado, o atributo *Referenced Study Sequence* não estará presente e terá valor nulo na primitiva de resposta do módulo de integração HIS/RIS. Além disso, a classe SOP será estendida para conter os atributos específicos que estiverem presentes na modelagem do esquema auxiliar e que não estejam contidos no dicionário de dados do padrão DICOM.

6.3.4 Nível de exportação

A principal função do nível de exportação é manter a autonomia de associação entre os integrantes do sistema federado. Este nível é composto pelo esquema de exportação, contendo a descrição dos dados dos esquemas locais dispostos para a federação, e o processador de filtragem. Na arquitetura do Portal de Teleradiologia, estes componentes são representados por uma (ou mais) entidade(s) de aplicação

DICOM. Esta entidade de aplicação de exportação implementará a classe de serviço *Storage* e *Query/Retrieve* com o papel de SCP para receber objetos DICOM das entidades locais e atender as requisições do sistema federado. Desta forma, para dispor objetos DICOM para o sistema federado, basta para um usuário local enviar esta imagem para a entidade de aplicação de exportação. O processador de filtragem é implementado na própria entidade de aplicação de exportação, que aceitará requisições de associação DICOM externas apenas do Portal de Teleradiologia. A decisão entre concentrar todos os estudos dispostos para o sistema federado em uma única entidade de aplicação provedora ou em várias é específica para cada integrante do sistema. Entretanto, a manutenção em uma única entidade de aplicação facilita o gerenciamento do esquema de exportação e sua possibilita sua integração com o módulo de integração HIS/RIS em um único aplicativo.

6.3.4.1 Serviço de notificação

O sistema federado do Portal de Teleradiologia, além da descrição do esquema federado, contem partes das instâncias dos bancos de imagens componentes que estão disponíveis para a federação. Existem duas razões para isso. O primeiro motivo é promover a integração semântica em nível de instância através do sistema MPI. O outro motivo é concentrar toda a política de controle de acesso sobre os estudos no nível federado. A concepção do Portal de Teleradiologia não prevê qualquer relação entre os usuários da federação e os usuários locais dos sistemas componentes. Este dois grupos de usuários podem ser completamente disjuntos. A realização do controle de acesso nos sistemas locais implicaria em replicar as informações de todos os usuários da federação em todos os integrantes, o que tornaria o gerenciamento de usuários ineficiente. Para aplicar o controle de acesso no servidor de integração da federação é necessário que os objetos de controle estejam presentes no nível federado, no caso objetos de informação DICOM até o nível de estudo.

O mecanismo de comunicação utilizado pelos integrantes da federação para enviar os objetos de informação DICOM para o SGBDF é a classe SOP *Basic Study Content Notification*. Esta classe SOP permite uma entidade de aplicação notificar outra entidade de aplicação sobre a existência, conteúdo e localização das informações associadas a um estudo. A operação é realizada através do elemento de serviço DIMSE C-STORE, o mesmo utilizado pela classe de serviço *Storage*. Neste caso será a entidade de aplicação

do nível de exportação terá o papel de SCU enquanto o SGBDF provê o papel de SCP desta classe SOP.

O subconjunto do *Study Component* IOD transmitido entre as entidades de aplicação é constituído pelas informações do estudo[1], juntamente com os atributos adicionais obtidos do módulo de integração HIS/RIS. Desta forma, antes do envio da notificação para o SGBDF sobre a inclusão de um novo estudo para o sistema federado, é realizada uma consulta ao módulo de integração HIS/RIS através da classe SOP *Detached Patient Management*, para obter as informações adicionais sobre o paciente. O IOD recebido na primitiva de resposta retornada pelo SCP do módulo HIS/RIS é integrado com os atributos do IOD *Basic Study Descriptor* e enviado para o sistema federado. No servidor de integração parte dos atributos deste IOD com as informações de identificação do paciente é utilizada como entrada para o sistema MPI e outra parte é mantida no catálogo do sistema onde serão aplicadas as políticas de controle de acesso. O resultado do processamento do sistema MPI, um identificador único para o paciente é mantido vinculado com as informações sobre a localização do estudo.

Toda a comunicação de dados DICOM entre o nível de exportação e o nível federado é realizada seguindo o perfil básico para conexão de transporte seguro TLS definido em [DM15]. A autenticação é realizada entre ambas as partes e os certificados ITU-T X.509 são obtidos do *Certification Authority* do Portal de Teleradiologia.

A figura 10 ilustra o modelo de execução para disposição de um IOD para o sistema federado. O modelo consiste basicamente em quatro etapas que estão presentes em um típico cenário em que um estudo é compartilhado com o sistema federado. A seguir será detalhada cada uma destas etapas:

1. Uma entidade de aplicação local transfere um *Composite* IOD para a entidade de aplicação de exportação através da classe de serviço *Storage*. A entidade de aplicação local, pode ser, por exemplo, uma estação de trabalho dedicada para selecionar IODs DICOM nos servidores de imagens locais para serem dispostos para a federação. Após decidir quais estudos serão compartilhados com a federação, o médico transfere o estudo através do elemento de serviço C-STORE SCU. O médico pode opcionalmente, caso o IOD não esteja armazenado localmente na estação de

[1] Não é necessário enviar os valores para o atributo *Referenced Series Sequence* desde que as informações sobre a série referenciada pelo estudo não são necessárias.

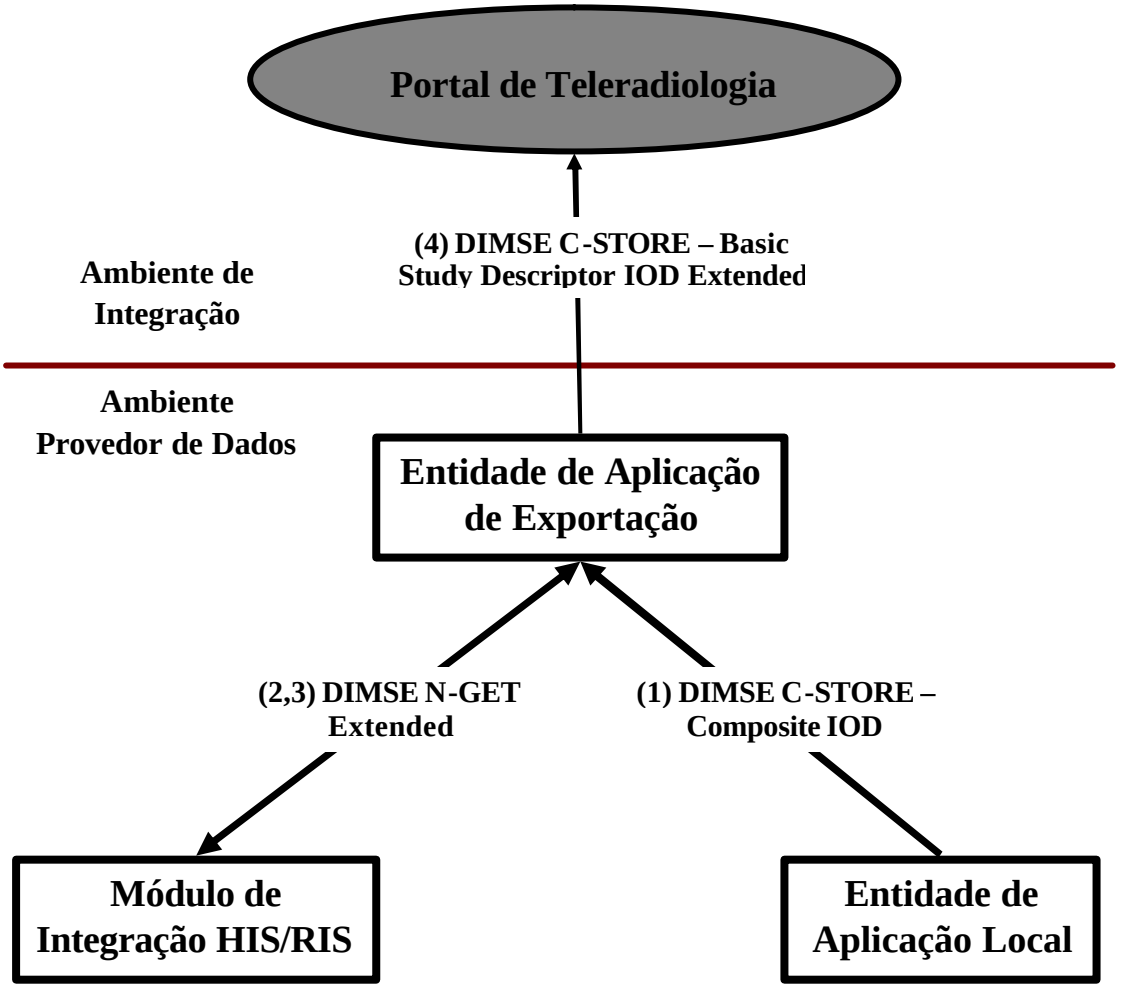

Figura 11: Modelo de execução de notificação para o sistema federado

trabalho, enviar um C-MOVE SCU para o banco local, com o título da entidade de aplicação de exportação no atributo *Move Destination* da primitiva de requisição. Nesta mesma estação de trabalho, o mesmo médico pode conectar-se via CORBA ao Portal de Teleradiologia para definir os atributos de controle de acesso para este estudo[1].

2. Após a finalização da associação C-STORE com a entidade de aplicação local, a entidade de aplicação de exportação verifica se o paciente associado ao IOD Composto se encontra armazenado localmente, verificando o atributo *Patient ID*.

[1] O desenvolvimento de um modelo de segurança e controle de acesso específico não faz do escopo deste trabalho, mas é previsto nos trabalhos futuros. Uma discussão sobre o tema é feita na seção 6.4

Neste ponto, assume-se que este campo é preenchido corretamente no ambiente PACS da organização. Caso contrário, com o campo com valor nulo, por exemplo, uma solução seria enviar a requisição para o módulo de integração HIS/RIS, e deixar que o sistema MPI do módulo encontre o paciente correto dentro do sistema HIS. Outra abordagem seria a entidade de aplicação rejeitar IODs com o campo *Patient ID* com valores inválidos, que de qualquer forma não estariam acordantes com o padrão DICOM. Caso seja determinado que o paciente já se encontra armazenado localmente, não é enviada a requisição para o módulo de integração HIS/RIS. Caso o IOD seja relacionado a um novo paciente, a primitiva de requisição DIMSE N-GET é enviada para o módulo de integração, com a entidade de aplicação assumindo o papel de SCU da classe SOP *Detached Patient Management*. No conjunto de dados enviados para consulta, estão os atributos adicionais à especificação DICOM, definidos durante a modelagem do esquema auxiliar (e.g CPF, RG, etc).

3. O módulo de integração HIS/RIS implementa um SCP para a classe SOP *Detached Patient Management*. Ao receber o elemento de serviço DIMSE N-GET contendo os atributos de paciente requisitados, é realizado o relacionamento entre os valores de atributos recebidos e o sistema HIS. Após buscar as informações relacionadas ao paciente no sistema HIS, o SCP envia a primitiva de resposta DIMSE N-GET com os atributos requisitados pelo SCU.

4. Após receber a primitiva de confirmação da máquina de estados DIMSE, a entidade de aplicação integra as informações do paciente presentes na primitiva com as informações de estudo presentes no IOD recebido na primeira etapa e cria um *Basic Study Descriptor IOD Extendido*. Este IOD é finalmente enviado para o Portal de Teleradiologia através do elemento de serviço C-STORE seguindo a semântica da classe SOP *Basic Study Content Notification*.

6.3.4.2 Instalação e configuração do nível de exportação

A quatro etapas descritas acima garantem um método consistente de notificação para os sistemas componentes dos estudos dispostos para o Portal de Teleradiologia. Todo o modelo de execução é realizado através de serviços de aplicação padronizados pelo padrão DICOM. Esta abordagem garante a compatibilidade com bancos de imagens e clientes DICOM existentes e provê um método prático para disposição seletiva de estudos para o sistema federado. Além disso, a etapa de negociação durante

o projeto do sistema federado é bastante simplificada, resumindo-se a instalação de uma entidade de aplicação DICOM com as funcionalidades de SCP para as classes de serviço *Query/Retrieve* e *Storage* e SCU para as classes SOP *Detached Patient Management* e *Basic Study Content Notification*. Quanto às duas classes SOP, apesar de ainda existirem poucas aplicações DICOM que suportem seus serviços, está previsto a implementação destas funcionalidades no CyclopsDICOMServer, servidor de imagens DICOM desenvolvido no mesmo âmbito deste trabalho.

O módulo de integração HIS/RIS é bastante dependente do domínio local. Em organizações onde os sistemas radiológicos e hospitalares são integrados ele consistirá em uma aplicação que consulte o sistema local e implemente a classe SOP *Detached Patient Management* com papel de SCP. Em organizações onde os sistemas não estejam integrados, será necessário o emprego de um sistema MPI. Uma situação bastante comum, é que os sistemas HIS e RIS compartilhem o mesmo identificador para pacientes, o que facilita bastante a tarefa de relacionar seus registros. Apesar de logicamente, representar um módulo independente da entidade de aplicação auxiliar, ambos subsistemas podem ser integrados em uma mesma aplicação.

Uma questão não abordada é a notificação quando um estudo for retirado da federação. O padrão DICOM especifica classes de serviço *Study Management* que provêm serviços de notificação de eventos relacionados a um estudo. Entretanto esta classe de serviço possui semântica ampla, relacionada ao controle de ciclo de vida de uma estudo. Uma solução mais simples é definir uma classe SOP especializada a partir de *Basic Study Content Notification* para notificar o sistema federado quando um estudo for retirado da entidade de aplicação de exportação. Neste caso, não é necessária interação com o módulo de integração HIS/RIS.

6.3.5 Nível federado

No nível federado está o esquema federado, que contém a integração dos múltiplos esquemas de exportação. A arquitetura do Portal de Teleradiologia apresenta duas características que o distinguem de uma arquitetura federada convencional. A primeira é total uniformização entre as entidades obtidas do esquema de exportação. O modelo de informação especificado pelo padrão DICOM permite manter a mesma descrição da estrutura dos dados entre o esquema federado e de exportação. A outra característica é a replicação de parte das instâncias dos bancos de imagens locais,

especificamente são mantidos os atributos relacionados à identificação do paciente e estudo. Como já discutido anteriormente, esta replicação é necessária para manter o sistema MPI e o mecanismo de controle de acesso no sistema federado.

O esquema federado também inclui informações sobre a distribuição dos dados, que são obtidas durante a etapa de integração. Estas informações constituem basicamente os parâmetros necessários para estabelecimento da associação DICOM com os servidores locais. Abaixo está um trecho do arquivo de configuração do CyclopsDICOMServer em sintaxe XML para uma entidade de aplicação cliente.

```
<dicom>
    <AETitle>Olho</AETitle>
    <label>DCMServer Client running on Olho</label>
    <host>olho.lisha.ufsc.br</host>
    <address>#[150 162 62 212]</address>
    <port>4007</port>
</dicom>
```

As instâncias dos bancos locais são atualizadas pelas notificações enviadas pelas entidades de aplicação de exportação de cada integrante da federação. Somente após a notificação ser recebida pelo sistema federado, é que um estudo local estará disponível para compartilhamento com os demais integrantes da federação. A notificação contém o IOD *Basic Study Content Notification* estendido com informações adicionais do paciente. Os atributos deste IOD irão ser armazenados no catálogo do sistema federado e utilizados durante a construção das consultas distribuídas.

Os identificadores do paciente serão utilizados pelo sistema MPI para geração de um identificador único para paciente. Este identificador irá referenciar todos os registros de atributos identificados como pertencentes a um mesmo paciente. A figura 11 apresenta um modelo entidade-relacionamento descrevendo a estrutura das informações mantidas no nível federado. As rotinas de relacionamento de registros do sistema MPI são efetuadas em dois momentos pelo sistema MPI: após novos estudos sobre um paciente serem enviados através de uma notificação e quando consultas são realizadas fornecendo apenas dados do pacientes. Estes rotinas possuem requisitos diferentes. A primeira rotina deve priorizar a precisão em detrimento ao tempo de processamento. O processamento pode ser realizado em uma máquina dedicada e/ou em *batch*. Interações

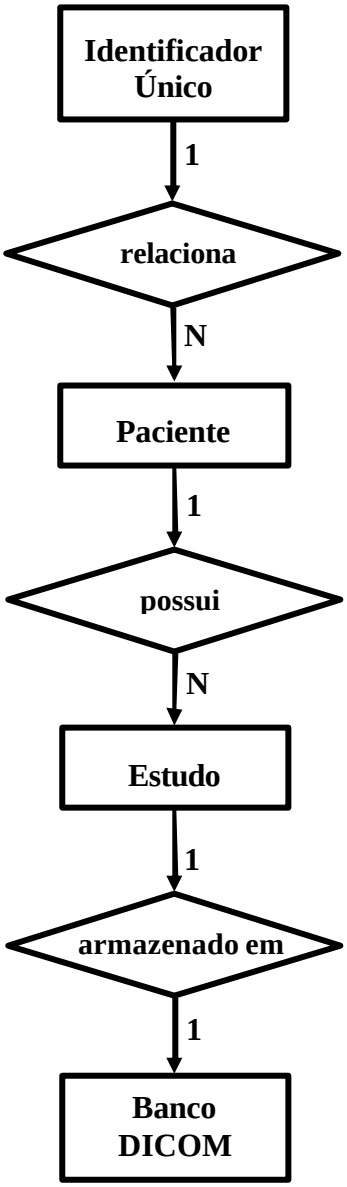

Figura 12: Modelo entidade-relacionamento do esquema federado

manuais são previstas para decisão final sobre o relacionamento de dois registros. Ao receber a notificação de um novo paciente de uma entidade de aplicação de exportação, o sistema poderá gerar um identificador único temporário para este paciente. Posteriormente este identificador poderá ser substituído pelo sistema MPI ou efetivado como definitivo. A rotina de relacionamento de registros executada quando um cliente consulta um paciente sem enviar seu identificador como parâmetro deve priorizar o

tempo de resposta em detrimento à precisão. O retorno de mais de um registro candidato ao cliente é aceitável.

A comunicação de dados no nível federado é realizada através de dois mecanismos distintos. Os clientes do sistema utilizam a arquitetura CORBA, através da invocação de operações para consulta, busca e inserção nos objetos CORBA do sistema federado. Para a associação com servidores DICOM, o sistema federado implementa as classes de serviço *Storage* e *Query/Retrive* como SCU e *Basic Study Content Notification* como SCP.

Dois processadores estão presentes no nível federado: um processador de transformação e um processador de construção. Ambos são implementados em um único objeto *servant* associado ao objeto CORBA invocado pelo cliente. A rotina que representa o processador de transformação no *servant* é responsável por receber as requisições de procedimento remoto CORBA e transformá-las nas primitivas de requisição para os servidores DICOM. Após as primitivas de confirmação serem recebidas da máquina de protocolos DIMSE, o processador de transformação realiza o *marshalling* do conjunto de dados do IOD e o retorna para o cliente. Para otimizar a comunicação entre os objetos CORBA do cliente e do nível federado, é utilizado o serviço de notificação de eventos. Neste caso o objeto CORBA do cliente associa-se ao canal de eventos localizado no sistema federado como consumidor e o objeto CORBA que encapsula o processador de construção associa-se como fornecedor de eventos. Desta forma, cada IOD recebido por uma associação DICOM gera um evento que é enviado ao cliente. A rotina do *servant* que implementa o processador de construção consulta os registros sobre os servidores DICOM componentes associados aos estudos associados e gera um conjunto de primitivas de requisições DIMSE. A manutenção das instâncias dos elementos de consulta do sistema, no caso os atributos de pacientes e estudos, elimina a necessidade de técnicas complexas para otimização de consultas e localização dos dados em sistemas federados [ZH01].

Como já referenciado anteriormente, o nível federado comporta uma CA para administração das chaves públicas usadas para identificação dos clientes e provedores de dados. Toda a comunicação de dados mediada pelo sistema federado, entre clientes e servidores DICOM é realizada utilizando o TLS como provedor de canal seguro.

6.3.6 Nível Externo

O nível externo define a visão do sistema federado para seus usuários. O esquema externo é o mesmo do esquema federado; as visões específicas do sistema para seus usuários são providas com base no mecanismo de controle de acesso, contido no TCB da federação.

Todos os usuários da federação precisam ter uma conta de acesso no sistema. O cliente utiliza o nível externo para acessar os servidores DICOM compartilhados e para executar operações de controle sobre os estudos sob sua responsabilidade. A discussão sobre as políticas de controle de acesso adequadas para o sistema federado é feita na seção 6.5.

O cliente precisa obter as referências para os objetos servidores e a definição das interfaces em IDL para invocar os serviços do Portal de Teleradiologia. As referências para os objetos do Portal de Teleradiologia podem ser obtidas através de um serviço de nomes CORBA ou em formato *stringfied*. Diversas maneiras podem ser utilizadas para divulgar as interfaces IDL e/ou a referência em formato *stringfied* dos serviços do sistema federado, como publicação WWW ou transferência por FTP. É prevista nos trabalhos futuros a disposição dos serviços da federação por *Web Services*.

6.4 Apresentação da Política de Segurança

O compartilhamento de informações clínicas entre diferentes instituições requer cuidados especiais com a segurança destas informações. A facilidade e agilidade para transmissão de dados clínicos proporcionada por um sistema federado enfatizam a necessidade de uma preocupação maior com questões como privacidade e ética médica. A garantia de um contexto seguro para a prática da Telemedicina em geral representa um grande desafio, pois reúne um domínio de informação extremamente sensitivo, no caso registros clínicos, e um ambiente com a disposição natural para o intercâmbio de informações como a Internet.

Questões ético-legais sobre a disposição de registros clínicos através da Internet e suas implicações sobre os requisitos de segurança para sistemas que provêm este tipo de serviço tem sido exaustivamente discutidas nos últimos anos. Entretanto, é seguro afirmar que os resultados destes debates ainda estão distantes de serem concludentes e que não existe o vislumbre de uma definição em curto prazo para estas questões, em especial no cenário brasileiro. Os atuais modelos de negócios e serviços, e o próprio

comportamento do profissional de saúde ainda estão se adaptando às novas condições políticas, sociais e legais decorrentes da prática da Telemedicina. Iniciativas de órgãos governamentais têm sido realizadas para regular a utilização da Internet para transmissão de registros clínicos. Entretanto, observa-se que o amplo domínio de aplicação ainda não está completamente especificado e que determinadas diretrizes terão que ser adaptadas para condições específicas que somente se revelarão em cenários concretos. No Brasil, generalidade apresentada na resolução do Conselho Federal de Medicina sobre as normas técnicas para a manutenção de prontuários eletrônicos [CFa02], [CFb02], corroboram a percepção de que a própria regulamentação do processo depende de um nível de entendimento do problema que apenas será alcançado com experiências práticas em situações reais.

Um fator determinante para a dificuldade em regulamentar a prática da Telemedicina é conciliar dois requisitos contrastantes: possibilitar o compartilhamento seletivo de informações médicas e ao mesmo evitar a exposição desnecessária destas informações. Um simples episódio médico pode envolver muitos profissionais, que devem ter acesso controlado aos dados clínicos, determinados por sua responsabilidade dentro da organização. Apenas o médico especialista responsável deve possuir autorização para acessar a seção de registros de seu paciente. Entretanto, em algumas situações, um médico especialista pode ter que compartilhar esta informação com outros especialistas fora dos limites da organização para fins de segunda opinião, por exemplo. Outro exemplo pode ser a necessidade de dispor exames médicos anônimos para fins de pesquisa. Todas estas possibilidades permitem uma complexa configuração para uma política de controle de acesso.

6.4.1 Política de segurança

As considerações acima ressaltam a necessidade de adotar um modelo de segurança flexível o suficiente para acomodar qualquer regulamentação ou recomendação aplicada ao sistema e ao mesmo tempo garantir a usabilidade em um ambiente de operação intrinsecamente colaborativo. Este requisito refere-se em particular a política de segurança e mecanismos de controle de acesso. Os demais objetivos de controle básicos definidos em [TCa85], *accountability* e garantia de segurança são relativamente independentes de um domínio de aplicação específico e estão previstas como requisitos no ambiente de operação do servidor de integração.

Entre as diversas políticas de segurança existentes, o *Role-based Access Control* (RBAC) é o modelo considerado como mais adequado para sistemas de atenção à saúde. RBAC provê uma solução elegante para o problema de gerenciar conjuntos de regras de controle de acesso em sistemas distribuídos. A noção básica do RBAC é o conceito de papéis, que é um mecanismo usado para categorizar usuários baseado em várias propriedades como cargo, função e responsabilidades. Permissões são associadas com papéis e usuários são atribuídos para papéis apropriados. Os usuários podem ser facilmente chaveados para outros papéis e permissões podem ser concedidas e revogadas se necessário. Restrições podem ser aplicadas para relações e funções definidas no modelo RBAC para estabelecer políticas organizacionais de alto nível. O padrão deixa em aberto a representação de usuários, papéis, autorizações e sessões, e a interpretação de autorizações, cabendo estas tarefas a implementações específicas. O RBAC provê um modelo de política de segurança que emprega primariamente controle de acesso obrigatório (MAC – *Mandatory Access Control*), com previsão de suporte a classificação de informação.

Além do escopo do controle acesso obrigatório, a delegação de privilégios é uma questão importante em um sistema médico para permitir que usuários possam assumir papéis temporários. A descentralização da administração da atribuição de usuários é importante em um controle de acesso baseado em papéis, permitindo uma granularidade mais fina e garantindo a escalabilidade do sistema. A idéia básica atrás da delegação é que um usuário pode conceder papéis para outros usuários para que estes desempenhem funções autorizadas por este usuário. Este tipo de controle de acesso é classificado como arbitrário (*Discretionary Access Control*) em [TCa85], e dentro do modelo RBAC é empregado como uma especialização do controle de acesso obrigatório principal.

A abordagem tradicional de RBAC apresenta dificuldades em capturar contextos relevantes para segurança que podem ter impacto na decisão de controle de acesso. Em um sistema de atenção à saúde, o nível de acesso específico e as permissões de um usuário devem ser determinados não apenas por seu papel na organização, mas também pelo contexto de segurança específico, como localização de origem e horário da requisição. Por exemplo, pode ser definido na política de segurança do sistema federado que um médico poderá ativar o papel de médico de plantão de emergência apenas quando estiver usando o sistema a partir da sala de emergência de um hospital. Outro

exemplo seria um médico exercendo o papel de *Residente*; neste caso o turno de trabalho determina janelas de tempo em que o acesso à identificação do paciente é permitido.

A aplicação de contextos de segurança ao modelo RBAC também se relaciona com a definição dinâmica das permissões relacionadas a um papel. Certas vezes as permissões para um determinado papel devem ser concedidas baseadas na atividade especifica que o usuário está desempenhando no momento. Por exemplo, suponha-se que a política de privacidade adotada no sistema federado conceda acesso a informações classificadas como altamente sensitivas apenas para o papel *Médico Principal de Tratamento* (MPT). Uma questão que surge é quais permissões devem ser atribuídas para o papel MPT, já que é inapropriado conceder permissões de todos os registros de pacientes. Um médico deve possui as permissões atribuídas para o MPT de um paciente apenas quando o paciente tenha designado este médico como seu MPT. Para tratar esta questão os atuais modelos de RBAC empregam a chamada atribuição de papéis dinâmica, onde as restrições de contexto são aplicadas durante o processo de ativação de um em uma sessão.

Diversos trabalhos tem sido realizados aplicando o modelo RBAC para sistemas distribuídos de atenção a saúde [ZG02], [MF01]. As implementações de referência para os padrões CORBA/OMG *Healthcare Security Service* e *Resource Access Decision Facility* utilizam o modelo RBAC. Em especial o modelo desenvolvido pelo NIST (*National Institute of Standards and Technology*) [FK92] vem se consolidando como o principal modelo para RBAC. A implementação e adaptação do NIST RBAC no Portal de Teleradiologia está prevista nos trabalhos futuros deste trabalho.

6.5 Protótipo desenvolvido

Um protótipo para a arquitetura federada do Portal de Teleradiologia foi desenvolvido durante o decurso deste trabalho. A principal finalidade deste protótipo foi servir como plataforma básica para testes, validações e refinamentos sobre a pesquisa realizada. Não houve o intuito de desenvolver um sistema federado completo e funcional nesta primeira etapa.

A protótipo foi desenvolvido utilizando a linguagem de programação *Smalltalk*. O ambiente de desenvolvimento utilizado foi o *Cincom VisualWorks*, versão 5i.1, não comercial. A implementação CORBA utilizada foi o *Distributed Smalltalk*, integrante

do ambiente de desenvolvimento. Os servidores DICOM utilizados nos testes foram o Offis [OF02] e o CyclopsDICOMServer [DP01]. O aplicativo utilizado como cliente foi o DicomEditor [DF99], onde adaptações foram feitas para adequá-lo aos propósitos deste trabalho.

A primeira etapa do desenvolvimento do protótipo priorizou a análise do desempenho da arquitetura CORBA como *middleware* entre os clientes e o servidor de integração, em particular na transmissão de grandes volumes de dados característicos das modalidades de imagens DICOM. Na abordagem, utilizada os objetos de informação DICOM passados por valor como parâmetros de retorno das chamadas de procedimento remotas. Esta é uma funcionalidade relativamente nova na especificação CORBA [SJ00], e portanto testes exaustivos foram necessários para avaliar a viabilidade desta solução. Os objetos de informação DICOM *Computer Tomography* e *Ultrasound* foram definidos em IDL e utilizados durante os testes.

O cliente de rede *DicomEditor* foi estendido para suportar a comunicação CORBA para busca de objetos de informação DICOM. Para otimizar a interação entre os objetos distribuídos e permitir a comunicação assíncrona foi utilizado o serviço de notificação de eventos CORBA. Desta forma, foram implementados os processadores de transformação e construção do nível federado representados por objetos distribuídos CORBA. Para comunicação com os servidores DICOM foram integradas classes utilizadas no DicomEditor para as classes de serviço *Storage* e *Query/Retrieve*, no papel de SCU. Os resultados dos testes foram bastante favoráveis quanto a utilização da tecnologia CORBA. A avaliação da performance para transmissão de grandes conjuntos de dados DICOM determinou uma performance bastante superior ao protocolo de comunicação DICOM.

O mecanismo de notificação da disposição de estudos para o sistema federado foi realizado através de uma aplicação CORBA, que permite selecionar os estudos na entidade de aplicação de exportação e enviar seus identificadores para o sistema federado. Esta aplicação foi integrada ao DicomEditor. O modelo apresentado na seção 6.3.4.1 está previstos nos trabalhos futuros.

Na segunda etapa de desenvolvimento do protótipo foi priorizado o modelo de segurança para o sistema federado, em especial os mecanismos para política de segurança e *accountability*. O mecanismo de controle de acesso foi baseado em um

modelo geral MAC com classificação seletiva dos estudos e especializações DAC, baseadas em ACL para delegação de privilégios. As permissões de controle sobre os estudos dispostos para a federação sãos feitas através de um aplicativo específico integrado ao DicomEditor. Com relação à *accountability*, a autenticação foi realizada através de senha, e foram implementadas rotinas configuráveis para auditoria, baseada em registros de ações. Foram ainda desenvolvidas rotinas de controle de exceções e garantia de robustez através dos mecanismos nativos CORBA e do ambiente *Visualworks.*

7. CONCLUSÕES

O projeto e desenvolvimento de uma federação de sistemas autônomos representa uma tarefa extremamente complexa, que exige esforços coordenados nas várias frentes do problema. O cenário de ampla heterogeneidade causado pela autonomia de projeto entre os integrantes da federação impossibilita soluções genéricas e boa parte das questões pertinentes tem que ser resolvidas caso a caso.

Neste trabalho foi desenvolvido um modelo para a integração de bancos de imagens médicas digitais em conformidade com o padrão DICOM baseado em uma abordagem federada. O sistema proposto provê uma visão única e transparente para os clientes enquanto mantém a autonomia e independência dos sistemas integrantes. No decurso do trabalho foi realizada uma avaliação do padrão DICOM em um sistema federado, deficiências foram identificadas e soluções foram propostas. Em especial, optou-se por utilizar a tecnologia para objetos distribuídos CORBA como *middeware* entre os clientes e o sistema federado. Entre as demais contribuições deste este trabalho estão a criação de um modelo para integração transparente de servidores DICOM pré-existentes à arquitetura federada; definição de um modelo de segurança condizente com os rígidos requisitos de um sistema de informação médico; pesquisa e avaliação de métodos de avaliação de equivalência semântica de registros. Um protótipo do modelo foi desenvolvido e usado como plataforma básica para testes, validações, refinamentos sobre a pesquisa realizada. É prevista uma futura extensão do Portal de Teleradiologia para prover acesso a prontuários eletrônicos de paciente, além de imagens radiológicas digitais.

REFERÊNCIAS BIBLIOGRÁFICAS

[AR00] ANDERSON R.; A security policy model for clinical information systems. University of Cambridge Computer Laboratory, Cambridge.

[BG92] BELL, D. GRIMSON J; Distributed Database Systems**.** Addison Wesley,1992.

[BH94] BRIGHT, M; HURSON, A.; PAKZAD, S. Automated Resolution of Semantic Heterogeneity in Multidatabases. ACM Transactions on Database Systems, Vol. 19, N° 2, June 1994, Pages 212-253.

[BS01] BELL, Gleen B.; SETHI, Anil; Matching Records in a National Medical Patient Index. Communications of the ACM, Association for Computing Machinery, September 2001, Vol. 44, Nº 44, pag. 83-88.

[CC00] CAMARGO, Kenneth R.. COELI, Cláudia M.; Reclink: aplicativo para o relacionamento de bases de dados, implementado o método *probabilistic record linkage*. Cad. Saúde Pública, Rio de Janeiro, 16(2):439-447, abr-jun, 2000.

[CC02] CHRISTEN P.; CHURCHES T. Febrl: Freely extensible biomedical record linkage, Joint Computer Science Technical Report Series – Disponível em http://cs.anu.edu.au/techreports/.

[CD01] COULOURIS, G; DOLLIMORE, J; KINDBERG, T. Distributed Systems: Concepts and Design - Terceira edição. Addison-Wesley, 2001.

[CFa02] Resolução do Conselho Federal de Medicina. nº 1.638/2002.

[CFb02] Resolução do Conselho Federal de Medicina. nº 1.639/2002.

[CT25] Comitié Européen de Normalisation Technical Committee 251, http://www.centc251.org. Último acesso em 25.02.2002.

[DF99] KRECHEL, Dirk; FABER, Kerstin; WANGENHEIM, Aldo Von; et al. Object-Oriented Implementation of a DICOM Client in Smalltalk. CBMS99 - 12TH IEEE Symposium On computer-based Medical Systems, 1999, Stamford. IEEE Computer Society Press, v.1. p.12- 17.

[DM01] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 1.

[DM03] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 3.

[DM04] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 4.

[DM05] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 5.

[DM07] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 7.

[DM08] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 8.

[DM015] *NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION*. *Digital Imaging and Communications in Medicine* (DICOM);. Part 15.

[DS00] http://www.datasus.gov.br/. Último acesso em 28.02.2002.

[DP01] DELLANI, Paulo Roberto; Desenvolvimento de um servidor de imagens médicas digitais no padrão DICOM; Dissertação de Mestrado - UFSC - CPGCC– 2001.

[EN94] ELMASRI, R. NAVATHE, S. Fundamentals of Database Systems. 2 ediçao – Addison-Wesley, 1998.

[FH93] TUIJNMAN, Frank; AFSARMANESH Hamideh. Distributed Objects in a Federation of Autonomous Cooperation Agents. 0-8186-3135-X/93, 1993 IEEE.

[FK92] FERRAIOLO, D.; KUHN, R.; Role-Based Access Control. Proceedings of 15th National Computer Security Conference, 1992.

[FS69] FELLEGI, Ivan; SUNTER, Alan; A theory for Record Linkage**.** Journal of the American Statistical Association, American Statistical Association, December 1969, Vol. 64, Nº 64, Nº 328, pp. 1183-1210.

[GR03] GU L.; BAXTER R.; Vickers D., et al. Record Linkage: Current Practice and Future Directions. Technical Report 03/83, April 2003, CSIRO Mathematical and Information Sciences, Canberra 2601, Australia

[GC03] GOMATAM, S, CARTER, R., ARIET, M., et. al; An Empirical Compassion of Record Linkage Procedures, 2003. Obtido diretamente com o autor.

[GC04] GOMATAM, S, CARTER, R.; A computerized Stepwise Deterministic Strategy for Record Linkage, 5 de julho 2004. Obtido diretamente com o autor.

[MF01] MOTTA, G.; FURUIE, S.; Um modelo de autorização e controle de acesso para o prontuário eletrônico de pacientes em ambientes abertos e distribuídos. Revista Brasileira de Engenharia Biomédica, v.17, n.3, p. 141-150, set/dez 2001.

[HI00] Healthcare Information Systems Architecture. http://www.hisa.org. Último acesso em 25.02.2002.

[HL07] Health Level Seven, Inc - http://www.hl7.org/ [site oficial]. Último acesso em 27.02.2002.

[HM85] HEIMBIGNER, H.; MCLEOD D.: A federated architecture for Information Management. ACM Transactions on Office Information Systems, Vol 3, Nº3, Julho de 1985, pp. 253-278.

[HP96] Heath Insurance Portability and Accountability Act - Standards for privacy of individually Identifiable Health Information. Office of the Assistant Secretary for Planning and Evaluation, Department of Health and Human Services. Obtido em http://aspe.hhs.gov/admnsimp/.

[HV99] HENNING, S; VINOSKI, S. Advanced CORBA® Programming with C++. Primeira Edição, 1999. Addison Wesley.

[LM84] LYNGAEK, Peter; McLEOD Dennis. Object Management in Distributed Information Systems. ACM Transactions on Office Information Systems, Vol.2, No. 2, April 1984, Pages 96-122.

[LM91] LI, Qing; McLEOD, Dennis. An Object-Oriented Approach to Federated Databases. TH0372-3, 1991 IEEE.

[LM90] LITWIN, W., MARK L., ROUSSOPOULOS, N.: Interoperability of Multiple Autonomous Databases. ACM Computing Surveys, Vol. 22, Nº. 3, Setembro 1990.

[JM85] JARO, M. A.; Advances in Record Linkage Methodology as Applied to atching the 1985 Census of Tampa, Florida. Journal of the American Statistical Society, 84(406):414–20, 1989.

[LS02] LANG, U.; SCHREINER R. Developing Secure Distributed Systems with CORBA. Primeira Edição, Artech House, London, 2002.

[MH03] MACHADO, C.; HILL K.; Probabilistic Record Linkage and an Automated Procedure to Minimize the Undecided-Matched Pair Problem. CEDEPLAR/FACE - UFMG, Belo Horizonte 2003.

[MS02] MARTIN, K. Michael; SHEVCHENKO, Ivan P.; REED-FOURQUET, L., Lori. A Strategy for Statistical Master Person Index Linking. The Connecticut Healthcare Research and Education Foundation, Wallingfor, CT.

[NK59] NEWCOMBE, H.; KENNEDY, J.; AXFORD, S. et al. Automatic Linkage of Vital Records. Science, American Association for the Advancement of Science, Vol.130, Nº 3381, October 16, 1959, pp. 954-959.

[NC92] A guide to understanding security modeling in trusted systems. National Computer Security Center – Technical guidelines program, 1992.

[OF02] Instituto para o Desenvolvimento de Sistemas de Informática de Oldenburger – Desenvolvedores do Conjunto de Ferramentas DICOM DCTK - http://www.offis.uni-oldenburg.de/indexe.htm. Ultimo acesso em 03.03.2002.

[OH03] Object Management Group, HealthCare DTF Task Force, http://www.omg.org. Último acesso em 13.11.2003.

[OH96] ORFALI, R.; HARKEY D.; EDWARDS, J.; The Essential Distributed Objects Suvival Guide; John Wiley and Sons, New York, 1996.

[OS01] Object Management Group, CORBAservices: Common Object Services Specifications, http://www.omg.org, 1991. Último acesso em 17.09.2002.

[OV91]ÖZSU, M.; VALDURIEZ Patrick; Principles of Distributed Database Systems, Second Edition. Prentice Hall, 1999.

[PW88] PORTER, E. H.; WINKLER, W. E.; Approximate string comparison and its effect on an advanced record linkage system. In Proc. of an International Workshop and Exposition - Record Linkage Techniques, Arlington, VA, USA, 1997.

[PZ84] POLLOCK, J.; ZAMORA, A. Automatic Spelling Correction in Scientific and Scholarly Text, *Communications of the ACM*, **2**7, 358-368, 1984.

[RE96] RADEKE, Elke; Extending ODMG for Federated Database Systems. 0-8186-7662-0/96, 1996 IEEE.

[RK00] ROANTREE, M.; KEANE, J.; MURPHY, J.. A Three-Layer Model for Schema Management in Federated Databases. School of Computer Applications, Dublin City University, Dublin, Ireland.

[RW01] ROOS LL, WAJDA A. Record Linkage Strategies. Methods of Information in Medicine, 1991; 30:117-123.

[SK99] SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S.; Sistema de banco de dados. 3ª. Edição; MARKON Books Ltda, 1999.

[SJ00] SIEGEL Jon; CORBA 3 – Fundamentals and Programming, Segunda edição, John Wiley and Sons, 2000.

[SR00] STEVENS, W. Richard; TCP/IP Illustrated The protocols – Volume 1. First Edition. Addison-Wesley, 2000.

[TCa85] Department of Defense Trusted Computer System Evaluation Criteria. Department of Defense Standard - DoD 5200.28-STD, 1985.

[TCb85] Computer Security Requirements – Guidance for applying the Department of Defense Trusted Computer System Evaluation Criteria in specific environments. Department of Defense Standard – Technical Rationale Behind CSC-STD-003-85, 1985.

[WM89] WANG, Y.; MADNICK S. The inter-database instance identification problem in integrating autonomous systems. In Proceeding of the 5th International Conference on Data Engineering. IEEE Computer Society, Washington, D.C., 46-55.

[WW88] WINKLER, W. Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage, *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 667-671, 1988.

[WX01] WU, Xuequn. A CORBA-Based Architecture for Integrating Distributed and Heterogeneous Databases. Deutsche Telekom AG, Research Center, Darmstadt, Germany.

[ZG02] ZHANG, L.; AHN, G.; CHU, B.; A role-based delegation framework for healthcare information systems. ACM SACMAT'02, June 3-4, 2002, Monterey, Califórnia, USA.

[ZH00] ZHANG, J. Classifying approaches to semantic heterogeneity in multidatabase systems. Queen's University, Kingston, Ont. Computing and Information Sciences.

[ZH01] ZHU, Q.; Query optimization in multidatabase systems. Department of Computer Science, University of Waterloo, Ontario, Canadá.

ANEXO 1

Artigo publicado em The 15th International Conference on Computer Based Medical System - CBMS. Maribor,2002.

CyclopsDistMedDB. - A Transparent Gateway for Distributed Medical Data Access in DICOM Format

Leonardo Andrade Ribeiro
The Cyclops Project,
Telemedicine Laborator,
University Hospital (UFSC)
Florianópolis – SC – Brazil
lar@inf.ufsc.br

Paulo Roberto Dellani
The Cyclops Project,
Telemedicine Laboratory,
University Hospita (UFSC)
Florianópolis – SC – Brazil
dellani@inf.ufsc.br

Aldo von Wangenheim
The Cyclops Project
Computer Sciences
Department (UFSC)
Florianópolis – SC – Brazil
awangenh@inf.ufsc.br

Michael M.Richter
The Cyclops Project,
Universität Kaiserslautern,
Germany
richter@informatik.uni-kl.de

Kerstin Maximini
The Cyclops Project,
Universität Kaiserslautern,
Germany
k_maximi@cs.uni-hildesheim.de

Eros Communello
The Cyclops Project,
Universität Kaiserslautern,
Germany
eros@informatik.uni-kl.de

Abstract

The image diagnosis area is the most propense medical field to Telemedicine, because it does not obligate a direct contact of the patient with the responsible radiologist during the building of the report. The persistent lack of specialists on places distant from urban centers makes the Telemedicine an important tool for improvement of healthcare services. In this work we present a framework, called CyclopsDistMedDB, for the integration of distributed DICOM medical record databases over wide areas. The present system has a central module that is responsible for receiving the clients requests about patient data (images, waveforms), performing the querying and retrieval of images, patient records etc. from the specific DICOM databases containing the data requested and delivering them to the clients. The data communication protocols adopted are DICOM, for retrieval of objects directly from DICOM data servers and CORBA (Common Object Request Broker) for the delivery of DICOM data to client applications.

1. Introduction

The need of storing and making available digital medical examination data containing images and biological signals, like computed tomography, magnetic resonance and electrocardiography, is increasingly present in the hospital and clinics environments. The “filmless radiology” represents a solution to improve accessibility, quality and quantity and, at the same time constrain the healthcare costs. In this scenario, the DICOM standard has become an effective international standard and the most important protocol used in Radiological Information Systems (RIS).

The Telemedicine is one of the most important supporting technologies to bring the healthcare services closer to citizens. In order to achieve this, the Telemedicine services need to become more global and ubiquitous; in particular a standardized communication is necessary. Isolated and self contained systems must move toward an integration and sharing of information throughout hospitals, increasing their interoperability and providing more general services.

The transparent interchange of medical image and signal data through the Internet by medical organizations is essential in order to eliminate completely the necessity of the physician's presence on the site where the images are produced. A deeper integration of image and signal databases between different, distant hospitals is necessary for a better follow-up of the patient's health history, enabling a drastic reduction of duplicate examinations in emergency and outpatient situations and also supporting the providing of second opinions and even other not so obvious benefits like better scheduling of work between clinics who operate on different places.

The main goal of this work is to provide a framework that allows hospitals of any size which have DICOM image and signal databases running in their Intranet environments to share studies among them under secure and functional conditions.

2. Problem Description

The DICOM standard has become a *de facto* world standard, but its use in an Internet wide context is still very limited. The transfer and storage of radiological images between peer entities is done mostly in Intranet environments. In order to build a general and transparent mechanism for sharing images between different clinics and hospitals, it is necessary to deal with two main issues: a strong support for secure access control to individual studies and the ability of uniquely identify patients among different domains, which are both not yet supported by the DICOM standard.

Any system working with sensitive information has to guarantee the security of transmission and a rigid access control over its data . On part 15 of the DICOM standard [6], there is specified the support for connections over Secure Socket Layer (SSL), but access control is still defined on database level, not patient or study levels. Currently, there is a proposal to the DICOM standard based on encryption within the application layer, which consists in new attributes for DICOM objects where the sensitive information about a person will be ciphered. This approach requires less time expenditure than the transmission over SSL/TLS.

On the other hand, the access control specified in the DICOM standard is entirely delegated to the Application Entities (AEs). It is assumed that the AEs involved in a DICOM data interchange are implementing appropriate security policies, as access control, audit trails and mechanisms to identify users and their rights to access information. Unfortunately, most of the DICOM database server implementations only follow the security levels specified in the standard, with access control based on the context of applications and hosts with no user-based authentication. Another important point to be taken into account is the subject of distribution of access rights. For a more strict access control, the access right distribution must be based on studies and not on database level. In spite of the fact that the process of managing authorization decisions on fine grained resources is an expensive action, a rigid control over the information interchanged is a critical requirement in regional healthcare network-compatible applications.

The unique patient identification among database objects located at different systems is still another issue that must be addressed to enable the sharing of those images and signal data between different locations in a regional healthcare network. The patient's moving or a change of healthcare provider will lead to a same person to have studies stored at different places. Historically, health care providers dealt with this issue by creating a Master Patient Index (MPI) that used a limited set of demographic data (for example, name, gender, date of birth, etc) to help retrieve the disparate elements of a patient's records. In order to gain access to information about a person stored in DICOM databases from heterogeneous systems, it will be necessary to create a centralized MPI service able to identify correlated data and to provide links between them.

3. Distributed DICOM Database Gateway

This work presents a model for a transparent access gateway for distributed medical data in DICOM format called *CyclopsDistMedDB*. This solution is based on a centralized approach, with a module managing transactions between clients and the DICOM databases. This module is responsible for receiving the client requests about patient data (images, waveforms), performing the querying and retrieval of these objects from the specific DICOM databases containing the data requested and delivering it to the clients. The data communication protocols adopted are DICOM, for retrieval of objects directly from DICOM data servers and CORBA (Common Object Request Broker) for the delivery of DICOM data to client applications. Both communications are performed over a secure channel using SSL. The choice for CORBA for the communication with the clients provides more flexibility to the system and allows the use of the system together with commercial Internet providers. The multi-language feature of CORBA allows easilly applications written in different languages to use the services of our approach . Figure 1 describes the CyclopsDistMedDB communication model.

The use of a centralized intermediary instance to interchange image and waveform objects among the servers and clients provides facilities for the easier configuration of both entities. The client applications just need to keep CORBA data like IDL interfaces and object references for request services. On the image servers side, it is necessary only to enable the connection with CyclopsDistMedDB as a DICOM Aplication Entity. The configuration of the whole system is performed on the central module, and the addition or removal of one participant does not pose the need of the reconfiguration of all the others. Additionally, this approach allows dial-up and DSL clients to gain access to images on DICOM databases, which is difficult in traditional DICOM configurations because these clients have temporary IP addresses, and DICOM servers request pre-knowledge about the IP address of their clients and callback enabling. A table of metadata in the gateway module contains information about patients, their studies and their places of origin and storage. This information is kept updated by the nodes containg the DICOM servers through a daemon which retrieves local information from

the DICOM database through C-FIND message service [5] and sends information about new studies to the central gateway.

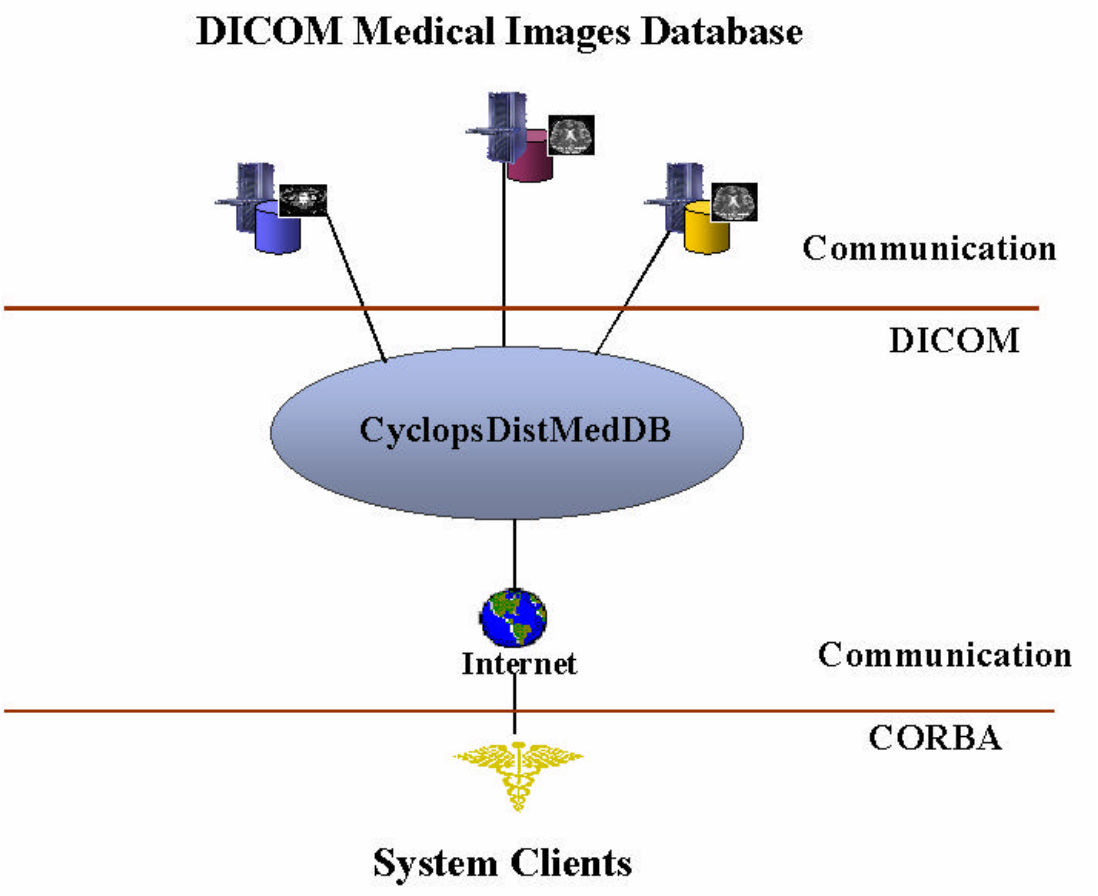

Figure 1. Communication model among CyclopsDistMedDB clients and DICOM servers

To be able to identify some amount of information from different places as concerned to a same patient, the system employs a MPI which is executed each time new data from the DICOM database demons arrives, correlating information related to the same patient but originated from different places. An implementation of a MPI system based in the probabilistic approach of [3] will deployed on second stage of the development of CyclopsDistMedDB.

4. Security, Ethics and Access Control

Medical ethics and sensitive information protection against unauthorized access are a key issue in this field and addressed in this work through connection security policies and a special access control strategy.

4.1. Security for Data Interchange

The support of secure channels during the data transmission is provided through TLS, both on the CORBA and DICOM side, providing for encrypted connections. In spite of the loss of generality because there are many DICOM databases which do not employ TLS, this requirement is essential to use DICOM network services on Internet/WAN level. A DICOM server that is not conformant with DICOM Secure Connection Transport Profile should not be used on public networks.

During data transmission from DICOM servers to CyclopsDistMedDB, the TLS communication employs host authentication of both sides. CyclopsDistMedDB allows DICOM connections only from hosts registered as DICOM data suppliers and with known public keys.

4.2. Access Control Strategies

The present work does not intend to provide security for each participant individually. It is assumed that each Application Entity insures that its own local environment is secure before even attempting secure communications with the system. Thus, access policies of the system are extending the Intranet secure context, that each clinic or hospital must have implemented, toward the Extranet environment.

To enable a fine-grained access control at study level, the system provides identification of all the interested parties involved in patient care, providing user-based and role-based authorization controls. The main objective is provide an essential guarantee for privacy of sensitive patient data, keeping the doctor-patient trust relationship during Telemedicine practice.

Implementing the basic precepts of the medical ethics in the Telemedicine field, is although not an easy task. The non-existence of international rules and mediator associations able to provide well-defined ethical rules, makes the building of a specific access policy hard. One of the common sense rules for the practical Telemedicine is the one that the patient must have pre-knowledge of any transmission of his/her studies. Emergency conditions, with eminent danger of life should constitute an exception, but without exempting the physician of any responsibility for the handling and the correct use of the sensitive information of the patient. Another requirement, recently highlighted by the rules defined by U.S. Health Insurance Portability and Accountability

Act (HIPAA), is that the patient must have extensive access to any kind of medical studies about him [4].

In order to satisfy the medical ethical issues without loosing the functionality, we propose that a system that distributs patients records over a public network must attend at least the following features:

- The access policies are applied at the study level, which means that each study that belongs to a patient can have its own access policy;
- Studies can have multiple security levels. A radiologist or responsible technician can determine that a study can be accessed only by users with explicit permission to do it (this is the appropriate default behavior), or by users with special rights (i.e., the one responsible for the emergency room of a hospital) or even though allow access to any system user.
- The system provides interfaces for the patient who has studies stored on it, to access its own data. The patient can view and control who has access, who is accessing its information and avoid inappropriate use of that information;
- Users which not are expressively allowed to access some information not must have any access or view to this information;
- A extensive log of the operations must be employed for audit purposes;

Information about other studies in the distributed database is completely filtered out, providing for each user different views about the data managed by system. It is possible for a physician who is participating directly on the treatment to grant access to a study for another physician for second opinion purposes. Each operation of granting permissions is recorded in the system for audit purposes.

In order to deploy the access control stated above, we used the basic lifecycle service provide by CORBA specifications. The client application holds a stringified reference to a CORBA Factory object on CyclopsDistMedDB. First, the only usable interface of this object is the one that receives the user login-password data. After a successfull validation and identification on the system, an instance of the class which holds the session context for this user is created. We use extensively the object-oriented polymorphism feature in order to deploy different user roles. The system objects wrapping DICOM data maintain information about the users who are allowed to access

them and which operations could be performed. The object encapsulation enables a good level of security.

5. Conclusions

We developed a model for a distributed DICOM-compliant medical database accessible through a central gateway, allowing the creation of regional healthcare networks providing image and signal data. This model solves various security and accessibility problems related to the DICOM Standard, without changing the standard itself, thus allowing the integration of existing PACS systems into a network of connected hospitals of any size.

This work aimed at the development of a technology of a distributed database of images and biological signals, able to be used on high speed networks as a central gateway for access to DICOM databases located in different and possible geographically distant places, providing the security and transparency requested by this kind of application. A first prototype implementation of this model has already been developed and is being tested in the scope of the German-Brazilian Cyclops Project.

References

[1] Dellani, Paulo Roberto; Desenvolvimento de um servidor de imagens médicas digitais no padrão DICOM; Dissertação de Mestrado - UFSC - CPGCC– 2001.

[2] Elmasri, R. Navathe, S. 1994 Fundamentals of Database Systems. Second edition – Addison-Wesley, 1998.

[3] Fellegi, Ivan; Sunter, Alan; A theory for Record Linkage. Journal of the American Statistical Association, American Statistical Association, December 1969, Vol. 64, Nº 64, Nº 328, pp. 1183-1210

[4] Health Insurance Portability and Accountability Act of 1996 http://www.hcfa.gov/hipaa/hipaahm.htm. Last Access in November, 10th 2001.

[5] National Electrical Manufacturers Association. Digital Imaging and Communications in Medicine (DICOM) - Part 15: Message Exchanges. Virginia, 2001. On-line available at: <ftp://medical.nema.org/medical/dicom/2001/draft/01_15dr.pdf>.

[6] National Electrical Manufacturers Association. Digital Imaging and Communications in Medicine (DICOM) - Part 15: Security Profiles. Virginia, 2001. On-line available at: <ftp://medical.nema.org/medical/dicom/2001/draft/01_15dr.pdf>.

[7] Object Management Group – http://www.omg.org. Last Access in March, 10th 2002.

[8] Request for Comments: 2437 PKCS #1: RSA Cryptography Specifications Version 2.0 On-line available at: http://www.rsasecurity.com

[9] Siegel Jon; CORBA 3 – Fundamentals and Programming, Second edition, John Wiley and Sons, 2000.

[10] Stallings, Willian; Cryptography and Network Security. Prentice Hall, 1998.

[11] Stevens, W. Richard; TCP/IP Illustrated The protocols – Volume 1 – First edition. Addison-Wesley, 2000.