

Construção de um Serviço de Identificação de Pacientes

VIVIAN RODRIGUES FIALES^{1,2}
ORIENTADORES: FABIANE BIZINELLA NARDON¹
SÉRGIO SHIGUEMI FURUIE¹

¹ Instituto do Coração do Hospital das Clínicas da Faculdade de Medicina da USP (InCor – FMUSP/HC)
Unidade de Pesquisa e Desenvolvimento do Serviço de Informática
Av. Dr. Enéas de Carvalho Aguiar, 44, São Paulo, Brasil
{vivian.fiales, fabiane.nardon, furuie}@incor.usp.br

² Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática
Rua da Consolação, 896, São Paulo, Brasil

Resumo: A identificação de pacientes, em uma instituição de saúde, é considerado um problema cuja solução é vital para recuperação dos históricos completos dos pacientes. Porém, solucionar tal problema não é uma tarefa trivial, uma vez que é muito comum haver uma grande heterogeneidade nos diferentes sistemas utilizados pelas diversas instituições de saúde onde o paciente recebeu atendimento ao longo de sua vida. Para que um sistema de identificação de pacientes possa executar satisfatoriamente esta tarefa, é necessária a implementação de métodos de busca que permitam encontrar pessoas dentro de um cenário de informações incompletas e que possui ambientes heterogêneos. Este artigo apresenta a solução escolhida pelo Instituto do Coração (InCor) para resolver tal problema.

Palavras Chave: CORBA, PIDS, informática em saúde, sistemas distribuídos.

1 Introdução

O Instituto do Coração (InCor) é um dos seis institutos independentes que formam o Hospital das Clínicas (HC) da Faculdade de Medicina da Universidade de São Paulo (FMUSP), maior hospital da América Latina, com 2.200 leitos, atendendo uma média de 3.500 pacientes ambulatoriais por dia. Apesar de estarem alocados no mesmo espaço físico e haver uma grande interação entre os institutos que formam o HC, cada um possui seus próprios sistemas de informação, projetados para atender suas necessidades específicas.

O InCor e o Instituto de Radiologia (InRad) vêm desenvolvendo há algum tempo um projeto de pesquisa, chamado PACS-HC, que tem como objetivo elaborar um modelo de ambiente distribuído para a transmissão, arquivamento, processamento e visualização de imagens médicas, integrando-as ao sistema de informações do HC, em geral, e do InCor, em particular. Este projeto tem como objetivo principal a recuperação do histórico completo dos pacientes. Porém, para que tal objetivo seja atingido, é necessária a implementação de um sistema de identificação de pacientes, uma vez que um sistema de informação capaz de recuperar o histórico completo de uma pessoa pressupõe a existência de um sistema capaz de identificar os pacientes dentro da instituição.

Para garantir a interoperabilidade entre os sistemas heterogêneos que compõem o projeto PACS-HC, é necessário utilizar mecanismos que permitam que sistemas construídos em diferentes linguagens de programação, sistemas operacionais e plataformas de hardware possam trocar informações entre si. Tal grau de interoperabilidade pode ser

atingido através da utilização de padrões que forneçam a infra-estrutura necessária para prover o nível adequado de integração. Dentre os padrões adotados neste projeto, estão:

- UML (*Unified Modeling Language*), linguagem orientada a objetos para modelagem de sistemas, padronizada pela OMG (*Object Management Group*), permitindo a utilização de técnicas de engenharia de software;
- *Middleware* CORBA (*Common Object Request Broker Architecture*), padrão aberto para desenvolvimento de sistemas que utilizem objetos distribuídos integrados em plataformas diferentes;
- Especificações aprovadas pelo comitê CORBAMED [OMG, 1999], que é a força tarefa da OMG, responsável pela padronização de serviços de software na área da Saúde.

A necessidade da construção de serviços independentes de plataforma, característica importante em um contexto que prevê a utilização em ambientes heterogêneos, levou à adoção da tecnologia Java®, sendo utilizadas várias de suas APIs (*Application Programming Interface*), tais como JDBC® (*Java® Database Connectivity*), para acesso a bases de dados; Swing®, para construção da interface com o usuário; Javadoc®, para geração automática de documentação.

Este artigo apresenta o serviço de identificação de pacientes, construído pelo InCor para o Projeto PACS-HC. Tal serviço foi baseado na especificação do Serviço de Identificação de Pacientes (*Person Identification Service – PIDS*) [OMG, 1998], proposto em 1998 pelo Comitê CORBAMED. Esta especificação define interfaces padronizadas, permitindo que diferentes sistemas, criados em plataformas de hardware e software diferentes, troquem informações entre si de forma transparente.

Após a apresentação do serviço de identificação de pacientes, apresenta-se a gerência de uma federação construída para o Projeto PACS-HC. Tal gerência permite que diversos serviços PIDS, cada um atuando sobre um domínio de identificadores específico, comuniquem-se entre si, garantindo a identificação de pacientes de forma descentralizada e possibilitando assim a identificação de um paciente onde quer que ele tenha recebido atendimento.

O texto está organizado da seguinte forma: a seção 2 detalha a especificação do PIDS; a seção 3 descreve a federação implementada para o Serviço de Identificação de Pacientes; a seção 4 apresenta os detalhes de implementação do PIDS construído; e a seção 5 apresenta uma conclusão sobre a implementação do serviço no InCor.

2 A Especificação do PIDS

O PIDS especifica um serviço cujo objetivo é a identificação de um mesmo paciente em diferentes sistemas de informação. Este serviço é capaz de encontrar identificadores de pacientes a partir de um conjunto de características conhecidas, atribuir identificadores a pacientes, recuperar informações de identificação de pacientes, avisar outros sistemas quando uma modificação numa característica de um paciente ocorrer, entre outras funcionalidades.

Por definir interfaces padronizadas, necessárias para realização de tarefas normalmente encontradas em um serviço de identificação de pessoas, tal serviço permite que diferentes instituições de saúde, utilizando plataformas potencialmente diferentes, possam trocar informações sobre pacientes, sem necessidade da criação de soluções proprietárias.

O PIDS vem sendo utilizado em diversos hospitais no mundo, como forma de atingir a interoperabilidade entre seus sistemas. Dentre os projetos mais ambiciosos que utilizam essa especificação, está o G-CPR (*Government Computer-Based Patient Record*) [GCPR, 2000], projeto do governo americano que tem como objetivo criar um ambiente onde informações clínicas dos pacientes possam ser compartilhadas dentro do serviço clínico, integrando os sistemas de alguns órgãos governamentais como o *Department of Defense* (DoD), o *Indian Health Service of Department of Health and Human Services* e o *Department of Veterans Affairs* (VA).

Outro projeto que utiliza a especificação do PIDS é o TeleMed® (ou OpenEMed®) [LANL, 2000], um sistema de informação em saúde, aberto e colaborativo, desenvolvido pelo *Los Alamos National Laboratory* (LANL). Mais especificamente, é um sistema de prontuário do paciente que suporta imagem, áudio e dados gráficos, utilizando as tecnologias Java® e CORBA. Este sistema está sendo utilizado no *National Jewish Center for Immunology and Respiratory Medicine*, em Denver, Colorado.

A especificação do PIDS define as seguintes interfaces: *IdentificationComponent*, interface principal cujo objetivo é oferecer acesso aos demais componentes; *IdentifyPerson*, responsável por encontrar pacientes a partir de informações de identificação; *IdMgr*, gerencia os identificadores dos pacientes, possibilitando a criação dos mesmos; *SequentialAccess*, fornece acesso seqüencial às informações de todos pacientes conhecidos pelo serviço; *ProfileAccess*, utilizada para consulta e alteração do perfil de um determinado paciente; *IdentityAccess*, gerencia as características de um ou mesmo vários pacientes, de uma única vez; *CorrelationMgr*, a partir de uma dada instituição, permite a consulta de pacientes que foram atendidos em outras instituições. O conjunto de interfaces definidas pelo PIDS estão representadas na figura 1.

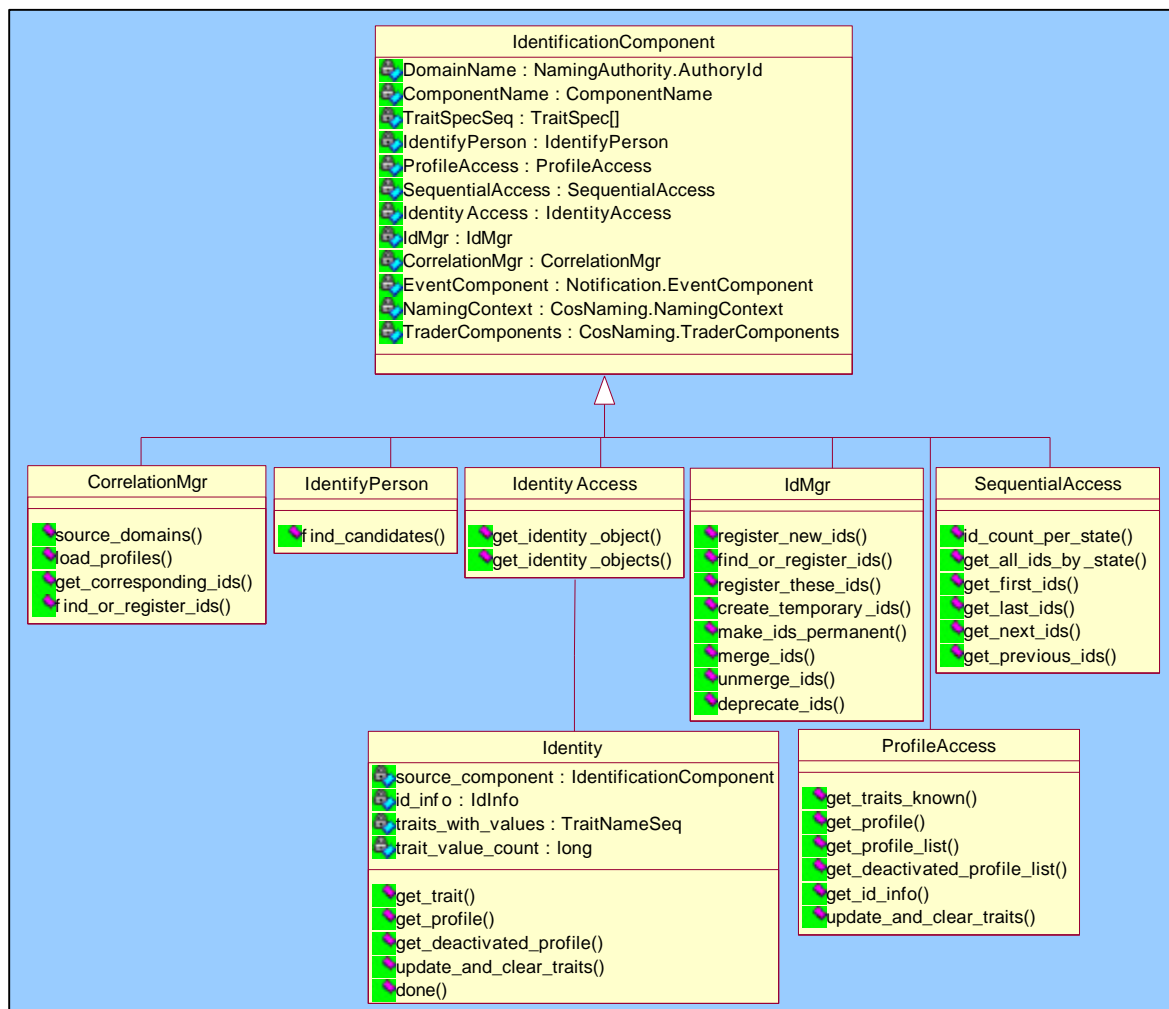


Figura 1: Interfaces do PIDS

A escolha de um serviço de identificação de pacientes a partir de uma especificação padronizada oferece várias vantagens. A primeira delas é o tempo reduzido de projeto, uma vez que o esforço de especificação já foi realizado quando da criação do padrão. Outra grande vantagem é a possibilidade da integração do serviço com sistemas de outras instituições que utilizem o mesmo padrão. Dessa forma, é possível a troca de informações entre ambientes heterogêneos com quase nenhum esforço adicional.

3 A Construção de uma Federação de Serviços de Identificação de Pacientes

A necessidade da construção de um sistema onde os pacientes atendidos em uma das instituições participantes do projeto pudessem ser encontrados nas demais instituições, sem a utilização de um sistema centralizado, levou à construção de uma estrutura de federação.

Numa federação, cada serviço atua sobre um domínio específico. Por exemplo, o PIDS do InCor atua sobre o domínio do conjunto de pacientes cadastrados em sua base de dados. Como os serviços atuam em domínios diferentes, é importante que um novo paciente, cadastrado em um dos sistemas que compõem a federação, possa ser encontrado pelos demais sistemas.

Tal estrutura de federação pode ser construída através de mecanismos de correlação, em que os diferentes identificadores, de um mesmo paciente, são correlacionados em um domínio especial, chamado domínio de correlação. Esse domínio tem a função singular de correlacionar tais identificadores nos demais domínios.

A figura 2 apresenta o esquema de federação adotado no protótipo implementado para o Projeto PACS-HC. Este protótipo considera cada um dos institutos do HC como um domínio a ser correlacionado.

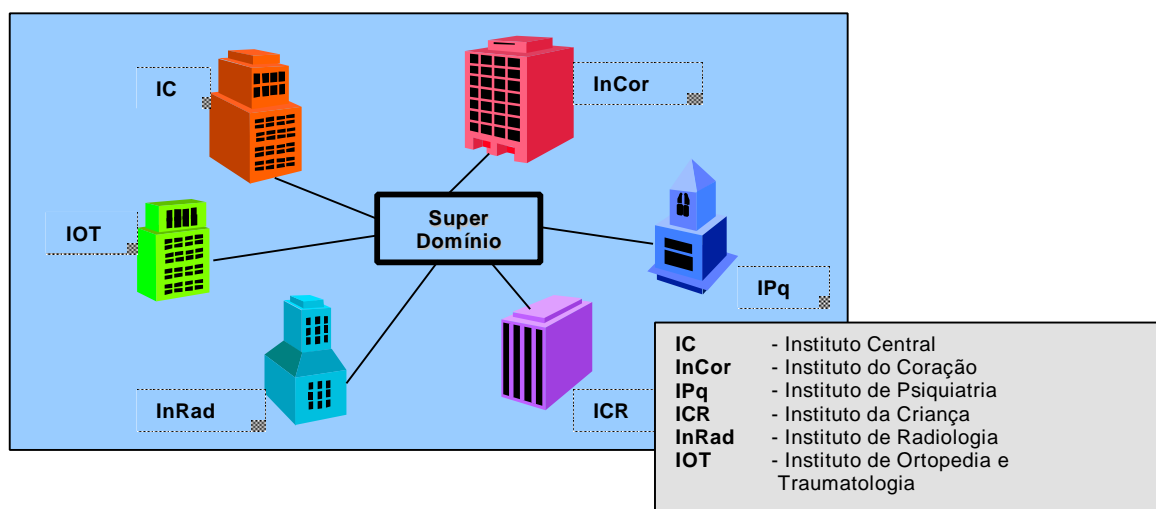


Figura 2: Esquema de Federação do PIDS

O esquema escolhido pelo InCor possui as seguintes partes:

- (a) *Super Domínio*, entidade que representa o domínio de correlação, ou seja, a base de dados que possui as informações básicas de identificação de um paciente (nome do paciente, nome da mãe, data de nascimento, local de nascimento e sexo, domínio do paciente - ex.: iot.usp.br, incor.usp.br, etc. -, identificador do paciente no domínio cadastrado, identificador do paciente no Super Domínio);

- (b) *Institutos*, entidades que possuem uma implementação de um sistema de identificação de pacientes, além do acesso ao sistema de identificação de pacientes do Super Domínio, ficando possível dessa forma, a identificação de pacientes em qualquer instituição;

A seguir exemplifica-se o cadastro de duas pessoas. No primeiro cadastro, o paciente nunca havia recebido atendimento, dentro do complexo do HC. No segundo cadastro, o paciente havia sido atendido em um instituto diferente do qual estava recebendo atendimento.

A figura 3 apresenta um diagrama de seqüência, ilustrando as operações realizadas quando um paciente que nunca recebeu atendimento em nenhum instituto é atendido no InCor.

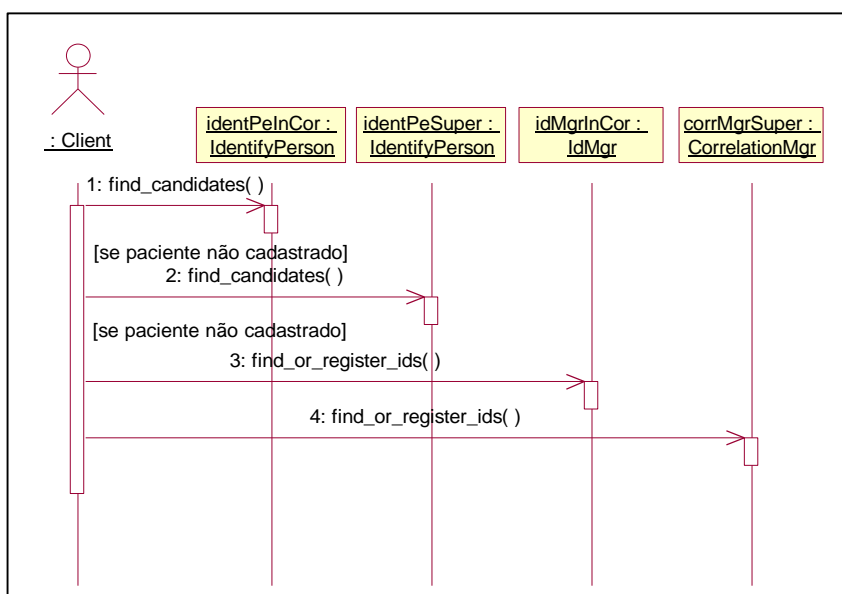


Figura 3: Paciente Novo

Os dados relevantes para identificação unívoca do paciente são fornecidos através de uma interface gráfica. A partir de tais dados, o sistema verifica se o paciente está cadastrado na base de dados local (do InCor, no exemplo citado), através da interface *IdentifyPerson*, método *find_candidates*. A seguir é verificado, na base de dados do Super Domínio, através da mesma interface, se o paciente está cadastrado. Como o paciente não estava cadastrado, procede-se com o fornecimento do restante dos dados. Primeiramente o cadastro é realizado na base de dados local, através da interface *IdMgr* do sistema de identificação do InCor, método *find_or_register_id*. Concluída tal operação, procede-se com a correlação do paciente no Super Domínio, através da interface *CorrelationMgr*, método *find_or_register_ids*.

A figura 4 ilustra a seqüência de operações realizadas quando um paciente que já recebeu atendimento em um dos institutos do complexo do HC é atendido em um outro instituto do complexo.

Os dados relevantes para identificação unívoca do paciente são fornecidos através de uma interface gráfica. A partir desses dados, o sistema verifica se o paciente está cadastrado na base de dados local (do InCor, no exemplo), através da interface *IdentifyPerson*, método *find_candidates*. A seguir é verificado, na base de dados do Super Domínio, através da mesma interface, se o paciente está cadastrado. Como o paciente estava cadastrado na base de dados remota, o sistema procede com a busca, no Super Domínio, dos dados do paciente, através do método *get_profile* da interface *ProfileAccess*. Com os dados do paciente, o sistema realiza o cadastro na base de dados local (InCor),

através da interface *IdMgr*, método *find_or_register_ids*. Após tal operação, é realizada a correlação do paciente, no Super Domínio.

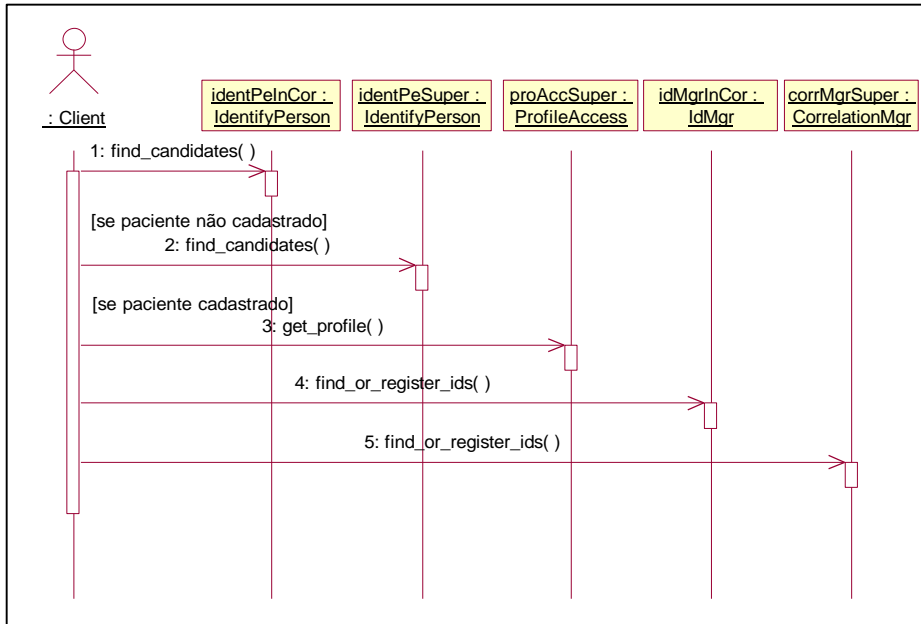


Figura 4: Paciente encontrado no Super Domínio

Dessa forma, a partir do momento que um identificador tenha sido atribuído ao paciente no Super Domínio, ou seja, que o paciente tenha recebido atendimento em algum instituto do complexo HC, ele poderá ser encontrado nos sistemas de informação de qualquer outro instituto.

3 A Implementação do PIDS

Para a implementação do serviço, foi utilizada a linguagem Java®, versão 1.2.2, e a implementação do padrão CORBA fornecida pela ferramenta Inprise Visibroker®, versão 3.3. O ambiente de desenvolvimento utilizado foi o JBuilder® 3.5, da Inprise.

Uma decisão importante na implementação do PIDS foi o conjunto de características que deveria ser utilizado no serviço. Dentro de um sistema de identificação de pacientes, considera-se como características de identificação, informações demográficas como nome, endereço, escolaridade, documentos, idade, nome dos pais, entre outras.

A especificação do PIDS pressupõe a utilização de um conjunto de características, sendo que a especificação criada pela OMG sugere dois conjuntos:

- (a) HL7 (*Health Level Seven*) [HL7, 1996] - padrão muito utilizado nos Estados Unidos; define o conteúdo das informações e não como as informações devem ser transmitidas; apesar de ser um conjunto de características muito utilizado em implementações do PIDS, tal conjunto possui algumas características que não são adequadas à realidade brasileira, tais como *SSN_NUMBER* - número de seguro social - e *MOTHERS_MAIDEN_NAME* - nome de solteiro da mãe -, enquanto que outras informações importantes como CPF ou RG não estão presentes.

- (b) Padrão dos *cartões Visa e Mastercard (vCard)* [Versit, 1996] - da mesma forma que ocorre com as características propostas pela especificação do HL7, as características do padrão Visa/Mastercard não são adequadas para identificar pacientes no Brasil; possui características que não se adaptam à realidade brasileira, tais como *FAX_TELEPHONE* - número do FAX -, *UNIFORM_RESOURCE_LOCATOR* - URL -, enquanto que informações importantes como CPF ou RG não estão presentes.

Em função de os dois conjuntos de características sugeridos pela especificação do PIDS serem inadequados, optou-se por adotar um novo conjunto, mais adequado à realidade brasileira. O conjunto de características escolhido foi o padrão proposto pelo Comitê de Padronização do Registro Clínico (PRC) [CPRC, 1999], que foi recentemente proposto pelo Ministério da Saúde. Este padrão foi resultado do trabalho de vários especialistas, na área da saúde, do Brasil e define bem as características essenciais para identificação de pacientes. Algumas das características suportadas pelo PRC são: Código, Nome, Data de Nascimento, Idade Aparente, Endereço, Local de Nascimento (Naturalidade), País de Nascimento (Nacionalidade), Sexo, Documentos, entre outras.

A implementação do PIDS do InCor também suporta o conjunto de características definidas pelo HL7. O objetivo dessa opção é permitir a comunicação com sistemas que utilizam tais características. Graças a essa funcionalidade, foi possível realizar testes de interoperabilidade com sistemas internacionais que utilizam o padrão HL7, como o PIDS do Projeto Teleméd®, de Los Alamos, e com o PIDS da empresa americana CareData.

Os apêndices A, B e C apresentam a IDL (*Interface Definition Language*) completa do PRC, HL7 e vCard, respectivamente. A IDL é uma linguagem de definição para interfaces, ou seja, uma linguagem para definição dos serviços que os objetos CORBA fornecem.

Como a construção de um sistema de identificação de pacientes como o PIDS tem como premissa fundamental a disponibilização de uma consulta capaz de encontrar nomes de pessoas de forma eficiente, tornou-se fundamental a escolha de um processo de fonetização para os nomes dos pacientes tratados pelo sistema. Optou-se por utilizar os Componentes de Fonetização do InCor [InCor, 2000], conjunto de componentes de fonetização desenvolvidos em linguagem Java® baseados em um módulo disponibilizado originalmente em C, ao CCS-SIS (Consórcio de Componentes de Software para Sistemas de Informação em Saúde) pela PROCEMPA (Companhia de Processamento de Dados do Município de Porto Alegre). A implementação destes componentes está disponibilizada no CCS-SIS, através de licença *GNU General Public Licence*. A utilização de tais componentes de fonetização permitiu que os pacientes fossem encontrados independentemente da grafia utilizada na consulta, melhorando assim, a qualidade da resposta provida pelo sistema.

4 Conclusão

Como o padrão PIDS já vem sendo utilizado em diversos hospitais no mundo, como forma de atingir a interoperabilidade entre seus sistemas, o Instituto do Coração vem aumentar essa lista de instituições que utilizam esse serviço. A implementação do PIDS permitiu a recuperação dos dados demográficos dos pacientes (nome, idade, telefone, endereço, etc.), possibilitando assim que se passe à próxima fase da construção do Projeto PACS-HC, ou seja, a recuperação das observações clínicas (laudos, sinais vitais, imagens, etc.), que é mais um passo importante para a recuperação do histórico completo dos pacientes no futuro.

Foram realizadas várias experiências com o PIDS implementado no InCor através de interações com outros sistemas que utilizavam o mesmo padrão e o resultado obtido foi bastante satisfatório. Foi possível trocar informações com PIDS implementados por outras instituições, com nenhuma implementação adicional e de forma bastante eficiente e confiável.

A utilização de padrões amplamente aceitos pela comunidade acadêmica e largamente discutidos entre profissionais de grande competência na área, permitiu a construção de um serviço que garante a interoperabilidade com outros sistemas das mais diversas plataformas que compõem o ambiente de uma instituição de saúde. A escolha de padrões como CORBA, veio ao encontro dos interesses da instituição uma vez que o padrão é aberto e definido por uma

organização de padronização internacional, sem fins lucrativos e composta pela maioria das grandes empresas e instituições de pesquisa em informática do mundo.

A adoção da tecnologia Java® na implementação deste projeto garantiu que o produto resultante fosse independente de plataforma, uma característica importante em um contexto que prevê a utilização em ambientes heterogêneos.

Por ter sido a primeira implementação brasileira do padrão PIDS, a experiência adquirida gerou uma quantidade de conhecimento importante para o país e que está sendo disseminado às demais instituições de saúde. A especificação do documento que representa as características de pacientes no Brasil foi colocada à disposição do Ministério da Saúde e do próprio Comitê CORBAMED. Várias demonstrações do sistema foram realizadas em eventos no país e espera-se que novas instituições iniciem a utilização deste padrão em breve.

6 Referências

- [CPRC, 1999] COMITÊ DE PADRONIZAÇÃO DO REGISTRO CLÍNICO. *Conjunto Essencial de Informações do Prontuário para Integração da Informação em Saúde*. Novembro, 1999. Disponível em: [ftp://ftp.datasus.gov.br/prc/Rec_001.zip]. Acessado em: março/2000.
- [GCPR, 2000] GOVERNMENT COMPUTER-BASED PATIENT RECORD. *Government Computer-based Patient Record Program*. December, 2000. Available at: [http://www.ihs.gov/gcpr/]. Visited in: december/2000.
- [HL7,1997] HEALTH LEVEL SEVEN. *HL7 Version 2.3 Specification*. 1997.
- [InCor, 2000] INSTITUTO DO CORAÇÃO. *Componentes de Fonetização do InCor*. Julho, 2000. Disponível em: [http://www.incor.usp.br/spdweb/ccsis/fonetica/]. Acessado em: abril/2000.
- [LANL, 2000] LOS ALAMOS NATIONAL LABORATORY. *OpenEMed*. December, 2000. Available at: [http://www.acl.lanl.gov/TeleMed/]. Visited in: august/2000.
- [OMG, 1999] OBJECT MANAGEMENT GROUP. *CORBAMED: Healthcare Domain Specifications*. March, 1999. Available at: [http://www.omg.org/cgi-bin/doc?formal/99-03-01.pdf]. Visited in: may/2000.
- [OMG, 1998] OBJECT MANAGEMENT GROUP. *Person Identification Service Specification*. February, 1998. Available at: [http://www.omg.org/cgi-bin/doc?formal/99-03-05.pdf]. Visited in: may/2000.
- [Seigel, 1996] SEIGEL, Jon. *CORBA Fundamentals and Programming*. United States, John Wiley and Sons, 1996. 693 p.
- [Versit, 1996] VERSIT CONSORTIUM. *vCard - The Electronic Business Card*, version 2.1, September 18, 1996.

Apêndice A – IDL PRC

A seguir apresenta-se a IDL criada pelo InCor na sua implementação do Serviço de Identificação de Pacientes para representação do conjunto de características proposto pelo Comitê de Padronização do Registro Clínico.

```
// Arquivo: PrcTraits.idl
/* Este arquivo descreve as características especificadas pelo Comitê de Padronização do Registro Clínico (PRC)
na sua recomendação final. */

#ifdef _PRC_TRAITS_IDL_
#define _PRC_TRAITS_IDL_
```

```

#include <orb.idl>
#include <PersonIdService.idl>

#pragma prefix "br.prc"

module PrcTraits
{
    typedef PersonIdService::TraitName PRCFieldName;

    /* Número do cartão SUS ou "N" se o numero não é conhecido */
    const PRCFieldName CODIGO_PACIENTE = "PRC/CodigoPaciente";
    typedef string CodigoPaciente;

    /* Nome do Paciente */
    const PRCFieldName NOME_PACIENTE = "PRC/NomePaciente";
    typedef string NomePaciente;

    /* Data de nascimento do paciente no formato DD/MM/YYYY */
    const PRCFieldName DATA_NASCIMENTO = "PRC/DataNascimento";
    typedef string DataNascimento;

    /* Idade presumida do paciente
       (se a data de nascimento não estiver disponível).
       Formato: 999 mais unidade (dias, meses, anos) */
    const PRCFieldName IDADE_APARENTE = "PRC/IdadeAparente";
    enum UnidadeTempo {DIAS, MESES, ANOS };
    struct IdadeAparente {
        short idade;
        UnidadeTempo unidade;
    };

    /* Local de nascimento do paciente. Deve ser usado o código do IBGE ou
       N se não é conhecido, ou E se o paciente é estrangeiro */
    const PRCFieldName LOCAL_NASCIMENTO = "PRC/LocalNascimento";
    typedef string LocalNascimento;

    /* País onde o paciente nasceu. Deve ser usado o padrão ISO 3166 */
    const PRCFieldName PAIS_NASCIMENTO = "PRC/PaisNascimento";
    typedef string PaisNascimento;

    /* Sexo do paciente. Valores possíveis:
       M - masculino; F - Feminino; I - Indeterminado; N - Não sabe */
    const PRCFieldName SEXO_PACIENTE = "PRC/SexoPaciente";
    enum SexoTipo {M, F, I, N };
    typedef SexoTipo SexoPaciente;

    /* Números e tipos dos documentos providos pelo paciente.
       Os valores possíveis para tipo são:
       CN - Certidão de Nascimento; CT - Carteira de Trabalho;
       RG - Registro Geral; CH - Carteira de Habilitação;
       TE - Título de Eleitor; CF - Cadastro de Pessoas Físicas;
       RE - Registro de Estrangeiro; CR - Certificado de Reservista
       CC - Certidão de Casamento; PA - Passaporte; SR - Sem Registro */
    const PRCFieldName DOCUMENTOS_PACIENTE = "PRC/DocumentosPaciente";
    enum DocumentoTipo {CN, CT, RG, CH, TE, CF, RE, CR, CC, PA, SR };

```

```

struct Documento {
    DocumentoTipo tipo;
    string numero;
    string complemento;
};
typedef sequence< Documento > DocumentosPaciente;

/* Nome da mãe do paciente */
const PRCTFieldName NOME_MAE = "PRC/NomeMae";
typedef string NomeMae;

/* Tipo e número do documento que identifica a mãe do paciente.
Veja tipos de documentos na característica DOCUMENTOS_PACIENTE. */
const PRCTFieldName DOCUMENTO_MAE = "PRC/DocumentoMae";
typedef Documento DocumentoMae;

/* Nome do pai do paciente */
const PRCTFieldName NOME_PAI = "PRC/NomePai";
typedef string NomePai;

/* Tipo e número do documento que identifica o pai do paciente.
Veja tipos de documentos na característica DOCUMENTOS_PACIENTE. */
const PRCTFieldName DOCUMENTO_PAI = "PRC/DocumentoPai";
typedef Documento DocumentoPai;

/* Situacao familiar. Tipos possíveis são:
Vive Sozinho; Vive com cônjuge ou companheiro(a);
Vive com outros familiares; Outros; Ignorado */
const PRCTFieldName SITUACAO_FAMILIAR = "PRC/SituacaoFamiliar";
enum TiposSituacaoFamiliar {VIVE_SOZINHO, VIVE_COM_CONJUGE,
VIVE_COM_FAMILIARES, OUTROS, IGNORADO };
typedef TiposSituacaoFamiliar SituacaoFamiliar;

/* Cor/Raça. Tipos possíveis são:
Branca; Negra; Amarela; Parda; Indígena; Indeterminada */
const PRCTFieldName COR_RACA = "PRC/CorRaca";
enum GruposEtnicos {BRANCA, NEGRA, AMARELA, PARDA, INDIGENA, INDETERMINADA };
typedef GruposEtnicos CorRaca;

/* Escolaridade do paciente e último ano de estudo completo.
Valores possíveis para grau de instrução são:
Alfabetização de adultos; Antigo primário; Antigo ginásio;
Antigo clássico, científico etc.; Ensino fundamental ou 1o Grau;
Ensino Médio ou 2o grau; Superior, graduação;
Superior, mestrado ou doutorado; Nenhum; Ignorado

Possíveis valores para último ano são:
Primeira; Segunda; Terceira; Quarta; Quinta; Sexta; Sétima;
Oitava; Nenhuma; Ignorado */
const PRCTFieldName ESCOLARIDADE_PACIENTE = "PRC/EscolaridadePaciente";
enum TipoEscolaridade {ALFABETIZACAO_ADULTOS, ANTIGO_PRIMARIO,
ANTIGO_GINASIO, ANTIGO_CLASSICO_CIENTIFICO, ENSINO_FUNDAMENTAL,
ENSINO_MEDIO, SUPERIOR, MESTRADO_DOUTORADO, NENHUM, IGNORADO1 };
enum UltimaSerieEscolaridade {PRIMEIRA, SEGUNDA, TERCEIRA, QUARTA,
QUINTA, SEXTA, SETIMA, OITAVA, NENHUMA, IGNORADA };

```

```

struct Escolaridade {
    TipoEscolaridade grau;
    UltimaSerieEscolaridade ultimaSerie;
};

/* Endereço do paciente. O município deve ser identificado com o código
do IBGE */
const PRCFieldName ENDERECO_PACIENTE = "PRC/EnderecoPaciente";
struct Endereco {
    string RuaAvenida;
    string numero;
    string complemento;
    string bairro;
    string cidade;
    string estado;
    string cep;
};

/* Status do endereço do paciente, se o endereço for desconhecido.
Valores possíveis são:
Ignorado; Sem Teto */
enum StatusEndereco { IGNORADO2, SEM_TETO };
enum TipoEndereco { CONHECIDO, DESCONHECIDO };

/* Se o TipoEndereco for CONHECIDO, o endereço é representado
pela estrutura "Endereco".
Se o TipoEndereco for DESCONHECIDO, o endereço é representado
pela enumeração StatusEndereco */
union EnderecoPaciente switch( TipoEndereco )
{
    case CONHECIDO : Endereco endereco_paciente;
    case DESCONHECIDO : StatusEndereco status_endereco;
};

/* Telefone para contato, no formato: <999> - <99-99-99-99>, onde 999
é o código de area e 99-99-99-99 é o número, e descricao do contato */
const PRCFieldName TELEFONE_CONTATO = "PRC/TelefoneContato";
struct Contato {
    string telefone;
    string descricao;
};

/* E-Mail do Pacinete */
const PRCFieldName E_MAIL = "PRC/EMail";
typedef string EMailPaciente;

/* Hora e data da ultima atualização do registro do paciente no
formato YYYY/MM/DD HH:MM:SS */
const PRCFieldName DATA_HORA_ATUALIZACAO = "PRC/DataHoraAtualizacao";
typedef string DataHoraAtualizacao;

/* Nome da pessoa que atualizou o registro do paciente */
const PRCFieldName NOME_REGISTRANTE = "PRC/NomeRegistrante";
typedef string NomeRegistrante;

```

```

/* Documento de identificação da pessoa que atualizou o registro
do paciente */
const PRCFieldName DOCUMENTO_REGISTRANTE = "PRC/DocumentoRegistrante";
typedef Documento DocumentoRegistrante;
};
#endif // _PRC_TRAITS_IDL_

```

Apêndice B – IDL HL7

Abaixo apresenta-se a IDL proposta pelo CORBAmed para representar o conjunto de características especificado pelo HL7.

```

// File: HL7Version2_3.idl
#ifndef _HL7_VERSION_2_3_IDL_
#define _HL7_VERSION_2_3_IDL_

#include <orb.idl>
#include <PersonIdService.idl>

#pragma prefix "omg.org"

module HL7Version2_3 {
    typedef PersonIdService::TraitName PIDFieldName;
    typedef string PIDFieldValue;

    const PIDFieldName PATIENT_NAME = "HL7/PatientName";
    const PIDFieldName MOTHERS_MAIDEN_NAME = "HL7/MothersMaidenName";
    const PIDFieldName DATE_TIME_OF_BIRTH = "HL7/DateTimeofBirth";
    const PIDFieldName SEX = "HL7/Sex";
    const PIDFieldName PATIENT_ALIAS = "HL7/PatientAlias";
    const PIDFieldName RACE = "HL7/Race";
    const PIDFieldName PATIENT_ADDRESS = "HL7/PatientAddress";
    const PIDFieldName COUNTY_CODE = "HL7/CountyCode";
    const PIDFieldName PHONE_NUMBER_HOME = "HL7/PhoneNumber_Home";
    const PIDFieldName PHONE_NUMBER_BUSINESS = "HL7/PhoneNumber_Business";
    const PIDFieldName PRIMARY_LANGUAGE = "HL7/PrimaryLanguage";
    const PIDFieldName MARITAL_STATUS = "HL7/MaritalStatus";
    const PIDFieldName RELIGION = "HL7/Religion";
    const PIDFieldName PATIENT_ACCOUNT_NUMBER = "HL7/PatientAccountNumber";
    const PIDFieldName SSN_NUMBER = "HL7/SSNNumber";
    const PIDFieldName DRIVERS_LICENSE_NUMBER = "HL7/DriversLicenseNumber";
    const PIDFieldName MOTHERS_IDENTIFIER = "HL7/MothersIdentifier";
    const PIDFieldName ETHNIC_GROUP = "HL7/EthnicGroup";
    const PIDFieldName BIRTH_PLACE = "HL7/BirthPlace";
    const PIDFieldName MULTIPLE_BIRTH_INDICATOR = "HL7/MultipleBirthIndicator";
    const PIDFieldName BIRTH_ORDER = "HL7/BirthOrder";
    const PIDFieldName CITIZENSHIP = "HL7/Citizenship";
    const PIDFieldName VETERANS_MILITARY_STATUS = "HL7/VeteransMilitaryStatus";
    const PIDFieldName NATIONALITY = "HL7/Nationality";
    const PIDFieldName PATIENT_DEATH_DATE_AND_TIME = "HL7/PatientDeathDateandTime";
    const PIDFieldName PATIENT_DEATH_INDICATOR = "HL7/PatientDeathIndicator";
};
#endif // _HL7_VERSION_2_3_IDL_

```

Apêndice C – IDL vCard

A seguir apresenta-se a IDL criada pelo Comitê CORBAmed para representar o conjunto de características do padrão Visa/Mastercard.

```
//File: vCardVersion2_1.idl
#ifndef _V_CARD_VERSION_2_1_IDL_
#define _V_CARD_VERSION_2_1_IDL_

#include <orb.idl>
#include <PersonIdService.idl>

#pragma prefix "omg.org"

module vCardVersion2_1
{
    typedef PersonIdService::TraitName PropertyName;
    typedef string PropertyValue;

    const PropertyName FORMATTED_NAME           = "vCard/FN";
    const PropertyName NAME                     = "vCard/N";
    const PropertyName PHOTOGRAPH              = "vCard/PHOTO";
    const PropertyName BIRTHDAY                = "vCard/BDAY";
    const PropertyName ADDRESS                 = "vCard/ADR";
    const PropertyName HOME_ADDRESS            = "vCard/ADR;HOME";
    const PropertyName WORK_ADDRESS            = "vCard/ADR;WORK";
    const PropertyName TELEPHONE               = "vCard/TEL";
    const PropertyName PREFERRED_TELEPHONE     = "vCard/TEL;PREF";
    const PropertyName HOME_TELEPHONE         = "vCard/TEL;HOME";
    const PropertyName WORK_TELEPHONE         = "vCard/TEL;WORK";
    const PropertyName VOICE_TELEPHONE        = "vCard/TEL;VOICE";
    const PropertyName FAX_TELEPHONE          = "vCard/TEL;FAX";
    const PropertyName MESSAGE_TELEPHONE      = "vCard/TEL;MSG";
    const PropertyName CELLULAR_TELEPHONE     = "vCard/TEL;CELL";
    const PropertyName BULLETIN_BOARD_TELEPHONE = "vCard/TEL;BBS";
    const PropertyName MODEM_TELEPHONE        = "vCard/TEL;MODEM";
    const PropertyName CAR_TELEPHONE          = "vCard/TEL;CAR";
    const PropertyName ELECTRONIC_MAIL        = "vCard/EMAIL";
    const PropertyName GEOGRAPHIC_POSITION    = "vCard/GEO";
    const PropertyName TITLE                  = "vCard/TITLE";
    const PropertyName ORGANIZATION           = "vCard/ORG";
    const PropertyName SOUND_ANNOTATION       = "vCard/SOUND";
    const PropertyName UNIFORM_RESOURCE_LOCATOR = "vCard/URL";
};
#endif // _V_CARD_VERSION_2_1_IDL_
```