

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA ABORDAGEM, BASEADA NA INTEGRAÇÃO DE
ARQUÉTIPOS A MENSAGENS HL7, PARA A
COMUNICAÇÃO DE APLICAÇÕES UBÍQUAS NO
CUIDADO DE SAÚDE PERVASIVO**

ANDERSON LUIZ MENEZES

ORIENTADOR: PROF. DR. WANDERLEY LOPES DE SOUZA

São Carlos - SP
Outubro/2011

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA ABORDAGEM, BASEADA NA INTEGRAÇÃO DE
ARQUÉTIPOS A MENSAGENS HL7, PARA A
COMUNICAÇÃO DE APLICAÇÕES UBÍQUAS NO
CUIDADO DE SAÚDE PERVASIVO**

ANDERSON LUIZ MENEZES

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes.
Orientador: Dr. Wanderley Lopes de Souza

São Carlos - SP
Outubro/2011

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

M543ab

Menezes, Anderson Luiz.

Uma abordagem, baseada na integração de arquétipos a mensagens HL7, para a comunicação de aplicações ubíquas no cuidado de saúde pervasivo / Anderson Luiz Menezes. -- São Carlos : UFSCar, 2011.
122 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2011.

1. Redes de computação. 2. Interoperabilidade. 3. Desenvolvimento orientado por modelos. 4. Computação ubíqua. I. Título.

CDD: 004.6 (20^a)


Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

“Uma Abordagem, Baseada na Integração
de Arquétipos a Mensagens HL7, para a
Comunicação de Aplicações Ubíquas no
Cuidado de Saúde Pervasivo”


ANDERSON LUIZ MENEZES

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Ciência da
Computação da Universidade Federal de São
Carlos, como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação

Membros da Banca:



Prof. Dr. Wanderley Lopes de Souza
(Orientador - DC/UFSCar)



Prof. Dr. Antonio Francisco do Prado
(DC/UFSCar)



Prof. Dr. Antonio Tadeu Azevedo Gomes
(LNCC)

São Carlos
Outubro/2011

Dedico este trabalho primeiramente à minha família que, mesmo distante, sempre esteve ao meu lado, ajudando-me a superar as dificuldades do dia-a-dia e sempre torcendo pelo meu sucesso nessa jornada. Em especial, dedico aos meus pais, Alceu e Vera, ao meu irmão Ererson e ao meu tio Darci, sem os quais eu jamais teria chegado até aqui. Mamãe, onde quer que esteja, saiba que sou muito grato por tudo o que a senhora me proporcionou. Sem dúvida, muito deste trabalho eu devo à senhora, pois nos momentos mais difíceis foram os conselhos que sempre deu que me mantiveram firme.

Não posso esquecer-me de uma pessoinha que entrou de vez em minha vida durante esse processo e que me fez enxergá-la com outros olhos, ensinando-me a extrair o máximo de cada momento. Amor, este trabalho também é seu! Sem você ao meu lado tudo teria sido ainda mais difícil.

AGRADECIMENTO

Agradeço primeiramente à minha família pelo apoio dado durante toda a minha vida. Foram vocês que me ensinaram tudo o que sei de mais importante e é de vocês que eu tiro a motivação para continuar sempre lutando, independentemente de quaisquer adversidades que possa encontrar. Agradeço principalmente aos meus pais, Alceu e Vera, por todo o esforço que empreenderam para sempre nos dar o melhor que puderam.

Pai, obrigado por nossas conversas diárias e obrigado por todo o apoio. O senhor é uma pessoa incrível, em quem sempre me espelhei e em quem sempre irei me espelhar. Mãe, mesmo a senhora não estando mais aqui presente, sei que tive sua ajuda para superar os momentos mais difíceis pelos quais passei. Obrigado por tudo! Espero que, com meu esforço, esteja conseguindo retribuir parte do que a senhora sempre fez. Everson, meu irmão, devo muito a você também. Você é mais um exemplo que sempre irei seguir. Agradeço ao senhor também, tio Darci. É sem dúvida a pessoa com o coração mais incrível que já conheci. A toda minha família o meu muito obrigado!

Agradeço também à minha namorada, Kamylla. Amor, você foi a coisa mais importante que poderia me ocorrer. Sua presença ao meu lado, mesmo distante, significou muito nessa jornada. Obrigado por me escutar, por me aconselhar, por me dar broncas e, principalmente, por me aguentar todo esse tempo. Você é muito importante para mim.

Jamais deixaria de agradecer aos meus amigos, a todos aqueles que eu já possuía e aos que consegui conquistar nesses anos de mestrado. Em especial ao Pablo, que iniciou comigo essa empreitada ainda na graduação e com quem tive o prazer de aproveitar ótimos momentos durante o mestrado; ao “irmão acadêmico” Jesús, que me recebeu muito bem e me presenteou com sua amizade; e ao Du, que me socorreu em muitos momentos de dificuldade no desenvolvimento deste trabalho. Agradeço também a todos os demais colegas e funcionários do PPG-CC por todo o apoio dado.

Por fim, deixo um agradecimento especial ao meu orientador, Prof. Wanderley, pelos ensinamentos, pela ajuda e, principalmente, pela compreensão. Professor, sou muito grato pela oportunidade que me propiciou. Tenha certeza de que aprendi muito nesses anos em que trabalhei com o senhor. Obrigado!

A ciência moderna analisou o mundo exterior; suas penetrações no Universo objetivo são profundas; isso será sua honra e sua glória; mas nada sabe ainda do universo invisível e do mundo interior. É esse o império ilimitado que lhe resta conquistar. Saber por que laços o homem se liga ao conjunto, descer às sinuosidades misteriosas do ser, onde a sombra e a luz se misturam, como na caverna de Plutão, percorrer-lhe os labirintos, os redutos secretos, auscultar o eu normal e o eu profundo, a consciência e a subconsciência; não há estudo mais necessário. Enquanto as Escolas e as Academias não o tiverem introduzido em seus programas, nada terão feito pela educação definitiva da Humanidade.

Léon Denis

RESUMO

Devido aos problemas enfrentados no Cuidado de Saúde, este caminha para uma mudança de paradigma. Tal mudança propõe que o cuidado deixe de estar centrado exclusivamente nos grandes centros hospitalares e nas mãos de profissionais especializados. O objetivo então é tornar o Cuidado de Saúde distribuído, onde o próprio paciente desempenha um papel fundamental no processo. Nesse sentido, o Cuidado de Saúde Pervasivo visa habilitar esse modelo distribuído através do emprego de Tecnologias de Informação e Comunicação (TIC), permitindo o acesso ao Cuidado de Saúde de qualquer local e a qualquer momento. A construção de ambientes para o Cuidado de Saúde Pervasivo apresenta diferentes desafios, dentre os quais estão a diversidade de dispositivos de acesso às informações e a diversidade de Sistemas de Informação Hospitalar (SIH) que podem constituir tal ambiente. Nesse contexto, a interoperabilidade aparece como um requisito fundamental, já que as informações devem ser compartilhadas independentemente das heterogeneidades existentes. Visando prover essa interoperabilidade, diferentes padrões surgiram com o objetivo de facilitar a interação entre SIHs heterogêneos. Exemplos são os protocolos de comunicação desenvolvidos pelo comitê *Health Level Seven (HL7)*, os quais são exclusivos do domínio Cuidado de Saúde. No entanto, os padrões desenvolvidos pelo comitê HL7 apresentam deficiências na representação dos conceitos clínicos, além de possuir um processo de desenvolvimento relativamente complexo. Uma alternativa para a representação do conhecimento no domínio Cuidado de Saúde é o uso de arquétipos, que são pedaços de conhecimento que indicam como representar conceitos ou informações de um dado domínio via expressões computáveis. Neste sentido, este trabalho apresenta uma abordagem, que integra arquétipos às mensagens HL7 v3, para a comunicação de aplicações ubíquas no Cuidado de Saúde Pervasivo. Com o auxílio do Desenvolvimento Dirigido a Modelos, é definido um processo composto por atividades que visam auxiliar os desenvolvedores na adoção desses padrões. Essa abordagem faz uso de modelagem e geração parcial de código para simplificar e tornar mais intuitivo o desenvolvimento das mensagens utilizadas no ambiente. A combinação dessas tecnologias permite que um padrão de comunicação já estabelecido tenha sua qualidade semântica aprimorada, contribuindo assim para o alcance da interoperabilidade.

Palavras-chave: Cuidado de Saúde Pervasivo, Interoperabilidade, Health Level Seven, Arquétipos, Computação Ubíqua, Desenvolvimento Dirigido a Modelos, Modelagem Específica de Domínio.

ABSTRACT

Due to the problems faced in Healthcare, this is heading towards a paradigm shift. This change suggests that the care is no longer focused exclusively on large hospitals and into the hands of specialized professionals. Thus, the goal is to distribute the Healthcare, making the patient play a key role in the process. Hence, the Pervasive Healthcare intends to provide the distribution of the model through the use of Information and Communication Technologies (ICT), enabling access to Healthcare anywhere and anytime. The creation of Pervasive Healthcare environments brings different challenges, which include the variety of devices to access information and the diversity of Hospital Information Systems (HIS) that may constitute such environment. In this context, the interoperability appears to be a fundamental requirement, since the information should be shared regardless of the existing heterogeneities. Aiming to provide this interoperability, different standards were set in order to facilitate the interaction between heterogeneous HISs. Some examples of these standards are the communication protocols developed by the Health Level Seven (HL7) committee, which are exclusive to the Healthcare domain. However, the standards set by the HL7 committee bring deficiencies to the clinical concepts representation and also has a relatively complex development process. An alternative to the knowledge representation in Healthcare is the use of archetypes, which are pieces of knowledge that show how to represent concepts or information of a given domain via computable expressions. Thus, this work presents an approach that integrates archetypes to HL7 v3 messages, for the communication of ubiquitous applications in Pervasive Healthcare. With the help of Model Driven Development, a process, which consists of activities designed to assist developers in adopting these standards, is defined. This approach makes use of modeling and partial code generation for simplifying and turning the development of the messages used in the environment more intuitive. The combination of these technologies enables an already established communication standard to improve its semantic quality, thus contributing to the achievement of the interoperability.

Keywords: Pervasive Healthcare, Interoperability, Health Level Seven, Archetypes, Ubiquitous Computing, Model Driven Development, Domain-Specific Modeling.

LISTA DE FIGURAS

Figura 2.1 – Dimensões da computação.....	20
Figura 3.1 – Modelo de Cuidado de Saúde Pervasivo	27
Figura 3.2 – Arquitetura do ABC	34
Figura 3.3 – Arquitetura do vSked.....	35
Figura 3.4 – Arquitetura do Ambiente.....	36
Figura 4.1 – Histórico do HL7.....	38
Figura 4.2 – HL7 e organizações de padronização.....	39
Figura 4.3 – Exemplo de mensagem em HL7	39
Figura 4.4 – Principais conceitos do HL7 v2.x.....	40
Figura 4.5 – Modelo de Referência do HL7 v3.....	42
Figura 4.6 – Communication Infrastruct Subject Area	44
Figura 4.7 – Exemplo de mensagem em HL7 v3	48
Figura 4.8 – Processo de desenvolvimento do HL7 v3	49
Figura 4.9 – Exemplo de R-MIM.....	50
Figura 5.1 – <i>Template</i> para o registro da concentração de glicose.....	53
Figura 5.2 – Arquétipo para o registro da concentração de glicose	54
Figura 5.3 – Meta-arquitetura dos arquétipos.....	55
Figura 5.4 – Estrutura <i>openEHR</i>	56
Figura 5.5 – Pacote support	57
Figura 5.6 – Pacote data_types.....	58
Figura 5.7 – Pacote data_structures	59
Figura 5.8 – Pacote common	59
Figura 5.9 – Pacote ehr.....	60
Figura 5.10 – Visão geral do EHR Information Model	61
Figura 5.11 – Pacote am	62
Figura 5.12 – Estrutura da ADL.....	63
Figura 5.13 – ADL para o registro da concentração de glicose.....	64
Figura 6.1 – Processo de criação de software no MDD	66
Figura 6.2 – Principais elementos do MDD	67

Figura 6.3 – Arquitetura clássica da metamodelagem	68
Figura 6.4 – Utilização dos adaptadores	71
Figura 6.5 – Arquitetura da ferramenta	71
Figura 6.6 – Fluxo das mensagens	72
Figura 7.1 – Integração de extratos de RESs a mensagens HL7 v3.....	75
Figura 7.2 – Proposta de Eric Browne para a integração.....	76
Figura 7.3 – Engenharia de Domínio.....	78
Figura 7.4 – Engenharia de Aplicação	80
Figura 7.5 – Busca pelo arquétipo para pressão sanguínea	81
Figura 7.6 – Descarga do arquétipo para pressão sanguínea.....	82
Figura 7.7 – Archetype Editor.....	82
Figura 7.8 – ADL gerada pelo Archetype Editor	83
Figura 7.9 – Metamodelo especificado.....	84
Figura 7.10 – Metamodelo com inclusão dos conceitos	85
Figura 7.11 – Implementação do metamodelo	85
Figura 7.12 – Geração do esquema da mensagem baseada em <i>templates</i> Jet	87
Figura 7.13 – Operação do MessageObjectMaker.....	89
Figura 7.14 – Operação do ContentObjectMaker.....	89
Figura 7.15 – Exemplo de operação de uma aplicação	90
Figura 7.16 – Arquitetura da aplicação.....	92
Figura 7.17 – Casos de uso do HiperRES	93
Figura 7.18 – Diagrama de sequência do caso de uso EnviarRegistroPressórico	93
Figura 7.19 – Modelo de mensagens	94
Figura 7.20 – Esquema de mensagens.....	96
Figura 7.21 – Teste de execução da aplicação	97
Figura 8.1 – Arquitetura abstrata do SMSCCS.....	98
Figura 8.2 – Arquitetura do SAIM.....	100
Figura 8.3 – Casos de uso do SAIM.....	102
Figura 8.4 – Diagrama de sequência do caso de uso EnviarMensagemRequisição	102
Figura 8.5 – Arquétipo para o conceito implante de marca-passo	104
Figura 8.6 – Envelope das mensagens	106

Figura 8.7 – Modelo para a mensagem de requisição	106
Figura 8.8 – Modelo para a mensagem do implante de marca-passo.....	107
Figura 8.9 – Modelo para a mensagem de ACK	108
Figura 8.10 – Esquema de mensagem para implante de marca-passo	109
Figura 8.11 – Teste de execução do SAIM	111
Figura 8.12 – Geração dos esquemas das mensagens	112
Figura 8.13 – Cenário para avaliação dos módulos de comunicação	114
Figura 8.14 – Vazão das mensagens trocadas	116

LISTA DE TABELAS

Tabela 4.1 – Conjunto de atributos estruturais.....	45
Tabela 4.2 – Principais tipos de dados do HL7 v3	46
Tabela 5.1 – Tipos de dados do <i>openEHR</i>	57
Tabela 5.2 – Estruturas genéricas do <i>openEHR</i>	58
Tabela 8.1 – Mensagens utilizadas nas simulações	115
Tabela 8.2 – Atraso verificado nas comunicações	117
Tabela 8.3 – Resultado dos formulários submetidos aos Pacientes/Médicos	118

LISTA DE ABREVIATURAS E SIGLAS

- ABC** - *Activity-Based Computing*
- ACK** - *Acknowledgement*
- ADL** - *Archetype Definitions Language*
- AM** - *Archetype Model*
- ANSI** - *American National Standards Institute*
- AOM** – *Archetype Object Model*
- API** – *Application Programming Interface*
- CCCM** – *Centro de Cirurgia Cardíaca de Marília*
- CDA** - *Clinical Document Architecture*
- CEC** – *Circulação extracórporea*
- CEN** - *European Committee for Standardization*
- CKM** – *Clinical Knowledge Manager*
- DC** – *Departamento de Computação*
- DSL** - *Domain Specific Language*
- EA** - *Engenharia de Aplicação*
- ED** - *Engenharia de Domínio*
- EMF** - *Eclipse Modeling Framework*
- EVN** - *Event Type Segment*
- GCU** - *Grupo de Computação Ubíqua*
- HiperRES** - *Hiper Registro Eletrônico de Saúde*
- HL7** - *Health Level Seven*
- ICM** – *Instituto do Coração de Marília*
- IDE** - *Integrated Development Environment*
- IHE** - *Integrating the Healthcare Enterprise*
- INCT** – *Instituto Nacional de Ciência e Tecnologia*
- ISO** - *International Organization for Standardization*
- JET** - *Java Emitter Templates*
- LNCC** – *Laboratório Nacional de Computação Científica*
- LOC** – *Lines Of Code*
- M2C** - *Modelo-para-Código*

M2M - *Modelo-para-Modelo*

MACC – *Medicina Assistida por Computação Científica*

MARTIN - *Mecanismos e Estruturas de Teleinformática*

MDD - *Model-Driven Development*

MIF - *Model Interchange Format*

MIM - *Message Information Model*

MSH – *Message Header Segment*

NHS - *National Health Service*

oAP - *openEHR Archetype Profile*

OO - *Orientação a Objetos*

OSI - *Open System Interconnection*

PDA - *Personal Digital Assistant*

PID - *Patient Identification Segment*

R-MIM - *Refined-Message Information Model*

RES - *Registro Eletrônico de Saúde*

RIM - *Reference Information Model*

RM - *Reference Model*

RMS – *Record Management System*

SADT - *Structured Analysis and Design Technique*

SAIM - *Serviço de Acesso às Informações Médicas*

SIH - *Sistema de Informação Hospitalar*

SMSCCS - *Contexto para o Cuidado de Saúde*

SNOMED-CT - *Systematized Nomenclature of Medicine - Clinical Terms*

SOAP - *Simple Object Access Protocol*

SUS - *Sistema Único de Saúde*

TAM – *Technology Acceptance Model*

TC - *Technical Committee*

TIC - *Tecnologia de Informação e Comunicação*

TOM – *Template Object Model*

UbiCon - *Ubiquitous Context Framework*

UCL - *University College of London*

UML - *Unified Modeling Language*

URL - *Universal Resource Locator*

UTI – *Unidade de Terapia Intensiva*

XMI - *XML Metadata Interchange*

XML - *eXtensible Markup Language*

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	13
1.1 Objetivos	15
1.2 Organização do Trabalho	16
CAPÍTULO 2 - COMPUTAÇÃO UBÍQUA.....	17
2.1 Computação Móvel	20
2.2 Computação Pervasiva	21
2.3 Computação Ubíqua no Cuidado de Saúde	23
CAPÍTULO 3 - CUIDADO DE SAÚDE PERVASIVO.....	25
3.1 Sistemas de Informação Hospitalar.....	27
3.1.1 Emprego dos Sistemas de Informação Hospitalar	30
3.2 Interoperabilidade.....	31
3.3 Terminologia Médica	32
3.4 Projetos no Cuidado de Saúde Pervasivo	33
CAPÍTULO 4 - HEALTH LEVEL SEVEN.....	37
4.1 HL7 v3.....	41
4.1.1 Atributos Estruturais	44
4.1.2 Tipos de Dados	45
4.1.3 Processo de Desenvolvimento	48
4.1.4 Clinical Document Architecture	51
CAPÍTULO 5 - ARQUÉTIPOS	52
5.1 Fundação <i>openEHR</i>	54
5.1.1 Reference Model	56
5.1.2 Archetype Model	62
5.1.3 Archetype Definition Language	63
CAPÍTULO 6 - DESENVOLVIMENTO DIRIGIDO A MODELOS.....	65
6.1 Principais Elementos do MDD.....	67
6.2 Metamodelagem.....	68

6.3 Linguagens Específicas de Domínio	69
6.4 MDD no Cuidado de Saúde.....	70
CAPÍTULO 7 - ABORDAGEM PROPOSTA.....	74
7.1 Objetivos da Abordagem.....	75
7.2 Visão Geral da Abordagem	77
7.3 Engenharia de Domínio.....	80
7.3.1 Especificar Arquétipos.....	81
7.3.2 Especificar Metamodelo do Domínio.....	83
7.3.3 Projetar Metamodelo do Domínio.....	84
7.3.4 Implementar Metamodelo do Domínio	85
7.3.5 Construir Transformações Modelo para Código.....	86
7.3.6 Implementar Parsers	88
7.4 Engenharia de Aplicação.....	91
7.4.1 Analisar	92
7.4.2 Projetar.....	94
7.4.3 Implementar	95
7.4.4 Testar	96
CAPÍTULO 8 - ESTUDO DE CASO.....	98
8.1 Engenharia de Aplicação.....	101
8.1.1 Analisar	101
8.1.2 Projetar.....	105
8.1.3 Implementar	108
8.1.4 Testar	109
8.2 Discussões	111
8.2.1 Aplicabilidade da Abordagem.....	112
8.2.2 Desempenho na Troca de Mensagens HL7 v3 com Arquétipos	113
CAPÍTULO 9 - CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	119
9.1 Trabalhos Futuros	120
REFERÊNCIAS.....	123

Capítulo 1

INTRODUÇÃO

O Cuidado de Saúde vem apresentando graves problemas no que diz respeito ao atendimento das reais necessidades da população, pecando principalmente no provimento de serviços com qualidade e baixo custo. Nos Estados Unidos, por exemplo, ocorreram casos onde cerca de 46% da população não recebeu os cuidados básicos recomendados e, ainda pior, casos onde mais de 11% da população recebeu cuidados que não foram recomendados e se tornaram potencialmente nocivos (MCGLYNN e tal., 2003).

Grande parte dos problemas no Cuidado de Saúde pode ser justificada pela centralização dos modelos vigentes, o qual sobrecarrega hospitais e profissionais de saúde responsáveis pelos atendimentos. Para uma prestação de serviços de saúde mais eficiente, faz-se necessário a descentralização de tais modelos e o desenvolvimento nos próprios pacientes de um senso de autogerenciamento, automonitoramento e autocuidado.

Neste contexto, a Computação Ubíqua (WEISER, 1991) pode ser extremamente útil na construção de um modelo de Cuidado de Saúde distribuído, já que o seu objetivo primordial é permitir o acesso à informação de qualquer local, a qualquer momento e a partir de qualquer dispositivo, viabilizando variadas formas de interação entre profissionais de saúde e pacientes. Dentre essas formas destacam-se o monitoramento remoto e o acesso remoto aos dados clínicos de pacientes.

O Cuidado de Saúde Pervasivo (BARDRAM; MIHAILIDIS; WAN, 2007) é um meio de alcançar a distribuição do Cuidado de Saúde, via o emprego de tecnologias da Computação Ubíqua, na medida em que visa prover o Cuidado de Saúde a qualquer pessoa, em qualquer local e a qualquer momento. Isso possibilita

“desafogar” os grandes centros hospitalares, os quais podem propiciar um atendimento de melhor qualidade aos casos mais agudos, principalmente aqueles cujo acompanhamento à distância é inviável.

Ambientes de Cuidado de Saúde Pervasivo demandam uma comunicação intensa entre os diferentes Sistemas de Informação Hospitalar (SIH), destacando-se os Registros Eletrônicos de Saúde (RES), os sistemas responsáveis pelo gerenciamento das prescrições de medicamentos, os responsáveis pelo gerenciamento das solicitações e resultados de exames clínicos, e os direcionados às questões administrativas das instituições. Esses sistemas possuem suas características próprias e muitas vezes são fornecidos por diferentes fabricantes, o que dificulta o alcance da interoperabilidade entre os mesmos.

Visando propor soluções para esse problema, foi criado em 1987 nos Estados o comitê *Health Level Seven (HL7)*, cujo objetivo principal foi desenvolver um protocolo de comunicação específico para o domínio Cuidado de Saúde, capaz de prover interoperabilidade entre sistemas heterogêneos (HL7, 2000). Diversas versões desse protocolo foram desenvolvidas, algumas destas alcançando grande aceitação junto aos fabricantes de sistemas e equipamentos médicos. Entretanto, com o passar do tempo, diversos problemas foram verificados para o provimento de interoperabilidade via o padrão estabelecido por esse comitê. Inicialmente muitos desses problemas relacionavam-se ao excesso de flexibilidade proporcionado pela família 2.x do padrão e, com o desenvolvimento da terceira família, apareceram também problemas de inconsistência na representação de conceitos clínicos.

Por outro lado, arquétipos (BEALE, 2001) foram propostos como forma de enriquecimento e padronização da representação de conceitos clínicos. Além de modelar numa única estrutura todo o conhecimento existente sobre um determinado conceito clínico, os arquétipos atendem tanto às variantes globais quanto às locais do conceito sem prejuízo da sua consistência. Dentre essas variantes estão os idiomas e as unidades de medida, as quais podem ser totalmente previstas na especificação dos arquétipos, permitindo assim o compartilhamento dos mesmos.

Buscando contribuir para o alcance dessa interoperabilidade, este trabalho propõe uma abordagem, que integra arquétipos a mensagens HL7 v3, para a comunicação de aplicações ubíquas em ambientes de Cuidado de Saúde Pervasivo. Empregando o Desenvolvimento Dirigido a Modelos (MDD) (FRANCE; RUMPE, 2007), é apresentado um processo que estabelece atividades e artefatos, os quais

auxiliam na modelagem e geração semi-automática de esquemas de mensagens a serem usados pelas aplicações do domínio. Para o cumprimento desse processo, devem ser executadas as etapas de Engenharia de Domínio (ED) e Engenharia de Aplicação (EA), sendo que na ED são especificados e desenvolvidos os principais artefatos que apoiarão a execução da EA, onde as aplicações ubíquas serão então desenvolvidas.

1.1 Objetivos

Este trabalho tem por objetivo geral propor uma abordagem, que integra arquétipos a mensagens HL7 v3, para a comunicação de aplicações ubíquas em ambientes de Cuidado de Saúde Pervasivo. Com essa integração pretende-se aliar um protocolo de comunicação, que é um padrão específico do domínio Cuidado de Saúde, a uma representação robusta de conhecimento nesse mesmo domínio. É importante ressaltar que a abordagem proposta foca no desenvolvimento dos módulos de comunicação das aplicações. O desenvolvimento das interfaces com o usuário e modelos de persistência, por exemplo, não são cobertos pela proposta deste trabalho. Além disso, a interoperabilidade mencionada limita-se a aplicações que dispõem desses módulos de comunicação, sendo que sistemas legados devem sofrer adaptações para que a comunicação possa ocorrer.

Como objetivos específicos, este trabalho visa avaliar o desempenho da troca de mensagens HL7 v3 com arquétipos, o emprego dessa abordagem no desenvolvimento de uma aplicação real para o domínio Cuidado de Saúde e a aceitação dessa aplicação no cotidiano da prestação de serviços de saúde, a fim de inferir o quanto uma aplicação desse gênero poderia influenciar nas atividades diárias dos profissionais de saúde.

1.2 Organização do Trabalho

Esta dissertação está organizada em nove capítulos, conforme descrito a seguir:

- **CAPÍTULO 2:** apresenta uma visão geral sobre a Computação Ubíqua, Computação Móvel e Computação Pervasiva, além de um exemplo de aplicação dos conceitos da Computação Ubíqua no Cuidado de Saúde;
- **CAPÍTULO 3:** introduz o conceito Cuidado de Saúde Pervasivo e outros tópicos relevantes para este trabalho, tais como Sistemas de Informação Hospitalar, Interoperabilidade e Terminologia Médica, apresentando projetos relacionados ao Cuidado de Saúde Pervasivo;
- **CAPÍTULO 4:** trata do padrão HL7, apresentando as principais características da segunda e da terceira famílias desse padrão;
- **CAPÍTULO 5:** discorre sobre os arquétipos, apresentando suas principais características e a abordagem de uso proposta pela Fundação *openEHR*, a qual é empregada neste trabalho;
- **CAPÍTULO 6:** aborda o Desenvolvimento Dirigido a Modelos, Metamodelagem e Linguagens Específicas de Domínio, apresentando alguns exemplos de seu emprego no Cuidado de Saúde;
- **CAPÍTULO 7:** apresenta a abordagem proposta neste trabalho, descrevendo cada uma de suas atividades;
- **CAPÍTULO 8:** apresenta o estudo de caso, seguido de discussões acerca das observações realizadas durante a sua avaliação; e
- **CAPÍTULO 9:** conclui esta dissertação, apresentando algumas considerações finais, as contribuições alcançadas e possibilidades de trabalhos futuros.

Capítulo 2

COMPUTAÇÃO UBÍQUA

Partindo do pressuposto de que o termo ubíquo é definido como algo onipresente, que está ao mesmo tempo em toda a parte (FERREIRA, 2009), pode-se afirmar que a ubiquidade já se tornou uma realidade no dia a dia das pessoas. Atualmente a informação deixou de estar presente apenas nos livros e pode ser encontrada em qualquer lugar. Ao caminhar pelas ruas de uma grande cidade, é comum deparar-se com grandes quantidades de informações, as quais são assimiladas sem nem ao menos tomar ciência de que isso está ocorrendo e de que forma a informação está disponível. A informação está presente nas vitrines das lojas, nas placas de trânsito, nos *outdoors*, e pode ser adquirida de forma totalmente transparente.

Sendo considerada a nova era da Computação, a Computação Ubíqua (HANSMANN et al., 2003) busca criar um novo paradigma computacional a partir da exata definição de ubiquidade, tendo como principal propósito permitir ao usuário, de qualquer lugar, a qualquer momento e de qualquer dispositivo, o fácil acesso e processamento da informação.

Segundo seu idealizador Mark Weiser, “*as mais profundas tecnologias são as que desaparecem*” (WEISER, 1991). Nesse sentido, a Computação Ubíqua busca inserir a Computação no mundo físico do usuário, permitindo que este foque apenas nas tarefas que pretende realizar e não nas ferramentas que necessitará para tal. Ou seja, o objetivo é permitir que as pessoas convivam diariamente com ambientes saturados de dispositivos computacionais, criando um mundo onde esses ambientes desempenham um papel fundamental no cotidiano dos cidadãos, auxiliando-os em todas as suas atividades (POSLAD, 2009).

Mesmo não apresentando todas as características propostas por Mark Weiser, a Computação Ubíqua já é uma realidade no cotidiano das pessoas. Sua existência pode ser comprovada pelas tecnologias cada vez mais presentes nas atividades diárias, tanto no âmbito empresarial quanto no doméstico. Dentre tais tecnologias estão os dispositivos móveis, etiquetas inteligentes, controles de transporte e as redes de comunicação sem fio (BELL; DOURISH, 2007).

O desenvolvimento dos sistemas ubíquos possui pelo menos três princípios básicos (ARAUJO, 2003):

- **Diversidade:** relacionada aos dispositivos e não às aplicações, a diversidade propõe o uso de computadores de propósito específico no desempenho de tarefas específicas. Por exemplo, enquanto um *tablet* é mais conveniente ao consumo de conteúdos, um *notebook* pode ser mais conveniente à redação de longos textos;
- **Conectividade:** a presença de diferentes dispositivos desempenhando tarefas específicas demanda uma alta conectividade, para propiciar a cooperação entre os mesmos e a mobilidade de seus usuários;
- **Descentralização:** mesmo com a presença de servidores, ambientes cooperativos e distribuídos demandam descentralização, já que é interessante que o fluxo da informação propague-se também entre os hospedeiros, independente das características particulares dos dispositivos do ambiente. A descentralização e a diversidade vêm acompanhadas da sincronização da informação, uma vez que é imprescindível que esta seja constantemente atualizada em todos os dispositivos para manter a consistência dos dados.

Na Computação Ubíqua há também grandes desafios relacionados à segurança, privacidade, expansividade e complexidade (ALMEIDA; BARRENHO, 2008):

- Devido ao alto nível requerido de conectividade, como prevenir tentativas de acesso mal intencionadas? A **segurança** possui grande relevância, sendo necessário disponibilizar infraestruturas embarcadas capazes de proteger os usuários do roubo de informações e de outras

formas de acesso não autorizadas. A **privacidade** possui a mesma relevância, pois todas as informações de âmbito particular armazenadas nos dispositivos devem estar protegidas, evitando que as mesmas possam ser acessadas por qualquer usuário;

- Como permitir que diferentes configurações de hardware e software interajam entre si? A **expansividade** é inerente à Computação Ubíqua, já que esta visa permitir o acesso e processamento da informação de qualquer local e usando qualquer tipo de dispositivo, sendo inviável restringir a expansão de ambientes e dispositivos. Essa questão está também relacionada à **complexidade**, tanto no que diz respeito à interação entre dispositivos quanto à interação homem-computador. É fundamental que ambientes ubíquos sejam capazes de selecionar o dispositivo que receberá determinada informação e irá exibí-la de forma apropriada ao usuário.

Além da Computação Ubíqua, outros termos surgiram nessa nova era da Computação, destacando-se Computação Móvel e Computação Pervasiva. Embora muitos autores utilizem esses termos como sinônimos, a Computação Ubíqua pode ser vista como a intersecção das demais, possuindo ao mesmo tempo o alto grau de mobilidade da Computação Móvel e o alto grau de embarcamento da Computação Pervasiva, conforme ilustrado na Figura 2.1 (LYYTINEN, 2002).

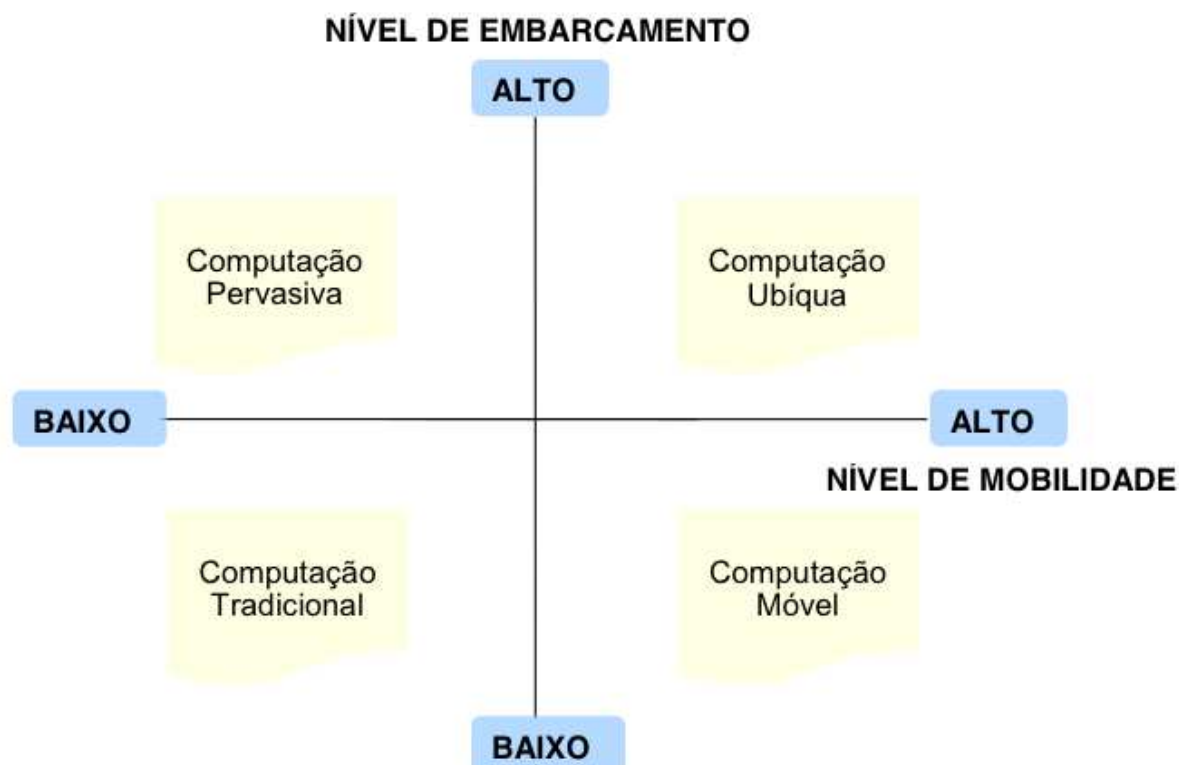


Figura 2.1 – Dimensões da computação

2.1 Computação Móvel

Segundo (MATEUS; LOUREIRO, 1998) a Computação Móvel “surge como uma quarta revolução da computação, antecedida pelos grandes centros de processamento de dados na década de sessenta, o surgimento dos terminais nos anos setenta e as redes de computadores na década de oitenta”.

A comunicação sem fio vem sendo usada há muito tempo nos sinais de rádio e televisão, mas a sua associação à portabilidade e à mobilidade é mais recente. A tecnologia que pode ser considerada como ponto de partida foi lançada em 1979 no Japão, sendo sucedida pela instalação das redes de telefonia celular em Chicago e Baltimore no ano de 1983 (MATEUS; LOUREIRO, 1998).

O objetivo da Computação Móvel é fornecer, com suporte à mobilidade, recursos computacionais comparáveis às estações de trabalho convencionais, ou seja, prover processamento e troca de informações via rede sem fio com o uso de dispositivos portáteis (FIGUEIREDO; NAKAMURA, 2003). Sendo uma área da

comunicação sem fio, seus dispositivos são caracterizados pelo tamanho reduzido, uso de baterias e a presença de uma tecnologia de acesso a redes sem fio.

Diante das particularidades que este modelo computacional impõe na forma de interação entre homem-máquina e entre dispositivos, surgem alguns desafios a ser superados (FIGUEIREDO; NAKAMURA, 2003):

- **Energia:** por se tratarem de dispositivos móveis, estes devem possuir sua própria fonte de energia. Ao mesmo tempo, deve-se garantir a melhor relação entre poder computacional e consumo, visto que essas baterias precisam apresentar uma duração satisfatória e permitir aos usuários tempos consideráveis de uso;
- **Segurança:** uma vez que todos os dados são propagados pelo ar, esquemas de autenticação e criptografia devem ser disponibilizados, visando reduzir a incidência de ataques maliciosos e o roubo de informações;
- **Adaptação:** é importante que os dispositivos possam agir de acordo com as características do ambiente e do usuário. Por exemplo, as informações manipuladas pelos usuários devem ser sincronizadas enquanto existir conectividade e, na ausência desta, o dispositivo deve armazená-las localmente para posterior sincronização das mesmas;
- **Aplicações e serviços:** na Computação Móvel serviços e aplicações devem ser disponibilizados de tal forma a permitir o seu acesso, por parte dos usuários, de qualquer local e a qualquer momento.

2.2 Computação Pervasiva

Na Computação Pervasiva os dispositivos devem atuar como um portal para o espaço das aplicações e não apenas como um repositório de software que o usuário deve gerenciar. As aplicações, por sua vez, são um meio através do qual o usuário efetua suas tarefas e não apenas um software escrito para explorar o poder computacional. Além disso, um ambiente computacional é um espaço físico de

informações aprimorado, não somente um ambiente virtual existente para armazenar e executar os softwares (SAHA; MUKHERJEE, 2003).

A Computação Pervasiva incorpora o conceito de uma Computação transparente, inteligente e adaptável, com os computadores embutidos no ambiente. De acordo com (ARAUJO, 2003), *“nessa concepção, o computador tem a capacidade de obter informação do ambiente no qual está embarcado e a utilizar para dinamicamente construir modelos computacionais”*.

A busca pela pervasividade computacional impõe uma série de mudanças não apenas na nossa forma de pensar, mas também no desenvolvimento das aplicações e dispositivos. Segundo (SAHA; MUKHERJEE, 2003), a Computação Pervasiva inclui-se em muitas das pesquisas relacionadas à Computação Móvel, gerando porém novas questões para discussão. Os principais desafios da Computação Pervasiva são (SAHA; MUKHERJEE, 2003):

- **Escalabilidade:** é a capacidade de manipular porções crescentes de trabalho de forma uniforme ou estar preparado para o seu crescimento. Na medida em que crescem os ambientes pervasivos, torna-se inevitável o aumento dos dispositivos e das interações homem-máquina nesses ambientes;
- **Heterogeneidade:** assumindo que implementações uniformes de ambientes inteligentes são impraticáveis, é necessário buscar meios para não deixar que essa heterogeneidade seja percebida pelos usuários. Por exemplo, um ambiente inteligente desenvolvido para um hospital terá grandes diferenças se comparado a outro desenvolvido para um supermercado. Dessa forma, o uso de *middlewares* e outros recursos computacionais torna-se inevitável, para que os dispositivos possam migrar por esses ambientes sem que as diferenças sejam percebidas pelo usuário;
- **Integração:** embora haja inúmeros esforços para o desenvolvimento de componentes para ambientes pervasivos, a integração entre esses componentes ainda é objeto de pesquisa. Por exemplo, os dispositivos necessitam de respostas rápidas dos servidores para suprirem as necessidades dos usuários e, ao mesmo tempo, esses servidores precisam gerenciar grandes quantidades de dispositivos concorrentes.

A busca pela integração e cooperação é inevitável para se prover serviços de forma eficiente;

- **Invisibilidade:** é um requisito fundamental para os sistemas que não necessitam de muita intervenção humana. Embora tais intervenções ocorram e são importantes ao processo de aprendizado do sistema, este deve possuir estratégias de autoajuste capazes de reduzir a necessidade de intervenção, o que permitirá ao sistema alcançar um nível de transparência “pleno” ao cotidiano de seus usuários;
- **Percepção:** está relacionada à capacidade de reconhecimento do contexto vigente do ambiente no qual o sistema está embarcado. A Computação Pervasiva faz da ciência de contexto um pré-requisito no desenvolvimento de seus ambientes, tornando-o bem mais complexo. Esses ambientes devem ser pró-ativos, isto é, devem ser capazes de tomar decisões antes que determinados fatos possam ocorrer;
- **Inteligência:** está relacionada ao poder de gerenciar as mudanças ocorridas no contexto. Como os sistemas percebem alterações no ambiente, devem usar essas percepções de forma eficiente.

2.3 Computação Ubíqua no Cuidado de Saúde

Um exemplo de como a Computação Ubíqua pode ser aplicada no Cuidado de Saúde é descrito no cenário proposto por (JORGENSEN, 2002). Nesse cenário uma enfermeira, que necessita administrar medicamentos aos seus pacientes, porta um crachá que emite sinais e habilita vários computadores a perceberem sua localização. Enquanto ela se aproxima da sala onde os medicamentos estão guardados, um computador nessa sala acessa automaticamente o registro eletrônico e verifica nos prontuários as prescrições emitidas para cada um dos pacientes, que são de responsabilidade daquela enfermeira, apresentando-as na tela. A enfermeira prepara as bandejas e usa o computador para registrar corretamente o momento em que os medicamentos de cada paciente foram retirados, colocando as bandejas em um carrinho. O carrinho é equipado com um computador sensível à localização que, ao se aproximar de um leito, automaticamente busca o prontuário do paciente e o

exibe na tela, facilitando e agilizando o registro da administração do medicamento. Caso o paciente possua alguma dúvida sobre os medicamentos, a enfermeira está capacitada também a prestar esclarecimentos através de guias *online* usando seu *Personal Digital Assistant (PDA)*.

O exemplo citado ilustra com bastante precisão três dos principais desafios enfrentados pela Computação Ubíqua: o reconhecimento das intenções do usuário, sensibilidade ao contexto e adaptabilidade. Tratando-se de um ambiente inteligente, tais características são cruciais para que a interação entre os usuários e as tecnologias existentes seja feita da forma mais transparente possível. Ainda no domínio do Cuidado de Saúde, outras importantes características, que devem ser garantidas pelos ambientes, são a segurança e a privacidade das informações, pois é sabido que a maioria dos pacientes somente concorda em usar sistemas para o Cuidado de Saúde Pervasivo se o acesso às informações contidas nesses sistemas for restrito aos profissionais de confiança, cuidadores de saúde envolvidos e familiares (VARSHNEY, 2009).

Capítulo 3

CUIDADO DE SAÚDE PERVASIVO

O Cuidado de Saúde vigente em vários países enfrentará em breve uma série de desafios, sobretudo devido: ao rápido crescimento no número de idosos; ao aumento de doenças crônicas; à necessidade de novos tratamentos e tecnologias; e à diminuição do número de profissionais de saúde comparada ao aumento populacional. Uma prova disso é o que se observou recentemente nos Estados Unidos, onde a *United States Census Bureau* estimou que o número esperado de habitantes com mais de 65 anos será de aproximadamente 70 milhões em 2030, que é o dobro em relação a 2000 (JIANG et al., 2009). Em Ontário, a província mais povoada do Canadá, a previsão em 2005 foi que o Cuidado de Saúde representaria 50% das despesas governamentais em 2011, 66% em 2017 e 100% em 2026 (SKINNER, 2005).

O modelo atual de Cuidado de Saúde encontra-se concentrado em torno de pessoas altamente especializadas, localizadas em grandes hospitais e focando em casos agudos de tratamento. Esse modelo pode ser comparado ao modelo de processamento de dados centralizado da primeira era da computação, ou seja, a era dos *mainframes*. Esse “Cuidado de Saúde Mainframe” precisa ser distribuído, a fim de produzir respostas mais rápidas e auxiliar o paciente a gerenciar a sua própria saúde. Um modelo de “Cuidado de Saúde Distribuído”, que permeie o cotidiano dos cidadãos, é muito mais apropriado para enfrentar os desafios citados. Sendo assim, o objetivo da *Pervasive Healthcare*, neste trabalho denominada Cuidado de Saúde Pervasivo, é habilitar esse modelo distribuído para o gerenciamento da saúde e bem estar (BARDRAM; MIHAILIDIS; WAN, 2007).

Segundo (ARNRICH et al., 2010), o Cuidado de Saúde Pervasivo está diretamente relacionado à Engenharia Biomédica, à Informática Médica e à Computação Ubíqua, podendo ser definido tanto como a aplicação das tecnologias da Computação Pervasiva no Cuidado de Saúde, quanto como um meio para disponibilizá-lo em qualquer lugar, a qualquer momento e para qualquer pessoa. Enquanto a Engenharia Biomédica e a Informática Médica estão preocupadas com o desenvolvimento de tecnologias para melhorar o atual modelo de Cuidado de Saúde, o Cuidado de Saúde Pervasivo foca numa mudança de paradigma.

Um modelo distribuído de Cuidado de Saúde é dependente do uso de novas tecnologias, ferramentas e serviços que possibilitem ao paciente desempenhar um papel mais ativo no gerenciamento de sua própria saúde e no tratamento de doenças, colocando-o no centro do processo de Cuidado de Saúde. A Figura 3.1 (VARSHNEY, 2003), ilustrativa de um modelo de Cuidado de Saúde Pervasivo, mostra um ambiente que suporta serviços de telemedicina móvel e monitoramento de pacientes, serviços médicos baseados em localização e resposta a emergências, e permite também o autogerenciamento, automonitoramento e autocuidado. Tais ambientes têm a capacidade de melhorar consideravelmente a qualidade dos atendimentos médicos, já que em ambientes tradicionais muitos erros ocorrem devido à dificuldade de acesso à informação no momento e local em que a mesma é requisitada, resultando em diagnósticos errôneos e administração equivocada de medicamentos (KOHN; CORRIGAN; DONALDSON, 2000 apud VARSHNEY, 2009).

A infraestrutura necessária para a criação de um ambiente de Cuidado de Saúde Pervasivo usualmente engloba redes de sensores e redes sem fio. O principal desafio é como integrar essas tecnologias, formando ambientes inteligentes capazes de reunir e disponibilizar informações de auxílio ao diagnóstico e à tomada de decisão, permitindo aos usuários o acesso a tais informações quando, onde e como melhor lhes aprouverem.

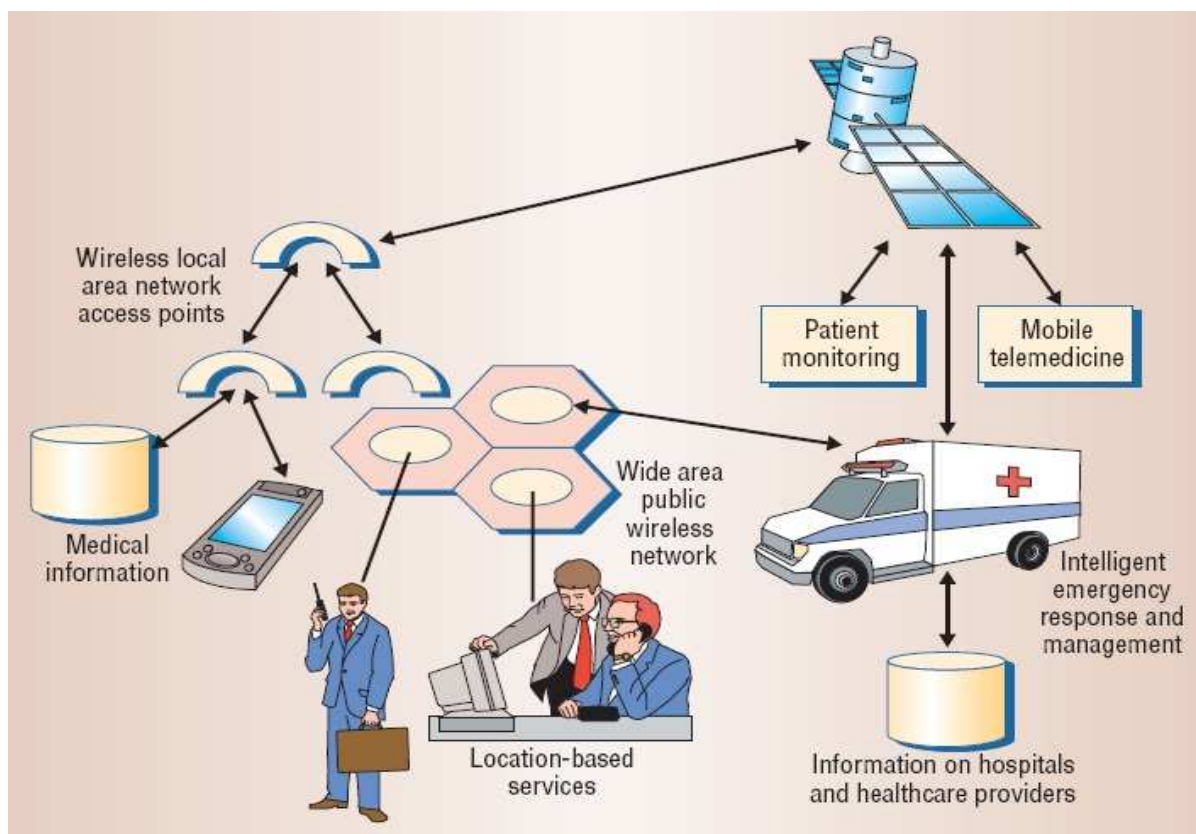


Figura 3.1 – Modelo de Cuidado de Saúde Pervasivo

3.1 Sistemas de Informação Hospitalar

O primeiro registro de emprego de recursos computacionais no Cuidado de Saúde remonta aos anos 50, época na qual o Dr. Robert S. Ledley trabalhou em um projeto, na área de odontologia, da *United States National Bureau of Standards*. Dentre seus inúmeros projetos, Ledley ficou conhecido pelo desenvolvimento do primeiro tomógrafo computadorizado de corpo inteiro, o qual revolucionou os diagnósticos médicos (SITTIG; ASH; LEDLEY, 2006).

Com o passar dos anos, a crescente demanda por cuidados médicos e o acelerado desenvolvimento tecnológico culminaram na criação dos Sistemas de Informação Hospitalar (SIHs). Um SIH é responsável por armazenar, gerenciar e disponibilizar aos interessados (*stakeholders*) diversos tipos de informação: dados administrativos; informações demográficas de pacientes; prontuários eletrônicos; e solicitações e resultados de exames clínicos. Além da informatização, esses SIHs e as Tecnologias de Informação e Comunicação (TICs) podem suportar uma mudança

de paradigma no Cuidado de Saúde, tornando-o distribuído e centrado no paciente, um requisito essencial do Cuidado de Saúde Pervasivo.

A *National Health Service (NHS)*, agência responsável pela saúde pública na Grã-Bretanha, apresenta na *NHS Informatics Review* de 2008 cinco requisitos fundamentais para que um SIH possa dar suporte ao Cuidado de Saúde centrado no paciente (BENSON, 2009): permitir sua integração aos demais sistemas, produzindo relatórios sofisticados; permitir o intercâmbio de documentos, tais como solicitações e relatórios de diagnósticos, incluindo todos os testes patológicos e aqueles efetuados nos cuidados primários; permitir agendamentos, tais como de leitos e exames; permitir prescrições eletrônicas de medicamentos, inclusive aqueles a serem administrados após a alta médica; e produzir documentos baseados numa codificação específica, que represente terminologias médicas.

Dois dos cinco requisitos levantados pela NHS mostram que os SIHs dependem diretamente da comunicação. Porém, independente do Cuidado de Saúde possuir ou não um suporte tecnológico, a maioria dos processos médicos envolve a troca de informações, devido principalmente à grande quantidade de documentos que é gerada (e.g., relatórios, solicitações, avaliações, prescrições). Essa comunicação ocorre tanto de forma restrita, por exemplo cobrindo apenas um grupo de trabalho para o registro e gerenciamento do cuidado de um único paciente, quanto de forma mais abrangente: cobrindo diferentes departamentos de diagnóstico e tratamento, para a requisição de serviços e relatórios; cobrindo diferentes provedores de Cuidado de Saúde, tais como unidades de saúde e hospitais, permitindo a continuidade do cuidado; e cobrindo provedores e agências reguladoras para fins de prestação de contas.

No Reino Unido podem ser observadas implicações do uso da comunicação no Cuidado de Saúde, cujo aprimoramento encontra-se estagnado devido à resistência dos hospitais na adoção de SIHs interoperáveis (BENSON, 2009). Enquanto os profissionais das unidades básicas de saúde esforçam-se para a informatização dessas unidades há mais de 40 anos, resultando hoje em 100% das mesmas informatizadas, os hospitais consideram a informatização um recurso menos necessário, representando apenas um custo a mais no orçamento. Como consequência, os poucos sistemas em uso nos hospitais não se integram aos sistemas das unidades básicas de saúde, dificultando o compartilhamento e o acesso à informação. Além disso, quase todas as prescrições emitidas pelas

unidades são digitais, enquanto os hospitais continuam com documentos em papel sendo apenas 5% em meio digital.

Nos Estados Unidos há também preocupação quanto aos SIHs na atual política de serviços de saúde desse país. A intenção do governo é de que todos os norte-americanos possuam Registros Eletrônicos de Saúde (RES) interoperáveis até 2014. Segundo o ex-secretário da saúde Tom Daschle, *“o sistema de saúde é incrivelmente primitivo quando existe a necessidade de usar a informação. Apenas 15% ou 20% dos médicos possuem registros computadorizados de seus pacientes e apenas uma pequena fração das bilhões de transações médicas, que ocorrem por ano nos Estados Unidos, são conduzidas eletronicamente”* (BENSON, 2009). Em 2009 somente 1,5% dos hospitais norte-americanos possuíam SIHs realmente interoperáveis.

O intercâmbio de informações entre os diversos sistemas usados no Cuidado de Saúde vem sendo discutido há anos. Nas últimas três décadas esforços consideráveis vêm sendo despendidos na criação de infraestruturas que permitam a integração de SIHs. Tais infraestruturas devem prover interoperabilidade entre os SIHs e o acesso ubíquo às informações geradas pelos mesmos. Prova da preocupação quanto a essa comunicação, foi a criação nos Estados Unidos do comitê *Health Level Seven (HL7)*, fundado com o objetivo de desenvolver um protocolo de comunicação para o intercâmbio de dados entre SIHs heterogêneos (HL7, 2000).

Particularmente no Brasil, os SIHs têm grande relevância na Saúde Coletiva. O SIH do Sistema Único de Saúde (SUS) é uma fonte de dados importante na elaboração de políticas de saúde e no planejamento e gestão de serviços de saúde no país. Foi possível verificar nas últimas décadas o crescimento do uso da base de dados do SIH/SUS em número, abrangência, diversidade de conteúdos e complexidade de análise. Contudo, identificam-se problemas de confiabilidade dos dados existentes devido a pouca padronização no preenchimento das informações, dificultando o alcance da interoperabilidade (BITTENCOURT; CAMACHO; LEAL, 2006).

3.1.1 Emprego dos Sistemas de Informação Hospitalar

Em (ORWAT; GRAEFE; FAULWASSER, 2008) é apresentado um *survey* sobre o emprego de SIHs, sendo que para elaboração desse trabalho foram efetuadas buscas nas principais bases de dados disponíveis, tendo sido selecionados 69 artigos que possibilitaram a identificação de 67 sistemas.

Os sistemas foram divididos em diferentes contextos, tais como ambulatorial, residencial, móvel e clínico, resultando em 57% de uso móvel e residencial e 36% de uso clínico. De acordo com esse artigo 63% dos sistemas conhecidos são direcionados para a prevenção e o Cuidado de Saúde, 39% para a organização dos provedores de Cuidado de Saúde e 12% para terapia e reabilitação¹.

Os usuários desses sistemas foram divididos em profissionais de saúde e outros “cuidadores” de saúde (e.g., pacientes, cuidadores particulares, familiares), sendo que os próprios pacientes são os principais usuários, já que usam 72% dos sistemas. Dentre os profissionais de saúde, médicos e enfermeiros são os usuários mais ativos, já que usam 54% e 51% dos sistemas respectivamente.

Em relação ao intercâmbio de dados, o artigo informa que 88% dos sistemas transmitem dados com o propósito de armazenamento, análise ou repasse, sendo que na maioria dos casos esses dados são transmitidos a um servidor central. Os demais sistemas não dependem diretamente da comunicação e desempenham suas funcionalidades de forma independente em um modelo computacional descentralizado.

Informações referentes aos impactos financeiros, provenientes do uso de SIHs no Cuidado de Saúde, foram encontradas em apenas 10 dos artigos selecionados, sendo que os autores não efetuaram no artigo análises mais profundas relativas aos custos e benefícios do emprego de tais sistemas.

Com base no estudo efetuado, percebe-se que o foco principal dos SIHs está no paciente e na sua qualidade de vida. A disponibilização desses sistemas em lares e dispositivos móveis, associada à transmissão de dados e o uso em larga escala dos mesmos pelos profissionais de saúde, comprovam o interesse pela adoção de um modelo distribuído de Cuidado de Saúde. Esse tipo de modelo possibilita aos médicos e cuidadores de saúde o acompanhamento remoto de seus pacientes e

¹Múltiplas entradas em diferentes categorias foram permitidas pelos autores.

auxilia os pacientes no gerenciamento de sua própria saúde, implicando na redução de custos e de demanda direta por profissionais de saúde.

3.2 Interoperabilidade

O termo interoperabilidade refere-se à habilidade de dois ou mais sistemas interagirem e trocarem informações visando um determinado objetivo (ISO, 2009). Alcançar interoperabilidade no domínio Cuidado de Saúde é uma tarefa complexa, visto que os dados de registros de pacientes contêm uma grande variedade de informações, as quais precisam ser trocadas e compreendidas por sistemas essencialmente heterogêneos. Neste domínio em especial, o intercâmbio de informações deve satisfazer a dois níveis de interoperabilidade: a *funcional* refere-se à legibilidade das informações, objetivando a sua compreensão e uso pelo humano; e a *semântica*, que tem o mesmo objetivo só que em relação à máquina (SCHLOEFFEL et al., 2006).

Só na Inglaterra são 30 mil unidades básicas de saúde encaminhando mais de 15 milhões de pacientes por ano para cerca de 60 mil hospitais e recebendo como retorno mais de 40 milhões de documentos clínicos e altas hospitalares. Cada médico, responsável pelos cuidados primários, pode encaminhar pacientes para qualquer especialista e cada um desses especialistas pode receber encaminhamentos de qualquer unidade básica de saúde. Além disso, qualquer médico pode comunicar-se com grandes quantidades de provedores de serviços ou entidades administrativas (BENSON, 2009).

Esse modelo de comunicação de muitos para muitos e o grande volume de informações trocadas representam apenas parte da complexidade para o provimento de interoperabilidade. O principal complicador é a diversidade de sistemas, refletindo na diversidade de arquiteturas e plataformas. Cada um desses 60 mil hospitais ingleses pode possuir seu próprio sistema, baseado numa arquitetura tipicamente proprietária, que organiza suas informações de forma totalmente particular.

Diante desse quadro, a interoperabilidade torna-se um dos principais desafios para a integração de SIHs e para a criação de ambientes de Cuidado de Saúde Pervasivo. Os dados trafegados devem possuir o mesmo significado, tanto para o

sistema A quanto para o sistema B, e envolvem não apenas o compartilhamento da informação propriamente dita, mas também todo conhecimento nela contido. Para tal, deve existir um consenso prévio entre os humanos quanto às definições dos conceitos, a fim de possibilitar a criação de modelos de representação do conhecimento, amenizando dessa forma as dificuldades de comunicação.

3.3 Terminologia Médica

Toda ciência possui um conjunto específico de terminologias, a fim de formalizar e facilitar os processos de comunicação e compreensão. O uso correto dos termos é de extrema importância em qualquer comunicação, pois permite aos interlocutores a exata compreensão da informação que é transmitida. Essa mesma realidade faz parte do contexto computacional, mas para que a terminologia possa fazer sentido aos computadores é necessária uma codificação da mesma, permitindo assim o seu processamento.

Uma codificação importante no domínio Cuidado de Saúde é a *Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT)* (SNOMED, 2011), que é a terminologia clínica multilíngue mais inteligível do mundo. Atualmente a SNOMED-CT possui mais de 310 mil conceitos, 990 mil descrições e mais de 1 milhão e 300 mil relacionamentos. Além de um esquema de codificação e identificação de conceitos e termos, a SNOMED-CT é também uma classificação multidimensional, que permite o relacionamento, agrupamento e análise desses conceitos sob diferentes critérios.

A disseminação e adoção de um esquema de codificação, tal como o SNOMED-CT, viabiliza uma maior concordância quanto às informações clínicas a serem transmitidas, facilitando assim a interoperabilidade entre os sistemas. Porém, além da codificação, a estruturação das mensagens e dos dados transportados é de extrema relevância para se alcançar essa interoperabilidade.

3.4 Projetos no Cuidado de Saúde Pervasivo

Activity-Based Computing (ABC) (BARDRAM; CHRISTENSEN, 2007) é um projeto que busca investigar que tipo de suporte a Computação Pervasiva pode fornecer aos ambientes hospitalares. O trabalho nos hospitais, por ser orientado a equipes e por depender diretamente da colaboração entre os diferentes campos de especialidade, é um grande desafio para a Computação Pervasiva. Além disso, o cotidiano dos cuidadores de saúde é inerentemente “nômade”, já que estes necessitam atender os pacientes, participar de reuniões, discutir com seus colegas e a maioria “não possui uma escrivaninha”.

ABC está relacionado a cinco temas médicos, que refletem as principais áreas de atuação dos grandes hospitais: administração de medicamentos por enfermeiros; prescrição de medicamentos pelos médicos; colaboração entre cuidadores de saúde; conferências médicas; e cirurgias. Para atender a essas necessidades, o projeto ABC apresenta soluções centradas em atividades, visando reduzir significativamente a sobrecarga dos usuários na abertura de aplicativos, navegação de dados e navegação na interface de interação.

A Figura 3.2 mostra a arquitetura do projeto, a qual usa comunicações cliente-servidor para as atividades e gerenciamento de estado, e comunicações ponto-a-ponto para conexões de voz e outras formas de colaboração.

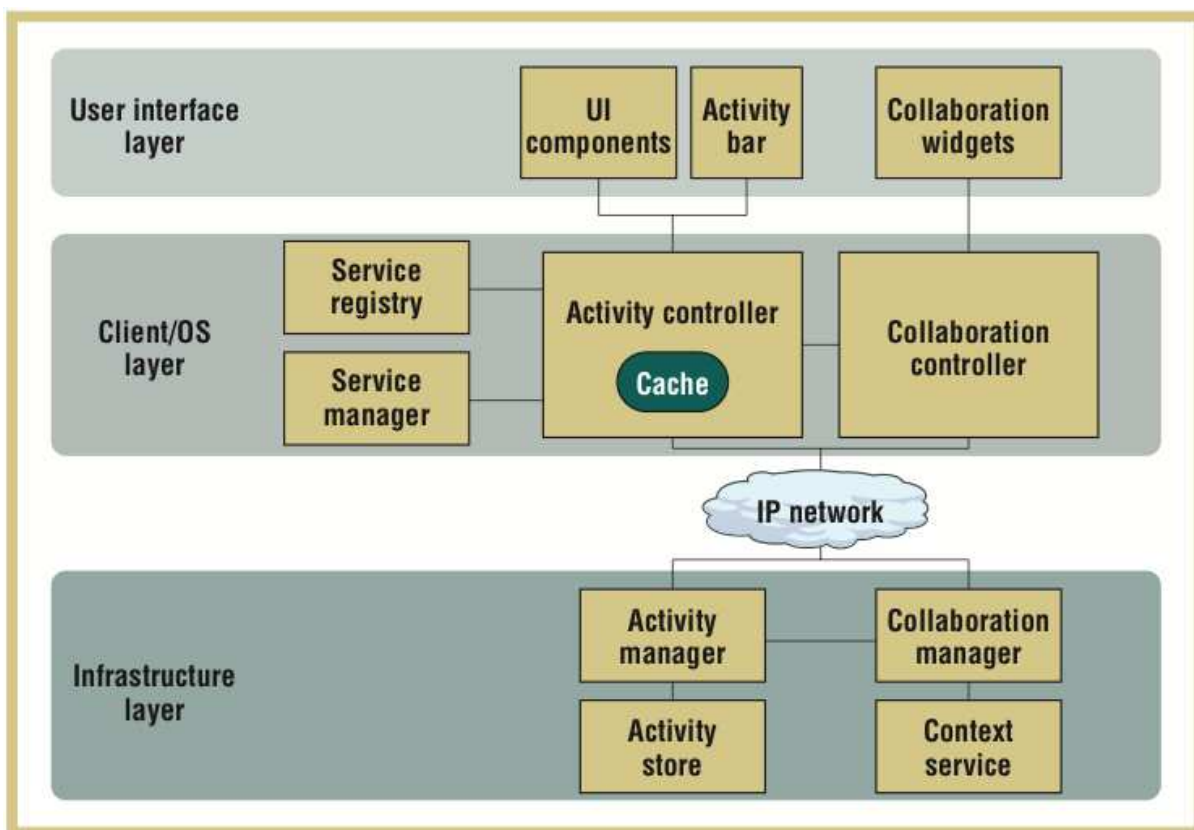


Figura 3.2 – Arquitetura do ABC

vSked (YEGANYAN et al., 2010) é um sistema interativo de itinerário visual criado para auxiliar crianças portadoras de autismo. Esse projeto foi motivado pelo fato de artefatos visuais reduzirem os sintomas associados às deficiências cognitivas, de comunicação e sociais.

Via um ambiente colaborativo composto por uma interface de visualização geral, disposta na sala de aula, e pelos dispositivos móveis dos alunos, vSked permite que atividades sejam criadas e disponibilizadas aos mesmos. Essas atividades são sempre associadas às imagens que facilitam o processo de perguntas e respostas. Sempre que um aluno indica corretamente a imagem que corresponde ao enunciado apresentado, espécies de “recompensas virtuais” são apresentadas, motivando-os a dar continuidade às atividades. Uma visão geral da arquitetura abstrata do vSked é apresentada na Figura 3.3.

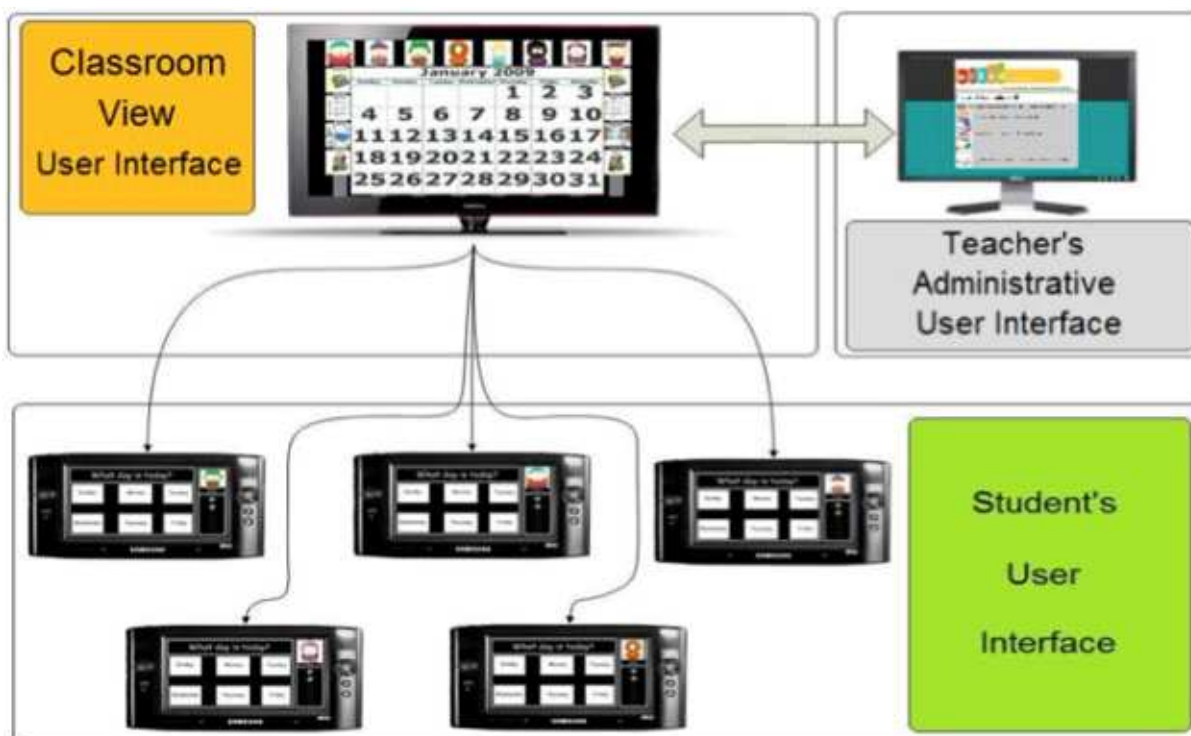


Figura 3.3 – Arquitetura do vSked

Os resultados coletados pelos autores mostram que o sistema pode promover a independência dos estudantes, reduzir a necessidade de intervenção dos educadores e o tempo necessário para a transição de uma atividade a outra. vSked permite a documentação e geração automática de relatórios, que indicam as metas e progressos. Um novo sistema, que visa a auxiliar crianças e adolescentes autistas no aprendizado das habilidades sociais, está sendo desenvolvido.

Designing a Pervasive Healthcare Assistive Environment for the Elderly (MADEIRA et al., 2010) propõe um ambiente de Cuidado de Saúde Pervasivo que fornece informações, via interfaces amigáveis com o usuário, provenientes de sensores que medem parâmetros fisiológicos. Como forma de automonitoramento, alertas simples e diretos, relacionados a possíveis riscos verificados nas medições, são apresentados em dispositivos embarcados no ambiente, melhorando a qualidade de vida dos idosos e aumentando a sua independência. Outro aspecto importante é a possibilidade de disponibilização das informações para que os cuidadores de saúde possam acessá-las remotamente.

Três tipos de usuários estão relacionados com o uso do sistema: os idosos em diferentes contextos (e.g., sobre a cama, fazendo uso de bengalas, fazendo uso de cadeiras de rodas); um observador que acompanha o idoso no ambiente (e.g.,

um familiar, um enfermeiro); e o médico, também visto como um observador que pode tomar decisões baseadas nos dados coletados. Usualmente esse médico encontra-se fora do ambiente, mas pode acessar os dados remotamente e monitorar seu paciente. A Figura 3.4 apresenta a arquitetura abstrata do ambiente.

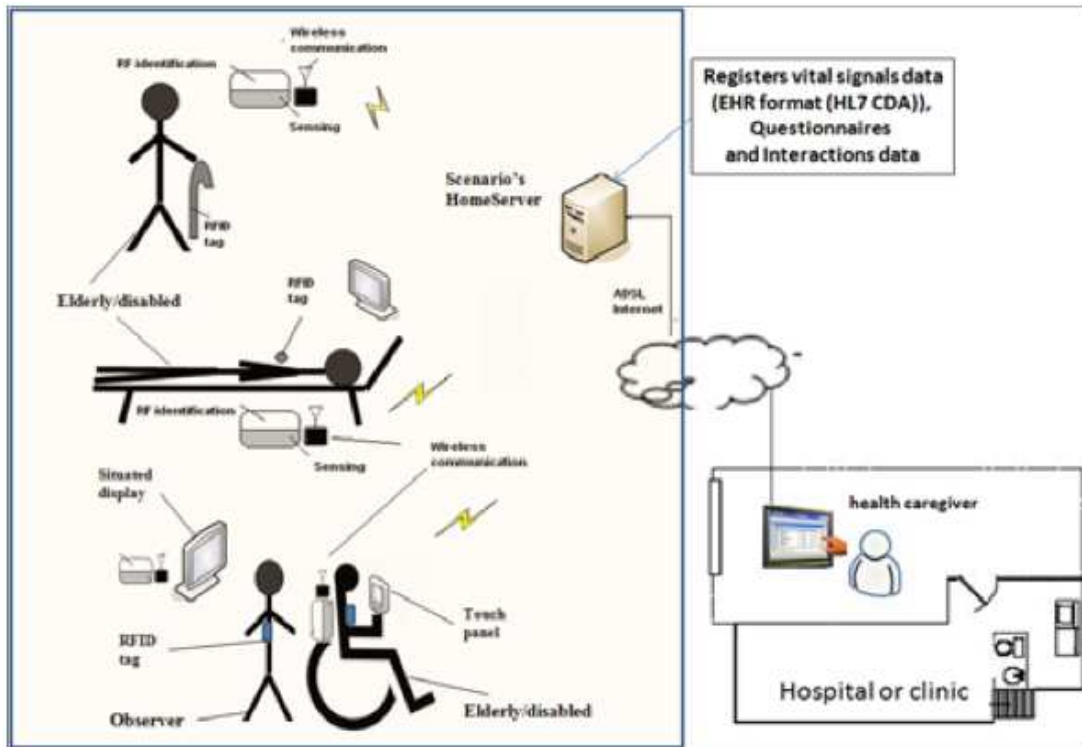


Figura 3.4 – Arquitetura do Ambiente

Capítulo 4

HEALTH LEVEL SEVEN

Além de representar o comitê, a sigla HL7 referencia os padrões estabelecidos pelo mesmo, sendo que o termo *Level Seven* está relacionado ao modelo *Open System Interconnection (OSI)* da *International Organization for Standardization (ISO)*, mais particularmente à camada de aplicação desse modelo.

O objetivo principal do comitê foi o estabelecimento de um protocolo capaz de transportar informações referentes à admissão, alta e transferência de pacientes. A primeira versão do HL7 surgiu em 1987, como consequência de um estudo, e não chegou a ser implementada (SOUZA, 2008). Em 1988 foi publicado o HL7 v2.0, o qual já apresentava uma gama maior de funcionalidades, permitindo o intercâmbio de ordens e relatórios de exames e tratamentos. Nos anos subsequentes surgiram novas versões, cada qual trazendo atualizações que permitiam cobrir novas áreas do Cuidado de Saúde. A primeira versão largamente empregada foi a 2.1, publicada em 1991 (BENSON, 2009). Em junho de 1994 o comitê HL7 foi reconhecido, pelo *American National Standards Institute (ANSI)*, como uma organização de desenvolvimento de padrões. A Figura 4.1 ilustra a evolução desse comitê e desse padrão ao longo dos anos.

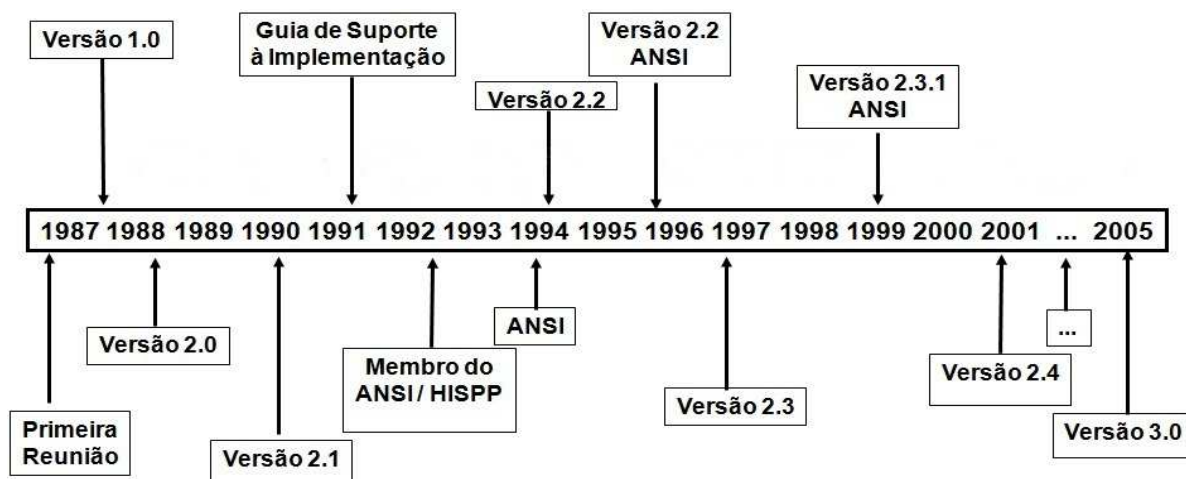


Figura 4.1 – Histórico do HL7

Mesmo sendo inicialmente uma entidade e um padrão exclusivamente norte-americanos, o HL7 foi difundido em diversos países (e.g., Canadá, Inglaterra, Alemanha, Japão, Austrália, África do Sul, Brasil), tornando-se o padrão para interoperabilidade entre SIHs mais usado em todo mundo e presente em mais de 90% dos hospitais norte-americanos (BENSON, 2009). A Figura 4.2 ilustra as relações entre o comitê HL7 e as principais organizações de padronização, em particular com o *Technical Committee 251 (TC)* do *European Committee for Standardization (CEN)* (CEN, 2009), que tem direcionado esforços na harmonização de seu padrão CEN 13606, para o intercâmbio de informações de saúde, com o HL7. Em junho de 2009 o ISO/TC215 (TC215, 2009), responsável pela padronização de informações de saúde e tecnologias de comunicação, reconheceu o HL7 v2.5 como um padrão internacional de protocolo de aplicação para a troca de dados em ambientes de Cuidado de Saúde. Em janeiro de 2010 o comitê anunciou a mudança de seu nome oficial *Health Level Seven* para *Health Level Seven International*.

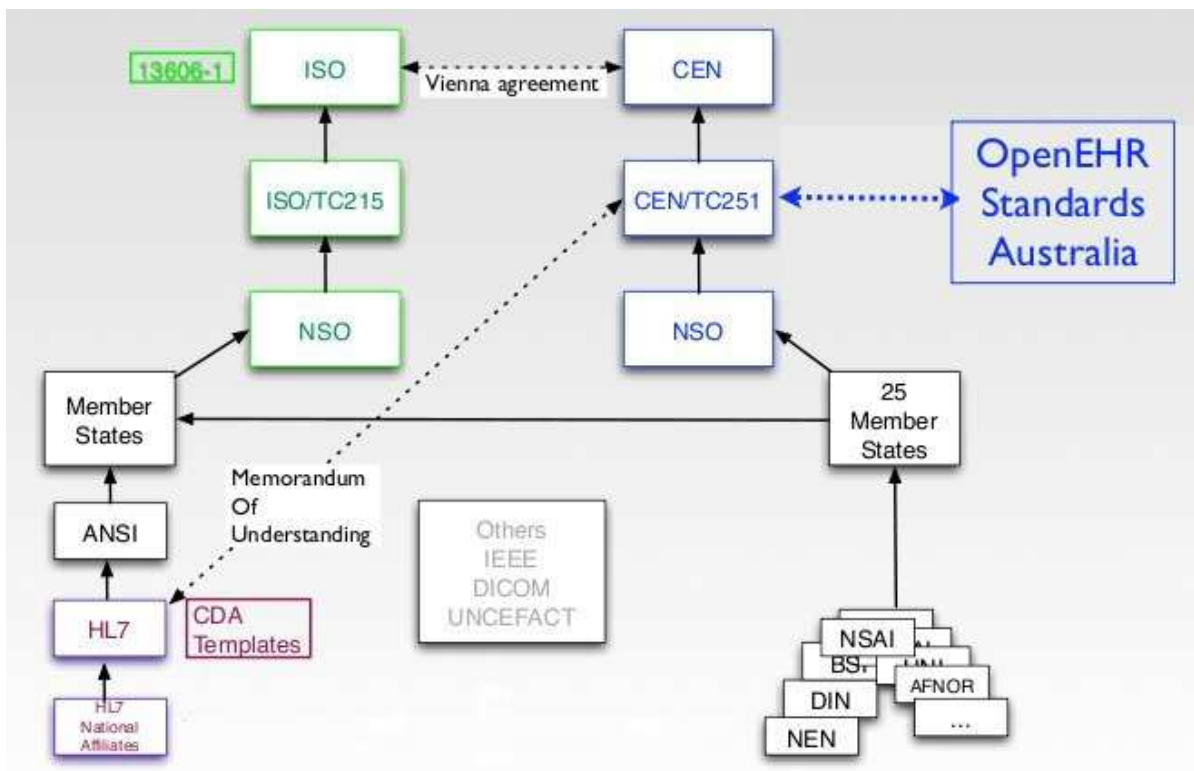


Figura 4.2 – HL7 e organizações de padronização

Essa ampla aceitação do HL7 deve-se principalmente à família v2.x, flexível e que elimina a necessidade de desenvolvimento de módulos de comunicação totalmente customizados, o que reduz drasticamente o custo. Embora flexibilidade seja uma característica chave para a aceitação e adoção desse padrão, também é a sua maior deficiência (NEOTOOL, 2007), pois possibilita o desenvolvimento de implementações com base em partes do mesmo, dificultando a interoperabilidade semântica entre essas implementações. Outra deficiência dessa família é a estruturação das mensagens, que dificulta a sua legibilidade (interoperabilidade funcional), conforme pode ser observado na Figura 4.3 (HL7, 2010).

```

MSH|^~\&|GHH LAB|ELAB-3|GHH OE|BLDG4|200202150930||ORU^R01
|CNTRL-3456|P|2.4<cr>
PID|||555-44-4444||EVERYWOMAN^EVE^E^L|JONES|
196203520|F|||153 FERNWOOD DR.^STATESVILLE^OH^35292||
(206)3345232|(206)752-121|||AC555444444||67-A4335^OH^20030520<cr>
OBR|1|845439^GHH OE|1045813^GHH LAB|1554-5^GLUCOSE|||200202150730|||
555-55-5555^PRIMARY^PATRICIA P^MD^LEVEL SEVEN HEALTHCARE, INC.
|||||F|||||444-44-4444^HIPPOCRATES^HOWARD H^MD<cr>
OBX|1|SN|1554-5^GLUCOSE^POST 12H CFST:MCNC:PT:SER/PLAS:QN||^182|mg/dl|
70_105|H|||F<cr>
    
```

Figura 4.3 – Exemplo de mensagem em HL7

Uma mensagem HL7 é criada como resposta a eventos. As mensagens HL7 v2.x são arquivos texto divididos em segmentos, que respeitam uma ordem pré-determinada (e.g., cabeçalho - MSH, evento – EVN, identificação do paciente - PID), os segmentos estão subdivididos em campos, também devidamente ordenados, cada campo possui um determinado tipo de dado, e cada dado pode ser subdividido em componentes. Para distinguir cada uma dessas divisões, a especificação define os delimitadores que devem ser empregados em cada caso. A Figura 4.4 (BENSON, 2009) apresenta uma visão geral dos principais conceitos que formam a família v2.x do protocolo.

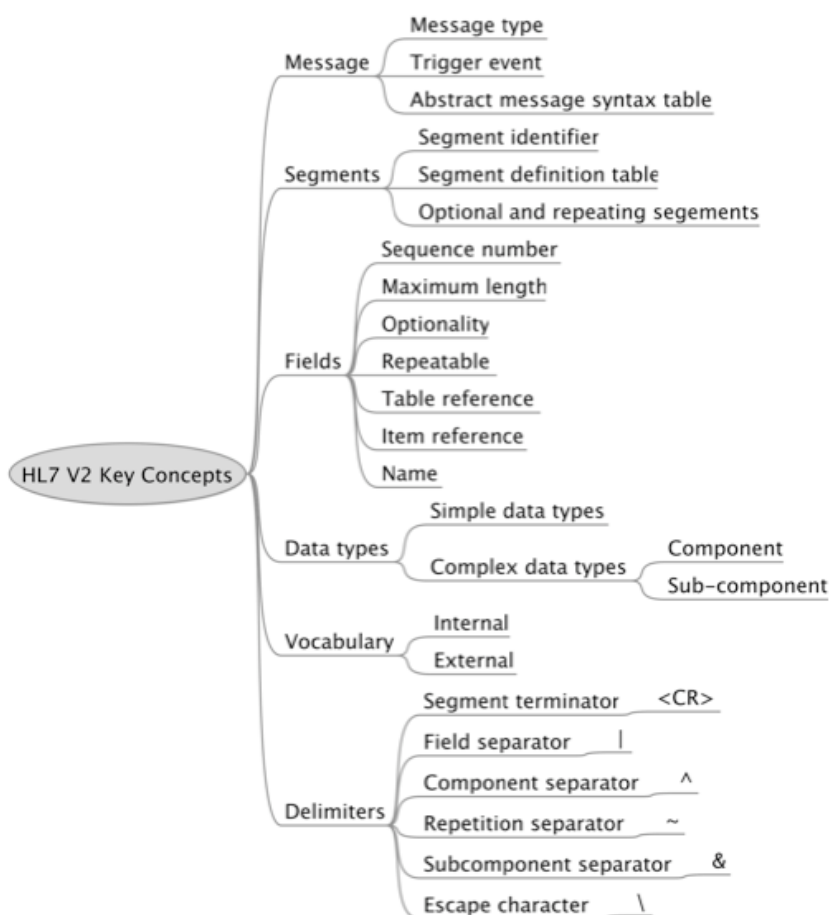


Figura 4.4 – Principais conceitos do HL7 v2.x

4.1 HL7 v3

Os trabalhos relativos à terceira família do padrão tiveram início em 1992 com o estabelecimento da *HL7 Version 3 Task Force* (BENSON, 2009), sendo que o foco principal foi a busca de uma solução para o excesso de flexibilidade da família v2.x. Para melhor descrever as motivações do comitê ao propor essa terceira família, segue parte da explicação relatada no site do comitê (HL7, 2011a):

Oferecendo grande flexibilidade, a serie V2.x de mensagens é largamente implementada e bem sucedida. Essas mensagens pertenceram há muitos anos a uma abordagem "bottom-up" que tratou necessidades individuais através de uma metodologia "ad-hoc". Não existe uma visão coerente dos dados que o HL7 transporta, bem como dos relacionamentos entre os dados.

O sucesso do HL7 [Versão 2] é amplamente atribuído à sua flexibilidade. Este padrão contém vários elementos e segmentos de dados opcionais, tornando-o adaptável em quase todos os locais. No entanto, essa grande flexibilidade dificulta a realização de testes de conformidade nas implementações, exigindo que os desenvolvedores gastem mais tempo analisando e planejando suas interfaces, para garantir que ambas as partes estejam usando os mesmos recursos opcionais.

A Versão 3 trata essas e outras questões através do emprego de uma metodologia bem definida, baseada em um modelo de informação de referência (dados). É o padrão mais definitivo até o momento. Utilizando técnicas rigorosas para a construção de mensagens e incorporando mais eventos e formatos de mensagens com pouca flexibilidade, o objetivo principal do HL7 para a Versão 3 é oferecer um padrão preciso e testável, provendo a habilidade de certificar a conformidade das implementações.

Em suma, o HL7 v3 foi desenvolvido para tornar-se um padrão mais robusto e para sanar as deficiências percebidas nas especificações das famílias precedentes. A terceira família de padrões do comitê HL7 emprega uma metodologia baseada em Orientação a Objetos (OO) e um *Reference Information Model (RIM)*. Esse modelo,

ilustrado no diagrama de classes *Unified Modeling Language (UML)* da Figura 4.5, provê uma representação explícita dos relacionamentos entre os objetos que constituem o domínio Cuidado de Saúde. Além disso, o padrão emprega *eXtensible Markup Language (XML)* para a troca de mensagens.

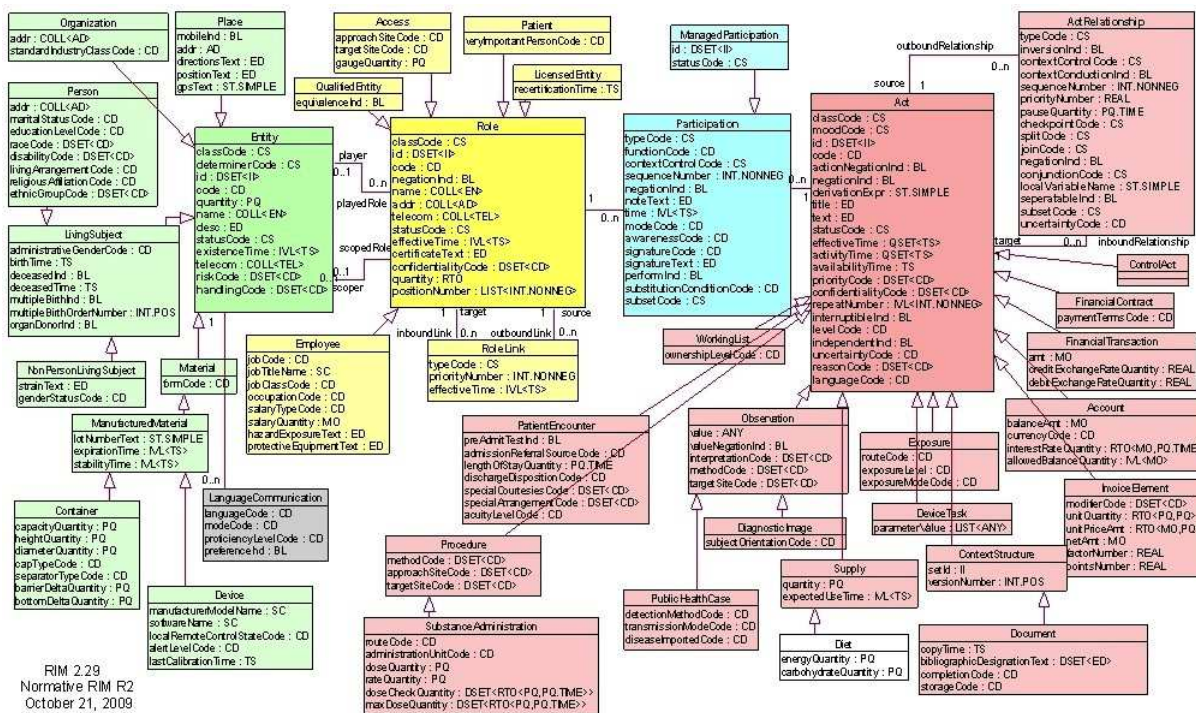


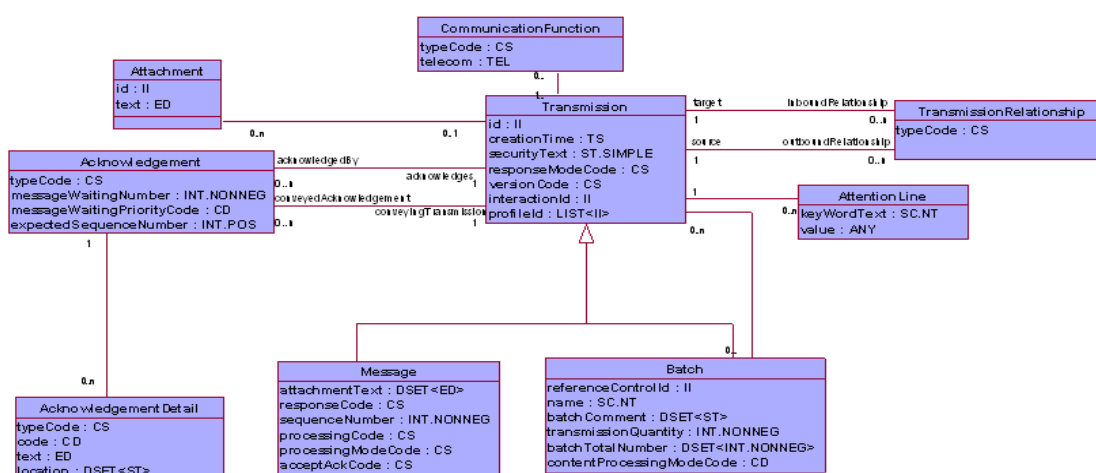
Figura 4.5 – Modelo de Referência do HL7 v3

Embora não represente exatamente um modelo de Cuidado de Saúde, o RIM HL7 foi criado para esse domínio, sendo parte essencial dessa família. Seu modelo especifica os componentes básicos, os relacionamentos permitidos e os tipos de dados usados no desenvolvimento das mensagens. Para compreender o HL7 v3, deve-se primeiro compreender o RIM (BENSON, 2009).

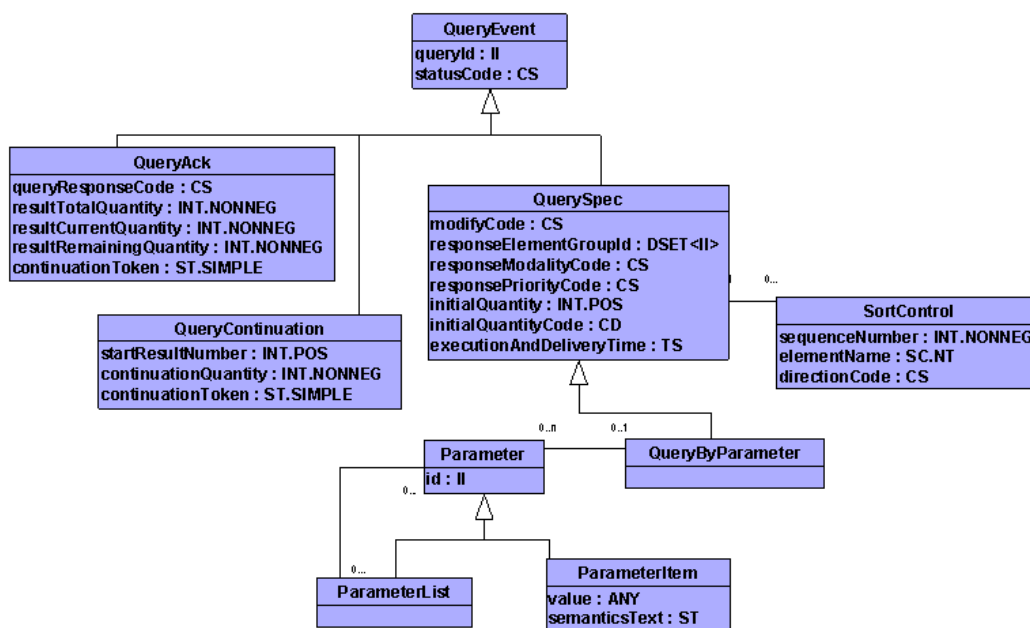
Embora aparentemente simples, o RIM HL7 é dotado de uma certa complexidade. Sua base é formada por três classes principais, **Act**, **Role** e **Entity**, cujos relacionamentos são representados pelas classes **ActRelationship**, **Participation** e **RoleLink** e por diversas especializações. Conforme ilustrado no diagrama da Figura 4.5, essas classes principais estão agrupadas em *Subject Areas*, as quais se distinguem pelas cores e representam as funções das classes no RIM. A *Acts Subject Area*, representada em vermelho, inclui a classe **Act** e suas especializações. Esse grupo está relacionado às ações e eventos que constituem os

serviços de Cuidado de Saúde. A *Entities Subject Area*, representada em verde, inclui a classe Entity, suas especializações e outras classes relacionadas. Este grupo representa os *stakeholders* e demais objetos de interesse do Cuidado de Saúde. Finalmente, a *Roles Subject Area*, representada em amarelo, inclui a classe Role e suas especializações. Esse grupo foca nos papéis que os participantes podem desempenhar nas interações do Cuidado de Saúde (HL7, 2011b). Toda mensagem HL7 v3 é derivada de Act, a qual pode ser associada a várias Participation, onde cada uma é associada a uma única Role, que por sua vez pode estar associada a uma ou nenhuma Entity. Portanto, toda interação é uma especialização de Act (e.g., **Observation**), sendo necessário identificar o ator que a executou (e.g., **Organization**) e os objetos ou sujeitos que foram afetados pela mesma (e.g., **Patient**).

Existe ainda uma quarta Subject Area, denominada *Communication Infrastructure Subject Area*, que é subdividida em *MessageControl Subject Area* e *QueryControl Subject Area* e está ilustrada na Figura 4.6. Essas *Subject Areas* estão relacionadas às definições técnicas e ao controle das comunicações em HL7, sendo que a *MessageControl Subject Area* (Figura 4.6(a)) contém os elementos do RIM envolvidos no controle, comunicação e reconhecimento de mensagens. A *QueryControl Subject Area* (Figura 4.6(b)) contém as classes necessárias para a formulação, comunicação e resposta às mensagens de consulta (HL7, 2011b).



(a) MessageControl subject area



(b) QueryControl Subject Area

Figura 4.6 – Communication Infrastruct Subject Area

4.1.1 Atributos Estruturais

A fim de criar um modelo menos complexo para o RIM (Figura 4.5), o comitê HL7 empregou Atributos Estruturais para representar as mais de cem classes originais via um diagrama simplificado formado por seis classes principais. Esses atributos dão significado às classes, dependendo da sua função em uma mensagem. Um atributo estrutural, presente em quase todas as classes do RIM, é o **classCode** o qual, por exemplo, pode definir: que a Act usada representa uma observação; que a Role trata-se de um paciente; e que a Entity especificada na mensagem é uma pessoa (BENSON, 2009). O conjunto completo de atributos estruturais está listado na Tabela 4.1.

Uma vez definidos os valores dos atributos estruturais durante a criação das mensagens, estes permanecem inalterados em todo o ciclo de vida de uso das mesmas. Tais atributos auxiliam os sistemas de informação a compreenderem os dados transmitidos, permitindo que sejam processados e usados corretamente.

De uma forma geral, o verdadeiro significado semântico das classes é definido pelos valores dos atributos estruturais. Os nomes das classes usados no diagrama UML da Figura 4.5 não possuem significado real no intercâmbio das

mensagens, sendo que esse significado é definido pela combinação dos atributos estruturais em cada uma dessas classes.

Tabela 4.1 – Conjunto de atributos estruturais

Classes	Atributos Estruturais
Act	classCode, moodCode, negationInd, levelCode
Entity	classCode, determinerCode
Role	classCode, negationInd
ActRelationship	typeCode, inversionInd, contextControlCode, ContextConductionInd, negationInd
Participation	typeCode, contextControlCode
RoleLink	typeCode

4.1.2 Tipos de Dados

Os tipos de dados são outros componentes importantes na atribuição de significado semântico às mensagens e às informações trocadas. A comunicação em ambientes de Cuidado de Saúde é demasiadamente complexa devido, por exemplo, à existência de diferentes e variados conceitos, unidades e terminologias, o que torna necessário o conhecimento da definição dos valores transportados. Por este motivo, todo elemento em HL7 v3 possui seu tipo de dado.

Para melhor representar a estruturação e definição desses valores, o comitê HL7 possui seu próprio conjunto de tipos de dados. Segundo o próprio comitê, os tipos de dados usados, pelas diferentes linguagens de programação, não são suficientemente poderosos para a representação das informações nesse domínio de aplicação. Porém, um número de tipos de dados empregados em diferentes linguagens de programação, como BASIC, Pascal, C, C++ e Java, foram usados como entrada para a especificação do conjunto de tipos de dados do padrão HL7 (HL7, 2011b). A Tabela 4.2 lista os principais tipos de dados especificados pelo comitê.

Tabela 4.2 – Principais tipos de dados do HL7 v3

Nome	Símbolo	Descrição
Bag	BAG	Uma coleção desordenada de valores, sendo que cada valor pode aparecer mais de uma vez na coleção.
Boolean	BL	Corresponde aos valores lógicos.
Character String	ST	Corresponde a dados do tipo texto. É usado para nomes, símbolos e expressões formais.
Concept Descriptor	CD	Representa qualquer conceito, normalmente atribuindo um código definido em um sistema de codificação. Pode conter a descrição original do conceito, bem como uma ou mais traduções em diferentes sistemas de codificação.
Coded Simple Value	CS	Tipo de dado codificado no qual o código e o sistema de codificação são definidos e fixados pelo contexto no qual o valor ocorre. O código CS não pode ser pré-determinado.
Coded with Equivalents	CE	Consiste de um valor codificado e, opcionalmente, valores codificados em outros sistemas de codificação que identificam o mesmo conceito.
Data Value	ANY	Define as propriedades básicas de todos os valores de dados. Este é um tipo abstrato, significando que nenhum valor pode ser simplesmente um ANY sem pertencer a um tipo de dado concreto.
Encapsulated Data	ED	Corresponde a dados compreendidos por humanos ou para futuro processamento computacional fora do escopo do HL7. Tais como conteúdo multimídia, documentos XML, etc.
Entity Name	EN	Nome para uma pessoa, organização, lugar, etc.
Instance Identifier	II	Identificador único. Exemplos são identificadores para objetos do RIM, números de registros médicos, identificadores de ordens, etc.
Integer Number	INT	Representa os números inteiros.
Person Name	PN	EN usada quando a especialização Person é instanciada.
Point in Time	TS	Quantidade que especifica um ponto na linha natural do tempo, como por exemplo as datas.
Real Number	REAL	Representa os números reais.
Sequence	LIST	Lista ordenada de valores.
Set	SET	Um conjunto desordenado de valores.
Telecommunication Address	TEL	Representa um endereçador para um recurso mediado por equipamentos de telecomunicação. Os endereçadores são especificados como um <i>Universal Resource Locator (URL)</i> .

Os atributos das classes e seus tipos de dados geram as *tags* XML das mensagens HL7 v3. A mensagem da família HL7 v2.x, exibida na Figura 4.3, seria estruturada em HL7 v3 como o exemplo da Figura 4.7 (HL7, 2010). A raiz de uma mensagem HL7 v3 é formada por um objeto **Message**, o qual define os responsáveis pelo envio e recebimento da mesma em termos de aplicações e instituições. A aplicação responsável pelo recebimento da mensagem é a GHH_OE, descrita na *tag* *executedByRcvApp*, e a aplicação que está enviando a mensagem é a GHH_LAB, descrita na *tag* *executedBySendApp*. Este elemento raiz encapsula todo *payload* da mensagem, que é o seu evento de controle (*tag* *has_payload_ControlActEvent*). O evento de controle da mensagem define as responsabilidades sobre os dados transmitidos (no caso da mensagem exemplo é o médico responsável pelo resultado do exame), que tipos de informação estão contidos na mensagem, bem como serve de invólucro para o corpo da mesma (região na qual os dados estão efetivamente listados). O conteúdo do domínio (corpo da mensagem) define o tipo específico de observação, seu identificador, em que momento essa observação ocorreu e seus resultados. Além disso, as informações do paciente também estão contidas nessa seção.

Para definir a organização dessas *tags* nas mensagens, bem como para definir uma tarefa específica para a mesma, faz-se necessário um processo de desenvolvimento. Durante esse processo são realizados refinamentos, onde o desenvolvedor seleciona as classes, atributos, relacionamentos e cardinalidades, para que a mensagem, ou conjunto de mensagens, possa suprir as necessidades do domínio alvo de atuação.

```

<?xml version="1.0"encoding="utf-8"standalone="no"?>
<Message xmlns="urn:hl7-org:v3"
  xmlns:xsi="http://www.w3.org/2002/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org:v3/MCCI.MT000101 MCCI.MT000101.xsd">
[... ]
  <executedBySendApp>
    <type_cd code="SND"/>
    <telecom value="127.127.127.255"/>
    <servedBy>
      <id extension="GHH LAB"root="2.16.840.1.113883.1122"/>
      <nm use="L">
        <given>An Entity Name</given>
      </nm>
      <telecom value="555-555-2005"use="H"/>
[... ]
    </servedBy>
  </executedBySendApp>
[... ]
  <has_payload.ControlActEvent xsi:type="MCAI.HD700200">
    <response_cd code="N"/>
    <verifier>
      <participant.COCT.MT090100>
        <id root="2.16.840.1.113883.1122"extension="444-444-4444"/>
      </participant.COCT.MT090100>
    </verifier>
    <interactionTarget xsi:type="POLB.MT004101">
      <ObservationEvent>
        <id root="2.16.840.1.113883.1122"extension="1045813"
          assigningAuthorityName="GHH LAB"/>
        <cd code="1554-5"codeSystemName="LN"displayName="GLUCOSE`POST 12H
          CFST:MCNC:PT:SER/PLAS:QN"/>
        <status_cd code="completed"/>
        <effective_time>
          <dt:center value="2002-02-15T07:30:00"/>
        </effective_time>
[... ]
        <value xsi:type="dt:PQ"value="182"unit="mg/dL"/>
        <interpretation_cd code="H"/>
        <patient>
[... ]
        </patient>
[... ]
        <referenceRange>
          <referenceObservationEventCriterion>
            <value xsi:type="dt:IVL:PQ">
              <dt:low value="70"unit="mg/dL"/>
              <dt:high value="105"unit="mg/dL"/>
            </value>
          </referenceObservationEventCriterion>
        </referenceRange>
      </ObservationEvent>
    </interactionTarget>
  </has_payload.ControlActEvent>
</Message>

```

Figura 4.7 – Exemplo de mensagem em HL7 v3

4.1.3 Processo de Desenvolvimento

Graças ao estilo abstrato e à aptidão para o uso de vocabulários externos do RIM HL7 na especificação das mensagens, o processo de desenvolvimento do HL7 v3 está centrado exclusivamente no refinamento do RIM, o que permite obter um modelo de informação específico do domínio em questão.

A obtenção de subconjuntos de especificações, via aplicação de restrições e refinamentos em modelos genéricos, é amplamente aceita nos padrões. Em especial nos padrões de comunicação, a ideia é que tal característica facilite o alcance da interoperabilidade, uma vez que não são aceitos “caminhos alternativos” e que todas

as mensagens trocadas são extraídas de um mesmo subconjunto do modelo de referência, respeitando estritamente a especificação.

Uma visão geral do processo de desenvolvimento de uma mensagem HL7 v3, proposto pelo comitê HL7, é apresentada na Figura 4.8. Esse processo consiste basicamente de quatro etapas principais: Especificar Casos de Uso; Especificar Modelo de Interação; Especificar Modelo de Informação; e Criar Descrição da Mensagem.

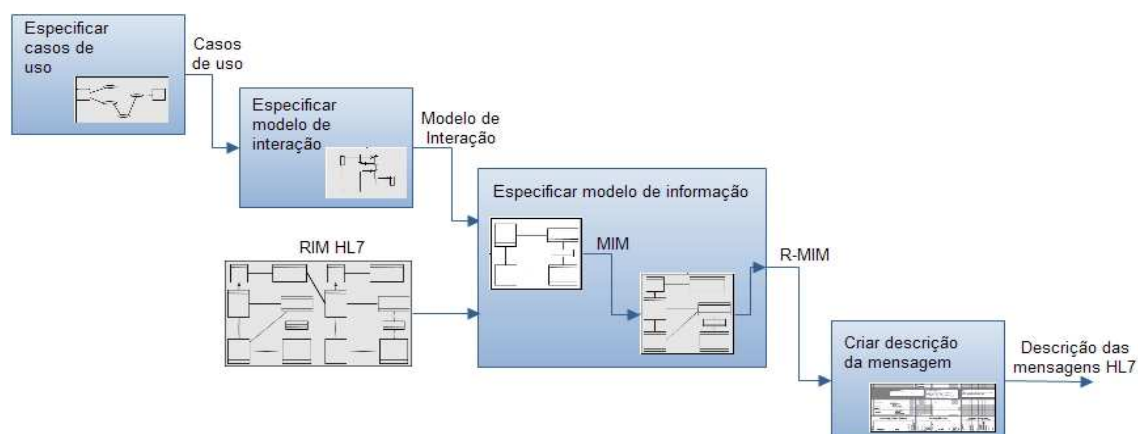


Figura 4.8 – Processo de desenvolvimento do HL7 v3

Na etapa **Especificação de Casos de Uso** são identificados os requisitos do domínio com foco nas mensagens (e.g., gerenciamento de pacientes, administração financeira), representando-os via diagramas de casos de uso. São identificadas as condições nas quais as mensagens são criadas, os atores que atuam nessas mensagens e outras informações pertinentes. Na etapa **Especificação do Modelo de Interação** os diagramas de casos de uso são detalhados nos diagramas de sequência, os quais especificam os fluxos das informações, as regras de comunicação e os eventos que disparam as mensagens. Na etapa **Especificação do Modelo de Informação** o RIM é especificado, conforme as especificações constantes nos diagramas de casos de uso e de sequência. Nessa etapa é inicialmente extraído um subconjunto do RIM, de forma a atender aos casos de uso elaborados, gerando o *Message Information Model (MIM)* e mantendo as especificações originais das classes. Em seguida o MIM é refinado, conforme as especificações dos casos de usos e respectivos diagramas de sequência, gerando um modelo de classes denominado *Refined-Message Information Model (R-MIM)*. No R-MIM tem-se a eliminação de atributos desnecessários, a adição de

cardinalidade aos relacionamentos e a especialização das classes. Finalmente, na etapa **Criar Descrição da Mensagem** é elaborada a descrição da mensagem HL7 com base nas especificações do R-MIM. Essa descrição define a estrutura das mensagens HL7 para o domínio alvo, a partir da qual as mensagens são criadas.

A Figura 4.9 apresenta um exemplo simples de R-MIM (HL7, 2011b). Este R-MIM retrata uma observação, representada pelo ponto de entrada **ObservationEvent**, nome designado a qualquer classe Act com os atributos estruturais classCode = OBS (observação) e moodCode = EVN (evento). O modelo exemplificado possui um ou mais **InvestigationEvents**, os quais possibilitam a captura e o uso de informações (e.g., provenientes de dispositivos, de reações sofridas pelo paciente mediante a administração de um medicamento), e as participações **Patient** e **Agent**. O sujeito Patient está associado a um nome, proveniente da classe **Person**, e a uma organização (classe **Organization**). A combinação do código do paciente com o código da organização gera uma identificação única para o primeiro. O autor Agent também possui um identificador único e pode representar um médico, um técnico ou um equipamento.

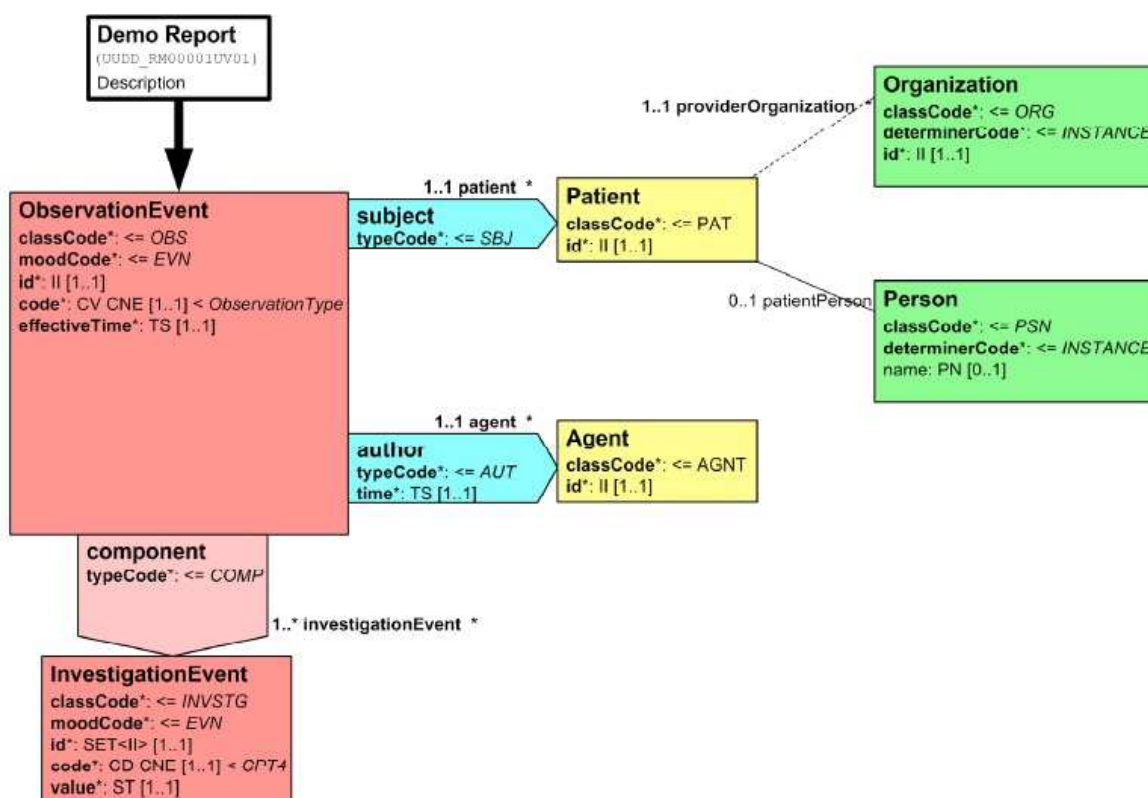


Figura 4.9 – Exemplo de R-MIM

4.1.4 Clinical Document Architecture

A terceira família do HL7 incorporou a *Clinical Document Architecture (CDA)* (DOLIN et al., 2006, 2009), uma especificação para a criação de documentos eletrônicos para o Cuidado de Saúde, a qual é a mais adotada do HL7 v3 no cenário mundial (BENSON, 2009). A CDA é um padrão que especifica a estrutura e a semântica de “documentos clínicos”², visando o seu intercâmbio e provendo interoperabilidade funcional.

Embora derivados do mesmo RIM, documentos HL7 são diferentes de mensagens HL7. Enquanto as mensagens tendem a ser temporárias, baseadas em eventos e não persistentes, os documentos são persistentes, completos, clinicamente certificados e visam principalmente à legibilidade por humanos, podendo ser transmitidos em mensagens HL7 ou qualquer outra forma de comunicação (DOLIN et al., 2006). Entretanto, mesmo com todas essas características a CDA apresenta inconsistências, que dificultam a representação do conhecimento (BROWNE, 2008).

²Um documento clínico é composto de toda documentação das observações clínicas e dos serviços.

Capítulo 5

ARQUÉTIPOS

Há tempo que a estratégia adotada pelo comitê HL7 para a representação de conceitos clínicos tem sido questionada. A estrutura de conhecimento para o domínio Cuidado de Saúde provida pelo RIM HL7 pode limitar-se a *templates*, muitas vezes uma solução simplista para a representação do conhecimento no nível conceitual (BROWNE, 2008). Devido à sua natureza estática, os *templates* não permitem que variações sejam representadas nos conceitos (e.g., rótulos, unidades de medida, idioma), o que em (BEALE, 2001) é denominado de Problema de Variabilidade. A Figura 5.1 exemplifica esse problema via um *template* para o registro da concentração de glicose sanguínea, utilizado no diagnóstico do diabetes. Neste é definido o rótulo concentração de glicose e uma verificação, via expressão booleana, que permite ao sistema emitir um alerta caso o valor inserido seja superior aos 140 mg/dl, confirmando o diagnóstico de diabetes (SBD, 2010). Já que não é possível estabelecer um padrão único para o registro da concentração de glicose, devido aos diversos contextos socioculturais envolvidos, variações nesse rótulo são comuns, tais como: glicose sanguínea; *glucose concentration*; *blood glucose*; *concentración de glucosa*.

Para lidar com esse problema e ampliar o significado semântico na representação de conceitos clínicos, o uso de arquétipos tem sido proposto (BEALE, 2001). Arquétipos são pedaços de conhecimento, que indicam como representar conceitos ou informações de um domínio via expressões computáveis. Tais expressões baseiam-se num modelo de referência e são definidas na forma de restrições estruturais (BEALE; HEARD, 2007a). Assim como no brinquedo LEGO, onde uma peça isolada não possui significado mas a combinação correta de várias

peças outorga um significado, os arquétipos buscam expressar um conceito de um domínio via a combinação de diferentes informações.

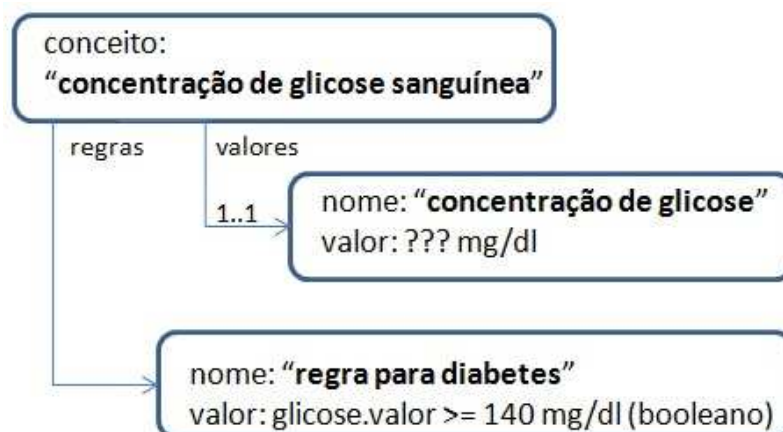


Figura 5.1 – *Template* para o registro da concentração de glicose

A Figura 5.2 (OPENEHR, 2011a) mostra parte da interface do arquétipo para o registro da concentração de glicose sanguínea, onde a definição do conceito compreende informações como a unidade de medida usada, condições nas quais o exame foi realizado (e.g., em jejum, após refeição, aleatória), quantidade de glicose administrada (para o caso do Teste Oral de Tolerância à Glicose). Além disso, o arquétipo da Figura 5.2 permite registrar informações adicionais, tais como: idioma usado; detalhes da amostra; representação multimídia do exame; anotações diversas provenientes das observações efetuadas pelo cuidador de saúde.

O que torna o uso de arquétipos uma boa alternativa para representação do conhecimento é a concentração de todo o conceito numa única estrutura. Sempre desenvolvido sob a orientação e auxílio de profissionais especialistas da área, o conceito descrito por um arquétipo pode ser visto como o refinamento da ontologia daquele domínio (BEALE, 2001), trazendo benefícios para a comunicação humana, interoperabilidade ao nível conceitual, enriquecimento do domínio e desenvolvimento de sistemas inteligentes.

The screenshot shows the 'openEHR-OBSEVATION.lab_test-blood_glucose.v1' form. It is organized into several sections:

- Header:** Definition | Terminology | Display | Interface | Description |
- History:** Any event (dropdown)
- Test name:** Free text
- Diagnostic service:** Free text
- Test status:** (dropdown)
- Slot: Specimen:** Blood glucose: 0,00 (input), mg/dl (dropdown), mmol/l (dropdown), Result: [Any] (dropdown), mg/dl (dropdown)
- Slot: Per-result:** Overall interpretation: Free text, Multimedia representation: (dropdown)
- Protocol:**
 - Test identification:** Placer order identifier: Free text, Filler order identifier: Free text, Laboratory test result identifier: Free text
 - Slot: Laboratory:** Datetime result issued: 2010 [yyyy], 1 [mm], 1 [dd], 0 [hh], 0 [mm], 0 [ss]
- State:**
 - Diet:** Intake: Random (dropdown), Duration: 0 (input), Description: Free text
 - Glucose challenge:** Dose: 0,00 (input) gm, Route: Oral (dropdown), Timing: Post-challenge (dropdown), Duration: 0 (input)
 - Insulin challenge:** Dose: 0,00 (input) U, Route: Subcutaneous (dropdown), Timing: Post-challenge (dropdown), Duration: 0 (input)
 - Event:** Event name: Free text, Timing: Post-challenge (dropdown), Duration: 0 (input)

Figura 5.2 – Arquétipo para o registro da concentração de glicose

5.1 Fundação openEHR

Fundada como uma parceria entre a *Ocean Informatics* (OCEAN, 2011) e a *University College of London (UCL)*, a *openEHR* (OPENEHR, 2011b) é uma fundação dedicada à pesquisa e ao desenvolvimento de RESs interoperáveis, que emprega arquétipos na especificação de uma arquitetura livre baseada numa abordagem de duas camadas (BEALE, 2001), a qual separa o modelo de dados (ou modelo de persistência) do modelo de representação do domínio.

O modelo de dados (primeira camada) é simbolizado por um Modelo de Referência (*Reference Model – RM*) composto por classes, que representam estruturas genéricas do RES ou de dados demográficos (FREIRE et al., 2008). O modelo de representação do domínio (segunda camada) é composto por regras que restringem o RM e especificam como os conceitos clínicos são usados no RES. Essas regras são definidas pelos arquétipos e essa camada é representada pelo Modelo de Arquétipo (*Archetype Model - AM*). Sendo o RM mais estável e menos

sujeito a alterações que o AM, essa separação habilita o desenvolvimento de “sistemas à prova de futuro” (*future-proof systems*) (BEALE, 2001), permitindo aos mecanismos de persistência ser pouco alterados e ao AM ser afetado pelas alterações de estrutura e de regras de negócio (FREIRE et al., 2008).

Nesse modelo dual empregado pela *openEHR*, apenas a primeira camada é implementada em software. Dessa forma, os sistemas tornam-se consideravelmente menores, mais sustentáveis e inerentemente auto-adaptativos, já que são construídos para consumir arquétipos, os quais poderão ser desenvolvidos em qualquer estágio de uso dos mesmos. A Figura 5.3 (BEALE; HEARD, 2007b) mostra que o uso dos arquétipos gera novos relacionamentos entre informações e modelos. Um sistema planejado de maneira clássica, com toda a semântica do domínio codificada no software ou banco de dados, estaria limitado do lado esquerdo da figura e seus dados conforme apenas a um modelo de dados. Com o uso do modelo dual, todos os dados em tempo de execução estão relacionados semanticamente aos arquétipos, expressos em *Archetype Definitions Language (ADL)*, bem como concretamente relacionados ao modelo de referência (BEALE; HEARD, 2007b).

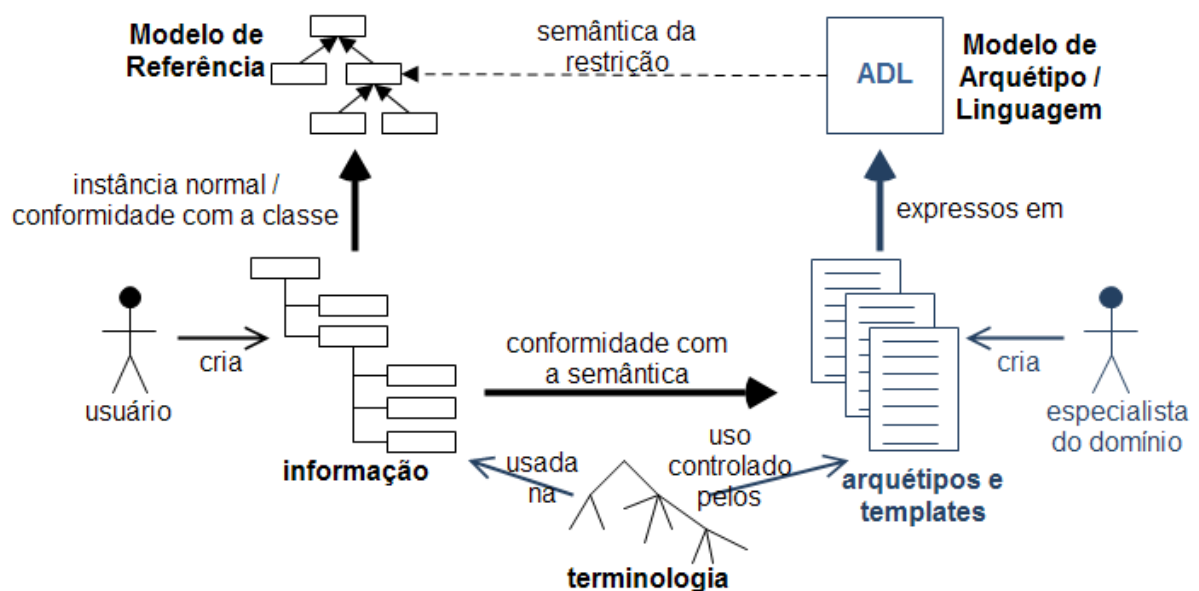


Figura 5.3 – Meta-arquitetura dos arquétipos

5.1.1 Reference Model

Conforme ilustrado na Figura 5.4 e descrito em (BEALE; HEARD, 2007b), o RM pode ser dividido em *core*, *patterns* e *domain*, cada qual com seus respectivos pacotes.

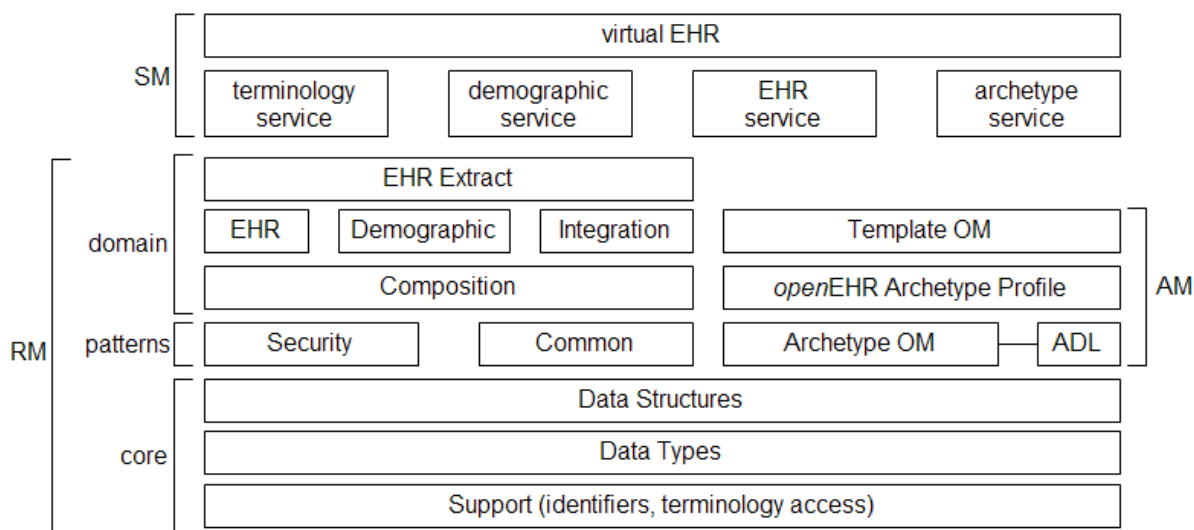


Figura 5.4 – Estrutura openEHR

O **Support Information Model** corresponde aos conceitos mais básicos requeridos por todos os demais pacotes, sendo composto pelos pacotes **definition**, **identification**, **terminology** e **measurement**. A semântica definida nesses pacotes permite que todos os outros modelos usem os identificadores e tenham acesso aos serviços de conhecimento, tais como terminologia e outros dados. Além disso, o pacote **support** compreende os tipos usados pelos modelos da openEHR, inclusive os tipos primitivos definidos por outras entidades. A Figura 5.5 (BEALE et al., 2008d) exhibe a estrutura desse pacote.

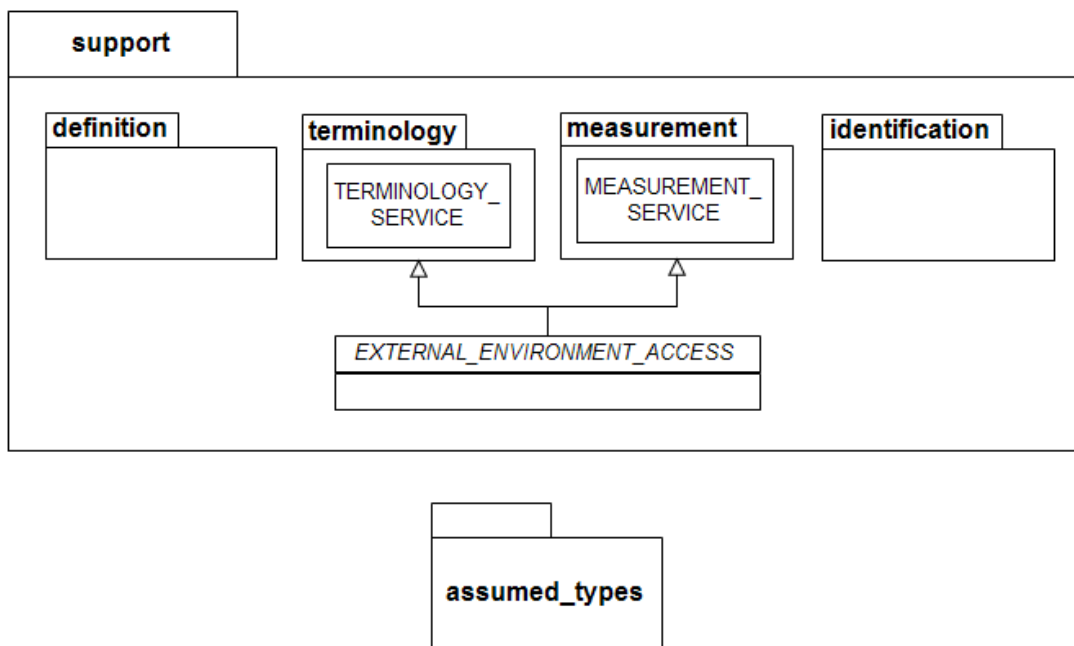


Figura 5.5 – Pacote support

O **Data Types Information Model** define um conjunto de tipos de dados subjacentes a todos os demais modelos, provendo tanto os tipos de dados genéricos quanto os específicos requeridos pelas formas de informação em Saúde. A Tabela 5.1 (BEALE et al., 2008b) lista as categorias de tipos de dados definidas pelo *data types information model* e a Figura 5.6 (BEALE et al., 2008b) exibe a estrutura do pacote.

Tabela 5.1 – Tipos de dados do *openEHR*

Categoria	Descrição
Text	texto simples, texto codificado, parágrafos
Quantities	qualquer tipo ordenado, incluindo valores ordinais, medições com valores e unidades, etc.
Data/Time	data, hora, tipos de data e hora, e tios de data e hora parciais
Encapsulated Data	multimídia, conteúdo processável
Basic Types	boolean, integer, real, etc.

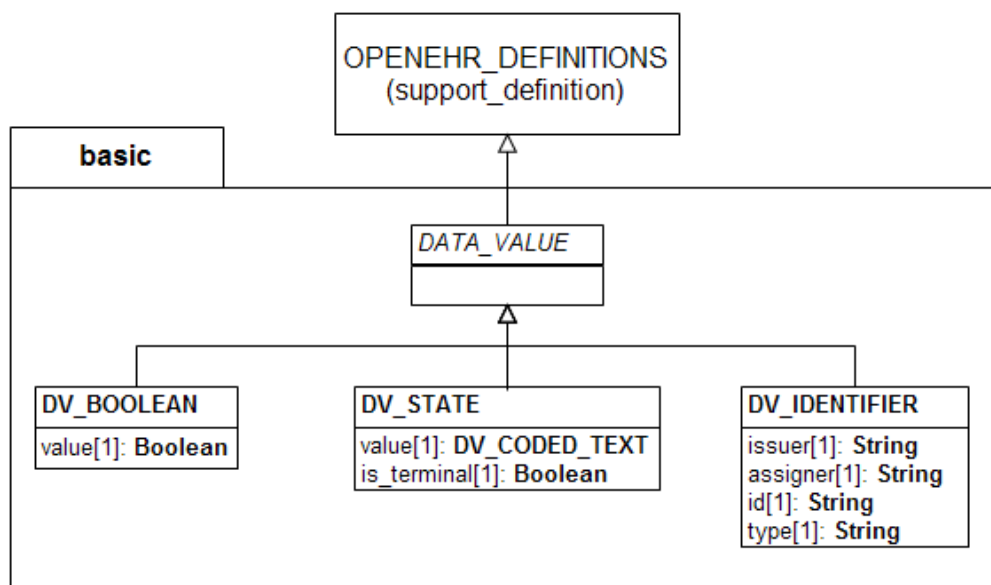


Figura 5.6 – Pacote data_types

O **Data Structures Information Model** é constituído pelas estruturas genéricas, listadas na Tabela 5.2, que são usadas para expressar o conteúdo da estrutura particular definida pelos arquétipos. Essas estruturas pertencem ao pacote **item_structure** exibido na Figura 5.7 (BEALE et al., 2008a). Além desse pacote, **data_structures** também contém o pacote **history**, o qual descreve uma noção genérica de história linear, permitindo o registro de informações ao longo do tempo.

Tabela 5.2 – Estruturas genéricas do openEHR

Estrutura	Descrição
Single	itens simples, utilizado para conter qualquer valor simples, como peso ou altura
List	lista linear de itens nominais, tais como alguns testes patológicos
Table	dado tabular, incluindo tabelas de tamanhos limitados e ilimitados, com colunas nominadas e ordenadas
Tree	dados em forma de árvore que podem ser conceitualmente uma lista de listas, ou outra estrutura de profundidade

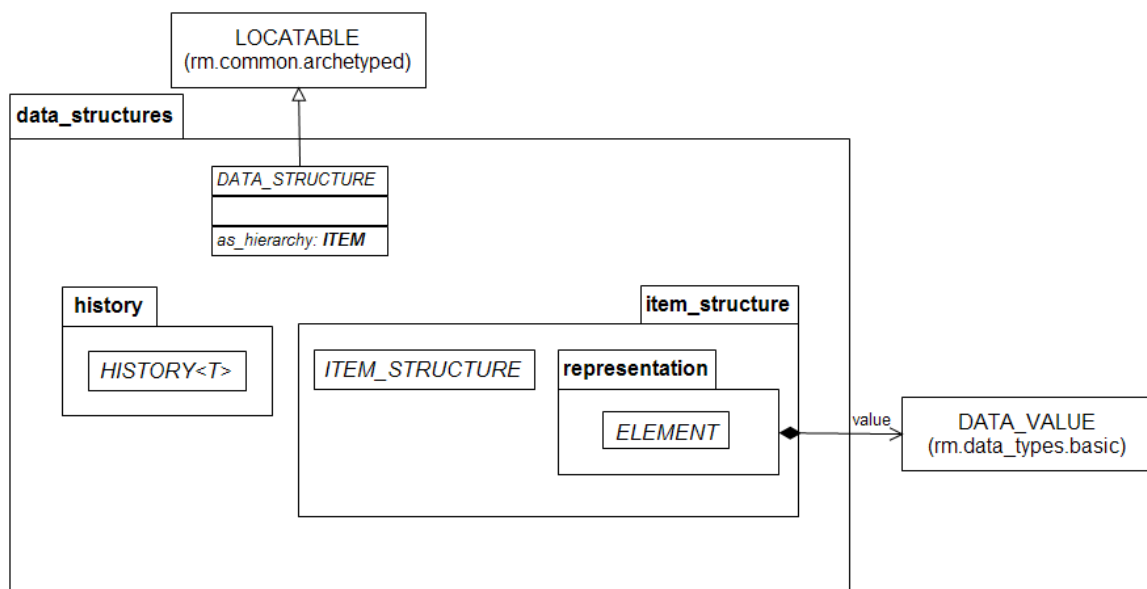


Figura 5.7 – Pacote data_structures

O **Common Information Model**, cuja estrutura está ilustrada na Figura 5.8 (BEALE et al., 2010), compreende vários pacotes contendo conceitos abstratos e padrões de projetos, que são usados em modelos de mais alto nível do *openEHR*.

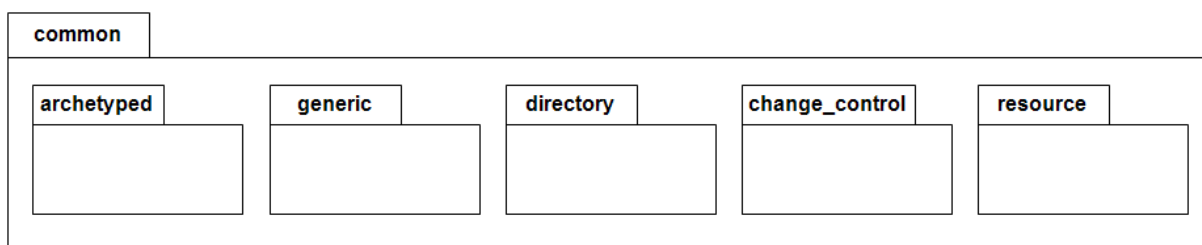


Figura 5.8 – Pacote common

O **archetyped** habilita o emprego do modelo dual da *openEHR*, fazendo a ligação entre os modelos de informação e de arquétipos; o pacote **generic** contém classes que formam “padrões de análise”, os quais são genéricos por todo o domínio e comumente relacionados a referências realizadas a entidades demográficas por outros dados; o pacote **directory** provê uma abstração simples e reutilizável de uma estrutura de pasta com controle de versão; o pacote **change_control** define uma semântica generalizada das alterações de um repositório lógico (e.g., um RES) ao longo do tempo, sendo que cada item no repositório tem sua versão controlada para permitir que o repositório como um todo tenha seu controle de versão garantido; o pacote **resource** define a semântica de um recurso de autoria (e.g., documentos),

suportando traduções em vários idiomas, metadados descritivos e históricos de revisão.

O **Security Information Model** define a semântica de controle de acesso e as configurações de privacidade para a informação no RES. O **EHR Information Model** é ilustrado na Figura 5.9 (BEALE et al., 2008c). O pacote **ehr** contém a estrutura de mais alto nível, que consiste de um objeto **HER_ACCESS**, um objeto **EHR_STATUS**, recipientes de controle de versão de dados na forma de **VERSIONED_COMPOSITION**, opcionalmente indexado por uma hierarquia de diretórios de **FOLDER**. Uma coleção de **CONTRIBUTION**, que documenta as mudanças do RES ao longo do tempo, também é incluída: o pacote **composition** é o recipiente de dados de mais alto nível num RES, sendo descrito pela classe **COMPOSITION**; o pacote **content** contém os pacotes **navigation** e **entry**, sendo que suas classes descrevem a estrutura e a semântica do conteúdo das composições num RES. Uma visão geral do EHR Information Model e seus relacionamentos é ilustrada na Figura 5.10 (BEALE et al., 2008c).

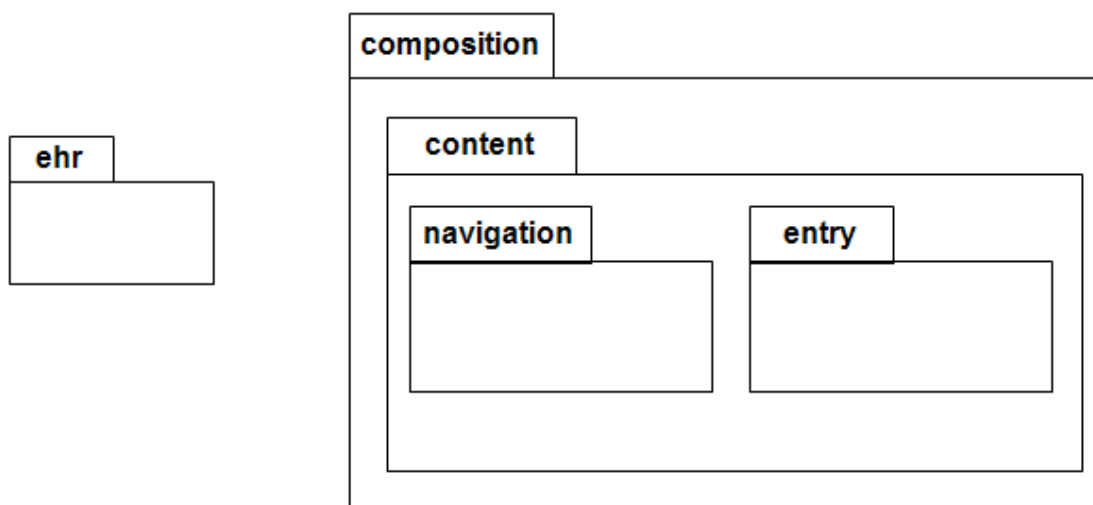


Figura 5.9 – Pacote ehr

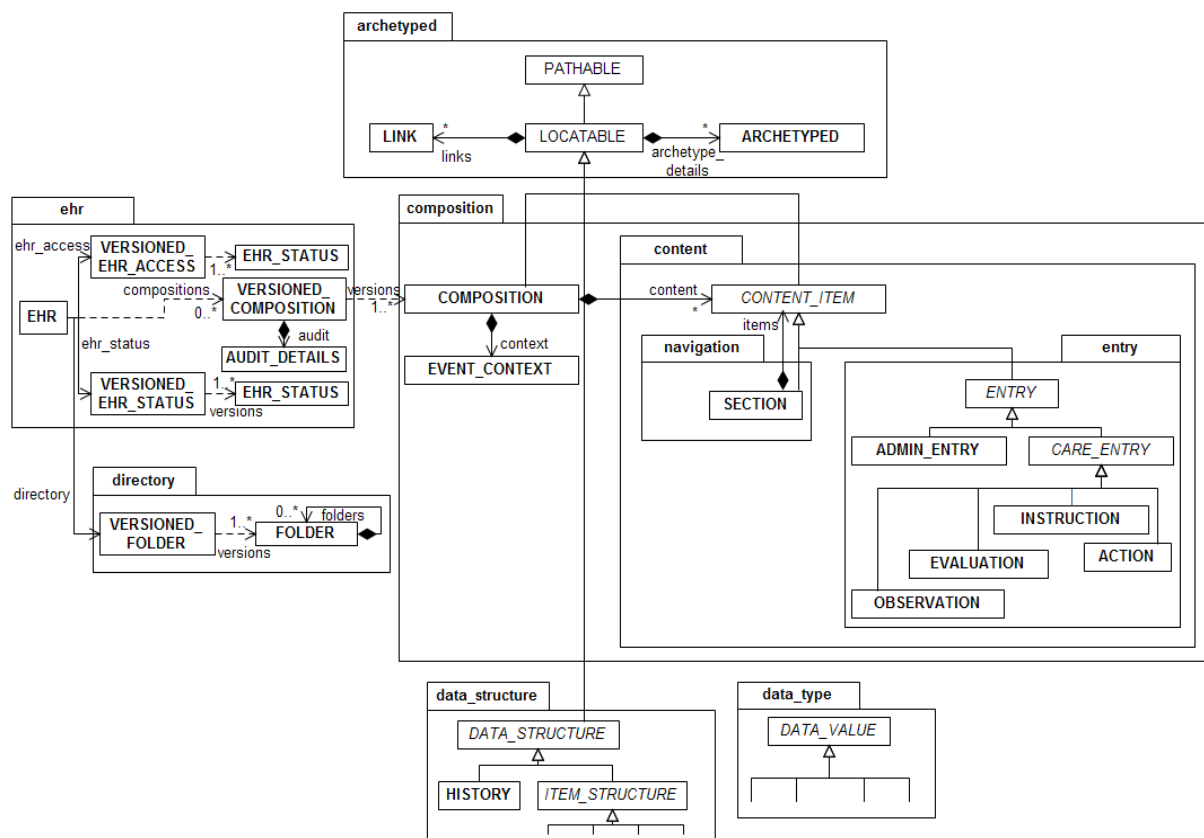


Figura 5.10 – Visão geral do EHR Information Model

O **Integration Information Model** define classes que permitem representar dados externos ou legados numa árvore de dados. Este tipo de entrada de dados possui seu próprio tipo de arquétipos, denominados arquétipos de integração, os quais podem ser usados em conjunto com os arquétipos clínicos em sistemas de integração de dados.

O **Demographics Information Model** define os conceitos genéricos de **PARTY**, **ROLE** e os detalhes relacionados aos endereços de contato. O modelo de arquétipos define a semântica das restrições nas **PARTY**, permitindo que arquétipos para qualquer tipo de pessoa, organização, papel e relacionamento de papéis possam ser descritos.

5.1.2 Archetype Model

O pacote **am** contém os modelos necessários para descrever a semântica dos arquétipos e *templates*³, bem como o seu uso no *openEHR*. Como pode ser observado na Figura 5.11, o Archetype Model inclui a ADL, os pacotes **archetype** e **template**, que definem a semântica orientada a objetos desses arquétipos e *templates*, e o pacote **openehr_profile**, que define um perfil genérico do modelo de arquétipo para o uso no *openEHR* (BEALE; HEARD, 2007b). A divisão ilustrada na Figura 5.4 para o Archetype Model é descrita em (BEALE, 2008).

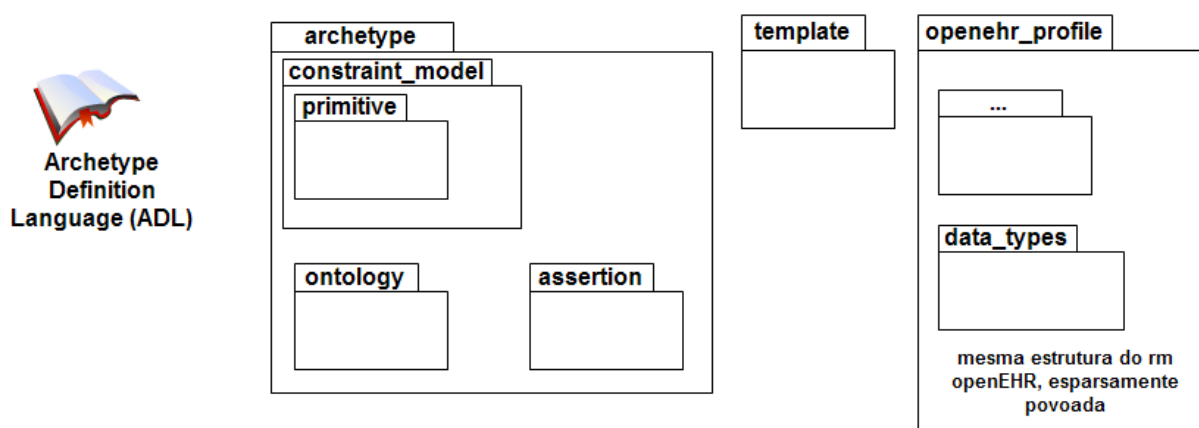


Figura 5.11 – Pacote am

O **Archetype Object Model (AOM)** define em UML o modelo de objetos equivalente aos arquétipos. Sendo um modelo genérico, o AOM pode ser usado para definir arquétipos para qualquer modelo de referência numa forma padronizada. O **openEHR Archetype Profile (oAP)** define classes de arquétipos personalizadas, e suas sintaxes personalizadas equivalentes, que podem ser usadas no refinamento de certas classes do RM. O **Template Object Model (TOM)** define o modelo de objetos para os *templates*, os quais são usados para combinar arquétipos em estruturas de informações específicas e assim suprir as necessidades de variações locais, usualmente correspondendo aos formulários de tela. A **Archetype Definition Language (ADL)** tem o objetivo de prover uma sintaxe abstrata capaz de representar textualmente arquétipos e *templates*.

³ A *openEHR* define o termo *template* de forma particular, considerando-os como arquétipos adaptados às necessidades locais de uma determinada instituição.

5.1.3 Archetype Definition Language

Archetype Definition Language (ADL) (BEALE; HEARD, 2008) é uma linguagem formal desenvolvida para expressar arquétipos a fim de torná-los processáveis. Para descrever as restrições nos dados, que são instâncias de algum modelo de informação, a ADL emprega três sintaxes: dADL para expressar dados; cADL para expressar restrições; e uma versão de lógica de predicados de primeira ordem. Via o uso da ADL, arquétipos podem ser escritos para qualquer domínio onde existam modelos formais que descrevem instâncias de dados.

Arquétipos expressos em ADL lembram códigos de linguagens de programação e possuem uma sintaxe definida. A cADL é usada para expressar a seção de definição de arquétipos, enquanto que a dADL é usada para expressar os dados que aparecem nas seções de idioma, descrição, ontologia e histórico de revisão (BEALE; HEARD, 2008). Uma visão geral da estrutura de uma ADL é apresentada na Figura 5.12.

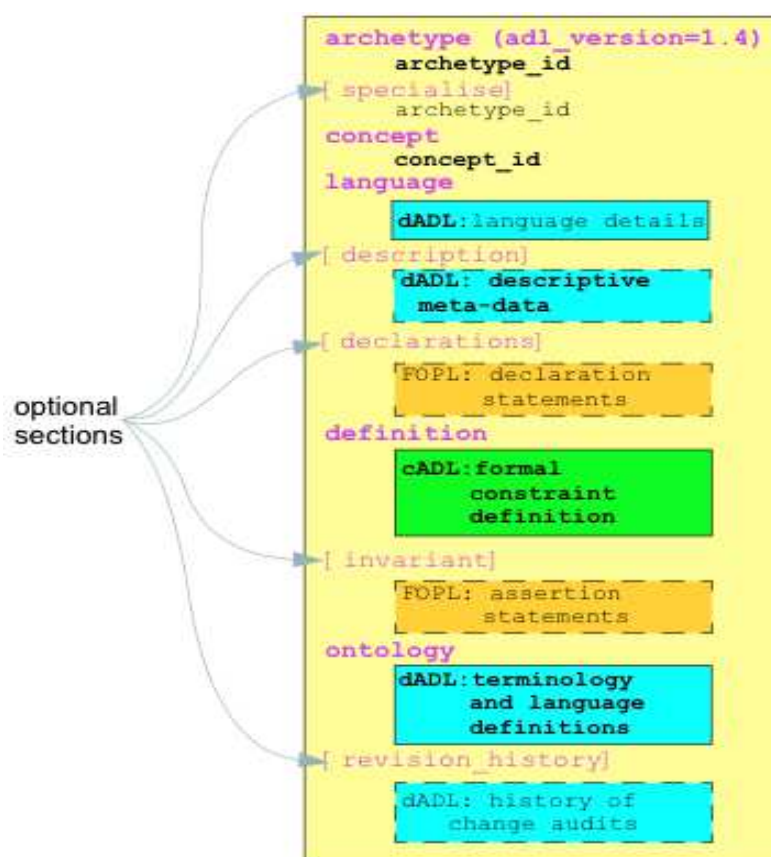


Figura 5.12 – Estrutura da ADL

Com uma sintaxe interpretável, a ADL possui um relacionamento formal com modelos estruturais, tais como os representados pela UML. Enquanto a sintaxe da ADL permanece como o principal formalismo para expressar arquétipos, o AOM estabelece a sua semântica, definindo principalmente os relacionamentos que precisam ser garantidos entre suas partes, para que esse arquétipo seja considerado válido como um todo. A Figura 5.13 (OPENEHR, 2011a) apresenta um trecho da ADL do arquétipo para o registro da concentração de glicose sanguínea.

```

archetype (adl.version=1.4)
  openEHR-EHR-OBSERVATION.lab.test-blood.glucose.v1
  specialise
    openEHR-EHR-OBSERVATION.lab.test.v1
  concept
    [at0000.1] -- Blood glucose
  language
    original_language = <[ISO.639-1::en]>
  [...]
  definition
    OBSERVATION[at0000.1] matches { -- Blood glucose
      data matches {
        HISTORY[at0001] matches { -- Event Series
          events cardinality matches {1..*; unordered} matches {
            EVENT[at0002] occurrences matches {0..1} matches { -- Any event
              data matches {
                ITEM.TREE[at0003] matches { -- Tree
                  items cardinality matches {0..*; unordered} matches {
                    ELEMENT[at0005] occurrences matches {0..1} matches { --
Test name
                                value matches {
                                  DV.TEXT matches {+}
                                }
                              }
                    ELEMENT[at0077] occurrences matches {0..1} matches { --
Diagnostic service
                                value matches {
                                  DV.TEXT matches {+}
                                }
                              }
                    ELEMENT[at0073] occurrences matches {0..1} matches { --
Test status
                                value matches {
                                  DV.CODED.TEXT matches {
                                    defining_code matches {
                                      [local::
                                        at0037, -- Interim
                                        at0038, -- Final
                                        at0039, -- Supplementary
                                        at0040, -- Corrected (amended)
                                        at0074, -- Aborted
                                        at0079] -- Never performed
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  [...]
}
[...]
```

Figura 5.13 – ADL para o registro da concentração de glicose

Capítulo 6

DESENVOLVIMENTO DIRIGIDO A MODELOS

Desde os primórdios da programação que os Engenheiros de Software buscam meios para aumentar o nível de abstração das tarefas associadas a essa atividade (ATKINSON; KÜHNE, 2003), reduzindo assim a complexidade e os problemas provenientes da mesma. Mesmo com os inúmeros avanços dessa área, vários problemas são verificados na forma com a qual o software é desenvolvido na maioria das organizações (KLEPPE; WARMER; BAST, 2003; LUCREDIO, 2009).

Muitos dos artefatos de alto nível (e.g., modelos UML) são empregados para facilitar a análise de um problema, sendo úteis apenas num primeiro momento. Conforme o desenvolvimento avança, esses artefatos vão se tornando obsoletos e deixam de representar fielmente a realidade. Isto ocorre porque normalmente as mudanças e manutenções são efetuadas diretamente no código, fazendo com que os esforços despendidos na construção desses artefatos não sejam aproveitados diretamente na produção de software (BITTAR et al., 2009; LUCREDIO, 2009).

Nesse sentido, o Desenvolvimento Dirigido a Modelos (*Model-Driven Development - MDD*) visa aprimorar a forma com que o software é desenvolvido, via a representação das principais características de um sistema em modelos, os quais são construídos e refinados conforme o sistema é desenvolvido (VOELTER; GROHER, 2007). Em contraste com o desenvolvimento de software não dirigido a modelos, no MDD os modelos não constituem apenas a documentação, sendo processados por ferramentas que permitem suas transformações em diferentes níveis de abstração (VOELTER; GROHER, 2007).

Conforme ilustra a Figura 6.1 (LUCREDIO, 2009), o MDD pode desobrigar o Engenheiro de Software da interação direta com o código fonte, abstraindo o seu trabalho das complexidades inerentes à implementação em diferentes plataformas (FRANCE; RUMPE, 2007; LUCREDIO, 2009). Mecanismos automáticos são responsáveis por efetuar transformações e gerar os códigos a partir dos modelos, fazendo com que estes deixem de ser apenas guias para o processo de desenvolvimento e também façam parte do software (LUCREDIO, 2009).

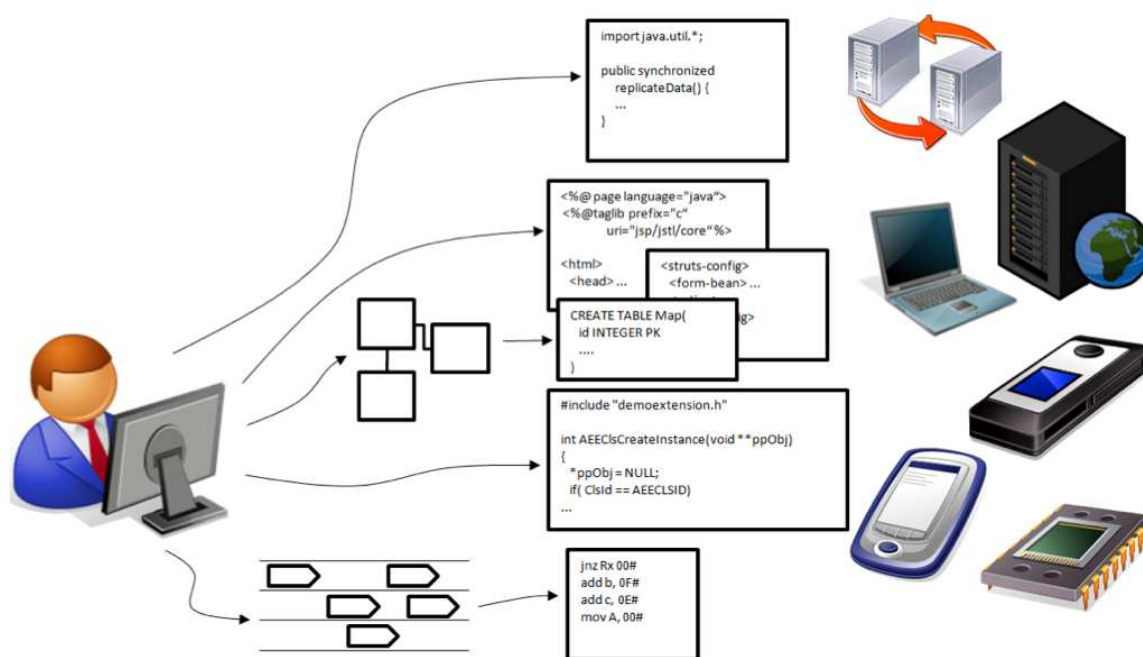


Figura 6.1 – Processo de criação de software no MDD

De acordo com (BAHNOT et al., 2005; DEURSEN; KLINT, 1998; KLEPPE et al., 2003; LUCREDIO, 2009; MERNIK et al., 2005), as principais vantagens do MDD relacionam-se à: **produtividade**, porque as tarefas repetitivas podem ser implementadas nas transformações, poupando tempo e esforço, e o tempo de desenvolvimento é melhor aproveitado, já que é empregado na produção de modelos de mais alto nível; **portabilidade**, porque um mesmo modelo pode gerar código para diferentes plataformas; **grau de integração**, porque diferentes partes do modelo podem ser transformados em códigos para diferentes plataformas, resultando num software que corresponde à heterogeneidade do ambiente sem perder sua funcionalidade global; **manutenção e documentação**, porque as alterações são realizadas diretamente nos modelos, mantendo-os em conformidade

com o código, e a documentação permanece sempre atualizada, facilitando assim a manutenção; **comunicação**, porque os especialistas de domínio podem usar os modelos para identificar os conceitos de negócio, não dependendo diretamente de conhecimento técnico relativo à plataforma; **reutilização**, porque o reuso de artefatos de alto nível traz maiores benefícios que o reuso de código-fonte; **verificação e otimização**, porque os modelos facilitam o emprego de verificadores semânticos e otimizações automáticas, o que pode reduzir a ocorrência de erros semânticos e fornecer implementações mais eficientes; e **correção**, porque as ferramentas usadas nas transformações não geram erros acidentais (e.g., erros de digitação) e erros conceituais podem ser identificados em níveis superiores de abstração.

Essas vantagens do MDD relacionam-se à minimização de tarefas repetitivas para a transformação de modelos em código fonte, as quais passam a ser realizadas com o auxílio de ferramentas, permitindo a sua automação e a manutenção da consistência dos modelos, isso mesmo mediante a realização de atividades de urgência, tais como a correção de erros (LUCREDIO, 2009).

6.1 Principais Elementos do MDD

Conforme ilustra a Figura 6.2 (LUCREDIO, 2009), o uso de MDD está diretamente relacionado aos seguintes elementos: uma ferramenta de modelagem; uma ferramenta para definir transformações; e um mecanismo para executar transformações.

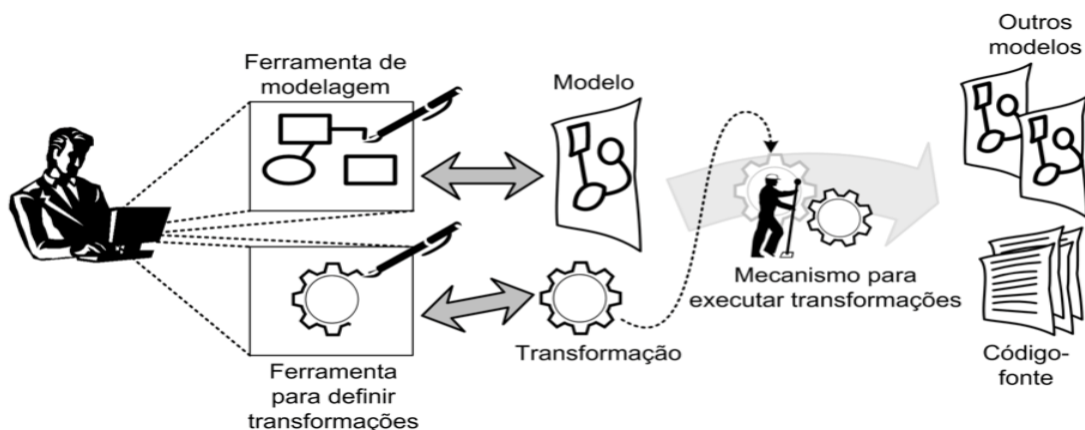


Figura 6.2 – Principais elementos do MDD

Os modelos são construídos com o auxílio da **ferramenta de modelagem**, a qual deve ser intuitiva e de fácil uso, permitindo que os modelos, que descrevem os conceitos do domínio, sejam produzidos facilmente. Além disso, a ferramenta de modelagem deve gerar modelos semanticamente completos e corretos, permitindo assim o processamento por parte dos computadores. Essas características são normalmente implementadas por uma Linguagem Específica de Domínio (DSL). Esses modelos são então submetidos às transformações que os converterão em outros modelos ou código-fonte. Tais transformações são construídas com o auxílio da **ferramenta para definir transformações**, que permite a definição de regras de mapeamento de modelo-para-modelo (M2M) ou modelo-para-código (M2C). O **mecanismo para executar transformações** aplica as transformações, permitindo que um modelo de entrada seja convertido em outro modelo de saída ou código-fonte, de acordo com as regras de mapeamento definidas (LUCREDIO, 2009).

6.2 Metamodelagem

Metamodelagem é empregada em diferentes abordagens do MDD, para garantir que os modelos construídos estejam semanticamente corretos e possibilitar a definição e execução de transformações (LUCREDIO, 2009). Para tal, a OMG (OMG, 2006) propôs a arquitetura em quatro camadas, ilustrada na Figura 6.3 (LIDDLE, 2010) através de um modelo para dados, a qual hoje é a mais comumente empregada.

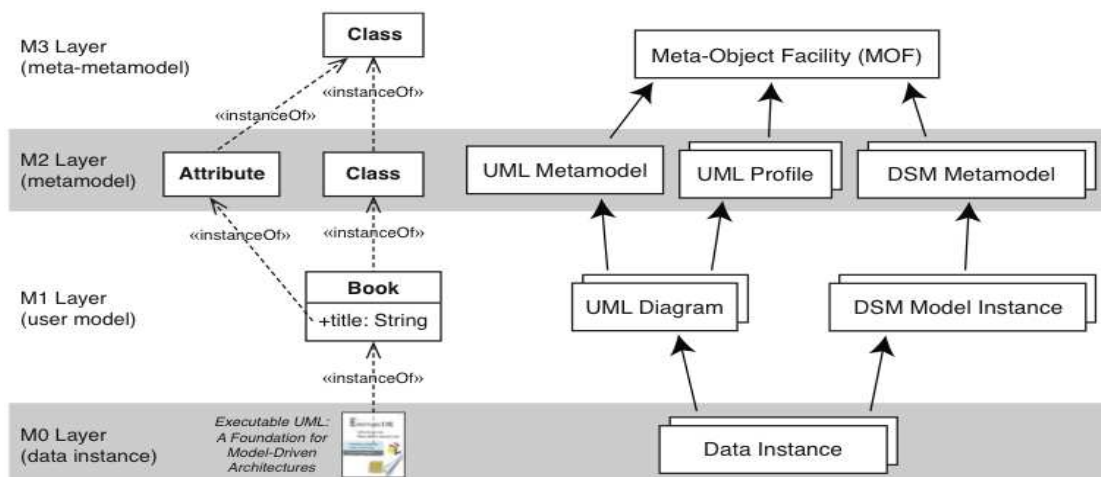


Figura 6.3 – Arquitetura clássica da metamodelagem

A primeira camada (M0) corresponde às instâncias de dados vigentes, sendo que na figura essa instância é representada pelo livro intitulado *Executable UML: A Foundation for Model-Driven Architectures*. Essa instância foi possível graças à segunda camada (M1), que corresponde aos metadados (LUCREDIO, 2009), a qual no exemplo da figura define que um livro é caracterizado pelo seu título.

A terceira camada (M2) é responsável pela definição dos modelos, ou seja, descreve os conceitos que formam uma linguagem de modelagem (LIDDLE, 2010). Por exemplo, a camada M2 da figura define a existência dos elementos classe e atributo usados pela M1. A quarta camada (M3) é responsável pela descrição dos conceitos que aparecem num metamodelo, sendo que não há uma camada superior a esta, pois normalmente um meta-metamodelo é uma instância de si próprio (LUCREDIO, 2009).

6.3 Linguagens Específicas de Domínio

Uma DSL é uma linguagem projetada para ser útil a um conjunto específico de tarefas num dado domínio (SADILEK, 2008; GRONBACK, 2009). Esse domínio refere-se a uma determinada área de competência e conhecimento, que possui terminologia e conceitos próprios e, no caso de desenvolvimento de sistemas, geralmente é chamado domínio do problema, já que trata do problema para o qual uma solução computacional está sendo buscada (LUCREDIO, 2009). Neste sentido, (DEURSEN; KLINT; VISSER, 2000) define:

Uma linguagem específica de domínio é uma linguagem de programação ou uma linguagem de especificação executável que oferece, através de notações e abstrações apropriadas, poder expressivo com foco em, e usualmente restrito a, um domínio de problema particular.

Uma DSL pode ser definida via um metamodelo, o qual representa o conhecimento do domínio do problema alvo. Por estarem restritas a domínios específicos, as DSLs normalmente são pequenas, consistindo de um conjunto de abstrações e notações, que são próximas dos termos reais conhecidos pelos

especialistas desses domínios (DEURSEN et al., 2000). Dessa forma, as DSLs expressam soluções no idioma e no nível de abstração do domínio do problema, reduzindo os esforços de tradução dos conceitos desse domínio para o domínio da solução (LUCREDIO, 2009).

O emprego de DSLs para a modelagem de aplicações, ao invés de linguagens de modelagem de propósito geral (e.g., UML), possibilita criar modelos mais específicos e completos. Recursos tais como *frameworks*, padrões de projeto e componentes podem ser incluídos nos modelos, criando uma infraestrutura que permite a execução de transformações M2C para a geração de uma maior quantidade de código a partir da modelagem (GREENFIELD et al., 2004).

6.4 MDD no Cuidado de Saúde

Em (OHR; VACLAVIK, 2009) é apresentada uma solução para auxiliar os desenvolvedores na manipulação de mensagens da família HL7 v2.x, baseada no uso de uma DSL programática para dar suporte à criação, envio, interpretação e validação de tais mensagens. É também apresentado o desenvolvimento de um módulo de comunicação, baseado nos perfis propostos pela *Integrating the Healthcare Enterprise (IHE)* (IHE, 2010), uma associação de profissionais de Saúde e indústria que visa aprimorar a forma com que os sistemas computacionais compartilham informações de saúde. Esses perfis baseiam-se em padrões bem estabelecidos, tais como o HL7 v2.x, e focam em necessidades específicas do Cuidado de Saúde. O uso da DSL, no desenvolvimento da interface de comunicação, além de facilitar o processo permitiu a criação de adaptadores que tornam a comunicação, entre os diferentes atores pertencentes aos cenários, menos acoplada, já que mensagens baseadas ou não nos perfis IHE podem ser usadas pelas aplicações. A Figura 6.4 apresenta uma visão dessas comunicações e de como os adaptadores são empregados.

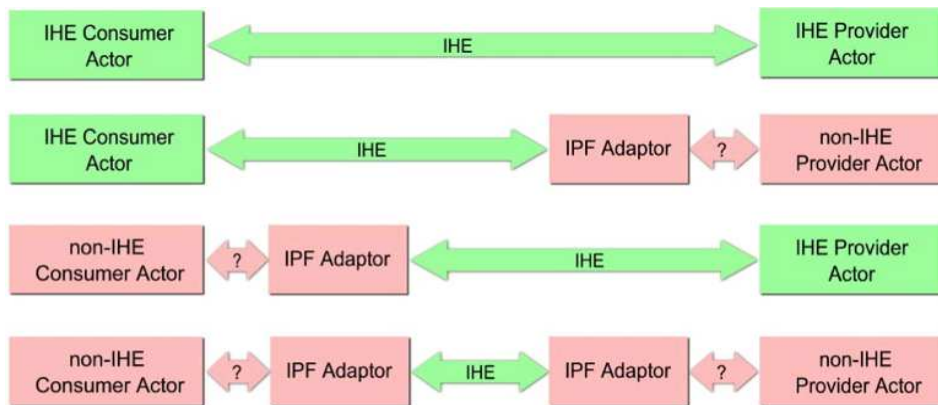


Figura 6.4 – Utilização dos adaptadores

Em (BANFAI et al., 2009) é discutida a implementação de uma ferramenta de modelagem para mensagens HL7 v3, desenvolvida a partir do IDE Eclipse e do *framework* EMF. Para a manipulação dos artefatos, essa ferramenta permite o uso tanto de um editor gráfico quanto textual, baseado num esquema de árvore, através dos quais os modelos são serializados para o padrão *XML Metadata Interchange (XMI)*, visando sua posterior utilização. A arquitetura, empregada nesse trabalho é ilustrada na Figura 6.5. Essa arquitetura foi projetada para ser extensível e possui como principais características: validação dos artefatos gerados, que é importante para reduzir o tempo despendido na busca e correção de erros; e transformação de modelos, que interpreta os esquemas dos artefatos HL7 de entrada e efetua transformações M2M.

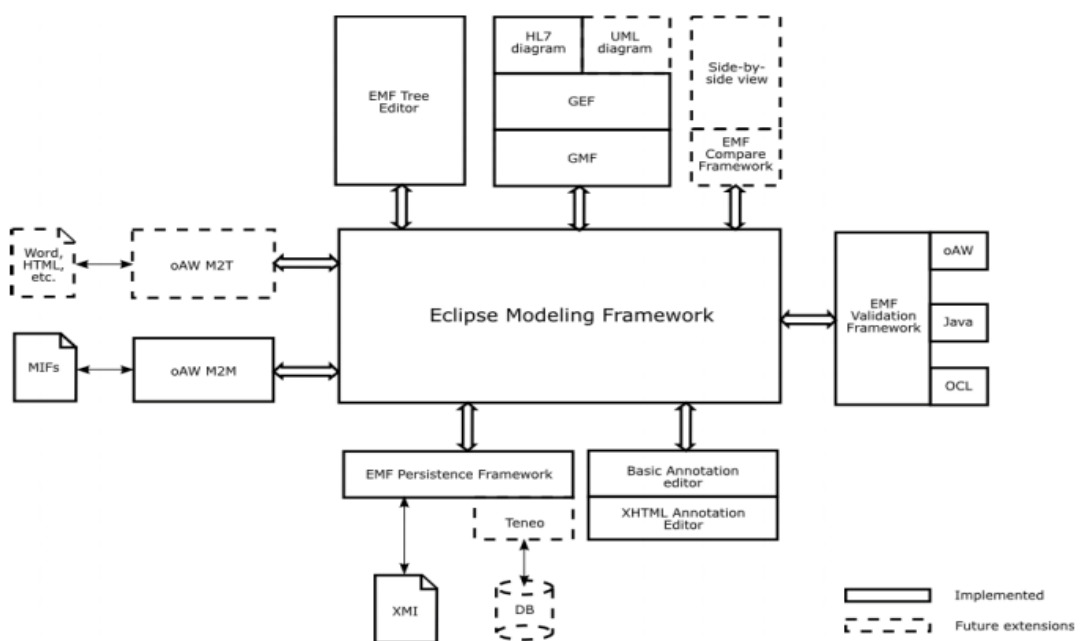


Figura 6.5 – Arquitetura da ferramenta

Em (REGIO et al., 2005) e (REGIO; GREENFIELD, 2005) são apresentadas uma arquitetura e a implementação de uma fábrica de software direcionada ao domínio Cuidado de Saúde, que visa prover os desenvolvedores de um meio mais prático para a especificação, projeto, implementação, teste e implantação de aplicações que fazem uso do protocolo HL7 v3. O objetivo é automatizar a criação de módulos de comunicação, chamados pelos autores de “portas de colaboração”, seguindo estritamente o processo de desenvolvimento das mensagens HL7 v3. A Figura 6.6 (REGIO et al., 2005) apresenta o fluxo das mensagens, que deve ser suportado pelas portas de colaboração geradas pela fábrica de software.

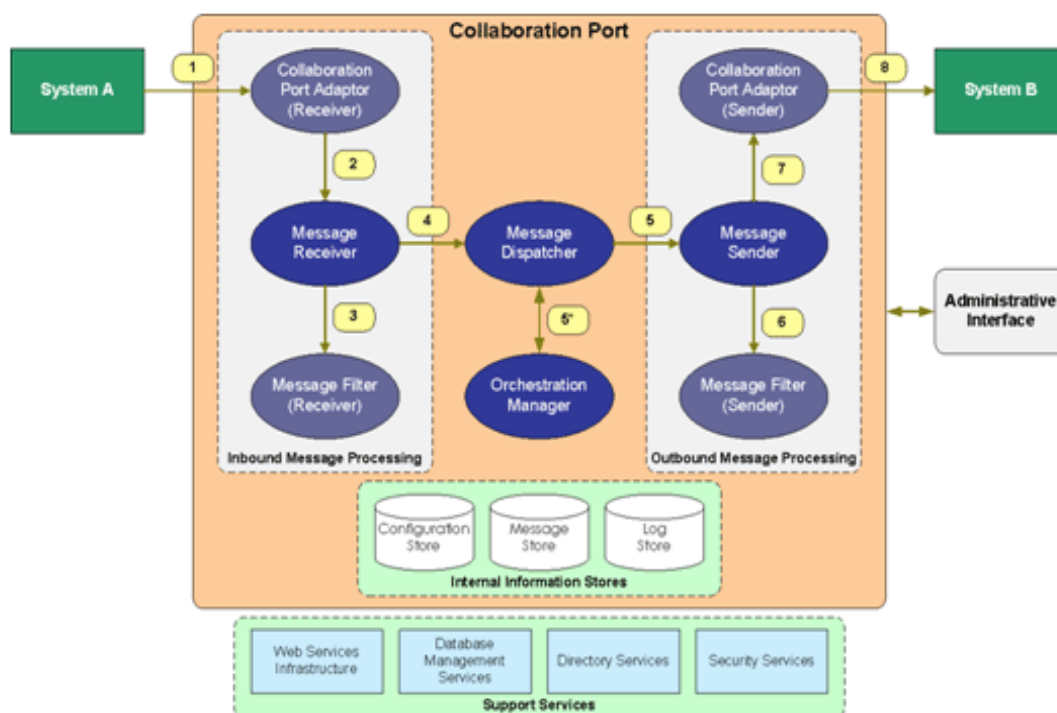


Figura 6.6 – Fluxo das mensagens

No exemplo dessa figura, o Sistema A determina que uma mensagem deve ser enviada ao Sistema B. A mensagem gerada é então colocada numa fila de mensagens e o *Collaboration Port Adaptor (Receiver)* é notificado da existência de uma mensagem para o Sistema B (1). Posteriormente, o *Collaboration Port Adaptor (Receiver)* retira a mensagem da fila e a envia para o *Message Receiver* (2). Com o auxílio do *Message Filter (Receiver)*, o *Message Receiver* gera um objeto para estruturar a mensagem (3). Em seguida, o *Message Receiver* notifica o *Message Dispatcher* do recebimento da mensagem (4). O *Message Dispatcher* encaminha

então o objeto para o *Message Sender* responsável por aquele tipo específico de mensagem (5). Opcionalmente, o *Orchestration Manager* pode ser invocado para efetuar um processamento de orquestração (5*). O *Message Sender* utiliza então o *Message Filter (Sender)* para processar o objeto e gerar o XML correspondente à mensagem (6). Finalmente, a mensagem é encaminhada ao *Collaboration Port Adapter (Sender)* (7), que empacota o XML em uma requisição *Simple Object Access Protocol (SOAP)* e a envia para o Sistema B (8).

Capítulo 7

ABORDAGEM PROPOSTA

Este capítulo apresenta a abordagem proposta para integrar arquétipos a mensagens HL7, visando utilizá-las na comunicação de aplicações ubíquas no Cuidado de Saúde Pervasivo. É importante ressaltar que essa integração ocorre entre as mensagens HL7 v3 e extratos de um determinado Registro Eletrônico de Saúde (RES). Portanto, o emprego das expressões “*criar mensagens HL7 v3 com arquétipos*” e “*enviar mensagens HL7 v3 com arquétipos*” refere-se à criação e ao envio de mensagens HL7 v3 que servem de envelope para extratos de um RES, sendo que estes são organizados conforme o modelo de referência *openEHR*, restringido pela especificação do arquétipo empregado.

A Figura 7.1 ilustra como ocorre o processo de integração e envio dessas mensagens HL7 v3 com extratos de RES. Conforme mencionado no Capítulo 5, os arquétipos indicam como representar conceitos ou informações de um domínio via expressões computáveis, sendo que essas expressões baseiam-se num modelo de referência e são definidas na forma de restrições estruturais. A estrutura do RM *openEHR* é restringida pelo arquétipo empregado, disponibilizando uma estrutura resultante que possui as características do conceito clínico definido pelo arquétipo. Os extratos do RES são atribuídos aos atributos do modelo resultante e envelopados por mensagens HL7 v3.

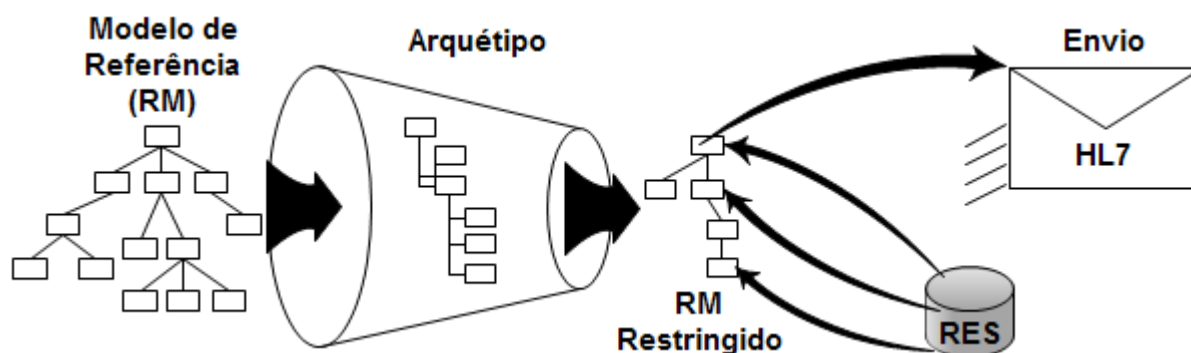


Figura 7.1 – Integração de extratos de RESs a mensagens HL7 v3

Para facilitar a redação e leitura desse texto, o processo descrito na figura acima será sempre referenciado pela expressão “*mensagens HL7 v3 com arquétipos*”. Na sequência deste capítulo são apresentados os objetivos da abordagem, uma visão geral da mesma e o detalhamento de suas etapas.

7.1 Objetivos da Abordagem

O objetivo principal deste trabalho foi prover uma abordagem visando aprimorar a qualidade semântica provida pelas mensagens HL7 via o emprego de conceitos clínicos representados por arquétipos. Essa solução foi apresentada inicialmente em (Browne, 2008) como uma alternativa para melhorar a qualidade da representação do conhecimento em documentos CDA, visto que muitas empresas e comunidades vêm adotando as especificações do comitê HL7 no desenvolvimento de seus sistemas. Uma vez que tanto mensagens HL7 quanto documentos CDA são derivados do mesmo modelo de referência RIM HL7, este trabalho emprega a solução proposta por Eric Browne para a geração de mensagens com maior capacidade de interoperabilidade. Conforme ilustrado na Figura 7.2 (Browne, 2008), a ideia para efetuar a integração sem modificação do RIM foi usar o par de atributos “code-value” disponível na classe **Observation**. Como “code” é responsável pela definição do significado do que é transmitido em “value”, atribui-se a “code” um código específico que permite o pronto reconhecimento do conceito clínico relacionado aos dados restringidos pelo arquétipo, os quais são serializados como XML e transmitidos em “value”.

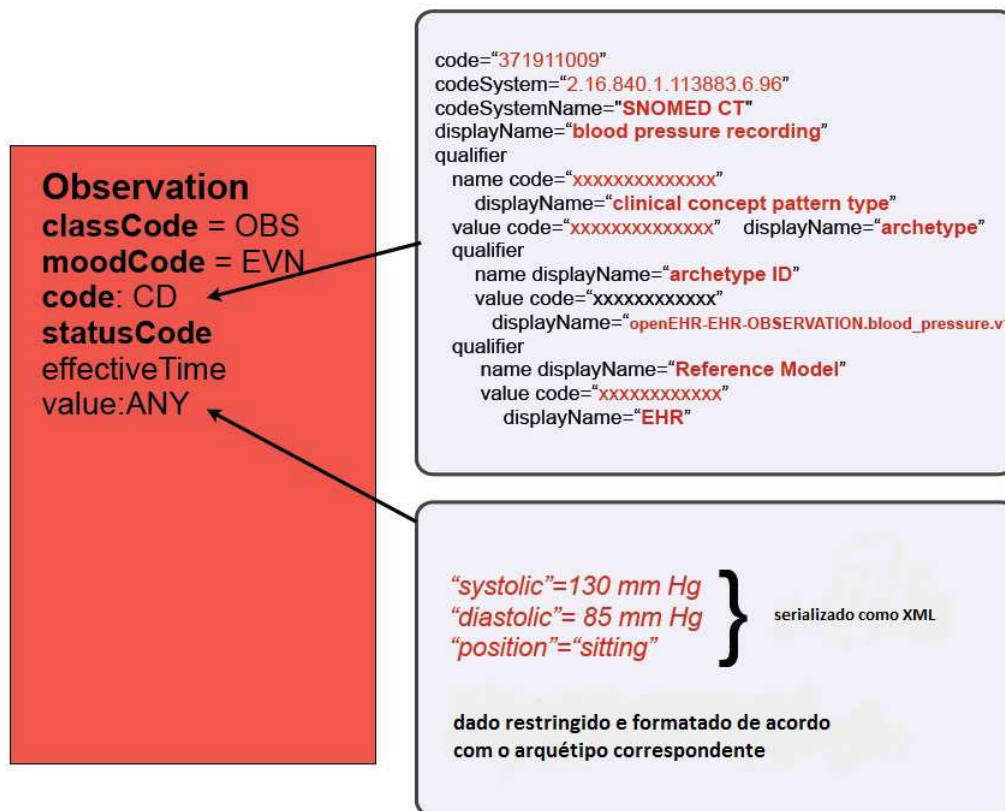


Figura 7.2 – Proposta de Eric Browne para a integração

Este trabalho também foi motivado pela complexidade verificada no processo de desenvolvimento de mensagens HL7 v3 (NEHTA, 2006; JAYARATNA, 2009) e pela escassez de ferramentas simples e intuitivas de apoio ao seu ciclo de vida. Apesar de algumas ferramentas estarem disponíveis no site *HL7 GForge* (GForge, 2010), as tentativas de uso das mesmas foram frustradas por diferentes razões: incompatibilidade com alguns sistemas operacionais; incompatibilidade com outras aplicações; e problemas não identificados.

Visando propor uma abordagem que facilite a integração de arquétipos às mensagens HL7 v3 e seja intuitiva aos usuários, esta emprega o conceito MDD no desenvolvimento dessas mensagens, permitindo que os refinamentos inerentes ao processo de desenvolvimento do HL7 v3 (Figura 4.8) sejam apoiados por modelos e que os artefatos gerados sejam reusáveis.

7.2 Visão Geral da Abordagem

Essa abordagem consiste das etapas **Engenharia de Domínio (ED)** e **Engenharia de Aplicação (EA)**. Na ED são identificados e implementados ou recuperados do repositório *Clinical Knowledge Manager (CKM)* (OPENEHR, 2011a) os arquétipos; é desenvolvida uma DSL representada via um metamodelo para comunicação no domínio da Cardiologia, pertencente ao domínio Cuidado de Saúde; são desenvolvidas transformações M2C para a geração dos esquemas das mensagens; e são desenvolvidos os *parsers* responsáveis por instanciar os objetos descritos nos esquemas das mensagens. Todos esses artefatos são usados como mecanismos de apoio à execução da EA, onde as aplicações ubíquas, dotadas de módulos de comunicação baseados em mensagens HL7 v3 com arquétipos, são implementadas instanciando-se o metamodelo construído na ED. As transformações M2C são reutilizadas para a geração dos esquemas das mensagens HL7 v3 com arquétipos a partir dos modelos. Os *parsers* são usados na forma de *Application Programming Interfaces (APIs)* para permitir que as aplicações criem e interpretem as mensagens.

A Figura 7.3 emprega a notação *Structured Analysis and Design Technique (SADT)* (ROSS, 1977) para ilustrar as seguintes atividades da ED: *Especificar Arquétipos*; *Especificar Metamodelo do Domínio*; *Projetar Metamodelo do Domínio*; *Implementar Metamodelo do Domínio*; *Construir Transformações Modelo para Código*; e *Implementar Parsers*.

Para especificar os arquétipos o Engenheiro de Domínio inicia com os requisitos do domínio do problema. Durante a realização dessa atividade esse engenheiro é guiado pelo *Archetype Model (AM)*, pela sintaxe da *Archetype Definition Language (ADL)* e pelo repositório de arquétipos CKM, sendo seus principais mecanismos um ou mais especialistas do domínio e a ferramenta *Archetype Editor* (OCEAN, 2011) ou outro editor de texto (e.g., Bloco de Notas, VI).

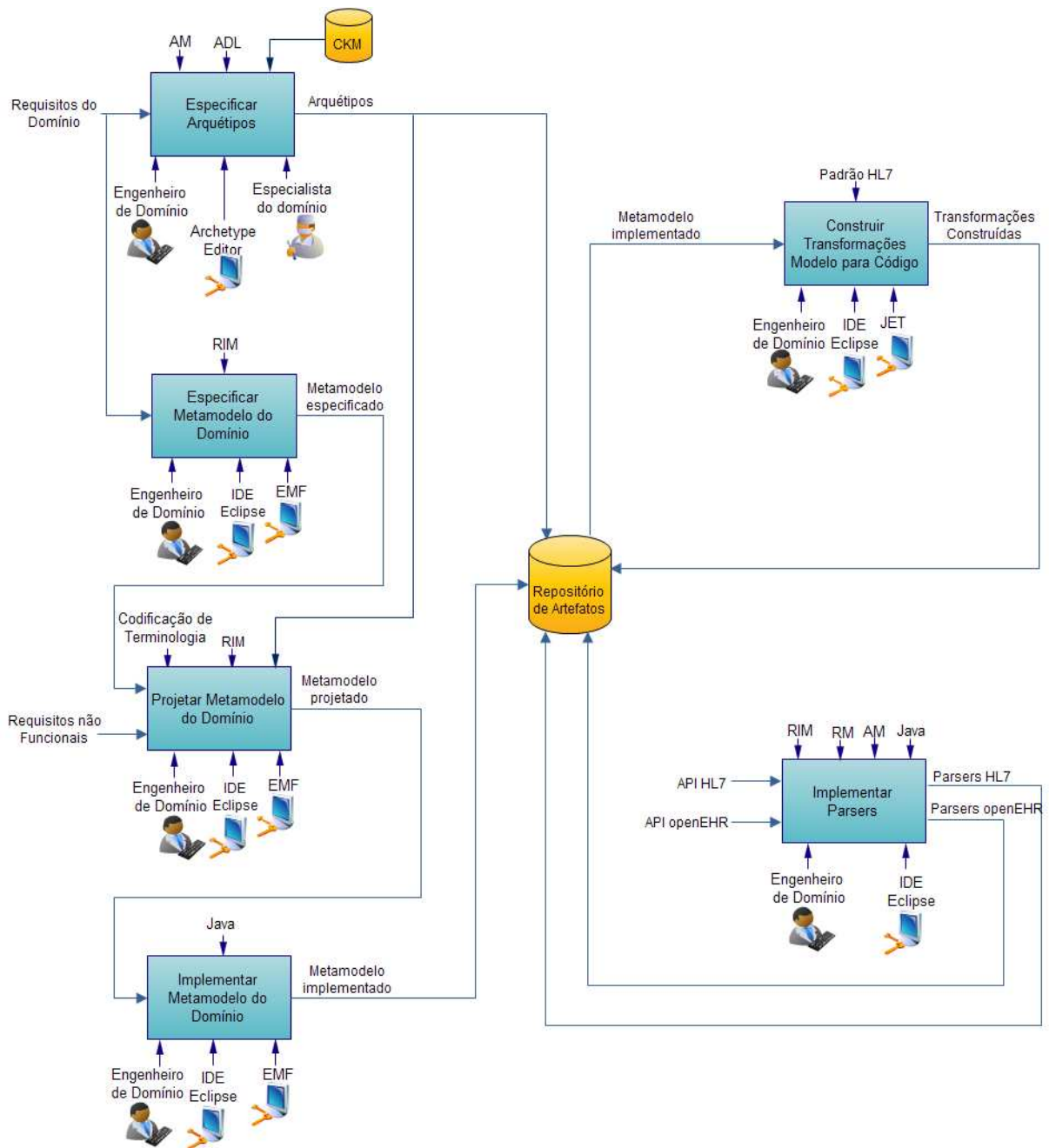


Figura 7.3 – Engenharia de Domínio

Para construir o metamodelo o Engenheiro de Domínio também inicia com os requisitos do domínio do problema. Durante a realização dessa atividade esse engenheiro é guiado pelo RIM HL7, pelos arquétipos especificados, por codificações de terminologia e pela sintaxe da linguagem de programação Java, sendo seus

principais mecanismos a *Integrated Development Environment (IDE)* Eclipse e o *Eclipse Modeling Framework (EMF)* (STEINBERG et al., 2008).

Para construir as transformações M2C o Engenheiro de Domínio inicia com as especificações dos artefatos gerados durante a construção do metamodelo. Durante a realização dessa atividade esse engenheiro é guiado pela especificação do padrão HL7 v3, já que é necessário conhecer o *Model Interchange Format (MIF)* empregado pelo comitê para a representação dos esquemas de todos os artefatos (HL7, 2011b), sendo seus principais mecanismos a IDE Eclipse e o *framework Java Emitter Templates (JET)* (ECLIPSE, 2010).

Para implementar os *parsers* o Engenheiro de Domínio inicia com a API desenvolvida pelo *HL7 Java SIG Project* (JavaSIG, 2010), doravante denominada API HL7, e com a API desenvolvida pelo *openEHR Java Reference Implementation Project* (CHEN, 2010), doravante denominada API *openEHR*. Durante a realização dessa atividade esse engenheiro é orientado pelo RIM HL7, pelo RM *openEHR*, pelo AM *openEHR* e pela sintaxe da linguagem de programação Java, sendo o seu principal mecanismo a IDE Eclipse.

O diagrama da Figura 7.4 apresenta as atividades da EA, que abrangem as disciplinas tradicionais de Análise, Projeto, Implementação e Testes da Engenharia de Software e que sofreram modificações graças ao processo de desenvolvimento de mensagens proposto pelo comitê HL7 (Figura 4.8). Durante a realização dessas atividades o Engenheiro de Aplicação é guiado pela *Unified Modeling Language (UML)* e pelo RIM HL7, sendo os seus principais mecanismos a IDE Eclipse, a ferramenta MVCASE (LUCRÉDIO et al., 2003), o metamodelo implementado, os arquétipos especificados e as transformações M2C construídas.

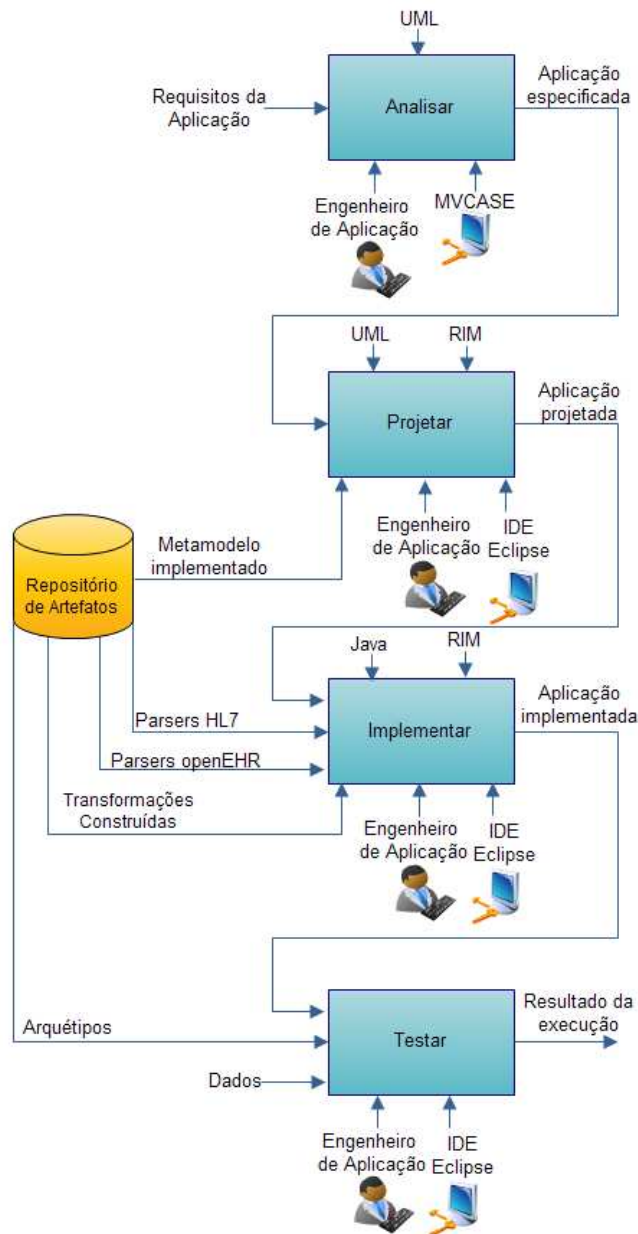


Figura 7.4 – Engenharia de Aplicação

7.3 Engenharia de Domínio

A ED tem por objetivo identificar e modelar tanto as características comuns quanto as variáveis de aplicações de um dado domínio, gerando modelos que podem ser reutilizados em aplicações específicas (PIETRO-DIAZ; ARANGO, 1991). Portanto, a ED é um processo para o desenvolvimento de artefatos reusáveis em um domínio particular de problema (LEITE; GIRARDI, 2009).

7.3.1 Especificar Arquétipos

Nessa atividade é identificado um conjunto inicial de conceitos clínicos, comumente empregados em diferentes especialidades do domínio Cuidado de Saúde. Dentre esses conceitos estão os que não dizem respeito diretamente a uma especialidade, tais como pressão sanguínea, temperatura corporal e peso.

Com base no levantamento efetuado, o Engenheiro de Domínio pode verificar a disponibilidade e baixar do repositório CKM arquétipos que representam tais conceitos. Além de um repositório que disponibiliza arquétipos, *templates* e outros artefatos *openEHR*, o CKM é também um sistema para o desenvolvimento colaborativo, gerenciamento e publicação desses artefatos (OPENEHR, 2011a). A Figura 7.5 exibe a tela de busca por arquétipos, sendo que o arquétipo buscado foi o de pressão sanguínea (*blood pressure*), enquanto a Figura 7.6 exibe a tela que permite aos usuários efetuar a descarga do artefato nas diferentes línguas para as quais o mesmo já foi traduzido.

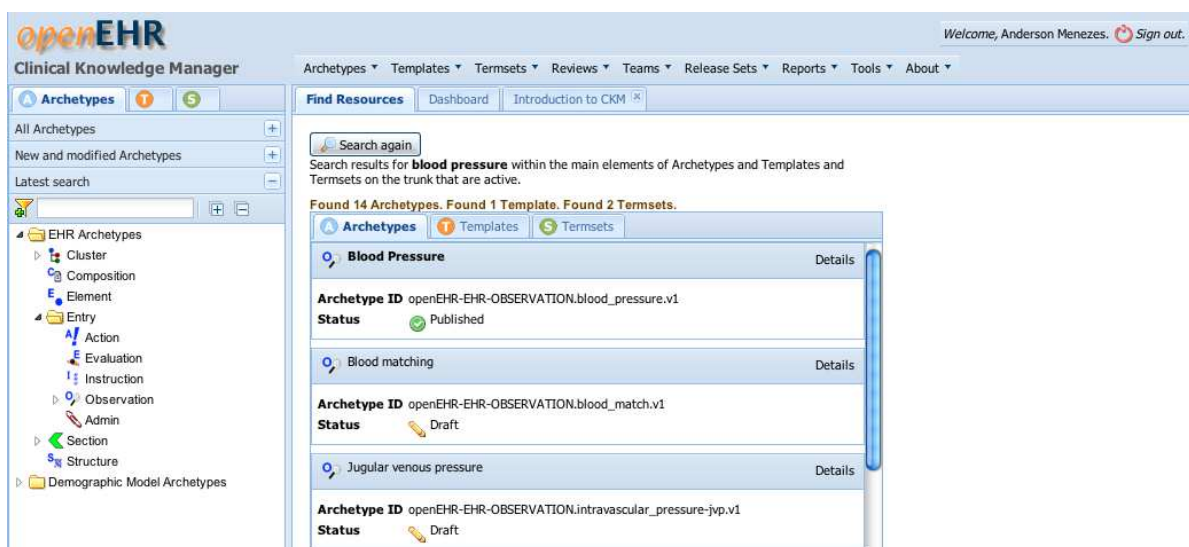


Figura 7.5 – Busca pelo arquétipo para pressão sanguínea

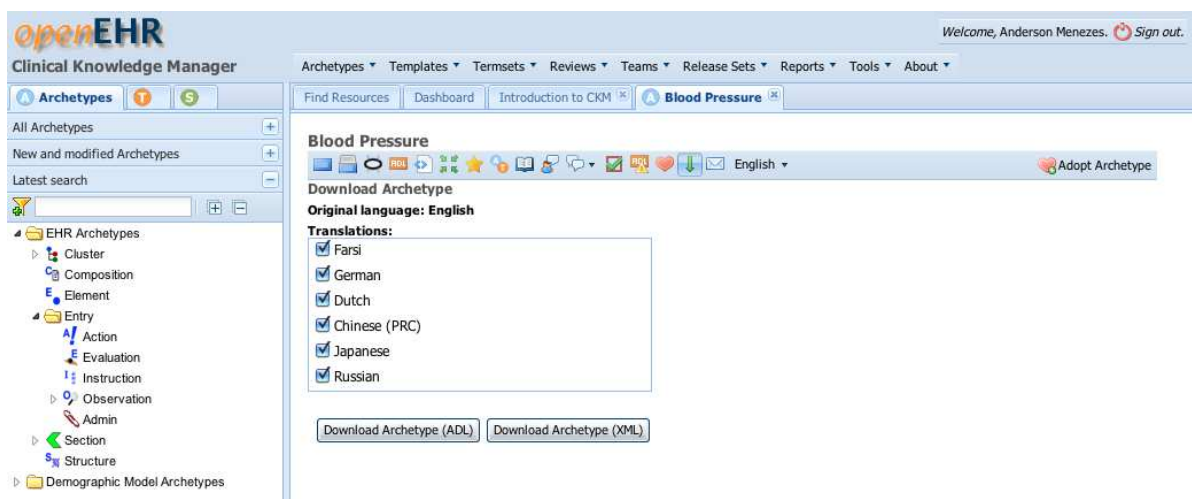


Figura 7.6 – Descarga do arquétipo para pressão sanguínea

Caso os arquétipos identificados não estejam disponíveis no CKM, o Engenheiro de Domínio tem a possibilidade de especificá-los com o auxílio do *Archetype Editor* ou de outro editor de texto. A Figura 7.7 ilustra o uso dessa ferramenta onde o engenheiro, sob orientação do especialista do domínio, seleciona o tipo de arquétipo a ser criado (e.g., **CLUSTER**, **COMPOSITION**, **OBSERVATION**) e define a árvore de elementos pertencentes ao conceito, seus relacionamentos com codificações de terminologia, sua descrição e as demais informações necessárias à especificação do arquétipo. A Figura 7.8 apresenta um trecho da ADL resultante dessa implementação.

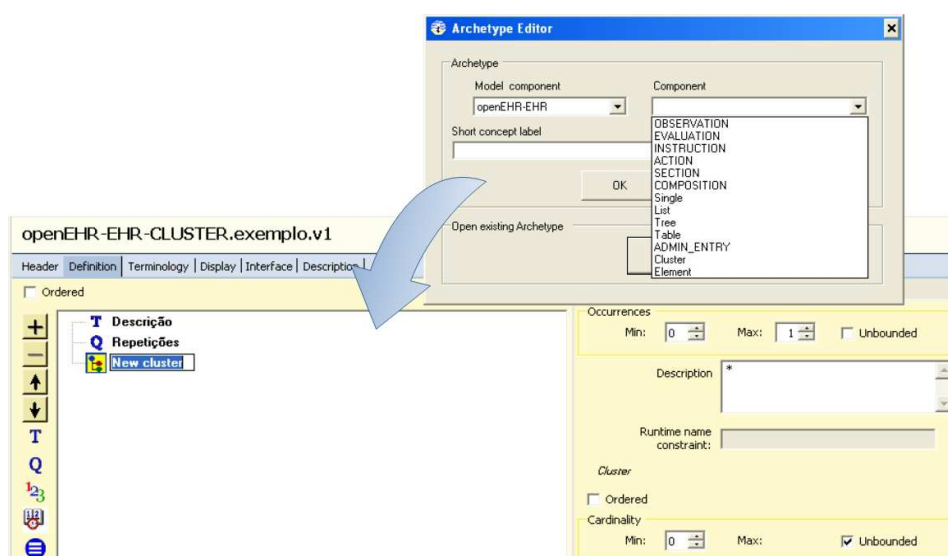


Figura 7.7 – Archetype Editor


```

archetype (adl_version=1.4)
  openEHR-EHR-CLUSTER.exemplo.v1

  concept
    [at0000] -- Exemplo
  language
    original_language = <[ISO_639-1::pt]>
  description
    [...]
    details = <
      ["pt"] = <
        language = <[ISO_639-1::pt]>
        purpose = <"Exemplificar a criação de um arquétipo">
      >
    >
  definition
    CLUSTER[at0000] matches { -- Exemplo
      items cardinality matches {0..*; unordered} matches {
        ELEMENT[at0001] occurrences matches {0..1} matches { -- Descrição
          value matches {
            DV_TEXT matches {*}
          }
        }
        ELEMENT[at0002] occurrences matches {0..1} matches { -- Repetições
          value matches {
            C_DV_QUANTITY <
              >
            }
          }
        }
      CLUSTER[at0003] occurrences matches {0..1} matches {*}
    }
  }
  [...]
}

```

Figura 7.8 – ADL gerada pelo Archetype Editor

7.3.2 Especificar Metamodelo do Domínio

Nessa atividade os requisitos do domínio Cardiologia são identificados, especificados, analisados e representados num metamodelo que expressa o conhecimento desse domínio. A identificação desses requisitos permite ao Engenheiro de Domínio reconhecer as principais características do domínio alvo, tais como os atores, as entidades, as ações e os relacionamentos existentes entre os mesmos.

No IDE Eclipse o Engenheiro de Domínio, guiado pelo RIM HL7, especifica o metamodelo usando o EMF. O RIM HL7 representa a informação num alto nível de abstração, o que permite o uso de diferentes terminologias na descrição dos conceitos desse domínio (BENSON, 2009). A Figura 7.9 mostra parte do metamodelo elaborado, onde as metaclasses, os meta-atributos e os metarrelacionamentos foram criados a partir de um mapeamento direto do diagrama de classes do RIM exibido na Figura 4.5.

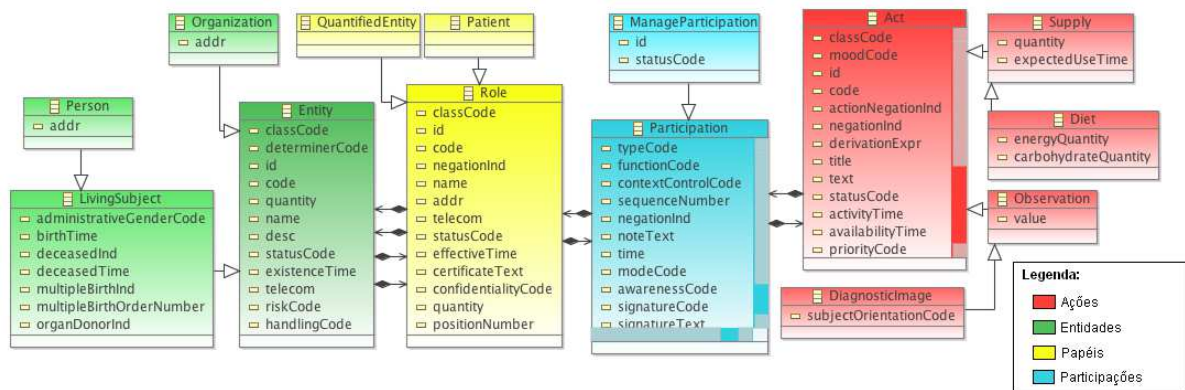


Figura 7.9 – Metamodelo especificado

7.3.3 Projetar Metamodelo do Domínio

Nessa atividade a especificação do metamodelo é refinada, de acordo com os padrões, tecnologias e plataformas que viabilizaram a construção do mesmo. Empregando o EMF, o Engenheiro de Domínio adiciona ao metamodelo elementos para a identificação dos principais conceitos clínicos do domínio Cardiologia. O processo é guiado pelas codificações de terminologias médicas, tais como a *Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT)*, pelo RIM HL7 e pelos arquétipos especificados.

A adição de elementos que tratam das codificações de terminologia ao metamodelo permite identificar de forma única os conceitos clínicos representados (e.g., pressão arterial, glicose sanguínea, temperatura corporal). A Figura 7.10 mostra parte do metamodelo com a inclusão da metaenumeração **Archetype**, que lista os conceitos “bpArchetype” e “glucArchetype”, os quais correspondem respectivamente aos conceitos clínicos de pressão sanguínea e concentração de glicose sanguínea definidos pela SNOMED-CT.

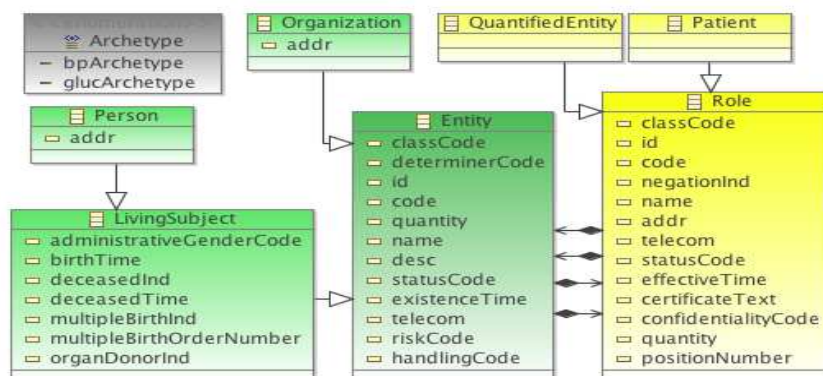


Figura 7.10 – Metamodelo com inclusão dos conceitos

Neste trabalho a inclusão da metaenumeração com códigos SNOMED-CT colabora diretamente para integrar os dados restringidos pelos arquétipos às mensagens HL7. Nessa atividade, o tipo de dados padrão associado ao meta-atributo “code” (CS), da metaclasses **Observation**, é substituído pelo novo tipo definido pela metaenumeração **Archetype**. Isso permite que o código do conceito seja atribuído ao esquema da mensagem no seu processo de modelagem, habilitando posteriormente as aplicações a identificar o conceito associado à mensagem e realizar os processamentos inerentes a cada um desses conceitos.

7.3.4 Implementar Metamodelo do Domínio

Nessa atividade é realizada a implementação com base nas especificações contidas no projeto do metamodelo. Apoiado pelo *framework* EMF, o Engenheiro de Domínio gera o código Java do metamodelo e de um editor de modelos. Na etapa EA, esse editor auxiliará na especificação dos modelos das mensagens a partir do metamodelo. A Figura 7.11 apresenta um trecho do código de implementação da metaclasses **Observation**, onde esta herda as características existentes na implementação da metaclasses **Act**, da mesma forma que ocorre na metamodelagem da Figura 7.9.

```

1 package RIM.impl;
2
3 import RIM.ANY;
4 import RIM.Observation;
5 import RIM.RIMPackage;
6
7 import org.eclipse.emf.common.notify.Notification;
8 import org.eclipse.emf.common.notify.NotificationChain;
9
10 import org.eclipse.emf.ecore.EClass;
11 import org.eclipse.emf.ecore.InternalEObject;
12
13 import org.eclipse.emf.ecore.impl.ENotificationImpl;
14
15 public class ObservationImpl extends ActImpl implements Observation {
16
17     protected ANY value;
18
19     public void setValue(ANY newValue) {
20         if (newValue != value) {
21             NotificationChain msgs = null;
22             if (value != null)
23                 msgs = ((InternalEObject)value).eInverseRemove(this, EOPPOSITE_FEATURE_BASE - RIMPackage.OBSERVATION__VALUE,
24                     null, msgs);
25             if (newValue != null)
26                 msgs = ((InternalEObject)newValue).eInverseAdd(this, EOPPOSITE_FEATURE_BASE - RIMPackage.OBSERVATION__VALUE,
27                     null, msgs);
28             msgs = basicSetValue(newValue, msgs);
29             if (msgs != null) msgs.dispatch();
30         }
31         else if (eNotificationRequired())
32             eNotify(new ENotificationImpl(this, Notification.SET, RIMPackage.OBSERVATION__VALUE, newValue, newValue));
33     }
34 }

```

Figura 7.11 – Implementação do metamodelo

A execução dessa atividade resulta na disponibilização do metamodelo e do editor de modelos correspondente como um *plugin* no IDE Eclipse. Durante a EA, esse *plugin* permite que o metamodelo seja instanciado para a modelagem das mensagens. Uma vez que o metamodelo baseia-se no RIM, os modelos gerados, usando a DSL expressa por esse metamodelo, correspondem ao refinamento que deve ser realizado, criando dessa forma as mensagens HL7 empregadas no intercâmbio dos dados das aplicações.

7.3.5 Construir Transformações Modelo para Código

Nessa atividade são criadas as transformações, que serão aplicadas aos modelos construídos na EA, para a geração dos esquemas das mensagens. Essas transformações buscam reduzir os esforços de desenvolvimento das mensagens e viabilizar o reuso dos modelos, já que os mesmos podem ser empregados para diferentes fins dentro do domínio.

O uso de *templates* na construção de transformações tem se destacado dentre as demais técnicas existentes (LUCRÉDIO, 2009). O *template* é um arquivo de texto qualquer, instrumentado com construções de seleção e expansão de código (CZARNECKI; EISENECKER, 2000). As soluções baseadas em DSLs e *templates* são compostas por três elementos principais: a entrada, que especifica as características do modelo e pode ser representado em diferentes formatos, tais como XML ou *XML Metadata Interchange (XMI)*; o *template*, responsável por guiar as consultas efetuadas na entrada; e o processador de *templates*, que instancia o *template* com base na entrada a fim de consultá-la e produzir o texto correspondente de saída (LUCRÉDIO, 2009).

Com o auxílio do *framework* JET, o Engenheiro de Domínio implementa os *templates* que interpretam os modelos de entrada, para a geração do código referente aos esquemas das mensagens HL7 v3 com arquétipos. A Figura 7.12 apresenta parte do *template* JET usado na geração dos elementos da classe **Observation** nos esquemas das mensagens. Esse *template* é executado sempre que o nó correspondente à classe *Observation* é encontrado durante a leitura do XMI. As iterações vistas no *template* percorrem todo o bloco referente à classe **Observation** e combina os trechos fixos e variáveis para formar a saída adequada. As partes fixas são comuns a todos os modelos de entrada e são simplesmente

replicadas no arquivo de saída. As partes variáveis dependem das consultas efetuadas na entrada e podem gerar resultados diferentes de acordo com o modelo consultado. Nesse caso específico a saída consiste num arquivo MIF, onde constam as informações referentes a um esquema de mensagem HL7 v3 com arquétipo.

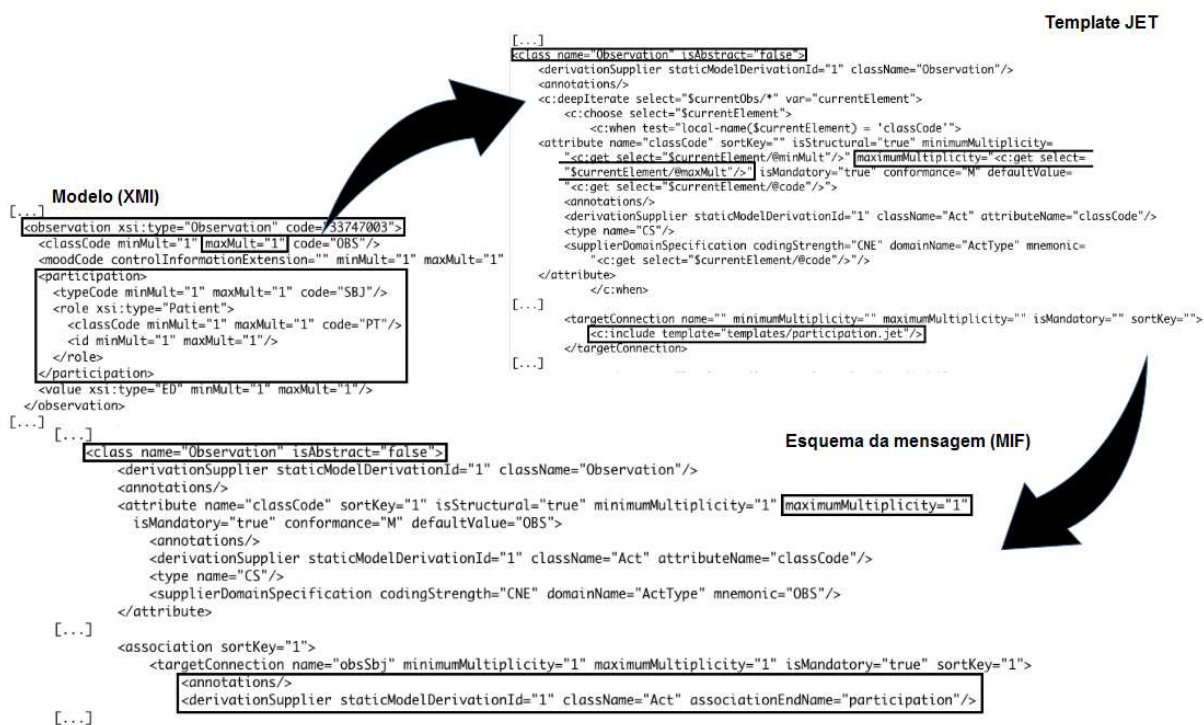


Figura 7.12 – Geração do esquema da mensagem baseada em *templates* Jet

É importante ressaltar que o desenvolvimento das transformações depende diretamente do conhecimento do Engenheiro de Domínio acerca do metamodelo. Como pode se observado na Figura 7.12, o *template* deve ser capaz de identificar os elementos de entrada para produzir o esquema de saída. Cada parte do *template* é processada para que leituras sejam efetuadas no modelo de entrada e para que sejam gerados os trechos correspondentes nos esquemas das mensagens.

O emprego de transformações apresenta grandes benefícios. Primeiramente percebe-se que a manutenção dos códigos gerados é facilitada, pois não existe a necessidade de verificação constante de todos os elementos gerados (LUCRÉDIO, 2009). Destaca-se também a possibilidade de geração de código para qualquer linguagem de programação ou formato, dando liberdade de escolha aos desenvolvedores (CZARNECKI; EISENECKER, 2000), e que soluções baseadas em

templates podem gerar códigos mais legíveis que os gerados por outras técnicas (VÖLTER, 2008).

7.3.6 Implementar Parsers

Nessa atividade são implementados os *parsers* responsáveis pela instanciação dos objetos do RIM HL7 e do RM *openEHR* a partir dos MIFs, que representam esquemas das mensagens, e a partir das ADLs, que representam os arquétipos. O Engenheiro de Domínio inicia pelas com as APIs HL7 e *openEHR*, que representam a implementação dos modelos de referência RIM HL7, RM e AM e permitem um mapeamento inicial dos metadados contidos nos esquemas, sendo guiado por esses modelos e pela linguagem Java e auxiliado pela IDE Eclipse.

A Figura 7.13 ilustra a operação do *parser* desenvolvido para o manuseio de mensagens HL7 v3. Assim que um esquema de mensagem HL7, na forma de arquivo MIF, é carregado para a aplicação, o módulo **MessageObjectMaker**, desenvolvido nessa atividade e responsável pela instanciação de objetos RIM, verifica o identificador da mensagem⁴ (1) e, com base neste identificador, usa o **MessageTypeLoader** (2), pertencente à API HL7, para instanciar um objeto que contém os metadados do esquema da mensagem. Posteriormente esses metadados são retornados para o **MessageObjectMaker** (3) e, após a leitura dos metadados e a instanciação do objeto RIM, o mesmo é retornado para a aplicação (4) para que as informações sejam imputadas aos seus atributos. O **MessageObjectMaker** efetua também a leitura do código SNOMED-CT, referente ao conceito clínico associado à mensagem, e sugere à aplicação que o arquétipo correspondente seja carregado.

⁴ O identificador é formado pelo código de uma subseção, tal como a *Operations (PO)* pertencente à seção *Health & Clinical Management Domains*, pelo domínio dessa subseção, tal como o *Laboratory (LB)*, e por um valor numérico de seis dígitos, os quais conjuntamente identificam de forma exclusiva uma mensagem ou qualquer outro artefato HL7 v3.

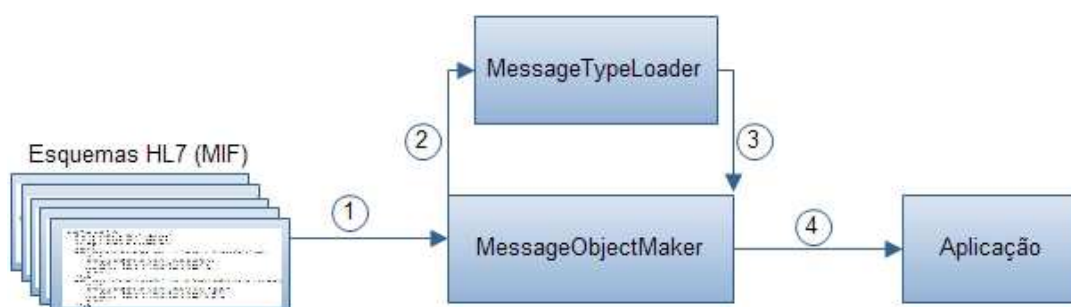


Figura 7.13 – Operação do MessageObjectMaker

A Figura 7.14 ilustra a operação do *parser* desenvolvido para o manuseio de arquétipos. O processo é semelhante ao ilustrado na Figura 7.13, sendo que um arquétipo, na forma de arquivo ADL, ao ser carregado para a aplicação, é verificado pelo módulo **ContentObjectMaker** (1), desenvolvido nessa atividade e responsável pela instanciação de objetos RM, e é repassado para o **ADLParser** (2), pertencente à API *openEHR* e cuja função é instanciar o objeto AM com todas as restrições listadas na ADL. O arquétipo, agora na forma de objeto AM, é retornado ao **ContentObjectMaker** (3), o qual verifica se o arquétipo instanciado possui outros arquétipos referenciados em seus *slots* e, em caso positivo, efetua chamadas recursivas (4) para possibilitar a instanciação dos objetos RM referentes a cada um desses arquétipos. Finalmente o **ContentObjectMaker** retorna o arquétipo para a aplicação (5), permitindo assim que valores sejam imputados aos seus atributos e que esse objeto seja futuramente usado como o conteúdo da mensagem.

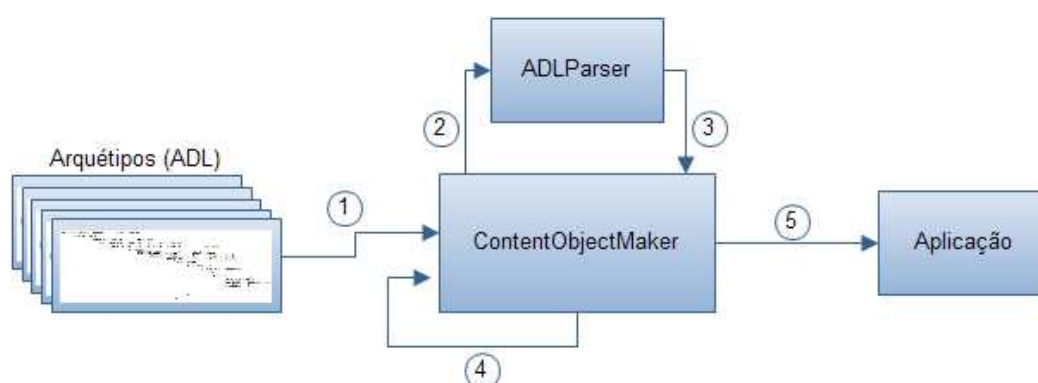


Figura 7.14 – Operação do ContentObjectMaker

Os *parsers* responsáveis pelo manuseio dos XML das mensagens são também desenvolvidos nessa atividade. O objetivo desses *parsers* é efetuar a leitura do XML, invocar o **MessageObjectMaker** e o **ContentObjectMaker**, conforme a

necessidade de instanciação dos objetos, e imputar os valores verificados na mensagem aos seus respectivos atributos nos objetos. Além disso, os módulos **MessageValuesSetter** e **ContentValuesSetter**, responsáveis pela imputação dos valores aos objetos e que possuem os métodos específicos para as características de cada modelo de referência, também são desenvolvidos. A Figura 7.15 ilustra a operação de uma aplicação que emprega os artefatos citados.

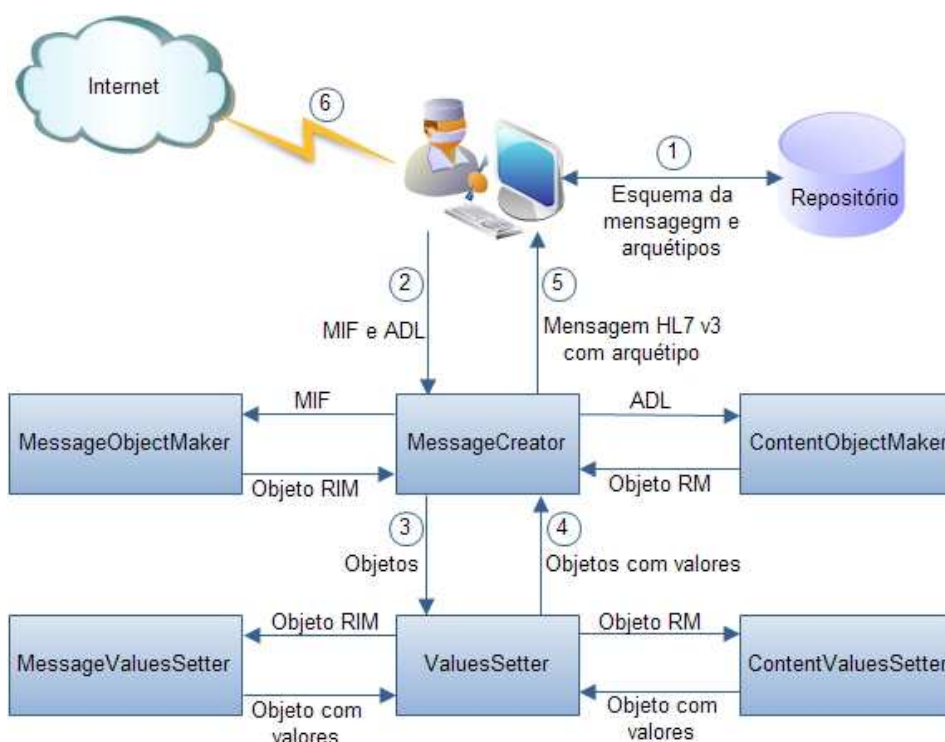


Figura 7.15 – Exemplo de operação de uma aplicação

O exemplo da Figura 7.15 sugere um cenário arbitrário, que permite a um usuário enviar resultados de exames de laboratório a uma clínica. O evento enviar resultados dos exames, disparado pelo usuário, faz com que a lógica da aplicação reconheça o tipo de informação a ser transmitida (e.g., resultado do Exame X) e solicite que tanto o esquema da mensagem quanto os arquétipos relacionados ao conceito clínico sejam carregados (1). Com o suporte do **MessageCreator**, os módulos **MessageObjectMaker** e **ContentObjectMaker** são empregados, conforme os exemplos das Figuras 7.13 e 7.14, para instanciar e disponibilizar os respectivos objetos para a imputação dos dados (2). De posse dos objetos, o **MessageCreator** invoca o **MessageValuesSetter** e o **ContentValuesSetter** para efetuarem essa imputação (3), que uma vez concluída fará com que os objetos sejam retornados ao

MessageCreator (4), o qual realizará a integração e serialização dos mesmos em XML, que por sua vez será retornado à aplicação (5). Nesse ponto a aplicação torna-se apta ao envio da mensagem HL7 v3 com arquétipo (6).

7.4 Engenharia de Aplicação

A EA visa a construção de aplicações específicas de um dado domínio reusando artefatos de software provenientes da ED (LEITE; GIRARDI, 2009). Essa etapa da abordagem tem foco no desenvolvimento das aplicações, no caso a criação de mensagens HL7 v3 com arquétipos. As disciplinas tradicionais da Engenharia de Software tem seu ciclo de vida estendido com os conceitos de MDD e são orientadas pelo processo de desenvolvimento proposto pelo comitê HL7. O uso do metamodelo e das transformações M2C agiliza as tarefas do Engenheiro de Aplicação, já que permitem a modelagem das mensagens e a automatização da geração de seus esquemas. Havendo necessidade, atividades da ED (e.g., Especificar Arquétipos) podem ser reiteradas para que o metamodelo abranja mais artefatos e cubra requisitos de uma quantidade maior de aplicações do domínio.

Para ilustrar a EA, esta seção utiliza uma aplicação do domínio Cardiologia, desenvolvida em caráter exploratório, que auxiliará no detalhamento das atividades. Trata-se de uma aplicação ubíqua, denominada *Hiper Registro Eletrônico de Saúde (HiperRES)*, que possibilita aos pacientes hipertensos registrarem suas pressões sanguíneas em seus dispositivos móveis. Esses registros são transmitidos a um servidor remoto e permitem aos profissionais de Saúde, responsáveis por esses pacientes, analisar os dados transmitidos e agir rapidamente caso necessário.

De acordo com a Figura 7.16 o HiperRES consiste de três partes: a primeira, executada no dispositivo móvel, registra e persiste os dados pressóricos inseridos pelo paciente, transmitindo-os para um servidor remoto via mensagens HL7 v3 com arquétipos; a segunda, executada no servidor, recebe essas mensagens, trata os dados baseados em arquétipos contidos nas mesmas e os persiste numa base de dados; a última, também executada no servidor, provê uma interface *web* permitindo aos profissionais de Saúde visualizar os dados pressóricos de seus pacientes.

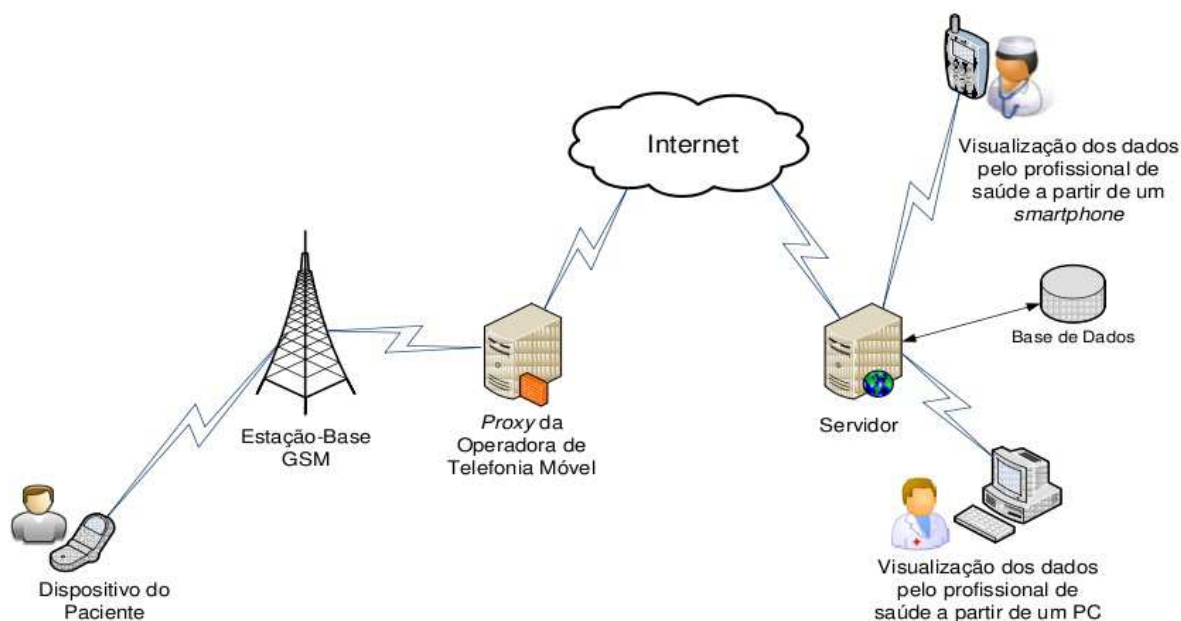


Figura 7.16 – Arquitetura da aplicação

7.4.1 Analisar

Nessa atividade a aplicação é especificada conforme seus requisitos. O Engenheiro de Aplicação faz uso de notações UML, tais como a diagramação dos casos de uso e classes, sendo guiado pela sintaxe e semântica da UML e auxiliado pela ferramenta MVCASE, empregada na construção dos diagramas.

Durante a Análise são também realizadas duas atividades relacionadas com o desenvolvimento de mensagens HL7 v3: **Especificar Casos de Uso** e **Especificar Modelo de Interação**. Essas atividades empregam notações UML nas suas especificações, o que permite o compartilhamento dos diagramas produzidos.

Em **Especificar Casos de Uso** são descritas as condições para a criação de cada mensagem HL7, os atores que agem nessas mensagens e outras informações relevantes. Por exemplo, o diagrama de casos de uso UML da Figura 7.17 descreve: os requisitos do HiperRES relativos ao registro de dados pressóricos; a submissão desses dados ao servidor via mensagens HL7 v3 com arquétipos; e o acesso desses dados pelo cardiologista.

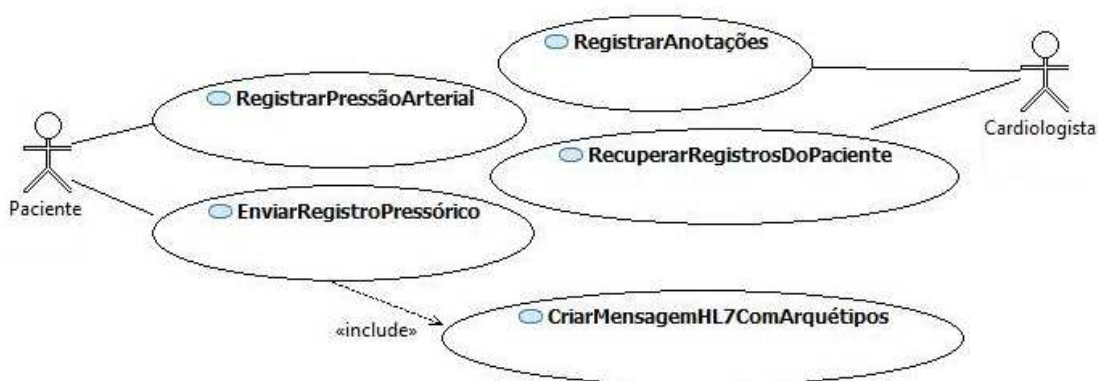


Figura 7.17 – Casos de uso do HiperRES

Em **Especificar Modelo de Interação** são descritas as interações dos atores e características das mensagens trocadas (e.g., fluxo de informações, eventos disparadores). Por exemplo, a Figura 7.18 descreve as interações do caso de uso EnviarRegistroPressórico, onde o ator paciente, ao registrar seus dados pressóricos, solicita o envio dessas informações ao servidor remoto. Nesse momento, a aplicação em execução no dispositivo do paciente instancia o objeto HL7 através do MessageObjMaker, instancia o objeto *openEHR* referente ao registro pressórico através do ContentObjMaker e, com método *setMessageValues*, atribui os dados pressóricos ao campo *value* da mensagem. Por fim, o objeto HL7 é serializado pelo MessageXMLBuilder e enviado ao servidor com a chamada ao método *hl7Data*.

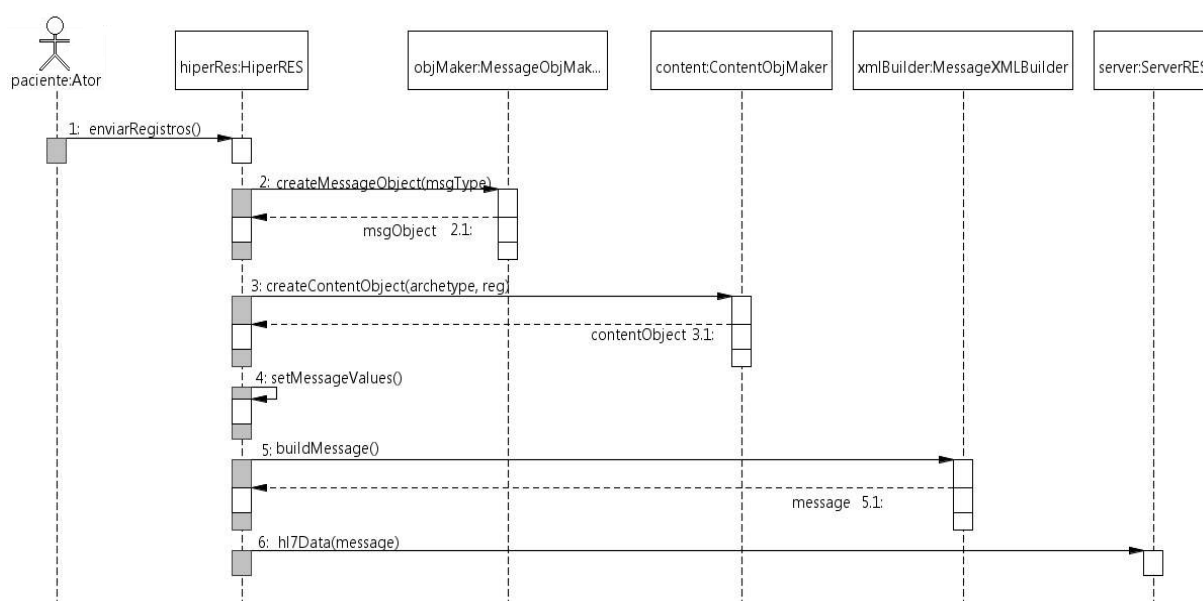


Figura 7.18 – Diagrama de sequência do caso de uso EnviarRegistroPressórico

7.4.2 Projetar

Nessa atividade as especificações da Análise são refinadas com as tecnologias de hardware e software pertinentes, tais como a plataforma Java ME (SUN, 2009) e o *framework Floggy* (FLOGGY, 2009), o qual facilita o trabalho de manipulação das rotinas de persistência contidas na API do *Record Management System (RMS)*.

Especificar Modelo de Informação também é realizada durante a disciplina de Projeto. Nessa atividade o Engenheiro de Aplicação modela as mensagens com base no metamodelo desenvolvido na ED e nos diagramas de casos de uso e de sequência, já que estes descrevem os atores e expressam suas interações com as mensagens. Uma vez que esse metamodelo reflete o conhecimento expresso no RIM HL7, os modelos criados como instâncias do mesmo refletem o refinamento do RIM, provendo assim um subconjunto de classes e atributos que representam as necessidades do domínio do problema. Por exemplo, a Figura 7.19 ilustra parte do modelo de mensagens do HiperRES, sendo que: à esquerda é apresentada uma visão em árvore desse modelo, desenvolvido a partir do metamodelo; e à direita é apresentado o diagrama de objetos instanciados do metamodelo.

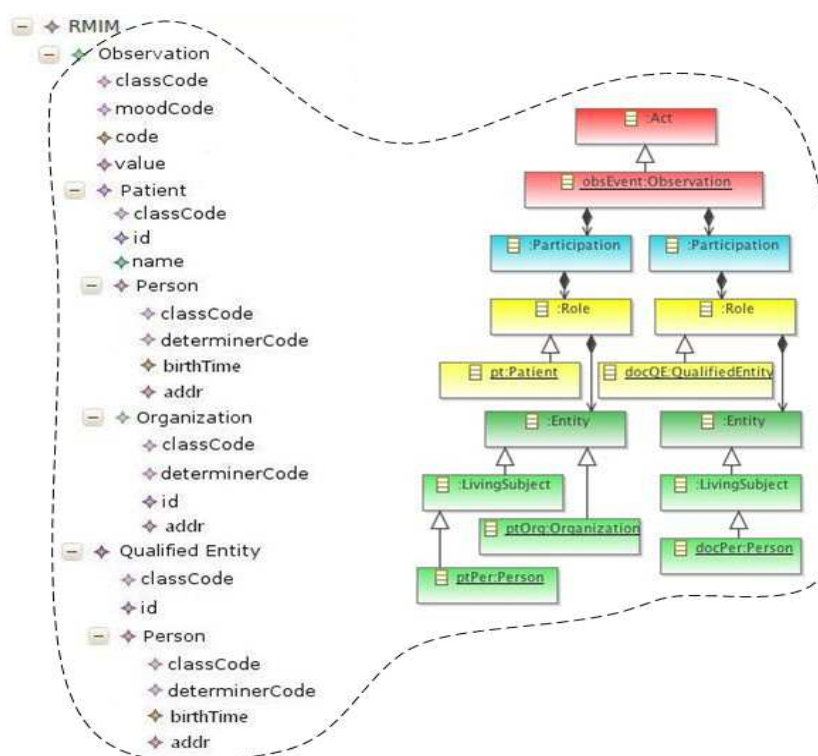


Figura 7.19 – Modelo de mensagens

Esse modelo representa as informações da aplicação para o intercâmbio de registros pressóricos. Como especificado no diagrama de casos de uso, o paciente registra a pressão sanguínea em seu dispositivo móvel. Esse registro é traduzido numa **Observation** no modelo de mensagens. A associação entre o paciente e seu registro de pressão sanguínea é indicada pelo elemento **Participation**. Esse elemento é composto de uma **Person** e uma **Organization**, as quais descrevem os atributos do paciente e a instituição de saúde na qual o mesmo recebe cuidados médicos. No modelo tem-se uma segunda **Participation** associada à **Qualified Entity** para representar o profissional de Saúde responsável pelo paciente.

Adotando a codificação SNOMED-CT, empregada no metamodelo, os arquétipos que definem os conceitos clínicos, representados nos modelos das mensagens, são combinados aos seus respectivos códigos. Por exemplo, no modelo da Figura 7.19 o código SNOMED-CT, que identifica o conceito clínico de pressão sanguínea, é atribuído ao campo “code” do elemento **Observation**. Além disso, são atribuídos ao campo “value” os dados restringidos pelo arquétipo, que correspondem ao registro de pressão sanguínea.

7.4.3 Implementar

Nessa atividade tem-se a codificação da aplicação e a geração dos esquemas das mensagens, correspondendo também à realização da atividade **Criar Esquemas das Mensagens**.

Com o auxílio da IDE Eclipse, o Engenheiro de Aplicação executa as transformações M2C, que são aplicadas aos modelos para gerar a estrutura das mensagens empregadas no transporte de dados de Saúde. Por exemplo, a Figura 7.20 mostra um trecho do esquema de mensagens do HiperRES, gerado pelas transformações da IDE Eclipse a partir do modelo ilustrado na Figura 7.19. Esse esquema define a estrutura das mensagens HL7 v3 com arquétipos, que serão criadas pela aplicação para o intercâmbio dos dados pressóricos.

```

<class name="Observation" isAbstract="false">
  <derivationSupplier staticModelDerivationId="1" className="Observation"/>
  <annotations/>
  <attribute name="classCode" sortKey="1" isStructural="true" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="M" defaultValue="OBS">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="classCode"/>
    <type name="CS"/>
    <supplierDomainSpecification codingStrength="CNE" domainName="ActType" mnemonic="OBS"/>
  </attribute>
  <attribute name="moodCode" sortKey="2" isStructural="true" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="M" defaultValue="EVN">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="moodCode"/>
    <type name="CS"/>
    <supplierDomainSpecification codingStrength="CNE" domainName="ActMood" mnemonic="EVN"/>
  </attribute>
  <attribute name="code" sortKey="3" isStructural="false" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="R">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="code"/>
    <type name="ARCHETYPE"/>
    <supplierDomainSpecification codingStrength="CWE" domainName="ActCode"/>
  </attribute>
  <attribute name="value" sortKey="4" isStructural="false" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="R">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Observation" attributeName="value"/>
    <type name="ED"/>
  </attribute>

```

Figura 7.20 – Esquema de mensagens

Além disso, os *parsers* desenvolvidos na ED são incorporados como APIs na aplicação, garantindo à mesma a capacidade de criação, serialização e interpretação das mensagens HL7 v3 com arquétipos. Devem também ser implementados nessa atividade os demais elementos da aplicação, tais como as interfaces com o usuário e a persistência dos dados, os quais extrapolam o contexto desta proposta e, por esse motivo, não são detalhados na mesma.

7.4.4 Testar

A Figura 7.21 ilustra os resultados do teste de execução da aplicação HiperRES: a imagem à esquerda mostra a execução do HiperRES no aparelho celular; a imagem no centro mostra a interface WebRES para a visualização dos dados, a qual é executada num *smartphone*; a imagem à direita mostra um trecho de mensagem HL7 que transporta os dados pressóricos, baseados em arquétipos, do dispositivo do paciente para o servidor. Essa mensagem foi instanciada pelo HiperRES a partir do esquema apresentado na Figura 7.20.

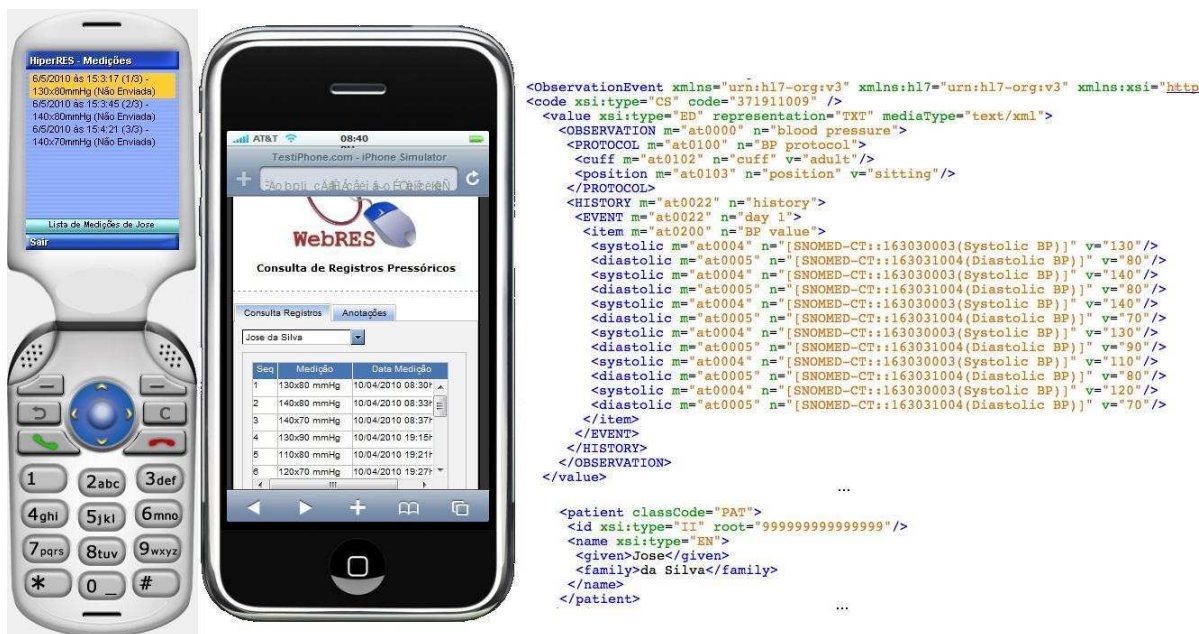


Figura 7.21 – Teste de execução da aplicação

A execução da EA facilita a modelagem e a geração parcial dos esquemas de mensagens HL7, reduzindo os esforços de desenvolvimento. O uso conjunto de arquétipos e HL7 permite criar mensagens semanticamente mais interoperáveis, que podem ser trocadas entre diferentes aplicações que empreguem as mesmas tecnologias.

Capítulo 8

ESTUDO DE CASO

No início das pesquisas foi cogitada a implementação de um módulo de comunicação, baseado no uso do protocolo HL7 v3, como objetivo central deste trabalho. Esse módulo então seria usado como parte de um projeto de doutorado em desenvolvimento neste programa de pós-graduação. O referido projeto diz respeito a um Sistema de Mensagens Sensível ao Contexto para o Cuidado de Saúde (SMSCCS), o qual visa suportar a mobilidade e o trabalho colaborativo dos profissionais de Saúde no cumprimento de suas atividades diárias. A Figura 8.1 (MORAES et al., 2009) apresenta uma visão abstrata da arquitetura do SMSCCS.

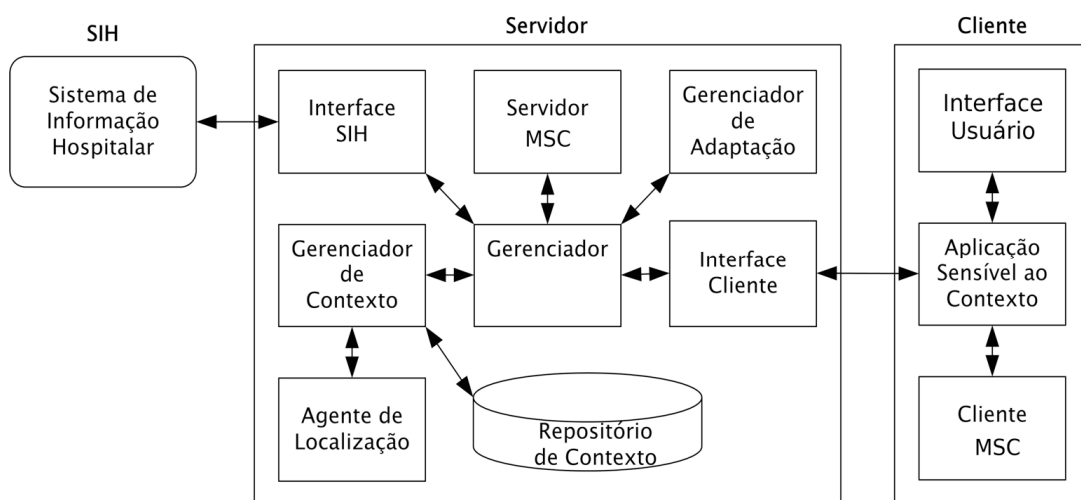


Figura 8.1 – Arquitetura abstrata do SMSCCS

A arquitetura ilustrada na figura acima é composta basicamente por um Sistema de Informação Hospitalar (SIH) integrado a um Servidor de Mensagens

Sensível ao Contexto, a Clientes de Mensagens Sensível ao Contexto e a um conjunto de Agentes. O **Gerenciador**, presente no servidor, é responsável por todas as comunicações efetuadas entre os módulos do mesmo. O **Gerenciador de Contexto** é responsável pelo processamento das informações dinâmicas de contexto, obtidas por meio do **Agente de Localização**, e pelo processamento das informações estáticas de contexto, obtidas por meio do **Repositório de Contexto**. O **Gerenciador de Adaptação** é responsável pela verificação e adaptação de conteúdo efetuada conforme as características dos dispositivos dos usuários. Após o processamento de adaptação, o conteúdo é repassado ao módulo **Aplicação Sensível ao Contexto**, presente no dispositivo do usuário, que se comunica com os módulos **Cliente de Mensagens Sensível ao Contexto** e **Interface de Usuário**. Os agentes, por sua vez, são responsáveis pela revisão do contexto e pela tomada de decisões relativas às atividades a desempenhar, quando desempenhá-las, que tipo de informação comunicar e a quem comunicá-la (MORAES et al., 2009).

O módulo de comunicação seria responsável por toda a comunicação existente entre o servidor, SIH e clientes. Entretanto, na medida em que foram verificadas a complexidade do processo de desenvolvimento das mensagens HL7 v3 e as deficiências na representação dos conceitos clínicos apresentados pelo RIM HL7, foi proposta a criação de uma abordagem que visasse simplificar esse processo e aprimorar a representação do conhecimento. Graças ao processo descrito na abordagem proposta no presente trabalho, partes essenciais a um módulo de comunicação, baseado no uso de mensagens HL7 v3 e arquétipos, são construídas de forma genérica e permitem seu reuso em diferentes aplicações pertencentes ao domínio Cardiologia. Por exemplo, um Engenheiro de Aplicação pode recuperar quaisquer artefatos desenvolvidos na Engenharia de Domínio, tais como os *parsers* e os arquétipos.

Uma vez que o SMSCCS ainda não dispõe de um protótipo funcional, que permita a aplicação e a avaliação de aplicabilidade da abordagem, este capítulo apresenta as etapas da abordagem sendo executadas no desenvolvimento de um sistema que permite ao profissional de Saúde, especialista no implante de marca-passos cardíacos, o acesso às informações de seus pacientes de qualquer local e usando diferentes tipos de dispositivos. Apesar de ser um protótipo simples, o mesmo apresenta um requisito comum a qualquer sistema atual de apoio ao cuidado de Saúde, ou seja, interoperabilidade. Portanto, seu desenvolvimento permite que

um módulo de comunicação, baseado em características comuns a diferentes sistemas do domínio, seja criado e avaliado.

O Serviço de Acesso às Informações Médicas (SAIM) é constituído de três partes, conforme ilustrado na Figura 8.2. A primeira é composta pelos dispositivos que os médicos cardiologistas usam para acessar as informações dos pacientes, tais como *tablets*, *laptops* e *smartphones*. A segunda é composta por um servidor que disponibiliza uma interface *web*, a qual permite aos profissionais de Saúde efetuar a requisição das informações de seus pacientes via a inserção de seu identificador, do identificador do paciente, da informação que deseja consultar e da clínica que possui tais informações. Esse servidor cria então uma mensagem HL7 v3 de requisição e a envia ao servidor da respectiva clínica. Outra atribuição do servidor *web* é a apresentação das informações recebidas como resposta do servidor da clínica. Para tal, a mensagem XML recebida é processada e disponibilizada na interface de visualização. A terceira parte do SAIM é formada pelos servidores e as bases de dados das clínicas. Esses servidores, ao receberem a mensagem HL7 de requisição, efetuam a consulta apropriada em suas bases de dados e, encontrando os registros correspondentes, atribuem os dados aos objetos referentes ao conceito clínico identificado, empacotam essas informações em mensagens HL7 v3 e as enviam como resposta ao servidor *web*. No caso da consulta à base de dados não retornar as informações requeridas, os servidores criam uma mensagem HL7 de reconhecimento (*Acknowledgement - ACK*) informando o ocorrido.

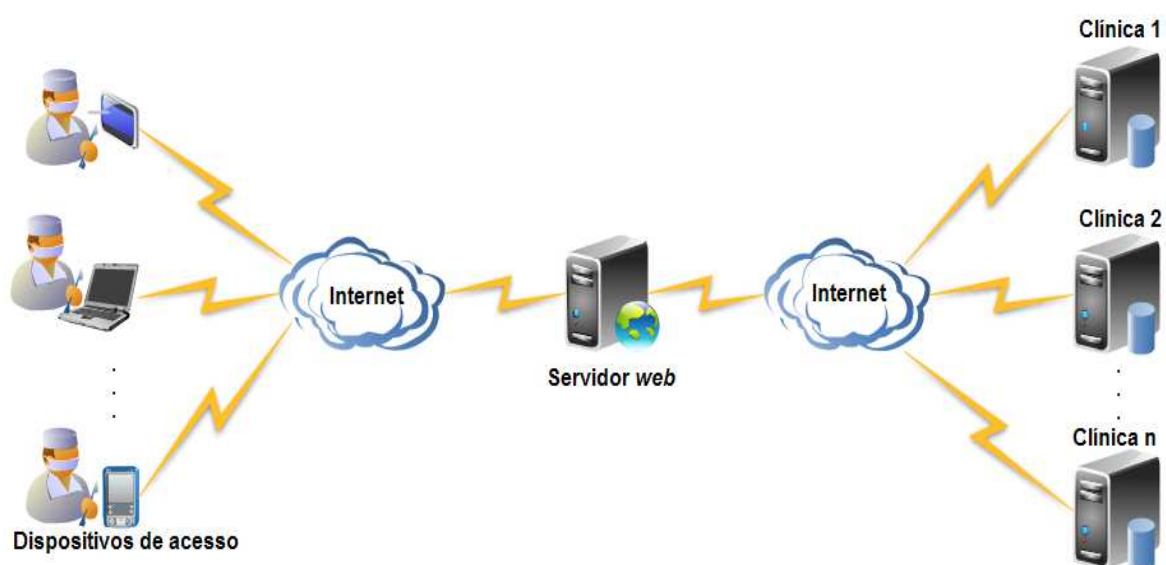


Figura 8.2 – Arquitetura do SAIM

O módulo de consulta e visualização das informações do SAIM foi desenvolvido com o auxílio do *framework Java Server Faces (JSF)* (SUN, 2010) e apresenta uma pequena adaptação de conteúdo, que permite aos usuários o acesso a partir de diferentes tipos de dispositivos. Para tal, foram construídas duas versões genéricas para as interfaces, uma correspondente à visualização via *desktops* e outra correspondente à visualização via *smartphones*. O chaveamento da versão correta a ser apresentada e a adaptação dinâmica das características das interfaces (e.g., redimensionamento de imagens, caixas de texto) são realizados graças ao *Ubiquitous Context Framework (UbiCon)* (CIRILO, 2011), desenvolvido no Grupo de Computação Ubíqua (GCU) do Departamento de Computação (DC) da UFSCar.

As próximas seções apresentam as atividades da abordagem aplicadas ao desenvolvimento do SAIM, para a criação das mensagens usadas no estudo de caso, e discussões referentes às avaliações realizadas.

8.1 Engenharia de Aplicação

Este estudo de caso possibilitou avaliar a etapa da EA com o reuso dos artefatos produzidos na ED. Serviu ainda para refinar a ED na especificação e projeto do metamodelo, na construção das transformações e na implementação dos *parsers*.

8.1.1 Analisar

A EA teve início com a especificação da aplicação via o levantamento de requisitos. Durante a análise as atividades Especificar Casos de Uso e Especificar Modelo de Interação foram também executadas. A Figura 8.3 mostra o diagrama de casos de uso, construído com o auxílio da ferramenta MVCASE, para especificar os requisitos do SAIM relacionados à inserção dos dados para a consulta, à criação da mensagem de requisição HL7 v3 e ao envio dessa mensagem para o servidor da clínica.

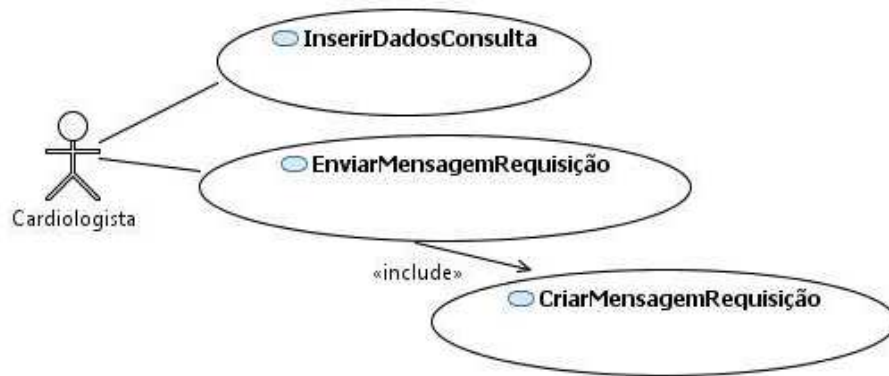


Figura 8.3 – Casos de uso do SAIM

Após o levantamento dos requisitos e a construção de seus respectivos diagramas de casos de uso, foi realizada a especificação do modelo de interação. Nessa atividade os requisitos foram refinados e apresentados na forma de diagramas de sequência. A Figura 8.4 mostra o diagrama de sequência do caso de uso EnviarMensagemRequisição, onde o ator cardiologista solicita as informações referentes ao implante de marca-passo de seu paciente. Após essa solicitação, a aplicação web cria o objeto HL7 de uma mensagem de requisição com o auxílio da MessageObjMaker e atribui as informações que o cardiologista deseja consultar com o método setMessageValues. Esse objeto HL7 é então serializado em XML pelo MessageXMLBuilder e enviado ao servidor da respectiva clínica com a chamada ao método sendMessage.

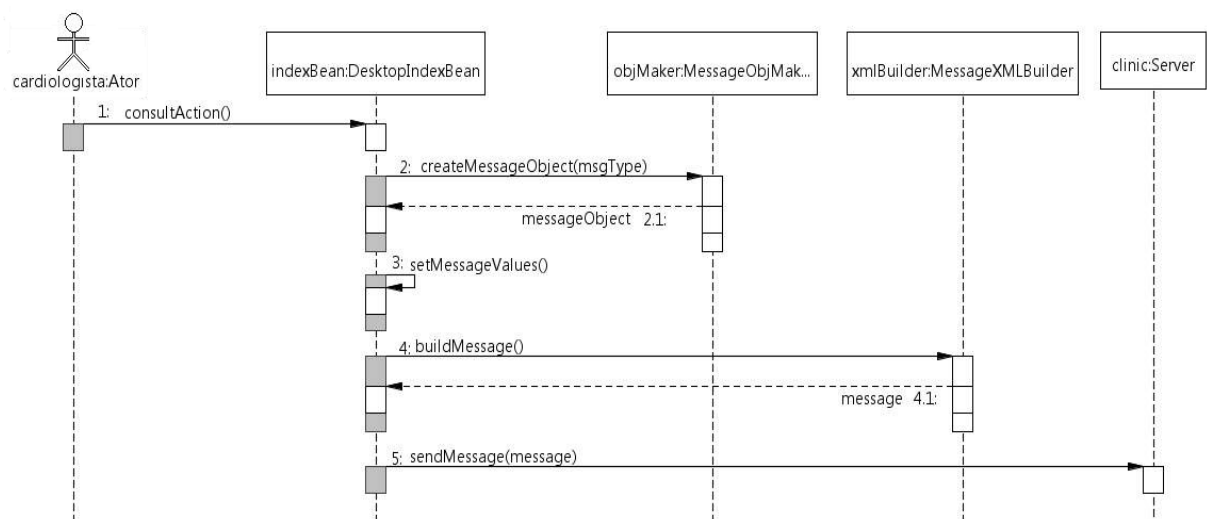


Figura 8.4 – Diagrama de sequência do caso de uso EnviarMensagemRequisição

Com o levantamento dos requisitos, foi identificada a necessidade de retorno às etapas da ED para o aprimoramento e adequação do metamodelo. Tratando-se de um sistema de acesso às informações referentes a implantes de marca-passo cardíaco, verificou-se a necessidade de inserção de novos arquétipos, os quais compreendem: informações e exames realizados no pré-operatório; informações do implante (e.g., marca e modelo dos eletrodos e do gerador, localização do implante); informações cirúrgicas; informações e exames do pós-operatório; e dados da alta médica. Dentre os conceitos identificados, os mais genéricos, tais como o da pressão arterial (openEHR-EHR-OBSERVATION.blood_pressure.v1), taxa de batimentos cardíacos (openEHR-EHR-OBSERVATION.heart_rate.v1) e registro de dados operatórios (openEHR-EHR-OBSERVATION.operation_record.v1), tiveram seus arquétipos recuperados do repositório disponibilizado pelo CKM. Conceitos mais específicos, relacionados à aplicação, tiveram seus arquétipos implementados, destacando-se: implante de marca-passo; cirurgia cardíaca, que estende o arquétipo para o registro de dados operatórios com informações específicas da cirurgia cardíaca; pós-operatório; e alta médica.

A Figura 8.5 mostra o arquétipo para o conceito implante de marca-passo cardíaco (openEHR-EHR-OBSERVATION.pacemaker_implant.v1). À esquerda da figura tem-se parte da interface gerada pela ferramenta Archetype Editor. À direita da figura tem-se um trecho da ADL correspondente. Conforme orientação dos especialistas do domínio, a descrição do conceito implante de marca-passo cardíaco deve conter o diagnóstico que convergiu para o implante, a etiologia, as informações dos eletrodos e gerador implantados, a descrição da cirurgia, procedimentos realizados e a avaliação do implante. Esse arquétipo foi desenvolvido de forma independente e não possui relacionamento com outros arquétipos⁵.

O conceito cirurgia cardíaca foi descrito no arquétipo openEHR-EHR-OBSERVATION.cardiac_operation.v1. Segundo os especialistas do domínio, as informações relevantes para a descrição desse conceito são os diagnósticos que convergiram na realização da cirurgia, os procedimentos realizados, informações sobre abertura e fechamento da cavidade, oxigenador e próteses usados, tempos de

⁵ A integração entre arquétipos foi efetuada com base na especificação, ainda em fase de desenvolvimento, dos *templates* openEHR (BEALE, 2010).

anoxia⁶ e circulação extracorpórea (CEC), inspeção das estruturas, dados de pós-operatório e dados de alta.

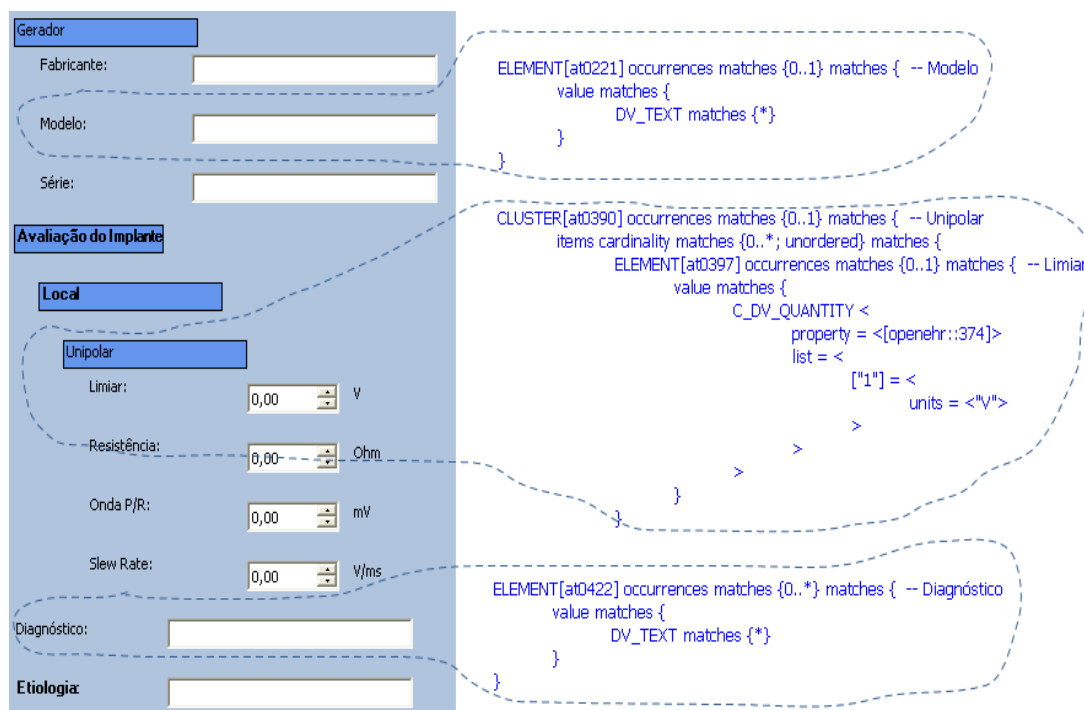


Figura 8.5 – Arquétipo para o conceito implante de marca-passo

Optou-se pela especificação dos dados do período de pós-operatório e dos dados de alta em arquétipos separados, por estes representarem conceitos que podem ser facilmente reutilizados. Dessa forma, o arquétipo especificado para o período de pós-operatório (openEHR-EHR-CLUSTER.postoperative_period.v1) apresenta as datas de entrada e saída da Unidade de Terapia Intensiva (UTI), medicamentos administrados, tempo de assistência ventilatória, drenagem mediastinal e pleural, se existe necessidade de re-operação e observações quanto a evolução do paciente. O arquétipo para os dados de alta médica (openEHR-EHR-CLUSTER.patient_discharge.v1) é composto pela data de alta, pela frequência cardíaca e pressão arterial verificadas, pelas observações do eco-cardiograma realizado, pelas observações do Raio-X do tórax, pelos exames laboratoriais realizados e observações provenientes desses exames e pelos medicamentos prescritos para o paciente. A frequência cardíaca e pressão arterial foram incorporadas via o reuso dos arquétipos recuperados do repositório do CKM.

⁶ Tempo de redução da oxigenação.

Dentre os exames relacionados ao implante de marca-passo cardíaco, tem-se o teste ergométrico que foi especificado no arquétipo openEHR-EHR-OBSERVATION.ergometric_test.v1. Esse arquétipo descreve as indicações para o teste, as queixas apresentadas pelo paciente, os exames físicos realizados, as observações do eletrocardiograma basal realizado, as medições da frequência cardíaca e da pressão arterial realizadas com o paciente em pé e sentado, a descrição do Protocolo de Bruce⁷, as verificações das alterações eletrocardiográficas por canal usado e as observações e conclusões efetuadas pelo médico. Os demais arquétipos especificados foram para os conceitos: *scan* de carótidas (openEHR-EHR-OBSERVATION.carotid_scanning.v1); eco-cardiograma Doppler (openEHR-EHR-OBSERVATION.doppler_ecg.v1); e eco-cardiograma transesofágico (openEHR-EHR-OBSERVATION.transesophageal_ecg.v1).

O Protocolo de Bruce também foi especificado em um arquétipo a parte (openEHR-EHR-CLUSTER.bruce_protocol.v1). Esse arquétipo descreve os estágios de realização do exercício em esteira, apresentando o tempo, a inclinação, a frequência cardíaca, a pressão arterial e as observações efetuadas durante cada estágio. São descritos também nesse arquétipo os estágios de recuperação, apresentando o tempo, frequência cardíaca, pressão arterial e observações.

8.1.2 Projetar

Além de identificar as tecnologias necessárias para o desenvolvimento da aplicação, tais como Java, JSF e Apache Axis (APACHE, 2010), nessa atividade foram também modeladas as mensagens pertencentes ao domínio da aplicação. Para tal, o Engenheiro de Aplicação foi auxiliado pela IDE Eclipse, pelo metamodelo e pelo editor de modelos gerados na ED.

Com base nos requisitos identificados na atividade de análise, verificou-se a necessidade de três tipos de mensagens: mensagem para busca/requisição; mensagem para o intercâmbio das informações; e mensagem de reconhecimento. Todos os tipos de mensagem possuem uma seção em comum (Message), responsável por identificar a aplicação que gerou a mensagem e a aplicação de

⁷ Protocolo em esteira comumente mais utilizado na estimativa da potência aeróbica.

destino, servindo assim como um envelope para a mensagem. A Figura 8.6 ilustra a seção Message, sendo que à esquerda tem-se a representação em árvore do modelo criado pelo Engenheiro de Aplicação e à direita tem-se o diagrama de objetos mostrando a instanciação dos conceitos contidos no metamodelo.

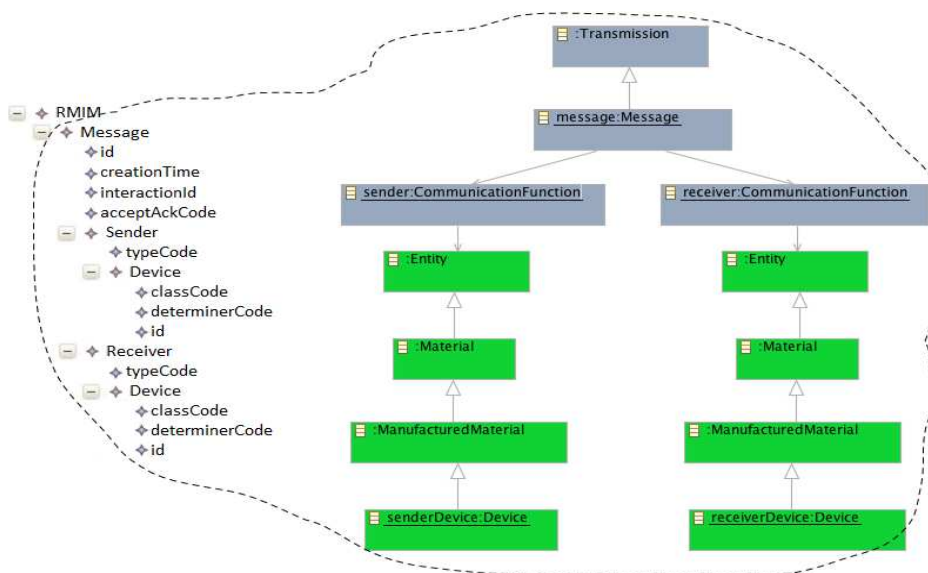


Figura 8.6 – Envelope das mensagens

A mensagem de requisição foi modelada para transportar, até o servidor da clínica, os dados do paciente a serem consultados. Esses dados são inseridos pelo profissional de Saúde via a interface *web* do SAIM. Dentre as informações constantes na mensagem estão o identificador do médico, o identificador do paciente e a informação que se deseja consultar. A Figura 8.7 mostra o modelo dessa mensagem.

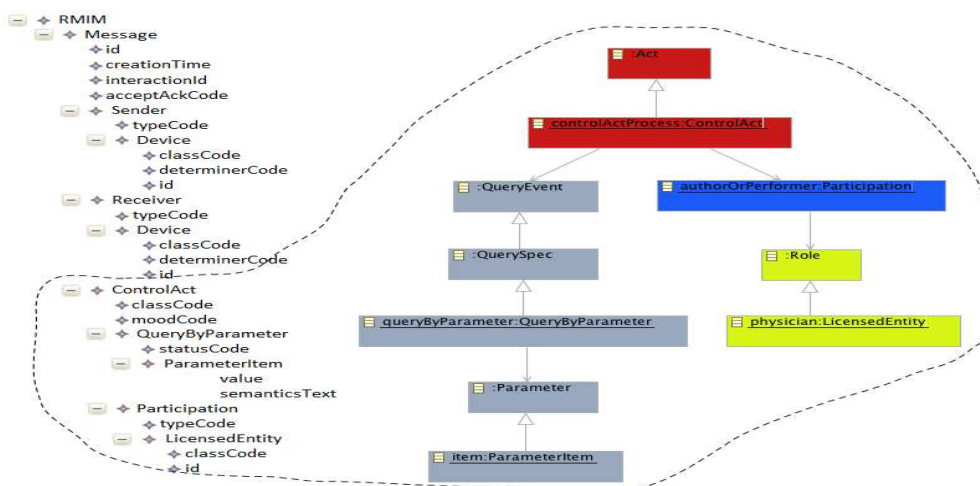


Figura 8.7 – Modelo para a mensagem de requisição

As mensagens para o intercâmbio das informações são diretamente dependentes do conceito clínico relacionado. A tomada de decisão, referente à inclusão do código SNOMED-CT do conceito durante o processo de modelagem, buscou respeitar as particularidades de cada conceito e das aplicações relacionadas. Por exemplo, em algum momento pode ser importante a identificação do profissional de Saúde que executou um determinado exame clínico. Entretanto o arquétipo, por limitar-se apenas à representação do próprio conceito, não dispõe de tais informações, as quais devem ser incluídas no corpo da mensagem HL7. Acredita-se que tal decisão pode colaborar com a interoperabilidade, uma vez que o conceito clínico não foi prejudicado pelo acréscimo de informações que não o competem. A Figura 8.8 mostra o modelo da mensagem que transporta o conceito implante de marca-passo.

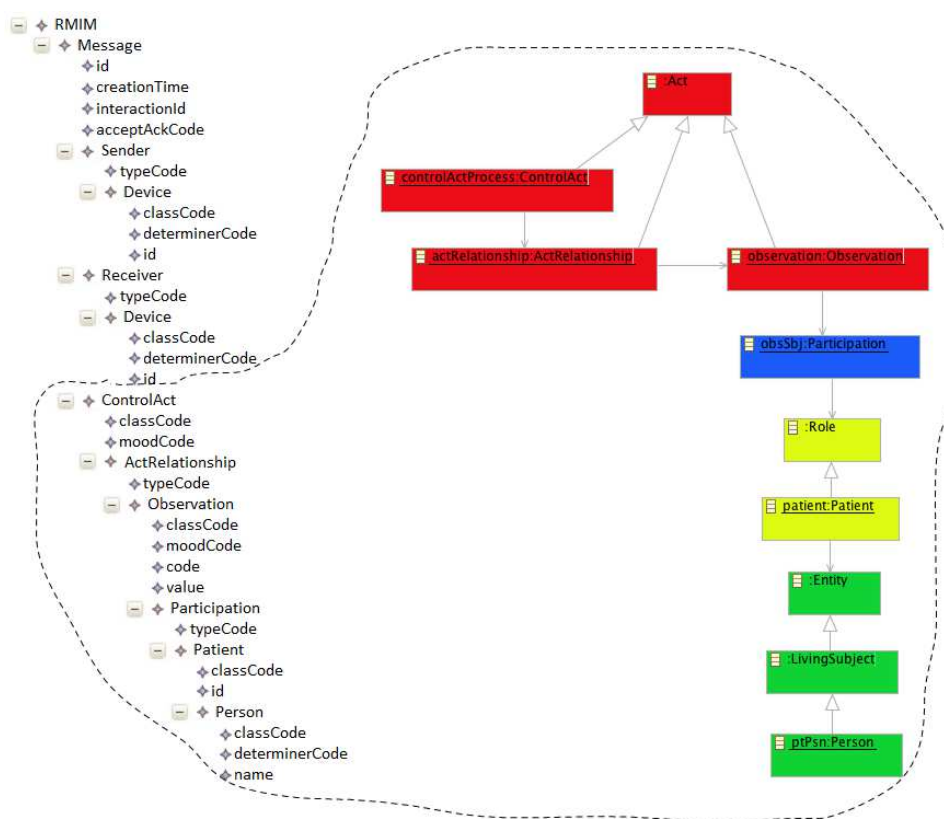


Figura 8.8 – Modelo para a mensagem do implante de marca-passo

A mensagem de reconhecimento, assim como a mensagem de requisição, baseia-se apenas no protocolo HL7 v3. Essas mensagens são enviadas a partir do servidor da clínica para informar a existência de algum erro (e.g., não ter localizado

em sua base de dados as informações requisitadas). Para tal, essa mensagem transporta principalmente o identificador da mensagem de requisição que a gerou. A Figura 8.9 mostra parte do modelo dessa mensagem.

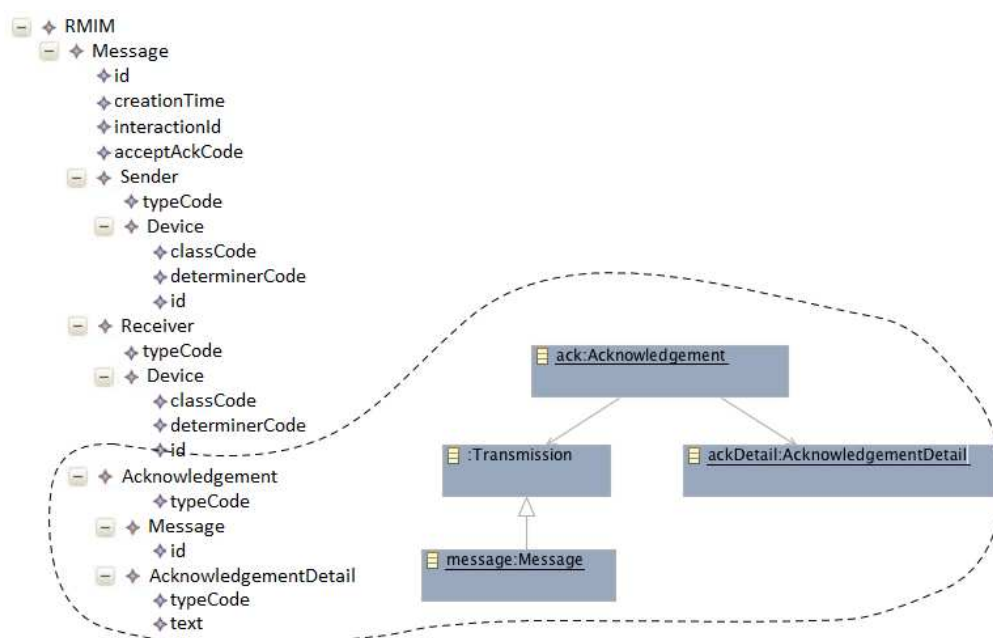


Figura 8.9 – Modelo para a mensagem de ACK

8.1.3 Implementar

Nessa atividade a aplicação é implementada, conforme as tecnologias adotadas na atividade de projeto e com base no reuso dos artefatos gerados na ED. Ademais, graças à execução conjunta das atividades tradicionais da Engenharia de Software e do processo de desenvolvimento das mensagens HL7 v3 (**Criar Esquemas das Mensagens**), são executadas também as transformações M2C para a geração dos esquemas das mensagens, tendo como entrada os modelos criados anteriormente.

Com o auxílio da IDE Eclipse, os modelos são submetidos aos *templates* JET para a criação dos respectivos arquivos MIF. A Figura 8.10 apresenta um trecho do esquema gerado para o transporte das informações do implante de marca-passo. Como é possível observar nas regiões destacadas, as transformações permitiram a criação de um esquema que contém todos os dados referentes a essa mensagem, tais como as cardinalidades e valores padrão. Um exemplo de valor padrão usado é

o código SNOMED-CT do conceito atribuído ao campo “code” da classe **Observation**.

```
<class name="Observation" isAbstract="false">
  <derivationSupplier staticModelDerivationId="1" className="Observation"/>
  <annotations/>
  <attribute name="classCode" sortKey="1" isStructural="true" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="M" defaultValue="OBS">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="classCode"/>
    <type name="CS"/>
    <supplierDomainSpecification codingStrength="CNE" domainName="ActType" mnemonic="OBS"/>
  </attribute>
  <attribute name="moodCode" sortKey="2" isStructural="true" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="M" defaultValue="EVN">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="moodCode"/>
    <type name="CS"/>
    <supplierDomainSpecification codingStrength="CNE" domainName="ActMood" mnemonic="EVN"/>
  </attribute>
  <attribute name="code" sortKey="3" isStructural="false" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="R" defaultValue="14106009">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Act" attributeName="code"/>
    <type name="ARCHETYPE"/>
    <supplierDomainSpecification codingStrength="CWE" domainName="ActCode"/>
  </attribute>
  <attribute name="value" sortKey="4" isStructural="false" minimumMultiplicity="1" maximumMultiplicity="1"
    isMandatory="true" conformance="R">
    <annotations/>
    <derivationSupplier staticModelDerivationId="1" className="Observation" attributeName="value"/>
    <type name="ED"/>
  </attribute>
</class>
```

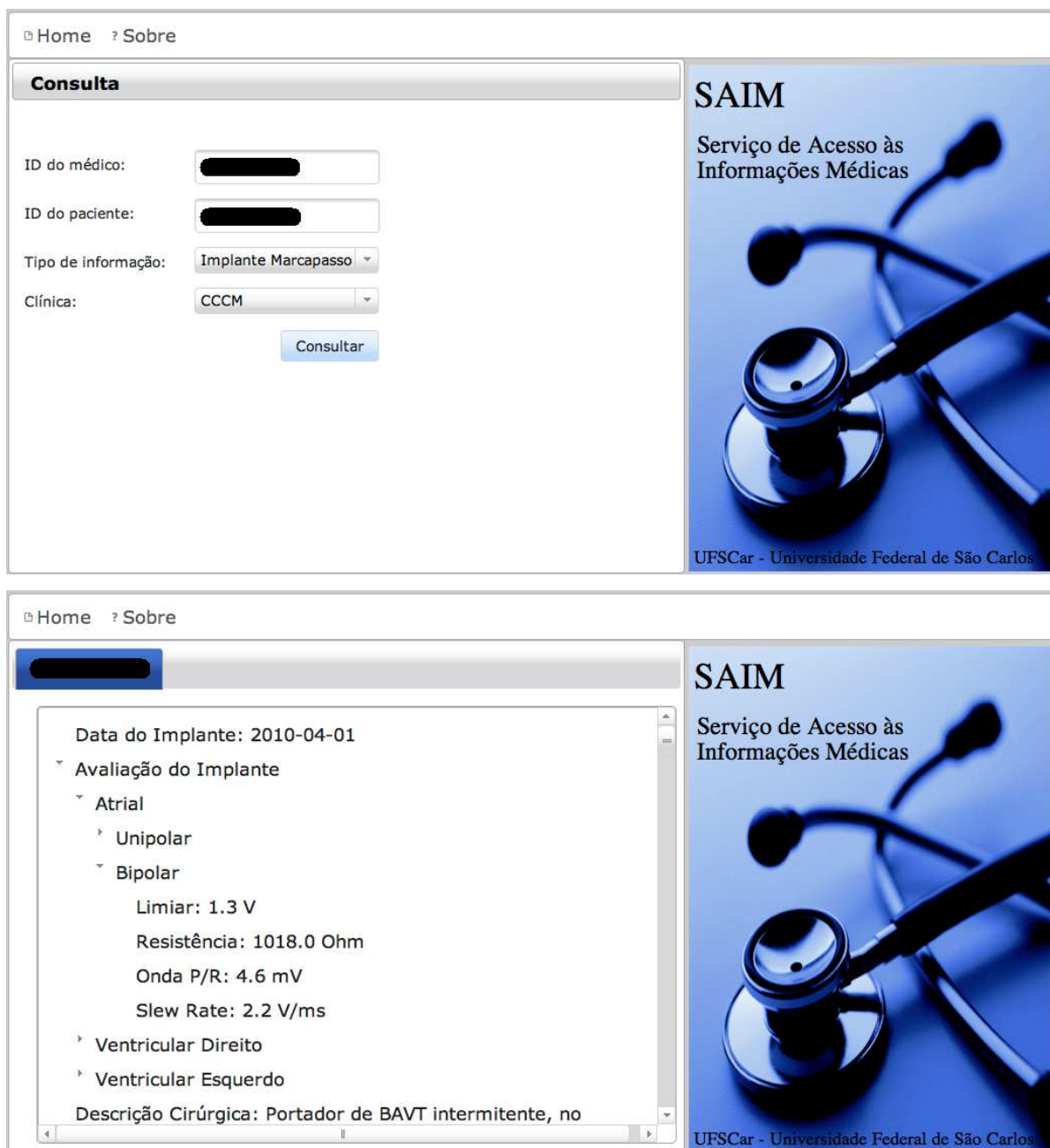
Figura 8.10 – Esquema de mensagem para implante de marca-passo

A execução da atividade de implementação permitiu a construção de um módulo de comunicação para o SAIM, que emprega arquétipos e o protocolo HL7 v3 no transporte de suas informações. Esse módulo confere ao SAIM a possibilidade de criação e interpretação dos XML das mensagens, tornando-o apto para a troca de mensagens HL7 v3 com arquétipos.

8.1.4 Testar

Para testar o funcionamento do SAIM, módulos de comunicação foram implantados também no servidor do Instituto do Coração de Marília (ICM) e do Centro de Cirurgia Cardíaca de Marília (CCCM) na forma de WebServices, permitindo assim a interpretação das mensagens de requisição e o acesso às suas bases de dados. As equipes médicas dessas clínicas vêm auxiliando diretamente no desenvolvimento deste trabalho, colaborando na especificação dos conceitos clínicos, sanando eventuais dúvidas e disponibilizando seus recursos

computacionais para testes. A Figura 8.11 exibe a visualização das interfaces do SAIM a partir do navegador de um *desktop* (a) e do navegador do *smartphone* Nokia N900 (b).



(a)

Mobile WebSAIM Sobre

Id do Médico:

Id do Paciente:

Tipo de Informação: **Cirurgia Cardíaca** ▼

Clínica: **CCCM** ▼

Consultar

Resultado

-
- Informações Cirúrgicas
- Pós Operatório
- Dados da Alta
- Diagnóstico Principal: Comunicação Interatrial.
- Diagnóstico Secundário: Hipertensão Arterial Sistêmica
- Instalação de CEC: Canulação da raiz da Aorta com cânula metálica e das veias Cavas em separado com cânulas metálicas (Pacífico); hipotermia moderada (32°), cardioplegia sanguínea anterógrada isotérmica intermitente na raiz da Ao.
- Tempo de Anóxia: 52.0 min
- Tempo de CEC: 70.0 min
- Oxigenador
- Próteses

Voltar

(b)

Figura 8.11 – Teste de execução do SAIM

8.2 Discussões

Esta seção apresenta discussões acerca das observações realizadas durante o emprego da abordagem e das avaliações realizadas no protótipo desenvolvido.

8.2.1 Aplicabilidade da Abordagem

Ao empregar a AE no desenvolvimento do SAIM, foi possível observar que as atividades do processo pertencente à abordagem proposta, bem como os artefatos gerados, colaboraram significativamente para a simplificação do desenvolvimento das mensagens HL7 v3. Uma mostra dessa simplificação está na geração dos esquemas das mensagens, realizada com o auxílio das transformações M2C. Conforme ilustra o gráfico da Figura 8.12, na geração dos esquemas para a mensagem ACK e para a mensagem de requisição, aproximadamente 88% das linhas de código (*Lines of Code* – LOC) foram geradas automaticamente via a execução das transformações M2C (194 LOC e 245 LOC respectivamente). Da mesma forma, cerca de 89% das linhas de código foram geradas automaticamente para o esquema da mensagem que transporta os dados referentes ao conceito implante de marca-passo (294 LOC). As linhas de código, que necessitaram de intervenção do Engenheiro de Aplicação, condizem principalmente com as informações mais subjetivas, tais como descrições e nomes de relacionamentos.

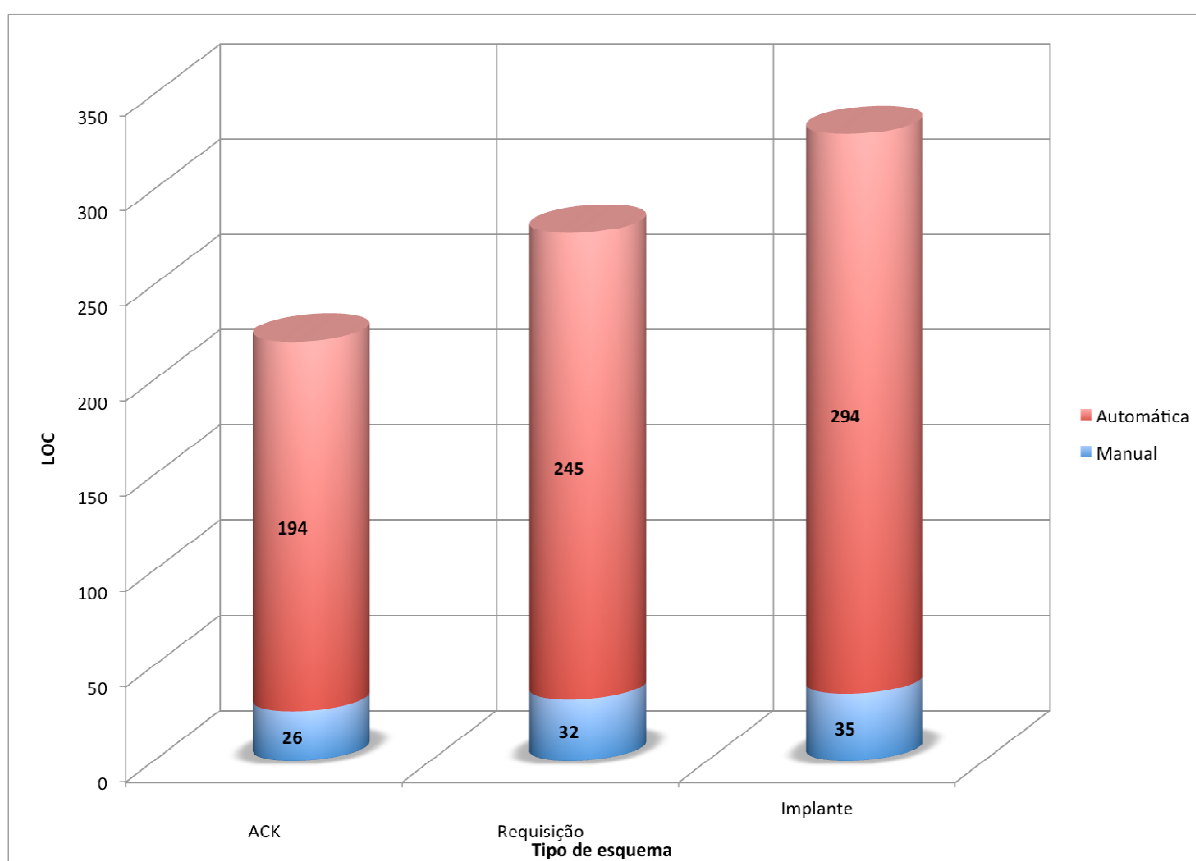


Figura 8.12 – Geração dos esquemas das mensagens

Embora a abordagem proposta omita algumas etapas existentes no processo de desenvolvimento originalmente proposto pelo comitê HL7, tais como a criação de *storyboards*, esta mostrou-se bastante intuitiva. Uma vez que a modelagem ocorre com base em termos e conceitos conhecidos do domínio Cuidado de Saúde, viabilizados pela DSL expressa via o metamodelo construído na ED, a especificação das mensagens tornou-se mais compreensível, eliminando o uso de detalhes mais técnicos relativos à implementação.

Acredita-se também que a tradução das atividades do processo de desenvolvimento das mensagens HL7 v3 para as atividades tradicionais empregadas na Engenharia de Software tenha colaborado com essa simplificação. Como já citado no Capítulo 4 desta dissertação, a curva de aprendizado relacionada ao HL7 v3 é relativamente grande. Levar essa complexidade para um cenário mais próximo do Engenheiro de Aplicação pode reduzir essa curva assim como o tempo e custo de desenvolvimento.

Outro destaque da abordagem é o uso de arquétipos na representação de conceitos clínicos. Além de permitir a associação de arquétipos durante a modelagem das mensagens, o processo permite que os mesmos sejam identificados e especificados, disponibilizando-os num repositório e colaborando com o reuso. Em princípio apenas um repositório local é abrangido pela abordagem, mas como o CKM permite que arquétipos sejam propostos e adicionados ao seu repositório, essa abordagem também pode contribuir com o seu compartilhamento.

8.2.2 Desempenho na Troca de Mensagens HL7 v3 com Arquétipos

Para avaliar o desempenho das comunicações baseadas no uso de mensagens HL7 v3 com arquétipos, optou-se pelo emprego do *Network Simulator 2 (NS2)* (NS, 2011), um simulador de eventos discretos desenvolvido especificamente para pesquisas em redes de computadores.

Uma vez que as mensagens HL7 v3 com arquétipos atuam na Camada de Aplicação, uma aplicação, que simula o comportamento dos módulos de comunicação baseados no emprego desses padrões, foi desenvolvida e incorporada ao NS2, permitindo assim o seu uso nas simulações. O comportamento do módulo de comunicação foi implementado com base no tamanho real das mensagens de requisição e resposta, usadas para as informações de implante de marca-passo

cardíaco, e por uma média aritmética entre os tempos totais verificados para o processamento da mensagem de requisição, consulta na base de dados e criação da mensagem HL7 v3 com arquétipo de resposta. Os valores citados foram obtidos através das execuções do SAIM.

Dessa forma, um nó $n0$, executando a aplicação que simula o comportamento do módulo de comunicação, é capaz de enviar uma mensagem de requisição HL7 v3 para um segundo nó $n1$, que também executa a aplicação. O nó $n1$, ao receber a requisição, aguarda o tempo de processamento necessário para a interpretação e criação da mensagem de resposta e a envia para o nó $n0$. Todas as mensagens trocadas pelos nós são enviadas com o auxílio do *Transmission Control Protocol (TCP)* e apresentam uma carga útil que representa a mensagem da aplicação empacotada por SOAP e este, por sua vez, empacotado pelo *Hypertext Transfer Protocol (HTTP)*.

O cenário desenvolvido para a avaliação leva em conta a existência de um servidor (nó $n0$), responsável pelo envio das mensagens de requisição e recepção das mensagens de resposta, de cinco servidores (nó $s1$ ao nó $s5$), responsáveis pela recepção e interpretação das mensagens de requisição e pela criação e envio das mensagens de resposta, por seis roteadores (nó $r0$ ao nó $r5$), pertencentes às instituições onde os servidores estão hospedados, e por dois gateways (nó $g0$ e nó $g1$), que fazem a conexão entre essas instituições. A Figura 8.13 ilustra o cenário desenvolvido para a avaliação, exibindo a largura de banda e o atraso definidos para cada ligação.

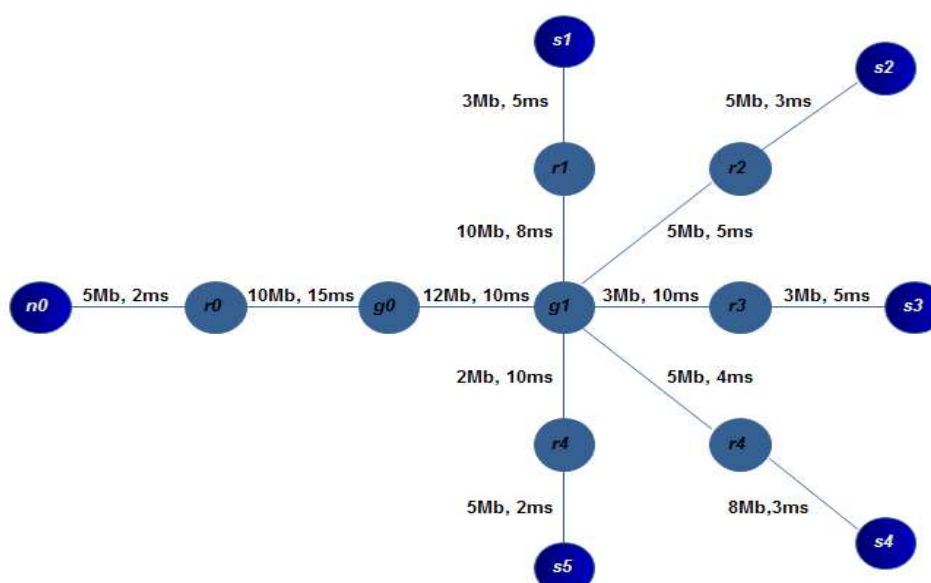


Figura 8.13 – Cenário para avaliação dos módulos de comunicação

Para efeito de comparação do desempenho da troca de mensagens HL7 v3 com arquétipos, dois outros tipos de mensagens foram usados como respostas às requisições: uma mensagem HL7 v3 pura, contendo os dados do implante de marca-passo cardíaco; e uma mensagem também com esses dados, baseada apenas nas restrições do arquétipo. Ambas as mensagens foram transportadas conforme citado anteriormente. A Tabela 8.1 apresenta as mensagens usadas pela aplicação na execução das simulações, juntamente com seus respectivos tamanhos.

Tabela 8.1 – Mensagens utilizadas nas simulações

Tipo de mensagem	Tamanho (bytes)
Requisição HL7 v3	2029
HL7 v3 pura	42561
Somente arquétipo	56641
HL7 v3 com arquétipo	71798

Buscando manter regularidade entre as execuções das simulações, o mesmo *script* foi empregado nos três casos, ou seja, o mesmo comportamento foi usado para os três tipos de mensagens de resposta. Os resultados foram obtidos com execuções equivalentes a 25 minutos de intensa troca de mensagens, tendo sido enviada uma requisição a cada 0,7 segundos.

A Figura 8.14 apresenta um comparativo entre a vazão média obtida em cada um dos casos de simulação ao longo de sua execução. A vazão indica a quantidade de dados úteis transmitidos por unidade de tempo. Como é possível observar, os três tipos de mensagem apresentaram um comportamento muito semelhante dentro do cenário proposto. Tendo em vista que a largura de banda média do ambiente é de aproximadamente 6Mb, pode-se afirmar que as mensagens apresentaram um desempenho baixo, não chegando a alcançar 1Mb/s de vazão. Dependendo da quantidade de informação que deve ser transmitida, essa baixa vazão verificada nas simulações pode representar um empecilho para o pleno uso das mensagens HL7 v3 com arquétipos no cotidiano das atividades médicas.

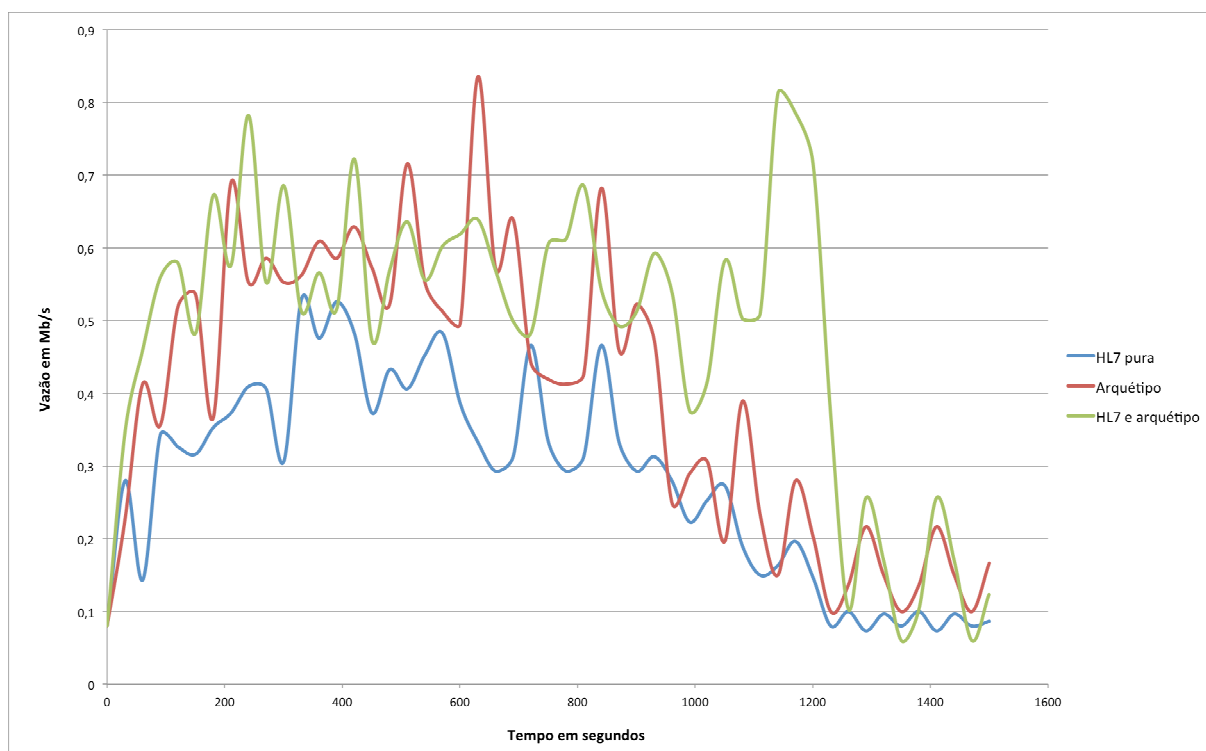


Figura 8.14 – Vazão das mensagens trocadas

Outra métrica de desempenho verificada foi o atraso (latência) de entrega das mensagens. O atraso indica o tempo médio existente entre o momento em que um quadro está apto para ser enviado e a sua recepção por um destino. A Tabela 8.2 exhibe a média dos resultados obtidos para cada caso de simulação. Como se pode observar, apesar de pequena, existe uma variação no tempo gasto para a entrega de cada tipo de mensagem. A mensagem HL7 v3 com arquetipo é a que apresenta maior atraso de propagação. Não se tratando de uma aplicação interativa, o atraso pode não representar um grande problema. Entretanto, considerando uma situação de emergência, é importante que a relação atraso *versus* vazão apresente um bom desempenho, já que o tempo gasto, para que as informações do paciente sejam apresentadas para um profissional que trata emergências, pode acarretar em consequências sérias ao paciente. Portanto, é importante manter uma infraestrutura adequada à propagação de informações, bem como buscar-se alternativas de implementação que colaborem com a melhora no desempenho do protocolo.

Tabela 8.2 – Atraso verificado nas comunicações

Tipo de mensagem	Atraso médio (segundos)
HL7 v3 pura	0,0422
Somente arquétipo	0,0425
HL7 v3 com arquétipo	0,0426

Apesar dos resultados de desempenho apresentados, uma avaliação preliminar (MENEZES et al., 2010) mostrou que estes não influenciaram na aceitação, por parte dos usuários, de aplicações que fazem uso de mensagens HL7 v3 com arquétipos, indicando que possíveis baixas de desempenho não são relevantes para o domínio abordado.

Para realizar tal avaliação, um protótipo da aplicação descrita na Seção 7.4 foi apresentado a vinte e seis pacientes hipertensos e cinco cardiologistas. Como estratégia de avaliação foi aplicado o modelo *Technology Acceptance Model (TAM)* (DAVIS, 1989). O TAM permite avaliar a utilidade e a facilidade de uso percebida pelos usuários, considerando os efeitos de variáveis externas, tais como as características da aplicação, o processo de desenvolvimento, o treinamento e a intenção de uso.

Após um período de cinco dias de uso da aplicação, formulários de avaliação baseados no TAM foram preenchidos pelos pacientes e cardiologistas. Nesses formulários os usuários especificaram seus níveis de concordância, com relação a uma determinada afirmação sobre a aplicação, conforme a escala de Likert (LIKERT, 1932). A Tabela 8.3 sumariza os resultados obtidos nessa avaliação. Para cada afirmação, a avaliação dos pacientes está localizada no topo das células e a avaliação dos médicos na parte inferior.

Uma avaliação de aceitação do SAIM também foi proposta aos médicos que auxiliaram no desenvolvimento do trabalho. No entanto, apesar de expressarem o desejo em participar, até o momento da escrita desta dissertação não foi possível encontrar tempo disponível nas agendas dos médicos para a realização dessa avaliação.

Tabela 8.3 – Resultado dos formulários submetidos aos Pacientes/Médicos

	Concorda plenamente	Concorda parcialmente	Nem concorda nem discorda	Discorda parcialmente	Discorda totalmente
1. É útil					
enviar/receber os dados por meio dessa tecnologia.	22 (84,6%)	3 (11,5%)	1 (3,9%)	-----	-----
	3 (60%)	2 (40%)	-----	-----	-----
2. O uso dessa tecnologia pode agilizar a relação médico- paciente na tomada de decisão.					
	24 (92,3%)	2 (7,7%)	-----	-----	-----
	3 (60%)	2 (40%)	-----	-----	-----
3. O uso dessa tecnologia interferiu na rotina das atividades diárias.					
	-----	-----	-----	2 (7,7%)	24 (92,3%)
	-----	-----	-----	-----	5 (100%)
4. Houve dificuldade em registrar/acessar os dados.					
	-----	-----	-----	1 (3,9%)	25 (96,%)
	-----	-----	-----	-----	5 (100%)

Capítulo 9

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Os problemas que o Cuidado de Saúde vem enfrentando estão conduzindo este domínio a uma considerável mudança de paradigma, deixando de ser centrado nos grandes hospitais e profissionais de saúde e tornando-se distribuído e centrado no paciente. Nesse contexto, o Cuidado de Saúde Pervasivo visa permitir que os pacientes tornem-se responsáveis pelo gerenciamento de sua própria saúde, reduzindo a sobrecarga dos hospitais e permitindo que os casos mais agudos de tratamento sejam atendidos com maior qualidade e com custos mais baixos.

A constante troca de informações, exigida pelos ambientes de Cuidado de Saúde Pervasivo, esbarra no problema da interoperabilidade, devido principalmente à heterogeneidade de aplicações nesses ambientes. Uma forma de reduzir os efeitos prejudiciais dessa heterogeneidade é o desenvolvimento e emprego de padrões de comunicação, os quais devem possibilitar a interoperabilidade entre sistemas heterogêneos.

Nesse sentido, este trabalho propôs uma abordagem que integra arquétipos a mensagens HL7 v3, para a comunicação de aplicações ubíquas em ambientes de Cuidado de Saúde Pervasivo. O objetivo principal, da integração dos arquétipos às mensagens HL7 v3, foi uma busca pelo aprimoramento da qualidade semântica provida por este protocolo de comunicação, o qual apresenta vários problemas para a representação de conceitos clínicos.

O emprego dessa abordagem, no desenvolvimento de aplicações ubíquas para o domínio Cardiologia, evidenciou sua eficiência, sobretudo devido ao uso da

DSL, expressa pelo metamodelo desenvolvido, e às transformações M2C. O uso de modelos, no desenvolvimento das mensagens, torna o processo mais intuitivo, abstraindo assim grande parte da complexidade existente. Da mesma forma, as transformações M2C permitem que grande parte dos esquemas das mensagens seja gerada automaticamente, reduzindo o trabalho de codificação.

Uma avaliação de desempenho foi realizada com o auxílio do *Network Simulator 2 (NS2)*, possibilitando verificar que as mensagens HL7 v3 com arquétipo podem apresentar baixo desempenho, provavelmente devido ao empilhamento de protocolos. Formas diferentes de implementação podem reduzir o impacto desse desempenho no usuário final.

Embora uma avaliação completa de aceitação do sistema SAIM, desenvolvido como estudo de caso, não tenha sido realizada, avaliações preliminares mostram que o uso de tecnologias, que estreitam a relação entre médico e paciente, são bem vistas por esses usuários. Mesmo que o meio empregado para a comunicação seja totalmente transparente, acredita-se que o fato desse meio permitir o acesso à informação de qualquer local e a qualquer momento tenha influenciado significativamente na opinião dos usuários.

9.1 Trabalhos Futuros

O uso de padrões no desenvolvimento de aplicações para o Cuidado de Saúde possibilita a expansão e o aprimoramento desse trabalho. Por exemplo, a construção de um *framework* que forneça um “esqueleto” para o desenvolvimento de aplicações ubíquas, que empregam HL7 v3 e arquétipos no intercâmbio de suas informações, para ambientes de Cuidado de Saúde Pervasivo.

Partes da proposta apresentada nesta dissertação estão sendo acrescentadas a um projeto de doutorado do PPG-CC/UFSCar, conforme citado no Capítulo 8. Portanto, está prevista a adequação e possível expansão das características aqui discutidas para viabilizar uma maior integração com esse projeto.

Existe também uma colaboração entre GCU/DC/UFSCar e o Laboratório Nacional de Computação Científica (LNCC)⁸, mais especificamente com o grupo de Mecanismos e Arquiteturas de Teleinformática (MARTIN)⁹, no contexto do Instituto Nacional de Ciência e Tecnologia (INCT) Medicina Assistida por Computação Científica (MACC)¹⁰. Há um projeto no grupo MARTIN que emprega arquétipos e consiste da criação de uma Fábrica de Software específica para o domínio Cuidado de Saúde. A ideia principal é o emprego de estratégias de modelagem para a criação de artefatos que dão suporte à geração semi-automática da estrutura básica das aplicações, as quais correspondem diretamente às reais necessidades dos *stakeholders* (e.g., médicos), intimamente ligados ao desenvolvimento das aplicações. O objetivo desse projeto é prover os mecanismos necessários, para que o processo de desenvolvimento das aplicações seja menos custoso e possa ser apoiado por profissionais de Saúde, que usam seus conhecimentos de uma determinada especialidade para enumerar as características relevantes de cada aplicação e definir os tipos de informação a serem gerenciados pela mesma.

Outro ponto importante, que pode vir a ser explorado em trabalho futuro, é o intercâmbio de dados baseados em arquétipos via o uso de extratos *openEHR*. O *openEHR Extract Information Model* (BEALE; FRANKEL, 2010) é uma especificação, ainda em fase de desenvolvimento, que propõe meios de serialização e comunicação, de partes de um Registro Eletrônico de Saúde (RES) *openEHR*, entre sistemas que implementam ou não as especificações da fundação. O emprego dos extratos *openEHR* nas comunicações, em contraste com a proposta apresentada neste trabalho, pode reduzir consideravelmente a sobrecarga gerada pela integração de especificações pouco correlatas. Além disso, o uso de “mensagens *openEHR*”, ao invés das mensagens HL7 v3, permite que todo o processo mantenha-se baseado numa arquitetura em camadas, facilitando futuras alterações e manutenções.

Pretende-se ainda realizar experimentos junto a desenvolvedores para avaliar detalhadamente a viabilidade da abordagem proposta. Para tal, será necessário familiarizar os participantes com os padrões envolvidos, bem como orientá-los

⁸ LNCC – <http://www.lncc.br>

⁹ MARIN – <http://martin.lncc.br>

¹⁰ INCT-MACC – <http://macc.lncc.br>

quanto aos meios tradicionais de uso dos mesmos. Dessa forma, tornar-se-á possível averiguar as dificuldades encontradas, tanto no desenvolvimento proposto pelas entidades quanto no proposto por esta abordagem. Todavia, uma avaliação desse tipo demandará um tempo considerável, sobretudo devido à curva de aprendizado relacionada aos padrões empregados neste trabalho.

REFERÊNCIAS

ALMEIDA, A.; BARRENHO, R. *A Computação Ubíqua: Onde estamos... para onde vamos...* 2008. Acesso em: 11/07/2008. Disponível em: <<http://alunos.di.uevora.pt/l19637/cub1/index.html>>.

APACHE. *Apache Tomcat*. Acesso em: 29/02/2010. Disponível em: <<http://tomcat.apache.org>>.

ARAÚJO, R. B. Computação ubíqua: Princípios, tecnologias e desafios. In: *XXI Simposio Brasileiro de Redes de Computadores*. Natal: [s.n.], 2003. p. 45–115.

ARNRICH, B. et al. Pervasive healthcare: Paving the way for a pervasive, user-centered and preventive healthcare model. *Methods of Information in Medicine*, v. 1, p. 67 – 73, 2010.

ATKINSON, C.; KUHNE, T. Model-driven development: a metamodeling foundation. *Software, IEEE*, v. 20, n. 5, p. 36 – 41, sept.-oct. 2003. ISSN 0740-7459.

BAHNOT, V. et al. Using domain-specific modeling to develop software defined radio components and applications. In: *Proceedings of the 5th OOPSLA Workshop on Domain-Specific Modeling*. San Diego, USA: [s.n.], 2005.

BANFAI, B. et al. Implementing an hl7 version 3 modeling tool from an ecore model. In: *Medical Informatics in a United and Healthy Europe*. [S.l.: s.n.], 2009. p. 157–161.

BARDRAM, J. E.; CHRISTENSEN, H. B. Pervasive computing support for hospitals: An overview of the activity-based computing project. *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, v. 6, n. 1, p. 44 – 50, 2007.

BARDRAM, J. E.; MIHAILIDIS, A.; WAN, D. *Pervasive Computing in Healthcare*. [S.l.]: CRC Press, 2007. 336 p.

BEALE, T. Archetype constraint-based domain models for future-proof information systems. p. 69, Agosto 2001. Disponível em: <<http://www.deepthought.com.au>>.

BEALE, T. *Archetype Object Model*. Novembro 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. *The openEHR Archetype Model: openEHR Templates*. Fevereiro 2010. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T.; FRANKEL, H. *The openEHR Information Model: Extract Information Model*. Maio 2010. Acesso em 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T.; HEARD, S. *Archetype Definition and Principles*. 2007. Acesso em: 11/05/2009. Disponível em: <<http://www.openehr.org>>.

BEALE, T.; HEARD, S. *openEHR Architecture Overview*. Abril 2007. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T.; HEARD, S. *Archetype Definition Language*. Dezembro 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. et al. *Data Structures Information Model*. Novembro 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. et al. *Data Types Information Model*. Novembro 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. et al. *EHR Information Model*. Agosto 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. et al. *Support Information Model*. Outubro 2008. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BEALE, T. et al. *Common Information Model*. Abril 2010. Acesso em: 22/03/2011. Disponível em: <<http://www.openehr.org>>.

BELL, G.; DOURISH, P. Yesterday's tomorrows: Notes on ubiquitous computing's dominant vision. In: *Personal and Ubiquitous Computing*. London: [s.n.], 2007. v. 11, n. 2, p. 133–143.

BENSON, T. *Principles of Health Interoperability HL7 and SNOMED*. 1. ed. [S.l.]: Springer, 2009. 263 p.

BITTAR, T. J. et al. Web communication and interaction modeling using model-driven development. In: *Proceedings of the 27th ACM international conference on Design of communication*. New York, NY, USA: ACM, 2009. (SIGDOC '09), p. 193–198. ISBN 978-1-60558-559-8. Disponível em: <<http://doi.acm.org/10.1145/1621995.1622033>>.

BITTENCOURT, S. A.; CAMACHO, L. A. B.; LEAL, M. do C. O sistema de informação hospitalar e sua aplicação na saúde coletiva. *Cadernos de Saúde Pública*, v. 22, n. 1, p. 19–30, Janeiro 2006.

BROWNE, E. *Archetypes for HL7 CDA Documents*. 2008. Acesso em: 15/12/2010. Disponível em: <<http://www.openehr.org/wiki/display/stds/openEHR+Archetypes+for+HL7+CDA+Documents>>.

CEN. *European Committee for Standardization*. 2009. Acesso em: 22/07/2009. Disponível em: <<http://www.cen.eu/cenorm/homepage.htm>>.

CHEN, R. *openEHR Java Reference Implementation Project*. 2010. Acesso em: 22/09/2010.

CIRILO, C. E. Model Driven RichUbi – Processo Dirigido a Modelos para a Construção de Interfaces Ricas de Aplicações Ubíquas Sensíveis ao Contexto. Dissertação (Mestrado) – Universidade Federal de São Carlos, São Carlos, 2011.

CZARNECKI, K.; EISENECKER, U. W. *Generative Programming: Methods, Tools, and Applications*. [S.l.]: Addison-Wesley Professional, 2000.

DAVIS, F. D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, v. 13, n. 3, p. 318–341, 1989.

DEURSEN, A.; KLINT, P.; VISSER, J. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.*, ACM, New York, NY, USA, v. 35, p. 26–36, June 2000. ISSN 0362-1340. Disponível em: <<http://doi.acm.org/10.1145/352029.352035>>.

DEURSEN, A. V.; KLINT, P. Little languages: little maintenance? *Journal of Software Maintenance*, v. 10, n. 2, p. 75–92, 1998.

DOLIN, R. H. et al. HI7 Clinical Document Architecture, release 2. *Journal of the American Medical Informatics Association*, v. 13, n. 1, p. 30–39, 2006.

DOLIN, R. H. et al. *Clinical Document Architecture*, release 2. 2009.

ECLIPSE. *EMFT JET Developer Guide*. 2010. Acesso em: 12/02/2010. Disponível em: <<http://help.eclipse.org>>.

FERREIRA, A. B. de H. *Dicionário Aurélio*. 4. ed. [S.l.]: Editora Positivo, 2009. 2160 p.

FIGUEIREDO, C. M. S.; NAKAMURA, E. Computação móvel: Novas oportunidades e novos desafios. *T&C Amazônia*, Manaus, v. 1, n. 2, p. 16–28, Junho 2003.

FLOGGY. *Floggy Persistence Framework*. 2009. Acesso em: 20/09/2009. Disponível em: <<http://floggy.sourceforge.net>>.

FRANCE, R.; RUMPE, B. Model-driven development of complex software: A research roadmap. In: *2007 Future of Software Engineering*. Washington, DC, USA: IEEE Computer Society, 2007. (FOSE '07), p. 37–54. ISBN 0-7695-2829-5. Disponível em: <<http://dx.doi.org/10.1109/FOSE.2007.14>>.

FREIRE, S. M. et al. Utilizando o modelo dual para a representação e persistência de contexto em aplicações ubíquas de telemonitoramento. *VIII Workshop de Informática Médica*, p. 252–255, 2008.

GFORGE. *HL7 GForge*. 2010. Acesso em: 22/09/2010. Disponível em: <<http://gforge.hl7.org/>>.

GREENFIELD, J. et al. *Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools*. [S.l.]: Wiley, 2004. 696 p.

GRONBACK, R. C. *Eclipse modeling project : a domain-specific language toolkit*. 1. ed. Addison-Wesley, 2009. Taschenbuch. ISBN 0321534077. Disponível em: <<http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0321534077>>.

HANSMANN, U. et al. *Pervasive Computing*. 2. ed. [S.l.]: Springer-Verlag, 2003. 448 p.

HL7. *Health Level Seven, Version 2.4: Final Standard*. Novembro 2000.

HL7. *HL7 V3 Guide*. Janeiro 2010. Acesso em: 27/09/2010. Disponível em: <<http://www.hl7.org/v3ballot2010jan/html/help/v3guide/v3guide.htm#v3gex1>>.

HL7. *Health Level Seven*. 2011. Acesso em: 20/01/2011. Disponível em: <<http://www.hl7.org>>.

HL7. *HL7 Version 3 Ballot Web Site*. 2011. Acesso em: 10/02/2011. Disponível em: <<http://www.hl7.org/v3ballot2011jan/html/welcome/introduction/index.html>>.

IHE. *Integrating the Healthcare Enterprise*. 2010. Acesso em: 28/07/2010. Disponível em: <<http://www.ihe.net/>>.

ISO. *International Organization for Standardization*. 2009. Acesso em: 10/07/2009. Disponível em: <<http://www.iso.org>>.

JAVASIG. *HL7 Java SIG Project*. 2010. Acesso em: 22/09/2010. Disponível em: <<http://aurora.regenstrief.org/javasig>>.

JAYARATNA, P. *Tool-Assisted Knowledge to HL7 v3 Message Translation*. Dissertação (Mestrado) — McMaster University, Canadá, 2009.

JIANG, S. et al. Robust medical data delivery for wireless Pervasive Healthcare. In: *2nd International Workshop on Smart Homes for Tele-Health (SmarTel)*. Chengdu: [s.n.], 2009.

JORGENSEN, J. B. Coloured petri nets in UML-based software development - designing middleware for Pervasive Healthcare. In: JENSEN, K. (Ed.). *Fourth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*. [S.l.: s.n.], 2002. v. 560, p. 61 – 80.

KLEPPE, A.; WARMER, J.; BAST, W. *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley, 2003. (The Addison-Wesley object technology series). ISBN 9780321194428. Disponível em: <<http://books.google.com/books?id=-uodwVPBG6YC>>.

KOHN, L. T.; CORRIGAN, J. M.; DONALDSON, M. S. To Err is Human. 2000.

LEITE, A.; GIRARDI, R. Um processo para a engenharia de domínio e de aplicações multiagente: As fases de projeto de domínio e de aplicações. In: *III Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software*. [S.l.: s.n.], 2009.

LIDDLE, S. W. *Model-Driven Software Development*. 2010. Acesso em: 07/12/2010. Disponível em: <<http://www.deg.byu.edu/papers/LiddleMDD.pdf>>.

LIKERT, R. A Technique for the Measurement of Attitudes. *Archives of Psychology*, v. 22, n. 140, 55 p, 1932.

LUCREDIO, D. *Uma Abordagem Orientada a Modelos para Reutilização de Software*. Tese (Doutorado) — Universidade de São Paulo, São Paulo, 2009.

LUCREDIO, D. et al. MVCCase Tool - working with design patterns. In: *Proceedings of Latin American Conference on Pattern Languages of Programming*. Porto de Galinhas, PE, Brasil: [s.n.], 2003.

LYYTINEN, K.; YOO, Y. *Issues and Challenges in Ubiquitous Computing*. [S.l.]: Communications of the ACM, 2002.

MADEIRA, R. N. et al. Designing a pervasive healthcare assistive environment for the elderly. *12th ACM International Conference on Ubiquitous Computing (UbiComp)*, p. 5, 2010.

MATEUS, G. R.; LOUREIRO, A. A. F. *Introdução à Computação Móvel*. Rio de Janeiro: 11^a Escola de Computação, 1998.

MCGLYNN, E. A. et al. The quality of health care delivered to adults in the United States. *The New England journal of medicine*, RAND, Santa Monica, v. 348, n. 26, p. 2635–2645, jun. 2003. ISSN 1533-4406.

MENEZES, A. L.; CIRILO, C. E.; MORAES, L. C.; SOUZA, W. L.; PRADO, A. F., Using Archetypes and Domain Specific Languages on Development of Ubiquitous Applications to Pervasive Healthcare. In: *Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems - CBMS*. Perth, 2010.

MERNIK, M.; HEERING, J.; SLOANE, A. M. When and how to develop domain-specific languages. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 37, p. 316–344, December 2005. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1118890.1118892>>.

MORAES, J. L. C.; PRADO, A. F.; SOUZA, W. L. Ambiente de computação ubíqua para o cuidado de saúde pervasivo (acucsp). In: *Anais do Seminário Integrado de Software e Hardware (SEMISH)*. Bento Gonçalves, RS: [s.n.], 2009.

NEHTA. *Review of Shared Electronic Health Record Standards*. [S.l.], 2006. Disponível em: <<http://publicaa.ansi.org/sites/apdl/Documents/Standards%20Activities/Healthcare%20Informatic>>.

NEOTOOL. *The HL7 Evolution*. 2007. Acesso em: 07/05/2009. Disponível em: <<http://www.neotool.com/pdf/HL7-Version-3-with-HL7-Version-2-History.pdf>>.

OCEAN. *Ocean Informatics*. 2011. Acesso em: 22/02/2011. Disponível em: <<http://www.oceaninformatics.com>>.

OHR, C.; VACLAVIK, M. Using hl7 processing capabilities of the open ehealth integration platform in the implementation of ihe profiles. In: ICW Developer Conference. [S.l.: s.n.], 2009. p. 9.

OMG. *Meta Object Facility Core Specification Version 2.0*. 2006. Acesso em: 12/02/2010. Disponível em: <<http://www.omg.org>>.

OPENEHR. *Clinical Knowledge Manager*. 2011. Acesso em: 10/20/2011. Disponível em: <<http://www.openehr.org/knowledge/>>.

OPENEHR. *openEHR Foundation*. 2011. Acesso em: 12/02/2011. Disponível em: <<http://www.openehr.org>>.

ORWAT, C.; GRAEFE, A.; FAULWASSER, T. Towards pervasive computing in health care - a literature review. *BMC Medical Informatics and Decision Making*, v. 8, n. 26, p. 18, Junho 2008.

POSLAD, S. *Ubiquitous Computing: Smart Devices, Environments and Interactions*. 1. ed. [S.l.]: Wiley, 2009. 502 p.

PRIETO-DIAZ, R.; ARANGO, G. *Domain analysis and software systems modeling*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1991. cap. Domain Analysis Concepts and Research Directions.

REGIO, M.; GREENFIELD, J. Designing and implementing an hl7 software factory. In: *20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Application*. San Diego, California: [s.n.], 2005. p. 7.

REGIO, M.; GREENFIELD, J.; THUMAN, B. *A Software Factory Approach to HL7 version 3 Solutions*. June 2005. Acesso em: 21/03/2010. Disponível em: <<http://msdn.microsoft.com/en-us/library/ms954602.aspx>>.

ROSS, D. T. Structured analysis: a language for communicating ideas. *IEEE Transactions on Software Engineering*, v. 3, n. 1, p. 16 – 34, 1977.

SADILEK, D. A. Prototyping domain-specific language semantics. In: *OOPSLA Companion '08: Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications*. New York, NY, USA: ACM, 2008. p. 895–896. ISBN 978-1-60558-220-7. Disponível em: <<http://dx.doi.org/10.1145/1449814.1449896>>.

SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21st century. *IEEE Computer Society*, v. 36, n. 3, p. 25–31, 2003.

SBD. *Sociedade Brasileira de Diabetes*. 2010. Acesso em: 25/03/2010. Disponível em: <<http://www.diabetes.org.br/>>.

SCHLOEFFEL, P.; BEALE, T.; HAYWORTH, G.; HEARD, S.; LESLIE, H. The relationship between CEN 13606, HL7 and openEHR. In: *Health Informatics Conference*. Sidney, Austrália:[s.n.], 2006. p. 4.

SITTIG, D. F.; ASH, J. S.; LEDLEY, R. S. The story behind the development of the first whole-body computadorized tomography scanner as told by Robert S. Ledley. *Journal of the American Medical Informatics Association*, v. 13, n. 5, p. 465–469, Setembro 2006.

SKINNER, B. J. Measuring the sustainability of provincial public health expenditure in canada. *Paying More, Getting Less 2005*, p. 28, 2005.

SNOMED. *Systematized Nomenclature of Medicine - Clinical Terms*. 2011. Acesso em: 12/01/2011. Disponível em: <<http://www.ihtsdo.org/snomed-ct/>>.

SOUZA, C. *Padronização Universal*. 2008. Acesso em: 22/11/2009. Disponível em: <<http://www.saudebusinessweb.com.br/noticias/index.asp?cod=48177>>.

STEINBERG, D. et al. *EMF Eclipse Modeling Framework*. [S.l.]: Addison-Wesley, 2008.

SUN. *Java Technology*. 2009. Disponível em: <http://java.sun.com>. Acesso em 25/10/2009.

TC215. ISO Technical Committee 215 (ISO/TC215). 2009. Acesso em: 22/07/2009. Disponível em: <http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54960>.

VARSHNEY, U. *Pervasive Healthcare Computer*, v. 36, n. 12, p. 138 – 140, Dezembro 2003.

VARSHNEY, U. *Pervasive Healthcare Computing: EMR/EHR, Wireless and Health Monitoring*. 1. ed. [S.l.]: Springer Publishing Company, 2009. 288 p.

VOELTER, M.; GROHER, I. Product line implementation using aspect-oriented and model-driven software development. *Software Product Line Conference*, International, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 233–242, 2007.

VOLTER, M. *MD* Best Practices*. December 2008. Acesso em: 17/11/2010. Disponível em: <<http://www.voelter.de>>.

WEISER, M. The computer for the 21st century. *Scientific American*, v. 265, n. 3, p. 94–104, 1991.

YEGANYAN, M. T. et al. VSked: An interactive visual schedule system for use in classrooms for children with autism. *4th International Symposium on Intelligent Distributed Computing (IDC)*, p. 4, 2010.