

**Christiano Pereira Pessanha**

**Implementando o Prontuário Eletrônico  
OpenEHR em CMS's: Uma Aproximação**

**Belo Horizonte**

**2014**

**Christiano Pereira Pessanha**

# Implementando o Prontuário Eletrônico OpenEHR em CMS's: Uma Aproximação

Tese apresentada ao curso de Doutorado do Programa de Pós Graduação da Escola de Ciência da Informação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Ciência da Informação.

Área de concentração:  
Produção, Organização e Utilização da Informação.

Linha de pesquisa:  
Gestão da Informação e do Conhecimento

**Orientador: Prof. Dr. Marcello Peixoto Bax**

**Belo Horizonte**

**2014**

Pessanha, Christiano Pereira.

P475i Implementando o prontuário eletrônico OpenEHR em CMS's  
[manuscrito]: uma aproximação / Christiano Pereira Pessanha. –  
2014.  
157 f. : enc., il.

Orientador: Marcello Peixoto Bax.

Tese (doutorado) – Universidade Federal de Minas Gerais,  
Escola de Ciência da Informação.

Referências: f. 110-120.

Anexos: f. 121-157.

1. Ciência da informação – Teses. 2. Representação do  
conhecimento – (Sistemas especialistas) – Teses. 3. Informática na  
medicina – Teses. 4. Ontologias (Recuperação da informação) –  
Teses. I. Título. II. Bax, Marcello Peixoto. III. Universidade Federal  
de Minas Gerais, Escola de Ciência da Informação.

CDU: 025.4:61



UFMG

**Universidade Federal de Minas Gerais**  
**Escola de Ciência da Informação**  
**Programa de Pós-Graduação em Ciência da Informação**

**FOLHA DE APROVAÇÃO**

"IMPLEMENTANDO O PRONTUÁRIO ELETRÔNICO OPENEHR EM CMS'S: UMA APROXIMAÇÃO"

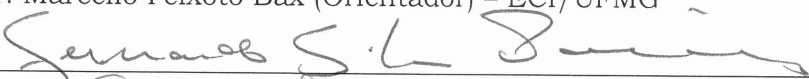
Christiano Pereira Pessanha

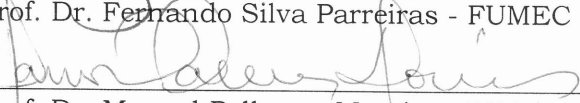
Tese submetida à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação da Universidade Federal de Minas Gerais, como parte dos requisitos à obtenção do título de "**doutor em Ciência da Informação**", linha de pesquisa "**Gestão da Informação e do Conhecimento**".

Tese aprovada em: 29 de setembro de 2014.

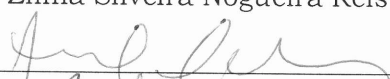
Por:

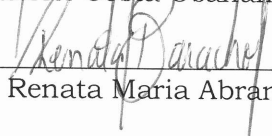
  
Prof. Dr. Marcello Peixoto Bax (Orientador) - ECI/UFMG

  
Prof. Dr. Fernando Silva Parreiras - FUMEC

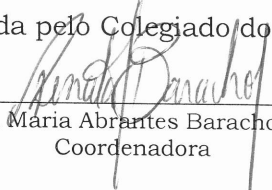
  
Prof. Dr. Manoel Palhares Moreira - PUC/MG

  
Profa. Dra. Zilma Silveira Nogueira Reis - Faculdade de Medicina/UFMG

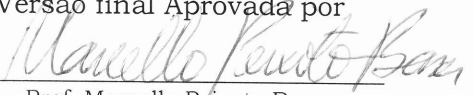
  
Prof. Dr. Gabriel Costa Osanan - Faculdade de Medicina/UFMG

  
Profa. Dra. Renata Maria Abrantes Baracho Porto - ECI/UFMG

Aprovada pelo Colegiado do PPGCI

  
Profa. Renata Maria Abrantes Baracho Porto  
Coordenadora

Versão final Aprovada por

  
Prof. Marcello Peixoto Bax  
Orientador



UFMG

**Universidade Federal de Minas Gerais**  
**Escola de Ciência da Informação**  
**Programa de Pós-Graduação em Ciência da Informação**

ATA DA DEFESA DE TESE DE **CHRISTIANO PEREIRA PESSANHA**, matrícula:  
2010654875


Às 9:00 horas do dia 29 de setembro de 2014, reuniu-se na Escola de Ciência da Informação da UFMG a Comissão Examinadora aprovada *ad referendum* pelo Colegiado do Programa de Pós-Graduação em Ciência da Informação em 11/09/2014, para julgar, em exame final, o trabalho intitulado **Implementando o prontuário eletrônico OpenEHR em CMS's: uma aproximação**, requisito final para obtenção do Grau de DOUTOR em CIÊNCIA DA INFORMAÇÃO, área de concentração: Produção, Organização e Utilização da Informação, Linha de Pesquisa: Gestão da Informação e do Conhecimento. Abrindo a sessão, o Presidente da Comissão, Prof. Dr. Marcello Peixoto Bax, após dar conhecimento aos presentes do teor das Normas Regulamentares do Trabalho Final, passou a palavra ao candidato para apresentação de seu trabalho. Seguiu-se a arguição pelos examinadores com a respectiva defesa do candidato. Logo após, a Comissão se reuniu sem a presença do candidato e do público, para julgamento e expedição do resultado final. Foram atribuídas as seguintes indicações:

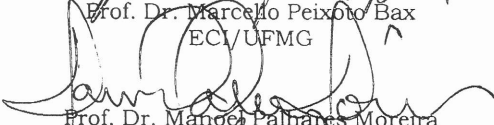
Prof. Dr. Marcello Peixoto Bax - Orientador	APROVADO
Prof. Dr. Fernando Silva Parreiras	APROVADO
Prof. Dr. Manoel Palhares Moreira	APROVADO
Profa. Dra. Zilma Silveira Nogueira Reis	APROVADO
Prof. Dr. Gabriel Costa Osanan	APROVADO
Profa. Dra. Renata Maria Abrantes Baracho Porto	APROVADO

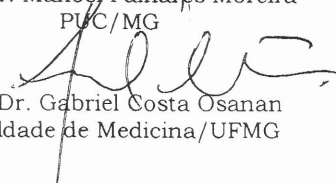
Pelas indicações, o candidato foi considerado APROVADO.

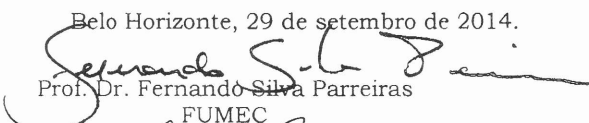
O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. Nada mais havendo a tratar, o Presidente encerrou a sessão, da qual foi lavrada a presente ATA que será assinada por todos os membros participantes da Comissão Examinadora.

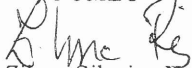
Belo Horizonte, 29 de setembro de 2014.

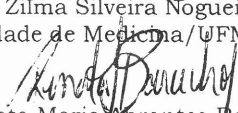
  
Prof. Dr. Marcello Peixoto Bax  
ECI/UFMG

  
Prof. Dr. Manoel Palhares Moreira  
PUC/MG

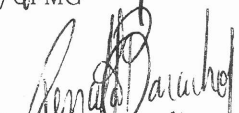
  
Prof. Dr. Gabriel Costa Osanan  
Faculdade de Medicina/UFMG

  
Prof. Dr. Fernando Silva Parreiras  
FUMEC

  
Profa. Dra. Zilma Silveira Nogueira Reis  
Faculdade de Medicina/UFMG

  
Profa. Dra. Renata Maria Abrantes Baracho Porto  
ECI/UFMG

Obs: Este documento não terá validade sem a assinatura e carimbo da Coordenadora.

  
Prof. Dr. Renata Maria Abrantes Baracho Porto  
Coordenadora do Programa de Pós-Graduação em Ciência da Informação

À minha mãe, Geralda Pereira Campos Pessanha, que, através de grande esforço, renúncia, sacrifício e fé, tornou o meu caminho possível.

Ao meu orientador, Prof. Marcello Peixoto Bax, pela oportunidade e aprendizado obtido durante o desenvolvimento desta pesquisa.

À minha tutora REUNI, Profa Renata M. A. B. Porto, pela oportunidade do convívio e aprendizado no desempenho da sempre surpreendente tarefa de ensinar.

Àqueles professores que, por seu destacado conhecimento, exemplo e amizade, fizeram deste um tempo de valioso aprendizado formal e uma rara oportunidade de convívio: Maurício Barcellos Almeida, Renato Rocha Souza, Carlos Alberto Ávila Araújo, Cátia Rodrigues Barbosa e Cláudio Paixão Anastácio de Paula.

Às sempre competentes Nely Ferreira e Gisele S. Reis, pelas incontáveis vezes em que me auxiliaram ao tornar os trâmites do curso bem mais simples de resolver do que os problemas com que lidava em minha pesquisa.

À minha esposa pelo companheirismo nas horas difíceis, em que todas as prioridades se resumiam à pesquisa.

Ao meu avô Waldemar Pereira, pelo contagiante amor aos estudos e exemplo de vida.

Aos grandes e inesquecíveis amigos Frei Francisco Pereira da Cunha (*in memorian*) e Frei Vicente Nunes (*in memorian*), pelo exemplo e incentivo que me influenciaram, fortemente, desde a mais tenra idade.

À Eide Pereira de Souza (*in memorian*), pelo inestimável incentivo e amizade.

À Maria da Conceição Pereira Dutra (*in memorian*) pelo inestimável auxílio quando do início da caminhada em minha graduação.

À Grande amiga Conceição Arreguy (e a todos os amigos do Centro José Grosso) pelo exemplo e incentivo tantas vezes transcendentais.

À Constance (*in memorian*), Juju e Liz, pela companhia e alegria em tantos momentos.



A maior parte das discussões provêm de nomes diferentes dados a um mesmo princípio pelos diversos estudiosos ou por diferentes escolas.

(Gerard A. V. Encausse, 1900)

## RESUMO

Esta pesquisa tem como motivação os desafios surgidos dos esforços que buscam atingir a interoperabilidade semântica nos Registros Eletrônicos de Saúde (RES). Mais especificamente, ela focaliza a utilização do padrão OpenEHR, desenvolvido sobretudo com vistas a viabilizar tal tipo de interoperabilidade. Argumenta-se que arquétipos (i.e., artefatos de conhecimento clínico do padrão OpenEHR) podem ser implementados em um sistema de gestão de conteúdos (CMS), com a vantagem de promover a separação entre o conhecimento de domínio (em saúde) e o código do RES, facilitando assim a interoperabilidade semântica. Outras vantagens são comentadas no texto. A pesquisa segue a metodologia de *Design Science*, que indica a realização de experimentos práticos e a geração de conhecimento teórico a partir de tais experimentos. Portanto, a título de exploração prática, a pesquisa tenta expressar os arquétipos OpenEHR no *framework* tecnológico de um CMS específico: o Plone CMS. Em primeira tentativa busca-se a expressão de arquétipos OpenEHR via ferramenta Plone/Archetypes (ferramenta de geração automática de conteúdos). Constatou-se que o custo-benefício é insatisfatório. Surgem também dúvidas sobre a suficiência do potencial expressivo da linguagem base do CMS utilizado (linguagem Python), frente à complexidade dos conceitos clínicos expressos nos arquétipos OpenEHR. Responder a tais dúvidas é fundamental para tentativas de expressar arquétipos no CMS Plone, por isso a pesquisa explora, em segunda tentativa, a implementação dos modelos de informação OpenEHR em linguagem Python (base do CMS Plone), pois estes modelos proveem a semântica necessária para a construção dos arquétipos OpenEHR. Como contribuição a tese realiza a implementação acima e abre a perspectiva de expressão dos artefatos de conhecimento clínico (Arquétipos OpenEHR) no Plone. Além de facilitar a interoperabilidade do RES implementado no CMS, esperam-se vantagens advindas da sua utilização. Com efeito, a utilização de um CMS tem custo-benefício positivo em relação à implementação de sistemas feita “do zero”. Algumas vantagens são: a geração automática de conteúdos; a gestão dos conteúdos semi-automatizada (via CRUD<sup>1</sup>); a criação facilitada de interfaces via *templates*, além de outras. A averiguação da possibilidade de expressar a semântica OpenEHR em Python sugere a viabilidade de implementações deste padrão nos diversos *frameworks* da linguagem.

---

<sup>1</sup> Acrônimo para as ações em banco de dados em língua inglesa: *Create, Read, Update, Delete*.

Além disso, a tese constata o alinhamento entre o conceito de arquétipo, chave do padrão OpenEHR, e o conceito de conteúdo, chave para os CMS's. O resultado desse esforço teórico sugere que, apesar da dificuldade inicial com a ferramenta Plone/Archetypes, existe proximidade entre estes dois conceitos. Abrindo possibilidades de expressão de arquétipos em CMS's de modo geral (não apenas no Plone).

Palavras-chave: Informática em Saúde; Registro Eletrônico de Saúde; Representação do Conhecimento; Interoperabilidade; Modelagem de dois níveis; OpenEHR; Modelo de Referência; Arquétipos; Python; Plone; CMS.

## ABSTRACT

This research has as motivation the challenges arising from efforts that seek to achieve semantic interoperability in Electronic Health Records (EHR). More specifically, it focuses on the use of OpenEHR standard, which was especially developed to facilitate this type of interoperability.

It is argued that archetypes (i.e., artifacts of clinical knowledge of the OpenEHR standard) can be implemented in a Content Management System (CMS) framework, with the advantage of promoting the separation between the domain knowledge (in health) and the EHR code, thus facilitating the semantic interoperability. Other advantages are discussed in the text.

The survey follows the *Design Science* methodology, which indicates the performance of practical experiments as well as the generation of theoretical knowledge from such experiments. Therefore, by way of practical exploration, the research attempts to express OpenEHR archetypes in the technological framework of a specific CMS: the Plone CMS. At first attempt, the expression of OpenEHR archetypes is sought by means of a Plone/Archetypes tool (tool for the automatic generation of content). It has been ascertained that the cost-benefit is unsatisfactory.

There are also doubts regarding the sufficiency of the expressive potential of the base language of the CMS used (Python language), due to the complexity of the clinical concepts expressed in the OpenEHR archetypes.

Replying to such questions is fundamental to the attempts of expressing CMS Plone archetypes, thus the research explores, in a second attempt, the deployment of OpenEHR information models in Python language (basis of CMS Plone), since these models provide all the required semantics for the construction of OpenEHR archetypes.

As a contribution, the thesis performs the previously mentioned deployment and opens the prospect of expression of the clinical knowledge artifacts (OpenEHR Archetypes) in the CMS Plone framework. Besides facilitating the interoperability of the EHR implemented in CMS, benefits arising from its use are also expected. In fact, the use of a CMS carries a positive cost-benefit when compared to the deployment of systems made "from scratch". Some advantages are: automatic generation of content; the management of content in an automated way (via CRUD<sup>2</sup>); the facilitated creation of interfaces via *templates*, in addition to others.

The investigation into the possibility of expressing OpenEHR semantics in Python suggests the viability of deployment of this standard in the various frameworks of the language.

---

<sup>2</sup> Acronym for the actions in databases in English: Create, Read, Update, Delete.

Furthermore, the thesis evidences the alignment between the archetype concept, key to the OpenEHR standard, and the content concept, key to the CMSs. The result of this theoretical effort suggests that, despite the initial difficulty with the Plone/Archetypes tool, there is equivalence between these two concepts. This opens possibilities of expression of archetypes in CMSs in general (not just in Plone).

Keywords: Health Informatics; Electronic Health Record; Knowledge Representation; Interoperability; Dual-level modeling; OpenEHR; Reference Model; Archetypes; Python; Plone; CMS.

## Sumário

<b>1 INTRODUÇÃO .....</b>	<b>1</b>
<b>2 INFORMAÇÃO CLÍNICA, REGISTRO ELETRÔNICO DE SAÚDE E O PRONTUÁRIO ELETRÔNICO DO PACIENTE .....</b>	<b>8</b>
2.1 CONTEXTUALIZAÇÃO: REGISTROS MÉDICOS EM PAPEL .....	8
2.2 O REGISTRO ELETRÔNICO DE SAÚDE .....	11
2.2.1 Evolução.....	11
2.2.2 RES <i>Versus</i> PEP.....	13
2.2.3 Interoperabilidade Semântica em RES.....	14
2.3 TERMINOLOGIAS, MODELOS DE INFORMAÇÃO E PADRÕES .....	19
<b>3 REPRESENTAÇÃO DE CONHECIMENTO E ONTOLOGIAS: O MODELO DUAL.....</b>	<b>22</b>
3.1 A ABORDAGEM DECLARATIVA PARA O PROCESSAMENTO DE CONHECIMENTO .....	22
3.1.1 O Modelo Dual.....	23
3.2 ONTOLOGIAS .....	25
<b>4 METODOLOGIA DE IMPLEMENTAÇÃO DE ARQUÉTIPOS OPENEHR .....</b>	<b>29</b>
4.1 <i>DESIGN RESEARCH</i> .....	29
4.2 ARQUÉTIPOS COMO ARTEFATOS .....	31
4.3 METODOLOGIA PARA A IMPLEMENTAÇÃO DOS ARQUÉTIPOS OPENEHR.....	32
4.3.1 Problemas Práticos e Questões de Conhecimento.....	33
4.4 TRABALHOS CORRELATOS .....	38
4.4.1 Rumo a um Registro Médico Eletrônico Interoperável e Baseado em Conhecimento via Metodologia de Arquétipos (CHEN, 2009).....	38
4.4.2 Implementação das Especificações OpenEHR em Ruby (Kobayashi; Tatsukawa, 2011).....	40
<b>5 INTEROPERABILIDADE SEMÂNTICA EM RES: O PADRÃO OPENEHR.....</b>	<b>42</b>
5.1 CONTEXTUALIZAÇÃO .....	42
5.2 O PADRÃO OPENEHR .....	43

5.2.1 Processos Clínicos e Processos de Negócio.....	44
5.2.2 A Geração de Informação nos Processos Clínicos e de Negócio.....	44
5.2.3 A Ontologia CIR.....	46
5.2.4 Uma Ontologia Voltada à Informação.....	49
5.2.5 A Hierarquia Conceitual da Ontologia CIR e a Classe <i>ENTRY</i> do Modelo de Referência OpenEHR.....	50
5.2.6 O Modelo de Conhecimento ou Modelo de Arquétipos do Padrão OpenEHR.....	53
5.3 ARQUÉTIPOS E <i>TEMPLATES</i> : A COMPONENTIZAÇÃO NO PADRÃO OPENEHR.....	60
5.4 TENTATIVA DE IMPLEMENTAÇÃO DE ARQUÉTIPOS OPENEHR EM PLONE/ARCHETYPES.....	62
<b>6 EXPRESSÃO DO PADRÃO OPENEHR EM PYTHON.....</b>	<b>64</b>
6.1 OS MODELOS DE INFORMAÇÃO: BASE PARA REPRESENTAR O CONHECIMENTO.....	64
6.2 A SEMÂNTICA DOS ARQUÉTIPOS EXPRESSA NO MODELO DE REFERÊNCIA.....	66
6.3 OS MODELOS DE INFORMAÇÃO: CARACTERÍSTICAS SEMÂNTICAS E IMPLEMENTAÇÃO.....	68
6.3.1 Modelo de Informação do Registro Eletrônico de Saúde (RES).....	68
6.3.1.1 Pacote <i>Composition</i> .....	69
6.3.1.2 Pacote <i>Content</i> .....	71
6.3.1.3 Pacote <i>Navigation</i> .....	72
6.3.1.4 Pacote <i>Entry</i> .....	73
6.3.1.5 Tentativa de Implementação de Arquétipos OpenEHR em Plone/Archetypes	76
<b>7 GESTÃO DA INFORMAÇÃO CLÍNICA DE UM RES OPENEHR VIA CMS.....</b>	<b>78</b>
7.1 QUESTÕES CONCERNENTES À IMPLEMENTAÇÃO DE PEP/RES CONFORME A ARQUITETURA COMPUTACIONAL PROPOSTA NO PADRÃO OPENEHR.....	78
7.2 <i>FRAMEWORKS</i> ORIENTADOS À METADADOS: OS SISTEMAS GESTORES DE CONTEÚDO - CMS.....	84
7.2.1 Informação Clínica e Metadados.....	84
7.2.2 Gestão do Conhecimento Clínico via Gestão de Conteúdo.....	87
7.2.2.1 Sistemas Gestores de Conteúdo (CMS): Definição e Características.....	88

7.3 ARQUÉTIPOS E CONTEÚDO: CONCEITOS DOS DOMÍNIOS CMS E OPENEHR.....	92
7.3.1 Ressalvas Quanto a Representação de Arquétipos OpenEHR .....	100
7.4. RES/PEP VIA CMS .....	100
<b>8 CONCLUSÕES.....</b>	<b>103</b>
8.1 MODELO DE REFERÊNCIA E MODELO DE ARQUÉTIPOS COMO SOLUÇÕES NA BUSCA DA INTEROPERABILIDADE EM REGISTROS ELETRÔNICOS DE SAÚDE. ....	103
8.2 VIABILIZANDO A EXPRESSÃO DE INFORMAÇÃO CLÍNICA NA PLATAFORMA PYTHON.....	104
8.3 OS SISTEMAS GESTORES DE CONTEÚDO COMO <i>FRAMEWORK</i> PARA A IMPLEMENTAÇÃO DE SOFTWARES DE RES/PEP.....	106
8.4 PERSPECTIVAS PARA TRABALHOS FUTUROS E QUESTÕES PENDENTES .....	107
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>110</b>
<b>ANEXO A - PORTARIA N 2.073, DE 31 DE AGOSTO DE 2011 .....</b>	<b>122</b>
<b>ANEXO B - MODELOS DE INFORMAÇÃO OPENEHR .....</b>	<b>128</b>



## LISTA DE FIGURAS

Figura 1 - Evolução da Tecnologia da Informação no Setor de Saúde .....	12
Figura 2 - Um Formulário de Pedido Decomposto em várias Tabelas.....	16
Figura 3 - Separação dos Modelos e Implementações no OpenEHR.....	32
Figura 4 - Estrutura de Problemas Aninhados Referentes à Trajetória da Pesquisa.....	37
Figura 5 - Metáfora da Resolução de Problemas .....	44
Figura 6 - Informações Criadas pelo Investigador Clínico .....	45
Figura 7 - Ontologia da Informação Clínica Inicial .....	46
Figura 8 - Categorias de Alto Nível .....	47
Figura 9 - A Ontologia da Informação Clínica (CIR) .....	48
Figura 10 - A Superclasse ENTRY, Baseada na Ontologia CIR.....	51
Figura 11 - A Classe ENTRY do Modelo de Referência OpenEHR, Baseada na Ontologia CIR.....	51
Figura 12 - Máquina de Estados Finitos Associada à Subclasse ACTION.....	52
Figura 13 - Construção de Arquétipos por Especialistas Médicos .....	54
Figura 14 - Elementos de uma Composition.....	56
Figura 15 - Hierarquia de Uma Composition.....	57
Figura 16 - Relação Entre os Modelos de Informação e de Conhecimento .....	59
Figura 17 - Construção de Aplicações por Especialistas em T.I. ....	59
Figura 18 - Utilização de Arquétipos em <i>Templates</i> .....	60
Figura 19 - Sequência de Artefatos Necessários para se Construir uma Interface de RES .....	61
Figura 20 - Relação Entre os Artefatos Formais do Padrão OpenEHR.....	62
Figura 21 - O Esquema Ontológico do padrão OpenEHR .....	65
Figura 22 - O Projeto da Especificação OpenEHR.....	66
Figura 23 - Modelo de Arquétipos: Meta Arquitetura.....	67
Figura 24 - O Pacote OpenEHR Composition .....	69
Figura 25 - O Mapeamento das Informações Oriundas de Eventos em Saúde para o Registro OpenEHR.....	71
Figura 26 - O Pacote OpenEHR <i>Content</i> .....	71
Figura 27 - O Pacote OpenEHR Navigation.....	72
Figura 28 - O Pacote OpenEHR Entry.....	73
Figura 29 - OpenEHR - Modelagem Dual .....	77
Figura 30 - Instância de Pressão Sanguínea na Abordagem Dual. ....	79
Figura 31 - Representação de Arquétipos em Forma de Árvore .....	80
Figura 32 - Endoscopia Gastrointestinal no Sistema GastrOS.....	80
Figura 33 - Validação de um Arquivo ADL no Processo de Parseamento de um Arquétipo .....	81
Figura 34 - Relação do AOM com a Linguagem de Arquétipos.....	82

Figura 35 - Exemplo de Importação Circular .....	83
Figura 36 - Importação Circular na Especificação OpenEHR.....	83
Figura 37 - Contexto da Correspondência entre Arquétipos e um Elemento do Domínio dos CMS's.....	92
Figura 38 - O Circulo da Gestão de Conteúdo. ....	94
Figura 39 - Correlação entre Modelo de Conteúdo e Modelo de Dados .....	98
Figura 40 - Equivalência entre Arquétipos/OpenEHR e Conteúdo/CMS.....	99
Figura 41 - O Modelo de Informação <i>Support</i> . ....	130
Figura 42 - O Pacote OpenEHR RM Identification .....	131
Figura 43 - O Pacote OpenEHR RM <i>Terminology</i> .....	137
Figura 44 - O Pacote OpenEHR RM <i>Measurement</i> .....	138
Figura 45 - O Pacote OpenEHR RM Definition.....	139
Figura 46 - O Modelo de Informação OpenEHR Data Types .....	139
Figura 47 - O Pacote OpenEHR Basic.....	140
Figura 48 - O Pacote OpenEHR Text .....	141
Figura 49 - OpenEHR Paragraph Elementos .....	142
Figura 50 - OpenEHR Paragraph Aparência Final .....	142
Figura 51 - O Pacote OpenEHR Quantity .....	143
Figura 52 - O Pacote OpenEHR Date Time .....	147
Figura 53 - O Pacote OpenEHR Time Specification.....	148
Figura 54 - O Pacote OpenEHR <i>Encapsulated</i> .....	149
Figura 55 - O Pacote OpenEHR URI .....	150
Figura 56 - O Pacote OpenEHR <i>Data_Structures</i> .....	151
Figura 57 - O Pacote OpenEHR Item Structure .....	152
Figura 58 - O Pacote OpenEHR Representation .....	153
Figura 59 - O Pacote OpenEHR History.....	154
Figura 60 - Uma History de Uma Medida de Peso .....	156
Figura 61 - Uma History de Medidas de Pressão Arterial .....	156

## LISTA DE ABREVIATURAS

AIM	–	<i>Advanced Informatics in Medicine</i>
ABNT	–	Associação Brasileira de Normas Técnicas
ADL	–	<i>Archetype Description Language</i>
CEN	–	Comité Europeu de Normalização
CIR	–	<i>Clinical Investigator Record</i>
CMS	–	<i>Content Management Systems</i>
DBC	–	<i>Design by Contract</i>
EI	–	Entidade de Informação
EMR	–	Entidade do Mundo Real
GEHR	–	<i>Good Electronic Health Record</i>
HIMSS	–	<i>Healthcare Information and Management Systems Society</i>
HL7	–	<i>Health Level 7</i>
IMR	–	Identificador do Mundo Real
ISO	–	<i>International Organization for Standardization</i>
OpenEHR	–	<i>Open Electronic Health Record</i>
OpenEHR RM	–	<i>OpenEHR Reference Model</i>
OpenEHR SM	–	<i>OpenEHR Service Model</i>
OpenEHR AR	–	<i>OpenEHR Archetype Model</i>
PEP	–	Prontuário Eletrônico do Paciente
RES	–	Registro Eletrônico de Saúde
REST	–	<i>Representational State Transfer</i>
SNOMED CT	–	<i>Systematized Nomenclature of Medicine Clinical Terms – Clinical Terms</i>
SYNOM	–	<i>Synapses Object Model</i>
UML	–	<i>Unified Modelling Language</i>
UCUM	–	<i>Unified Code for Units of Measure</i>
XML	–	<i>Extended Markup Language</i>

# 1 INTRODUÇÃO

## Contexto

O prontuário eletrônico do paciente (PEP) foi desenvolvido para que médicos e enfermeiros recuperem fatos clínicos de forma sistemática, permitindo, a todos o acesso às mesmas informações, agilizando o tratamento e diminuindo erros. O arquivamento em fichas individuais leva a problemas (Massad, 2003) como ilegibilidade, ambiguidade, equívocos de leitura, ausência e perda de informações (por extravio ou desorganização), entre outros. Tais problemas, à luz de novas tecnologias de informação e comunicação (TICs) inspiram uma mudança na forma tradicional de registrar informações em saúde para uma nova forma, onde a informação eletrônica e sua integração por meio da rede Internet figuram como essenciais. Assim, tem-se proposto o Registro Eletrônico de Saúde (RES) como meio para organizar e agilizar o registro e o acesso à informação, ou “um repositório de informação a respeito da saúde de um ou mais indivíduos numa forma processável eletronicamente” (Leão *et al.*, 2004, p.17).

Uma vez implantado, um desafio ao RES é que, cada vez mais, os registros dos pacientes se encontram distribuídos em bases de dados de diferentes sistemas de informação de hospitais e clínicas. Desse desafio adveio o esforço mundial em tornar a interoperabilidade entre esses sistemas uma realidade. Considera-se hoje importante que as instituições de saúde tenham a capacidade de trocar dados entre si, disponibilizando a informação entre profissionais e para o próprio paciente; que pode acompanhar melhor o desenrolar de seu tratamento (Santos, 2011). Torna-se, portanto, essencial aos sistemas de RES buscar a sua adaptação a padrões como, por exemplo, o OpenEHR, HL7<sup>3</sup> e a norma ISO 13606<sup>4</sup>. O governo brasileiro, através da Portaria 2073 de 31 de Agosto de 2011, por sua vez, demonstra estar partícipe dessa busca (veja Anexo A).

O padrão desenvolvido pela fundação OpenEHR<sup>5</sup> foi proposto visando, sobretudo, permitir a interoperabilidade entre os sistemas de RES (Beale; Heard, 2008, p. 11). A ideia é

---

<sup>3</sup>A organização *Health Level Seven* (HL7) desenvolveu um padrão conhecido internacionalmente, utilizado para intercâmbio de dados na área da saúde: <<http://hl7.org>>.

<sup>4</sup>A norma ISO 13606 é um padrão internacional para a interoperabilidade de sistemas de RES:<<http://www.en13606.eu>>.

<sup>5</sup>Fundação *Openehr*: <<http://www.openEHR.org>>.

representar os conhecimentos clínicos de modo estruturado, mantendo-os como objetos externos ao *software*. Tais objetos, denominados "arquétipos" (padrões de metadados), representam informações de várias naturezas; padronizando e organizando os dados do domínio de conhecimento. Como são mantidos externos aos códigos do sistema, especialistas da área médica podem gerenciá-los de forma mais independente dos analistas de sistemas, e especialistas de TI. O objetivo é que os próprios profissionais de saúde representem conceitos complexos, como "pressão sanguínea" ou "histórico familiar", sobretudo através do reuso ou da definição de novos arquétipos. Talvez o maior fator motivador da adoção de arquétipos seja a perspectiva do reuso do conhecimento clínico bem especificado e validado por organizações de referência (Nardon *et al.*, 2008), extremamente especializado e caro.

### **Problema**

Para a construção de arquétipos reutilizáveis, o padrão especifica um **modelo de referência**<sup>6</sup>, núcleo estável e genérico que define conjuntos de classes, relações e atributos que representam os conceitos do domínio em baixo nível de abstração. São os blocos genéricos para se construir os arquétipos. (Beale, 2002).

E, uma vez que o modelo de referência é a base para a criação dos modelos de conhecimento, qualquer esforço de implementação do OpenEHR, numa plataforma de programação deve primeiro responder à questão: **é possível expressar informação clínica conforme o padrão OpenEHR nesta plataforma?** É alinhado a esse esforço de pesquisa que este trabalho se propõe colaborar, num primeiro momento, ao verificar a possibilidade de expressão de informações clínicas no padrão OpenEHR, na linguagem de programação Python<sup>7</sup>.

Num segundo momento, uma vez verificada a possibilidade de expressar a semântica do padrão OpenEHR em Python, busca-se analisar os ganhos trazidos à gestão da informação clínica por sua implementação em **sistemas gestores de conteúdo** (CMS). Para tanto, deve-se responder à seguinte questão de pesquisa: **que relação há entre o conceito de arquétipo**

---

<sup>6</sup> Na especificação OpenEHR, o modelo de referência é composto de vários modelos de informação (como: *data\_types*, *data\_structures*, *support*, *EHR*, *Demographic*, entre outros), porém, a mesma expressão "modelo de informação" é utilizada, de maneira mais abrangente, pelo paradigma do modelo dual para designar seu primeiro nível, constituído pela referência estável que será implementada em software. Por sua vez, o segundo nível será responsável pela definição formal dos conceitos clínicos via arquétipos (que se utilizam das classes dos modelos de informação) e templates (BEALE; HEARD, 2008).

<sup>7</sup> Linguagem Python, página institucional: < <http://www.python.org/> >.

## **OpenEHR, próprio do domínio clínico e o conceito de conteúdo, próprio do domínio dos CMS's ?**

É no contexto especificado acima que esta pesquisa apresenta contribuições para o problema genérico da gestão de informação em saúde.

### **Justificativa**

Um dos grandes motivadores para a adoção de arquétipos na construção de aplicações é a perspectiva de se reutilizar conhecimento clínico bem especificado e validado por organizações de referência. Torna-se, portanto, essencial aos programas de RES, que buscam interoperabilidade, a adaptação a padrões voltados a este fim, como o OpenEHR.

Conforme Kobayashi e Tatsukawa (2012), as atuais implementações do padrão OpenEHR, considerando-se as estatísticas de adoção das linguagens utilizadas, proveem recursos para aproximadamente 25% dos desenvolvedores de *softwares*, desta forma, implementações adicionais mostram-se necessárias para tornar mais ampla a base de usuários.

Assim, um resultado positivo quanto ao problema expresso pela primeira questão de pesquisa permitirá que aplicativos voltados ao gerenciamento de informações clínicas possam ser desenvolvidos em mais uma linguagem. Pode-se, dessa forma, aproveitar os recursos da plataforma de desenvolvimento Python, que vem apresentando crescimento nos últimos anos, em número de usuários e em recursos, conforme indicam os índices de classificação<sup>8</sup>.

Por sua vez, uma resposta positiva ao problema da segunda questão de pesquisa permitirá o avanço teórico de alinhar os conceitos de arquétipo/OpenEHR e conteúdo/CMS, até então, disjuntos. Permitirá, também, realizar uma análise, que possibilite aos proponentes de tais sistemas, decidir entre uma implementação cuja codificação parta do zero e a utilização dos CMS's. Um *framework* sistêmico como um CMS possibilitaria a gestão de conteúdos (neste caso: arquétipos) através de uma interface amigável, além de acrescentar recursos como a separação do conteúdo de sua apresentação, indexação dos conteúdos, gestão avançada de usuários, entre outros.

### **Objetivos, Hipóteses e Pressupostos**

Ao se iniciar este trabalho de pesquisa, alguns pressupostos foram assumidos como

---

<sup>8</sup> Índice TIOBE, página institucional:< <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>

norteadores para as escolhas entre os inúmeros caminhos passíveis de se seguir durante seu desenvolvimento:

- O desenvolvimento de um sistema de RES apresenta desafios inerentes à dinâmica e à complexidade da informação médica, manutenção evolutiva, interoperabilidade etc.
- A separação entre o conhecimento clínico a ser utilizado pelo sistema de RES dos detalhes envolvidos na implementação, permite melhor enfrentar os desafios acima citados.
- Padrões internacionais voltados ao registro da informação clínica são considerados parte importante para o enfrentamento dos desafios envolvidos no desenvolvimento de sistemas de RES.
- O OpenEHR facilita o intercâmbio de informações clínicas sobre uma **base semântica** comum.
- Um *framework* sistêmico voltado à gestão de conteúdos (um CMS) possui propriedades que são adequadas à representação da informação clínica, e este trabalho de pesquisa busca correlacionar estes domínios de forma a esclarecer seus pontos em comum e sua influência numa possível solução de implementação.
- Implementar o OpenEHR com auxílio de um CMS apresenta vantagens diversas decorrentes dos pressupostos acima, algo que este trabalho de pesquisa tem por objetivo revelar.

De modo sucinto: arquétipos (ou artefatos de conhecimento clínico do padrão OpenEHR) podem ser implementados num *framework* sistêmico voltado à gestão de conteúdos (CMS) com o acréscimo de promover a separação do conhecimento da codificação, facilitando assim obtenção da interoperabilidade. Isso apresenta vantagens diversas frente a uma implementação feita a partir do zero.

Alinhado com os pressupostos assumidos, este trabalho de pesquisa tem o objetivo geral de possibilitar, na plataforma Python/Plone<sup>9</sup>, a expressão de arquétipos em conformidade com a semântica definida pelo padrão para interoperabilidade de dados clínicos OpenEHR. A escolha do Plone/CMS para tal esforço deve-se a algumas características, como

---

<sup>9</sup> Plone: < <http://plone.org/> >

ser construído sobre o servidor ZOPE, utilizar o banco de dados OO ZODB e os *frameworks* CMF e ZMI, cujas propriedades vão de encontro a objetivos como: expressar arquétipos como conteúdos, vantagens em utilizar bancos de dados não-relacionais para expressar informação clínica, workflow, versionamento, entre outras. O detalhamento da motivação para a escolha do Plone/CMS será feito no Capítulo 7. Para tanto, busca atingir os seguintes objetivos específicos:

**O1:** Verificar se os modelos (de referência e de arquétipos) do padrão OpenEHR podem ser expressos na plataforma de desenvolvimento Python. Uma vez respondida esta questão - referente à explicitação e comparação dos formalismos sintáticos e semânticos da plataforma Python e do padrão OpenEHR - abre-se caminho para a criação de artefatos de conhecimento compatíveis com este padrão por sistemas construídos sobre essa plataforma de programação.

**O2:** Verificar a possibilidade de se utilizar um *framework* como o dos sistemas gestores de conteúdo ou CMS's para a implementação de sistemas do tipo PEP/RES, conforme o padrão OpenEHR.

Dessa forma, tomando como base os pressupostos e objetivos anteriores a pesquisa desenvolvida busca argumentar as seguintes hipóteses:

**H1** - A semântica proposta pelo padrão OpenEHR para a expressão de informações clínicas pode ser implementada na plataforma Python.

**H2** - Um *framework* sistêmico voltado à gestão de conteúdos (um CMS) possui propriedades que são adequadas à representação da informação clínica. Assim, implementar o padrão OpenEHR com auxílio de um CMS apresenta vantagens diversas frente a uma implementação feita a partir do zero.

**H3** - Há proximidade semântica entre o conceito de arquétipo do padrão OpenEHR e o de conteúdo dos CMS's.

## **Metodologia**

A necessidade de lidar com questões de natureza prática e teórica, aninhadas e



interdependentes em suas soluções, levou à adoção do *design research*<sup>10</sup> como paradigma epistemológico norteador da trajetória metodológica desta pesquisa. A abordagem permite, no caso específico de sistemas de informação, trabalhar com a criação de novos conhecimentos através da concepção de algoritmos, interfaces, metodologias, entre outros resultados práticos.

Apointa-se, assim, a possibilidade de se trabalhar com conhecimentos caracterizados como formais, bem como materiais ou empíricos. De fato esta pesquisa, em sua primeira parte, busca possibilitar a expressão de arquétipos via semântica OpenEHR em Python, ou seja, um esforço **empírico**. A seguir, sugere a equivalência conceitual entre arquétipos/OpenEHR e conteúdos/CMS, num esforço **teórico**. Tal contexto, permeado de problemas de diferentes naturezas, aninhados e influenciando suas soluções de modo encadeado, mostra-se consistente com a metodologia *design research*.

### **Contribuições da Tese**

A trajetória de pesquisa percorrida possibilitou apresentar contribuições em diferentes níveis para a gestão de informação em saúde.

Do ponto de vista tecnológico, a expressão da semântica OpenEHR na linguagem Python abre a perspectiva de expressão dos artefatos de conhecimento clínico OpenEHR nesta plataforma, bem como para implementações e provas de conceito em seus diferentes ambientes e *frameworks*.

Do ponto de vista teórico, a análise das características e do uso de CMS's como opção de implementação de RES oferece subsídios e amplia o leque de opções dos desenvolvedores de diversas plataformas em projetos deste tipo. Isso permite a criação de conteúdos via *frameworks* especificamente projetados para esse fim, oferecendo àqueles que avaliam gerir informações clínicas via CMS's a possibilidade de criar interfaces flexíveis e reutilizáveis de modo rápido e com menor dispêndio de recursos. Com efeito, a utilização de um CMS apresenta melhor custo-benefício frente à implementação de sistemas feita “do zero”, sendo que geração automática de conteúdos; gestão automatizada de conteúdos e o uso de *templates* na criação de interfaces, caracterizam algumas de suas vantagens.

---

<sup>10</sup> Inicia-se, formalmente, com a obra de Simon (1969), quando este destaca a importância de se desenvolver uma ciência que se dedique ao estudo dos artefatos produzidos pelo homem e como projetá-los de maneira satisfatória.

## **Estrutura do Trabalho**

Definiu-se a ordem de apresentação e temáticas de cada capítulo da seguinte maneira: o Capítulo 2 descreve conceitos essenciais para o desenvolvimento da argumentação deste trabalho, tratando da Informação Clínica e Registro Eletrônico de Saúde. O Capítulo 3 trata da Representação de Conhecimento e Ontologias. O Capítulo 4 apresenta a *design research* como base metodológica na condução da pesquisa. O Capítulo 5 trata do tema da Interoperabilidade Semântica em RES e o padrão OpenEHR, como apresentará os trabalhos relacionados com este projeto de pesquisa. No Capítulo 6 será apresentado o resultado do esforço de estudo e implementação da especificação do padrão OpenEHR, com maior foco nas especificações referentes ao seu modelo de referência, responsável por prover a semântica do padrão seguido da implementação. Uma vez concluída com sucesso, esta implementação permite responder satisfatoriamente à questão central da pesquisa: se é possível expressar informações clínicas conforme tal semântica na linguagem de programação Python.

Em seguida, uma vez obtida resposta positiva à questão posta no capítulo anterior, o Capítulo 7 analisa as possibilidades de utilização de CMS's na implementação de *softwares* destinados a gerir informação clínica, contrapondo o uso de tais sistemas à implementação tradicional feita somente via codificação direta (a partir “do zero”). Este esforço visa traçar um cenário de implementação alternativo para aqueles que estejam analisando os requisitos para o desenvolvimento de softwares que acessem e manipulem informações clínicas. As características específicas desejáveis para tal tipo de *software*, somadas às associadas a esse tipo específico de informação são analisadas em paralelo às características de um CMS. Espera-se, assim, oferecer subsídios para os tomadores de decisão quanto a futuras implementações do tipo.

Finalmente, no Capítulo 8 são apresentadas as conclusões da pesquisa, bem como as perspectivas de trabalhos futuros abertas pelos resultados teóricos e tecnológicos obtidos.

## **2 INFORMAÇÃO CLÍNICA, REGISTRO ELETRÔNICO DE SAÚDE E O PRONTUÁRIO ELETRÔNICO DO PACIENTE**

Este capítulo busca contextualizar o domínio dos registros em saúde, partindo de um breve histórico dos registros em saúde, sua evolução a partir de anotações em papel, discutindo sua função e problemas advindos da prática de registrar em prontuários de papel, bem como a possibilidade de transcender tais problemas via adoção de novas tecnologias propondo-se o registro eletrônico em saúde. A seguir, um histórico do RES é apresentado, ressaltando-se a evolução de suas funções e a crescente abrangência do seu uso nas instituições de saúde com o passar dos anos sendo feita, em seguida, a diferenciação entre RES e PEP.

Discute-se, a seguir, como a evolução dos sistemas de registro computadorizados levou ao aumento da complexidade informacional devido à distribuição das informações dos pacientes em diferentes bases de dados e a possibilidade da utilização das novas tecnologias na busca por uma solução. A heterogeneidade da informação causada pela diversidade de modelos de dados é discutida, assim como os conceitos de integração, interoperabilidade e interoperabilidade semântica. Ao fim, propostas de formas para se obter a interoperabilidade semântica são apresentadas, como o uso de terminologias e com especial destaque para o uso de modelos de informação genéricos ou de referência que, base para a modelagem em dois níveis, abordagem advinda da inteligência artificial, apresenta-se como promissora para lidar com os desafios deste contexto.

### **2.1 Contextualização: Registros Médicos em Papel**

Registros médicos contendo descrições de detalhes de uma cirurgia bem como prescrições de medicamentos e tratamentos foram encontrados em papiros do antigo Egito (Tan, 2005), indicando que a importância dada pelos médicos ao ato de registrar os dados referentes ao tratamento dos pacientes não é algo apenas da atualidade, mas permeia a atividade médica através da história. No século V A.C., a forma de se realizar o registro médico foi altamente influenciada por Hipócrates que, segundo van Bommel e Musen (2000), asseverava dois objetivos para o registro médico:

1 - Deveria refletir, de modo acurado, o curso da doença.

2 - Deveria indicar as possíveis causas da doença.

Tais registros continham descrições dos eventos que precediam as doenças dos pacientes, buscando esclarecer as suas reais causas, registrando as observações dos médicos numa ordem cronológica, no que se poderia chamar de registro orientado ao tempo. Até o século XIX, segundo os mesmos autores, os médicos baseavam suas observações no que ouviam, sentiam e viam. Porém, com a invenção do estetoscópio em 1816 e novos instrumentos para diagnósticos como o oftalmoscópio<sup>11</sup> e o laringoscópio<sup>12</sup> ampliou-se o escopo dos registros cuja ênfase estava, anteriormente, nas informações obtidas do paciente e seus familiares para as informações obtidas pelo médico e enfermeiras, tornando necessário, para fins de registro, o desenvolvimento de uma nova terminologia capaz de expressá-las.

Apesar da antiguidade da prática do registro médico a forma utilizada para fazê-lo sempre foi a da anotação manual em papel. Portanto, sob tal contexto, pode-se considerar o desenvolvimento dos computadores e a consequente perspectiva de se registrar informações clínicas em meio digital como um ponto de inflexão na evolução da prática do registro médico.

O modelo tradicional de arquivamento em fichas individuais leva, facilmente, à ocorrência de erros, podendo prejudicar o trabalho da equipe de saúde, conforme Bleich e Lawrence<sup>13</sup>:

O registro médico [...] muito frequentemente distribui mal o espaço, é esfarrapado, desorganizado e ilegível; notas de progresso, notas de consulta, relatórios radiológicos e notas das enfermeiras todas misturadas numa sequência ascendente. Estes registros confundem mais do que esclarecem; eles provêm um proibitivo desafio a qualquer um que tente entender o que está acontecendo com o paciente (BLEICH; LAWRENCE, 1993, apud BENSON, 2010, p.11-12, tradução nossa).

Conforme Massad (2003), algumas das principais desvantagens dos prontuários em papel são:

---

<sup>11</sup> O **oftalmoscópio** é um instrumento utilizado para a observação das estruturas oculares, mais especificamente na fundoscopia ou oftalmoscopia (observação do fundo do olho). O oftalmoscópio é fundamental para o diagnóstico de doenças como glaucoma, catarata, além de lesões e tumores na retina. Costa (2012).

<sup>12</sup> O **laringoscópio** é um instrumento utilizado para o exame da laringe. Costa (2012)

<sup>13</sup> BLEICH; LAWRENCE, 1993 apud BENSON, 2010, p. 11-12.

- Só pode estar em um lugar ao mesmo tempo - pode não estar acessível ou mesmo ser perdido.
- Conteúdo é livre, sem ordem definida, algumas vezes é ilegível, incompleto e com informação ambígua.
- Para estudos científicos, o conteúdo precisa ser transcrito, o que predispõe ao erro.
- As anotações em papel não podem disparar lembretes de alerta aos profissionais.

Hamilton (2011, p.6), levanta alguns dos erros dos prontuários em papel, como: “Categorias inteiras de erros médicos e potenciais erros podem ser eliminados, incluindo erros de transcrição, erros advindos da má compreensão de abreviaturas bem como de medicamentos que possuam nomes semelhantes”.

Arquivos em papel encontram-se unicamente localizados e podem não ser recuperados quando necessários, escrita pobre, ilegível, ou incompleta, datação incompleta ou ausente e notas tão ambíguas que levam a interpretações errôneas estão entre algumas das limitações do registro em papel que van Bommel e Musen (2000) apontam. Realizar uma pesquisa retrospectiva num largo número de registros em papel é uma tarefa extremamente laboriosa e sendo que muitos dados obtidos neste esforço podem estar faltantes ou inutilizadas. Porém, para estes autores, a principal limitação do registro em papel é que estes últimos podem contribuir apenas de modo passivo para o processo de decisão do médico, não podendo de modo ativo informar ao clínico sobre alguma anormalidade em um exame laboratorial, contra indicações ou alergias do paciente a certas medicações. Notas em papel não conseguem trazer, de modo ativo, a lembrança de informações, advertências e recomendações.

Da mesma forma, Lusk coloca que:

Todos os processos da medicina são orientados pela documentação coletada na hora do encontro. Todos, médicos, laboratórios, companhias de seguro, etc., necessitam das informações que os médicos coletam. O médico necessita delas para nas visitas subsequentes avaliar o tratamento do paciente, validar suas avaliações e gerir códigos e acompanhar prescrições. Portanto, a documentação necessita ser tão acurada, completa e eficiente quanto possível. Realizar isso com sistemas em papel é algo desajeitado e desorganizado. Um sistema computadorizado, porém, pode coletar e armazenar esta informação de maneira mais eficiente (LUSK, 2002, p.1223, tradução nossa).

## 2.2 O Registro Eletrônico de Saúde

### 2.2.1 Evolução

Como consequência deste cenário, houve uma forte tendência de mudança no modelo tradicional de atendimento à saúde, propondo-se o registro eletrônico de saúde (RES), que pode ser definido como "[...]a estrutura eletrônica para a manutenção de informação sobre o estado de saúde e o cuidado recebido por um indivíduo durante todo o seu tempo de vida". (Massad, 2003, p.5). O *Institute of Medicine of National Academies* dos Estados Unidos define o prontuário eletrônico do paciente como o registro eletrônico inserido em um sistema projetado para apoiar os usuários, “[...] fornecendo acesso a um completo conjunto de dados corretos, alertas, sistemas de apoio à decisão e outros recursos, como links para bases de conhecimento médico.” (Dick *et al.*, 1997, s/p, tradução nossa).

Por sua vez, a ISO define o RES como:

Um ou mais repositórios, física ou virtualmente integrados, de informação em forma processável por computadores, relevantes para o bem-estar, saúde e atendimento em saúde de um indivíduo, capaz de ser armazenado e transmitido de forma segura e de ser acessível por múltiplos usuários autorizados, representado de acordo com um modelo de informação lógico padronizado e comumente acordado. Seu objetivo principal é o apoio à atenção em saúde ao longo da vida de forma integrada, efetiva, de alta qualidade e segura (ISO/DIS 18308, 2009, p.11, tradução nossa).

Segundo a definição do HIMSS<sup>14</sup>

É um registro eletrônico longitudinal da informação de saúde de um paciente, gerado por um ou mais encontros em qualquer organização de atenção a saúde. Incluído nesta informação estão dados demográficos de paciente, notas de progresso, problemas, medicações, sinais vitais, história pregressa, imunobiológicos, dados laboratoriais e relatórios radiológicos (HIMSS, 2010, p.14, tradução nossa).

Autores como Beaver (2003) e Velde e Degoulet (2003) concordam em localizar o início do RES na década de 1960, com os primeiros sistemas voltados para a parte administrativa dos hospitais em que funcionavam, particularmente registros de admissão de pacientes e contabilidade. Esses sistemas funcionavam nos laboratórios farmácia e radiologia dos hospitais, tendo sido desenvolvidos como suporte aos procedimentos manuais existentes, sem mais adição de valor, trazendo maior precisão e otimizando tempo. Nos anos da década

---

<sup>14</sup> *Healthcare Information and Management Systems Society* (HIMSS). Página institucional: <<http://www.himss.org/ASP/index.asp>>

de 1980 os sistemas desenvolvidos para automatizar processos já existentes tornaram-se sistemas de suporte a médicos, enfermeiras e outros profissionais envolvidos no cuidado dos pacientes. O objetivo era garantir um padrão no atendimento ao paciente e proporcionar níveis de tomada de decisão. Na década de 1990 torna-se claro que, para reduzir custos, dever-se-ia criar sistemas de informação capazes de assistir aos médicos em suas atividades diárias no cuidado ao paciente, integrando e alocando recursos. Atualmente, com o avanço das tecnologias em rede, a *web*/internet com uma cada vez maior mobilidade, a evolução dos sistemas de saúde caminha rumo a um cenário com informações compartilhadas entre várias instituições, conforme mostra a Figura 1, o que pode levar a uma visão muito mais detalhada da saúde do paciente bem como de comunidades inteiras atendidas por tais sistemas de informação.

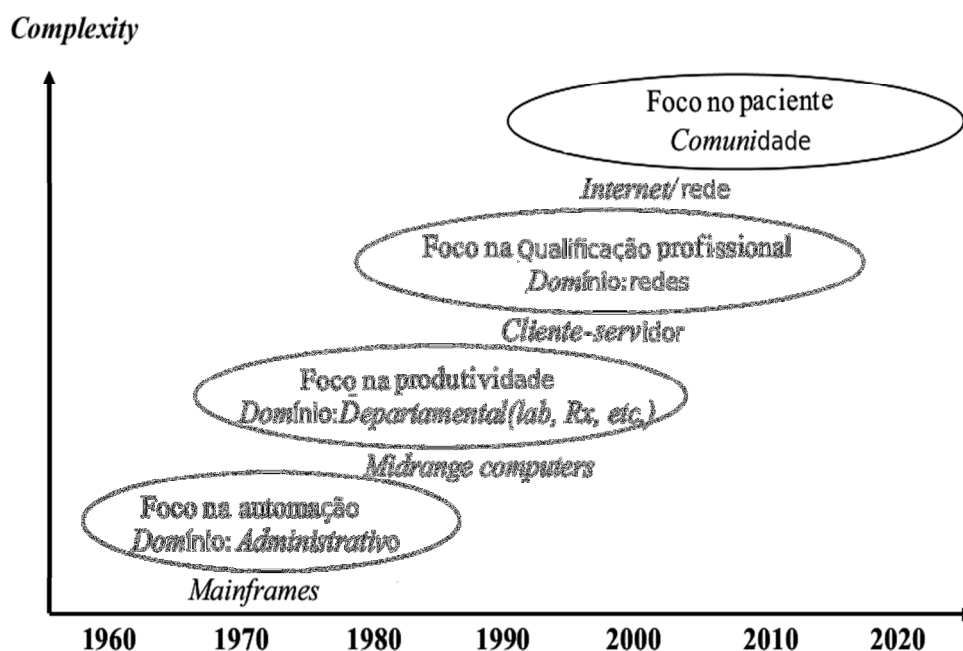


Figura 1 - Evolução da Tecnologia da Informação no Setor de Saúde  
Fonte: Traduzido de Velve e Degoulet (2003)

Embora os meios de armazenagem dos dados ao se fazer um registro médico tenham evoluído consideravelmente, os objetivos ao realizar os registros não só permaneceram como foram acrescidos de outros trazendo novos desafios às modernas formas de registro médico. De acordo com Böhm, Wen e Silveira (2003), o registro médico eletrônico traz vantagens. Há

o ganho de espaço físico, a qualidade da informação, a agilidade do processamento e, sobretudo, a mobilidade dos dados pelo pequeno volume que ocupam. Os sistemas eletrônicos possuem dispositivos de proteção eficientes e de menor custo. Contra furtos e outras violações eletrônicas, essa proteção não se mostrou mais problemática do que a dos arquivos tradicionais, sendo que a experiência mundial em segurança eletrônica é muito grande: há muitos anos os bancos, departamentos de defesa e vários órgãos governamentais tiveram que desenvolver sistemas eficientes contra apropriações indevidas de todas as espécies (Tachinardi; Furuie, 2003).

### 2.2.2 RES *Versus* PEP

Neste ponto, torna-se interessante esclarecer a principal diferença entre sistemas de Prontuário Eletrônico do Paciente (PEP) e sistemas de RES. De acordo com Garets e Davis (2006), o PEP é o registro de informação relacionada à saúde de um paciente, utilizado para suporte ao processo assistencial, gerado a partir de eventos ocorridos em uma **organização de saúde em particular**. Um sistema de PEP é um *software* utilizado para a gestão da informação clínica e demográfica, que pode alimentar um sistema de RES. O RES registra a informação de saúde de um paciente, porém é gerado a partir de eventos ocorridos em **múltiplas organizações de saúde**, e composto de um ou vários repositórios de dados clínicos e demográficos de pacientes. Ou seja, seus dados são coletados através de sistemas de PEP de organizações prestadoras de serviço em saúde, e o RES disponibiliza a informação clínica integrada do paciente.

Conforme visto no breve histórico do RES, apresentado inicialmente, a alocação e o consumo dos recursos informacionais dos sistemas implementados nos hospitais dava-se num nível local, o que tornava limitada a utilização dos dados originários desses últimos. Assim, a significativa variação entre os sistemas utilizados por diferentes hospitais impossibilitava a comparação entre os dados provenientes do sistema de informação de um hospital com os dados provenientes do sistema de outro hospital.

Tal situação advém do crescimento do número de aplicações na área de saúde, que aumentou a complexidade informacional. Cada vez mais os registros dos pacientes se encontram distribuídos em diferentes bases de dados nos sistemas de informação de hospitais e clínicas, abrindo a possibilidade, segundo Bui (2001) para a pesquisa e proposta de novas



tecnologias de informação desde que a condição de acesso a essas bases de dados distribuídas seja satisfeita. Em contrapartida, surgiram tecnologias como *web services*, algoritmos de busca cada vez mais sofisticados, a família de linguagens de marcação XML (Bax, 2001) que aliadas à internet possibilitaram a formulação de soluções capazes de reunir as informações de saúde de uma pessoa presentes em sistemas heterogêneos distribuídos. A grande vantagem da flexibilidade da representação de dados semiestruturada via linguagem XML sobre a rigidez do modelo de dados relacional aliada ao fato de salvar os dados em arquivos de texto viabiliza a troca de mensagens entre diferentes sistemas possibilitando sua integração (Nardon, 2008).

De acordo com Velde e Degoulet:

A eliminação das barreiras geográficas com o desenvolvimento das redes e da internet, e a crescente mobilidade dos pacientes requer que se reconsidere o design e a forma com que os sistemas de registro em saúde são estruturados. A integração de aplicações heterogêneas tornou-se um objetivo central. Dados necessitam ser mapeados e organizados entre aplicações de diferentes domínios. Isso é percebido por cada departamento médico/de saúde (VELDE; DEGOULET, 2003, p.3, tradução nossa).

### 2.2.3 Interoperabilidade Semântica em RES

A heterogeneidade de informação é consequência direta da grande quantidade de modelos de dados e formas de representação existentes, estrutura e sintaxes de dados diferentes entre sistemas de informação. Neste nível também se considera o **aspecto semântico**, ou seja, as **variadas interpretações que estes dados podem receber em diferentes contextos de sistemas de informação**. Estas diferenças tornam a integração de sistemas de informação um processo complexo.

Tal complexidade advém das particularidades dos dados gerados no processo de cuidado ao paciente que, conforme Wetter (2008), realiza múltiplas visitas onde são solicitados múltiplos exames que, por sua vez geram múltiplos resultados. As visitas devem ser documentadas com as razões da visita (sintomas), diagnóstico, lista de terapias aplicadas; as notas podem estar em formato estruturado ou não estruturado. Da mesma forma há os resultados de laboratório, que podem ser acessados por data, por laboratório ou como resultado único; e as imagens geradas em exames como radiologia, cardiologia, *echo*, exames vasculares entre outros necessitam ser arquivadas, acessadas e gerenciadas, entre outros.

Como não poderia deixar de ser, essa complexidade reflete-se nas bases de dados que, conforme salienta El-Sappagh é o componente mais crítico do RES:

O componente mais crítico do registro eletrônico de saúde é a base de dados. Possuir uma base de dados altamente estruturada, dinâmica, completa e consistente é importante para atingir os objetivos do RES. Sistemas de saúde moveram-se de sistemas isolados em hospitais para soluções que incluem múltiplos profissionais de saúde e instituições, interoperabilidade e compartilhamento de informações. Torna-se, portanto, crucial permitir que todos os dados clínicos relevantes estejam disponíveis a qualquer momento, em qualquer lugar para que se obtenha melhoria na qualidade e fornecimento do cuidado em saúde (EL-SAPPAGH, 2012, p.6, tradução nossa).

Dessa maneira, a complexidade dos modelos de dados de um registro eletrônico de saúde gera uma situação diferenciada quando comparado a outras áreas, em termos das dificuldades ao registrar seus dados em bases de dados para registros de saúde, conforme salienta Paul:

**Dados de saúde diferem dos dados de outras indústrias.** Dados esparsos ocorrem porque os médicos fazem apenas poucos tipos diferentes de testes laboratoriais entre milhares de possibilidades de testes para um dado paciente em toda a vida deste último. Isso requer frequentes alterações nos esquemas de dados porque torna-se necessário acomodar novos testes laboratoriais, como novas doenças que surgem constantemente. Por estas razões, a representação de dados relacional requer muitas colunas e, portanto, dados de alta dimensão. Relações muitos-para-muitos entre pacientes e diagnósticos **requerem complexas modelagens de dados.** **Consultas incorrem na penalidade do desempenho** se os registros de dados são extensos mas apenas poucas colunas são usadas na consulta (PAUL, 2010, grifo nosso, p.1810, tradução nossa).

Assim, de modo a representar um *design* de objeto numa base de dados relacional todas as classes necessitam ser mapeadas para tabelas relacionais, num processo conhecido como mapeamento objeto-relacional, onde um objeto é representado em tabelas de duas dimensões. O mapeamento objeto-relacional traz consigo um substancial problema de desempenho, pois os dados necessitam ser separados, ou desconstruídos, em múltiplas tabelas relacionais, sempre que forem armazenados e reconstruídos quando da sua recuperação, o que pode consumir significativamente recursos do sistema, dada a complexidade da operação. A Figura 2, a seguir, mostra a desconstrução em tabelas:

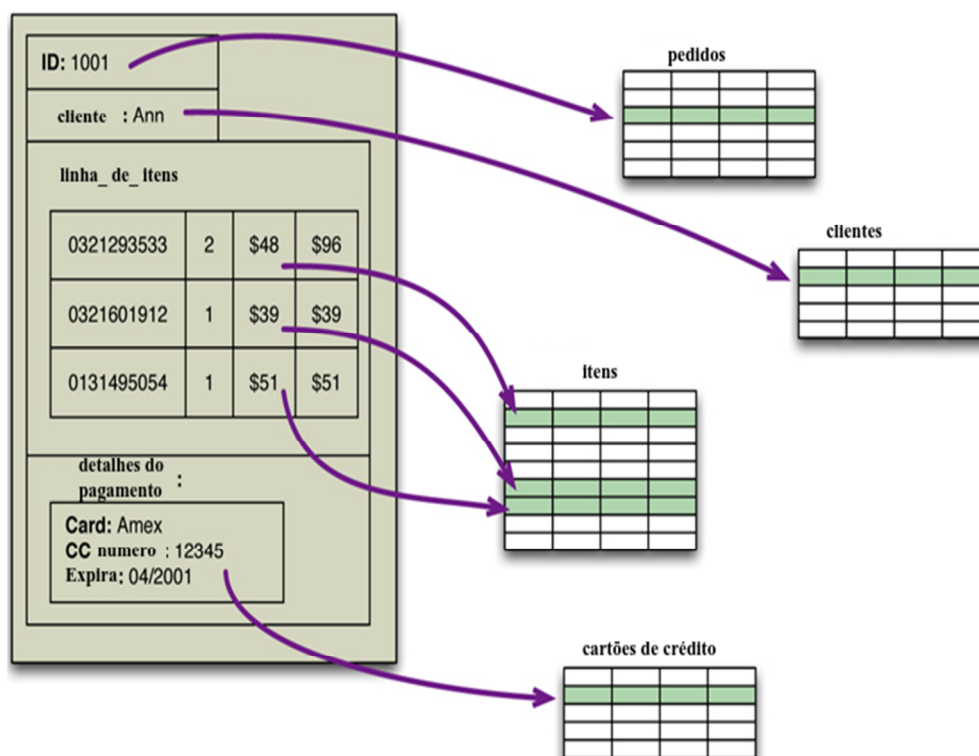


Figura 2 - Um Formulário de Pedido Decomposto em várias Tabelas  
Fonte: Traduzido de Sadalag e Fowler (2013)

Devido ao contexto exposto, justifica-se para sistemas que, normalmente, trabalham com o tipo de dados complexos dos RES, a busca por outras opções de bancos de dados, como os bancos de dados do tipo NoSQL, conforme ressaltam Lee *et al.*:

Os bancos de dados relacionais são os mais comuns e, comprovadamente, a abordagem mais utilizada para armazenar e consultar dados em várias formas. Porém, sua maior desvantagem é a necessidade do exato *pré-design* das estruturas dos campos de dados, o que é requerido no processo de normalização da base de dados para assegurar a consistência dos dados. Assim, **modelos relacionais não se mostram práticos para certos formatos de dados** que requerem muitos campos para lidar com diferentes tipos de dados envolvidos, onde muitos dos campos de dados envolvidos são pouco utilizados devido à natureza dos dados. Uma base de dados que registre esse tipo de dado certamente conterá muitos campos vazios, resultando num armazenamento ineficiente e num desempenho pobre. Dados médicos, especialmente notas clínicas, são um exemplo dessa situação. Para lidar com essas questões, tenta-se usar uma classe de bancos de dados conhecidos como NoSQL e linguagem de marcação extensível (XML) para desenvolver bases de dados que consigam lidar mais efetivamente com as características especiais dos dados clínicos (LEE *et al.*, 2013, p.99, grifo nosso, tradução nossa).

Bancos de dados NoSQL operam sem um esquema de dados pré-definido, permitindo que se adicione novos campos aos registros sem a necessidade de realizar alterações em sua estrutura. Essa característica é interessante ao lidar com dados não uniformes ou campos customizados.

A mudança significativa é a possibilidade de se ter opções para armazenagem de dados. Sadalage e Fowler (2013) se referem a esse ponto de vista como **persistência poliglota**, que significaria o uso de diferentes tipos de armazenagem de dados em diferentes circunstâncias. Ao invés de, simplesmente escolher a base de dados relacional porque esta é uma unanimidade, necessita-se compreender a natureza dos dados que se está armazenando e como se deseja manipulá-lo.

As características comuns das bases de dados NoSQL são (Sadalage; Fowler, 2013):

- Não utilizam modelo relacional.
- Bom desempenho em *clusters*.
- Código aberto.
- Construídas para os domínios da internet do século 21.
- Livre de esquemas.
- Persistência poliglota, considerado o mais importante resultado trazido pelas bases NoSQL.

Das demandas específicas associadas ao contexto dos RES discutido até aqui, adveio o esforço num nível mundial em tornar a interoperabilidade entre esses sistemas uma realidade, sob três aspectos (HL7, 2010):

- Técnico: possibilitar o envio de dados entre sistemas diferentes.
- Semântico: garantir que os diferentes sistemas compreendam os dados compartilhados da mesma maneira.
- Processo: possibilitar que sistemas diferentes presentes na mesma instituição trabalhem juntos.

O RES apresenta-se, portanto, como um componente central na construção de um

sistema integrado de saúde, podendo constituir-se num repositório de dados para pesquisa e melhoria nas condições de tratamento de pacientes e para a definição de políticas de saúde pública. Esta construção também exige a integração de diferentes sistemas de informação utilizados pelas instituições de saúde. Como nos parágrafos anteriores os termos integração e interoperabilidade foram utilizados livremente, cabe aqui contextualizá-los de maneira apropriada.

Segundo o HIMSS (2010), a **integração** é o arranjo dos sistemas de informação da organização de um modo que esses possam se comunicar eficientemente, reunindo as partes relacionadas em um único sistema. Já a **interoperabilidade** é a capacidade dos sistemas de informação de trabalharem juntos, interna e externamente às fronteiras organizacionais, a fim de promover uma prestação de serviços eficaz. O dicionário *Webster* define interoperabilidade como “a habilidade de um sistema de trabalhar com ou usar as partes de outro sistema”.

Destas duas definições, pode-se inferir que a integração está mais relacionada com os sistemas incorporarem uma forma única de trabalho, almejando um objetivo comum, de maneira unificada e padronizada. Por outro lado, a interoperabilidade implica em diferentes sistemas de informação agregando suas forças em prol de um objetivo comum, sem, no entanto, alterarem sua autonomia e características próprias. De acordo com Sheth (1999), a interoperabilidade é um requisito básico para os sistemas de informação. O autor sugere diferentes níveis de interoperabilidade, a saber: de sistema, de sintaxe, estrutural e semântica. Nesse último nível, o significado da informação que é intercambiada deve ser garantido entre os sistemas de informação participantes do processo. Especificamente, no cenário de interoperabilidade semântica, os sistemas interoperam **compartilhando um vocabulário comum**.

O cenário em que se encontram os sistemas de RES caracteriza-se pela heterogeneidade, dentre outros fatores, fruto de uma complexidade informacional advinda das várias especialidades médicas, terminologias, culturas e idiomas. Há ainda os vários sistemas de PEP disponíveis no mercado, com funcionalidades específicas para diferentes tipos de organizações. Estes fatores ajudam a explicar a importância dada à obtenção de interoperabilidade semântica para os sistemas de RES. Conforme Kalra (2007), a comunidade internacional considera a obtenção de interoperabilidade semântica entre sistemas de RES essencial quando se pensa o futuro dos serviços em saúde, onde o uso de terminologias,

ontologias e arquétipos compõe o núcleo de tal desafio.

O mesmo Kalra coloca o objetivo da interoperabilidade semântica como:

Tornar os sistemas capazes de reconhecer e processar a informação equivalente semanticamente de forma homogênea, mesmo se as instâncias são representadas de forma heterogênea. Ou seja, elas são diferentemente estruturadas, e/ou usam diferentes sistemas de terminologias, e/ou usam diferentes idiomas (KALRA, 2007, p.14, tradução nossa).

Dos vários desafios e mudanças culturais necessárias ao desenvolvimento de um sistema de RES, sob o ponto de vista técnico, o desafio da interoperabilidade e a complexidade das informações, fazem com que seu desenvolvimento seja mais difícil do que o desenvolvimento de outros sistemas de informação (NARDON, 2002).

### 2.3 Terminologias, Modelos de Informação e Padrões

Os vocabulários utilizados na área médica são complexos, não estruturados e repletos de variações regionais e especificidades. Mesmo nessa condição, os sistemas de RES devem gerar dados importantes para outros sistemas e disponibilizar recursos para trabalhar com diferentes tipos de informação estruturada e não estruturada, em diferentes formatos e volumes.

As tentativas atuais para obtenção de interoperabilidade semântica entre sistemas de RES se concentram no desenvolvimento de três tipos artefatos (KALRA, 2007): (i) modelos de informação genéricos para representação de dados clínicos; (ii) definições de estruturas de dados clínicos; e (iii) sistemas de terminologia clínica.

A utilização de **modelos de informação genéricos ou modelos de referência** (i) tem sido uma abordagem internacionalmente aceita para a construção de arquiteturas de sistemas de RES. Em linhas gerais, eles **definem um padrão de representação do dado clínico e suas propriedades comuns**. Quando um mesmo modelo de informação genérico é utilizado por sistemas de RES que precisam interoperar, as partes do registro de saúde do paciente trocadas conterão a estrutura, nomes e informações médico-legais necessárias para que o dado clínico seja compartilhado no formato correto, mesmo se a natureza do conteúdo clínico não tenha sido previamente acordada (KALRA, 2007). Em complemento ao modelo de informação genérico existem estruturas de dados clínicos e as terminologias clínicas. As estruturas de dados clínicos (ii) permitem que as classes do modelo de informação sejam

utilizadas para representar de forma consistente os conceitos clínicos relevantes. Essas estruturas de dados clínicos introduzem uma abordagem sistemática para representação dos conceitos em um sistema de RES, e sendo padronizadas, podem ser compartilhadas entre diferentes instituições de saúde (KALRA, 2007). Com relação às terminologias (iii) são essenciais para a interoperabilidade de sistemas de RES, pois são utilizadas na codificação do conhecimento nas diversas partes do RES e aumentam a possibilidade de interpretação correta dos dados envolvidos, tanto para humanos quanto para máquinas (HL7, 2010).

A importância das terminologias ou vocabulários controlados para se atingir a interoperabilidade semântica entre os diferentes sistemas é ressaltada por Neira (2008) ao destacarem o aspecto semântico da interoperabilidade, ou seja, como a habilidade de dois ou mais sistemas trocarem dados com base em um vocabulário previamente acordado garantindo a mesma interpretação das noções para seus usuários. Os vocabulários controlados, ou terminologias, se mostram úteis para (HL7, 2010):

- Padronização de textos livres ou conteúdo estruturado do RES.
- Representação de observações clínicas e avaliações.
- Codificação de testes e resultados.
- Identificação de drogas.
- Colaboração de dados clínicos em tempo real.
- Representação de aspectos sintáticos e semânticos dos conceitos médicos.
- Recuperação e análise de dados, e suporte aos processos de tomada de decisão.

**O desenvolvimento de padrões ou modelos de referência veio, portanto, como resposta para o tipo de problema apresentado ao se buscar sistemas de RES interoperáveis e com melhor relação custo-benefício. Conforme o padrão ABNT ISO/TR 20514 (ABNT, 2008) um RES deve incorporar um modelo de referência da informação em saúde.**

Ao procurar implementar, através da abordagem tradicional de desenvolvimento num único nível, sistemas que apresentem o grau de complexidade e exigência que os sistemas de RES necessitam, obteve-se um sistema de difícil manutenção, curta vida útil e alto custo (Beale, 2007). A abordagem em dois níveis, advinda das técnicas de representação de

conhecimento da I.A., mostrou-se adequada para evitar essas consequências, sendo que um nível refere-se ao modelo de domínio e terminologias e outro ao modelo de informação, procurando deixar a geração do conhecimento diretamente sob a responsabilidade dos especialistas com pequena (idealmente nenhuma) dependência dos profissionais em tecnologia de informação.



### 3 REPRESENTAÇÃO DE CONHECIMENTO E ONTOLOGIAS: O MODELO DUAL

Uma vez que, conforme visto no capítulo anterior, a modelagem dual, proveniente da Inteligência Artificial, mostrou-se promissora quando se visa à construção de *softwares* do tipo RES, neste capítulo, a fundamentação referente à representação de conhecimento, base para esta abordagem, é apresentada.

Inicialmente, a distinção entre as abordagens declarativa e procedimental para a construção de sistemas computacionais são vistas buscando distingui-las e contextualizá-las, pois suas fundamentações e motivações distintas são aplicáveis a diferentes situações. Particularmente a abordagem declarativa, fundamentada na lógica matemática, ao buscar criar sistemas de modo declarativo, provê a base para a separação do conhecimento sobre um dado domínio do processamento associado a ele.

Assim, em seguida, discute-se a importância e vantagens do uso do modelo dual em implementações onde se necessita lidar com conhecimento associado a dados domínios, mantendo características como o engajamento ontológico, portabilidade, fidelidade semântica e independência da máquina. Em seguida, a ontologia, enquanto artefato para representação formal de conhecimento é apresentada. Algumas de suas características e principais definições são discutidas, bem como as vantagens advindas do seu uso. Isso fundamenta a apresentação, no Capítulo 5, da ontologia CIR, base para a classe *Entry* do modelo de referência do padrão para interoperabilidade de dados clínicos OpenEHR.

#### 3.1 A Abordagem Declarativa para o Processamento de Conhecimento

Duas abordagens para elaboração de sistemas computacionais surgiram após o advento das linguagens de alto nível (Sebesta, 2012):

- A abordagem procedimental ou imperativa, que procura descrever o funcionamento dos processos em seus detalhes, sem compromisso em descrever também as características das entidades representadas – metáfora do “como fazer”.
- A abordagem declarativa, que procura descrever as entidades e os fatos acerca de um dado domínio, utilizando-se de máquinas de inferência para deduzir novos fatos a partir dos já existentes bem como a solução desejada ou ação a ser tomada.

Nas décadas de 70 e 80 travou-se um debate sobre qual das duas abordagens seria a mais adequada para se programar sistemas. Concluiu-se que tais abordagens não são excludentes entre si, pois são mais bem adequadas a determinados tipos de tarefas. Quando se trata, porém, de controle e eficiência, a abordagem procedimental apresenta os melhores resultados, tornando-se assim o paradigma de programação mais comumente empregado.

O paradigma declarativo, fundamentado de modo consistente na lógica matemática seguiu uma trajetória diferente, sua motivação é modelar os sistemas num nível de abstração mais elevado, descrevendo o conhecimento do domínio que se deseja modelar sem preocupações a respeito da implementação (Russel; Norvig, 2004). Seu objetivo principal é, de modo declarativo, criar sistemas diretamente a partir do conhecimento, permitindo ao computador realizar inferências de modo automático, utilizando processos de dedução da lógica matemática como resolução, dedução natural, *tableau* semântico (Smullyan *et al.*, 2009), entre outros.

O paradigma lógico, implementando a ideia de computação como dedução, usa um modelo computacional que é substancialmente diferente de todos os outros vistos até agora. [...] Linguagens lógicas permitem expressar soluções para problemas complexos de uma maneira puramente declarativa e de modo extremamente simples e compacto (GABBRIELLI; MARTINI, 2010, p.369, tradução nossa).

O paradigma declarativo serve como fundamento, sempre que for necessário **separar o conhecimento a respeito de um dado domínio do processamento associado a ele.**

### 3.1.1 O Modelo Dual

O desenvolvimento de sistemas dessa natureza exige primeiro que se especifique de modo abstrato o conhecimento a respeito do domínio para em seguida, converter a especificação obtida em sentenças lógicas. Após essas duas etapas, é feita a codificação para uma linguagem apropriada. Ao seguir esse procedimento para gerar um sistema, concentrando-se em especificar o conhecimento sem relacioná-lo com um código específico, mantendo-o externo ao sistema, os pesquisadores visualizaram as vantagens do paradigma declarativo como **engajamento ontológico ou de conhecimento** (significando que as sentenças lógicas **guardam uma relação mais direta com o domínio que se está modelando**), legibilidade, capacidade de inferência, fidelidade semântica, reusabilidade, portabilidade do conhecimento por si, independente do código de implementação e, portanto,

independente da máquina. Estas características apresentam-se como os grandes atrativos para os esforços em desenvolver sistemas em **modelo dual**, separando o conhecimento do domínio do resto do sistema, conforme descrito acima.

O estado da arte deste paradigma se consolidou no desenvolvimento dos chamados sistemas baseados em conhecimento (Brachman; Levesque, 2004) que incorporam a programação lógica e, diferente dos programas de uso geral que buscam soluções completas para problemas utilizando-se de passos elementares de raciocínio desenvolvidos no início das pesquisas em Inteligência Artificial (Russell; Norvig, 2004). Estes sistemas possuem um conhecimento mais amplo e específico de um dado domínio, permitindo passos de raciocínio maiores e mais complexos para se resolver problemas em uma área específica. Como exemplos, são geralmente citados os sistemas especialistas: MYCIN, DENDRAL, SHRDLU, CYC, entre outros.

A Inteligência Artificial apresenta três importantes técnicas de decomposição para construção de sistemas:

- 1- Separar o conhecimento dos programas através de uma linguagem de representação do conhecimento declarativa.
- 2- Identificar classes gerais e suas relações, organizando o conhecimento para possibilitar herança desses construtos.
- 3- Caracterizar o problema geral resolvendo tarefas e classes de inferência, e projetando métodos correspondentes e algoritmos (Gruber, 1993).

A importância da separação do conhecimento já havia sido reconhecida por Allen Newell (1982), com a proposta de inserção de um novo nível de sistemas para os já existentes – chamado nível de conhecimento. Segundo este autor, os níveis existentes eram insuficientes para a descrição da lógica no ambiente que cerca os sistemas computacionais. Sua hipótese levou em consideração que existia um nível de sistema computacional distinto, imediatamente acima do nível simbólico, caracterizado pelo conhecimento.

Na busca por uma melhor forma de representação do conhecimento, os formalismos orientados a predicados, como as regras de produção e programação lógica (Sterling, 1999), bem como os formalismos orientados a classes e relações, como redes semânticas, lógica de

descrições e *frames* (Copin, 2004), procuraram, enfatizando a declaração dos predicados lógicos do domínio e inferências ou nas classes de objetos pertencentes ao domínio e suas relações, representar o conhecimento e também processá-lo. As ontologias surgiram na esteira desses esforços.

Brachman (1979 apud GUARINO, 1995)<sup>15</sup> definiu uma classificação dos tipos de primitivas usadas por sistemas de representação do conhecimento, propondo uma classificação de cinco níveis: Linguístico, Conceitual, Epistemológico, Lógico e Implementacional. Considerando os ganhos do uso de ontologias para a área de representação do conhecimento, Guarino (1995) propôs a inserção do **nível Ontológico**, posicionado entre o nível conceitual e o epistemológico, ou seja, um nível onde **o significado associado a uma linguagem de representação do conhecimento poderia ser restringido formalmente**.

### 3.2 Ontologias

O termo Ontologia tem sua origem nos trabalhos de Aristóteles, quando este se dedicou a estudar as características do Ser em geral (Lalande, 2003). Em Inteligência Artificial foi inserido num novo contexto, bem mais restrito, embora próximo do seu significado original, como um índice de tipos de entidades que podem existir num dado domínio elaborado sob a perspectiva de um indivíduo que faz uso de uma dada linguagem (Sowa, 2000). Há ainda a definição proposta por Gruber (1995), onde uma ontologia é a especificação formal, explícita e compartilhada de uma conceitualização.

As ontologias podem ser vistas como resposta contemporânea a uma necessidade dos sistemas baseados em conhecimento. Um propósito é favorecer o compartilhamento e o reuso do conhecimento armazenado em sistemas criados para as mais diversas finalidades. Este último, antes do predomínio da Internet, não podia ser compartilhado ou reutilizado. Em geral organizava-se em bases de conhecimento isoladas umas das outras, em linguagens diferentes, sem interfaces capazes de integrá-las e, portanto, sem interoperabilidade.

Com o crescimento do uso social e economicamente relevante da Internet e o processo de modelagem conceitual dos diversos domínios do conhecimento, sente-se hoje a necessidade do reuso do conhecimento. Principalmente em áreas como a Medicina ou a Biologia, repletas de conceitos complexos, dinâmicos e fortemente inter-relacionados (Baker;

---

<sup>15</sup> BRACHMAN, 1979 apud GUARINO, 1995, p. 5.

Cheung, 2007).

Além do problema da interoperabilidade, muitos pesquisadores viam a especificação dos sistemas baseados em conhecimento sob a ótica funcional, com o foco na estratégia de resolução de inferências via regras, consoante à visão dos formalismos orientados a predicados (Russell; Norvig, 2004). Essa visão terminava por reduzir o volume de conhecimento que poderia ser reutilizado, pois muitos dos conceitos e relações do domínio acabavam implícitos, expressos como parte das regras e premissas criadas no processo de modelagem, o que tornava essas bases de conhecimento aplicáveis apenas na resolução de problemas no domínio original não deixando possibilidade de reuso, mesmo para situações em que havia proximidade conceitual entre os domínios.

Em face de propostas como *frames*, redes semânticas, thesaurus, entre outras, as ontologias podem ser vistas como uma etapa significativa na evolução dos processos de especificação de conhecimento. Elas abstraem o conhecimento de seus processos de tratamento, tornando-o modularizável, interoperável, reutilizável e com engajamento ontológico mais forte. Ou seja, homogeneizado tanto em relação à forma como é organizado, como em relação à terminologia empregada no domínio modelado.

Uma das mais importantes definições de ontologia advém da linha de pensamento da Ciência da Computação. Segundo Gruber (1995, p.1): “uma ontologia é uma especificação explícita de uma conceitualização”. Logo, para este autor, todo conhecimento formalmente representado é baseado em uma conceitualização: os objetos, conceitos e seus supostos relacionamentos. Essa conceitualização é uma **visão simplificada e abstrata do mundo** que se deseja representar.

Guarino (1995, p.2) critica a definição de conceitualização assumida por Gruber, e apresenta uma revisão para sua definição de ontologia: “uma teoria lógica que explica de forma explícita e parcial uma conceitualização”. Na sua visão, uma conceitualização não pode ser representada por completo, dentre outros fatores, pois depende da percepção dos observadores.

Existe uma diferença ontológica entre “termos em uma linguagem” e “objetos universais na realidade”, e uma ontologia se preocupa com esses últimos. Diferentemente das ontologias, as terminologias lidam com o uso da linguagem, mostrando como a linguagem é usada em um campo científico específico, e como seus termos são relacionados. **Uma**

**terminologia é projetada para prover delimitações claras de conceitos via definições, enquanto uma ontologia é um artefato representacional cujas unidades representacionais são intencionadas para representar objetos universais na realidade e suas relações.** Por se basearem em conceitos, as terminologias podem diferir em função do ponto de vista de seus criadores, pois existem diferentes formas de se entender um conceito: podem ser entendidos como uma entidade mental, uma entidade teórica compartilhada, ou uma entidade linguística (Lacasta; Nogueras-Iso; Zarazaga-Soria; 2010).

Definindo-se ontologia a partir dos requisitos que permitem sua aplicação em computação (Daconta *et al.*, 2003, p.185) tem-se:

“Uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada”. Analisando os termos da definição:

- Especificação explícita corresponde às definições de conceitos, instâncias, relações, restrições e axiomas.
- Formal corresponde a ser definida declarativamente, logo, acessível por agentes e sistemas ou, legível para computadores.
- Conceitualização corresponde a modelo abstrato de algum domínio do real.
- Compartilhada corresponde a conhecimento consensual enquanto vocabulário comum do domínio modelado ou definido por comum acordo entre os desenvolvedores dos agentes e sistemas que lhe farão uso.

Dessa definição verifica-se que uma ontologia fornece um entendimento sobre uma conceitualização compartilhada acerca dos dados e metadados de um dado domínio, um vocabulário comum isento de ambiguidades. Idealmente, qualquer instância que fizer uso dos dados e metadados de um domínio dever aderir à ontologia correspondente (Guarino, 1995).

Uma ontologia requer um vocabulário específico que descreva um domínio e também um conjunto de axiomas lógicos que garantirão a semântica ao significado que se deseja para os termos do vocabulário. Isso implica que duas ontologias podem ser diferentes em termos de seus vocabulários e referirem-se à mesma conceitualização, ou seja, ao mesmo domínio de conhecimento.

A organização dos conceitos bem como a integração semântica, possibilitando a

interoperabilidade entre os sistemas é feita através do desenvolvimento de ontologias, que contextualizam os dados e lhes dão significado. Se um RES registrar que um determinado paciente apresenta “alergia”, isso quer dizer que este dado possui o mesmo significado do termo “alergia” que se encontra na ontologia, caracterizando assim a necessidade de um mapeamento correto e consistente entre dados dos modelos de informação e os termos da ontologia (Cannoy; Yier, 2009).

Uma ontologia pode ser entendida como uma especificação explícita e formal de uma conceitualização compartilhada (Gruber, 1995), caracterizando-se pela definição de um vocabulário comum para o compartilhamento de informações sobre um dado domínio. Ou seja, uma ontologia pode ser utilizada por uma aplicação computacional para explicitar quais termos utiliza e também o seu significado (Daconta; Obrst; Smith, 2003). Pode ser encarada, também, como um formalismo capaz de retirar a ambiguidade da linguagem natural e de notações informais permitindo, através de um vocabulário pré-definido, a criação de um canal de comunicação preciso entre diferentes aplicativos que compartilham significações comuns a respeito de um dado domínio de conhecimento (Gašević; Djurić; Devedžić, 2006).

Ontologias trazem, também, através do seu uso, conforme Arancon *et al.* (2008), o que se pode denominar de **clareza estrutural**, pois ontologias proveem uma **organização hierárquica dos elementos através de princípios similares à programação orientada à objetos** e aos bancos de dados orientados à objetos. A informação estruturada resultante torna possível representar as relações entre objetos do domínio de maneira mais natural do que num banco de dados relacional. Além disso, **permite representar relações hierárquicas difíceis de formalizar via álgebra relacional**. Como resultado, a estrutura final é muito mais clara do que a obtida utilizando-se a clássica base de dados relacional.

Na busca pela interoperabilidade entre sistemas de RES, muita ênfase tem sido dada à utilização de terminologias com a justificativa de que são os “códigos” que permitirão associar os termos utilizados pelos seres humanos a uma sintaxe uniforme e significados que podem ser interpretados por um *software*. Porém, a aplicação de ontologias para o domínio de RES pode tornar as terminologias logicamente mais coerentes e compatíveis com as intuições de senso comum (SMITH; CEUSTERS; TEMMERMAN, 2005). Tal contexto poderá ser visto no Capítulo 5, quando se discutirá o padrão para uma arquitetura de RES da fundação OpenEHR, que faz uso de uma ontologia.

## 4 METODOLOGIA DE IMPLEMENTAÇÃO DE ARQUÉTIPOS OPENEHR

Ao iniciar a pesquisa, a trajetória a ser percorrida na busca por expressar os arquétipos OpenEHR no CMS Plone indicava que, antes disso, era necessário verificar se a linguagem Python, em que o CMS é escrito, era capaz de expressar a semântica definida pelo padrão. Ficou claro que esta trajetória envolveria questões a serem respondidas teoricamente, bem como as que seriam respondidas via implementação de artefatos computacionais.

A necessidade de lidar com questões de diferentes naturezas, práticas e teóricas, aninhadas e interdependentes em suas diferentes soluções, encaminhou a pesquisa a buscar suporte no *design science research*<sup>16</sup>, uma vez que este, enquanto proposta metodológica, posiciona-se exatamente neste contexto.

Ao buscar posicionar e valorizar os problemas de *design*, advindos das ciências dos fenômenos artificiais, colocando-os no nível dos das ciências naturais, pesquisadores como Simon inserem possibilidades do desenvolvimento de posturas e metodologias para trabalhos que envolvem a criação de artefatos:

Todos [realizam] *design* quando divisam cursos de ação objetivando alterar situações existentes em prol de outras que sejam preferidas. A atividade intelectual que produz artefatos materiais não é fundamentalmente diferente de uma que prescreve remédios para um paciente doente, que cria novos planos de vendas para uma companhia ou uma política de bem estar social para um estado. O *design*, visto sob essa ótica, apresenta-se como o núcleo de todos os treinamentos profissionais (SIMON, 1969, p.111, tradução nossa).

A seguir será feita uma apresentação dos pontos fundamentais do *design research* para, em seguida, ser discutido o seu uso, enquanto metodologia, no esforço de expressar os arquétipos OpenEHR como conteúdos em CMS; por fim os passos realizados na pesquisa, surgidos sob a sua ótica, serão apresentados.

### 4.1 *Design Research*

Conforme o parágrafo anterior, ao procurar trazer a atividade do *design* para um novo patamar intelectual, Simon (1969) procurará esclarecer e distinguir ciência natural e ciência do artificial (ou *design science*). Ciência natural seria um corpo de conhecimento sobre

---

<sup>16</sup> Optou-se, no texto, pelo uso do termo original “*design science research*”, uma vez que o termo traduzido não se mostrou, até o momento da escrita deste trabalho, ter sido adotado na literatura acadêmica pesquisada.



classes de fenômenos ou objetos do mundo (natural ou da sociedade) que descreve como se comportam e interagem uns com os outros. Por sua vez, a ciência do artificial é um corpo de conhecimento sobre o *design* do artificial, que podem ser objetos ou fenômenos, denominados assim de artefatos, que são concebidos para cumprir certos objetivos pré-definidos.

O objetivo geral do *design research* seria criar ou colaborar para o acréscimo de novos e interessantes conhecimentos de *design* numa área de interesse, ser uma doutrina sobre o processo de *design* num corpo de conhecimentos intelectualmente robusto, analítico, **parte formal e parte empírico**. Assim, pode-se entender que o *design research* envolve a criação de novos conhecimentos através do *design* de artefatos (coisas ou processos) bem como da análise do seu uso e desempenho. Tais artefatos, conforme ressaltado por Vaishnavi e Kuechler (2014), especificamente sob a ótica dos sistemas de informação, vão incluir, mas não se limitar a algoritmos, interfaces, metodologias para o *design* de sistemas e linguagens.

Deve-se reforçar a atenção para a característica do conhecimento de acordo com o *design research*, conforme a descrição anterior: “parte formal e parte empírico” (Bax, 2014), sendo essa dupla característica a responsável por sua escolha no caso da pesquisa aqui desenvolvida. Pois nela ocorre o desenvolvimento de artefatos que, por sua vez, permitem conclusões e o acréscimo de novos conhecimentos e o desenvolvimento de novos artefatos de forma cíclica. Conforme ressaltaram Hevner e Chatterjee:

*Design research* é um paradigma de pesquisa em que o *designer responde a questões relevantes para problemas humanos via criação de artefatos inovadores, contribuindo, assim, com o acréscimo de novos conhecimentos ao corpo das evidências científicas*. Os artefatos concebidos são, simultaneamente, utilizáveis e fundamentais na compreensão do problema (HEVNER; CHATTERJEE, 2010, p.5, grifo nosso, tradução nossa).

Da mesma forma, Wieringa chama a atenção para a resolução de ciclos de problemas:

*Design research enfatiza a conexão entre conhecimento e prática* mostrando que se pode produzir conhecimento científico através do *design* de coisas úteis. [...] para resolver um problema prático, o mundo real é modificado para se adaptar a propósitos humanos, mas para resolver um problema de conhecimento, nós adquirimos conhecimento sobre o mundo sem, necessariamente, modificá-lo. No *design research*, esses dois tipos de problemas estão mutuamente aninhados, mas este aninhamento não deve nos cegar para o fato de que suas soluções e métodos de solução são diferentes (WIERINGA, 2009, s/p, tradução nossa).

A possibilidade de lidar com problemas de naturezas diversas como de conhecimento

e práticos, o aninhamento e influência entre os mesmos torna o *design research* uma opção adequada como referência para o percurso a ser desenvolvido por esta pesquisa.

## 4.2 Arquétipos como Artefatos

No caso dos arquétipos OpenEHR, especificamente, a adequação da pesquisa que os envolve à metodologia proposta pelo *design research* ocorre, graças à percepção dos tipos de problemas envolvidos e sua relação mútua. O problema prático envolvendo sua implementação nas diversas plataformas é recorrente no esforço de viabilização do padrão OpenEHR como solução para RES interoperáveis, embora, como se vê nos trabalhos correlatos visitados, a contribuição ao conhecimento também surja, naturalmente, durante o percurso. Por sua vez, o correspondente metodológico no *design research* é salientado, por Vaishnavi e Kuechler (2014, s/p) ao afirmarem que “O que faz o processo de modelagem do *design research* diferente do processo de modelagem correspondente do *design* é o fato de que a contribuição ao conhecimento permanece como sendo o foco chave do *design research*”.

Entre os princípios de *design* norteadores do padrão OpenEHR, aquele responsável pelas características que mostraram ser o *design research* uma proposta metodológica coerente com os objetivos deste trabalho está o modelo dual (discutido neste mesmo capítulo), responsável por separar os modelos de informação e de conhecimento, que Garde *et al* consideram o mais inovador de seus princípios:

Os princípios do *design* do OpenEHR são descritos em maiores detalhes em sua especificação disponível em sua página institucional, mas a **inovação principal da arquitetura do OpenEHR é que este separa o relativo aos registros que desejamos preservar dos dados clínicos utilizando arquétipos** (GARDE *et al.*, 2007, p.4, grifo nosso, tradução nossa).

Essa separação entre o modelo de referência e o modelo de arquétipos em camadas distintas (Figura 3 abaixo) estabelece uma ordem quando de qualquer tentativa de implementação da especificação do padrão OpenEHR numa linguagem computacional. Uma vez que a camada do modelo de referência contém os modelos de informação definidos na especificação do padrão, a semântica do padrão também se encontra nessa camada (veja o Capítulo 5), dessa forma antes de qualquer tentativa de expressão de um arquétipo, como conteúdo num CMS ou codificando-o a partir “do zero”, deve-se realizar uma **verificação, via**

**implementação**, referente à adequação da linguagem de implementação pretendida à semântica do OpenEHR.

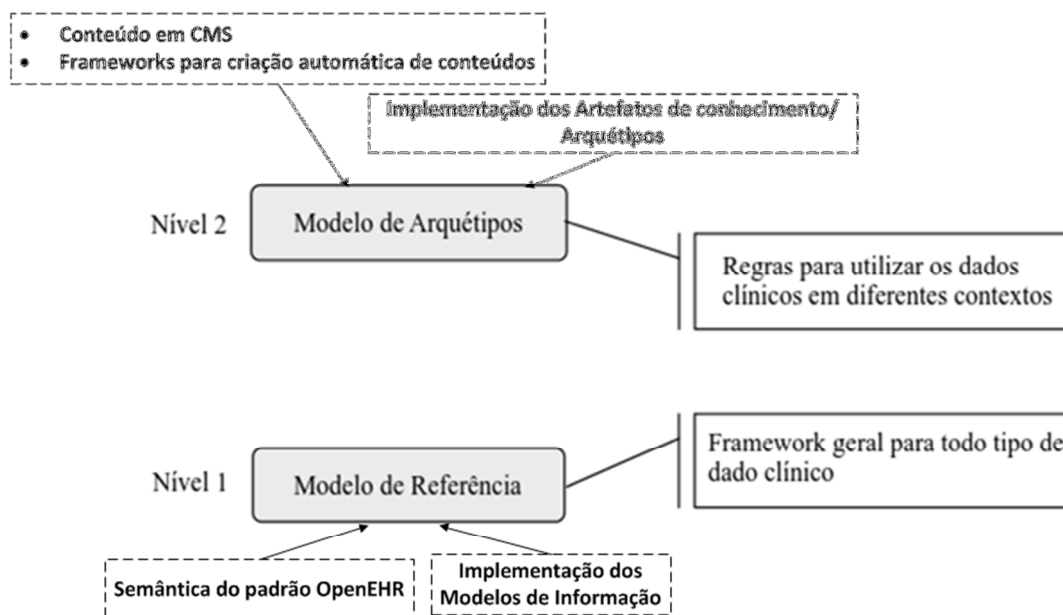


Figura 3 - Separação dos Modelos e Implementações no OpenEHR

Fonte: Adaptado e Traduzido de Sinha *et al.* (2013)

O resultado de tal verificação, que corresponde a uma contribuição ao **conhecimento** sobre o problema advém, necessariamente, da resolução de um problema **prático**, que é a implementação dos modelos de informação da camada de referência do padrão OpenEHR na linguagem escolhida. Essa situação, composta de problemas de diferentes naturezas, aninhados e, encadeadamente, influenciando suas soluções, mostrou-se consistente com a metodologia proposta pelo *design research*, descrita anteriormente, justificando sua escolha como norteador da trajetória da pesquisa desenvolvida neste trabalho.

#### 4.3 Metodologia para a Implementação dos Arquétipos OpenEHR

O *design research* enfatiza, conforme visto acima, a ligação entre duas naturezas de problemas, prática e o conhecimento, procurando mostrar que o conhecimento científico pode ser produzido através do *design* de artefatos úteis, numa lógica destacada por Wieringa:

A lógica do *design research* [pode ser entendida] como a resolução de ciclos de **problemas mutuamente aninhados, em que um problema prático pode levar à questões de conhecimento, e a resposta a essas questões de**

**conhecimento podem levar a novos problemas práticos** (WIERINGA, 2009, s/p, grifo nosso, tradução nossa).

Como a resolução desses dois tipos de problemas aninhados perfaz o âmago do *design research*, e, conforme ressaltado anteriormente, suas soluções e métodos de solução são diferentes, portanto, equívocos de identificação entre os tipos de problema pode gerar situações negativas, conforme Wieringa (2009, s/p): “[...] problemas práticos e problemas de conhecimento no *design research* podem ser confundidos, e isto pode criar problemas metodológicos”. Cabe, portanto, antes de se utilizar seus conceitos na descrição dos passos da pesquisa aqui desenvolvida, procurar esclarecer as distinções entre estes conceitos.

#### 4.3.1 Problemas Práticos e Questões de Conhecimento

Problemas práticos visam uma mudança no mundo de modo que este venha a corresponder a algum(s) objetivo(s) esperado(s). Problemas de conhecimento, por sua vez, não buscam causar modificações no mundo, mas modificar o conhecimento que se possui a respeito deste último. Estes dois tipos de problema são cruciais para o *design research*, apresentando, cada um, por sua vez, os seus desafios de resolução, porém, deve-se notar as diferenças metodológicas correspondentes a cada um. A solução de um problema de conhecimento possui natureza lógica, pois são proposições que requerem como resposta serem verdadeiras, sendo que estas respostas envolvem a aplicação de critérios de verdade e falsidade, independentes dos interessados na resposta.

A razão para que se tenha clara a distinção entre os dois tipos de problema é ressaltada por Wieringa:

A razão para se fazer a distinção [entre problemas práticos e de conhecimento] é que problemas práticos são resolvidos por modificações no mundo de acordo com as pretensões dos interessados, mas os problemas de conhecimento são resolvidos pela formulação de proposições a respeito do mundo (WIERINGA, 2009, s/p, tradução nossa).

É usual distinguir quatro razões não exclusivas para investigar problemas práticos (Wieringa, 2009), cada um deles leva a diferentes ênfases no processo de investigação:

- Investigação orientada a **problemas**: Experimenta-se um problema que necessita ser diagnosticado antes de ser solucionado;
- Investigação orientada a **objetivos**: Refere-se a uma situação em que não se

experiência um problema, porém, há razões para modificar o mundo em concordância com objetivos dados;

- Investigação orientada a **soluções**: Refere-se a uma situação em que busca-se para uma dada tecnologia, já existente, um problema que pode ser solucionado por seu intermédio. Neste caso, se inicia por uma investigação das propriedades da nova tecnologia, e o *design* das soluções será uma exploração das maneiras em que poderá ser utilizada para se atingir novos objetivos;
- Investigação orientada a **impactos**: também chamada de pesquisa avaliativa, onde se detém em ações passadas para se preparar para o *design* de futuras soluções.

Pelas características do problema que se deseja resolver, buscando encontrar uma representação para os artefatos de conhecimento clínico, ou arquétipos OpenEHR, como conteúdos num CMS a pesquisa proposta neste trabalho pode, pela classificação exposta acima, ser categorizada como orientada a problemas.

Como ressaltado anteriormente, o aninhamento de problemas apresenta-se algo comum no processo do *design research*, onde, por exemplo, problemas de *design* podem conter problemas de conhecimento e estes, por sua vez, podem conter problemas práticos. Dessa forma, Wieringa (2009) propõe um refinamento na classificação dos problemas

- Problemas práticos
  - Problemas de *design*: Como construir ou aperfeiçoar algo?
  - Problemas de implementação: Implementar o *design* de um aperfeiçoamento.
- Problemas/questões de conhecimento
  - Questões Conceituais
    - Modelagem conceitual: Quais conceitos serão usados?
    - Análise conceitual: quais são as relações conceituais entre esses conceitos?
  - Questões empíricas
    - Descrição: Quais são os fatos?
    - Explicação: Quais são suas causas?

- Predição: Quais são seus efeitos?
- Avaliação: Como fatos/causas/impactos comparam-se aos critérios?

Dessa forma, pela proposta metodológica de Wieringa (2009), **um projeto de *design* será estruturado como um conjunto de problemas aninhados em que o problema de nível mais alto é sempre um problema prático**. Na busca por sua solução, decompõem-se, então, os problemas práticos em subproblemas de conhecimento e subproblemas práticos. Sendo que, os problemas poderiam ser decompostos sempre, embora na prática, a decomposição possa parar quando se atinge um subproblema para o qual já se sabe o que fazer para obter a sua resposta.

Seguindo a categorização dos problemas mostrada acima, apresenta-se a seguir a sequência de passos que descreve o caminho percorrido no desenvolvimento da pesquisa. Para melhor compreensão do fluxo descrito pela figura a seguir, deve-se ter em conta a ordem de leitura determinada pelos dois tipos de setas: linha em cor negra (sempre na vertical) e cheia na mesma cor de fundo dos blocos (na horizontal). As setas cheias descrevem o desenvolvimento de um dado problema num fluxo horizontal, enquanto as setas negras descrevem a passagem de um problema para outro na vertical. Assim, sempre que houver um bloco que dê saída para os dois tipos de seta, deve-se, primeiro, acompanhar o fluxo descrito pelas setas cheias até o seu final, para, só então, seguir a seta negra que leva ao problema descrito por um quadro abaixo.

Os Quadros (1), (2), (3) e (4) referem-se à trajetória que parte de um problema de *design* (1), que leva à fundamentação teórica da pesquisa, referente aos Capítulos iniciais até o Capítulo 5. Tal caminho versa sobre os fundamentos do padrão OpenEHR, permitindo que se caracterize funcionalmente os modelos de referência, arquétipos e serviço, identificando o responsável pela semântica do padrão, neste caso, **o modelo de referência, bem como os modelos de informação que o compõe**, associados aos dados clínicos.

Os quadros (5), (6), (7), (8), (9) e (10) descrevem a trajetória de pesquisa apoiada num problema prático (8) que deu origem ao Capítulo 6, onde, inicialmente, se elenca e verifica a adequação dos tipos e estruturas da linguagem Python com os tipos assumidos pela especificação OpenEHR. Em seguida, os diversos pacotes de classes propostos na especificação para a representação de dados clínicos são implementados corretamente nesta linguagem, permitindo concluir que esta última suporta a semântica da especificação, abrindo-

se espaço para seu uso na criação de arquétipos.

Os quadros de números (11) a (16) retratam a tentativa de expressar artefatos de conhecimento ou arquétipos OpenEHR via *framework*<sup>17</sup> Plone/Archetypes<sup>18</sup>. Tal esforço levou à percepção do custo benefício insatisfatório ao se buscar expressar arquétipos OpenEHR através desse framework, bem como trazendo o problema de conhecimento (16) expresso pela dúvida sobre o potencial expressivo dos conteúdos CMS frente à complexidade dos arquétipos.

A busca por responder a tal questão, cuja resposta positiva poderia expressar obstáculo a implementações satisfatórias dos arquétipos como conteúdos em CMS's, é descrita pelos quadros (17) à (21). Este esforço de resolução busca fundamentar os conceitos de arquétipo e conteúdo, identificando seus elementos mais relevantes para, em seguida, buscar aqueles que são comuns a ambos, nos quadros (17) e (18). O conceito de metadata apresenta-se como elemento sugestivo para tal ligação conceitual (19), pois a caracterização dos CMS's como metadata frameworks (20) permite estabelecer a ligação entre os dois conceitos chave de ambos os domínios: arquétipo e conteúdo (21). As inferências obtidas através deste resultado (22), sua contextualização frente ao problema da tese e suas consequências são expostas e analisadas nos Capítulos 7 e 8 deste trabalho.

---

<sup>17</sup> Plone/Archetypes: < <http://plone.org/products/archetypes>>

<sup>18</sup> É importante não confundir os conceitos de *Plone/Archetypes* e *OpenEHR/Archetypes*. Embora similares, como mostra a pesquisa. *Plone/Archetypes* é uma ferramenta para representar conteúdos no CMS Plone, já *OpenEHR/Archetypes* é a parte do padrão OpenEHR voltada à representação de conhecimento clínico.

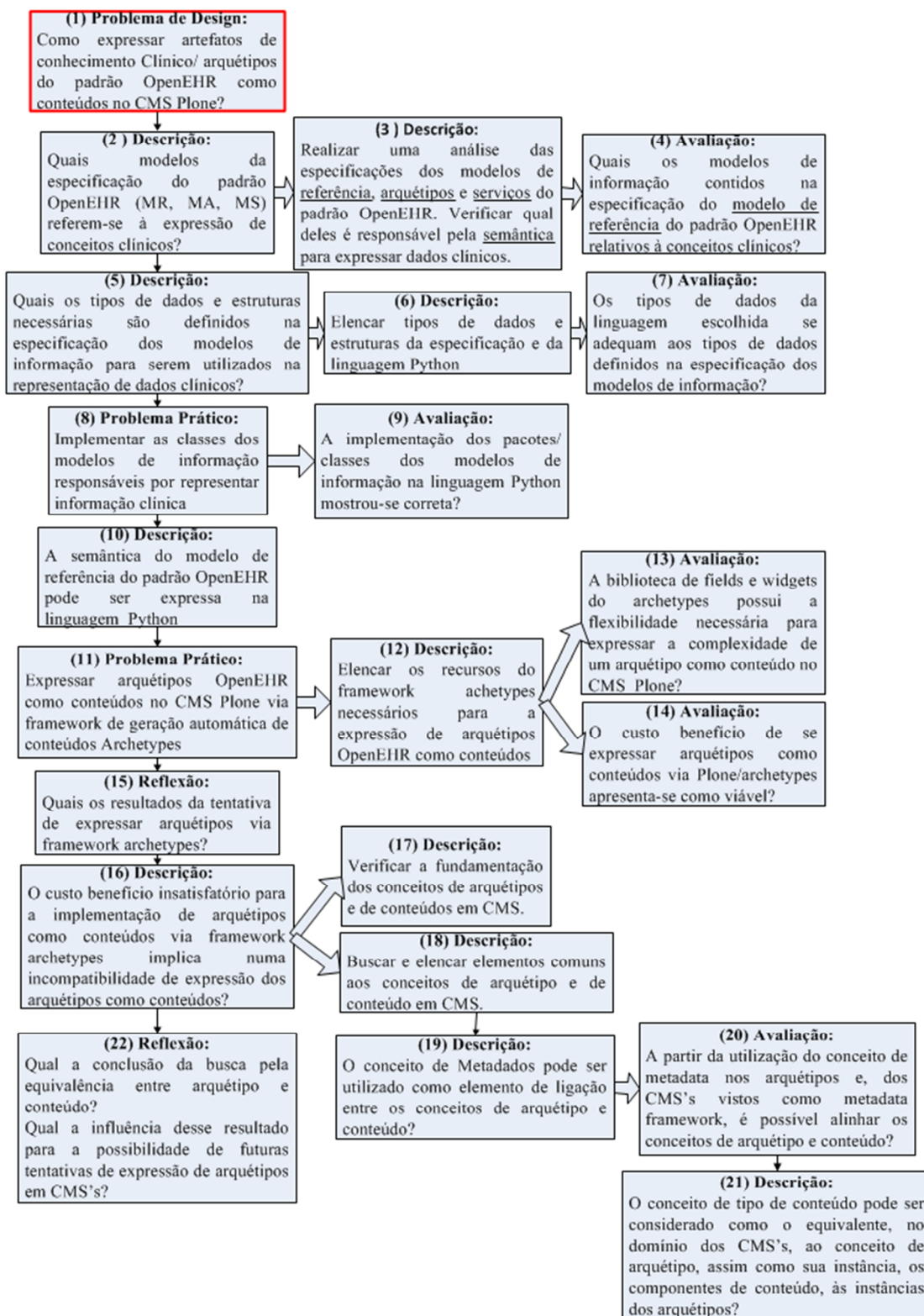


Figura 4 - Estrutura de Problemas Aninhados Referentes à Trajetória da Pesquisa  
Fonte: Elaborado pelo Autor



#### 4.4 Trabalhos Correlatos

Apresenta-se, nesta seção, um breve relato sobre projetos e iniciativas relacionados ao tema central deste trabalho de pesquisa, que busca verificar a possibilidade de expressar arquétipos OpenEHR na plataforma Python e em CMS's.

A necessidade e os ganhos advindos da ampliação das possibilidades expressivas do padrão OpenEHR para outras plataformas de programação para além das utilizadas em seu desenvolvimento (destacadas nos parágrafos a seguir), é ressaltada pelos autores dos trabalhos descritos nesta seção. O esforço de implementação do padrão OpenEHR em Java, encorajou, conforme Chen (2007), o seu uso em diversos projetos ao redor do mundo. De modo semelhante, Kobayashi e Tatsukawa (2012) ressaltam a importância de ampliar a base de desenvolvedores que utilizam o padrão OpenEHR provendo implementações que os atendam nas diversas plataformas que adotam.

Os trabalhos acadêmicos descritos a seguir configuram-se em esforços objetivando obter condições para a expressão de informações clínicas conforme a semântica definida pela especificação do padrão OpenEHR nas linguagens Java e Ruby.

Embora a implementação descrita em Atalag, Yang e Warren (2010) faça uso de uma biblioteca que implementa o modelo de referência OpenEHR na linguagem C#, o desenvolvimento desta última deu-se em ambiente e para uso comercial, especificamente pela *Ocean Informatics*<sup>19</sup>, não estando seu esforço de implementação associado a trabalho de cunho acadêmico. Por este motivo optou-se por não elencá-lo junto aos demais trabalhos relacionados. O mesmo critério foi aplicado à implementação do padrão OpenEHR, feita pela mesma *Ocean*, na linguagem Eiffel.

##### 4.4.1 Rumo a um Registro Médico Eletrônico Interoperável e Baseado em Conhecimento via Metodologia de Arquétipos (CHEN, 2009)

O autor inicia sua tese constatando a importância da aquisição, estruturação e uso da informação biomédica e do papel chave das pesquisas de melhores maneiras de modelar a estrutura dessa informação, do que aquelas existentes hoje. O reconhecimento dos vários

---

<sup>19</sup> Ocean Informatics. Página institucional da plataforma OpenEHR .NET: <<http://oceaninformatics.com/platform/technology>>.

benefícios advindos do RES são ressaltados, bem como sua consequência imediata, a saber, a multiplicação de sistemas RES construídos sobre diferentes plataformas e sem padronização conceitual ou de estruturas de dados, limitando sua possibilidade de compartilhamento.

Com a constatação de que o volume de conhecimento no domínio da biomedicina vem crescendo exponencialmente, discute-se como a tradicional disseminação manual do conhecimento requer cada vez mais esforços por parte dos profissionais clínicos. O conceito de interoperabilidade é, então, apresentado e discutido, mostrando-se essencial para o desenvolvimento futuro dos sistemas RES.

A seguir, são apresentados os esforços atuais em busca da interoperabilidade via padronização, como o da norma ISO 13606, HL7 CDA, SNOMED CT e o padrão OpenEHR, expondo-se algumas de suas características. Definições de interoperabilidade são apresentadas e discutidas, bem como a questão dos limites existentes entre tais modelos, tendo em vista a troca de informações. O modelo de arquétipos é apresentado com potencial de uso em tais situações. O seu uso não resolve automaticamente o problema, serve, porém, como espaço de discussão, para que especialistas trabalhem e clareiem a definição do arquétipo.

A tese consiste em cinco artigos dedicados à investigação de diferentes aspectos da interoperabilidade dos sistemas de registro médico eletrônicos. O primeiro descreve o sistema de *templates* Julius, que busca prover uma solução mais flexível que permita aos clínicos definirem os dados a serem salvos e compartilhados. Tais *templates*, usados para entrada de dados, podem ser usados em combinação com sistemas já existentes. O segundo trata da implementação da especificação OpenEHR na linguagem de programação Java, gerando a especificação OpenEHR para esta linguagem (Java<sup>20</sup>), agora incluída como parte da especificação OpenEHR. Na forma de um projeto de código livre, possibilitou a geração de sinergia entre projetos acadêmicos e comerciais que passaram a dispor de um *framework* comum, indicando possíveis contribuições e direcionamentos para a interoperabilidade de projetos de *software* livre. O terceiro trata de um estudo de conversão entre modelos de conteúdo de diferentes sistemas de RES através de arquétipos OpenEHR, usando uma técnica de mapeamento semântico. O quarto trata da representação de um guia de cromoterapia utilizando a especificação OpenEHR e o quinto e último da modelagem de componentes estruturais de planos de cuidado em saúde através das classes do modelo de referência

---

<sup>20</sup> Linguagem Java, página institucional: <<http://www.oracle.com/us/technologies/java/overview/index.html>>.

OpenEHR.

Entre as diversas conclusões e análises deste trabalho acadêmico, ressalta-se o uso de um modelo de referência, no caso o do padrão OpenEHR, como base para a representação de instâncias de informação em diferentes e relevantes cenários: proposta de arquétipos para uso em diferentes *templates* para planos de cuidado e representação de guias clínicos. A simplificação das tarefas de autoria pelo reuso de modelos torna sua manutenção mais fácil bem como a governança através de um ambiente compartilhado.

#### 4.4.2 Implementação das Especificações OpenEHR em Ruby (Kobayashi; Tatsukawa, 2012).

Os autores, como motivação para o projeto, partem da constatação de que a especificação do padrão OpenEHR possui implementação formalmente descrita nas linguagens de programação Eiffel, C# e Java mas não numa linguagem de *scripting*. Tais linguagens são apreciadas por proporcionarem eficiência e desenvolvimento rápido<sup>21</sup>.

Um grupo de médicos japoneses, especializados na programação de sistemas de informação motivados pelos conceitos de *design* do padrão OpenEHR, propuseram sua implementação, como código *open source*, numa versão em Ruby<sup>22</sup>. A linguagem de *scripting* Ruby é escolhida pela sua capacidade de desenvolvimento eficiente e, também, visando investigar a universalidade da aplicação da especificação OpenEHR.

O modelo de referência do padrão OpenEHR, que define os conceitos básicos, tipos de dados e estruturas de dados foi inicialmente definido na linguagem Eiffel<sup>23</sup> via padrão de *design* programação por contrato<sup>24</sup> (abreviada por *dbc*, do inglês *design by contract*). O tratamento e verificação das invariâncias, tornou-se um dos pontos de dificuldade da implementação, uma vez que a linguagem Ruby não suporta, nativamente, a programação por

---

<sup>21</sup> Linguagens de scripting são similares às linguagens de programação imperativas em vários aspectos. Assim, linguagens de scripting suportam variáveis, comandos, e procedimentos, que são conceitos chave da programação imperativa. Várias linguagens de scripting são dinamicamente tipadas. Quando usadas para “ligar” sistemas diferentes, scripts necessitam possuir a capacidade de enviar dados para e entre subsistemas escritos em diferentes linguagens. (Watt; Findlay, 2004)

<sup>22</sup> Página institucional da linguagem Ruby: <<http://www.ruby-lang.org/>>

<sup>23</sup> Página institucional da linguagem Eiffel: <<http://www.eiffel.com/>>

<sup>24</sup> A programação por contrato é um abordagem de desenvolvimento de software que prescreve que os desenvolvedores devem definir Métodos formais, especificações de interface precisas e verificáveis dos componentes de desenvolvimento de software, com pré-condições, pós-condições e constantes. Estas especificações são definidas como um "contrato", de acordo com os próprios conceitos de condições e obrigações dos contratos de negócio. (Meyer, 2000)

contrato e, portanto, não consegue realizar tal verificação. Conforme Meyer (2000, p. 363), “O conceito de invariância de classe é um dos conceitos principais do dbc, onde se verifica que o estado de uma classe é mantido dentro de dadas tolerâncias especificadas ao final de cada execução”.

A herança múltipla não se mostrou um problema, apesar de nem toda linguagem orientada à objetos implementá-la, pois a linguagem Ruby consegue fazê-lo mantendo a simplicidade. O mesmo ocorreu com a importação circular, outro obstáculo advindo da influência da linguagem Eiffel na especificação OpenEHR que causaria problemas em outras linguagens como o Java, pode ser resolvido em Ruby. Tal situação sugere que a especificação possa vir a ser rerepresentada num formato que exclua tais problemas advindos do *dbc*.

Uma vez que o modelo de arquétipos é construído, criou-se, em seguida, um *parser* ADL visando a serialização de arquétipos nesta linguagem (ADL: *archetype description language*).

Várias especificações OpenEHR foram implementadas bem como o parser ADL, tornando possível a comparação com as *core libraries* de outras implementações. Como as terminologias representam papel importante na seção *ontology* de um arquétipo, em caráter experimental, um servidor de terminologias (via protocolo REST) também foi implementado.

Nos testes de desempenho, porém, ficou claro a desvantagem da linguagem Ruby em termos de velocidade, a diferença ficou de 20 a 300 aquém da implementação em Java o que se mostra claramente um problema em termos de desempenho.

Como conclusão, os autores levantam problemas encontrados na implementação, entre eles o próprio *parser* ADL que necessita ser trabalhado, além da inexperiência dos desenvolvedores de sistemas em saúde. Porém, ressaltam que a linguagem possui potencial para resolver tais problemas, podendo tornar-se uma plataforma moderna e prática para sistemas de informação em saúde.

## 5 INTEROPERABILIDADE SEMÂNTICA EM RES: O PADRÃO OpenEHR

Neste capítulo, o padrão para interoperabilidade de dados clínicos OpenEHR é apresentado. Inicialmente, apresenta-se a geração de informações clínicas a partir dos processos clínico e administrativo, bem como a caracterização de cada tipo de informação gerada para, a seguir categorizá-las através da ontologia CIR.

A seguir, mostra-se a definição do modelo de referência a partir da ontologia CIR, permitindo a especificação da classe ENTRY, responsável pela definição das entradas do modelo de referência OpenEHR.

Após a discussão desta parte da especificação do padrão, a representação e a organização de fatos clínicos através do modelo de conhecimento (ou de arquétipos) é apresentado discutindo-se as vantagens do modelo dual como a camada de conhecimento não ser implementada em *software*, o reuso dos arquétipos definidos por especialistas, a componentização do sistema, entre outras.

### 5.1 Contextualização

Tornar os registros eletrônicos de saúde interoperáveis é, conforme salienta Chen (2009), um pré-requisito para o suporte à cada vez mais sistemas de saúde distribuídos.

Trazer conhecimento a tais sistemas para obter suporte à decisão médica é um passo crítico para promover o cuidado baseado em evidência, ou seja, dados provenientes de diferentes sistemas necessitam ser analisados em busca de novas evidências a serem acrescentadas à prática. Para tanto, torna-se necessário o arranjo de sistemas de PEP/RES, capazes de compartilhar dados clínicos de pacientes com outros sistemas, via uso de um **modelo de referência, estruturas de dados clínicos e terminologias, preservando a semântica existente no domínio de conhecimento**, atualizando e recuperando dados de cada paciente de forma consistente, sem ambiguidade.

Conforme salientam Ingram e Sevket (2013), a necessidade de uma abordagem *open source* para o compartilhamento de informações apresenta-se como inadiável:

Estas áreas de interesse [*design* para adaptabilidade e governança] relacionam-se diretamente às razões para se requerer transparência nos sistemas e no aumento das perspectivas quanto ao progresso na pesquisa para a promoção de uma abordagem de *software open source*, notavelmente

em áreas chave como infraestrutura de informação, onde a **interoperabilidade entre sistemas é essencial** (INGRAM; SEVKET, 2013, p.35, grifo nosso, tradução nossa).

Buscando contextualizar num breve histórico os esforços na direção descrita acima, pode-se, sem perda de generalidade, marcar seu início em 1988, ano em que a união Europeia estabeleceu a iniciativa *Advanced Informatics in Medicine* (A.I.M.), como parte de seu programa para desenvolvimento de pesquisa e tecnologia. O projeto de pesquisa *Good Electronic Health Record* (GEHR) se iniciou em 1992, como parte da iniciativa A.I.M., e mais tarde continuou com forte participação de grupos de pesquisa da Austrália. Seu principal objetivo foi o estudo de abordagens de modelagem de objetos para sistemas de RES e questões sobre implementação desses sistemas. Pode-se afirmar que o conceito de **arquétipos** foi inicialmente posto neste projeto (OPENEHR, 2010). Em 1995, foi criado um novo projeto denominado *Synapses*, liderado pela *Trinity College Dublin*, na Irlanda. Baseada na utilização de tecnologias de sistemas de bancos de dados federados, sua arquitetura utilizava um modelo de informação (canônico) – denominado *Synapses Object Model* (*Synom*) - que fornecia uma **representação formal para as características genéricas dos registros de pacientes, e um modelo de conhecimento** que armazenava as definições - chamado *Synapses Object Dictionary* (*Synod*) (OPENEHR, 2010). Em 1998, a **fundação *OpenEHR*** foi constituída dando continuidade a esse esforço de pesquisa, criando uma **arquitetura fundamentada em um modelo de informação (canônico) e um modelo de conhecimento baseado em arquétipos**, denominando tal abordagem como **modelagem de dois níveis** (BEALE, 2002).

## 5.2 O Padrão OpenEHR

Dentro de tal contexto e objetivos, a fundação OpenEHR apresenta-se como uma fundação não lucrativa que objetiva tornar realidade a interoperabilidade entre sistemas de RES (OPENEHR, 2010).

De acordo com a descrição de Leslie:

A fundação *openEHR* oferece um conjunto de especificações para uma arquitetura de RES. Seu propósito é habilitar a interoperabilidade semântica da informação em saúde entre, e dentro, de sistemas de RES – tudo em um formato não proprietário, evitando a detenção de direitos por parte de fornecedores (LESLIE, 2007, p.51, tradução nossa).

Constatando-se a necessidade de uma base teórica robusta aos modelos de informação

clínica, bem como objetivando garantir aos sistemas de RES requisitos desejáveis como interoperabilidade, computabilidade, escalabilidade, viabilidade econômica e desempenho, partiu-se de uma **ontologia** para desenvolver a base formal do modelo OpenEHR.

### 5.2.1 Processos Clínicos e Processos de Negócio

A trajetória de desenvolvimento da ontologia da informação clínica descrita por Beale (2007) cujos pontos relevantes serão apresentados na sequência se inicia a partir de dois tipos de processo:

- Processo clínico: que descreve a interação entre o sistema responsável pela investigação clínica e o sistema que representa o paciente.
- Processo de negócio: contém o processo clínico descrito acima e se insere no contexto administrativo conforme mostra a Figura 5 a seguir.



Figura 5 - Metáfora da Resolução de Problemas  
Fonte: Traduzido de Beale (2007)

A ilustração apresenta o processo clínico como a interação de dois sistemas/entidades: o paciente, normalmente compreendido como um indivíduo, embora possa representar mais de um e o investigador, representando a totalidade dos profissionais de saúde que podem realizar alguma ação no cuidado do paciente, além do próprio paciente (automedicação), parentes e outras pessoas. Este sistema de investigação, orientado a objetivos, utiliza observações como suporte para avaliações e intervenções.

### 5.2.2 A Geração de Informação nos Processos Clínicos e de Negócio

Uma vez que se deseja a serialização e troca de mensagens entre sistemas de informação,

Beale (2007) apresentará a figura abaixo, que busca apresentar os tipos de informação que podem ser criadas nos processos anteriormente descritos.



Figura 6 - Informações Criadas pelo Investigador Clínico  
Fonte: Traduzido de Beale (2007)

Na figura acima podemos ver a representação da geração de informações durante o processo de resolução de problemas clínicos. O processo se inicia com o médico recebendo o paciente, sendo que a primeira coisa que faz é **observá-lo**, realizar medições (temperatura, pressão arterial etc.), fazer perguntas. A informação vinda destas observações iniciais é registrada. A seguir, o médico realiza **avaliações**, define objetivos e planos de cuidado para com o paciente, baseando-se em seu conhecimento e em fontes externas de conhecimento. A informação sobre as avaliações também deve ser registrada. O passo seguinte é dar **instruções**, indicações e ordens. Desde solicitar que se faça um estudo de laboratório, até passar indicações a um paciente que está recebendo alta. Por último, encerra-se o ciclo com a **execução das instruções**, o que vemos nomeado na figura como "**ações**". Se o problema não é considerado resolvido, o ciclo é novamente executado, com tudo sendo, novamente, registrado.

Conforme se vê na figura, cinco tipos distintos de informação podem ser criados durante o processo de atendimento ao paciente. A seguir apresenta-se sua descrição.

- Observações: informações criadas por um ato de observação, medição, questionamento ou teste de um paciente ou substância relacionada (ex: urina ou



tecido).

- Opiniões: inferências realizadas pelo investigador utilizando seus conhecimentos pessoais a respeito do conteúdo das observações. Inclui diagnósticos, avaliações, planos e objetivos.
- Instruções: instruções, baseadas em observações, detalhadas o suficiente para serem executadas por agentes investigadores (pessoas ou máquinas).
- Ações: registro de ações de intervenção ocorridas via instruções ou outra causa.
- Eventos administrativos: registro de eventos ocorridos no contexto administrativo como admissões, agendamentos, liberações.

### 5.2.3 A Ontologia CIR

A partir destas categorias Beale (2007) propõe uma ontologia inicial, que tem o objetivo de situar os tipos de informação apresentados com respeito às categorias de informações administrativas (*admin information*) e informações de cuidados (*care information*) conforme figura abaixo:



Figura 7 - Ontologia da Informação Clínica Inicial  
Fonte: Traduzido de Beale (2007)

Analisando as categorias criadas sob a perspectiva da temporalidade e, assumindo que “observações” são entendidas como “quaisquer informações recebidas pelo investigador clínico, usadas para caracterizar o estado do paciente” (Beale, 2007, p.760), retira-se daí que estas estão no passado, pois expressam fatos já ocorridos no tempo, logo, de natureza histórica. Similarmente, a categoria de informação “ação” pode ser considerada como de

natureza histórica, uma vez que registros de ações efetuadas não podem ser feitos até que estas sejam completadas (ações a serem tomadas serão “instruções”).

Utilizando as categorias de alto nível propostas por Sowa (2000), estas duas categorias da ontologia clínica se adequam à categoria denominada por este como “História”, conforme mostra a Figura 8:

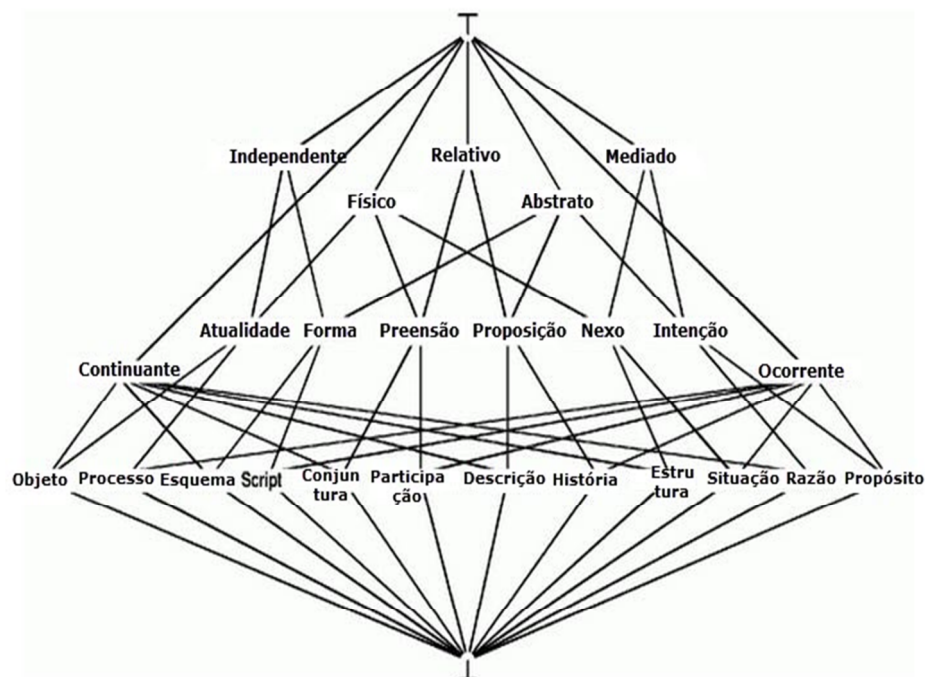


Figura 8 - Categorias de Alto Nível  
Fonte: Traduzido de Sowa (2000)

Mantendo a temporalidade como baliza, pode-se situar análises, avaliações e planejamentos. Em suma, atividades analíticas e de tomada de decisão, como estando no tempo presente, pois estas vêm entre as evidências percebidas e as ações a serem tomadas. A categoria de informação clínica “opinião” versará sobre pensamentos e análises sobre fatos registrados e conterà as subcategorias acima e corresponderá à categoria “descrição” nas categorias de alto nível de Sowa (2000), conforme a Figura 8.

A categoria “opinião” divergira da categoria “observação” por representar inferências retiradas das evidências observadas e não as evidências em si mesmas.

O tempo futuro será expresso pela categoria denominada por Beale (2007) como “instruções”, estando suas informações na forma de ordens, requisições e outros

direcionamentos a serem seguidos, referindo-se a eventos que ainda não ocorreram ou ainda não apresentaram seu término. Conterá as duas subcategorias denominadas requisição de investigação para se realizar observações adicionais e requisição de intervenção visando a modificação do estado do paciente. Em termos das categorias de alto nível mostradas na Figura 8, as instruções correspondem à categoria “Script”.

Na figura 9 pode-se ver a ontologia inicial proposta (Figura 7) acrescida das categorias descritas acima, formando a ontologia da informação clínica CIR (*Clinical Investigator Record*).

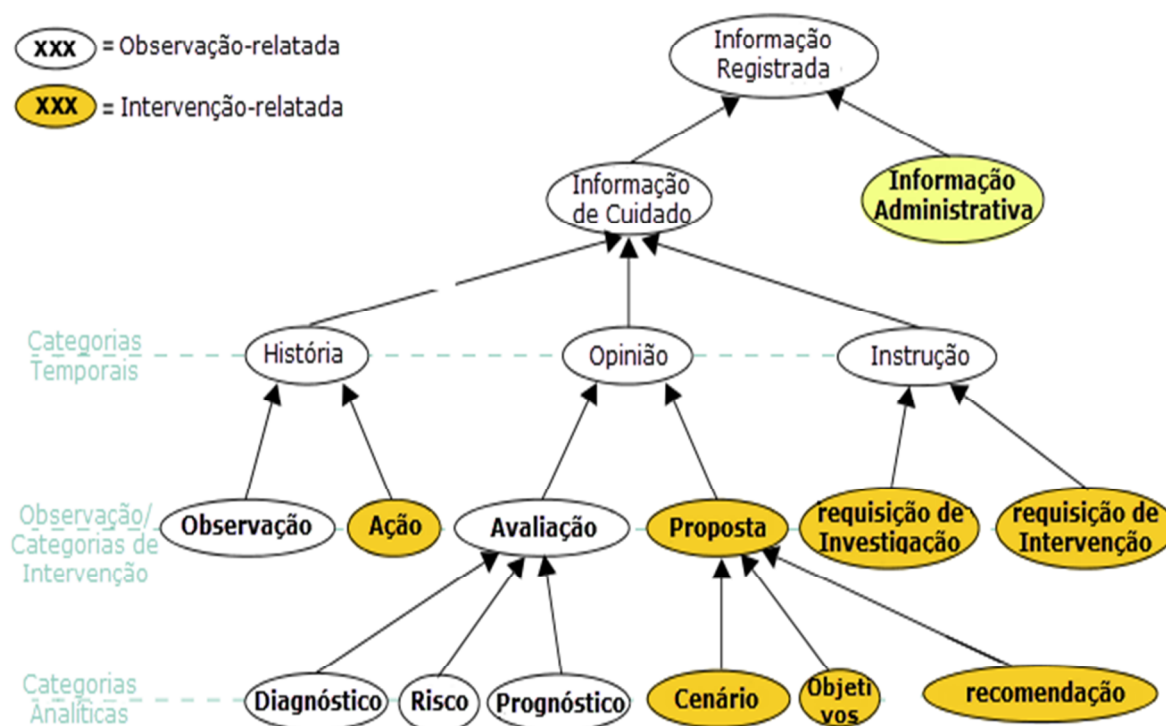


Figura 9 - A Ontologia da Informação Clínica (CIR)  
 Fonte: Traduzido de Beale (2007)

Com as categorias da ontologia CIR, pode-se classificar entradas para representar conceitos clínicos, como nos exemplos abaixo:

- Tomar a pressão arterial: observação
- Frequência cardíaca: observação

- Frequência respiratória: observação
- Resultado de laboratório: observação
- Avaliação das vias aéreas: avaliação
- Classificação da gravidade do estado do paciente (Triagem): avaliação
- Avaliação de disfunção neurológica: avaliação
- Ordem para realização exames de laboratório: instrução
- Indicações para o paciente: instrução
- Diagnósticos: instrução
- Implante de cateter: ação
- Administração de substâncias: ação

#### **5.2.4 Uma Ontologia Voltada à Informação**

Conforme apresentado no Capítulo 3 uma ontologia pode ser entendida como uma descrição de algum aspecto da realidade. Para efeito de um melhor entendimento da motivação da ontologia CIR, poder-se-ia categorizar as ontologias como sendo:

- Ontologias da realidade: Que versam sobre coisas reais, eventos e processos, em vez de informação.
- Ontologias da informação: Que versam sobre qualquer tipo de informação, ou seja, entidades que possuam comprometimento com algum tipo de meio como o escrito, audiovisual, etc.

Pode-se situar a ontologia CIR na categoria das ontologias de informação. Com tal categorização não se está negando a pertença das informações à realidade, porém, ressaltando-se que algo da realidade está sendo registrado e possui características como:

- Tipo de entidade registrada, como notas, resultados de exames, diagnósticos, etc.
- A estrutura dos registros feitos.
- Relações entre informações registradas, como “maiores detalhes” de algo, “informações suplementares”, ou outra informação advinda de quem faz o registro e que não ocorrem entre os entes da realidade que são os motivos dos registros.

Pois, claramente, a ontologia CIR, base da classe ENTRY (entradas) do padrão OpenEHR é definida para versar sobre informações, mais especificamente sobre informações em saúde. Pois, o ponto mais importante sobre os arquétipos, que serão construídos a partir dela, é que estes não são descrições de coisas reais, mas são registros de algo que despertou o interesse/chamou a atenção do profissional de saúde durante o processo clínico, ou seja, capturam o que os profissionais de saúde, por sua experiência ou seguindo um procedimento padrão, julgam ser necessário registrar. Desse modo, norteados pelas categorias definidas na ontologia CIR, os profissionais de saúde, ao realizar as entradas de informações via formulários, o farão de maneira mais intuitiva e com menor curva de aprendizado, pois as categorias nela definidas buscam corresponder a informações geradas durante o fluxo de trabalho desses profissionais. Conforme ressaltado por Sinha *et al.*:

O modelo OpenEHR, baseado na abordagem problema-resolução para representar a informação clínica provê uma abordagem realista para sistemas de RES que se estendam ao longo do fluxo de trabalho do hospital. O modelo ontológico OpenEHR tem sido referenciado por vários programas nacionais voltados para a construção seus próprios sistemas de RES e padrões para a conteúdos de informação clínica em RES (SINHA *et al.*, 2013, p.164, tradução nossa).

### **5.2.5 A Hierarquia Conceitual da Ontologia CIR e a Classe ENTRY do Modelo de Referência OpenEHR**

A ontologia CIR provê a base para as classes de entrada (classes *Entry*) do modelo de referência definido pelo padrão OpenEHR, com a principal característica de ser um modelo genérico, que permite representar a informação gerada na assistência à saúde, porém, sem a especificação semântica dos conceitos clínicos particulares.

Na Figura 10 observa-se, de modo esquemático, a hierarquia conceitual definida na ontologia CIR, iniciando-se com uma superclasse abstrata denominada ENTRY seguida de duas subclasses responsáveis pelas entradas para informações do tipo administrativo (ADMIN\_ENTRY), bem como a de cuidados médicos (CARE\_ENTRY) seguida de quatro subclasses correspondendo às categorias explicitadas anteriormente quando da sua apresentação, ou seja, observação, avaliação, instrução e ação.

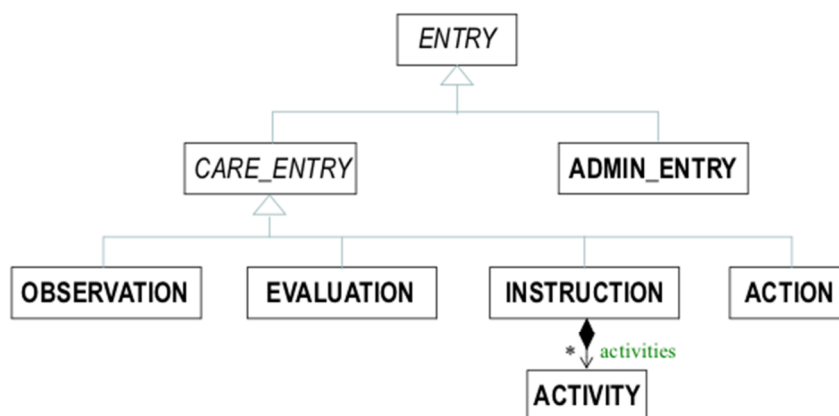


Figura 10 - A Superclasse ENTRY, Baseada na Ontologia CIR  
Fonte: Beale (2008)

Na Figura 11 apresenta-se a especificação UML correspondente à classe ENTRY do modelo de informação EHR, do modelo de referência OpenEHR, no que pode ser considerado o cerne da proposta de RES da fundação OpenEHR.

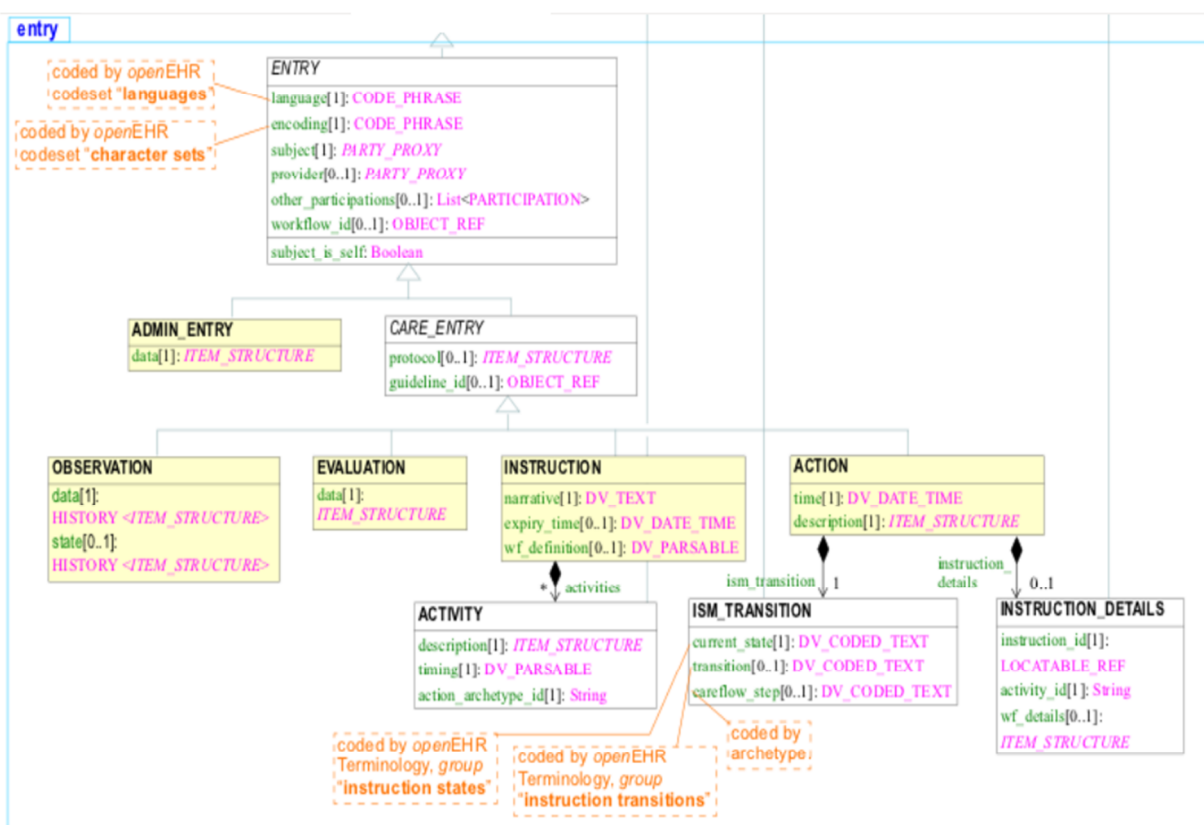


Figura 11 - A Classe ENTRY do Modelo de Referência OpenEHR, Baseada na Ontologia CIR  
Fonte - Beale (2007)

Analisando o diagrama de classes, pode-se notar que instrução possui várias atividades

relacionadas permitindo modelar situações em que as tarefas a ser desempenhadas ao se seguir tal instrução podem ser complexas e necessitar de várias atividades, ou passos, para serem realizadas. Uma ação, por sua vez, pode estar relacionada aos detalhes de uma instrução, para indicar que tal ação foi realizada como parte das atividades requisitadas por uma dada instrução.

Como uma ação pode ser não pontual necessitando ser executada por um dado período de tempo, como por exemplo, a administração de algum medicamento durante um período estabelecido, em intervalos regulares de tempo, torna-se necessário expressar as diversas situações que podem ocorrer durante esta ação não pontual, como início da ação, término, interrupção, cancelamento da ação etc. Para representar tais situações, cada ação traz associada uma máquina de estados finitos (Figura 12), que permite que se conheça o estado de qualquer intervenção que esteja sendo realizada junto ao paciente.

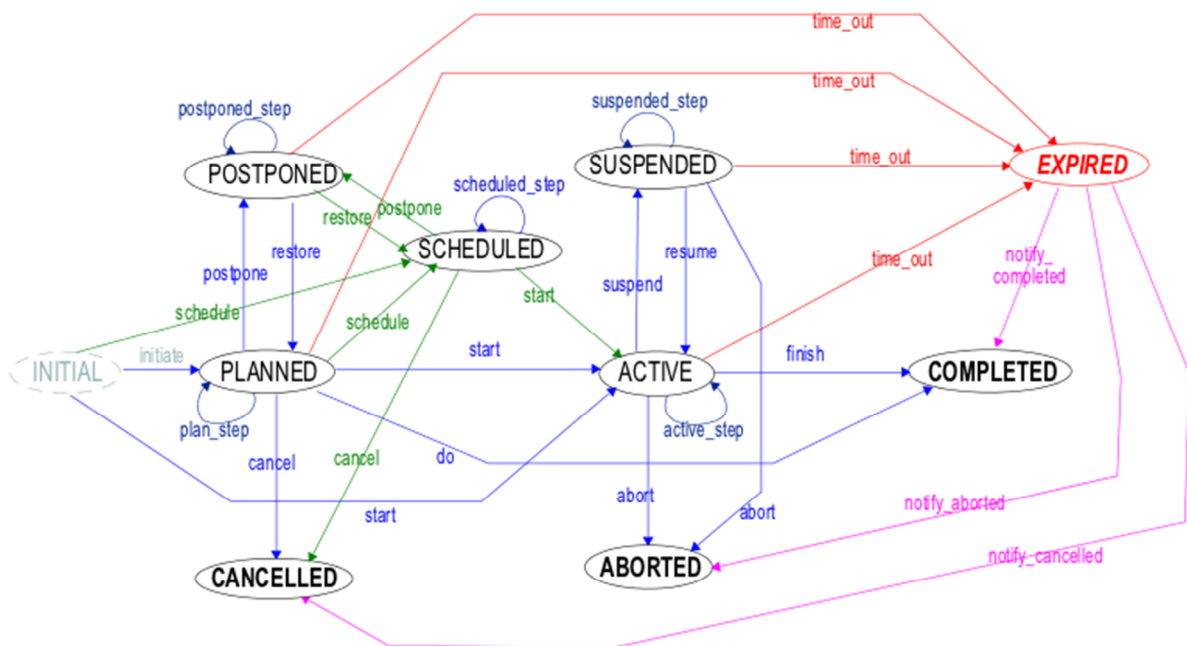


Figura 12 - Máquina de Estados Finitos Associada à Subclasse ACTION  
Fonte: Beale (2008)

Conforme foi explicitado anteriormente, o modelo de referência OpenEHR possui a característica de ser genérico. Como, então, poderão ser representados conceitos clínicos específicos como, por exemplo, a pressão arterial do paciente? Por exemplo, nos modelos de informação vistos até o momento, não há nenhuma classe específica para representar pressão

arterial, mas existe uma classe que possui os conceitos clínicos necessários para expressar medições a respeito do paciente: a classe (subclasse de ENTRY) observação (OBSERVATION). Portanto, pressão arterial será definida como uma observação. Mas surge, em seguida, a questão: como expressar as particularidades desta observação? A resposta a esta questão levará ao modelo de conhecimento proposto pelo padrão OpenEHR, ou modelo de arquétipos.

### 5.2.6 O Modelo de Conhecimento ou Modelo de Arquétipos do Padrão OpenEHR

Tomando como elementos constituintes os modelos de informação clínica, que permitem representar conceitos clínicos gerais, chega-se ao **modelo de conhecimento** do padrão OpenEHR, que tem por objetivo representar conceitos clínicos particulares (ao contrário do modelo de referência, que representa conceitos clínicos gerais e reside dentro do software).

Conforme Beale e Heard:

Sob a abordagem do modelo dual, **um modelo de informação, ou de referência estável constitui o primeiro nível da modelagem**, enquanto as definições formais do conteúdo clínico na forma de arquétipos e templates constituem o segundo. **Somente o primeiro nível (o modelo de referência) é implementado em software**, reduzindo significativamente a dependência de sistemas legados e dados em definições de conteúdo variáveis (BEALE; HEARD, 2008, p.15, grifo nosso, tradução nossa).

Os conceitos clínicos particulares são representados como um conjunto de restrições sobre o modelo de informação genérico. A abordagem do OpenEHR, através da modelagem de dois níveis, habilita o corpo médico a determinar as características do registro de saúde mais adequadas para suas necessidades, ilustra a separação entre a atividade dos especialistas de domínio (neste caso, em específico, a área médica) na criação dos arquétipos através do modelo genérico de informação e que, por sua vez, irão compor as bases de conhecimento clínico conforme a Figura 13.



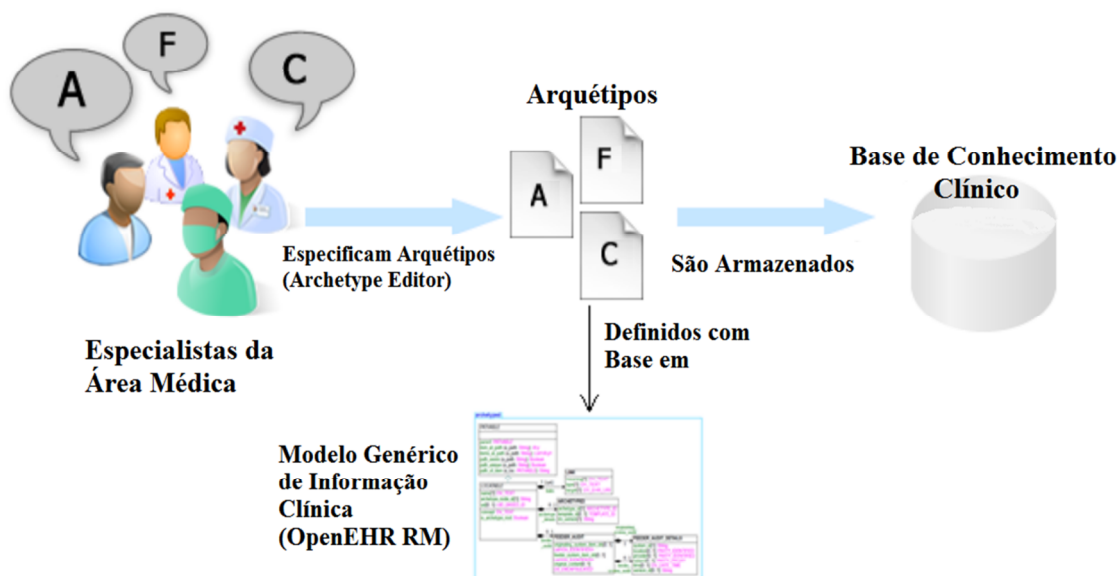


Figura 13 - Construção de Arquétipos por Especialistas Médicos  
 Fonte: Traduzido de Gutiérrez e Carrasco (2013)

Descrito sob a ótica da restrição aos objetos do modelo de referência, por Martínez-Costa *et al.*, tem-se:

O modelo de referência representa as características globais das anotações dos registros de saúde, como elas são agregadas e o contexto de informação requerido. **Este modelo define o conjunto de classes que formam os blocos genéricos de construção do registro eletrônico de saúde e contém as características não voláteis do registro eletrônico de saúde.** [...] Arquétipos aplicam restrições aos objetos, que podem ser considerados descritores dos níveis ontológicos do domínio, definidos num modelo de referência. Os arquétipos fazem a ponte entre a generalidade dos conceitos definidos no modelo de referência e a variabilidade da prática clínica, tornando-se assim, uma ferramenta para representar estes conceitos (MARTÍNEZ-COSTA *et al.*, 2009, p.151, grifo nosso, tradução nossa).

O modelo OpenEHR propõe uma estrutura de alto nível, em formato árvore, para manter os registros clínicos composta dos seguintes elementos, em ordem crescente de hierarquia:

- ELEMENT: corresponde ao nó (folha) da árvore que carrega um dado simples, como pressão sistólica, pressão diastólica, peso corporal, nome de um medicamento, etc.
- CLUSTER: permite a criação de estruturas de dados mais complexas como tabelas, árvores, como lista que reúne as medições de pressão sistólica e diastólica, ou todas as

possíveis posições em que o paciente pode estar no momento em foram tomadas as medidas ou a interpretação de um eletroencefalograma.

- ITEM: representa o tipo de dado que se está representando conforme o modelo de referência anteriormente apresentado, como observação, inferência, ação, etc.
- ENTRY: contém uma estrutura (não recursiva) de dados que utiliza uma estrutura de Clusters/Elements, como medida da pressão sanguínea, resultado de um exame etc.
- SECTION: cabeçalhos de uso opcional, que possibilitam a organização das *entries* (entradas) numa composição, como histórico familiar, alergias, sintomas subjetivos etc.
- COMPOSITION: representa o conjunto de componentes do RES que é criado durante um encontro clínico, como resumo de alta, relatório clínico, etc.
- FOLDER: é o nível mais alto do RES, utilizado para agrupar composições, podendo conter outros *folders* e informações multiprofissionais. Pode ser dividido em compartimentos como Hospital <X>, Hospital <Y>, Pediatria, Geriatria, Laboratório de análises etc.

Em um dado contexto, os dados coletados de acordo com os arquétipos são armazenados em um sistema de RES em grandes estruturas denominadas *Compositions*, que tem seus próprios arquétipos. As *Compositions* são comparáveis com um documento que resulta de um evento clínico. Ou seja, os arquétipos contêm um conjunto máximo de dados sobre cada conceito clínico, incluindo dados de atendimento requeridos como: protocolos, ou métodos de medição; eventos relacionados; e contexto que é requerido para o dado clínico ser interpretado corretamente.

Uma ilustração da hierarquia dos elementos em árvore de uma *composition* pode ser vista na Figura 14.

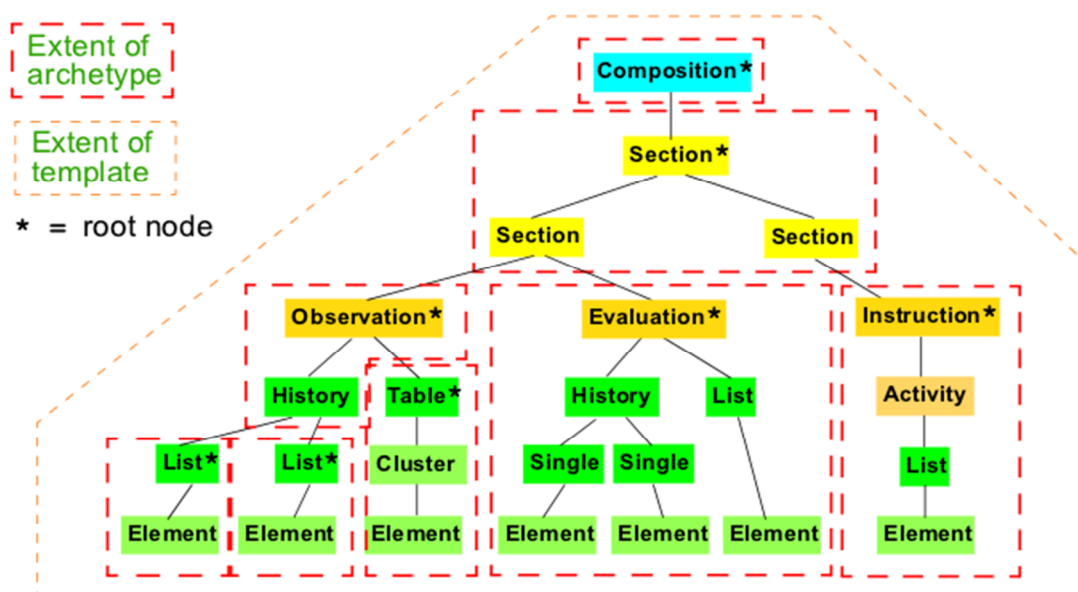


Figura 14 - Elementos de uma Composition

Fonte: Beale (2008)

A hierarquia estendida até o tipo de dado mais simples, ELEMENT, que permite representar diversos tipos de dados como booleanos, datas, uri's, duração de tempo etc., pode ser vista na Figura 15.

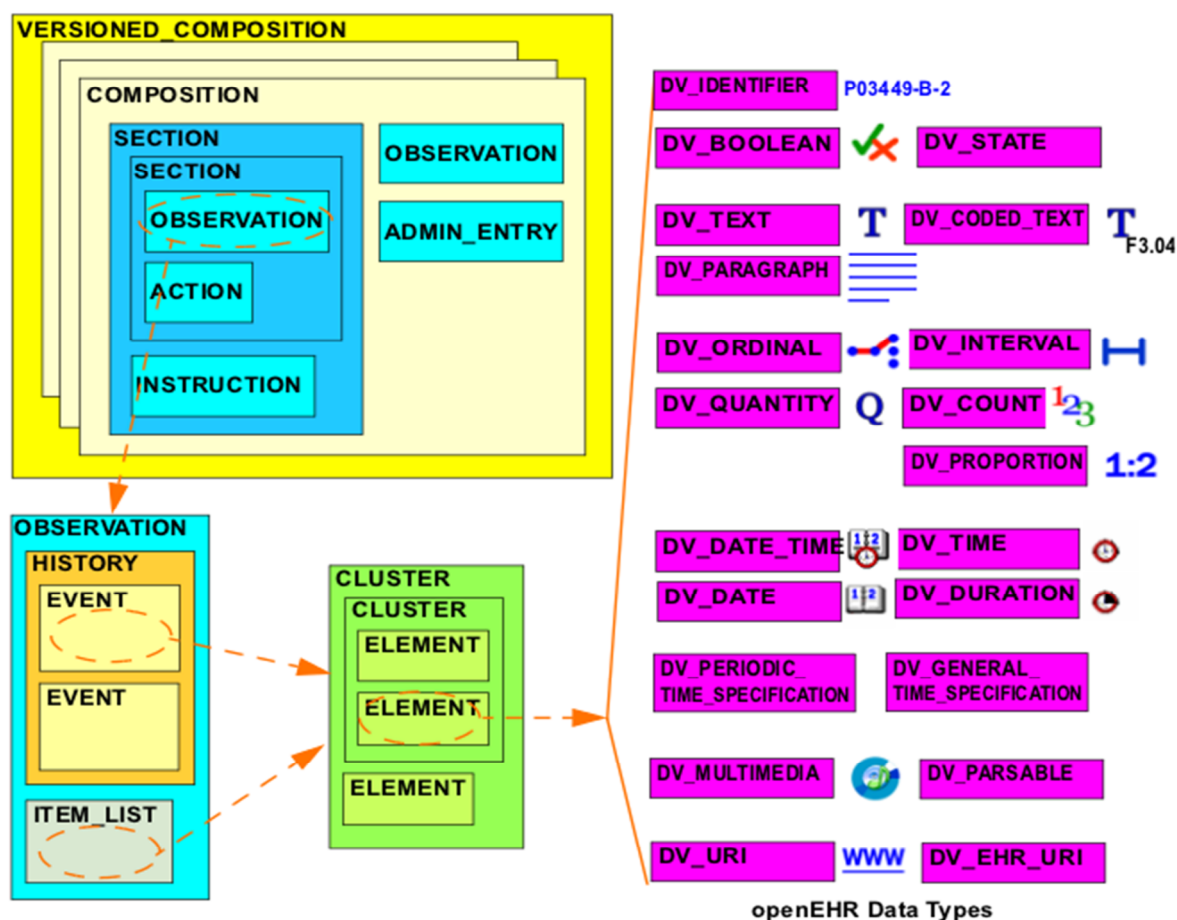


Figura 15 - Hierarquia de Uma Composition  
Fonte: Beale (2008)

O **modelo de arquétipos**, portanto, pode ser visto como **uma representação de metadados** desenvolvida para organizar e padronizar dados de domínios de conhecimento. Através dos arquétipos os conceitos clínicos são capturados estruturadamente **fora** do *software*. Os tipos de arquétipos suportam o registro das atividades clínicas mais comuns, incluindo observações, avaliações, instruções e ações conforme já visto (Leslie, 2008).

Ao se desenvolver o RES como um sistema baseado em conhecimento, construído segundo a abordagem em dois níveis, obtém-se um aplicativo robusto, de mais fácil manutenção e modificação, cujo acréscimo de conhecimentos é feito via arquétipos, que são a expressão computável de um conceito em nível de domínio na forma de declarações de restrições estruturadas, baseada em algum modelo de informação de referência (ABNT, 2008).

O modelo de arquétipos, conforme proposto pela fundação OpenEHR, permite

capturar de modo estruturado os conceitos referentes aos conhecimentos e atividades clínicas como observações, avaliações, instruções e ações; são representações de meta-dados que auxiliam na organização e padronização dos dados referentes a um dado domínio de conhecimento (Beale, 2007). Arquétipos podem ser descritos como um modelo formal e, ao mesmo tempo, reutilizável de um conceito pertencente a um dado domínio que, uma vez representado por um arquétipo pode vir a ser novamente utilizado em vários cenários que exijam sua aplicação.

Sob o ponto de vista da **interoperabilidade semântica**, o modelo de arquétipos da fundação OpenEHR pode ser entendido como especificações que buscam garantir a semântica das informações trocadas entre os diferentes sistemas de RES, permitindo a incorporação de conhecimento ao sistema diretamente no modelo de domínio, tornando viável a interoperabilidade semântica na troca de informações entre os sistemas de registro eletrônico de saúde, tornando-se peça chave no cenário destacado por Massad (2003) onde “[...]a tendência na informática em saúde para a construção do prontuário eletrônico é cada vez mais uma realidade”.

De acordo com Beale (2007), o projeto de especificação do OpenEHR consiste do Modelo de Referência (RM), o Modelo de Serviços (SM) e o Modelo de Arquétipos (AR). Os dois primeiros correspondem à camada de informação/computação respectivamente, o último formaliza a ponte entre o modelo de informação e o modelo de conhecimento. O tipo mais elementar de distinção em qualquer modelo de sistema é o ontológico, isto é, no nível de abstração da descrição do mundo real, pois todos os modelos carregam algum tipo de conteúdo semântico. **O modelo de conhecimento, portanto, posiciona o sistema no nível ontológico, isto é, no nível de abstração da descrição do mundo real, pois todos os modelos carregam algum tipo de conteúdo semântico.**

A Figura 16 ilustra a relação entre os modelos de informação e de conhecimento, onde o modelo de informação genérico anteriormente visto sofre restrições, via expressão através da linguagem ADL (*Archetype Definition Language*) ou XML (*Extensible Markup Language*), para que seja possível expressar conhecimento clínico específico através do modelo de arquétipos, que podem ser serializados e disponibilizados em repositórios para reuso em diferentes projetos de RES.

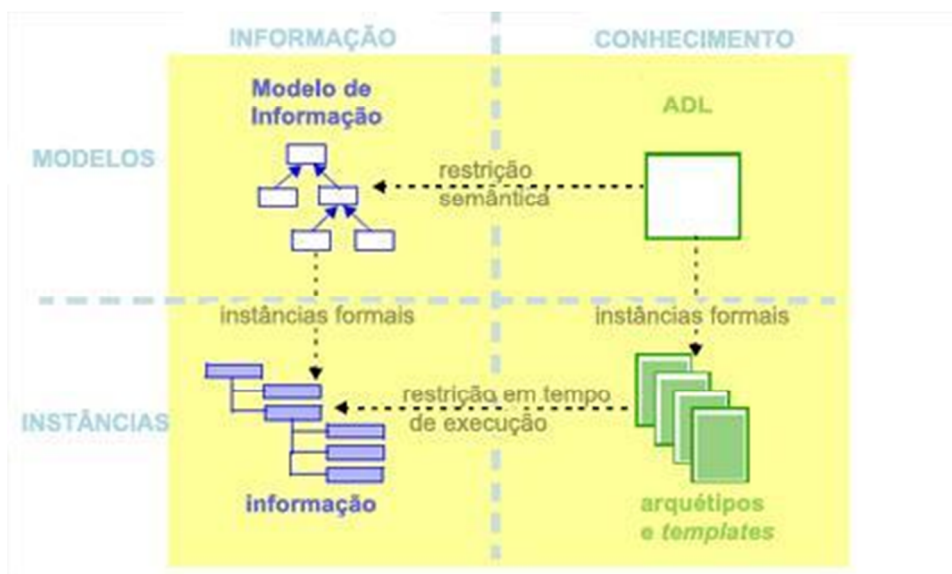


Figura 16 - Relação Entre os Modelos de Informação e de Conhecimento  
 Fonte: Traduzido de Beale (2008)

A maior expectativa é a possibilidade de **reuso** das estruturas de registro de informação, em geral complexas. Sendo assim, a criação de arquétipos e *templates* – forma de apresentação dos dados clínicos para os usuários – torna-se uma tarefa um pouco mais direcionada pelo corpo médico, especialistas do domínio. Uma vez criados os repositórios de arquétipos, estes podem ser utilizados pelos especialistas em tecnologias da informação para a criação dos programas de RES (Figura 17) respeitando a divisão proposta pelo modelo dual.

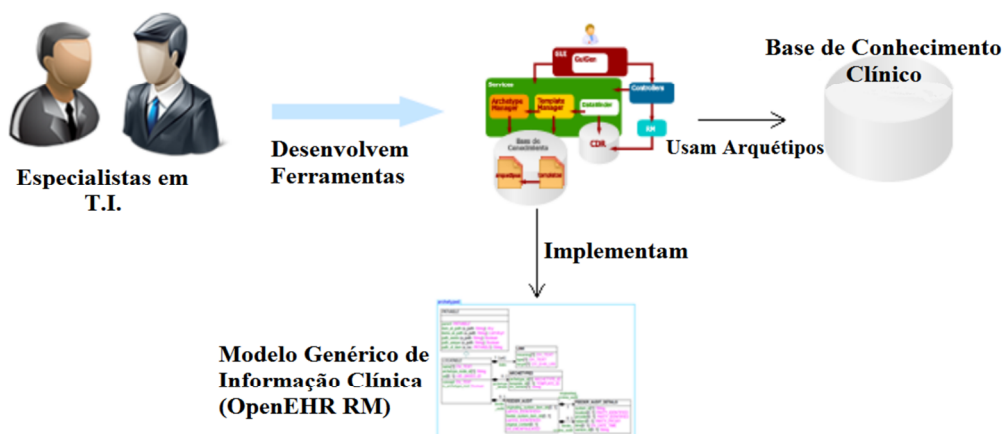


Figura 17 - Construção de Aplicações por Especialistas em T.I.  
 Fonte: Traduzido de Gutiérrez e Carrasco (2013)

**Além de fomentar o maior reuso do conhecimento, a utilização de arquétipos**

pode ser vista como uma solução possível para a heterogeneidade das informações em saúde. Uma vez que promovem a separação entre processos e dados, tornam os sistemas mais flexíveis e interoperáveis. Sistemas de RES baseados no modelo de arquétipos podem ser constantemente atualizados sob a supervisão mais direta de equipes médicas, até mesmo sem gerar interrupções no sistema.

### 5.3 Arquétipos e *Templates*: A Componentização no Padrão OpenEHR

Ao separar claramente os três modelos, a arquitetura OpenEHR permite que cada um destes modelos tenha uma estrutura bem definida, um alcance limitado e interfaces claras. Isso aumenta a componentização e melhora a capacidade de manutenção e adaptação do sistema. Conseqüentemente, os arquétipos podem ser também, considerados como mantenedores da semântica das informações passíveis de uso nas trocas de mensagens entre os sistemas de RES.

Grupos de arquétipos são combinados em *templates* com o objetivo de capturar o conjunto de dados correspondente a uma tarefa clínica particular. Conforme salientado anteriormente, devido ao seu caráter componentizado, os arquétipos podem ser utilizados na definição de diversos *templates* como ilustrado pela Figura 18.

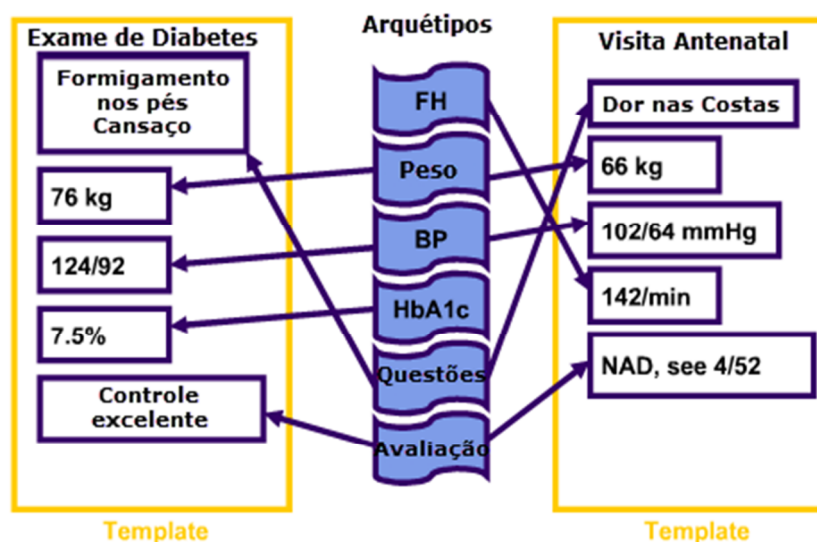


Figura 18 - Utilização de Arquétipos em *Templates*  
 Fonte: Traduzido de Cook (2010)

Quando médicos olham para os *templates*, a informação contida dentro deles faz

sentido diretamente e não requer treinamento intensivo para que eles sejam capazes de criar novos *templates* para seus próprios propósitos.

Os *templates* podem ser utilizados para construir formulários genéricos que representam o *layout* aproximado do RES em uma forma prática, e estes podem ser utilizados por fornecedores, contribuindo com o desenvolvimento de interfaces para usuários.

Na Figura 19 em ordem ascendente, é mostrado o caminho percorrido partindo do modelo de referência, explicado anteriormente, passando pela definição dos arquétipos, a seguir pela definição de *templates* que servirão de base para se criar os formulários da interface de aplicativos de RES.

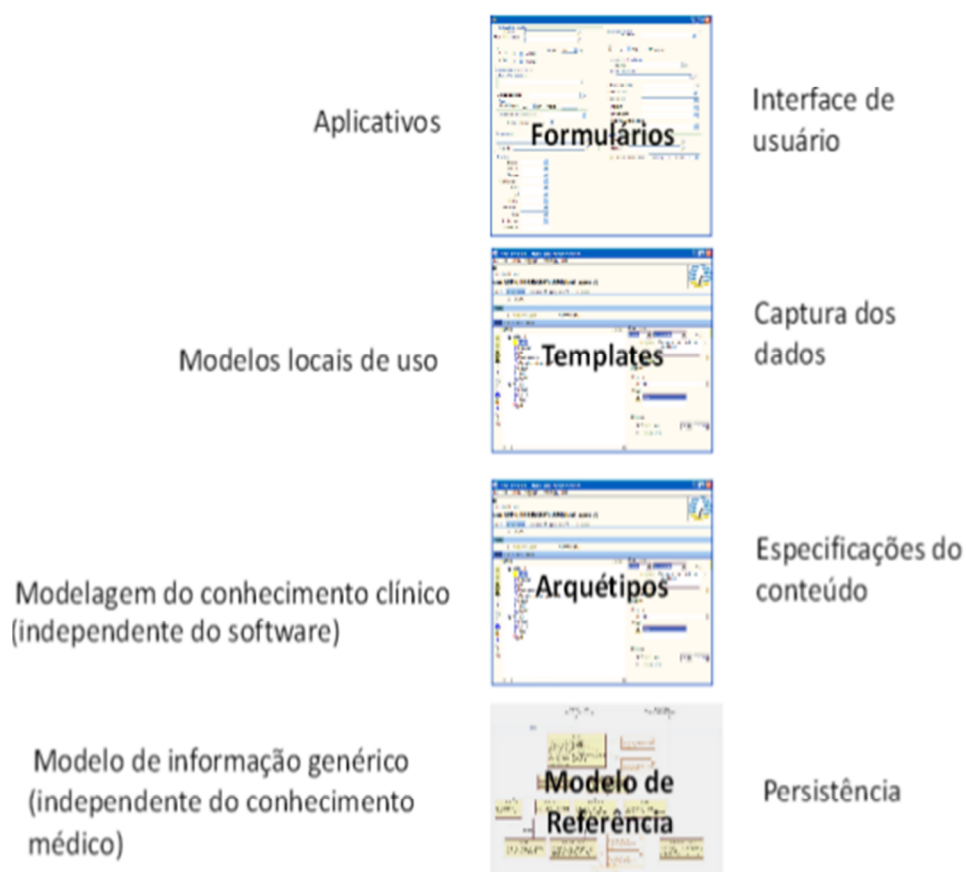


Figura 19 - Sequência de Artefatos Necessários para se Construir uma Interface de RES  
Fonte: Elaborado pelo Autor

Além disso, **arquétipos e *templates* podem ser combinados com terminologias** ou subconjuntos de terminologias contextualmente apropriadas que suportem a seleção de termos pelo provedor de atendimento médico no ponto de entrada de dados, conforme esquematizado na Figura 20:



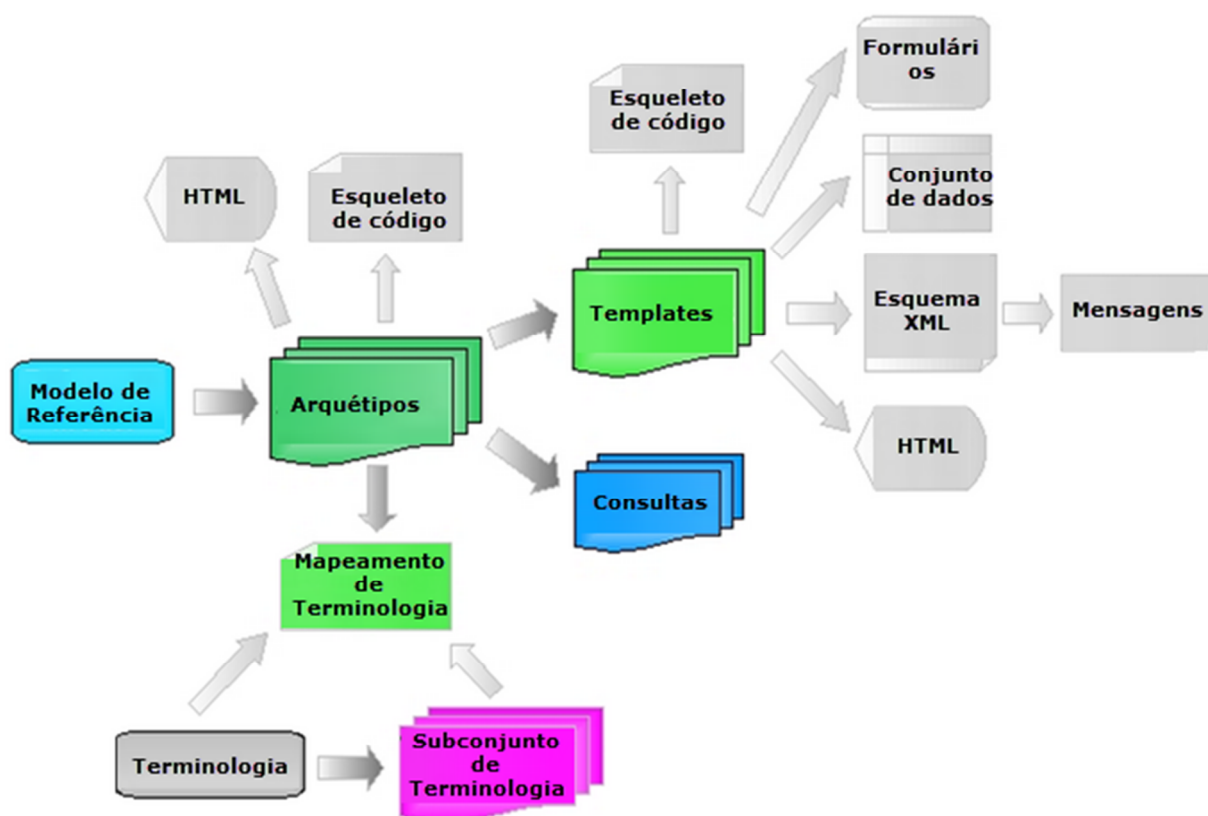


Figura 20 - Relação Entre os Artefatos Formais do Padrão OpenEHR  
 Fonte: Traduzido de Chen (2008)  
 Acesso em novembro de 2010

#### 5.4 Tentativa de Implementação de Arquétipos OpenEHR em Plone/Archetypes

Visando verificar a possibilidade da construção de arquétipos num sistema gestor de conteúdos/CMS com o objetivo de eliminar etapas custosas da construção padrão de um RES, e buscando agregar possibilidades de expressão dos arquétipos através de um *framework* para geração de conteúdo, escolheu-se o par *Plone/Archetypes* para tal tentativa.

Após, aproximadamente, três meses de estudos e tentativas de implementação dos arquétipos em *Plone/Archetypes*, concluiu-se pela sua impossibilidade, numa relação aceitável de custo-benefício. Tal resultado teve como consequência o desvio para uma nova etapa do questionamento quanto à possibilidade de expressar arquétipos num CMS. Solucionar a questão era necessário, pois caso a resposta fosse negativa, a pretensão inicial, da qual se partiu por hipótese, de utilizar CMS's como *framework* para a implementação de sistemas de

RES teria de ser revista ou abandonada. O capítulo 7 apresenta o resultado desse esforço elucidativo.

## 6 EXPRESSÃO DO PADRÃO OPENEHR EM PYTHON

Uma vez discutidos, no Capítulo 4, os modelos de referência e conhecimento do padrão OpenEHR e como a ontologia C.I.R. fundamenta a especificação da classe *ENTRY*, neste capítulo, as classes dos modelos de referência são apresentadas, bem como as principais características e detalhes da semântica a elas associada. O código Python, correspondente às classes encontra-se hospedado num repositório *online* de acesso público<sup>25</sup>.

### 6.1 Os Modelos de Informação: Base para Representar o Conhecimento

Conforme Beale e Heard (2008), a abordagem adotada pelo padrão OpenEHR para modelar informações e conhecimentos clínicos é baseada em princípios de *design* que visam separar os modelos da arquitetura OpenEHR. Conforme salientado no Capítulo 5 (p. 42-44), a capacidade de manutenção, a flexibilidade e a escalabilidade dos sistemas desenvolvidos sob essa ótica são melhorados.

Visando a realização de tal cenário, o padrão OpenEHR utiliza a separação em níveis. Tal escolha se justifica, uma vez que modelos sempre carregarão conteúdos semânticos, que podem ser utilizados num processo de categorização. Como exemplos de tais situações, Beale e Heard (2008) citam a terminologia SNOMED-CT que procura descrever infecções causadas por certos tipos de bactérias, partes do corpo humano, assim como, sintomas das mais diversas patologias. Um modelo de informação pode ter de especificar um tipo lógico referente a quantidades, bem como informações coletadas num exame pré-natal, e estes tipos de informação possuem diferenças qualitativas, as quais necessitam ser categorizadas e separadas dentro do “ecossistema” que encampa os modelos de informação. O esquema dessa separação pode ser visto na Figura 21, que distingue ontologias de informação e ontologias da realidade. Elas possuem propósitos diferentes, sendo que a primeira **define modelos para os conteúdos de informação**, enquanto a segunda visa **descrever e classificar os fenômenos do “mundo real”**:

---

<sup>25</sup> Endereço eletrônico do repositório: <[https://github.com/chrispess/Doutorado\\_ECI/tree/master/openehr](https://github.com/chrispess/Doutorado_ECI/tree/master/openehr)>.

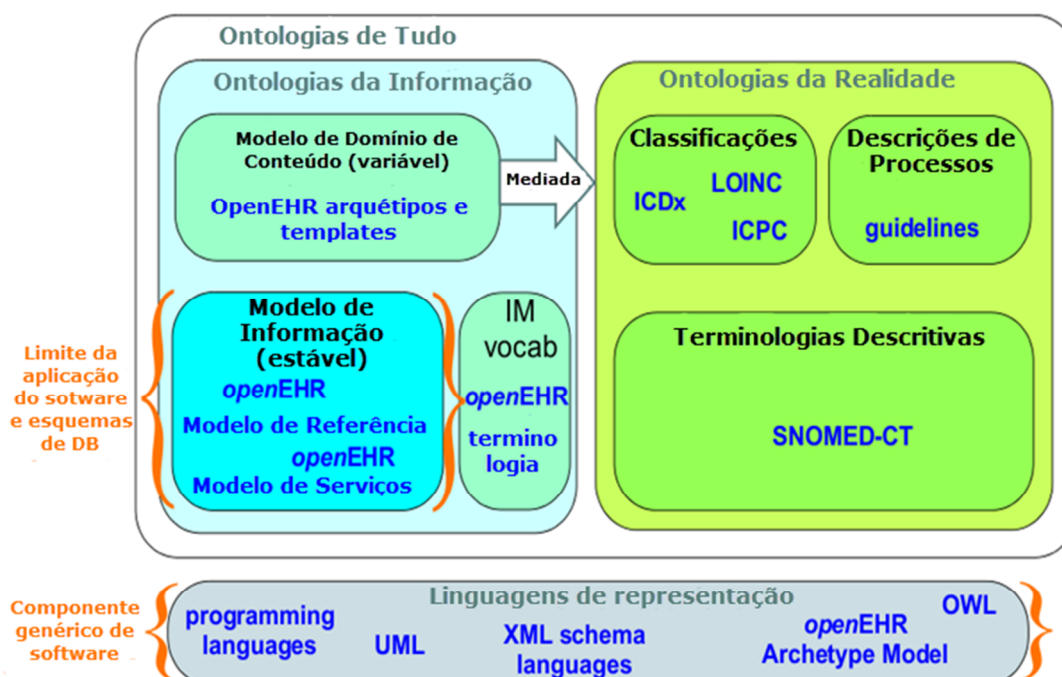


Figura 21 - O Esquema Ontológico do padrão OpenEHR  
 Fonte: Traduzido Beale e Heard (2008)

Um segundo nível de separação, pode ser visto dentro do nível da “ontologia da informação” (lado esquerdo da figura), entre **modelo de informação** e **modelo de domínio (de conteúdo)**. O critério utilizado para essa separação é a **invariância**. No primeiro há invariância dos tipos e estruturas de dados, id’s, listas, etc.; diferente do último, que corresponde ao nível dos conteúdos de domínio, ou estruturas de informação que variam conforme descrevem fenômenos do mundo real, como temperatura corporal, pressão arterial, infecção por micróbios, entre outros.

Tal separação apresenta-se como de suma importância. Beale e Heard ressaltam que:

Esta separação não é geralmente bem entendida, e historicamente, uma grande porção da semântica do domínio tem sido projetada no *software* e nas bases de dados, levando a sistemas de manutenção inviável (BEALE; HEARD, 2008, p.15, tradução nossa).

Em sua busca por separar de modo claro os modelos de informação, de domínio e as terminologias, a arquitetura do padrão OpenEHR almeja definir de maneira precisa o escopo de cada um, bem como suas interfaces gerando, como consequência, sistemas com mais alta capacidade de manutenção e adaptação.

Uma consequência de tal abordagem é a adoção, pelo padrão, da modelagem em dois

níveis (*cf.* Capítulo 5), onde um modelo de referência, estável e único a ser implementado em *software*, constitui o primeiro nível de informação, enquanto as definições dos conteúdos clínicos, expressas na forma de arquétipos e *templates* constituem o segundo nível. Sistemas projetados assim são mais adaptáveis do que aqueles concebidos em camada única.

Baseado em tais premissas, a fundação *OpenEHR* disponibiliza um conjunto de especificações que consiste em Modelo de Referência (RM), Modelo de Serviços (SM) e Modelo de Arquétipos (AM). O modelo de referência corresponde ao ponto de vista das informações que serão serializadas e processadas pelo sistema. O modelo de serviços corresponde ao ponto de vista computacional que busca descrever o sistema como um conjunto de objetos e interfaces que interagem, possibilitando que o sistema se torne compartilhável. O modelo de arquétipos formaliza a ponte entre os modelo de referência e os Arquétipos.

A correspondência entre as especificações e a plataforma computacional do padrão OpenEHR pode ser vista em seus elementos constituintes e relações na Figura 22:

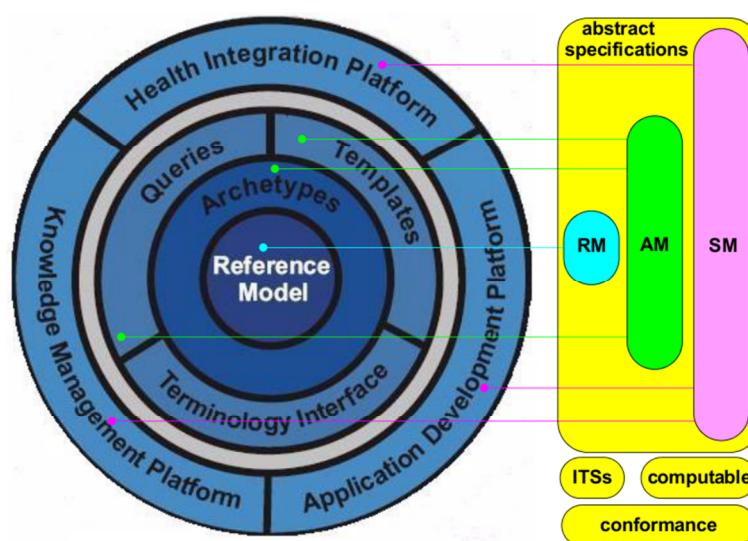


Figura 22 - O Projeto da Especificação OpenEHR  
Fonte: Beale e Heard (2008)

## 6.2 A Semântica dos Arquétipos Expressa no Modelo de Referência

Em tal arquitetura, o modelo de referência é responsável pela semântica da informação que será trabalhada via restrições na camada de conhecimento, conforme salientado em Beale:

A estrutura e a **semântica da informação são definidas por um modelo de referência**; **itens de informação são instâncias deste modelo**, conforme as relações classe/objeto usuais na orientação à objetos. No lado Conhecimento, são arquétipos, *i.e.* definições no nível de conhecimento, cujo trabalho é restringir no tempo de execução a estrutura e a semântica da informação (BEALE, 2002, p.14, grifo nosso, tradução nossa).

A Figura 23 esquematiza tal relação de restrição sobre as classes do modelo de referência para gerar instâncias no nível do conhecimento.

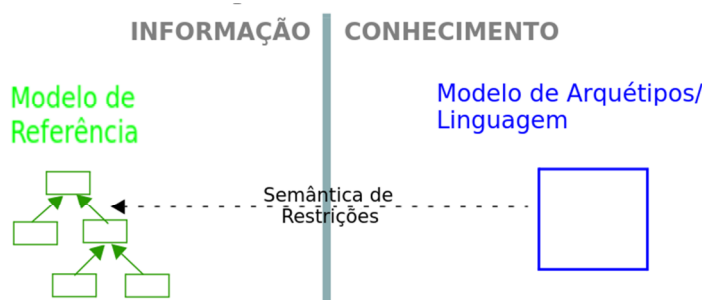


Figura 23 - Modelo de Arquétipos: Meta Arquitetura  
Fonte: Adaptado e Traduzido de Beale (2002)

Logo, para que se expresse os conceitos referentes ao nível de conhecimento, deve-se ter à mão a estrutura de blocos básicos necessários para sua construção, ou seja, as diversas classes definidas na especificação dos modelos de informação do modelo de referência. Pois, conforme Martínez-Costa *et al.*:

O **modelo de referência** representa as características globais das anotações dos registros de saúde, como elas são agregadas e o contexto de informação requerido. **Este modelo, define o conjunto de classes que formam os blocos genéricos de construção do registro eletrônico de saúde e contém as características não voláteis do registro eletrônico de saúde. [...] o sistema de informação é baseado no modelo de referência, e os extratos válidos do registro de saúde são instâncias desse modelo de referência** (Martínez-Costa *et al.*, 2009, p.151, grifo nosso, tradução nossa).

Da mesma forma, conforme ressaltado por Chen e Klein (2007, p.58, tradução nossa, grifo nosso): “[...] de acordo com a abordagem via modelo dual adotada pelo padrão OpenEHR, **o núcleo de um sistema de registro eletrônico de saúde pode ser construído baseado nos componentes do modelo de referência**”. Ou, conforme afirmam Beale e Heard (2008), **estes modelos constituem a referência primária para toda semântica OpenEHR**.

Portanto, do visto anteriormente, retira-se que, para se obter uma resposta quanto à questão de se é possível expressar informações clínicas de acordo com a semântica definida pelo padrão OpenEHR na plataforma Python, torna-se necessária a implementação dos

modelos de informação nesta linguagem. Uma vez obtido um resultado positivo quanto a esse esforço, pode-se afirmar, de modo categórico, a possibilidade de se expressar informação clínica conforme a semântica OpenEHR nesta plataforma.

A obtenção de tal resultado abre a possibilidade para os desenvolvedores de sistemas de informação escreverem códigos nesta linguagem, e expressar um modelo clínico capaz de criar e utilizar dados clínicos em acordo com tal semântica. Tal resultado torna o formalismo de arquétipos mais acessível para projetos acadêmicos que desejem expressar, utilizando a plataforma Python, informação clínica sob a semântica da especificação do padrão OpenEHR.

No texto a seguir, neste capítulo, procurar-se-á realizar uma exposição/análise da semântica dos modelos de informação que compõe o modelo de referência do padrão OpenEHR. Tal compreensão se faz necessária se se objetiva a construção de aplicativos que utilizem tais modelos de informação.

Uma vez que o objetivo principal desta pesquisa inquirir sobre a possibilidade de expressar, na plataforma Python, informação clínica conforme a semântica definida na especificação OpenEHR, alguns modelos de informação não diretamente relacionados a tal objetivo tiveram sua implementação postergada a trabalhos futuros. Portanto, por este critério, modelos de informação que tratam de demografia, versionamento, privacidade, etc., não estando diretamente relacionados aos modelos de informação clínica derivados da ontologia CIR (ver Capítulo 5) não tiveram sua implementação considerada no âmbito deste trabalho.

O código em Python para cada modelo de informação pode ser acessado no repositório *online* indicado na página 63. Optou-se por tal forma de apresentação visando manter o texto do capítulo dentro de uma dimensão razoável. Mantendo o propósito de clareza textual, no desenrolar do texto, procurar-se-á apresentar os modelos de informação de maneira concisa e objetiva. Maiores detalhes podem ser consultados na especificação, cujos volumes estão no sítio do padrão OpenEHR<sup>26</sup>.

## **6.3 Os Modelos de Informação: Características Semânticas e Implementação**

### **6.3.1 Modelo de Informação do Registro Eletrônico de Saúde (RES)**

Nesta seção, apresenta-se a semântica das classes do modelo de informação,

---

<sup>26</sup>Especificação OpenEHR: <<http://www-test.openehr.org/programs/specification/releases/currentbaseline>>

responsável pela arquitetura de informação de um RES (Beale *et al.*, 2008d). Este modelo define as estruturas de nível mais alto de um RES como, por exemplo, *composition*, *navigation* e *entry*. O detalhamento dos demais modelos de informação, suas classes e a semântica que definem, encontra-se no Anexo B.

### 6.3.1.1 Pacote *Composition*

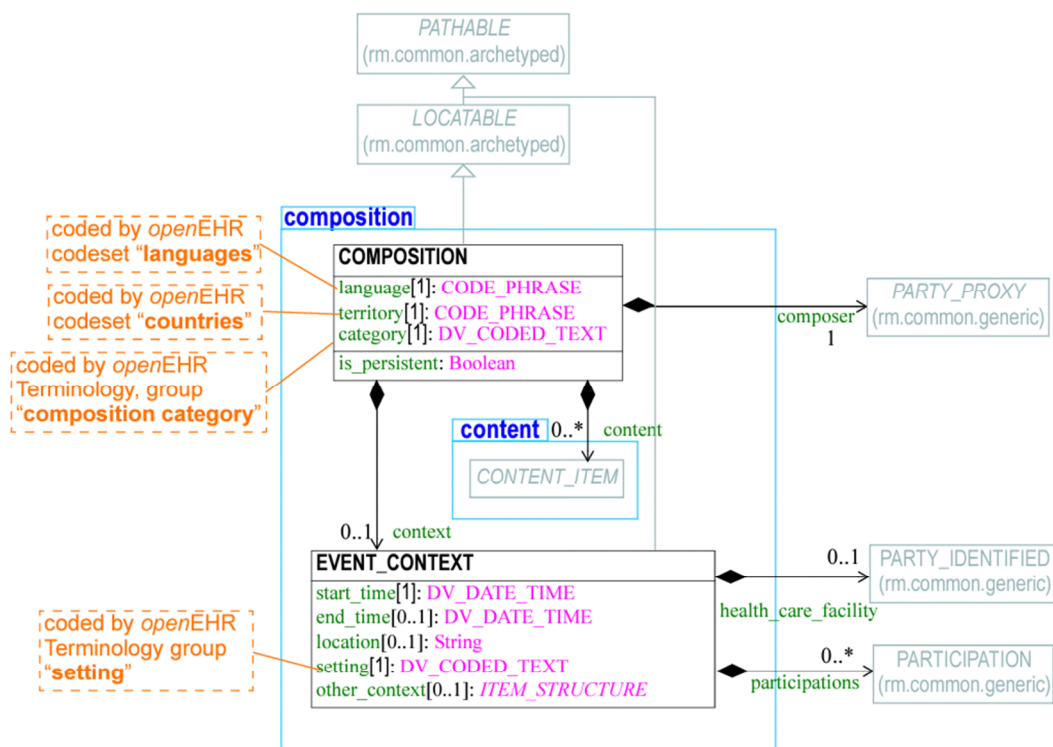


Figura 24 - O Pacote OpenEHR Composition  
Fonte: Beale *et al.*(2008d)

A classe *Composition* é o “container” de dados primário e a raiz dos conteúdos clínicos de um registro eletrônico de saúde baseado no padrão OpenEHR, é o agregador de dados ou documentos num sistema orientado à documentos. Uma instância da classe COMPOSITION permite recuperar informações em seu contexto, o próprio contexto, bem como os atributos de quem o criou (o *composer*), pois o modelo proposto pelo padrão fundamenta-se na análise sistemática de contextos do “mundo real” que são mapeados para os níveis apropriados do modelo de informação conforme nos mostra a Figura 25.

Visando uma melhor compreensão do esquema da Figura 25, deve-se lembrar que um **documento clínico** é uma composição discreta e eletrônica sobre um paciente identificado,



com a intenção primária de ser lida por seres humanos, como um relatório ou carta, ou seja, um documento em papel. Documentos clínicos podem incluir em suas **declarações clínicas**, arquivos pdf, links para páginas web, imagens, áudio e vídeo.

Uma **declaração clínica** é uma instanciação de um tipo de dado clínico. Pode ser pensada como componente molecular que possui existência independente, como um diagnóstico, um procedimento ou o resultado de um teste. Uma leitura da pressão sanguínea de um paciente é um exemplo de tipo de dado clínico. Declarações clínicas são utilizadas para estruturas dados de modo apropriado ao processamento computacional, como tabelas com resultados de laboratório, procedimentos, medicações, resultados de testes e apontamentos. Em termos do esquema da Figura 25, **declarações clínicas** possuem um nível de granularidade similar ao nível de granularidade de um arquétipo *Entry* do padrão OpenEHR (descrito abaixo, neste capítulo).

No lado inferior esquerdo pode-se ver o contexto das sessões de entrada das informações clínicas ou declarações clínicas geradas num encontro (ou evento) de cuidado em saúde mapeado através das estruturas informacionais do ambiente de registros clínicos proposto pelo padrão. Para o modelo proposto pelo padrão, uma declaração clínica é uma unidade de informação mínima e indivisível passível de registro, carregando consigo estruturas temporais e espaciais chamadas `DATA_VALUES`.

O sistema de registro eletrônico em saúde (lado direito da Figura 25) consiste em distintos itens chamados *Compositions* adicionados ao longo do tempo e organizados em pastas (*folders*). Cada *composition* consistindo de entradas (*entries*) organizadas em seções (*sections*) dentro da *composition*.

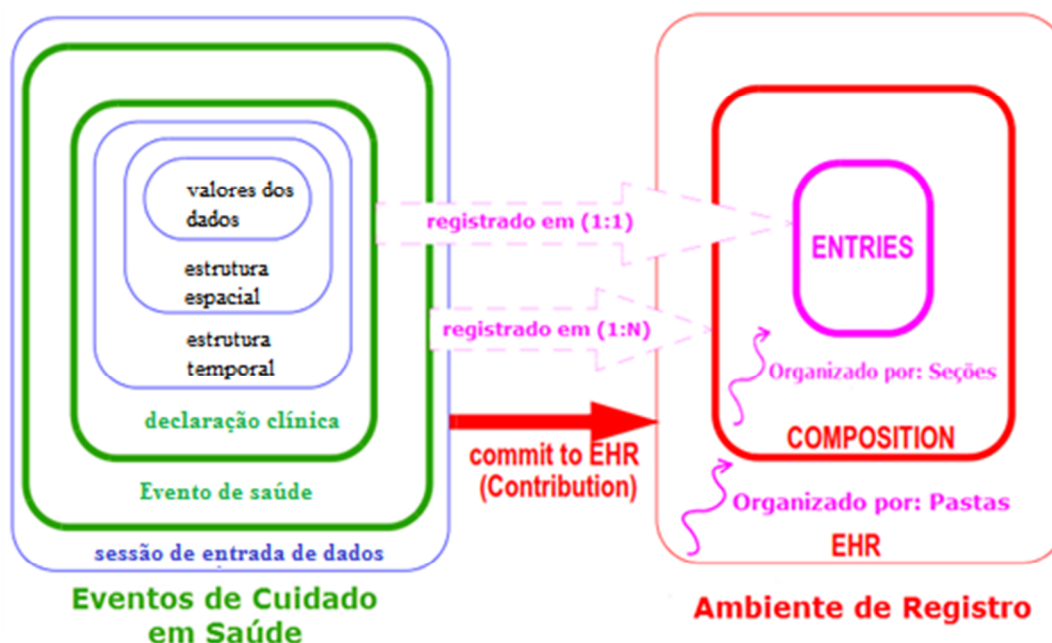


Figura 25 - O Mapeamento das Informações Oriundas de Eventos em Saúde para o Registro OpenEHR

Fonte: Beale *et al.* (2008d)

### 6.3.1.2 Pacote *Content*

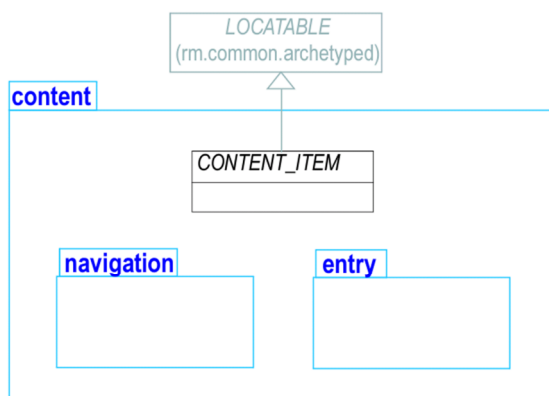


Figura 26 - O Pacote OpenEHR *Content*

Fonte: Beale *et al.*(2008d)

No pacote *content* é definida a classe `CONTENT_ITEM`, ancestral de todos os tipos de conteúdo, bem como os pacotes *navigation* e *entry*, contendo as classes definidoras de `SECTION`, `ENTRY` e todos os tipos relacionados (Figura 26).

### Composition: O atributo *Content*

Os dados numa *composition* são registrados via atributo *content*, havendo quatro modos possíveis de estruturação dos dados num atributo *content*, que são:

- Pode ser vazio, havendo dois casos em que uma *composition* vazia faria sentido:
  - A *Composition* está em estado de “rascunho”.
  - Quando o sistema registra um fato referente a um evento mas sem interesse em seus detalhes, como ao registrar a visita de um médico sem adicionar nenhuma informação.
- Pode conter uma ou mais SECTION’s.
- Pode conter uma ou mais ENTRY’s inseridas diretamente, sem estarem organizadas por SECTION’s.
- Pode ser uma combinação dos três itens anteriores.

#### 6.3.1.3 Pacote *Navigation*

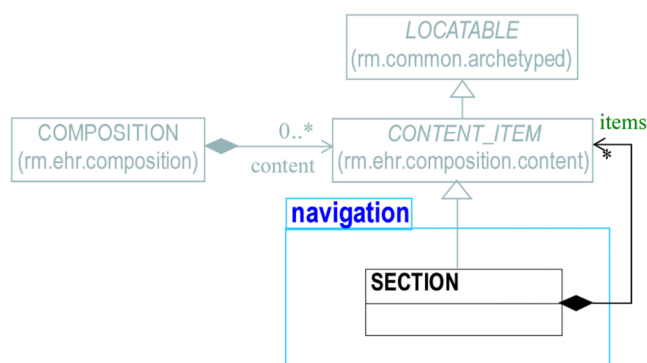


Figura 27 - O Pacote OpenEHR Navigation  
Fonte: Beale *et al* (2008d)

O pacote *navigation* (Figura 27) define uma hierarquia de estruturas que possibilita a organização de conteúdos em diretórios de modo a que todo diretório seja considerado como pertencente a uma árvore de diretórios (instâncias da classe SECTION). Estruturas do tipo SECTION são facultativas e podem ser removidas ou ignoradas, sendo utilizadas para agrupar, pelo tipo de conteúdo, instâncias da classe *Entry* em categorias como: “histórico familiar”, “problemas”, “observações”.

### 6.3.1.4 Pacote *Entry*

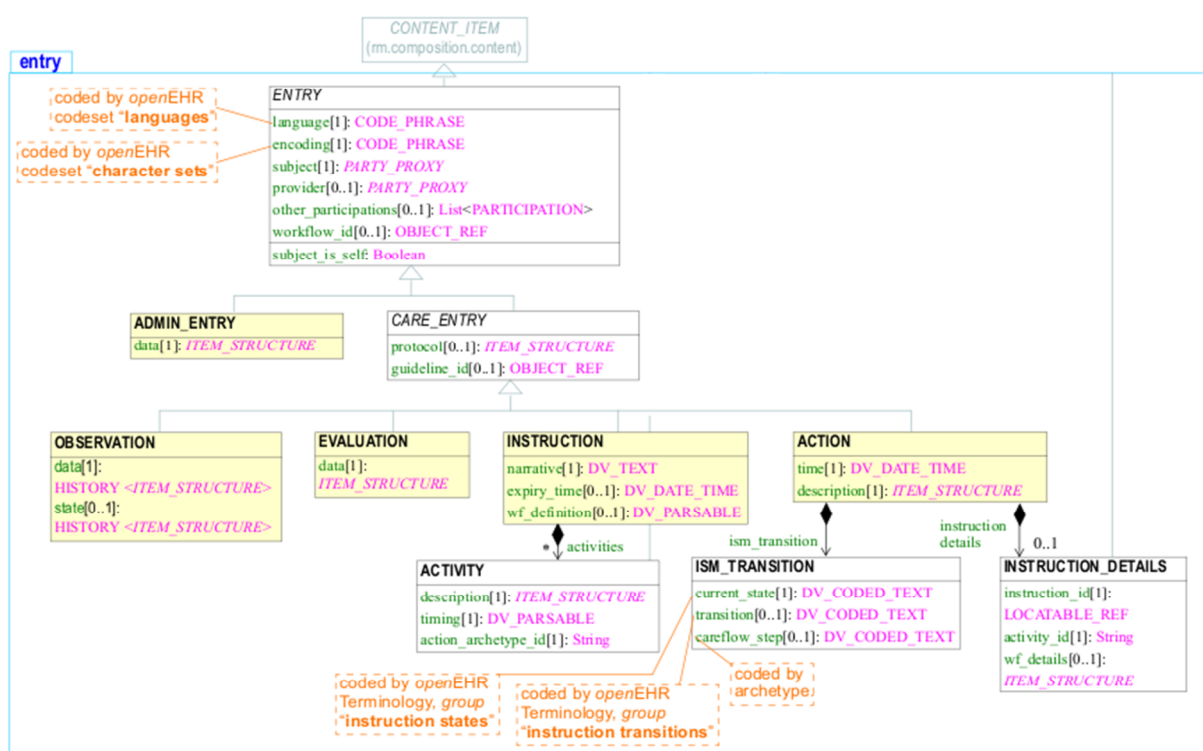


Figura 28 - O Pacote OpenEHR Entry  
Fonte: Beale *et al.*(2008d)

Toda informação criada num registro eletrônico de saúde baseado no padrão OpenEHR é expresso como uma instância de uma classe do pacote *Entry* (Figura 28). Uma instância da classe *Entry* corresponde a uma declaração clínica simples, podendo ser uma frase narrativa curta ou uma grande entrada de dados como, por exemplo, um resultado de exame microbiológico, um exame psiquiátrico ou uma prescrição mais complexa. Em termos de conteúdo clínico, a classe *entry* é a que possui maior relevância para o modelo de informação proposto pelo OpenEHR, pois define a semântica de toda a informação a ser registrada.

### Os Tipos Padrão de Dados Clínicos

Os tipos OpenEHR definidos como possíveis entradas da classe *entry*, foram pensados para, normalmente, representar os tipos de informação clínica mais comumente utilizados. Conforme o modelo de informação OpenEHR há dois tipos possíveis de compositions: *persistent* e *event*. O persistente corresponde às informações que caracterizam o paciente por

um longo prazo e é mantido pelo(s) clínico(s), podendo ser denominado de modelo do paciente. Como exemplos de entradas para *compositions* persistentes podemos citar (os subtipos como observações, avaliações, ações, etc. que serão vistos adiante):

- Informações Básicas como sexo, altura, peso, gravidez, etc; registradas como observações ou *admin\_entries*.
- Lista de problemas mantida como uma ou mais avaliações mantidas por clínicos como resultado de observações feitas no registro.
- Lista de medicamentos: Derivada de instruções e ações no registro.
- Precauções terapêuticas (alergias e alertas): Registradas como avaliações de vários tipos.
- Preferências do paciente: Registradas como avaliações caso este tenha restrições contra certos tipos de drogas ou tratamentos.
- Consentimentos do pacientes: Registrados como instâncias da classe *Admin*.
- Histórico familiar: registrado como observações.
- Estilo de vida: registrado como observação cobrindo aspectos como fazer uso de álcool/tabaco, drogas, etc.
- Registro de vacinações: vacinação é um tipo de instrução (caso a vacinação esteja sendo feita é registrada como *action* no registro).

### A Classe Entry

Toda entrada possui um certo número de atributos em comum, como linguagem e codificação indicando como todos os textos das entradas podem ser interpretados linguisticamente e qual conjunto de caracteres deve ser utilizado.

### Care\_entry e Admin\_entry

Uma divisão básica das entradas é a divisão entre informação clínica e não clínica (veja ontologia CIR no Capítulo 4). A classe *CARE\_ENTRY* é a classe base para todas as classes que expressam informações referentes à qualquer atividade clínica no processo de cuidado ao paciente, enquanto a classe *ADMIN\_ENTRY* é utilizada para capturar informação administrativa.

Informação administrativa possui as seguintes características:

- É criado por não clínicos ou clínicos agindo num contexto administrativo.
- Expressa detalhes da coordenação do processo clínico, registrando informações como: admissão, apontamentos, desembarque/desligamento (permitindo registrar que um paciente foi enviado para casa ou transferido para outra instituição), faturamentos.
- A remoção de informações administrativas do registro eletrônico de saúde não compromete sua integridade clínica.

### Observações

São instancias da classe OBSERVATION responsáveis pelo registro de observações de quaisquer fenômenos ou estados referentes ao paciente, como resultados de patologia, leituras de pressão sanguínea, histórico familiar e circunstâncias sociais conforme narradas ao clínico pelo paciente, respostas do paciente a questões formuladas pelo médico durante o exame clínico bem como respostas a questionários de avaliações psicológicas. Observações distinguem-se de ações pelo fato de ações serem intervenções, enquanto observações apenas registram relativas à situação do paciente e não o que é feito a ele. As informações significantes numa observação são expressas em termos de *data*, *state* e *protocol* (os termos originais foram mantidos devido ao seu uso mesmo nos arquétipos traduzidos).

- *Data*: O dado que está sendo registrado no momento, expresso na forma de um histórico (veja classe *History*) de eventos, cada um consistindo de estruturas de dados complexas como listas, tabelas, árvores ou o valor simples (instância da classe *Single*). como exemplos pode-se citar pressão sanguínea, ritmo cardíaco, traço ECG, etc.
- *State*: Qualquer informação particular sobre o estado do paciente, cujas entradas de dados estão sendo registradas, necessárias para a correta interpretação dos dados como fatos de uma paciente estar grávida ou submetendo-se à quimioterapia, nível de esforço em exercícios (repouso, pós-maratona,...), posição (deitado, de pé), etc. A forma do atributo é a mesma de *data*, um histórico de eventos na forma de *item\_structures*.
- *Protocol*: Detalhes de como a observação foi realizada, incluindo informações sobre instrumentos, métodos, procedimentos clínicos particulares e outros métodos

observacionais. Tais informações podem ser sempre omitidas de uma interface do usuário, não possuindo importância na interpretação dos dados registrados.

### **Avaliações**

Pela ontologia CIR (veja Capítulo 5), uma avaliação pode cobrir conceitos como problemas e diagnósticos, avaliação de risco, cenário, objetivos e recomendações.

O *design* da classe é bastante simplificado. Possui um único atributo (além dos herdados das classes ENTRY e CARE\_ENTRY) denominado data que é uma instância do tipo ITEM\_STRUCTURE, não havendo atributos de tempo associados à emissão da avaliação.

### **Instruções e Ações**

Para o padrão OpenEHR, instruções especificam ações a serem executadas no futuro, diferindo da avaliação pelo maior nível de detalhamento, suficiente para sua execução sem a necessidade de quaisquer decisões posteriores a respeito da sequência de passos a serem seguidos para a sua realização. Podendo, por exemplo ser realizadas por enfermeiros ou pacientes. Quaisquer decisões a serem tomadas durante a realização da instrução devem estar especificadas na própria instrução, como por exemplo, a suspensão da administração de um medicamento devido a reações alérgicas.

#### **6.3.1.5 Tentativa de Implementação de Arquétipos OpenEHR em Plone/Archetypes**

Após a conclusão da primeira sequência de passos visando a implementação dos arquétipos, conforme descrito neste capítulo, obteve-se os elementos necessários para a sua implementação. Ou seja, implementou-se em Python o modelo de referência da especificação que constitui o primeiro nível do modelo dual (Figura 29), proposto no *design* do padrão OpenEHR, como ilustrado abaixo (Sinha *et al*, 2013).

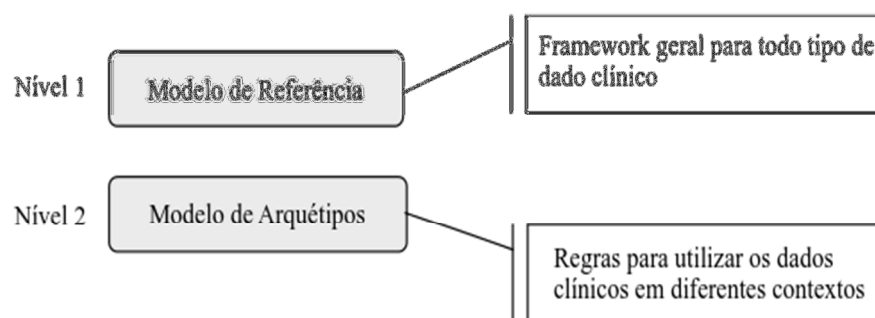


Figura 29 - OpenEHR - Modelagem Dual  
Fonte: Sinha *et al.*(2013d)

O resultado obtido abre possibilidades para a utilização de arquétipos OpenEHR em softwares RES/PEP desenvolvidos sobre a plataforma Python, pois conforme discutido no Capítulo 5, o modelo de referência provê as classes cujas instâncias servem à construção dos arquétipos.



## 7 GESTÃO DA INFORMAÇÃO CLÍNICA DE UM RES OpenEHR VIA CMS.

Neste capítulo, parte-se, inicialmente de uma discussão sobre o custo benefício e problemas que podem surgir ao se seguir uma proposta de implementação de softwares de PEP e RES a partir "do zero", como a proposta da arquitetura para um sistema de cinco camadas ("*5-tier System Architecture*") definida pelo OpenEHR (BEALE; HEARD, 2008, p.71-72). A seguir parte-se do conceito de arquétipos sob a ótica de uma estrutura de metadados ou informação **cercada de metadados**, e como podem ser mais facilmente implementados em *frameworks* voltados a metadados.

Em seguida, argumenta-se como um CMS pode ser visto como um *framework* voltado a metadados; o conceito de modelo de conteúdos e o de conteúdo são apresentados. Em seguida, discute-se o conceito de conteúdo à luz do conceito de arquétipo, buscando as similitudes entre os dois. Obtida esta aproximação conceitual, busca-se o similar para a instância de um arquétipo no domínio dos CMS's e discute-se como o conceito de componente atende a este requisito.

Finalmente, uma ressalva sobre o fato de que, apesar de todo CMS possuir um modelo de conteúdo não implica, necessariamente, em facilidades para a expressão de arquétipos em qualquer CMS, conforme mostrou a tentativa de expressar arquétipos via Plone/*Archetypes*.

O mesmo raciocínio vale ao se mencionar as facilidades de um *framework* de conteúdos como o *dexterity*<sup>27</sup> ou o *entity*<sup>28</sup>. Sem entrar em detalhes técnicos que fugiriam ao escopo do capítulo, objetivou-se ressaltar como um CMS, dotado de *frameworks* desse tipo, apresenta vantagens para a expressão/manipulação (CRUD) de arquétipos, melhorando o custo benefício frente à forma *default* da arquitetura computacional proposta pelo OpenEHR.

### 7.1 Questões Concernentes à Implementação de PEP/RES Conforme a Arquitetura Computacional Proposta no Padrão OpenEHR

Conforme o resultado apresentado no capítulo anterior, através da implementação do modelo de referência, é possível expressar informações clínicas conforme a semântica OpenEHR na linguagem Python. A obtenção de tal resultado abre a possibilidade para

---

<sup>27</sup> Plone Dexterity: <<http://plone.org/products/dexterity>>.

<sup>28</sup> Drupal Entity: <<https://www.drupal.org/project/entity>>.

desenvolvedores de sistemas de informação escreverem códigos nesta linguagem, e expressar um modelo clínico capaz de criar e utilizar dados clínicos.

Ademais, como foi visto anteriormente, de acordo com a abordagem do modelo dual adotada pelo padrão OpenEHR, o modelo de referência define a estrutura e a semântica da informação clínica, enquanto na camada de conhecimento existem os arquétipos, que aplicam restrições às classes do modelo de referência para chegar aos conceitos ou artefatos de conhecimento definidos pelos especialistas clínicos. Tal situação de instanciação é ilustrada na Figura 30, onde o modelo de referência é utilizado para a construção do arquétipo pressão arterial.

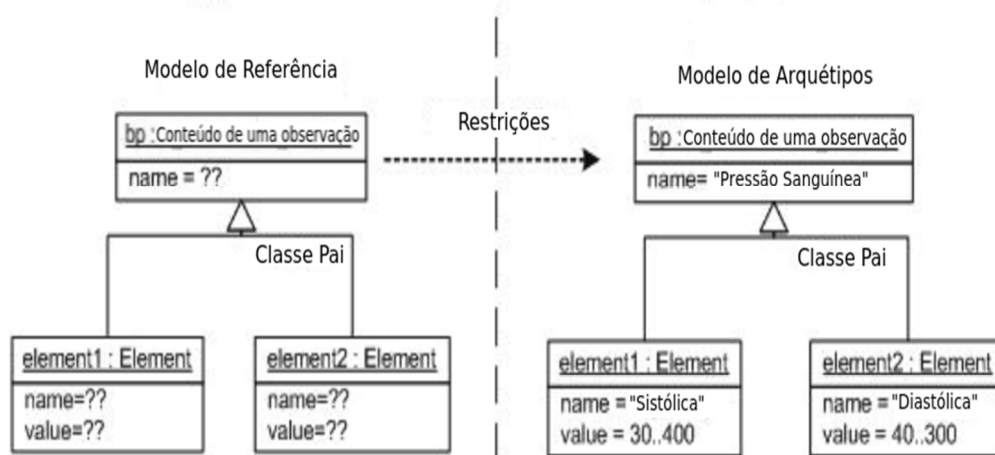


Figura 30 - Instância de Pressão Sanguínea na Abordagem Dual.  
Fonte: Adaptado e Traduzido de Kitsiou, Manthou e Vlachopoulou (2009)

Tais artefatos de conhecimento quando usados por *templates*<sup>29</sup> permitem dar forma às entradas dos prontuários eletrônicos, conforme podemos ver nas Figuras 31 e 32. As figuras mostram arquétipos OpenEHR agrupados em *templates* que servem de base para as entradas de dados das interfaces de PEP's. O leitor pode consultar aqueles desenvolvidos nas pesquisas de Nardon, França e Naves (2009) e Atalag e Yang (2010):

<sup>29</sup> *Templates*: São gabaritos voltados à formatar e facilitar a entrada de dados em um sistema.

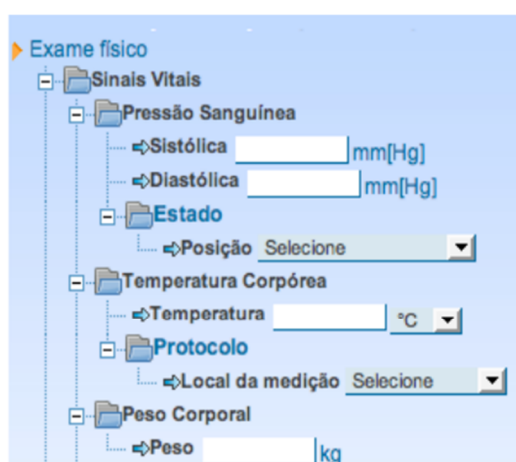


Figura 31 - Representação de Arquétipos em Forma de Árvore  
Fonte: Nardon e França (2009)

Figura 32 - Endoscopia Gastrointestinal no Sistema GastrOS  
Fonte: Atalag e Yang (2010)

Para se construir sistemas como os propostos nas pesquisas dos autores acima citados, além da serialização de *compositions* e arquétipos, codificação das regras de negócio em alguma linguagem de programação, há a necessidade da utilização de outros recursos, como

por exemplo, de bibliotecas de componentes para a interface visual com o usuário, *frameworks* ORM (no caso de linguagens orientadas à objetos e banco de dados relacionais), entre outros.

O padrão OpenEHR prevê ainda, entre vários requisitos, a necessidade de um *parser* ADL, que transforma arquétipos em ADL (*archetype description language*) para o formato de objetos na memória (OpenEHR, 2013). Tais objetos são definidos pelo *archetype object model* (AOM). O *parser* valida e gera, dinamicamente, uma instância a partir de um arquétipo ADL (*cf.* Figura 33).

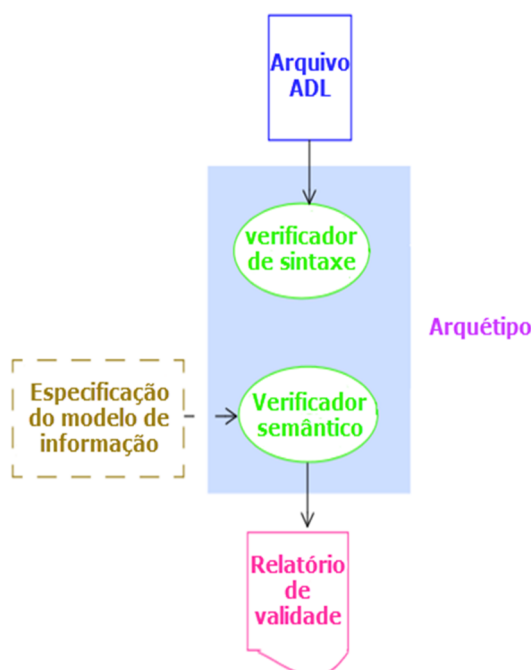


Figura 33 - Validação de um Arquivo ADL no Processo de Parseamento de um Arquétipo  
Fonte: Adaptado e Traduzido de Beale (2008)

A Figura 34 ilustra o processo de parseamento, onde um arquivo ADL é convertido pelo parser em uma árvore parseada. Esta árvore é uma representação estruturada via objetos na memória, da semântica do arquétipo, numa forma conveniente para seu posterior processamento. A árvore ADL necessita ser processada por um construtor de arquétipo (Beale, 2008), para criar, na forma de objeto, um arquétipo válido para ser utilizado num sistema de informação particular. Como entrada, o construtor requer a especificação dos modelos de informação do modelo de referência, ou seja, acessar as classes dos modelos de informação cujas instâncias formarão os dados do sistema. Isso pode ser via XML ou arquivos

de linguagens de programação validadas (OpenEHR *Specification Program*, 2013).

Para realizar o caminho inverso do procedimento acima descrito, torna-se necessário implementar um serializador ADL, que parte dos arquétipos no formato AOM da memória, para a representação textual em ADL. Segundo Chen e Klein (2007): “[...] tal procedimento é frequentemente usado antes que um arquétipo seja armazenado e transmitido entre sistemas”.

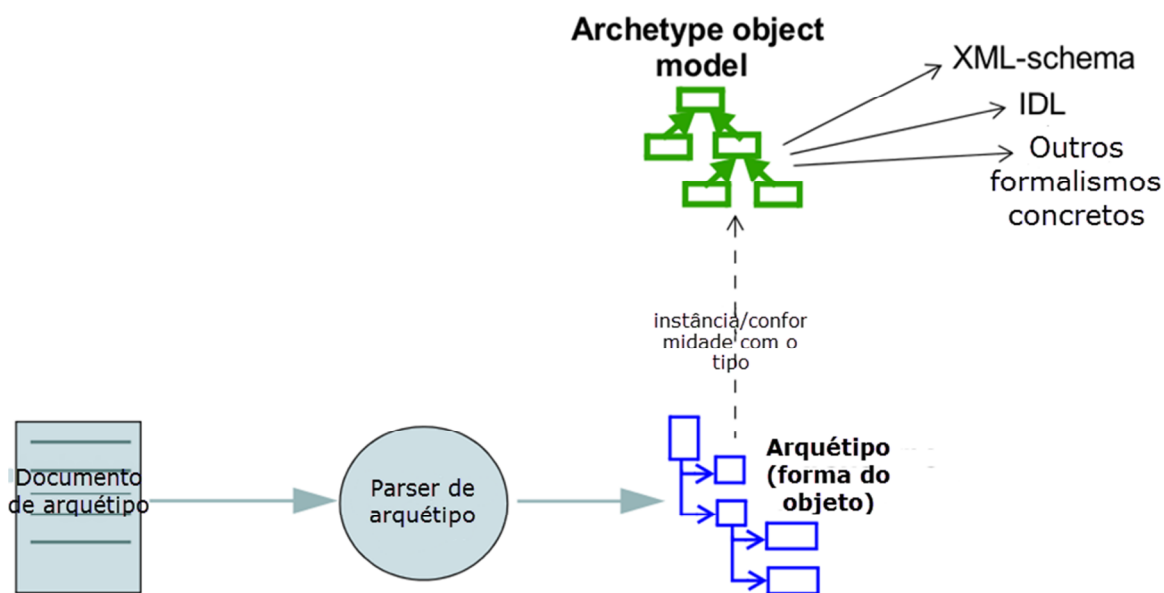


Figura 34 - Relação do AOM com a Linguagem de Arquétipos  
Fonte: Adaptado e Traduzido de Beale (2008)

Os componentes de software acima descritos podem ser considerados como parte essencial de um software para gerenciamento de informação clínica, no caso da implementação deste último seguir análoga às apresentadas por Atalag (2010) ou Nardon, França e Naves (2009), que seguem a arquitetura computacional proposta pelo padrão OpenEHR.

Diante da necessidade das funcionalidades exigidas pelos sistemas de PEP, pode-se levantar a questão: haveria a necessidade de se construir tais *softwares* seguindo sempre os mesmos requisitos?

No caso específico do padrão OpenEHR, uma vez que sua especificação foi implementada, inicialmente, na linguagem Eiffel<sup>30</sup>, a forma como se modela conceitos no OpenEHR sofreu influência desta linguagem. Tal influência na modelagem dos conceitos da

<sup>30</sup> Linguagem de programação Eiffel. Página institucional: <<http://www.eiffel.com/>>.

especificação pode gerar problemas quando da implementação das classes que se utilizam de recursos da linguagem Eiffel (Rist, 1995) não comuns às linguagens mais utilizadas, como Java, C++, C, C#, entre outras. Um exemplo de tal situação é relatado por Kobayashi e Tatsukawa (2012) em sua busca por implementar o padrão OpenEHR na linguagem Ruby, conforme mostrado no Capítulo 4 (p.40-41). Na tentativa de implementação da classe ARCHETYPE ocorre **importação circular** (suportada em Eiffel) sendo, na definição desta última, necessária a importação da classe ARCHETYPE\_ONTOLOGY e, na definição de ARCHETYPE\_ONTOLOGY, a importação da classe ARCHETYPE, conforme mostra Figura 35 abaixo.

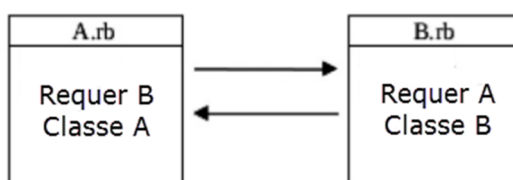


Figura 35 - Exemplo de Importação Circular  
Fonte: Adaptado e Traduzido de KOBAYASHI e TATSUKAWA (2012)

A situação de importação cruzada descrita acima pode ser vista, na figura 36, conforme visto em Beale (2008), retirada nas seções de descrição das classes ARCHETYPE e ARCHETYPE\_ONTOLOGY, onde as setas indicam os atributos do tipo de cada classe importado na classe ao lado, configurando a circularidade:

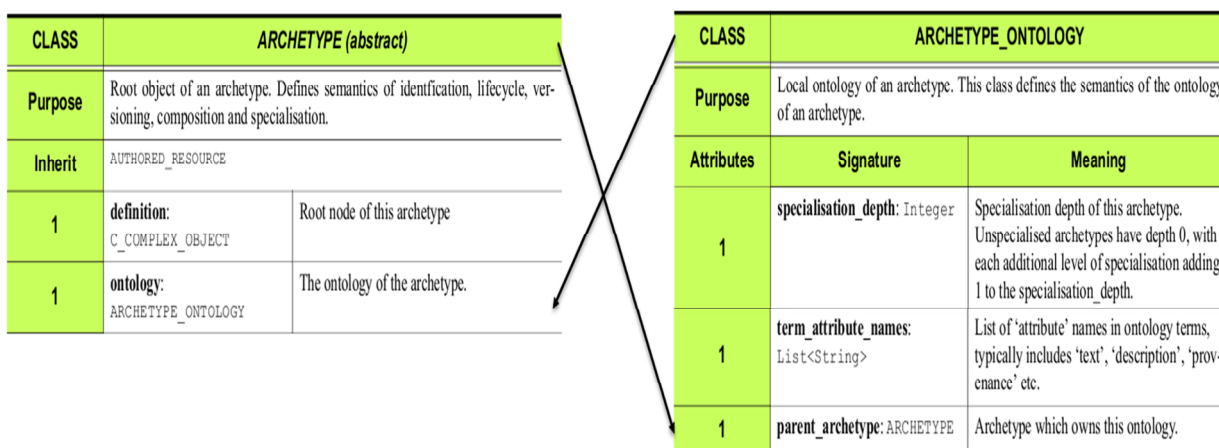


Figura 36 - Importação Circular na Especificação OpenEHR  
Fonte: Adaptado de Beale (2008)

Tal situação reforça a importância da questão, anteriormente levantada, que se busca responder: Se haveria a possibilidade de se utilizar um *framework* sistêmico que, além de agregar valor a *softwares* do tipo RES/PEP, evitasse a queda recorrente em problemas como o descrito acima, bem como permitisse sua implementação de acordo com as características desejadas e necessidades específicas, poupando o tempo/esforço normalmente dispendido numa implementação codificada a partir “do zero” dos diversos componentes como os acima citados.

Tal situação é, também, salientada por Gök (2008, p 50), onde fica claro o interesse legítimo que há em mostrar as vantagens do modelo através de implementações práticas do OpenEHR. Numa linha de pesquisa, harmônica ao trabalho desenvolvido nesta pesquisa: “[...] ao longo do tempo a abordagem do padrão OpenEHR amadureceu, porém, persiste um hiato de conhecimento de como criar um sistema baseado em OpenEHR (estratégias de implementação e migração)”.

## **7.2 Frameworks Orientados à Metadados: Os Sistemas Gestores de Conteúdo - CMS**

### **7.2.1 Informação Clínica e Metadados**

Ao se buscar uma solução tecnológica que permita o trabalho com arquétipos OpenEHR, deve-se ter em conta que não se está trabalhando com o processamento de dados, mas com informação num nível de complexidade mais elevado, advinda das várias especialidades médicas, dados demográficos, terminologias, etc.

Coloca-se, portanto, a necessidade de, uma arquitetura pensada não apenas para lidar com **dados**, mas manipular **informação** que se encontra num nível de complexidade e granularidade que advém da natureza da informação clínica. Conforme afirmam Velde e Degoulet (2003, p.108): “A granularidade das declarações clínicas/de saúde variam da orientação a texto (não-estruturado), passando pela orientação à seção (semiestruturado), às estruturas orientadas à conteúdo (codificadas)”.

Pelo que foi visto na fundamentação teórica, na discussão sobre informação clínica e na explanação sobre a semântica da especificação do modelo de referência, o nível de complexidade de um **documento clínico**, contendo diversas **declarações clínicas** é

consideravelmente alto (Capítulos 2 e 6). Todo documento clínico compartilha, além das informações que lhe são específicas, várias propriedades comuns, que necessitam ser gerenciadas, como persistência, coerência, completude, capacidade de ser lido por um ser humano, autenticabilidade, possuir prazo para ser considerado válido, entre tantas outras.

Segundo Eichelberg *et al* (2005), a especificação do OpenEHR adota a abordagem de modelagem em dois níveis. Assim, **propõem um modelo de informação em um primeiro nível**, que estabelece as classes mínimas para o gerenciamento dos registros de pacientes e suporte dos requisitos médico-legais. O segundo nível **envolve o uso de estruturas de metadados** que podem ser processadas entre emissores e receptores de informação clínica, buscando possibilitar a validação semântica de tais informações. No caso do OpenEHR, **tais estruturas de metadados são denominadas arquétipos** e, normalmente, serializadas através da linguagem ADL. Nesse segundo nível aparecem também as terminologias médicas, estruturas essenciais para a interoperabilidade de sistemas de RES. Ressaltando o que foi dito no Capítulo 5 sobre arquétipos no OpenEHR: **o modelo de arquétipos, portanto, pode ser visto como uma representação de metadados desenvolvida para organizar e padronizar dados de domínios de conhecimento.**

A adoção de padrões como o OpenEHR, que visam a interoperabilidade tem como proposta que tais documentos **compartilhem, idealmente, os mesmos metadados**, que incluiriam informações sobre o documento, quem o criou e quando, onde foi criado e qual o seu propósito, entre outros. Como exemplo pode-se tomar as funcionalidades das classes de alguns dos pacotes dos modelos de informação codificadas, discutidas no Capítulo 6, como o pacote *identification*, que define identificadores para id's diversos, *templates*, terminologias, o pacote *date\_time*, que provê classes para datas, duração, intervalo de tempo, etc.

Ou seja, tais metadados buscam responder a questões como as exemplificadas abaixo, registrando os dados via instâncias das classes do modelo de referência visando eliminar ambiguidades na informação serializada:

- Diagnóstico via instância da classe DV\_TEXT
- Severidade do problema via instância da classe DV\_CODED\_TEXT.
- Data da ocorrência do problema via instância da classe DV\_DATE\_TIME.
- Idade do paciente quando da resolução do problema via instância da classe DV\_DURATION.



- Autor do registro via instância da classe DV\_TEXT.
- Linguagem utilizada no registro via instância da classe CODE\_PHRASE.

Tal tensão entre a necessidade de gerenciar informação, compartimentalizando-a, e a necessidade de manter seu sentido é ressaltada por Boukar:

Computadores utilizam informação separada em seus elementos primários (dados) arriscando-se a perder o significado original e o contexto em que a informação foi inserida. Para gerenciar informação via computadores é **necessário separá-la em um conjunto de elementos, ou metadados, que permite que esta seja tratada como informação e não como dados** (BOUKAR, 2012, p.49, grifo nosso, tradução nossa).

Sem perda de generalidade, pode-se, para o uso neste trabalho de pesquisa, entender metadado conforme a definição de Boiko:

Metadado é o que se necessita além do próprio dado para compreender e utilizar esse mesmo dado. Metadados agem como instruções que vem junto ao dado. Em contrapartida, metadado é o que não será visto se se olhar apenas para o dado em si. **Metadados existem em adição ou por trás dos dados. Eles acrescentam contexto e uma interpretação mais extensa ao dado** (BOIKO, 2005, p.493, grifo nosso, tradução nossa).

Bem como a definição de Nadkarni:

[Metadados são] Dados cujo propósito primário é **descrever, definir e/ou anotar outro dado que o acompanha**. O dado que acompanha pode estar na mesma unidade física que o metadado (por exemplo, no mesmo arquivo no disco) ou na mesma unidade lógica (por exemplo, a mesma base de dados). [...] A informação guardada em metadados pode estar numa forma compreensível por pessoas não técnicas ou numa forma que pode ser utilizada por softwares. Frequentemente as duas formas coexistem (NADKARNI, 2011, p.1, grifo nosso, tradução nossa).

Num sistema de informação, construído conforme a proposta do padrão OpenEHR, com a complexidade da arquitetura em dois níveis, tais **metadados apresentam-se como de fundamental importância para a construção dos artefatos de conhecimento, sendo parte indispensável da própria natureza dos arquétipos**.

O comentário feito por Nadkarni (2011, p.1) a respeito da constatação de Marco (2000) onde este último afirma que “Quando falamos a respeito de **metadados**, nós, realmente, estamos falando a respeito de **conhecimento**.”, explicita o aspecto de indispensabilidade dos metadados para tal categoria de sistemas de informação:

“De uma perspectiva prática, isso significa que se você possui metadados

acompanhando dados, isso significa que é possível fazer coisas com os dados que seriam muito mais difíceis, ou talvez, **impossíveis de fazer se esses metadados não existissem**” Nadkarni (2011, p.1).

Este mesmo autor, ao tratar especificamente de sistemas orientados a metadados, os define através dessa característica referente à indispensabilidade dos metadados. Conforme se vê abaixo:

Softwares orientados a metadados são sistemas que contam com componentes de metadados para estender capacidades e funções que **não funcionariam ou se tornariam inaplicáveis se tais metadados não existissem, ficassem inutilizados, ou se tornassem inconsistentes ou corrompidos**. Isso acontece porque tais sistemas são desenvolvidos para consultar constantemente os metadados de modo a determinar seu curso de ação durante suas várias operações: **mudanças nos metadados correspondem a mudanças no comportamento do sistema** (NADKARNI, 2011, p.2, grifo nosso, tradução nossa).

O mesmo Nadkarni salienta, conforme os parágrafos anteriores, a adequação dos registros eletrônicos de saúde na categoria dos sistemas orientados a metadados:

Sistemas em ambientes de biomedicina - como sistemas de **Registro Eletrônico de Saúde**, sistemas de gerenciamento de dados de estudos clínicos e sistemas de gerenciamento de informações de laboratórios de pesquisa básica - são candidatos ideais para a aplicação de técnicas baseadas em metadados. (NADKARNI, 2011, p.v, grifo nosso, tradução nossa).

Portanto, pelos tipos de instâncias possíveis para as classes do modelo de referência do padrão OpenEHR, discutidas no capítulo anterior, os arquétipos podem conter informações sob as várias formas de comunicação passíveis de registro, como por exemplo:

- Texto.
- Som.
- Imagens.
- Vídeos.
- Arquivos de computador como planilhas, apresentações e outros arquivos diversos.

### 7.2.2 Gestão do Conhecimento Clínico via Gestão de Conteúdo

Em tal contexto, em que a camada de conhecimento ou de arquétipos corresponde a uma camada de artefatos de conhecimento contendo informações estruturadas com elevada

variação de complexidade, cabe utilizar o conceito de **capitalização do conhecimento** definido, conforme Arancon, Polo, Berrueta, Lesafre e De Abajo que levanta a associação natural entre **sistemas orientados a metadados** com o gerenciamento via o uso de **sistemas gestores de conteúdo**:

Capitalização do conhecimento cobre um grupo de **aplicações devotadas a gerenciar conteúdo, documentos e informação, estruturadas** de modo a permitir aos usuários fácil acesso ao conhecimento e adição ou modificação de dados. Atualmente, soluções diferentes estão disponíveis para tais propósitos, sob categorias como **Sistemas Gestores de Conteúdo (CMS)**, Sistemas Gestores de Documentos (DMS), *wikis*, portais web dinâmicos, máquinas de busca, etc. (ARANCON *et al.*, 2008, p.245, grifo nosso, tradução nossa).

#### 7.2.2.1 Sistemas Gestores de Conteúdo (CMS): Definição e Características.

Com o crescimento da Internet, cresceu também a necessidade de gerir conteúdos. A publicação de informações online não poderia mais ser feita de maneira manual, deixando-as sem manutenção, após sua publicação, pois esta passa a necessitar de revisão contínua visando novos consumidores e máquinas de busca, que necessitam acessar a informação sempre atualizada.

Assim, percebe-se que a internet exige uma cada vez mais rápida manutenção e atualização da informação, superando a época em que as informações eram transmitidas individualmente num modo de distribuição, manutenção e atualização complexo como, por exemplo, via email. Seu advento reduziu em muito os custos em comunicação enquanto tornou o acesso à informação praticamente instantâneo para uma larga audiência. Antes de seu advento, era aceitável a publicação de novas informações numa base mensal ou semanal, atualmente, porém, espera-se que informações importantes sejam imediatamente publicadas online. Visando possibilitar tais resultados, surgiu o sistema gerenciador de conteúdos, ou CMS. Conforme Suh *et al.*:

Gestão de conteúdo se refere a sistema e processos onde a informação é criada, gerenciada, publicada e arquivada. A informação, tipicamente passa através desse ciclo de vida por um período de tempo finito. Um sistema gestor de conteúdo (CMS) provê a infraestrutura necessária para que várias pessoas contribuam efetivamente com o conteúdo e colaborem durante seu ciclo de vida (SUH *et al.*, 2003, s/p, tradução nossa).

Um sistema gestor de conteúdo Web foi desenvolvido para suprir as necessidades de organização da informação de forma online. Um CMS possuirá, tipicamente, as seguintes características:

- facilidade na criação e edição de conteúdo por colaboradores não técnicos.
- Segurança nos direitos de acesso.
- Fluxo de trabalho (*workflow*) estruturado no processo para aprovação de conteúdo.
- Versionamento e arquivamento de conteúdo.
- *Templates* para gerar saídas consistentes.

As características acima permitem aos CMS's agregarem vantagens frente aos métodos tradicionais, particularmente na situação em que grupos distribuídos de usuários são responsáveis por coordenar e contribuir para diferentes repositórios de conteúdo. Seguem algumas vantagens a título de ilustração, outras podem ser vistas em Boukar (2012), Boiko (2005) e Suh (2003):

- Gestão do conteúdo: envolve fazer um melhor uso da informação colocando o controle de sua gestão nas mãos de seus proprietários/criadores. Os usuários podem realizar atualizações *online* de suas informações de maneira rápida e eficiente sem a necessidade de recorrer a auxílio técnico, permitindo a especialistas publicarem informações para consumidores específicos.
- Acessibilidade do conteúdo: um CMS pode adaptar o conteúdo a múltiplos formatos de saída, partindo de uma única fonte de informação e aplicando as mudanças necessárias para gerar, automaticamente, a saída em padrões como HTML, PDF, XML etc.
- Redução de custo: a criação/manutenção de conteúdo torna-se menos custosa quando os usuários podem contribuir ao disponibilizar informação online, diretamente sem a intermediação de especialistas em TI para reconfigurar o conteúdo para algum formato de apresentação. A remoção desses passos do processo de disponibilização do conteúdo deixa os especialistas técnicos livres para a realização de tarefas especializadas.

Da caracterização dos CMS's vista acima, pôde-se constatar sua inserção, também, junto à Ciência da Informação, mais especificamente como sistemas gestores de informação. Tal constatação é corroborada por Han (2005, p.356) quando afirma que: “[Um] CMS ideal é um sistema gerenciador de informação que preserva, organiza, dissemina e gerencia documentos desenvolvidos localmente e documentos externos com metadados associados”.

Numa situação específica da área de saúde, mais próxima da temática deste trabalho, especificamente envolvendo a gestão de informações num laboratório, Mooney e Baenziger (2008, p. 70) também realiza a mesma constatação ao afirmar que: "um CMS pode prover uma alternativa atraente ao uso de sistemas gestores de informação laboratorial caros, ao possibilitar o desenvolvimento da sua própria infraestrutura web".

Assim, do visto acima, ao buscar posicionar os arquétipos junto ao domínio dos sistemas gestores de conteúdo (conforme se buscará na sequência), estes também serão posicionados junto à gestão da informação, ou seja, junto à Ciência da Informação.

A escolha do Plone/CMS como framework para o esforço de expressão dos arquétipos OpenEHR deve-se a algumas de suas características, como ser construído sobre o servidor de aplicações web Zope (*Z Object Publishing Environment*), de código aberto e escrito na linguagem Python. No "topo" deste servidor, foi construído o CMF (*Content Management Framework*), que permite a separação dos objetos de conteúdo da lógica da aplicação e da apresentação. O mesmo ocorre com o ZMI (*Zope Management Interface*), uma interface de gestão para os usuários que permite adicionar e persistir objetos de software como scripts Python, *templates*, e que, a partir da versão Zope 3.0, segundo Weitershausen (2008, p. 41) "É agora uma interface para todos os propósitos, mas com um **forte foco na gestão de conteúdos**" o que vem de encontro à expectativa para o gerenciamento de arquétipos como conteúdos argumentada neste trabalho (*cf.* Seção 7.3).

Some-se isso a ser um servidor orientado a objetos, o que implica que todos os conteúdos por ele trabalhados como páginas web, imagens, links, arquivos são objetos contidos em um banco de dados orientado a objetos denominado ZODB (*Zope Object Database*). Este último, por ser uma base de dados orientada a objetos, é hierárquica e particularmente apropriada para estruturas de arquivos hierárquicos ou aninhados (Delmonte *et al.*, 2009), o que corresponde à caracterização das informações clínicas apresentada no Capítulo 2.

Sendo o ZODB orientado a objetos, tudo o que aparecer em um site Zope/Plone (páginas web, imagens, links, arquivos, etc) é um objeto e está contido em um banco de dados orientado a objetos. Uma base de dados não relacional, orientada a objetos é particularmente apropriada para o trabalho com estruturas de arquivos hierárquicos e reflete, melhor do que uma base relacional, a estrutura de um sistema de arquivos com objetos dentro de objetos. Tal propriedade mostra-se adequada para lidar com o tipo de objetos instanciado das classes do modelo de referência OpenEHR e utilizado na construção dos arquétipos (*cf.* Capítulo 6).

A base de dados ZODB preserva todas as versões antigas de um objeto: isto é particularmente útil para desfazer alterações, para o controle de versões de documentos colaborativos.

Clark *et al.* (2009) ressaltará ainda a facilidade de, no servidor Zope, desenvolver documentos XML estruturados, com *workflow* vinculado ao documento.

Para os objetivos desta pesquisa, a definição de Boukar, abaixo, para gestão de conteúdo pode, sem perda de generalidade, ser utilizada:

Defino gestão de conteúdo como um sistema de métodos e técnicas para automatizar os processos para colecionar, gerenciar e publicar conteúdos utilizando tecnologias da informação. A **gestão de conteúdo baseia sua lógica na separação entre o conteúdo e sua estrutura**. Sistemas gestores de conteúdo [CMS's] proveem o controle da criação e distribuição da informação. Permitem o conhecimento e o monitoramento do valor da informação como também gerenciar a transmissão desses dados e decidir o seu receptor (BOUKAR, 2012, p.50, grifo nosso, tradução nossa).

Pode-se utilizar, junto à definição acima, as constatações de Boiko sobre a importância dos metadados nos CMS's, permitindo, pelas definições anteriormente apresentadas, sua inclusão na categoria dos **sistemas orientados a metadados**:

**Metadados são críticos**, não apenas para permitir que o CMS se integre a outras fontes de dados distintas, mas também para possibilitar o CMS a unificar e fazer um uso mais eficiente e automatizado das funcionalidades e informações que gerencia. [...] Se o gerenciamento é a arte de nomear a informação, metadados são o conjunto de nomes. Em outras palavras, **gestão de conteúdo é totalmente a respeito de metadados** (BOIKO, 2005, p.493, grifo nosso, tradução nossa).

O mesmo Boiko reforça, novamente, essa percepção de um CMS como sendo um sistema orientado a metadados:

**O sistema de metadados por trás de um CMS é o que define o sistema.** O conjunto de nomes e relações contidos no framework orientado a metadados

é o esqueleto onde são alocados os conteúdos. Sem essa estrutura, o conteúdo torna-se disforme e flácido como um corpo humano sem os ossos (BOIKO, 2005, p.491, grifo nosso, tradução nossa).

### 7.3 Arquétipos e Conteúdo: Conceitos dos Domínios CMS e OpenEHR.

Uma vez atingida a percepção de um CMS como sistema orientado a metadados, pôde-se mostrar que é possível entender os arquétipos OpenEHR como estruturas de metadados, bem como verificar que um sistema de RES pode se enquadrar nessa categoria de sistemas.

O cenário acima descrito abre a possibilidade para que se busque a correlação entre uma estrutura de metadados, no caso dos arquétipos OpenEHR, e seu correspondente (o que deve ainda ser verificado adiante) num sistema orientado a metadados, neste caso os CMS's. Tal correlação entre elementos, até o momento, pertencentes a **dois domínios diferentes**: sistemas gestores de conteúdo e padrões para interoperabilidade semântica de dados clínicos (Figura 37 abaixo), abre possibilidades quanto à construção de softwares de RES/PEP e ao gerenciamento de seus artefatos de conhecimento. Ou seja, pode-se ter melhor custo-benefício que a forma padrão apresentada pela arquitetura computacional do OpenEHR.

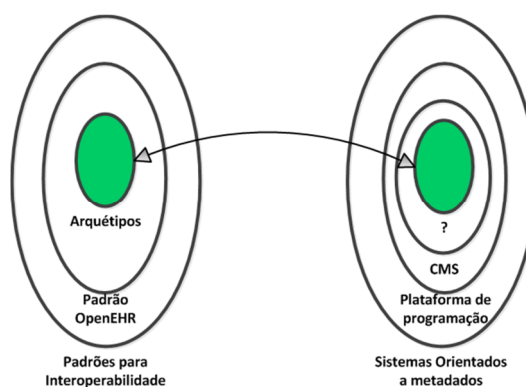


Figura 37 - Contexto da Correspondência entre Arquétipos e um Elemento do Domínio dos CMS's

Fonte: Elaborado pelo Autor

Buscando atingir o objetivo acima, torna-se necessário encontrar, na arquitetura dos CMS's, o conceito cujas propriedades e funções assemelhem-se às do conceito de arquétipo. É essencial auferir a similitude entre estes conceitos dos dois domínios se se busca a gestão de arquétipos via CMS's.

O conceito de **conteúdo**, definido por Boukar, pela semelhança com a noção central de **arquétipos** (informação acrescida de metadados), apresenta-se como suporte para sua representação no domínio dos CMS's:

Para gerenciar conteúdo, é necessário contextualizar a informação. Na prática, **conteúdo é informação enriquecida com dados**. Basicamente, **conteúdo é uma suíte de dados estruturados** que um computador pode organizar em um sistema para sua coleção, gerenciamento e publicação (BOUKAR, 2012, p.49, grifo nosso, tradução nossa).

O mesmo autor amplia a definição que, da mesma forma que a anterior, remete à conceituação de arquétipo apresentada nos capítulos anteriores (contexto dos registros eletrônicos de saúde - RES), conforme se pode ver abaixo:

**Informação torna-se conteúdo** quando é utilizada para um ou mais propósitos. Seu valor é a soma da constituição primária (informação), aplicação, usabilidade, significância e unicidade. É **informação acrescida de uma camada de conjuntos de dados** [metadados] num contexto específico (BOUKAR, 2012, p.49, grifo nosso, tradução nossa).

De modo análogo, Boiko define conteúdo em função de informação e metadados, reforçando a percepção da equivalência conceitual com arquétipos:

**Conteúdo**, portanto, é **informação que se rotula com [meta]dados e reunida em coleções** que um computador pode organizar, sistematizar, gerenciar e publicar. Tal sistema, um sistema gestor de conteúdos, é bem sucedido, se pode aplicar metodologias [orientadas a metadados] em seus dados sem deixar que se perca, ao longo do percurso, o interesse e significado da informação (BOIKO, 2005, p.495, grifo nosso, tradução nossa).

Boiko (2005) elenca as diversas entidades existentes num CMS, em torno das quais se estrutura os projetos nele desenvolvidos. Entidades tais como autores, audiências, metas e requerimentos, tipos de conteúdo, publicações, *workflow*, entre outras, onde, numa visão esquemática, um sistema gestor de conteúdo pode ser visto como uma interação balanceada, cujo balanço é mantido pelos metadados, entre as forças dessas entidades que o definem. Em resumo, são os **metadados que mantêm o sistema coeso** e lhe proveem sua forma. A Figura 38 a seguir, ilustra tal esquema:



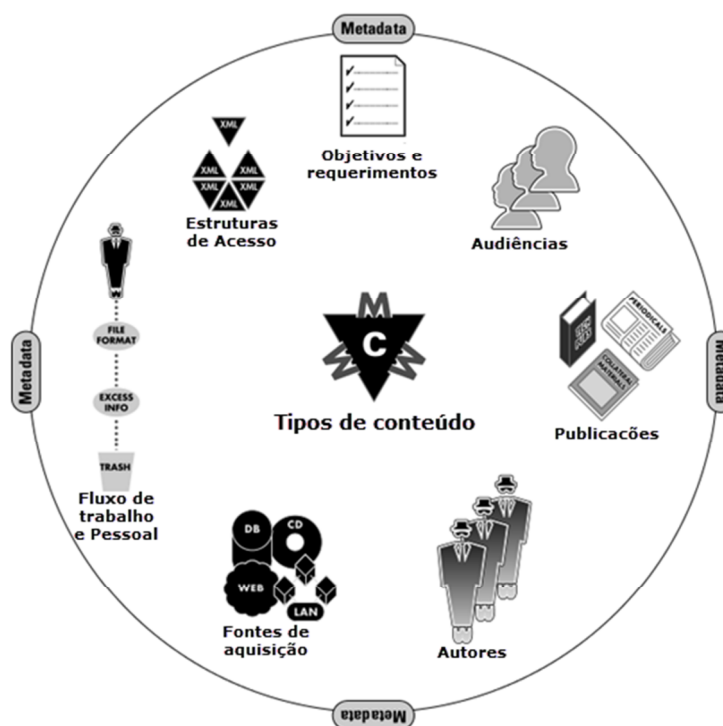


Figura 38 - O Círculo da Gestão de Conteúdo.  
 Fonte: Traduzido de Boiko (2005)

Esta maneira, com alto grau de abstração, de se enxergar um sistema gestor de conteúdos se adequa à característica de independência de hardware e software desse tipo de sistema, pois gestão de conteúdos é independente de hardwares e softwares particulares sobre os quais se constrói o sistema. Antes, apresenta-se como um processo organizacional para alinhar, colaborativamente, forças competitivas visando reunir e fornecer conteúdos valorados.

Percebe-se assim, o papel central desempenhado pelo conceito de conteúdo nos CMS's e sua relação com os metadados, pois metadados e tipos de conteúdo relacionam-se fortemente um com o outro. A maioria dos metadados é armazenada nos elementos dos tipos de conteúdo que são definidos. **Tipos de conteúdo** podem ser vistos como porções de informação ou de funcionalidades envolvidos em metadados. Visto sob essa ótica, fica patente a estreita similaridade entre o conceito de tipos de conteúdo com o conceito de **arquétipos** discutido ao longo de todo o texto, não só em termos dos conceitos de informação e metadados, mas pelo conceito de reuso.

Conforme visto anteriormente objetivos como a interoperabilidade e o reuso

estão entre os principais norteadores para o esforço de definição de padrões como o OpenEHR. Ressaltando parte do que foi dito (Capítulo 5, p. 59): “A maior expectativa é a possibilidade de **reuso** das estruturas de registro de informação, em geral complexas. [...] Além de fomentar o maior reuso do conhecimento, a utilização de arquétipos pode ser vista como uma solução possível para a heterogeneidade das informações em saúde”.

Logo, se até o ponto atual do texto, a argumentação apresentada procurou mostrar como os conceitos de arquétipo e conteúdo, podem, em seus respectivos domínios, serem considerados como similares, cabe verificar, então, se tal similitude persiste no aspecto funcional que envolve os arquétipos. Tal busca e verificação passa pelo reuso, visto que, sem tal propriedade, o esforço dispendido na definição de arquétipos mostrar-se-ia como tendo, claramente, um custo benefício insatisfatório.

Guiando-se pelos aspectos essenciais do modelo de arquétipos do padrão OpenEHR, partindo do fato de que o reuso dos artefatos de conhecimento, no padrão OpenEHR, advém da possibilidade de se definir um arquétipo e, posteriormente, reutilizá-lo em quantos projetos de software o conhecimento nele encapsulado for necessário; A questão a ser colocada é: **Se conteúdo é o similar conceitual, no domínio dos CMS's, ao conceito de arquétipos; possui o primeiro um mecanismo de instanciação análogo, que possibilite o seu reuso no sistema definido neste domínio?**

A resposta à indagação anterior pode ser respondida pelos conceitos do domínio dos CMS's denominados, **modelo de conteúdo, tipos de conteúdo e componentes de conteúdo**, cuja importância no sistema é ressaltada por Boiko:

No coração de qualquer sistema gestor de conteúdos há **agrupamentos de informação que são reutilizáveis** em suas várias publicações. Cada um desses agrupamentos, que denomino **componentes** de conteúdo, origina-se de um grupo que chamo **tipo de conteúdo**. **Dividir suas informações em tipos** é o primeiro e maior passo para se tornar capaz de gerenciar a criação e distribuição de suas informações (BOIKO, 2005, p.559, grifo nosso, tradução nossa).

Estes três conceitos devem ser entendidos numa hierarquia de abstração em que o modelo de conteúdo se apresenta como o nível mais abstrato (conforme explicado no parágrafo a seguir), seguido do tipo de conteúdo (análogo a classes em programação orientada

a objetos) e, por fim os tipos de conteúdos (seriam como as instâncias das classes em programação orientada a objetos).

Procurando compreender o modelo de conteúdo comparando-o com o conhecido modelo de dados, vemos que um modelo de conteúdo é análogo a um modelo de dados. Este último trata estritamente sobre **como** a informação é armazenada; o tipo de conteúdo, porém, diz respeito a **o que a informação é**. Assim, **o modelo de conteúdo deve ser considerado mais abstrato que o modelo de dados**. Pois, inicialmente define-se o modelo de conteúdo e, então, quando da implementação do CMS, define-se qual modelo de dados é necessário para armazenar os conteúdos contidos no modelo de conteúdo.

Assim, conforme salienta Boiko (2005), todo sistema gestor de conteúdos ou **CMS deve possuir um modelo de conteúdos**. E, como um modelo de dados, um modelo de conteúdos é designado para descrever os usos específicos dos conteúdos que modela.

O modelo de conteúdo define o que seu conteúdo é, enquanto o modelo de dados define como os conteúdos serão armazenados na base de dados (ou, numa estrutura XML).

O modelo de conteúdo consiste nas seguintes partes (o termo “*element*” foi mantido na grafia original em inglês devido à sua similitude com o termo de mesma grafia dos arquétipos OpenEHR):

- O domínio de conteúdo é o mais alto nível de abstração, o universo de todos os conteúdos a serem gerenciados.
- Estruturas de acesso especificam como o conteúdo poderá ser organizado, gerenciado, hierarquizado, indexado, sequenciado e receber referências cruzadas.
- Tipos de conteúdo são o maior subconjunto de conteúdos dentro de um dado domínio definido no CMS. Tipos de conteúdo definem o que é o conteúdo num nível abaixo do domínio de conteúdo.
- *Element* é o principal integrante de um tipo de conteúdo a carregar informação. Os elementos especificam o que o conteúdo é num nível mais detalhado que o do tipo de conteúdo. Por exemplo, um tipo de conteúdo denominado serviços pode incluir um título, o tipo de serviço a ser prestado (gerência de projeto, consultoria ou suporte técnico) e uma descrição do serviço, onde, no tipo de

conteúdo, cada um será representado por um *element* distinto.

- Tipos de *elements* e valores permitidos especificam que tipos de dados são permitidos constar em cada *element*. Por exemplo, tipo de serviço pode ser uma lista para que se escolha entre gerência de projeto, consultoria ou suporte técnico. Assim, Tipos de *elements* e valores permitidos especificam no menor nível de detalhamento, o que é o conteúdo.

Ressalta-se não só a semelhança sintática, citada em nota acima, entre “*Element*” para um arquétipo e “*Element*” para o modelo de conteúdo de um CMS, mas a completa semelhança funcional entre eles. Ambos encarregados de portar os menores dados que comporão as estruturas maiores dos arquétipos, ocorrendo o mesmo, de modo análogo, com os conteúdos.

A relação entre um modelo de dados e um modelo de conteúdos deve ser vista como intrínseca. Ao fim, muito do modelo de conteúdo irá se tornar um modelo de dados, pois se o CMS é construído sobre uma base de dados que sustenta seu conteúdo, a base de dados possui um modelo de dados para representar seu conteúdo. Conforme Boiko:

Durante o projeto lógico, define-se com o que o modelo de conteúdo deve parecer. Durante o projeto físico, define-se o modelo de dados que o conteúdo definido pelo modelo de conteúdo e durante a implementação, inicia-se a criação e armazenagem dos conteúdos de diversos tipos que serão utilizados. Quando se cria e armazena conteúdos, se está criando porções de conteúdo que podem ser de vários tipos. Essas porções são denominadas componentes. **Componentes são porções de conteúdo que seguem o modelo de conteúdo definido para um tipo de conteúdo particular** (BOIKO, 2005, p. 563, grifo nosso, tradução nossa).

Portanto, um CMS permite a criação de diversos **tipos de conteúdo** em um esquema coerente em que **componentes de conteúdo** relacionam-se uns com os outros possibilitando a criação de sistemas que permitam sua coleção, gestão e publicação.

A relação entre modelo de conteúdos e modelo de dados é expressa conforme mostra a Figura 39 a seguir:

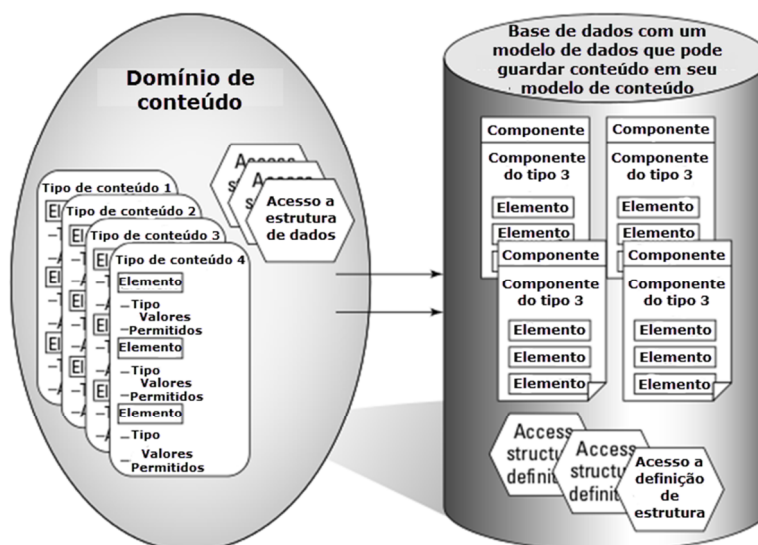


Figura 39 - Correlação entre Modelo de Conteúdo e Modelo de Dados  
 Fonte: Traduzido de Boiko (2005)

Uma vez que se busca identificar um isomorfismo funcional entre elementos dos domínios dos CMS's e do padrão OpenEHR, o esquema conceitual e a discussão apresentados permitem confirmar, em resposta a indagação anteriormente posta, a relação entre **tipos de conteúdo** e **arquétipos**, como também entre **componentes de conteúdo** e **instâncias de arquétipos**.

Buscando apresentar tal correlação de uma maneira mais intuitiva, Boiko utiliza uma analogia com a programação orientada a objetos que vale, tanto para o domínio OpenEHR/arquétipos quanto para CMS/conteúdos:

Como um programador de linguagens orientadas à objeto, o arquiteto de conteúdo cria classes de conteúdo (chamadas tipos de conteúdo) e instâncias de conteúdo (chamadas componentes). A classe/tipo define a estrutura geral e a instância/componente o conteúdo específico para dentro da estrutura geral do sistema (BOIKO, 2005, p.565, grifo nosso, tradução nossa).

Assim, tipos de conteúdo dividem as informações com as quais se necessita lidar em porções gerenciáveis e com dimensão conveniente. Para tanto, são definidos tipos de conteúdo visando estabelecer um conjunto de objetos de conteúdo que se pode criar, manter e distribuir. No CMS, o conteúdo é armazenado na forma de componentes, que são instâncias de tipos de conteúdo particulares.

A equivalência entre os elementos dos domínios discutida pode ser expressa na Figura 40:

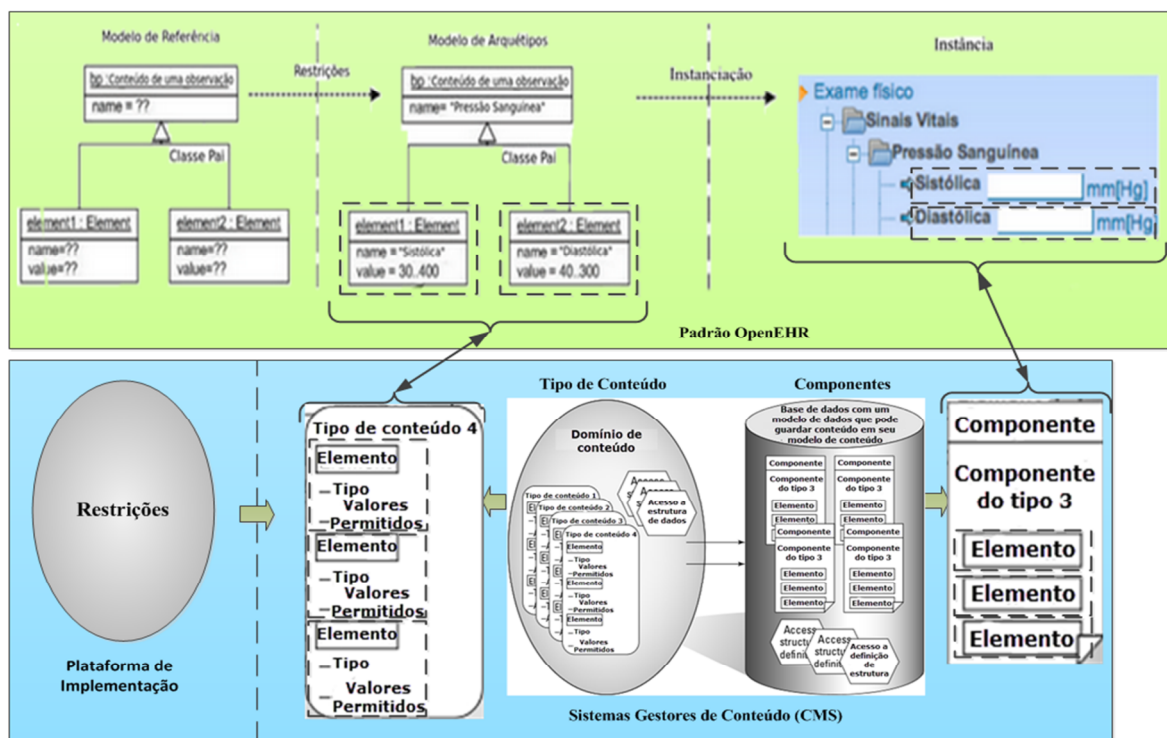


Figura 40 - Equivalência entre Arquétipos/OpenEHR e Conteúdo/CMS  
 Fonte: Adaptado e Traduzido de Kitsiou, Manthou e Vlachopoulou (2009); Boiko (2005); Nardon e França (2009)

### 7.3.1 Ressalvas Quanto a Representação de Arquétipos OpenEHR como Conteúdos

Uma vez discutida, em termos conceituais, a similitude entre os conceitos de arquétipo e conteúdo, instância de arquétipo e componente de conteúdo, cabe ressaltar que, apesar da garantia de que, conforme dito anteriormente, todo CMS possui um modelo de conteúdo, isso não implica necessariamente numa facilidade imediata para se expressar e utilizar qualquer arquétipo publicado como conteúdo.

Os requisitos para a expressão das instâncias das classes do modelo de referência e do próprio(s) arquétipo(s) com o qual se deseja trabalhar devem ser confrontados com as possibilidades de modelagem de conteúdos expressa pelo modelo de conteúdo e pelo modelo de dados do CMS com o qual se pretenda trabalhar, pois este é construído sobre uma plataforma de desenvolvimento em particular para atender a requisitos estabelecidos e, tais requisitos podem não ser compatíveis com aqueles necessários para a expressão de um arquétipo.

O mesmo raciocínio expresso no parágrafo anterior vale para *frameworks* construídos e integrados a CMS's com o objetivo de automatizar e tornar mais intuitiva a criação de conteúdos. *Frameworks* como o *Entities* (Spets, 2013), construído para o CMS *Drupal*<sup>31</sup> ou o *Dexterity*, construído para o CMS *Plone*<sup>32</sup>. Tais ferramentas buscam tornar mais intuitiva a criação de novos conteúdos em projetos que estejam sendo desenvolvidos nos CMS's aos quais se integram. Tornam a criação e definição da apresentação dos conteúdos mais fácil para o projetista, automatizando o CRUD dos conteúdos nas bases de dados. Porém, tais facilidades não implicam, necessariamente na ausência do levantamento e verificação de requisitos, conforme comentado anteriormente, visando verificar as possibilidades expressivas de tais CMS/*frameworks* frente às restrições necessárias à representação dos arquétipos OpenEHR.

### 7.4. RES/PEP Via CMS

A possibilidade da representação de arquétipos via conteúdos em sistemas gestores de conteúdos abre possibilidades para o desenvolvimento de sistemas de PEP/RES implementados de forma diferente ao padrão “codificado a partir do zero” discutido no início

---

<sup>31</sup> CMS Drupal. Página institucional: <<https://drupal.org/>>

<sup>32</sup> Plone. Página institucional: <<http://plone.org/>>

do capítulo. Conseqüentemente, pode-se evitar o dispêndio de tempo e esforço na implementação das etapas e componentes necessários à solução que siga por essa via.

Além do ganho assinalado no parágrafo anterior, a possibilidade da representação de arquétipos como conteúdos num CMS torna mais intuitiva e menos dispendiosa a prototipagem de *interfaces*. Aliado à natural capacidade dos CMS's para gerenciar os mais diversos tipos de conteúdo, isso permite lidar com um dos problemas que, segundo Lusk mais levam a falhas em sistemas dessa natureza, conforme ressaltado abaixo:

Muitos registros médicos eletrônicos falham devido a interfaces de entrada de dados não intuitivas. [...] muitos sistemas são inflexíveis e não permitem aos médicos o *design* de uma interface que satisfaça suas necessidades. Um bom sistema de registro médico eletrônico reúne informação de outros sistemas de informação existentes e a apresenta num formato intuitivo e consistente, agindo como uma interface universal para imagens, texto, e acesso a documentos legados (LUSK, p.1227, 2002, tradução nossa).

O mesmo autor assinala que o registro médico eletrônico ideal deve possuir uma interface intuitiva que modela os hábitos naturais dos médicos para entrada e revisão de informação. Deve ser adaptável para permitir as variações necessárias nas documentações e deve prover uma entrada fácil de informação altamente detalhada de qualquer paciente. Tais requisitos, conforme a caracterização dos CMS's já vista e sua facilidade na gestão de conteúdos, podem ser atendidos por sistemas construídos de modo mais eficiente e maior custo benefício, sobre esse tipo de plataforma.

Outra característica a compor um sistema de RES ideal, apontada por Lusk é o *workflow*, conforme se pode ver na referência abaixo:

Se implementado apropriadamente, o sistema de registro médico eletrônico ideal deve produzir avanços **significativos em *workflow*** e na real redução em uso de papel, decrescendo a carga de documentação. Um sistema que provê vias clínicas bem definidas, *workflow* eficiente para documentação e a precisa captura de informação colabora com os protocolos de cuidados médicos bem como na efetiva monitoração do tratamento (LUSK, 2002, p.1232, grifo nosso, tradução nossa).

No caso da implementação dessa característica em particular, novamente, o uso de um sistema desenvolvido sobre um CMS se mostra interessante. Gerir o *workflow* é uma característica essencial de todo sistema CMS, conforme salienta Suh:

*Workflow* ajuda [o CMS] a manter o fluxo de conteúdo através de sua estrutura tecnológica, e garante que a estrutura humana o aprove e esteja ciente de tudo o que acontece durante [o fluxo]. [...] A habilidade de obter



uma visão geral de como as páginas estão sendo construídas é chave para manter o trabalho. Um workflow concebe o processo, mas esse tipo de gestão é como olhar para o chão de uma fábrica para ver quem está trabalhando e quem está deixando o trabalho acumular. (Suh *et al.*, 2003, s/p, tradução nossa).

Da mesma forma, características elencadas pelo modelo de referência do padrão OpenEHR, como necessárias aos arquétipos como o versionamento (Beale *et al.*, 2008e), podem ser implementadas através do uso da estrutura de um sistema gestor de conteúdos. O mesmo vale para as características de recuperação, segurança, escalabilidade, robustez, geração de *logs* e de relatórios (Boiko, 2005).

## 8 CONCLUSÕES

Neste capítulo, sob a ótica dos dois questionamentos norteadores que guiaram a pesquisa aos seus principais resultados, são revistos: a trajetória da pesquisa desenvolvida na tese, os resultados alcançados e a importância do tema abordado.

Inicialmente examina-se a busca por viabilizar, pela implementação do OpenEHR em Python, mecanismos que possibilitem a interoperabilidade semântica em sistemas de RES/PEP nela construídos. A seguir, o mesmo é feito à luz do esforço na busca por um *framework* sistêmico que viabilize a representação de informação clínica e que permita a criação/gestão de artefatos de conhecimento com vantagens diversas, frente a uma implementação dos mesmos "a partir do zero". Esta análise ocorre sob a ótica das especificidades dos sistemas gestores de conteúdo ou CMS. Apresentar-se-á, também, conjunto de temas que, derivados deste trabalho de pesquisa, podem ser abordados em trabalhos futuros.

### 8.1 Modelo de Referência e Modelo de Arquétipos como Soluções na Busca da Interoperabilidade em Registros Eletrônicos de Saúde.

Conforme foi discutido ao longo deste trabalho, o modo tradicional de realizar o registro de saúde, através de fichas em papel deixa margem para a ocorrência de diversos tipos de erros, que comprometem tais registros. Na busca por soluções, utilizando-se de novas ferramentas da TIC, foi proposto o Registro Eletrônico de Saúde, que por definição superava diversos erros dos registros em papel, mas trouxe consigo, o aumento da complexidade informacional quando de sua implementação em diversas plataformas e modelos de dados.

Frente a esse novo desafio, a modelagem dual, oriunda das pesquisas voltadas à representação de conhecimento na área da Inteligência Artificial, apresenta solução ao possibilitar a separação das camadas de informação e de conhecimento. Assim, apenas a primeira camada, menos abstrata, é implementada em *software*. Os artefatos de conhecimento da segunda camada são construídos por especialistas da área médica, buscando o máximo possível de desvinculação em relação a plataformas de desenvolvimento e modelos de dados (Beale *et al.*, 2008a, p. 12). Estes últimos são as peças intercambiáveis que permitem a interoperabilidade entre os sistemas de RES/PEP.

Tais sistemas, embora sejam projetados sob diferentes plataformas de desenvolvimento utilizam artefatos de conhecimento construídos com base na mesma semântica. O que torna possível a troca de mensagens mutuamente inteligíveis entre sistemas que adotam plataformas e modelos de dados heterogêneos, bem como a coerência semântica (manutenção do significado) das informações registradas dentro das bases de dados.

## 8.2 Viabilizando a Expressão de Informação Clínica na Plataforma Python

Uma discussão sobre assunto amplo como a interoperabilidade semântica em sistemas de registro de saúde poderia incluir considerações e trajetórias tão variadas que uma providência salutar no início do trabalho foi delimitar o escopo do problema a ser tratado. A proposta adotada foi, após estudo exaustivo da especificação OpenEHR, verificar como tornar possível a expressão dos artefatos de conhecimento ou arquétipos na plataforma de programação Python.

A percepção propiciada, neste caso, pela modelagem dual tornou patente que, antes de se obter a expressão de artefatos de conhecimento OpenEHR, pertencentes à **camada de conhecimento** do referido modelo, seria necessário verificar/mostrar que os elementos da semântica proposta pelo padrão, contidos nos modelos de informação, pertencentes à **camada de informação** (abaixo da camada de conhecimento), poderiam ser expressos nesta mesma plataforma.

Tal verificação, que deve ser realizada, necessariamente, antes da implementação da camada de conhecimento é crucial para a pretensão de expressar informação clínica em qualquer plataforma, pois todo artefato de conhecimento, ou arquétipo, é “construído” a partir de instâncias das classes definidas pelos modelos de informação.

Em suma, utilizando-se as classes dos modelos de informação, um número ilimitado de arquétipos pode ser construído. Com efeito, tais classes representam as características globais dos componentes do registro em saúde, como são agregados, e a informação contextual necessária para atender aos seus requisitos. Ele define o conjunto de classes que formam os blocos genéricos para construir o RES.

Embora todas as exigências da análise quanto aos tipos primitivos necessários, conforme as definições da especificação OpenEHR tenham sido atendidas, garantindo assim a possibilidade de se trabalhar na construção de todas as classes dos modelos de informação,

especificidades oriundas da utilização da linguagem Eiffel na especificação do padrão (Eiffel adota o *design by contract* - dbc) exigiram a adoção de procedimentos como a aplicação de testes unitários<sup>33</sup> nas funções codificadas nas classes Python criadas, visto serem estes últimos considerados como equivalentes aos contratos do dbc.

Após esforço de estudo da especificação, adaptação à plataforma escolhida e codificação (em ordem crescente de complexidade das classes dos seus vários modelos de informação), o objetivo, posto como necessário para que se pudesse afirmar, de modo categórico, a possibilidade de se expressar informação clínica (no padrão OpenEHR) na plataforma Python foi atingido.

Tal resultado, não apenas abre caminho para a utilização de arquétipos OpenEHR na plataforma Python, mas também para artefatos de conhecimento de padrões que utilizem, ou possam vir a se utilizar do modelo de referência OpenEHR. Este é o caso do padrão *Multi-Level Healthcare Information Modelling* ou MLHIM<sup>34</sup>, cujo equivalente ao arquétipo é denominado “Definições de Restrições para os Conceitos” (*Concept Constraint Definitions*) ou CCD, e que se utiliza do modelo de referência do padrão OpenEHR.

Deve-se lembrar, porém, que as benesses advindas da utilização da modelagem dual, particularmente a tão buscada interoperabilidade semântica, dependem de uma governança centralizada dos artefatos de conhecimento. Não basta que os arquétipos sejam elaborados por especialistas da área médica e representados corretamente. Com efeito, como a construção de arquétipos ocorre de maneira isolada, a não exposição do trabalho à comunidade e a falta de uma governança centralizada poderão levar ao desenvolvimento de arquétipos tratando dos mesmos conceitos, mas com diferentes níveis de granularidade. Um obstáculo à interoperabilidade entre os diferentes sistemas que os adotarem.

No caso do padrão OpenEHR, a governança é realizada através do CKM<sup>35</sup> ou *Clinical Knowledge Manager*, um sistema *web* desenvolvido para ser um repositório de arquétipos. Além de facilitar o desenvolvimento colaborativo, a gestão e publicação de arquétipos,

---

<sup>33</sup> Testes unitários são criados durante a fase de desenvolvimento de um programa ou em seu início. O objetivo dos testes unitário é testar a menor unidade existente do software (onde uma unidade é a menor parte testável de um programa), ou seja, isolar parte dos códigos e métodos, e analisar se essas funcionalidades apresentam o retorno esperado mediante valores informados. Idealmente, cada teste de unidade é independente dos demais, o que possibilita ao programador testar cada módulo isoladamente.

<sup>34</sup> *Multi level healthcare Information Modelling*. Página Institucional: <<http://mlhim.org/>>.

<sup>35</sup> *Clinical Knowledge Manager*. Página institucional: <<http://openehr.org/ckm/>>.

permite a implementação de um modelo de governança que evita situações como a descrita no parágrafo anterior.

### **8.3 Os Sistemas Gestores de Conteúdo como *Framework* para a Implementação de Softwares de RES/PEP.**

Com o resultado positivo quanto à possibilidade da expressão de informações clínicas em Python, ou seja, a construção de arquétipos nessa plataforma, passou-se à análise de como utilizar um *framework* sistêmico capaz de atender a etapas da construção de um software de RES/PEP no padrão OpenEHR. O objetivo seria eliminar etapas como a utilização de parser e serializador ADL ou as dificuldades como a importação circular no modelo de arquétipos, advindas do uso da linguagem Eiffel na modelagem da especificação (cap. 7, p. 113).

Neste trabalho, o *framework* analisado restringiu-se aos sistemas gestores de conteúdo. O que nos permitiu fazer a relação entre esse tipo específico de *software* e os artefatos de conhecimento do padrão OpenEHR foi o fato de arquétipos serem estruturas de metadados, enquanto que os CMS's são classificados como sistemas orientados à metadados.

A importância dos metadados para este tipo de *software* é crucial, pois permite organizar e padronizar o conhecimento dos domínios trabalhados como conteúdos. Os sistemas gestores de conteúdo possibilitam o uso eficiente e automatizado da informação do domínio de conhecimento através de sua representação via conteúdos.

A análise realizada, tendo os metadados como pano de fundo, possibilitou estabelecer o argumento da similitude conceitual entre arquétipos e conteúdos (cap. 7, p. 123). Uma ligação, antes inexistente, entre o domínio dos sistemas gestores de conteúdo e dos padrões para a interoperabilidade de dados clínicos.

Tal resultado posiciona a temática norteadora deste trabalho (o uso de padrões para interoperabilidade semântica em sistemas de registro eletrônico de saúde) junto a pesquisas análogas do campo da Ciência da Informação. Mais especificamente junto a pesquisas de gestão da informação. Dessa forma, o desafio da interoperabilidade semântica em RES passa a ser visto e tratado também sob a ótica da teoria e das técnicas da Ciência da Informação. O que agrega valor à busca, já existente, de soluções sob a perspectiva da Ciência da Computação.

#### 8.4 Perspectivas para Trabalhos Futuros e Questões Pendentes

Advinda da primeira parte da pesquisa, em que se analisou a possibilidade da expressão de informações clínicas na plataforma Python, a implementação, nesta mesma plataforma, do modelo de informação demográfico<sup>36</sup>, que constitui parte essencial do padrão OpenEHR e permite a inserção de conteúdo demográfico nos arquétipos criados, bem como do seu modelo de conhecimento, ou de arquétipos, apresenta-se como desenvolvimento natural da pesquisa. O mesmo pode ser dito para as ferramentas necessárias para o *parseamento* de arquivos ADL e para a geração de objetos na memória conforme especificado pelo *archetype object model* ou AOM proposta pelo OpenEHR (ver Capítulo 7, p. 109-110).

Assim como trabalhos futuros que, advindos diretamente do resultado da segunda parte desta pesquisa devem considerar, fortemente, a construção de artefatos de conhecimento, na forma de conteúdos em sistemas gestores de conteúdo, utilizando o CMS Plone (desenvolvido sobre a plataforma Python), confrontando os requisitos dos arquétipos OpenEHR com as possibilidades de modelagem e a geração automática de conteúdos oferecida pelo *framework dexterity*. Tal resultado possibilitará a realização de provas de conceito envolvendo a geração automática de conteúdos/arquétipos neste CMS, abrindo caminho para futuras implementações de softwares RES/PEP bem como a comparação dos resultados obtidos via o uso de *frameworks* de conteúdo com os obtidos através da codificação “pura”, sem o seu uso.

O resultado, também abre caminho para que novas contribuições, vindas por outros trabalhos do mesmo gênero sejam acrescentadas e disponibilizadas, possibilitando que implementações nesta plataforma, utilizando recursos ainda não testados como *framework* para desenvolvimento web em Python Django<sup>37</sup>, entre outros, bem conhecido pelos desenvolvedores Python que poderia agregar valor ao trazer a possíveis implementações web de softwares RES/PEP as suas funcionalidades facilitadoras.

Implementações e provas em CMS's, como as propostas acima, tornam possível, também, a realização de testes envolvendo a utilização de padrões de *design* (Schmidt; Stal; Rohnert; Buschmann, 2000) como o MVC, acrônimo para o padrão *Model-View-Controller*,

---

<sup>36</sup>Especificação do Modelo de Informação Demográfico OpenEHR: <<http://www-test.openehr.org/programs/specification/releases/currentbaseline>>

<sup>37</sup>Framework web em Python, Django. Página institucional: <<https://www.djangoproject.com/>>

que descrevem uma solução geral reutilizável para algum problema recorrente no desenvolvimento de sistemas de software orientados a objetos, podendo ser utilizada em muitas situações diferentes.

Os CMS's, embora, a princípio, possam permitir a aplicação de padrões de *design* não os impõe à priori. Isso abre espaço para a verificação sobre possíveis ganhos de desempenho da combinação arquétipos OpenEHR, expressos como conteúdos em CMS's aplicando o padrão de *design* MVC. Um resultado positivo quanto ao uso de um padrão de *design* em arquétipos/conteúdos num CMS agregaria valor ao permitir que uma solução testada, bem documentada e de conhecimento geral seja utilizada em futuros RES/PEP desenvolvidos sobre esse tipo de plataforma.

Conforme trabalho realizado por (Frade *et al.*, 2013), que levantou as varias tecnologias utilizadas para armazenamento dos dados de diversas implementações envolvendo arquétipos, estudos comparando o desempenho desses sistemas ainda devem ser realizados. O estudo indica uma dominância dos sistemas relacionais. Porém, a impedância do modelo OpenEHR frente às estratégias de indexação e procedimentos automáticos das implementações em bancos relacionais sugerem e justificam testes e verificações da utilização de outras soluções que não a relacional, como bancos orientados a objetos e bases de dados do tipo NoSQL.

Outro ponto relevante relacionado a futuros sistemas de RES/PEP que venham a seguir o padrão OpenEHR é a existência de dois problemas com a forma como as terminologias são tratadas, conforme salienta Nardon (2008). O primeiro problema é a forma como uma terminologia é associada com um elemento de um arquétipo. De acordo com a especificação, esta associação é feita através de uma URL. Ou seja, se um elemento, por exemplo, “diagnóstico” deve receber como valores possíveis itens do capítulo “A” da tabela CID-10, basta que o criador do arquétipo defina uma URL arbitrária que indique isso. Como **o formato da URL não é controlado, perde-se, em portabilidade semântica ao se realizar tal procedimento**, uma vez que quem utilizar o mesmo arquétipo não terá mecanismos para saber o que a URL definia.

Outra questão envolvendo o serviço de terminologias diz respeito a suas interfaces, bastante simples e ineficientes para acesso a grandes terminologias. Por exemplo, não há formas, dentro do padrão, que permitam limitar o numero de elementos retornados em uma

consulta ao serviço de terminologias, algo essencial em sistemas reais, onde os resultados são frequentemente apresentados paginados.

Pode-se ressaltar que, a implementação Python do modelo de referência do padrão OpenEHR, apresenta-se, não apenas como uma contribuição de caráter acadêmico, mas também social, pois Python é uma linguagem *open source*. Acrescentando que o aumento das possibilidades de implementação associadas ao padrão OpenEHR, viabilizando-o nas diversas linguagens associadas às diferentes plataformas de programação existentes traz, em seu bojo, o estímulo à modelagem e compartilhamento de arquétipos em português, bem como a tradução para o português de arquétipos já existentes no repositório do CKM. Tal participação no desenvolvimento dos artefatos de conhecimento que, por definição, se caracterizam pelo forte apelo comunitário não pode ser negligenciado, sob o risco da indisponibilidade dos arquétipos necessários quando do desenvolvimento de *softwares* neste padrão no país.



## REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT ISO/TR 20514**: Informática em saúde - Registro eletrônico de saúde - Definição, escopo e contexto. Rio de Janeiro, 2008.

ARANCÓN, José; POLO, Luis; BERRUETA, Diego; LESAFFRE, François-Marie; DE ABAJO, Nicolás; CAMPOS, Antonio. Ontology-based Knowledge Management in the Steel Industry. In CARDOSO, J.; HEEP, M.; LYTRAS, M. D. (Eds.). **The Semantic Web: Real-World Applications from Industry (Semantic Web and Beyond)**. v. 6. New York: Springer, 2008.

ATALAG, K.; YANG, H. Y.; WARREN, J. On the Maintainability of openEHR Based Health Information Systems: an Evaluation Study in Endoscopy. In: 18th Annual Health Informatics Conference: Informing the Business of Healthcare. 2010, Melbourne. **Proceedings of 18th Annual Health Informatics Conference: Informing the Business of Healthcare**. Melbourne: Health Informatics Society of Australia, 2010.

BAKER, C. J. O.; CHEUNG, Kei-Hoi. **Semantic Web-Revolutionizing Knowledge Discovery in the Life Sciences**. New York: Springer, 2007.

BAX, M. P. **Introduction to markup languages**. Ci. Inf., Brasília, v. 30, n. 1, 2001. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652001000100005&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652001000100005&lng=en&nrm=iso)>. Acesso em: 10 abr. 2010.

BAX, M. P.. **DESIGN SCIENCE: FILOSOFIA DA PESQUISA EM CIÊNCIA DA INFORMAÇÃO E TECNOLOGIA**. In: XV Encontro Nacional de Pesquisa em Ciência da Informação (ENANCIB 2014), Belo Horizonte, 2014.

BEALE, T. Archetypes: Constraint-based domain models for future-proof information systems. In: Eleven OOPSLA Workshop on Behavioural Semantics: Serving the Customer, 2002, New York. **Proceedings of OOPSLA '02 Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**. Baclawski, K.; Kilov, H. (Eds). New York: ACM, 2002.

BEALE, T. **The OpenEHR Archetype Model**: Archetype Object Model. OpenEHR, 2008.

Disponível em: <[http://www.openehr.org/svn/specification/TAGS/Release1.0.1/publishing/architecture/aechetype\\_model.pdf](http://www.openehr.org/svn/specification/TAGS/Release1.0.1/publishing/architecture/aechetype_model.pdf)>. Acesso em: 10 nov. 2011.

BEALE, T.; HEARD, S. An Ontology-based Model of Clinical Information. In: MEDINFO, 2007, Brisbane. **Proceedings of the 12th World Congress on Health (Medical) Informatics (Studies in Health Technology and Informatics)**. K. Kuhn *et al.* (Eds). Brisbane: IOS Press, 2007.

BEALE, T.; HEARD, S. **OpenEHR architecture overview**. OpenEHR, 2008. Disponível em: <<http://www.openehr.org/svn/specification/TAGS/Release1.0.1/publishing/architecture/overview.pdf>>. Acesso em: 20 set. 2011.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR Reference Model: Data Types Information Model**. OpenEHR, 2007. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 30 jun. 2012.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR Reference Model: Support Information Model**. OpenEHR, 2008a. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 15 abr. 2012.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR Reference Model: Data Types Information Model**. OpenEHR, 2008b. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 30 abr. 2012.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR Reference Model: Data Structures Information Model**. OpenEHR, 2008c. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 29 abr. 2012.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR Reference Model: EHR Information Model**. OpenEHR, 2008d. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 30 abr. 2012.

BEALE, Thomas; HEARD, S.; KALRA, Deepak; LLOYD, David. **The OpenEHR**

**Reference Model:** Common Information Model. OpenEHR, 2008e. Disponível em: <[http://www-test.openehr.org/releases/trunk/architecture/rm/support\\_im.pdf](http://www-test.openehr.org/releases/trunk/architecture/rm/support_im.pdf)>. Acesso em: 20 abr. 2012.

BEAVER, K. **Healthcare Information Systems**. 2.ed. Florida: CRC Press LLC, 2003.

BENSON, Tim. **Principles of Health Interoperability HL7 and SNOMED**. London: Springer, 2010. p. 11-12.

BOIKO, B. **Content Management Bible**. 2. ed. Indianapolis: Wiley Publishing Inc, 2005.

BÖHM, G. M.; WEN, C. L.; SILVEIRA, P. P. Telemedicina: O Acesso à Distância aos Registros de Saúde. In: MASSAD *et al.* **O prontuário eletrônico do paciente na assistência, informação e conhecimento médico**. São Paulo: H. de F. Marin, 2003.

BOUKAR, M. M. Content Management System (CMS) Evaluation and Analysis. **Journal of Technical Science and Technologies**. v. 1. n. 1, 2012. Disponível em: <<http://journal.ibsu.edu.ge/index.php/jtst/article/view/240>>. Acesso em: 25 fev. de 2013.

BRACHMAN, R. J.; LEVESQUE, H. J. **Knowledge Representation and Reasoning**. San Francisco: Morgan Kaufmann Publishers, 2004.

BUI, Alex A.T.; Weinger ,Gregory S.; Barretta, Susan J.; Dionisio, John David N.; McCoy, J. Michael. **An XML Gateway to Patient Data for Medical Research Applications**. Los Angeles: University of California, 2001. Disponível em: <[www.mii.ucla.edu/papers/bui\\_metmbs\\_2001b.pdf](http://www.mii.ucla.edu/papers/bui_metmbs_2001b.pdf)>. Acesso em: 30 abr. de 2011.

CANNOY, S. D.; IYER, L. Semantic Web Standards and Ontologies in the Medical Sciences and Healthcare. In: TAN, J. (Org). **Medical Informatics: Concepts, methodologies, tools, and applications**. New York: Medical Information Science Reference, 2009. p 65-77.

CHEN, R.; KLEIN, G. The OpenEHR Java Reference Implementation Project. In: MEDINFO-2007, 2007, Brisbane. **Proceedings of the 12th World Congress on Health (Medical) Informatics (Studies in Health Technology and Informatics)**. K. Kuhn *et al.* (Eds). Brisbane: IOS Press, 2007. p. 58-62.

CHEN, R.; GARDE, Sebastian; BEALE, Thomas; NYSTRÖM, Mikael; KARLSSON, Daniel; KLEIN, Gunnar O.; AHLFELDT, Hans. An Archetype-based Testing Framework. In: MIE-2008, 2008, Göteborg. **Proceedings of MIE2008, The XXIst International Congress of the European Federation for Medical Informatics**. Andersen, S. K. *et al.* (Eds). Göteborg: IOS Press, 2008. p. 401-406.

CHEN, R. **Towards Interoperable and Knowledge-Based electronic Health Record Using Archetypes Methodology**. Tese (Doutorado em Informática Médica) - Departamento de Engenharia Biomedica, Universidade de Linköping, Linköping, 2009.

CHUN, W. J. **Core Python Programming**. 2. ed. Stoughton: Pearson Education, 2007.

CLARK, A. **Practical Plone 3: A Beginner's Guide to Building Powerful Websites**. 1. ed. Birmingham: Packt Publishing, 2009.

COOK, T. W. **Implementing OpenEHR** : Architecture Review Board. Disponível em: <<http://uff.academia.edu/TimothyCook/Talks>>. Acesso em: 12 mai. 2011.

COPIN, B. **Artificial Intelligence Illuminated**. 1. ed. Sudbury: Jones and Bartlett Publishers, 2004.

COSTA, M. F. E. **Dicionário de Termos Médicos**. Porto: Porto Editora, 2012.

DACONTA, M. C.; Obrst, L. J.; Smith, K. T. **The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management**. 1. ed. Indianapolis: Wiley Publishing Inc, 2003.

DELMONTE, Maurizio; MORO, Davide; NARDUZZO, Alice; FABRIZIO, Reale; MACKAY, Andy. **The Definitive Guide To Plone**. 2. ed. New York: Apress. 2009.

DICK, R. S.; STEEN, E. B.; DETMER, D. E. **The Computer Based Patient Record : An Essential Technology for Health Care**. Institute of Medicine, 1997, Washington: National Academic Press. Disponível em: <[http://books.nap.edu/openbook.php?record\\_id=5306&page=R1](http://books.nap.edu/openbook.php?record_id=5306&page=R1)>. Acesso em: 20 jul. de 2010.

EL-SAPPAGH, Shaker H.; EL-MASRI, Samir; RIAD, A. M.; ELMOGY, Mohammed. **Electronic Health Record Data Model Optimized for Knowledge Discovery**. International Journal of Computer Science Issues. v. 9. Issue 5. n. 1, 2012.

EICHELBERG, Marco; ADEN, Thomas; RIESMEIER, Jorg; DOGAC, Asuman; LALECI, Gokce B.. A Survey and Analysis of Electronic Healthcare Record Standards. **ACM Computing Surveys**, v. 37, Issue 4, p. 277-315, 2005.

FRADE, S.; MIRANDA FREIRE, S.; SUNDVALL, E.; PATRIARCA-ALMEIDA, J.H.; CRUZ-CORREIA, R.. Survey of OpenEHR Storage Implementations. In: 26th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2013). Porto, 2013. **Apresentação Oral**. Porto, 2013.

GABBRIELLI, M.; MARTINI, S. **Programming Languages: Principles and Paradigms**. Springer, 2010.

GARDE S; KNAUP P; HOVENGA EJS; HEARD S.. **Towards Semantic Interoperability for Electronic Health Records: Domain Knowledge Governance for openEHR Archetypes**. In: Methods of Information in Medicine. Issue 3. Schattauer, 2007. p.332-343.

GAŠEVIĆ, D.; DJURIĆ, D.; DEVEDŽIĆ, V. **Model Driven Architecture and Ontology Development**. Berlin: Springer, 2006.

GARETS, D.; DAVIS, M. **Electronic Medical Records vs. Electronic Health Records: Yes, There Is a Difference**. Chicago: HIMSS Analytics, 2006. Disponível em: <[https://www.himssanalytics.org/docs/WP\\_EMR\\_EHR.pdf](https://www.himssanalytics.org/docs/WP_EMR_EHR.pdf)>. Acesso em: 15 jul. 2011.

GÖK, M. **Introducing an openEHR-Based Electronic Health Record System in a Hospital Case Study, Emergency Department, Austin Health, Melbourne**. Tese (Doutorado em Informática Aplicada) - Centro de Informatica, Universidade de Göttingen, Göttingen, 2008.

GUARINO, N. Formal Ontology, Conceptual Analysis and Knowledge Representation. In: **International Journal of Human and Computer Studies**, v.43. Issue 5-6. Duluth: Academic Press, 1995. p. 625-640.

GUTIÉRREZ, P. P.; CARRASCO, L. Open EHR-Gen Framework Generador de sistemas normalizados de historia clínica electrónica basados en el estándar OpenEHR. In: Jornadas de Sistemas de Información en Salud del Hospital Italiano de Buenos Aires. 2013, Buenos Aires. **Oficina do Programa de Internacionalização do OpenEHR**. Buenos Aires, 2013. Disponível em: <<http://informatica-medica.blogspot.com.br/2013/12/talleres-de-openehr-en-hiba-2013.html>>. Acesso em: 15 jul. 2013.

GRUBER, T. R. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In: Principles of Knowledge Representation and Reasoning. 1993, São Francisco. **Proceedings of the Second International Conference**. São Francisco: Morgan Kaufmann Publishers Inc., 1993.

GRUBER, T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In: **International Journal of Human-Computer Studies**, v.43, Issue 5-6. Duluth: Academic Press, 1995. p. 907-928.

HAN, Y. **Digital Content Management: The Search for a Content Management System**. Library Hi Tech. v. 22. Issue 4. 2005. p. 355-365.

HAMILTON, B. R. **Electronic Health Records**. 2 ed. New York: Mac Graw Hill, 2011. p. 2-26.

HEVNER, A.; CHATTERJEE, S. **Design Research in Information Systems: Theory and Practice**. Integrated Series in Information Systems, v. 22. New York: Springer, 2010.

HIMSS. **Electronic Health Record**, 2010. Disponível em: <[http://www.himss.org/ASP/topics\\_ehr.asp](http://www.himss.org/ASP/topics_ehr.asp)>. Acesso em: 25 jun. 2011.

HL7. **What is HL7?** Health Level Seven, 2009. Disponível em: <[www.cplire.ru/alt/telemed/HL7/WhatIsHL7.doc](http://www.cplire.ru/alt/telemed/HL7/WhatIsHL7.doc)>. Acesso em: 20 maio. 2010.

ISO/DIS 18308. **TC 215 Health Informatics: Requirements for an Electronic Health Record Reference Architecture**. ISO Technical Specification 18308, 2009.

INGRAM, D.; SEVKET, S. A. **The Evolving Role of Open Source Software in Medicine and Health Services**. Technology Innovation Management Review, jan. 2013. p. 32-39.

Disponível em: <<http://timreview.ca/article/648>>. Acesso em: 12 abr. 2013.

KALRA, D. Barriers. Approaches and Research Priorities for Semantic Interoperability in Support of Clinical Care Delivery. In: **SemanticHealth Project IST 027328**. Bruxelas: Comissão Europeia, 2007. Disponível em: <[http://www.semantichealth.org/DELIVERABLES/SemanticHealth\\_D4\\_1\\_final.pdf](http://www.semantichealth.org/DELIVERABLES/SemanticHealth_D4_1_final.pdf)>. Acesso em: 20 abr. 2011.

KITSIOU, S.; MANTHOU, V; VLACHOPOULOU, M. Overview and Analysis of Electronic Health Record Standards. In: LAZAKIDOU, A. A.; SIASSIAKOS, K. M. **Handbook of research on distributed medical informatics and e-health**. Hershey: Medical information science reference, 2009. p.84-102.

KOBAYASHI, S.; TATSUKAWA, A. Ruby Implementation of the OpenEHR Specifications. **Journal of Advanced Computational Intelligence and Intelligent Informatics**. v.16. n.1, 2012. p.42-47.

LACASTA, J.; NOGUERAS-ISO, J.; ZARAZAGA-SORIA, F. J. **Terminological Ontologies: Design, Management and Practical Applications**. New York: Springer, 2010.

LALANDE, A. **Vocabulário Técnico e Crítico da Filosofia**. Martins Fontes Editora, 2003.

LEÃO, Beatriz de Faria; COSTA, Cláudio Giuliano Alves; MARTINS, Gerson Zafalon; D'AVILA, Roberto Luiz. **Manual de Requisitos de Segurança, Conteúdo e funcionalidades para Sistemas de Registro Eletrônico em Saúde (RES)**. São Paulo: SBIS, 2004. Disponível em: < <http://www.sbis.org.br/manual.htm>>. Acesso em: 15 maio 2010.

LEE, K. Ka-Yin; TANG, Wai-Choi; CHOI, Kup-Sze. Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage. **Computer Methods and Programs in Biomedicine**. v. 110. Elsevier, 2013. p. 99-109.

LESLIE, H. **OpenEHR: The Word's Record**. PULSE+IT, 2007. Disponível em: < [http://issuu.com/pulseitmagazine/docs/pulseit\\_november2007/50](http://issuu.com/pulseitmagazine/docs/pulseit_november2007/50) > Acesso em: 20 set. 2011.

LUSK, R. Update on the Electronic Medical Record. **Otolaryngologic Clinics of North America**. v. 35, p. 1223-1236. 2002.

MARCO, D. **Building and Managing the Meta Data Repository: A Full Lifecycle Guide**. New York: John Willey and Sons, 2000.

MARTÍNEZ-COSTA, Catalina; MENÁRGUEZ-TORTOSA, Marcos; FERNANDEZ-BREIS, Jesualdo Tomás; MALDONADO, José Alberto. A model-driven approach for representing clinical archetypes for Semantic Web environments. **Journal of Biomedical Informatics**. v. 42. Elsevier, 2009. p. 150-164.

MASSAD, E.; MARIN, H. de Fátima; AZEVEDO, R. S. de. **O Prontuário Eletrônico do Paciente na Assistência, Informação e Conhecimento Médico**. São Paulo, 2003. Disponível em <[http://www.lampada.uerj.br/lampada/ementas/aulas/info\\_med/Prontuario\\_livro.pdf](http://www.lampada.uerj.br/lampada/ementas/aulas/info_med/Prontuario_livro.pdf)>. Acesso em: 20 fev. 2010.

MEYER, B. **Object Oriented Software Construction**. 2. ed. Prentice Hall, 2000.

MOONEY, S. D.; BAENZIGER, P. H. Extensible open source content management systems and frameworks: a solution for many needs of a bioinformatics group. **Briefings in Bioinformatics**. v. 9. n.1. Oxford: Oxford University Press Journals, 2008. p. 69-74.

NADKARNI, P. **Metadata-driven Software Systems in Biomedicine** : Designing Systems that can adapt to Changing Knowledge. London: Springer-Verlag, 2011.

NARDON, F. B.; FRANÇA, T.; NAVES, H. Construção de Aplicações de Saúde Baseadas em Arquétipos. In: XI Congresso Brasileiro de Informática em Saúde, 2008, Campos de Jordão. **Anais do XI Congresso Brasileiro de Informática em Saúde**. São Paulo, 2008. Disponível em: < <http://www.sbis.org.br/cbis11/arquivos/947.pdf> >. Acesso em: 30 mar. 2010.

NARDON, F. B. N. **Utilizando XML para Representação de Informação em Saúde**. Unidade de Pesquisa e Desenvolvimento, Instituto do Coração do Hospital das Clínicas da Faculdade de Medicina da USP, São Paulo, 2002. Disponível em: <<http://www.tridedalo.com.br/fabiane/publications/XML-SBISNews.pdf>>. Acesso em: 30 set. 2011.

NEIRA, R. A. Q. ; NARDON, F. B. ; MOURA JR., L. A. ; LEAO, B. F. Como Incorporar Conhecimento aos Sistemas de Registro Eletrônico em Saúde?. In: XI Congresso Brasileiro de Informática em Saúde, 2008, Campos de Jordão. **Anais do XI Congresso Brasileiro de**



**Informática em Saúde.** São Paulo, 2008. Disponível em:  
<<http://www.tridedalo.com.br/fabiane/publications/XML-SBISNewsII.pdf>>. Acesso em: 27 abr. 2011.

NEWELL, A. **The Knowledge Level.** Artificial Intelligence, n. 18, Pittsburgh 1982. p. 87-127. Disponível em:  
<<http://repository.cmu.edu/cgi/viewcontent.cgi?article=2616&context=compsci>>. Acesso em: 20 out. 2011.

OpenEHR. **Origins of the openEHR.** 2002. Disponível em:  
<<http://www.openehr.org/about/origins.html>>. Acesso em: 25 out. 2010.

OpenEHR. **The openEHR Archetype Model: Archetype Object Model.** 2013. Disponível em : <<http://www-test.openehr.org/releases/trunk/architecture/am/aom1.5.pdf>>. Acesso em: 25 nov. de 2011.

PAUL, R.; SAYED, A.; HOQUE, L. Search Efficient Representation of Healthcare Data based on the HL7 RIM. **JOURNAL OF COMPUTERS.** v. 5, n. 12. Academy Publisher. 2010. p. 1818-1818. Disponível em:  
<<http://ojs.academpublisher.com/index.php/jcp/article/view/051218101818/2383>>. Acesso em: 15 jul. 2012.

RIST, R.; TERWILLIGER, R. **Object Oriented Programming in Eiffel.** 1. ed. Upper Saddle: Prentice Hall, 1995.

RUSSELL, S; NORVIG, P. **Inteligência Artificial.** 2. ed. Rio de Janeiro: Editora Campus, 2004.

SADALAGE, P. J.; FOWLER, M. **NoSQL Distilled : A Brief Guide to the Emerging World of Polyglot Persistence.** Upper Saddle River: Addison-Wesley, 2013.

SANTOS, M. R. **Sistema de registro eletrônico de saúde baseado na norma ISO 13606: aplicações na Secretaria de Estado de Saúde de Minas Gerais.** Tese (Doutorado em Ciência da Informação) - Escola da Ciência da Informação da Universidade Federal de Minas Gerais. Belo Horizonte, 2011.

SCHMIDT, Douglas; STAL, Michael; ROHNERT, Hans; BUSCHMANN, Frank.. **Pattern-Oriented Software Architecture** : A System of Patterns. v. 1. New York: John Wiley & Sons, 2000.

SEBESTA, R. W. **Concepts of Programming Languages**. 10. ed. Addison-Wesley, 2012.

SIMON, H. A. **The sciences of the artificial**. 3. ed. Cambridge: M.I.T. Press, 1969.

SHETH, A. P. Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In: Goodchild MF, Egenhofer MJ, Fegeas R, Kottman CA (eds). **Interoperating Geographic Information Systems**. Norwell: Kluwer; 1999. Disponível em < <http://lstdis.cs.uga.edu/library/download/S98-changing.pdf> >. Acesso em: 30 set. 2012.

SINHA, Pradeep; SUNDER, Gaur; BENDALE, Prashant; MANTRI, Manisha; DANDE, Atreya. **Electronic Health Record Standards, Coding Systems, Frameworks, and Infrastructures**. New Jersey: John Wiley & Sons, 2013.

SMITH, B.; CEUSTERS, W.; TEMMERMAN, R. Wüsteria. In: Studies in Health Technology and Informatics. 2005, Geneva. **Proceedings of Medical Informatics Europe 2005**. Geneva: Stud Health Technol Inform, 2005. p. 617-652.

SMULLYAN, Raymond M.; LOPARIC, Andrea; DE CAMPOS, M. A.; MAZAK, Rene Pierre; VICENTE, Luciano. **LÓGICA DE PRIMEIRA ORDEM**. 1.ed. São Paulo: Editora da UNESP, 2009.

SOWA, J. F.; **Knowledge Representation**: Logical, Philosophical, and Computational Foundations. Pacific Groove: Brooks/Cole, 2000.

STERLING, Leon; SHAPIRO, Ehud. **The Art Of Prolog: Advanced Programming Technics**. 2 ed. MIT Press. 1999.

SUH, Phil; ADDEY, Dave; THIEMECKE, David; ELLIS, James. **Content Management Systems**. Glasshaus, 2003.

SPETS, S. **Programming Drupal 7 Entities**. Birmingham-Mumbai: Packt Publishing, 2013.

TACHINARDI, H.; FURUIE, S. S. Arquivamento e Transmissão de Imagens Médicas. In: MASSAD *et al.* **O Prontuário eletrônico do paciente na assistência, informação e conhecimento médico**. São Paulo: H. de F. Marin, 2003. p.97-108.

TAN, J. **E-health care information systems: An Introduction for Students and Professionals**. San Francisco: Jossey-Bass, 2005.

VAISHNAVI, V.; KUECHLER, W. **Design Science Research in Information Systems**. Disponível em: <<http://www.desrist.org/design-research-in-information-systems/>>. Acesso em: 15 mar. 2014.

VAN BEMMEL, J. H.; MUSEN, M.A.. **Handbook of Medical Informatics**. The Netherlands: Springer-Verlag, 2000.

VELDE, R. Van de; DEGOULET, P. **Clinical Information Systems: a Component-Based Approach**. New York: Springer. 2003.

WATT, David A.; FINDLAY, William . **Programming Language Design Concepts**. Hoboken: John Willey and Sons. 2004.

WEITERSHAUSEN, P. von. **Web Component Development with Zope 3**. 3. ed. Dresden: Springer. 2008.

WETTER, T. Why is medical software so hard? In: **Informatik Forschung und Entwicklung**. v.22. Issue 3. Springer, 2008.

WIERINGA, R.J. Design Science as Nested Problem Solving. In: 4th International Conference on Design Science Research in Information Systems and Technology. 2009, Philadelphia. **Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology**. Philadelphia, 2009. p. 1-12.

## **ANEXOS**

**ANEXO A - PORTARIA N 2.073, DE 31  
DE AGOSTO DE 2011**

## PORTARIA Nº 2.073, DE 31 DE AGOSTO DE 2011

Legislações - GM

Sex, 02 de Setembro de 2011 00:00

## PORTARIA Nº 2.073, DE 31 DE AGOSTO DE 2011

Regulamenta o uso de padrões de interoperabilidade e informação em saúde para sistemas de informação em saúde no âmbito do Sistema Único de Saúde, nos níveis Municipal, Distrital, Estadual e Federal, e para os sistemas privados e do setor de saúde suplementar.

O MINISTRO DE ESTADO DA SAÚDE, no uso das atribuições que lhe conferem os incisos I e II do parágrafo único do art. 87 da Constituição, e

Considerando a [Lei nº 8.080](#), de 19 de setembro de 1990, que dispõe sobre as condições para a promoção, proteção e recuperação da saúde, a organização e o funcionamento dos serviços correspondentes;

Considerando a [Lei nº 8.159](#), de 8 de janeiro de 1991, que dispõe sobre a política nacional de arquivos públicos e privados;

Considerando o [Decreto nº 7.508](#), de 28 de junho de 2011, que regulamenta a Lei nº 8.080, de 19 de setembro de 1990;

Considerando a [Portaria nº 399/GM/MS](#), de 22 de fevereiro de 2006, que divulga o Pacto pela Saúde 2006 - Consolidação do SUS e aprova as diretrizes operacionais do referido Pacto;

Considerando a Portaria nº 2.072/GM/MS, de 31 de agosto de 2011, que redefine o Comitê de Informação e Informática em Saúde (CIINFO/MS) no âmbito do Ministério da Saúde, cuja atribuição é emitir deliberações, normas e padrões técnicos de interoperabilidade e intercâmbio de informações em conformidade com a política de informação e informática em saúde;

Considerando a necessidade de adotar medidas no campo da saúde que objetivem a melhoria e a modernização do seu sistema de gerenciamento de informações e dos preceitos da Política Nacional de Informação e Informática em Saúde (PNIIS), em conformidade com o

art. 47 da [Lei nº 8.080, de 1990](#), e deliberações das 11ª, 12ª e 13ª Conferências Nacionais de Saúde;

Considerando a racionalização e a interoperabilidade tecnológica dos serviços nos diferentes níveis da Federação para permitir o intercâmbio das informações e a agilização dos procedimentos;

Considerando que um efetivo e eficiente sistema de registro das ações e eventos de saúde contribui para o gerenciamento do Sistema Único de Saúde (SUS), garantindo ao cidadão o registro dos dados relativos à atenção à saúde, que lhe é garantida, num sistema

informatizado;

Considerando a necessidade de inovação e fortalecimento do sistema de informação e informática em saúde e do processo de consolidação da implantação do Cartão Nacional de Saúde (CNS);

Considerando que um efetivo e eficiente sistema de registro de atendimento em saúde contribui para a organização de uma rede de serviços regionalizada e hierarquizada para a gestão do SUS; e

Considerando a necessidade de garantir ao cidadão o registro dos dados relativos à atenção à saúde, resolve:

## CAPÍTULO I

## DAS DISPOSIÇÕES PRELIMINARES

Art. 1º Esta Portaria regulamenta o uso de padrões de informação em saúde e de interoperabilidade entre os sistemas de informação do SUS, nos níveis Municipal, Distrital, Estadual e Federal, e para os

sistemas privados e de saúde suplementar.

Parágrafo único. Os padrões de interoperabilidade e de informação em saúde são o conjunto mínimo de premissas, políticas e especificações técnicas que disciplinam o intercâmbio de informações entre os sistemas de saúde Municipais, Distrital, Estaduais e Federal, estabelecendo condições de interação com os entes federativos e a sociedade.

Art. 2º A definição dos padrões de informação em saúde e de interoperabilidade de informática em saúde tem como objetivos: I - definir a representação de conceitos a partir da utilização de ontologias, terminologias e classificações em saúde comuns, e modelos padronizados de representação da informação em saúde, criar e padronizar formatos e esquemas de codificação de dados, de forma a tornar célere o acesso a informações relevantes, fidedignas e oportunas sobre o usuário dos serviços de saúde;

II - promover a utilização de uma arquitetura da informação em saúde que contemple a representação de conceitos, conforme mencionado no inciso I, para permitir o compartilhamento de informações em saúde e a cooperação de todos os profissionais, estabelecimentos de saúde e demais envolvidos na atenção à saúde prestada ao usuário do SUS, em meio seguro e com respeito ao direito de privacidade;

III - contribuir para melhorar a qualidade e eficiência do Sistema Único de Saúde e da saúde da população em geral;

IV - fundamentar a definição de uma arquitetura de informação nacional, independente de plataforma tecnológica de software ou hardware, para orientar o desenvolvimento de sistemas de informação em saúde;

V - permitir interoperabilidade funcional, sintática e semântica entre os diversos sistemas de informações em saúde, existentes e futuros;

VI - estruturar as informações referentes a identificação do usuário do SUS, o profissional e o estabelecimento de saúde responsáveis pela realização do atendimento;

VII - estruturar as informações referentes aos atendimentos prestados aos usuários do SUS visando à implementação de um Registro Eletrônico de Saúde (RES) nacional e longitudinal; e

VIII - definir o conjunto de mensagens e serviços a serem utilizados na comunicação entre os sistemas de informação em saúde;

## CAPÍTULO II

### DA DEFINIÇÃO E ADOÇÃO DOS PADRÕES DE INTEROPERABILIDADE DE INFORMAÇÕES DE SAÚDE

Art. 3º O Ministério da Saúde estabelecerá uma arquitetura de conceitos em saúde, que identificará os detalhes e os principais atributos dos serviços, seus componentes, atividades e políticas necessárias.

Parágrafo único. A arquitetura em saúde será a fundação para a definição do conjunto de especificações técnicas e padrões a serem utilizados na troca de informação sobre eventos de saúde dos usuários do SUS pelos sistemas de saúde locais, regionais e nacionais, públicos e privados.

Art. 4º Os padrões de interoperabilidade constarão do Catálogo de Padrões de Interoperabilidade de Informações de Sistemas de Saúde (CPIISS), publicado pelo Departamento de Informática do SUS (DATASUS/SGEP/MS), disponível para a sociedade em geral, encontrando-se a primeira versão nos termos do Anexo a esta Portaria.

§ 1º O CPIISS é constituído de especificações e padrões em uso, aprovados pelo Comitê de Informação e Informática em Saúde (CIINFO/MS) e pactuados na Comissão Intergestores Tripartite (CIT).

§ 2º O CPIISS conterá links para as organizações que produziram os padrões adotados, incluindo os padrões de jure e os de fato.

§ 3º O CPIISS será atualizado regularmente, de acordo com o processo de trabalho do CIINFO/MS, e todas as alterações serão enumeradas em versões acordadas após negociações na CIT.

§ 4º Os padrões publicados no CPIISS conterão um conjunto de metadados que seguirão o formato

definido pelo Padrão de Metadados do Governo Eletrônico Brasileiro (E-PMG).

Art. 5º Serão adotados padrões de interoperabilidade abertos, sem custo de royalties.

Parágrafo único. Quando não houver possibilidade técnica ou disponibilidade no mercado para adoção de padrões abertos, o CPIISS adotará os padrões apropriados aos objetivos estabelecidos nesta Portaria, levando em consideração os benefícios a seus usuários.

Art. 6º O processo de definição e adoção de padrões de interoperabilidade deve estar alinhado com o Guia de Boas Práticas e Regulamentação Técnica, definido pelo Conselho Nacional de Metrologia, Normalização e Qualidade Industrial (CONMETRO) e elaborado pelo Comitê Brasileiro de Regulamentação (CBR).

Art. 7º Os entes federativos que decidirem não utilizar os padrões de interoperabilidade de que trata esta Portaria deverão utilizar mensagens formatadas em padrão eXtensible Markup Language (XML) para troca de informações, de forma a atender aos XML schemas definidos pelo Ministério da Saúde e respectivas definições dos respectivos serviços -Web Service Definition Language (WSDL), quando for o caso.

Parágrafo único Cabe ao Ministério da Saúde, por meio do DATASUS/SGEP/MS, definir o padrão de importação e exportação baseado na tecnologia de serviços Web, com publicação dos schemas e respectivas WSDL.

### CAPÍTULO III

#### DA OPERACIONALIZAÇÃO E IMPLEMENTAÇÃO DOS PADRÕES DE INFORMAÇÃO EM SAÚDE E DE INTEROPERABILIDADE

Art. 8º A implementação dos usos dos padrões de informação em saúde e de interoperabilidade será coordenada pelo Grupo de Trabalho de Gestão da Câmara Técnica da CIT, ao qual caberá:

I - definir os sistemas a serem padronizados, com prioridade para os sistemas de base nacional vinculados à atenção primária à saúde; e

II - mapear mensagens a serem trocadas, indicando o conjunto de ontologias, terminologias e classificações em saúde aplicáveis.

Art. 9º Para implementar a utilização dos padrões de interoperabilidade, caberá ao Ministério da Saúde:

I - prover capacitação, qualificação e educação permanente dos profissionais envolvidos no uso e na implementação dos padrões de interoperabilidade;

II - garantir aos entes federados a disponibilização de todos os dados transmitidos, consolidados ou em sua composição plena; e III - prover plataforma de interoperabilidade para troca de informações entre os sistemas do SUS.

### CAPÍTULO IV

#### DO FINANCIAMENTO

Art. 10. O Ministério da Saúde ficará responsável pelos recursos financeiros necessários à efetivação da:

I - utilização dos padrões de interoperabilidade e informação em saúde estabelecidos nos termos desta Portaria, seja para subscrição, associação ou licenciamento, sendo a liberação de uso estendida a Estados, Distrito Federal e Municípios;

II - tradução de termos, nomenclaturas e vocabulários, bem como para a inserção de novos que



sejam imprescindíveis para atender às exigências do SUS, estendida sua utilização a Estados, Distrito Federal e Municípios; e

III - manutenção do arcabouço dos padrões de interoperabilidade e informação em saúde estabelecidos nos termos desta Portaria.

Art. 11. Os custos relacionados à adequação de sistemas de informação para uso dos padrões de interoperabilidade e informação em saúde serão de responsabilidade dos proprietários dos respectivos sistemas.

§ 1º Os Estados, o Distrito Federal e os Municípios arcarão com todas as despesas para adequação de seus sistemas próprios.

§ 2º O Ministério da Saúde arcará com as despesas para adequação de seus sistemas de informação.

Art. 12. Esta Portaria entra em vigor na data de sua publicação.

ALEXANDRE ROCHA SANTOS PADILHA

ANEXO

CAPÍTULO I

CATÁLOGO DE SERVIÇOS

1. Para a interoperabilidade entre os sistemas dos SUS será utilizada a tecnologia Web Service, no padrão SOAP 1.1 (Simple Object Access Protocol) ou superior.

2. Para a garantia de segurança e integridade de informações será adotado o padrão WS-Security para criptografia e assinatura digital das informações.

3. Os Web Services são identificados por um URI (Uniform Resource Identifier) e são descritos e definidos usando WSDL (Web Service Description Language).

CAPÍTULO II

CATÁLOGO DE PADRÕES DE INFORMAÇÃO

4. Os padrões são definidos em nível lógico (negócios) e não físico de arquivamento de banco de dados. Estes padrões não documentam propriedades de exibição. Os sistemas legados podem ter suas respostas, para integração e interoperação, encapsuladas em padrões XML aderentes aos padrões do Catálogo, de forma que, mesmo sem obedecer internamente ao padrão catalogado, possam comunicarse fazendo uso dele, por meio de XML Schemas

4.1. Para a definição do Registro Eletrônico em Saúde (RES) será utilizado o modelo de referência OpenEHR, disponível em <http://www.openehr.org/home.html>.

4.2. Para estabelecer a interoperabilidade entre sistemas, com vistas à integração dos resultados e solicitações de exames, será utilizado o padrão HL7 - Health Level 7.

4.3. Para codificação de termos clínicos e mapeamento das terminologias nacionais e internacionais em uso no país, visando suportar a interoperabilidade semântica entre os sistemas, será utilizada a terminologia SNOMED-CT, disponível em <http://www.ihtsdo.org/snomed-ct/>.

4.4. Para a interoperabilidade com sistemas de saúde suplementar serão utilizados os padrões TISS (Troca de Informações em Saúde Suplementar).

4.5. Para a definição da arquitetura do documento clínico será utilizado o padrão HL7 CDA.

4.6. Para a representação da informação relativa a exames de imagem será utilizado o padrão DICOM.

4.7. Para a codificação de exames laboratoriais será utilizado o padrão LOINC (Logical Observation Identifiers Names and Codes).

4.8. Para a codificação de dados de identificação das etiquetas de produtos relativos ao sangue humano, de células, tecidos e produtos de órgãos, será utilizada a norma ISBT 128.

4.9. Para a interoperabilidade de modelos de conhecimento, incluindo arquétipos, templates e metodologia de gestão, será utilizado o padrão ISO 13606-2.

4.10. Para o cruzamento de identificadores de pacientes de diferentes sistemas de informação, será utilizada a especificação de integração IHE-PIX (Patient Identifier Cross-Referencing).

4.11. Outras classificações que serão utilizadas para suporte à interoperabilidade dos sistemas de saúde: CID, CIAP-2 (Atenção primária de saúde), TUSS e CBHPM (Classificação brasileira hierarquizada de procedimentos médicos) e tabela de procedimentos do SUS.

**ANEXO B - MODELOS DE INFORMAÇÃO**  
**OpenEHR**

### **B1) A Classe BASIC\_TYPES**

Diferente de linguagens como C, C++ ou Java que são fortemente tipadas (ou de tipagem estática), ou seja, os tipos das variáveis devem ser declarados no código, obrigando que estas sejam sempre declaradas antecipadamente no código do programa, a linguagem de programação Python é dinamicamente tipada. Numa linguagem dinamicamente tipada a determinação do tipo de uma variável é feita em tempo de execução e não devido a declarações presentes no código. Conforme salienta Chun (2007): Python é dinamicamente tipada, significando que nenhuma pré-declaração de uma variável ou do seu tipo é necessária. O tipo (e valor) é inicializado na hora em que é usado.

Apesar dessa característica, buscando maior clareza dentro de uma boa prática de programação, optou-se por definir os tipos básicos mais simples a serem utilizados na construção das diversas classes. Isso foi feito na classe BASIC\_TYPES que define tipos como *String*, inteiro, flutuante, booleano e o dia inicial de uma data (para uso na instanciação das classes definidoras de datas).

### **B2) Modelo de Informação *Support***

O modelo de informação *Support* (Figura 41) abrange os tipos a serem utilizados nos outros modelos de informação. O modelo também descreve um pseudo pacote de tipos denominado *assumed\_types*, descrevendo os tipos que a especificação assume existir para que haja a possibilidade de uma implementação num *framework* tecnológico envolvendo linguagem de programação, base de dados, etc.

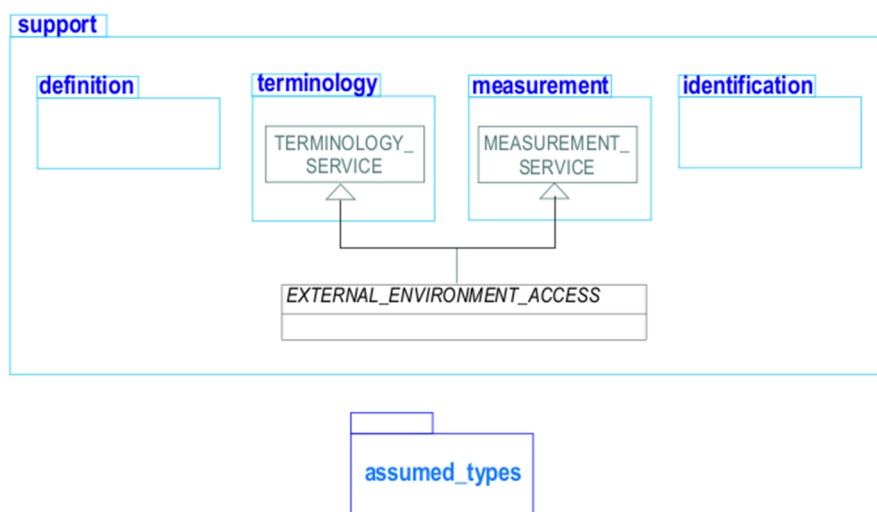


Figura 41 - O Modelo de Informação *Support*.  
Fonte: Beale *et al.* (2008a)

Os demais pacotes definem a semântica das constantes, acesso a terminologias, unidades de medida científicas e conversão de informações.

## Pacote Identification

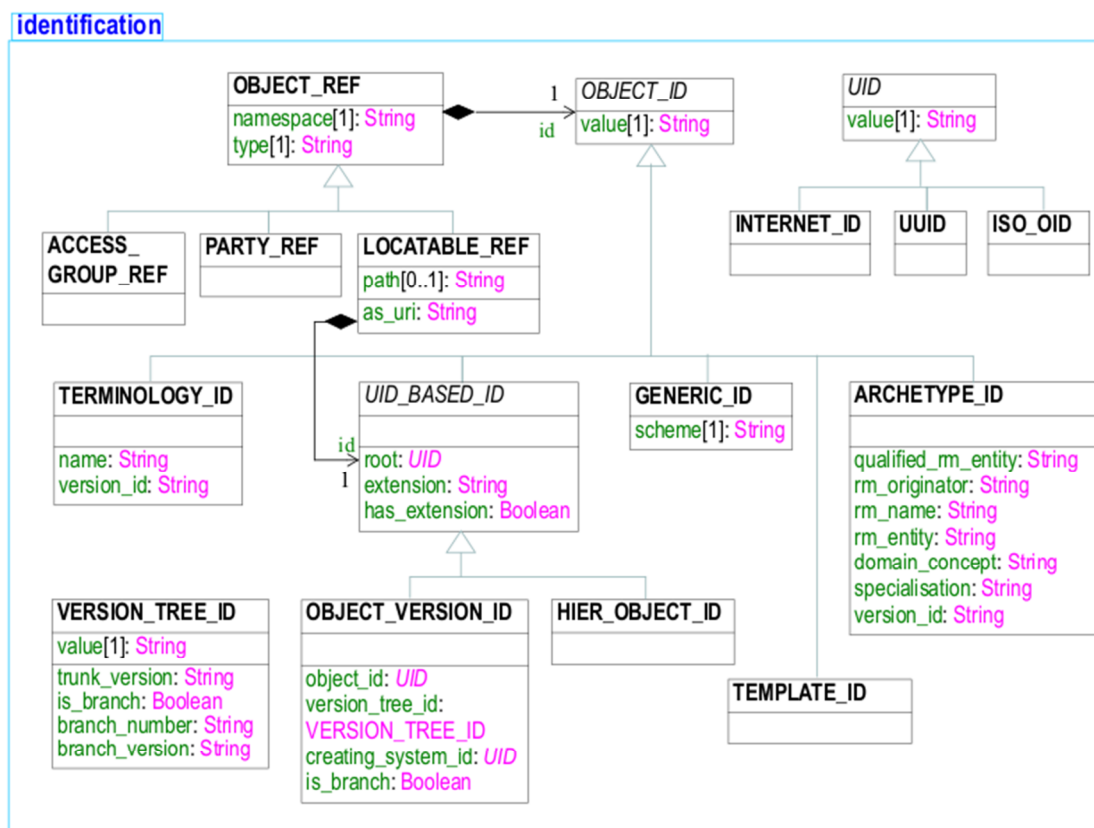


Figura 42 - O Pacote OpenEHR RM Identification  
Fonte: Beale *et al.* (2008a)

A identificação de uma entidade, seja no mundo real seja num sistema de informação, configura desafio não trivial. Segundo Beale (2008) o necessário para a identificação entre sistemas num ambiente de informação em saúde inclui:

- Identificadores do mundo real como números de segurança, ids diversos, etc. necessitam ser registrados.
- Identificadores para entidades informacionais que representam entidades ou processos do mundo real, necessitam ser únicos.
- Deve ser possível determinar se dois identificadores que se referem a entidades de informação representam as mesmas entidades do mundo real, mesmo se as instâncias das entidades de informação forem mantidas em sistemas diferentes.
- Versões ou modificações em entidades informacionais que possuam links a entidades do mundo real devem ser acomodadas de duas maneiras:

- Deve ser possível dizer se dois identificadores se referem a versões distintas da mesma entidade informacional na mesma árvore de versão.
- Não deve ser possível confundir versões de entidades de informação mantidas em múltiplos sistemas com o propósito de representar a mesma entidade do mundo real.

Deve ser possível para uma entidade em um sistema ou serviço (como um RES) se referir a uma outra entidade em outro sistema ou serviço de modo que:

- O alvo da referência é fácil localizar dentro do ambiente compartilhado.
- A referência é válida sem levar em conta a arquitetura física dos servidores e aplicações.

### **Identificação de Entidades do Mundo Real (EMR)**

Entidades do mundo real, como pessoas, carros, apontamentos e faturas podem ser designadas por identificadores. Apesar de muitos desses identificadores serem designados para serem únicos, frequentemente não o são, devido a erros em entradas de dados, equívocos em seu *design* (ids muito pequenos ou que incorporam características não únicas às entidades identificadas), erros de processo como pontos de emissão de ids não sincronizados; roubo de id via roubo de documentos ou ação de *hackers*. Conforme Beale, Heard, Kalra e Lloyd (2008d), em geral, enquanto alguns identificadores do mundo real (IMR) são “quase únicos”, nenhum pode ser garantido como sendo. Tal ocorre no caso de se dois IMR são iguais, estes referem-se à mesma EMR, o que, frequentemente, segundo os mesmos autores, não é o caso. Logo, para propósitos práticos, IMRs não podem ser considerados computacionalmente seguros.

### **Identificação de Entidades de Informação (EI)**

Assim como sistemas de informação são usados para registrar fatos sobre EMR's, a situação pode se tornar mais complexa devido à natureza intangível da informação. Em particular:

- O mesmo EMR pode ser representado simultaneamente em mais de um sistema (“multiplicidade espacial”).
- O mesmo EMR pode ser representado por mais de uma versão da mesma EI num

sistema (“multiplicidade temporal”).

À primeira vista, pode parecer que se pode ter apenas EI’s “puras”, ou seja, IE’s que não se referem a EMR como documentos online, softwares, etc. Porém, conforme ressaltam Beale, Heard, Kalra e Lloyd (2008d) na especificação deste modelo informacional, sempre se pode ter uma versão “autoritária” (ou imposta) para uma entidade dada. Tais entidades podem ser melhor compreendidas como “EMR virtuais”. Assim, pode-se dizer que múltiplas EI’s podem se referir a qualquer EMR dada.

A razão subjacente para a existência de tal multiplicidade de EI’s é que as dimensões tempo e espaço, em sistemas computacionais não são contínuas, mas discretas, pois cada “entidade” é, em última instância, um instantâneo de um número limitado de valores de atributos de uma EMR, num dado ponto no tempo, num sistema em particular. Se identificadores são atribuídos a EI’s sem que seja feita referência a versões ou duplicações, então, **nenhuma asserção pode ser feita** sobre a EMR quando tais EI’s forem comparadas.

### **Identificação de Versões**

A noção de versão aplicada a EI’s refere-se a instâncias diferentes de um conteúdo, cada uma representando um instantâneo de uma dada entidade. Como tais instâncias são armazenadas e gerenciadas dentro de um dado sistema de informação, torna-se necessário uma forma explícita de identificação de versões.

Os requisitos para tal podem ser resumidos como:

- Deve ser possível distinguir duas versões de uma mesma entidade, ou seja saber através dos identificadores se são a mesma ou diferentes versões de uma mesma coisa.
- Deve ser possível distinguir duas versões de uma mesma entidade em dois sistemas distintos.
- Deve ser possível traçar o relacionamento entre os itens numa linha progressiva de versões através dos seus respectivos identificadores.

A situação de entidades informacionais não conectadas diretamente por referências diretas, em um ambiente distribuído, apresenta dois requisitos, a seguir:



- A separação de objetos em um ambiente computacional distribuído não compromete a semântica do modelo.
- Diferentes tipos de informação podem ser gerenciadas de modo relativamente independente, por exemplo, registros de saúde e informações demográficas podem ser gerenciadas por diferentes grupos em uma organização ou comunidade.

### **Proposta do Padrão OpenEHR para o *Design* dos Identificadores**

No pacote *Identification*, modela-se apenas identificadores informacionais, ou seja, identificadores transparentes assim entendidos pelo OpenEHR ou sistemas computacionais relacionados. EMR's como números de identidade são modeladas utilizando o tipo de dado DV\_IDENTIFIER.

Uma decisão chave foi escolher uma representação via *string* para todos os identificadores, com subpartes acessíveis através de funções apropriadas que simplesmente parseiam um *string*. Isso garante que o dado que representa os identificadores (ex: XML) seja tão pequeno quanto possível.

### **Identificadores Primitivos**

O tipo abstrato UID e seus subtipos correspondem a identificadores primitivos, o que implica dizer que não possuem estruturas internas (*substrings*). Os três subtipos são: UUID, ISO\_OID e INTERNET\_ID. Normalmente são utilizados como partes de outros identificadores. Uma consequência da abordagem utilizada nessas classes é que definir um atributo do tipo UID via um string, desserializado de um XML ou outra forma textual, torna-o pronto para ser lido por uma base de dados.

### **Identificadores Compostos**

O tipo OBJECT\_ID e sua hierarquia de subtipos define todos os tipos de identificadores utilizados num sistema OpenEHR. Muitos desses identificadores possuem uma estrutura em várias partes, sendo que algumas são legíveis para os seres humanos. Estes tipos de identificadores podem ser divididos, semanticamente, em dois grupos: os definidos pelo próprio OpenEHR e aqueles definidos por organizações externas. Identificadores definidos pelo tipo HIER\_OBJECT\_ID são utilizados tanto pelo OpenEHR como por outras

organizações.

### **Identificadores do Tipo UID-based**

O tipo abstrato UID\_BASED\_ID bem como seus dois subtipos HIER\_OBJECT\_ID e OBJECT\_VERSION\_ID proveem respectivamente, UID-based identificadores para itens versionados e não versionados.

### **Identificadores de Arquétipos**

O subtipo ARCHETYPE\_ID define um identificador multiaxial para arquétipos, significando que cada instância do identificador denota um arquétipo simples dentro de um espaço multidimensional. Este espaço multidimensional seria composto pelos seguintes eixos:

- modelo de referência da entidade, definido como:
  - nome do emissor do modelo.
  - nome do modelo (pode haver mais de um do mesmo emissor).
  - nome do conceito modelado, ou seja, o nome da classe
- conceito do domínio.
- versão.

As três seções são delimitadas pelo caractere “.”, enquanto as partes da primeira seção são delimitadas pelo caractere “-”. O princípio que rege a construção dos identificadores de arquétipos é que todas as suas partes são imutáveis. Portanto, a inclusão do número da versão no identificador significa que, existindo duas versões de um dado arquétipo, na verdade tem-se dois arquétipos distintos.

Exemplos de identificadores podem ser:

- openEHR-EHR-SECTION.physical\_examination.v2
- openEHR-EHR-SECTION.physical\_examination-prenatal.v1
- H17-RIM-Act.progress\_note.v1
- openEHR-EHR-OBSERVATION.progress\_note-naturopathy.v2

### **Identificador de *Templates***

O identificador de templates é similar ao identificador de arquétipos. Isso provê um identificador multiaxial na forma UID-style. Na versão atual não se definiu uma estrutura exata para isso, sendo que a proposta corrente é dada como:

- Nomes reversos das organizações autoras.
- Identificador da classe do modelo de referência do qual está sendo feito o *template*.
- Nome do *template*.
- Identificador de versão na forma de “vn” onde n é um identificador numérico da versão.

Exemplo de um identificador pode ser visto a seguir:

- uk.nhs.cfh:openehr-EHR-COMPOSITION.admission\_ed.v5

### **Identificadores de Terminologias**

O subtipo TERMINOLOGY\_ID define um único e global identificador para terminologias (maiores detalhes na análise do pacote *Terminology*).

Exemplos de identificadores de terminologias podem ser:

- “SNOMED-CT”
- “ICD9(1999)”

## Pacote Terminology

[terminology](#)

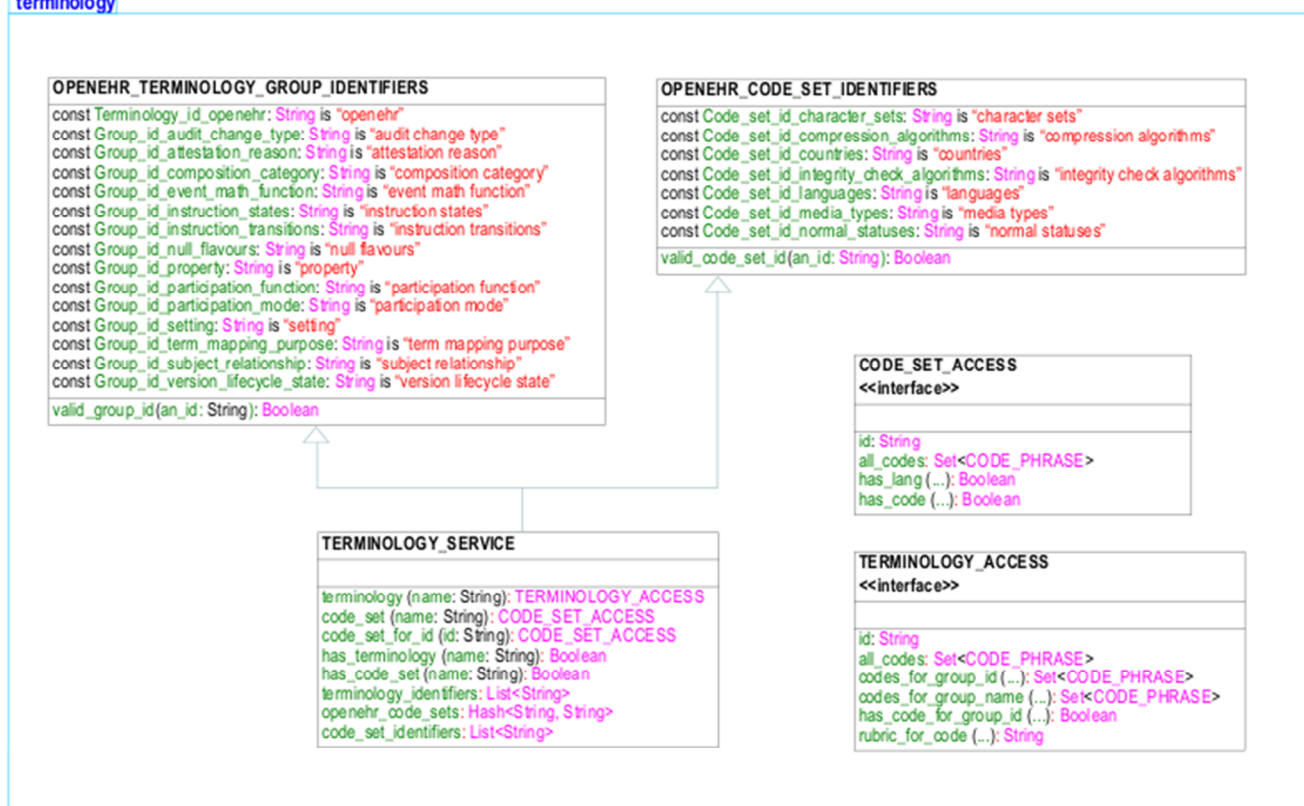


Figura 43 - O Pacote OpenEHR RM *Terminology*  
 Fonte: Beale *et al.* (2008a)

Uma interface de serviço de terminologia simples, definido conforme a Figura 43 possibilita ao padrão OpenEHR utilizar, referenciando diretamente do modelo de referência, conjuntos de códigos e terminologias. Dois tipos de entidades codificadas são distinguidas no OpenEHR. O primeiro definido como conjunto de códigos (“*code sets*”), que é o tipo de terminologia cujo código é padrão por si mesmo, como a ISO\_639-1, como por exemplo para atributos cujo valor é o código de uma linguagem. Tais padrões são referidos por constantes internas definidas na classe OPNEHR\_CODE\_SET\_IDENTIFIERS. O segundo são as terminologias, acessadas através das funções da classe TERMINOLOGY\_SERVICE, que possui argumentos como “openehr”, “centc251”, SNOMED-CT, etc.

## Identificadores

No OpenEHR, o identificador de uma terminologia ou conjunto de códigos é encontrado no atributo `terminology_id` da classe `CODE_PHRASE` (pacote *text*, do modelo de informação *Data Types*). Identificadores de conjuntos de códigos (por exemplo: “languages”) usados no OpenEHR são definidos na classe `OPENEHR_CODE_SET_IDENTIFIERS`. Identificadores externos assumidos como o “ISO\_639-1” para conjuntos de códigos usados pelo modelo de referência do OpenEHR são definidos no documento *openEHR Terminology*<sup>38</sup>

## Pacote Measurement

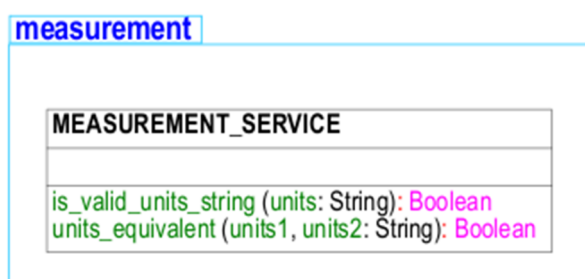


Figura 44 - O Pacote OpenEHR RM *Measurement*  
Fonte: Beale *et al.* (2008a)

O pacote *measurement* (Figura 45) define uma semântica mínima relativa a medições quantitativas, unidades, e conversões, habilitando o pacote *quantity* (Modelo de informação *Data Types*) a expressá-las corretamente. Da mesma forma que a definida no pacote *terminology*, um serviço simples de interface é proposto, provendo funções para esse fim. As definições de medições e unidades podem vir de diversas fontes, como: CEN ENV 12435<sup>39</sup>, UCUM<sup>40</sup>, etc.

<sup>38</sup>OpenEHR *Terminology*, endereço para acesso ao documento:

<<http://www.openehr.org/releases/1.0.2/architecture/terminology.pdf>>

<sup>39</sup> Medical Informatics - Expression of results of measurements in health sciences: < <http://www.cenc251.org>>

<sup>40</sup> The Unified Code for Units of Measure (UCUM), desenvolvido por Gunther Schadow e

<sup>40</sup>Clement J. McDonald do The Regenstrief Institute. Disponível em HL7v3 *ballot materials*:

<<http://www.hl7.org>>

## Pacote Definition

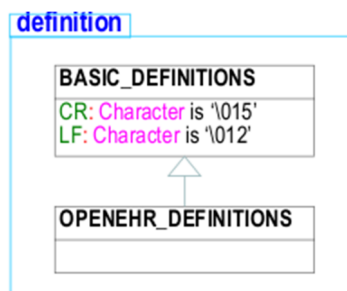


Figura 46 - O Pacote OpenEHR RM Definition  
Fonte: Beale *et al.* (2008a)

O pacote *definition* (Figura 47), define símbolos utilizados pelos modelos OpenEHR. Atualmente, somente um pequeno número é definido.

### B3) Modelo de Informação Data Types

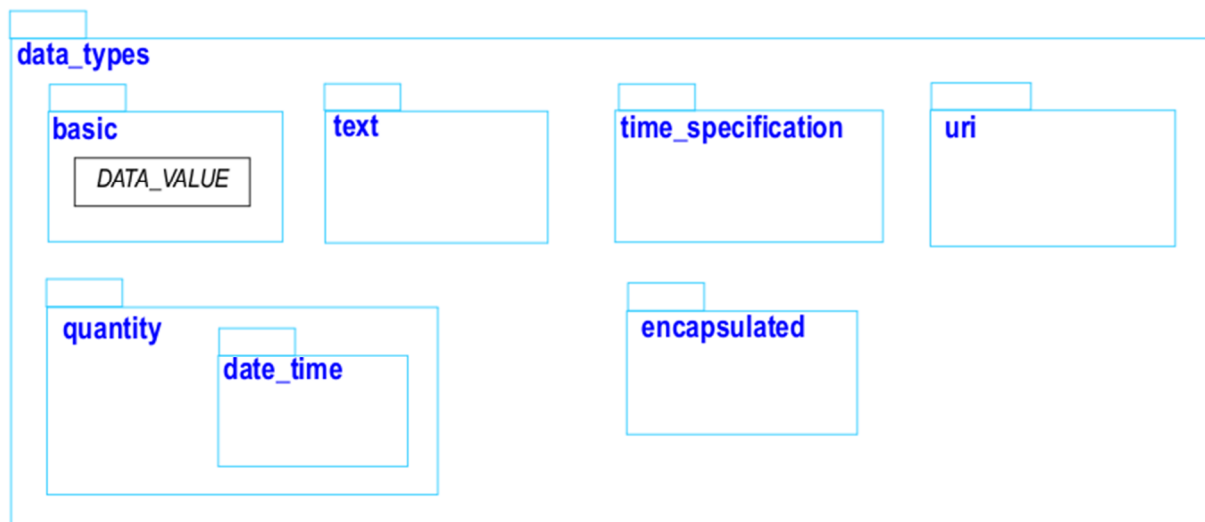


Figura 48 - O Modelo de Informação OpenEHR Data Types  
Fonte: Beale *et al.* (2008b)

Este modelo de informação (Figura 49) compõe-se de um conjunto de tipos de dados que serão utilizados por todos os outros modelos de informação, viabilizando a expressão de informações médicas que se utilizam de tipos como:

- Tipos básicos: booleanos, variáveis de estado.
- Texto: texto plano, texto codificado, parágrafos.
- Quantidades: qualquer tipo ordenado, incluindo valores ordinais, que representam

valores ordenados simbólicos, como “+”, “++”, “+++”, quantidades mensuráveis com valor e unidades, entre outras.

- Datas/tempo: datas e tempo.
- Dados encapsulados: multimídia, conteúdos parseáveis.

Os tipos de dados descritos aqui tem seus nomes iniciados com o prefixo “DV\_”, e possuem a classe DATA\_VALUE como superclasse e dois tipos de uso nos modelos de referência. Como “data values” em estruturas de dados onde são utilizadas instâncias da classe DATA\_VALUE ou subtipos que podem ser atributos em outras classes, como *date/times*, *coded terms*, etc.

### Pacote *Basic*

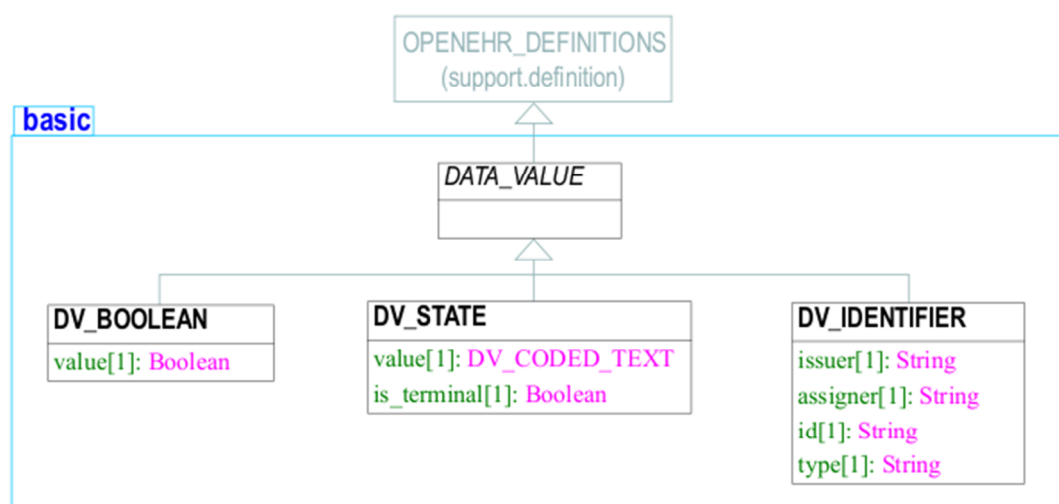


Figura 50 - O Pacote OpenEHR Basic  
Fonte: Beale *et al.* (2008b)

Contém tipos (Figura 51) para lidar com os conceitos de estados booleanos (bivalência) ou DV\_BOOLEAN, um estado numa máquina de estado (por exemplo, numa máquina de estado que modela o ciclo de vida da administração de um dado medicamento) ou DV\_STATE e identificadores para entidades do “mundo real” (como nomes e endereços) ou DV\_IDENTIFIER.

## Pacote *Text*

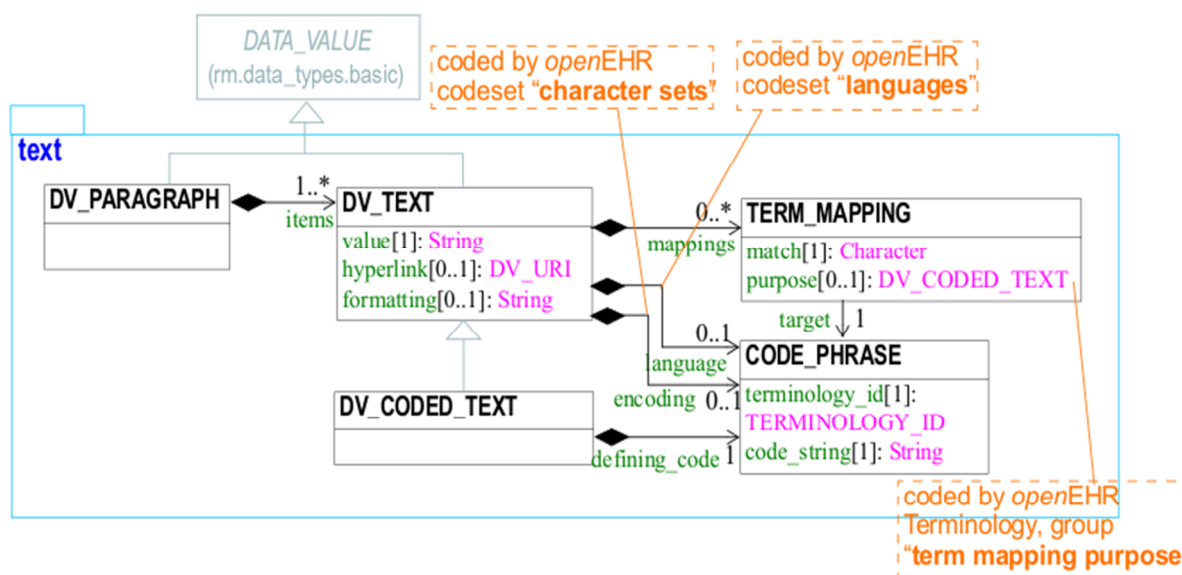


Figura 52 - O Pacote OpenEHR Text  
Fonte: Beale *et al.* (2008b)

O pacote *Text* (Figura 53), contém as classes que possibilitam representar todas as possíveis entradas de dados textuais no registro de saúde, como texto plano, texto codificado e texto narrativo.

Parte-se do princípio de que todos os casos clínicos podem ser registrados tão exatamente quanto necessário, estando ou não disponíveis ao clínico/software servidores de terminologias. Ou seja, registros com alto grau de generalidade como “frio”, “quente” ou muito específicos como “infecção pela bactéria *Staphylococcus aureus*” devem ser possíveis mesmo sem o acesso a terminologias externas.

O tipo **DV\_TEXT** pode ser utilizado quando o uso de um item de texto codificado ou não codificado é permitido, enquanto o tipo **DV\_CODED\_TEXT** deve ser utilizado sempre que um item de texto pode ser codificado.

O tipo **DV\_PARAGRAPH** permite que se construa textos de maior extensão utilizando-se instâncias das classes **DV\_TEXT** e **DV\_CODED\_TEXT**, conforme mostra a Figura 54:



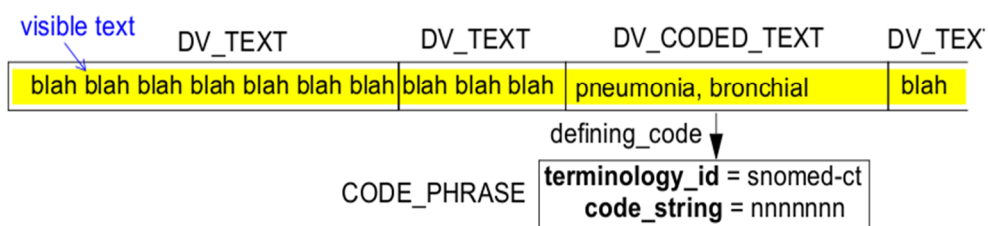


Figura 55 - OpenEHR Paragraph Elementos  
Fonte: Beale *et al.* (2008b)

O que gera um parágrafo com a aparência visual conforme a Figura 50:

XXXXX xxxx .xxx XXXXXXXX XX XX XXXXXXXX XXX .xxxxxxxxx XXXX  
 XXXXXXXXXXXX xxxxx XXX XXXXXXX XXXXXXXXXXXX .xxxxxx .xxxxx XXXXX XXXX  
 XXXXXX a XXXXXXXX XXXX XXXXXX .xxxxx .xxxxxxxxx XXXXXX X XXX

Figura 56 - OpenEHR Paragraph Aparência Final  
Fonte: Beale, Heard, Kalra e Lloyd (2008b)

No caso do acesso a terminologias, assume-se que o serviço (de terminologias) provê uma interface completa à terminologia que se esteja acessando e a classe **DV\_CODED\_TEXT** reflete tal assunção. Portanto, parte-se da ideia de que o serviço de terminologia disponibiliza apenas uma “chave” que se refere a uma entidade léxica que pode vir a ser um termo simples ou uma frase codificada.

Textos narrativos, por sua vez, são utilizados em situações como:

- No registro de respostas subjetivas ou imprecisas de pacientes, particularmente quantidades ou datas não consideradas como precisas o bastante para serem representadas por tipos de dados como *date/time* ou tipos do pacote *quantity*.
- Registro de declarações narradas pelo paciente, como observações de cunho visual.
- Registro de conclusões e prognósticos
- Registro de valores que, normalmente seriam codificados, mas não podem sê-lo pois não há serviço de terminologia disponível.

## Pacote Quantity

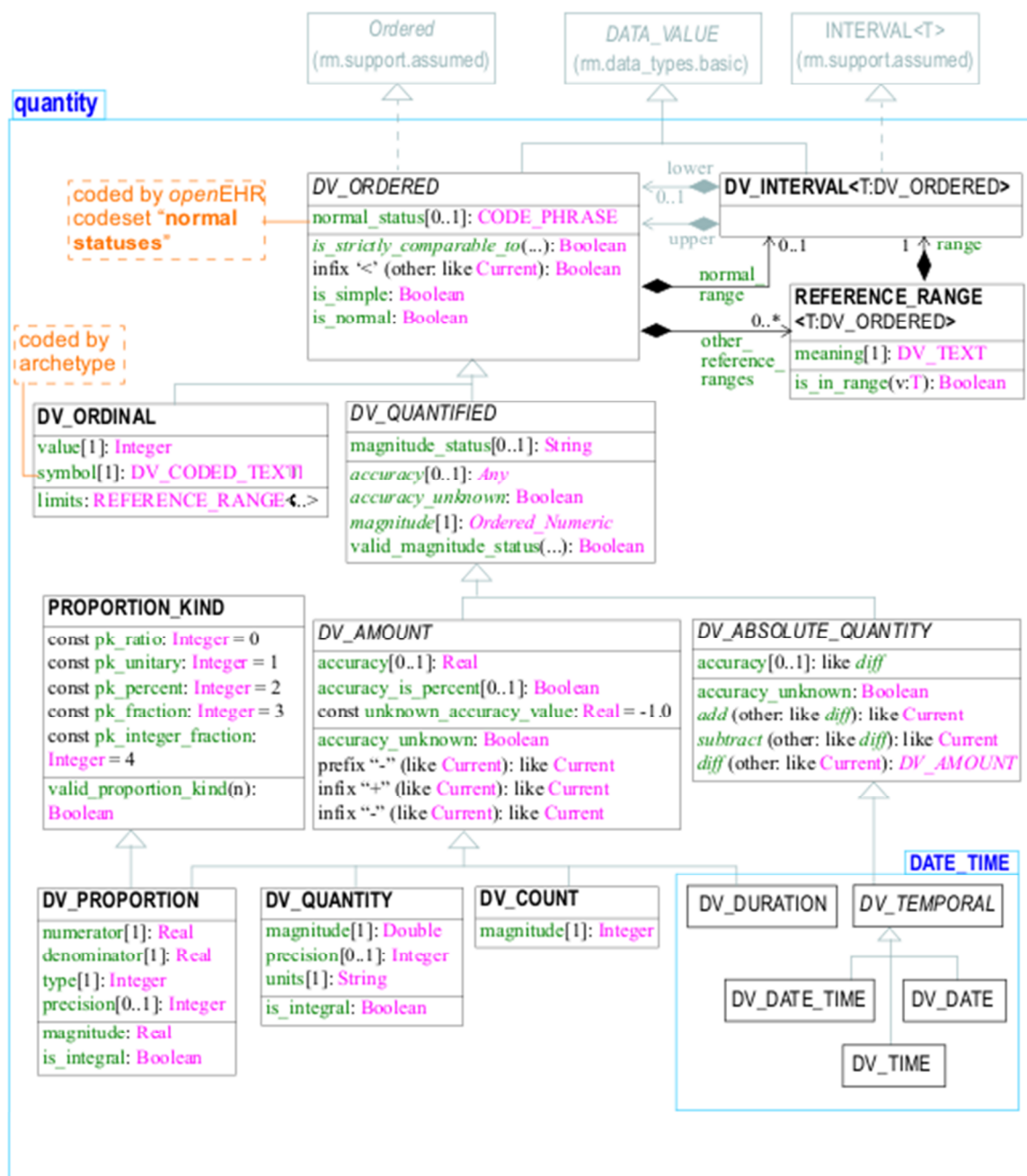


Figura 57 - O Pacote OpenEHR Quantity  
Fonte: Beale *et al.* (2008b)

O pacote *Quantity* (Figura 51) provê classes utilizadas para que se possa representar qualquer tipo ordenado o que inclui valores ordinais (para representar ordens simbólicas como “+”, “++”, “+++”) bem como quantidades mensuráveis.

## A Semântica Básica

A característica básica de todos os valores designados como quantidades é a ordenação, ou seja, o operador “<” pode ser definido entre quaisquer dois valores do domínio.

Neste pacote, uma classe designada por DV\_ORDERED dá origem a todas as outras classes, dividindo-se, inicialmente, por herança entre ordinais e quantidades, representadas pelas classes DV\_ORDINAL e DV\_QUANTIFIED. A primeira classe, DV\_ORDINAL, representa dados cujos valores numéricos exatos não são conhecidos com exatidão e são expressos de maneira simbólica como “+”, “++”, “+++”, ou por expressões como “brando”, “médio”, “severo”, onde a cada símbolo pode ser associado um valor numérico possibilitando uma base para comparações. Já as instâncias da classe DV\_QUANTIFIED e todos os seus subtipos exigem que se conheça com precisão seus valores numéricos.

### Valores Ordinais

A Medicina é um domínio de conhecimento em que, é um fato relativamente comum, o uso de símbolos para representar magnitudes relativas cujos valores exatos não são conhecidos. O propósito mais comum nessa prática é a classificação de pacientes em grupos que se localizam em intervalos com limites difusos, onde os valores concretos não são importantes, mas a categorização sim. Pode-se tomar como exemplo a categorização da dor expressa por um paciente como “branda”, “média” ou “severa”, ou a resposta reflexa da percussão no tendão como “-”, “+/-”, “+”, “++”, “+++” ou “++++”, onde não há necessidade de precisão científica dos valores de tais medidas uma vez que refletem a avaliação do clínico para experiências subjetivas do paciente que este está analisando, porém é possível compreender a existência de um ordenamento nos símbolos utilizados.

### Coisas Contáveis

Uma situação comum no domínio médico é lidar com quantidades contáveis e não dimensionáveis, como um número de doses de algum medicamento, número de nascimentos prematuros, número de comprimidos, etc., cujos valores são sempre inteiros, embora algumas entidades contáveis, como comprimidos, possam ser divididas em frações como “metades” ou “quartos”.

### Quantidades Dimensionáveis

O tipo de quantidade mais comum é a dimensionável, que corresponde a qualquer coisa passível de mensuração e que apresenta alguns aspectos advindos de sua própria natureza quantificável, como:

- Magnitude expressa por um número real.
- Uma propriedade física sendo medida e possuindo unidades apropriadas.
- Precisão, ou o número de casas decimais em que o valor é registrado.
- Exatidão, ou o erro assumido na medição feita por instrumento ou julgamento de uma pessoa.

Exemplos de quantidades dimensionáveis:

- Pressão sistólica: 110 mmHg
- Altura de um paciente: 178 cm
- Média de ocorrências de ataque de asma: 7 /mês
- Perda de peso: 2.5 kg

A classe DV\_QUANTIFIED dá origem a duas subclasses, sendo a primeira, DV\_AMOUNT utilizada para representar porções de algo sob o ponto de vista de uma propriedade física como massa ou sob o ponto de vista da quantidade cigarros sendo a semântica dos operadores “+“ e “-” semelhante à dos números aritméticos, pois a adição de duas quantidades relativas de algo sob a mesma unidade produz outra quantidade relativa do mesmo tipo.

A segunda subclasse é DV\_ABSOLUTE\_QUANTITY que modela quantidades cujos valores absolutos pertencem a uma escala ou “linha” com origem definida como, por exemplo, o conceito temporal de data. Porém, diferente da semântica da classe anterior, as operações sobre instâncias desse tipo não geram um resultado do mesmo tipo como, por exemplo, a diferença entre duas datas não é outra data e sim uma duração.

As subclasses da classe DV\_AMOUNT são DV\_PROPORTION, DV\_QUANTITY, DV\_COUNT e DV\_DURATION. O tipo DV\_COUNT é um inteiro e é utilizado para o registro de coisas contáveis como o número de vezes que uma paciente esteve grávida, ou seja um número sem a necessidade de unidades ou indicação de precisão.

Uma observação a respeito da permissividade da semântica proposta pelo padrão

OpenEHR para os tipos quantificados é que uma vez criada uma instância de DV\_QUANTITY representando algum valor acompanhado de magnitude, precisão e unidade, torna-se possível a comparação entre duas instâncias compatíveis pela igualdade de unidades, como duas medidas de pressão em mmHg. Deve-se ter em conta que a semântica proposta não garante a exclusão de comparações sem sentido, como comparar a pressão arterial de alguém com a pressão atmosférica (ambas em mmHg), estando fora do escopo do pacote QUANTITY, isso deve ser feito no nível da implementação do software, o qual deve garantir a fonte dos dados quantificados utilizados nos arquétipos.

### **Proporções**

A classe DV\_PROPORTION representa razões, ou seja valores relativos, através de um numerador e um denominador e uma magnitude. Tipos de proporção suportados são:

- Percentual: denominador = 100.
- Unitário: denominador = 1.
- Fração: numerador e denominador são valores inteiros, como 1/2, 3/4, etc.

## Pacote Date\_Time

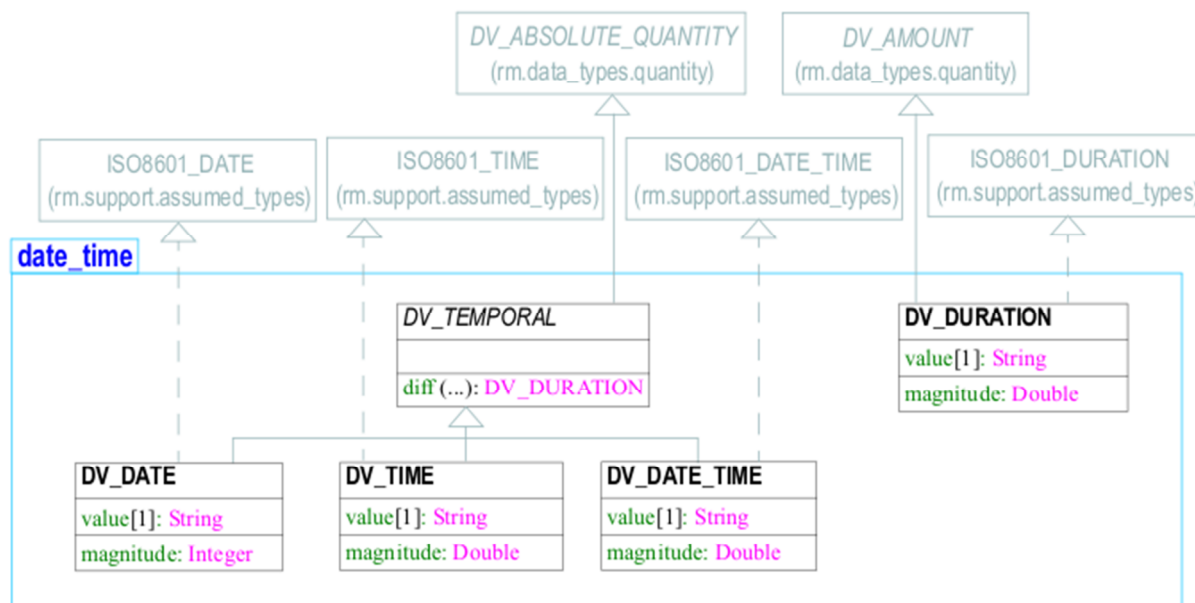


Figura 58 - O Pacote OpenEHR Date Time  
Fonte: Beale *et al.* (2008b)

### Abordagem Geral para o Registro do Tempo/Datas

Os tipos de dados date/time (Figura 52) podem ser divididos em duas categorias: absoluta e relativa. A primeira, compreende os conceitos de data e tempo mensurados a partir de uma origem conhecida, como exemplo, Date e Date\_time possuem como origem a data 0001-01-01, enquanto medidas de tempo se iniciam a partir da meia noite. A segunda categoria contém o conceito de duração que expressa o tempo existente entre dois pontos quais quer no tempo, assim, instâncias da classe DV\_DURATION são utilizadas para expressar a duração de fenômenos clínicos.

Portanto, dentro dessa abordagem, o pacote date\_time inclui três conceitos de data e tempo absolutos: DV\_DATE, DV\_TIME, DV\_DATE\_TIME, e um conceito relativo: DV\_DURATION.

Os conceitos possíveis de serem representados por tais tipos são:

- Data: um tipo para registrar ano, mês e dia do mês, como datas de nascimento e data de início de um tratamento para um dado problema de saúde.
- Tempo: registra hora, minuto, segundo, e a *timezone*, como horário da refeição do paciente ou a hora do dia em que algum problema ocorreu. A *timezone* específica é

requerida no caso de um registro eletrônico compartilhado entre países.

- Date\_time: registra ano, mês e dia do mês, hora, minuto, segundo, e a *timezone*. Como, por exemplo, data e hora da morte de uma pessoa,
- Duração: registra a duração de um evento ou (in)atividade como dias, horas, minutos e segundos.

### Pacote Time\_Specification

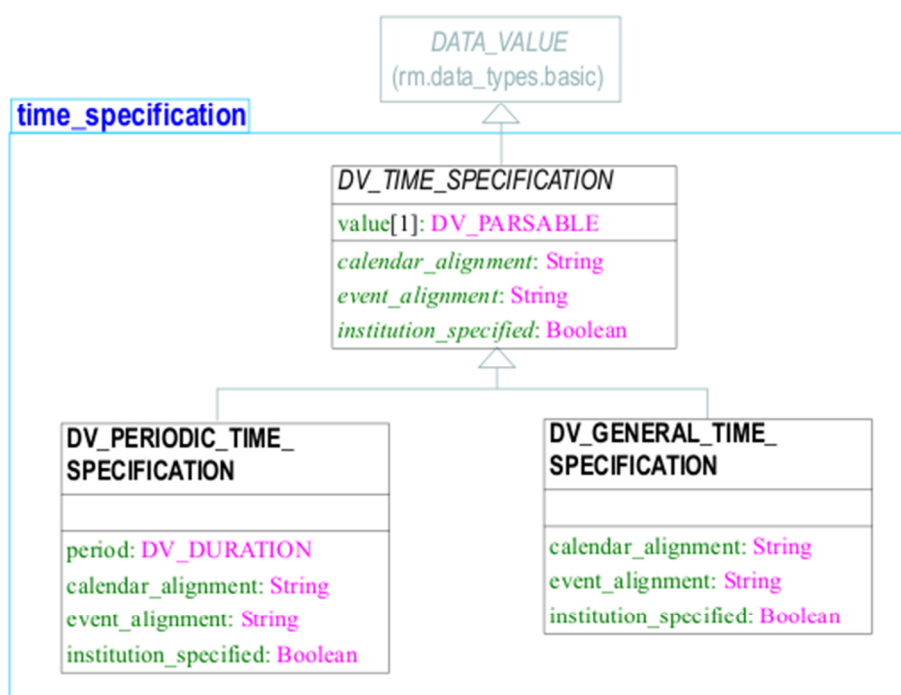


Figura 59 - O Pacote OpenEHR Time Specification  
Fonte: Beale *et al.* (2008b)

Pacote de classes definido para o caso de um dado sistema necessitar definir seus próprios tipos de data e tempo além dos anteriormente expostos, (Figura 53).

## Pacote Encapsulated

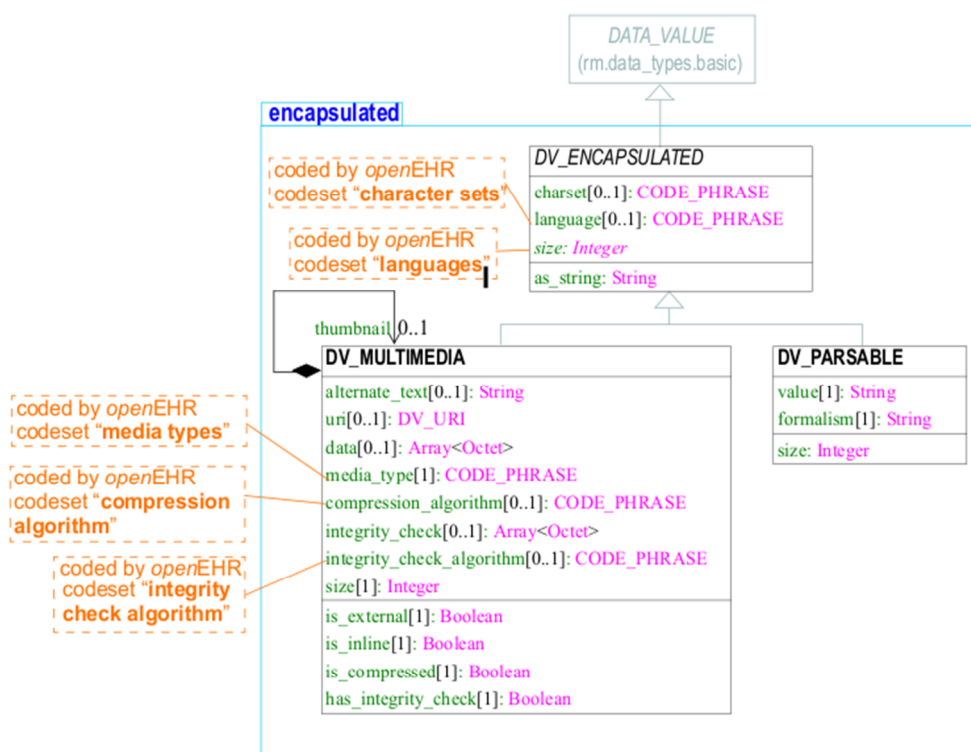


Figura 60 - O Pacote OpenEHR *Encapsulated*

Fonte: Beale *et al.* (2008b)

O pacote *encapsulated* (Figura 54) define classes para representar dados que se encontram externos ao modelo de referência do RES, mas suficientemente bem documentados por atributos de metadados para serem processados por ferramentas específicas, como multimídia e dados parseáveis. Tipos de conteúdo nesta categoria são:

- Imagens em formatos comprimidos ou não comprimidos.
- Séries de dados do tipo Bio-sinais, como, por exemplo, um conjunto de valores representando um traçado ECG.
- Conteúdo textual num arquivo que contenha alguma linguagem parseável como instâncias XML, HTML, etc., sendo que o nome do formalismo utilizado deve ser registrado como meta.
- Conteúdo binário que pode ser processado por ferramentas dedicadas.
- Assinaturas digitais.

Deve-se notar que metadados devem ser incluídos junto ao conteúdo de modo a



possibilitar que este possa ser processado, normalmente a indicação do tipo, como “jpeg”, “mp3”, “documento do MS-word” ou o nome da ferramenta que pode ser utilizada para processá-lo. Outros metadados que podem ser incluídos são:

- Tamanho do conteúdo.
- Linguagem natural, se houver.
- Um *thumbnail* ou outra forma reduzida do conteúdo original.
- *Checksums* caso esteja disponível.

### Pacote *URI*

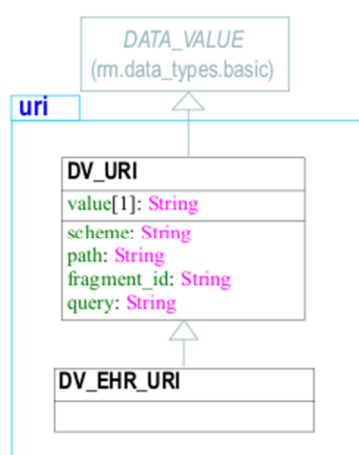


Figura 61 - O Pacote OpenEHR URI  
Fonte: Beale *et al.* (2008b)

Este pacote (Figura 55) representa o *Uniform Resource Identifier* (URI) definido pelo W3C<sup>41</sup>. Um exemplo comum de uso pode ser a referência a uma guia clínica ou outro documento associado ao tratamento de um paciente registrado no sistema de RES.

O tipo DV\_URI permite a criação de tipos referentes a objetos na rede mundial de computadores ou *world wide web* e seus subtipos especializam esta propriedade, sempre mantendo o *design* mais simples de Strings.

<sup>41</sup>O World Wide Web Consortium (W3C) é a principal organização de padronização da World Wide Web. Página institucional: <<http://www.w3.org/>>.

#### B4) O Modelo de Informação *Data Structures*

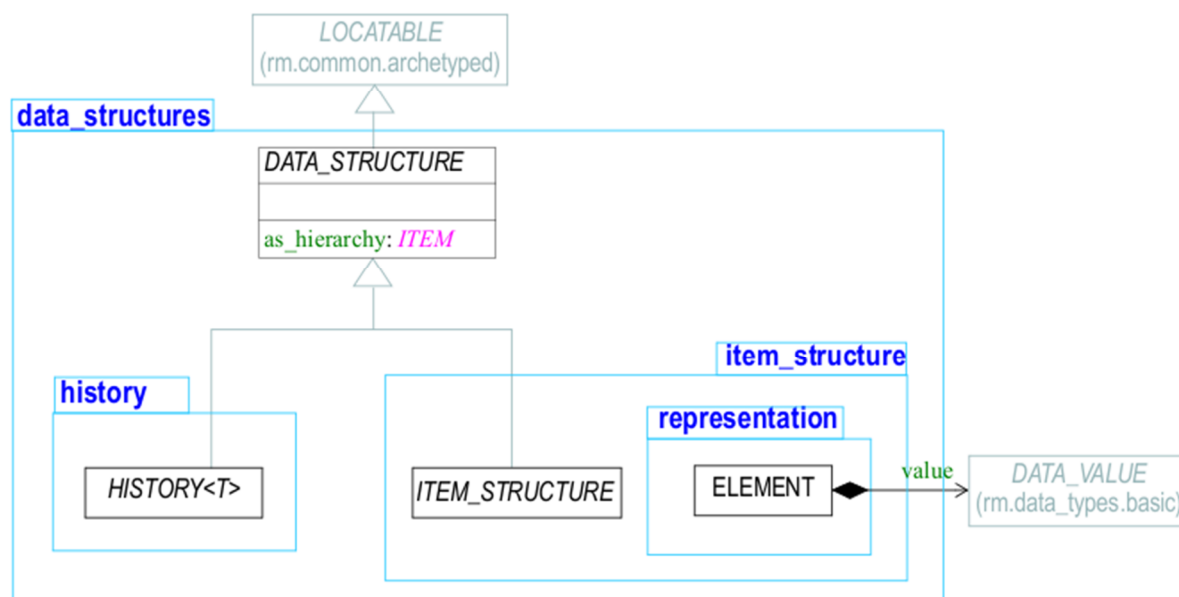


Figura 62 - O Pacote OpenEHR *Data Structures*  
 Fonte: Beale *et al.* (2008c)

O modelo de informação *data structures* (Figura 56) busca suprir os outros modelos de informação de estruturas de dados genéricas para serem utilizadas na expressão de conteúdos em arquétipos que dependam de tais tipos de organização de dados. Estruturas como:

- Single: item único que contém qualquer valor único, como, por exemplo, uma medida de temperatura, peso ou altura de uma pessoa.
- List (lista): representa uma lista linear de itens nomeados, como no resultado de um teste de patologia.
- Table (tabela): Utilizada para representar dados tabulados, na forma de tabelas de comprimento limitado ou ilimitado constando de colunas nomeadas e ordenadas, e, caso necessário, linhas nomeadas.
- Tree (árvore): Consta de dados em formato de árvore, que podem ser utilizados para representar, por exemplo, uma lista de listas ou outra estrutura de dados de complexidade similar.

## Pacote Item Structure

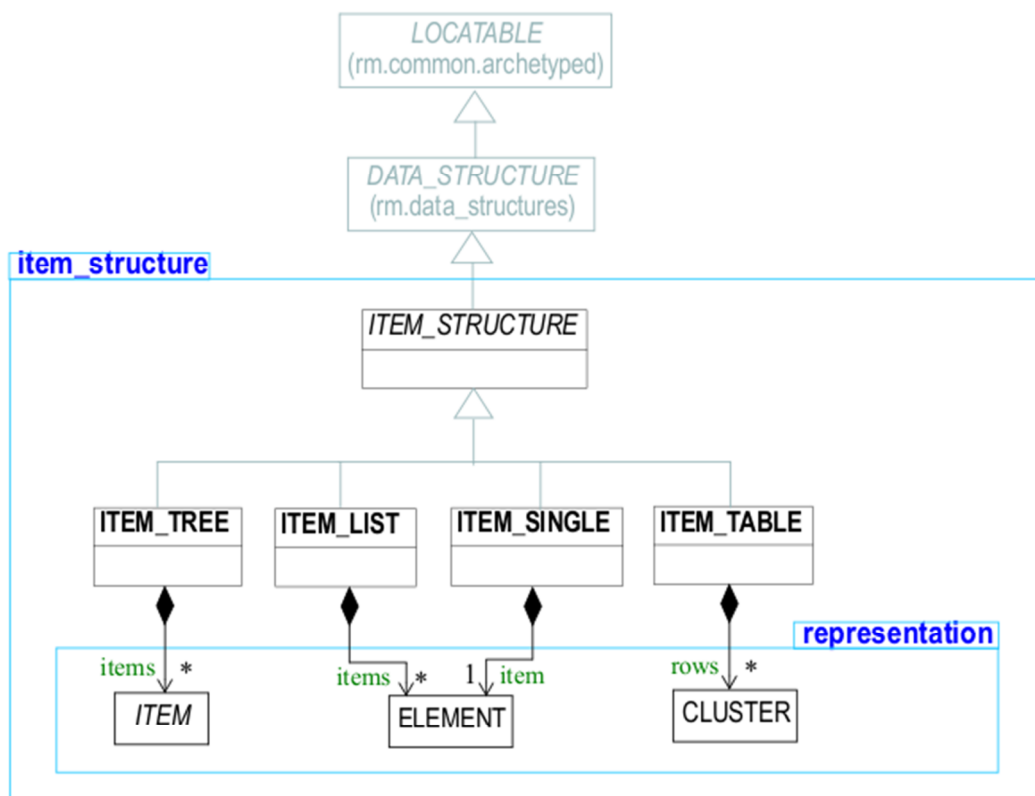


Figura 63 - O Pacote OpenEHR Item Structure  
Fonte: Beale *et al.* (2008c)

As classes que herdam da classe ITEM\_STRUCTURE (Figura 57) procuram modelar as estruturas de dados que mais comumente ocorrem nos registros de saúde, como ITEM\_SINGLE (para valores únicos, como o peso de um paciente), ITEM\_LIST (para listas, como as partes de um endereço), ITEM\_TREE (para dados hierarquicamente estruturados como um relatório de microbiologia) e ITEM\_TABLE (para dados tabulados como de testes de acuidade visual ou resultados de testes de reflexos).

Os valores dos dados são conectados às estruturas de dados espaciais através do valor do atributo da classe ELEMENT. Esta classe também possui um atributo denominado *null\_flavour*, que indica como ler o valor registrado.

Deve-se notar que estas classes anteriormente descritas não obrigatoriamente são equivalentes às classes de nomes similares nas várias bibliotecas de estruturas de dados das diferentes linguagens de programação.

## Pacote Representation

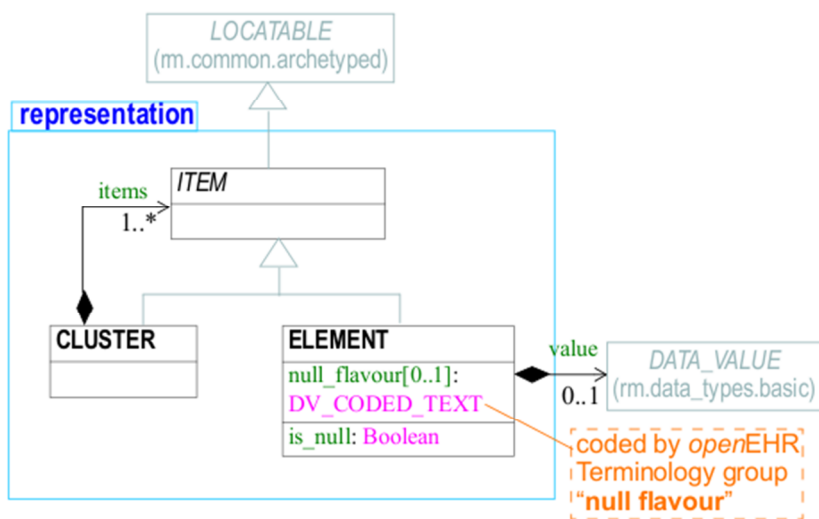


Figura 64 - O Pacote OpenEHR Representation  
Fonte: Beale *et al.*(2008c)

Este pacote (Figura 65) contém classes para uma representação hierárquica, se necessária, de qualquer estrutura de dados.

## Pacote *History*

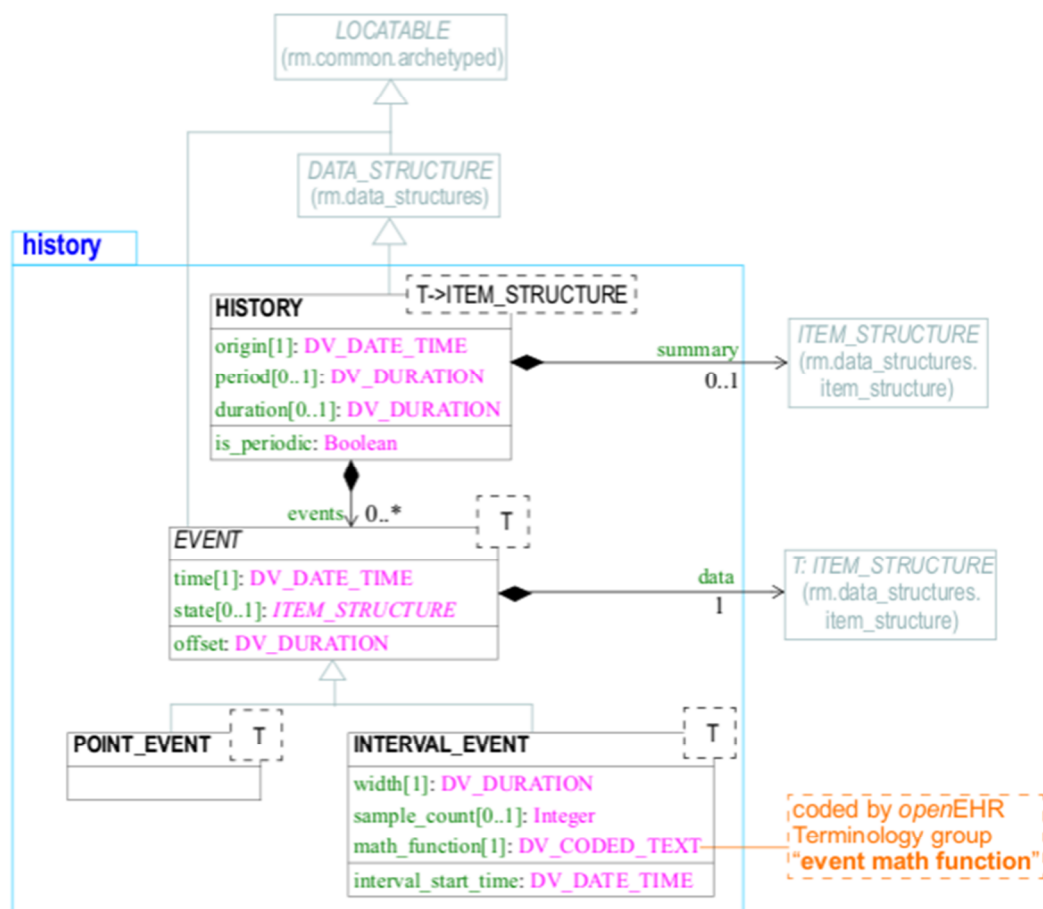


Figura 66 - O Pacote OpenEHR History  
Fonte: Beale *et al.* (2008c)

O pacote *history* (Figura 59) foi desenvolvido com o intuito de formalizar o conceito de passado e de tempo linear através do qual dados históricos de qualquer complexidade estrutural podem ser registrados. Os dados a serem registrados podem consistir de um exemplo simples ou de muitos, o registro será feito da mesma maneira: um histórico de eventos, ou uma série temporal que permite o acesso aos dados de maneira uniforme por qualquer software. Pode-se tratar de uma simples medida de temperatura, peso ou pressão arterial, uma série de imagens ou de arquivos multimídia (veja o pacote *encapsulated*, Figura 54).

O objetivo do modelo HISTORY é representar dados no tempo, em que cada item na série corresponde à medição de um mesmo fenômeno obtido pelo mesmo método de medição. Dessa forma, pode-se entender os itens de um “percurso” de medições de uma instância de

HISTORY como representando modificações num dado fenômeno ao longo do tempo, permitindo a obtenção de gráficos no tempo, análises estatísticas, etc.

O modelo define os tipos HISTORY<T>, EVENT<T>, POINT\_EVENT<T>, e INTERVAL\_EVENT<T> onde o parâmetro T é do tipo ITEM\_STRUCTURE e define o tipo de dados que pode ser registrado numa instância da classe HISTORY. O efeito obtido é a possibilidade da repetição no tempo de estruturas de dados, mostrando o percurso da medição até os dados atuais. Caso se deseje representar medições repetidas no tempo e realizadas manualmente uma série não periódica de instâncias da classe POINT\_EVENT (representa o “instantâneo” de um valor) pode ser utilizada. Para representar as saídas do monitoramento de sinais vitais, que normalmente vem como média acompanhada dos valores máximo e mínimo, pode-se utilizar instâncias periódicas da classe INTERVAL\_EVENT.

### **Marcação do Tempo**

Uma instancia da classe HISTORY contém sua origem no tempo através do atributo DV\_DATE\_TIME, indicando o que é considerado como o ponto zero no tempo da série de eventos. O tempo de origem da série não necessita corresponder ao do primeiro registro, podendo-se citar como exemplo o teste Apgar1 tomado aos 1 e 3 minutos. Periodicidade e aperiodicidade são expressas via atributos *is\_periodic* e *period*. Para uma série de periódica no tempo, o atributo *period* recebe o valor da duração do período e o atributo *is\_periodic* deve retornar *True*. A duração total da History é dada pela função *duration*.

### **Point\_Event**

O tipo mais simples de evento numa History é um “*point event*”, expresso por instâncias da classe POINT\_EVENT, representando um valor instantâneo. Uma instância de HISTORY pode ser composta apenas por “*point event*”, como numa situação em que várias medidas de pressão arterial são tomadas no tempo com o paciente em diferentes posições. O resultado de um teste Apgar é um típico exemplo de dado aperiódico, consistindo de 2 ou 3 eventos, cada um contendo 5 valores e um 6º valor representando o escore Apgar para este ponto no tempo. A Figura 60 mostra uma History da medida de peso de um paciente:

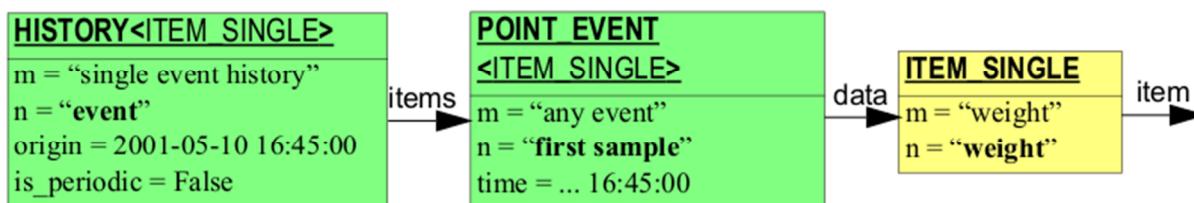


Figura 67 - Uma History de Uma Medida de Peso  
Fonte: Beale *et al.*(2008c)

### Interval\_Event

Instancias da classe INTERVAL\_EVENT são usadas para expressar valores correspondentes a um intervalo no tempo. O atributo width define a duração do intervalo. O significado de um INTERVAL\_EVENT no modelo é o de um valor que, efetivamente, summarize o valor de dados que ocorram num dado intervalo de tempo. A Figura 61 ilustra uma History contendo 2 pares de intervalos de 4 horas de eventos de medidas de pressão, onde cada par contém um valor máximo e a média dos valores de pressão para os pontos no tempo correspondentes a 4h e 8h.

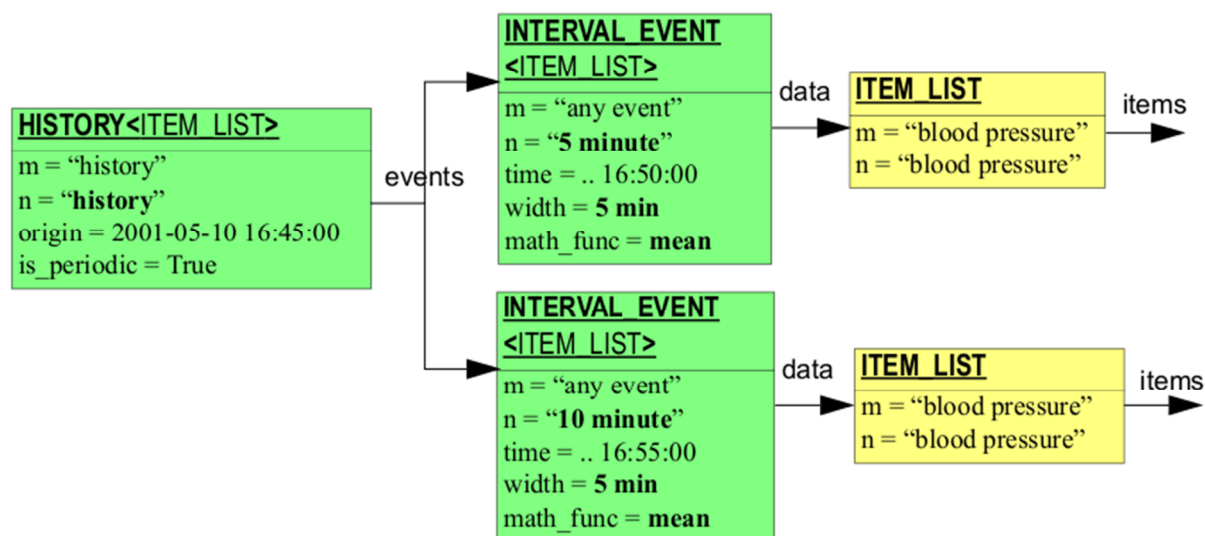


Figura 68 - Uma History de Medidas de Pressão Arterial  
Fonte: Beale *et al.* (2008c)

**State**

O atributo “state” é entendido como contendo as informações pertinentes para a correta interpretação dos dados primários. Um exemplo seria o dado referente à medida da frequência cardíaca de um paciente, onde o atributo state teria como valor a indicação do nível de esforço a que o paciente foi submetido, como repouso, ciclo de 3 minutos, etc. Tal atributo mostra-se crucial para o uso seguro dos dados registrados no atributo *data*, pois a depender do valor do atributo *state*, estes podem ser normais ou anormais.