



Universidade do Minho
Escola de Engenharia

Tiago Oliveira

**Guias Clínicas:
Representação e Raciocínio**

Tiago Oliveira
**Guias Clínicas:
Representação e Raciocínio**

UMinho | 2011

Setembro de 2011



Universidade do Minho
Escola de Engenharia

Tiago José Martins Oliveira

Guias Clínicas:
Representação e Raciocínio

Tese de Mestrado
Mestrado Integrado em Engenharia Biomédica

Trabalho efectuado sob a orientação do
Professor Paulo Jorge Freitas de Oliveira Novais

Agradecimentos

A realização de uma tese de dissertação reúne contributos de várias pessoas. Como tal não posso deixar de agradecer a todos que contribuíram para o desenvolvimento e conclusão deste trabalho.

Em primeiro lugar gostaria de agradecer ao meu orientador, Professor Paulo Jorge Freitas de Oliveira Novais, pela disponibilidade que demonstrou, pela motivação que transmitiu, pelo bom aconselhamento a todos os níveis e pela confiança no meu trabalho.

Ao Professor José Carlos Maia Neves, pela atenção disponibilizada, pelos esclarecimentos prestados, pela oportunidade que me proporcionou e pela confiança que em mim depositou.

Aos Doutores Mário Freitas e Pedro Pereira pelas guias disponibilizadas e pela ajuda na componente de saúde desta dissertação.

Gostaria de agradecer ao Ângelo Costa pela amizade, companheirismo e paciência que demonstrou ao longo de todo o processo, no esclarecimento de dúvidas e na realização de trabalho conjunto.

Também gostaria de agradecer ao Mário São João pela paciência e pelo *feedback* do trabalho realizado.

Agradeço aos meus amigos por me apoiarem nos bons e maus momentos.

Por fim, agradeço aos meus Pais e Avós o carinho e atenção que sempre me deram e por me proporcionarem os meios para chegar até este ponto.

Dedico esta dissertação aos meus Pais e aos meus Avós.

Título

Guias Clínicas: Representação e Raciocínio

Resumo

Os ambientes de cuidados de saúde são extremamente exigentes para os profissionais. Nestes ambientes há uma grande exposição a situações de tensão, que se repercutem na qualidade da prática clínica. O stress ocupacional origina situações de erro médico, de variações indesejadas na prática clínica e de medicina defensiva.

As Guias Clínicas (GCs), como recomendações clínicas baseadas em investigação científica sólida, podem solucionar tais problemas, fornecendo um suporte para a prática da medicina, baseado na evidência, e preenchendo eventuais vazios de conhecimento dos profissionais de saúde. Contudo, o seu formato actual não responde às exigências de um processo clínico que obriga a tomar decisões rápidas, com segurança. A solução pode passar pela implementação de formatos informáticos de GCs, as chamadas Guias Interpretáveis por Computador (*Computer-Interpretable Guidelines* - CIGs), em sistemas de apoio à decisão. As abordagens actuais de CIGs concentram-se sobretudo em aspectos relacionados com a modelação de tarefas, restrições temporais à execução de guias e integrações com sistemas de informação locais. No entanto, não providenciam um tratamento de casos de Informação Imperfeita, que são comuns nos processos clínicos.

Há necessidade de uma representação de guias, que combine a capacidade de modelação de tarefas das abordagens actuais de CIGs, com linguagens de programação que permitam expressar casos de Informação Imperfeita e métodos que permitam quantificá-los. Para o efeito, recolheram-se as principais características das abordagens de CIGs actuais e propôs-se um modelo, utilizando a Extensão à Programação em Lógica (EPL) e o método da Qualidade da Informação (*Quality of Information* - QoI). A aplicabilidade deste modelo foi estudada através de um caso de estudo com uma GC para detecção e tratamento de elevados níveis de colesterol.

Concluiu-se que, embora careça de melhoramentos ao nível de um suporte para o estado do paciente e ao nível da estruturação da informação, o modelo apresenta potencial para melhorar os resultados do processo clínico.

Title

Clinical Guidelines: Representation and Reasoning

Abstract

Healthcare environments are very demanding. In these environments healthcare professionals are exposed to many stressful situations that affect negatively the quality of clinical practice. Occupational stress is among the causes of medical errors, undesirable variations in clinical practice and defensive medicine.

The use of Clinical Guidelines may be a solution to these issues. They are evidence based recommendations that support good clinical practice and may compensate for knowledge gaps of healthcare professionals. However, their current format does not meet the requirements of real time decision support in the clinical process.

Implementing Computer-Interpretable Guidelines (CIGs) in clinical decision support systems shows promises of both changing the process of healthcare delivery and improving its outcomes. The existing CIG approaches focus mainly on task modeling, temporal constraints to the execution of guidelines and integration with local information systems. Yet, they fail to address the issue of Imperfect Information, which is common in many clinical cases.

There is a need for a guideline representation, which combines the task modeling capabilities of the existing CIG approaches with programming languages that enable the expression of cases of Imperfect Information and methods to quantify them. For this purpose we collected the main features of the existing approaches and proposed a model that uses the Extension to Logic Programming (ELP) and the method of Quality of Information. The applicability of this model was studied with a guideline for detection and treatment of elevated levels of cholesterol.

Although the model needs improvements in the support for the patient state and the structuring of information, it has the potential to improve clinical results.

Índice Geral

Agradecimentos.....	iii
Título.....	iv
Resumo.....	iv
Title.....	vi
Abstract.....	vi
Índice Geral.....	vii
Notação e Terminologia.....	xii
Notação.....	xii
Acrónimos.....	xii
Índice de Figuras.....	xv
Índice de Tabelas.....	xviii
Índice de Programas.....	xix
1. Introdução.....	1
1.1. Motivação.....	2
1.1.1. Stress Ocupacional e Erro Médico.....	2
1.1.2. Variações na Prática Clínica.....	3
1.1.3. Medicina Defensiva.....	4
1.2. Objectivos.....	5
1.3. Metodologia de Investigação.....	6
1.4. Estrutura.....	6
2. Guias Clínicas.....	9
2.1. Introdução.....	9
2.2. Definição.....	10
2.3. Características.....	11

2.4.	Objectivos das Guias Clínicas.....	13
2.5.	Benefícios e Limitações	13
2.5.1.	Para os Profissionais de Saúde	13
2.5.2.	Para os Pacientes.....	14
2.5.3.	Para as Organizações de Saúde	15
2.6.	Metodologia de Desenvolvimento	15
2.6.1.	Definição do Tema.....	15
2.6.2.	Composição do Grupo de Desenvolvimento de Guias Clínicas.....	16
2.6.3.	Análise do Problema.....	17
2.6.4.	Formulação de Questões-chave.....	17
2.6.5.	Pesquisa Bibliográfica	18
2.6.6.	Avaliação Crítica da Bibliografia Seleccionada.....	18
2.6.7.	Resumo da Bibliografia em Tabelas de Evidências.....	19
2.6.8.	Redacção de um Esboço da Guia Clínica.....	19
2.6.9.	Classificação das Evidências.....	19
2.6.10.	Outras Considerações.....	21
2.6.11.	Avaliação Externa.....	21
2.6.12.	Aprovação e Publicação da Guia Clínica	21
2.7.	<i>Living Guidelines</i>	22
3.	Guias Interpretáveis por Computador	23
3.1.	Introdução.....	23
3.2.	Obstáculos à Implementação de CIGs.....	23
3.3.	Benefícios da Implementação de CIGs	24
3.4.	Principais Aspectos do Desenvolvimento de CIGs.....	25
3.4.1.	Modelação e Representação de Guias Clínicas	25
3.4.2.	Aquisição de Guias Clínicas.....	26

3.4.3.	Verificação de Guias Clínicas.....	27
3.4.4.	Execução de Guias Clínicas.....	27
3.5.	Abordagens à Modelação de CIGs.....	27
3.5.1.	Arden Syntax	28
3.5.2.	Guideline Interchange Format (GLIF).....	31
3.5.3.	PROforma.....	35
3.5.4.	Asbru	39
3.5.5.	EON	43
3.5.6.	Standards-Based Sharable Active Guideline Environment (SAGE)	47
3.6.	Análise.....	50
4.	Representação do Conhecimento.....	55
4.1.	Introdução.....	55
4.2.	Dados, Informação e Conhecimento	55
4.3.	Informação Imperfeita	56
4.4.	Técnicas Numéricas para lidar com Informação Imperfeita.....	58
4.4.1.	Redes Bayesianas.....	58
4.4.2.	Teoria de Dempster-Shafer.....	59
4.4.3.	Conjuntos Fuzzy e Lógica Fuzzy	60
4.4.4.	Lógica Probabilística	61
4.5.	Verificação da Validade de Guias através de Lógica Temporal.....	61
4.6.	Programação em Lógica.....	63
4.7.	Extensão à Programação em Lógica	64
4.8.	Qualidade da Informação.....	68
4.9.	Análise.....	69
5.	Proposta de um Modelo de CIGs.....	71
5.1.	Introdução.....	71

5.2.	Primitivas de Representação.....	72
5.2.1.	<i>Root task</i>	74
5.2.2.	<i>Plan</i>	75
5.2.3.	<i>Decision</i>	79
5.2.4.	<i>Action</i>	80
5.2.5.	<i>Enquiry</i>	82
5.3.	Nível Computacional.....	83
5.3.1.	Representação do Conhecimento.....	83
5.3.2.	Qualidade da Informação e Pontuação de uma Conclusão.....	86
5.3.3.	Motor de Inferência para a Tomada de Decisão.....	87
5.3.4.	Partilha e Validação de Guias Clínicas.....	90
6.	Caso de Estudo.....	92
6.1.	Introdução.....	92
6.2.	Colesterol e Doença Cardíaca Coronária.....	92
6.3.	Descrição da Guia para Detecção e Tratamento de Níveis Elevados de Colesterol.....	94
6.4.	Representação da Guia Clínica no Modelo Proposto.....	95
6.4.1.	<i>Root Task</i> RT1.....	95
6.4.2.	<i>Plan</i> P1.....	96
6.4.3.	<i>Plan</i> P2.....	99
6.4.4.	<i>Plan</i> P3.....	100
6.4.5.	<i>Plans</i> P4, P5 e P6.....	101
6.4.6.	<i>Plan</i> P8.....	103
6.4.7.	<i>Plan</i> P9.....	104
6.4.8.	<i>Plan</i> P10.....	106
6.4.9.	<i>Plan</i> P7.....	106
6.4.10.	<i>Plan</i> P11.....	108

6.4.11. <i>Plans</i> P12, P13 e P14	110
6.4.12. <i>Plan</i> P15	110
6.4.13. <i>Plan</i> P16	111
6.4.14. <i>Plan</i> P17	112
6.5. Tomada de Decisão.....	113
6.6. Análise.....	119
7. Conclusões e Trabalho Futuro	122
7.1. Conclusões	122
7.2. Contribuições	123
7.3. Trabalho Relevante.....	124
7.4. Trabalho Futuro.....	125
Referências Bibliográficas	126
Referências Electrónicas	135
Anexos	136
Anexo A - Programas Auxiliares do Motor de Inferência.....	137
Anexo B - XML Schema do Modelo de CIGs Proposto	142
Anexo C – Guia para Detecção e Tratamento de Níveis Elevados de Colesterol	145
Anexo D – <i>Plans</i> da Guia para Detecção e Tratamento de Elevados Níveis de Colesterol	152

Notação e Terminologia

Notação

Ao longo do documento utiliza-se texto em itálico para palavras em língua estrangeira, equações e fórmulas matemáticas e lógicas. Utiliza-se o tipo de letra `Courier New` para programas e para o nome de predicados.

Alguns termos não foram traduzidos, nomeadamente os que dizem respeito aos modelos de representação de guias, de forma a representar fielmente os modelos.

Acrónimos

AHRQ	Agency for Health Research and Quality
API	Application Programming Interface
ASTM	American Society for Testing and Materials ASTM
BNF	Backus-Naur Form
CEMBE	Centro de Estudos de Medicina Baseada na Evidência
COGS	Conference on Guideline Standardization
EPL	Extensão à Programação em Lógica
FNCLCC	National Federation of Cancer Centers
GAC	Guideline Advisory Committee
GC	Guia Clínica
GEL	Guideline Expression Language
G-I-N	Guideline International Network

GLEE	Guideline Execution Engine
GLIF	Guideline Interchange Format
GRADE	Grading of Recommendations Assessment, Development and Evaluation
HDL	High Density Lipoprotein
HDL - C	HDL Cholesterol
HL7	Health Level Seven
IM	Informática Médica
KIF	Knowledge Interchange Format
LDL	Low Density Lipoprotein
LDL-C	LDL Cholesterol
NGC	National Guideline Clearinghouse
NHLBI	National Heart Lung and Blood Institute
NICE	National Institute for Clinical Excellence
NZGG	New Zealand Guidelines Group
OMS	Organização Mundial de Saúde
PAL	Protégé Axiom Language
PIB	Produto Interno Bruto
PL	Programação em Lógica
R ² L	Red Representation Language
RDF	Resource Description Framework
SADC	Sistema de Apoio à Decisão Clínica
SAGE	Standards-Based Sharable Active Guideline Environment

SIGN	Scottish Intercollegiate Guidelines Network
TLC	Therapeutic Lifestyle Changes
TNM	Task Network Model
UE	União Europeia
UML	Unified Modeling Language
UMLS	Unified Medical Language System
VLDL	Very Low Density Lipoprotein
VMR	Virtual Medical Record
XML	Extensible Markup Language
XSD	XML Schema Definition

Índice de Figuras

Figura 1 -- Representação do ciclo de vida de uma GC (adaptado de (Rosenbrand, Croonenborg, & Wittenberg, 2008)).	16
Figura 2 -- Representação das diferentes abordagens à modelação de CIGs, posicionadas sobre um eixo temporal de acordo com a altura em que se iniciou o seu desenvolvimento. Uma seta entre duas abordagens significa que a abordagem antecedente influenciou a abordagem consequente (adaptado de (Elkin, et al., 2000)).	28
Figura 3 -- Exemplo de um MLM (reproduzido de (Teije, Miksch, & Lucas, 2008)).	30
Figura 4 -- Representação esquemática dos tipos de passos do modelo GLIF3.	33
Figura 5 -- Excerto de uma representação gráfica de uma guia para vacinação em GLIF3 (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).	35
Figura 6 -- Representação esquemática das principais classes de tarefas do modelo PROforma.	36
Figura 7 -- Visualização de uma guia no ambiente de criação de tarefas PROforma (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).	39
Figura 8 -- Visualização de uma guia para o tratamento de distúrbio respiratório em crianças em AsbruView (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).	43
Figura 9 -- Representação esquemática das principais classes do modelo EON.	45
Figura 10 - Representação esquemática das principais classes do modelo SAGE.	48
Figura 11 -- Esquema da arquitectura de execução de SAGE (reproduzido de (Ram, et al., 2004)).	49
Figura 12 -- Representação esquemática das principais características dos modelos de representação de CIGs.	51
Figura 13 -- O modelo dados-informação-conhecimento, segundo o processo de tomada de decisão (adaptado de (Aamodt & Nygård, 1995)).	56
Figura 14 -- Relação entre as diferentes classes de tarefas. <i>Actions</i> , <i>decisions</i> e <i>enquiries</i> são tarefas atómicas, ao passo que <i>plans</i> são tarefas compostas, na medida em que podem ser constituídos por um número variável de tarefas atómicas e sub-planos.	73
Figura 15 -- Formação de um trombo a partir de uma placa instável (reproduzido de (Silva, 2004)).	93

Figura 16 – Representação da <i>root task</i> RT1 e da guia como <i>plan</i> P1.....	96
Figura 17 – Representação do <i>plan</i> P1 (1ª parte).....	97
Figura 18 – Representação do <i>plan</i> P1 com os planos P4, P5 e P6 (2ª parte).	98
Figura 19 – Representação do <i>plan</i> P2.	100
Figura 20 – Representação do <i>plan</i> P3.	101
Figura 21 – Representação do <i>plan</i> P4.	102
Figura 22 – Representação do <i>plan</i> P8.	103
Figura 23 – Representação do <i>plan</i> P9.	105
Figura 24 – Representação do <i>plan</i> P10.	106
Figura 25 – Representação do <i>plan</i> P7.	107
Figura 26 – Representação do <i>plan</i> P11 (1ª parte).....	108
Figura 27 – Representação do <i>plan</i> P11 (2ª parte).....	109
Figura 28 – Representação do <i>plan</i> P12.	110
Figura 29 – Representação do <i>plan</i> P15.	111
Figura 30 – Representação do <i>plan</i> P16.	112
Figura 31 – Representação do <i>plan</i> P17.	113
Figura 32 – Representação do painel de bordo do cenário 1 na decision D2: (a) pontuação do nível de LDL-C inferior a 100 mg/dL, (b) pontuação do nível de LDL-C inferior a 130 mg/dL e (c) pontuação do nível de LDL-C inferior a 160 mg/dL.	117
Figura 33 – Representação do modelo de tomada de decisão baseado em GCs.	119
Figura 34 – Primeira página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	146
Figura 35 – Segunda página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	147
Figura 36 – Terceira página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	148
Figura 37 – Quarta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	149
Figura 38 – Quinta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	150
Figura 39 – Sexta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).....	151

Figura 40 – Representação do <i>plan</i> P5.	153
Figura 41 – Representação do <i>plan</i> P6.	154
Figura 42 – Representação do <i>plan</i> P13.	155
Figura 43 – Representação do <i>plan</i> P14.	155

Índice de Tabelas

Tabela 1 – Comparação das características de diferentes abordagens de CIGs (adaptado de (Wang, Peleg, Tu, Shortliffe, & Greenes, 2001)).....	53
Tabela 2 – Operadores temporais, utilizados na lógica temporal para avaliação da qualidade de GCs (adaptado de (Lucas, 2003)).	62
Tabela 3 – Atributos de uma root task e respectivos exemplos.	74
Tabela 4 – Atributos de um <i>plan</i> e respectivos exemplos.	75
Tabela 5 – Atributos de uma <i>decision</i> e respectivos exemplos.	79
Tabela 6 – Atributos de uma <i>action</i> e respectivos exemplos.....	81
Tabela 7 – Atributos de um <i>enquiry</i> e respectivos exemplos	82

Índice de Programas

Programa 1 -- Extensão do predicado <code>temperature</code> com representação de um valor nulo do tipo desconhecido.....	66
Programa 2 -- Extensão do predicado <code>temperature</code> com representação de um valor nulo do tipo desconhecido de um conjunto de valores.	66
Programa 3 -- Extensão do predicado <code>temperature</code> com representação de um valor nulo do tipo desconhecido de um conjunto de valores, com invariante.....	67
Programa 4 -- Extensão do meta-predicado <code>demo</code> (reproduzido de (Analide, Novais, Machado, & Neves, 2006)).	67
Programa 5 -- Extensão do predicado <code>rhinorrhea</code>	84
Programa 6 -- Extensão do predicado <code>sore_throat</code> com representação de um valor nulo do tipo desconhecido.....	84
Programa 7 -- Extensão do predicado <code>temperature</code> com representação de um valor nulo do tipo desconhecido de um conjunto de valores.	84
Programa 8 -- Extensão do predicado <code>cough</code> com representação de um valor nulo do tipo desconhecido de um conjunto de valores, com invariante.....	85
Programa 9 – Regras de produção para diagnóstico de infecção respiratória viral.....	86
Programa 10 – Meta-predicado <code>demo_forward</code>	88
Programa 11 – Cláusula do predicado <code>composition</code> que lida com casos em que há informação positiva sobre as condições.	88
Programa 12 – Cláusula do predicado <code>composition</code> que lida com casos em que a condição é um valor nulo do tipo desconhecido.	89
Programa 13 – Cláusulas do predicado <code>composition</code> que lidam com valores nulos do tipo desconhecido de um conjunto de valores. A primeira cláusula refere-se a casos sem invariante e a segunda cláusula refere-se a casos com invariante.....	90
Programa 14 – Regras de produção da tarefa D2 para determinação dos níveis LDL-C a atingir.	114
Programa 15 – Extensões dos predicados do cenário 1.....	115

Programa 16 – Extensões dos predicados do cenário 2.....	115
Programa 17 – Cláusulas do predicado <code>composition</code> que lidam com condições compostas.	138
Programa 18 – Cláusulas do predicado <code>composition</code> que lidam com os operadores ">", "<", ">=", "<=" para o predicado <code>ten_year_risk</code>	140
Programa 19 – Predicado <code>score</code> que calcula a pontuação de uma conclusão.	141
Programa 20 – Predicado <code>conclusions</code> que cria uma lista de conclusões ordenadas por ordem decrescente de pontuação.	141
Programa 21 – XML <i>schema</i> do modelo de CIGs proposto.	144

1. Introdução

A Informática Médica (IM) (Haux, 2010), como disciplina, é relativamente recente. Hoje em dia, é vista como uma disciplina transversal, que constitui a base da medicina e prestação de cuidados de saúde actuais. Consequentemente existe uma pressão considerável sobre a IM no sentido de melhorar a saúde da população, através das suas contribuições para cuidados de saúde eficazes, de qualidade e para inovações na biomedicina e ciências relacionadas. Os principais campos de investigação em IM podem ser agrupados de acordo com a aplicação e avaliação dos sistemas de informação em saúde, representação do conhecimento médico e análise de dados e sinais.

Os primeiros Sistemas de Apoio à Decisão Clínica (SADCs) (Musen, Shahar, & Shortliffe, 2006) foram introduzidos nos finais dos anos sessenta. Contudo, são poucos os que apresentam uma utilização generalizada em contexto clínico. Existe um consenso generalizado de que os SADCs apresentam o potencial de melhorar os cuidados de saúde e modificar o comportamento dos médicos. Apesar da sua utilidade, a falta de aceitação clínica por parte de profissionais de saúde e pacientes representa um entrave à disseminação destes sistemas. Dado que a forma de representação do conhecimento constitui a base de qualquer SADC, a integração nestes sistemas de Guias Clínicas¹ (GCs) pode contribuir para aumentar a sua aceitação.

GCs (Mead, 2000) são recomendações de boas práticas médicas para profissionais de saúde e pacientes, baseadas em evidências científicas. As GCs visam tornar os processos clínicos mais eficientes e estabelecem padrões de qualidade pelos quais os profissionais de saúde se devem orientar. Contudo, as GCs encontram-se sob a forma de documentos longos, que são difíceis de consultar, sobretudo no momento do acto clínico. Além disso, o tempo de revisão e actualização das guias não se coaduna com a velocidade de crescimento e actualização do conhecimento científico. O formato em que estão disponíveis também dificulta a sua actualização. Por estas razões as GCs não cumprem totalmente o seu objectivo de fornecer conhecimento aos profissionais de saúde e promoção de boas práticas médicas. O desenvolvimento de formatos digitais para representação de GCs (Guias Interpretáveis por

¹ Na literatura anglo-saxónica o termo utilizado é *Clinical Guidelines*, pelo que se optou por traduzir este termo para Guias Clínicas.

Computador²) (Kaiser & Miksch, 2009) apresenta-se como uma solução para a falta de dinamismo do formato em papel de GCs, o que permite o desenvolvimento de mecanismos automáticos de revisão e actualização de guias e a sua integração em SADCs. Isto também possibilita a incorporação de métodos de inferência em cenários de Informação Imperfeita, uma vez que o contexto clínico é fértil neste tipo de situações.

1.1. Motivação

A confiança nos profissionais de saúde tem uma importância vital em qualquer procedimento clínico. Os pacientes devem ter a certeza que os profissionais de saúde tomarão sempre a melhor decisão sobre o rumo do processo clínico, com base nas condições de saúde de cada um. Existem três situações que, pelo seu impacto na saúde, exigem atenção especial e são alvo de investigação, nomeadamente o stress ocupacional aliado ao erro médico, as variações na prática clínica e a medicina defensiva.

1.1.1. Stress Ocupacional e Erro Médico

Nos últimos anos, o stress ocupacional (Melo, Gomes, & Cruz, 1997) tem sido objecto de uma intensa investigação científica, sobretudo devido ao seu impacto negativo na saúde, quer a nível individual, quer a nível organizacional. Stress ocupacional é descrito como a soma de repostas físicas e psicológicas à ansiedade, motivada por questões de trabalho, apresentando um efeito negativo na saúde física e mental dos indivíduos, juntamente com uma redução da sua satisfação pessoal, produtividade e eficiência.

Os profissionais de saúde são particularmente afectados pelo stress ocupacional, pois lidam com os pacientes cara a cara e presenciam situações de extremo sofrimento, dor e morte. Pesquisas revelaram que existe um fenómeno de contágio emocional, quando os profissionais de saúde não conseguem evitar o envolvimento pessoal nos seus casos clínicos (Maslach, Schaufeli, & Leiter, 2001). Este contágio emocional é também motivado pelo sentido de responsabilidade por terceiros e contribui em grande medida para tornar o ambiente de trabalho das organizações de saúde, um dos mais tensos que existem. Os profissionais de saúde figuram

² O termo utilizado na literatura anglo-saxónica para descrever este tipo de guias é *Computer-Interpretable Guidelines* (CIGs).

entre os trabalhadores que trabalham mais horas, o que contribui para o aumento da exposição a situações potencialmente causadoras de *stress*. Os profissionais de saúde consultam um elevado número de pacientes todos os dias e é-lhes exigido que realizem uma gestão do tempo das suas consultas de forma a responder ao volume de pacientes. Uma vez que a gestão da prática clínica requer muita atenção da parte do profissional de saúde, este pode descurar a prática clínica em si, dando origem a pequenas negligências, que por sua vez, podem resultar em erro médico (Linzer, 2005). Segundo a Organização Mundial de Saúde (OMS), entre os países membros da União Europeia (UE), em média, ocorre erro médico em cerca de 8-12% das hospitalizações (OMS, 2010). Estes valores, apesar de não muito elevados, ganham uma dimensão maior no contexto dos cuidados de saúde, pois a ocorrência de um erro médico pode trazer consequências graves.

Para solucionar estes problemas, é necessário fornecer aos profissionais de saúde a informação que necessitam no momento do acto clínico, de forma a atenuar o seu desgaste físico, emocional e psicológico. Neste campo, a utilização de GCs está a revelar-se muito útil e a receptividade dos profissionais de saúde a estas ferramentas é elevada. Estudos realizados em organizações de saúde espalhadas pelo mundo apontam para que, em média, cerca de 75 % dos profissionais de saúde partilham da opinião de que as GCs são fontes de recomendações úteis, e cerca de 71 % concorda que as GCs são boas ferramentas de ensino direccionadas à melhoria da qualidade (Farquhar, Kofa, & Slutsky, 1992).

1.1.2. Variações na Prática Clínica

As variações na prática clínica podem dividir-se em variações justificadas e injustificadas (Hampton, 1997). As variações justificadas são aquelas que ocorrem devido a diferenças nos sistemas de saúde, diferenças nas características da população (e.g., sociodemográficas, culturais, estado de saúde, entre outras) ou devido às preferências do paciente ou profissional, no caso de haver mais de uma opção cientificamente válida. Este tipo de variações tem origem na liberdade inerente aos profissionais de saúde e não deve ser eliminada. As variações injustificadas são as que surgem quando controlados os factores citados anteriormente, que não trazem benefícios, são prejudiciais para os pacientes e traduzem-se numa redução da qualidade da prática clínica.

Na origem destas variações injustificadas está o stress ocupacional, a falta de exactidão dos dados recolhidos acerca dos pacientes e a tendência dos profissionais de saúde para seguirem a prática efectuada pela maioria, na tentativa de preencher vazios de conhecimento. Como consequência, há uma dificuldade acrescida de avaliação da qualidade dos cuidados de saúde, uma vez que as variações na prática clínica tornam impossível a definição de parâmetros de qualidade. Há portanto, uma necessidade de normalização e homogeneização dos cuidados de saúde, mantendo, contudo, a liberdade dos profissionais de saúde para tomar decisões, desde que cientificamente suportadas.

Para atenuar os efeitos das variações injustificadas na qualidade da prática clínica é necessário implementar mecanismos que permitam diferenciá-las das variações justificadas, e que apontem vazios de conhecimento relativamente aos resultados de determinados procedimentos. As GCs são um meio eficaz de promoção de boas práticas, pois enumeram os diferentes procedimentos clínicos disponíveis para o mesmo problema, melhorando o conhecimento acerca da eficácia destas soluções alternativas (Pearson, 1998).

1.1.3. Medicina Defensiva

A medicina defensiva (Studdert, et al., 2005) define-se como a decisão de um profissional de saúde de desviar-se do que se considera como boa prática, de forma a evitar eventuais queixas e processos jurídicos dos pacientes e respectivas famílias. A medicina defensiva positiva demonstra uma preocupação pelo bem-estar do paciente e em evitar os riscos de má prática clínica. Por seu lado, a medicina defensiva negativa passa por evitar determinados procedimentos e pacientes de alto risco de forma a preservar a integridade profissional. Estes actos nem sempre são conscientes e com o tempo impregnam-se de tal forma na prática clínica, que os profissionais já não se apercebem que assumem tais comportamentos. Na maioria dos casos, quanto maior a experiência do profissional de saúde, mais conservadora é a sua abordagem aos cuidados de saúde. Por sua vez, o aumento da prática de medicina defensiva leva a um aumento da despesa, sobretudo com os meios complementares de diagnóstico, acentuado pelos avanços tecnológicos dos últimos anos no campo das técnicas de exame. De acordo com a OMS, entre 1995 e 2009, os gastos com a saúde em Portugal subiram de 7.8% para 11% do Produto Interno Bruto (PIB) (OMS, 2011). O uso desnecessário de procedimentos

invasivos e medicação cria riscos significativos para os pacientes. Aliás, a prescrição de fármacos constitui cerca de 20-30% das despesas globais com a saúde.

De forma a solucionar este problema, é necessário incentivar o desenvolvimento e implementação de GCs em formatos que apresentem uma maior disponibilidade para os profissionais de saúde e permitam uma maior e mais fácil disseminação da medicina baseada na evidência (Toker, Shvarts, Perry, Doron, & Reuveni, 2004).

1.2. Objectivos

Este trabalho de dissertação apresenta como tema “Guias Clínicas: Representação e Raciocínio”. O seu principal objectivo é o desenvolvimento de um modelo de representação de GCs genérico que seja de fácil entendimento e que capte todas as dimensões da informação que estas contêm, assim como um modelo de raciocínio com base em GCs.

O trabalho foi projectado e desenvolvido com o intuito de aumentar a disponibilidade de GCs no momento do acto clínico, aumentando e melhorando o conhecimento dos profissionais de saúde, de forma a mitigar os efeitos do stress ocupacional na prática clínica e prevenir casos de erro médico, de variações na prática clínica e de medicina defensiva. Para o efeito, é necessário realizar um levantamento e análise dos modelos existentes de GCs para integração em sistemas de apoio à decisão. Da análise deve resultar um conjunto de características a utilizar no modelo a propor.

A decisão em cenários de Informação Imperfeita não tem sido uma prioridade na modelação de CIGs. Contudo, a Informação Imperfeita está presente no processo clínico e deve ser considerada na tomada de decisão no contexto de GCs. Desta forma, a incorporação de Informação Imperfeita no modelo de representação de guias também é um dos objectivos deste trabalho, assim como um método de avaliação da Qualidade da Informação (QoI) (Neves, Machado, Analide, Abelha, & Brito, 2007; Novais, Salazar, Ribeiro, Analide, & Neves, 2010), permitindo classificar as recomendações das guias de acordo com a sua solidez, quando aplicadas ao contexto de um paciente. Desta forma, será possível a definição de um modelo de tomada de decisão com base em GCs.

Os objectivos do presente trabalho de dissertação podem ser resumidos nas seguintes linhas mestras:

- Levantamento das principais abordagens à modelação de CIGs;

- Desenvolvimento de um formalismo de representação de CIGs, que responda às necessidades de informação do processo clínico;
- Incorporação de Informação Incompleta no modelo de representação de CIGs;
- Definição de um método de avaliação da QoI associada às recomendações das GCs;
- Definição de um modelo de processo de tomada de decisão, baseado em GCs;

1.3. Metodologia de Investigação

O trabalho da presente dissertação foi desenvolvido durante o último ano do Mestrado Integrado em Engenharia Biomédica. De forma a alcançar os objectivos definidos adoptou-se a metodologia de Acção/Investigação (Somekh, 2005). Inicialmente identificou-se um problema numa área com pouco desenvolvimento, elaborando-se posteriormente um plano para o desenvolvimento do trabalho nessa área, de forma a responder ao problema colocado.

Em seguida iniciou-se a pesquisa dos conceitos e projectos relevantes para o trabalho. A assimilação dos conceitos e projectos foi sujeita a uma constante renovação, à medida que surgiam novas ideias e informação. A última parte do trabalho consistiu no desenvolvimento de um modelo funcional que permitisse atingir os objectivos traçados.

A metodologia aplicada engloba as seguintes fases de desenvolvimento:

- Definição do problema e respectivas características;
- Actualização constante do estado da arte e objectivos do trabalho;
- Desenvolvimento de um protótipo que permita alcançar os objectivos definidos;
- Análise e correcção do protótipo com base nos resultados obtidos;
- Difusão do conhecimento e resultados obtidos com a comunidade científica.

1.4. Estrutura

A presente dissertação foi estruturada em sete capítulos, organizados da seguinte forma:

1. Introdução

No primeiro capítulo realiza-se uma descrição breve da situação actual e o respectivo enquadramento do trabalho, uma introdução aos principais conceitos e uma apresentação da

motivação, objectivos e da metodologia de investigação. Também é realizada uma breve descrição da estrutura da dissertação.

2. Clinical Guias

Este capítulo pretende fornecer uma visão sobre a evolução das GCs ao longo do tempo, apresentando também diferentes definições deste conceito segundo diferentes organizações espalhadas pelo mundo. Também se abordam as principais características de uma GC, os seus benefícios, limitações e a sua metodologia de desenvolvimento.

3. Guias Interpretáveis por Computador

O terceiro capítulo trata de CIGs, referindo os principais obstáculos á sua implementação, os benefícios da sua utilização e os principais aspectos do seu desenvolvimento. Estes aspectos são utilizados posteriormente para apontar as principais características de alguns formalismos de CIGs, como *Arden Syntax* (Clayton, Pryor, Wigertz, & Hripcsak, 1989), *Guideline Interchange Format (GLIF)* (Peleg, Boxwala, Bernstam, Tu, Greenes, & Shortliffe, 2001), *PROforma* (Fox, Johns, & Rahmanzadeh, 1998), *Asbru* (Miksch, Shahar, & Johnson, 1997), *EON* (Musen, Tu, Das, & Shahar, 1996) e *Standards-Based Sharable Active Guideline Environment (SAGE)* (Tu, et al., 2007). No final do capítulo realiza-se uma análise aos pontos fortes e fracos de cada abordagem.

4. Representação do Conhecimento

Neste capítulo sobre representação do conhecimento começa-se por realizar a distinção entre dados, informação e conhecimento. De seguida fornece-se uma definição de Informação Imperfeita e aborda-se algumas das técnicas mais utilizadas para a sua quantificação. A Programação em Lógica (PL) (McCarthy, 1959) e a Extensão à Programação em Lógica (EPL) (Analide, Novais, Machado, & Neves, 2006; Neves, Machado, Analide, Abelha, & Brito, 2007) são discutidas como formas de representação da Informação Imperfeita. Por fim aborda-se a Qol como método para tratar Informação Imperfeita e realiza-se uma análise aos assuntos abordados no capítulo.

5. Proposta de um Modelo de CIGs

Com base nas abordagens discutidas no capítulo quatro, propõe-se um modelo de representação de CIGs. Começa-se por apresentar as primitivas de representação e posteriormente aborda-se a representação ao nível computacional. Ao nível computacional apresenta-se a implementação da Informação Imperfeita no modelo, através da linguagem de representação e da forma de cálculo da QoI. Também se apresenta um motor de inferência capaz de tratar esta informação e discute-se um formato para partilha e validação de guias.

6. Caso de Estudo

Este capítulo trata da representação de uma guia para detecção e tratamento de elevados níveis de colesterol no modelo proposto. Inicialmente aborda-se alguns conceitos necessários ao entendimento da guia e realiza-se uma breve descrição da mesma. No seguimento, apresenta-se a estrutura, em termos de primitivas de representação da guia. A tomada de decisão é abordada em seguida, com exemplos de uma tarefa de decisão. No final do capítulo, realiza-se uma análise da representação obtida e uma comparação com as abordagens mencionadas no capítulo três.

7. Conclusões e Trabalho Futuro

Por fim, o último capítulo desta dissertação sintetiza o trabalho realizado e as principais conclusões a retirar. São também mencionadas as perspectivas de trabalho futuro.

2. Guias Clínicas

2.1. Introdução

Durante a última década muitos países começaram a demonstrar um crescente interesse no desenvolvimento e utilização de GCs, reconhecendo-as como ferramentas essenciais no melhoramento da qualidade dos cuidados de saúde.

Em 1977, o National Institute of Health, nos Estados Unidos iniciou um programa de desenvolvimento de GCs com base no consenso entre especialistas da área da saúde (Fink, Kosecoff, Chassin, & Brook, 1984). Este programa denominava-se Consensus Development Program e foi o primeiro programa do género a ser criado. Em 1980 foi a vez do Dutch Institute for Healthcare Improvement CBO, na Holanda, de iniciar o seu programa de desenvolvimento de GCs com base no consenso de especialistas, do qual resultou a primeira GC de transfusões sanguíneas, publicada em 1982. A Agency for Health Research and Quality (AHRQ) iniciou o seu programa de desenvolvimento de guias, desta vez baseado em evidências científicas, em 1989, prolongando-se até 1996 (Rosenbrand, Croonenborg, & Wittenberg, 2008). A partir de 1996 a AHRQ começou a produzir estudos e relatórios que serviriam de base a outros programas de desenvolvimento de guias. Desde meados dos anos 1990, muitas organizações a nível global iniciaram programas de desenvolvimento de GCs (Rosenbrand, Croonenborg, & Wittenberg, 2008), como por exemplo a Scottish Intercollegiate Guidelines Network (SIGN), o National Institute for Clinical Excellence (NICE), o New Zealand Guidelines Group (NZGG), o Guidelines Advisory Committee (GAC) no Canadá, a Current Care/Duodecim - Finnish Medical Society, a National Federation of Cancer Centers (FNCLCC) em França, entre outras (Rosenbrand, Croonenborg, & Wittenberg, 2008). Em Novembro de 2002 foi fundada a Guidelines International Network (G-I-N) (Ollenschlager, et al., 2004), uma iniciativa internacional que envolve organizações responsáveis pelo desenvolvimento de guias provenientes de todo o mundo, com o objectivo de disseminar o desenvolvimento de Guias Baseadas na Evidência. A G-I-N é constituída por cerca de 85 organizações, de 43 países. O Centro de Estudos de Medicina Baseada na Evidência (CEMBE) é a única organização portuguesa que figura entre os membros da G-I-N (G-I-N, 2011).

Inicialmente, as GCs baseavam-se no consenso de especialistas da área da saúde. Contudo, este processo possui as suas limitações, visto que inclui algumas mas não todas as perspectivas, podendo levar a conclusões erradas. A opinião de especialistas nem sempre reflecte o estado actual do conhecimento. As organizações dedicadas ao desenvolvimento de guias reconheceram a necessidade de realizar uma análise sistemática da bibliografia existente, de modo a evitar conclusões tendenciosas. Actualmente é prática comum, no desenvolvimento de guias, a identificação e síntese das melhores evidências científicas disponíveis (Lohr, Eleazer, & Mauskopf, 1998). Para além desta mudança de paradigma na criação de guias, a ideia de que apenas os profissionais de saúde deveriam envolver-se na sua criação começou a ficar ultrapassada. O processo de desenvolvimento passou a englobar não só administradores, mas também pacientes, adquirindo um carácter multidisciplinar e focando-se nas questões relacionadas com a implementação de GCs.

Este capítulo começa por apresentar uma definição de GC. De seguida são referidos as suas características, objectivos e benefícios. Por fim, aborda-se a metodologia de desenvolvimento de GCs utilizada pela generalidade das organizações.

2.2. Definição

O termo GC é muitas vezes utilizado de forma alternada com protocolo. No entanto, as GCs distinguem-se dos protocolos, na medida em que estes são mais específicos e contêm critérios rígidos que definem as medidas para uma única condição clínica, as indicações que fornecem são obrigatórias. O conteúdo dos protocolos vai mais de encontro à gestão da prática clínica, apresentando normas de conduta ética dos profissionais de saúde em determinadas situações. Por seu lado, as GCs auxiliam a tomada de decisão sobre os cuidados de saúde apropriados para uma dada condição e visto que estes cuidados podem mudar com os avanços no conhecimento médico, as GCs apresentam um carácter mutável (Rosenbrand, Croonenborg, & Wittenberg, 2008).

As GCs são ferramentas de decisão, que permitem encurtar a distância entre a prática clínica real e a prática clínica óptima. Contudo, são também descritas como mecanismos de melhoria da qualidade dos cuidados de saúde e de redução de custos, recomendações que visam influenciar decisões em intervenções clínicas e processos de implementação da medicina baseada na evidência (Rosenbrand, Croonenborg, & Wittenberg, 2008).

Segundo o Institute of Medicine, nos Estados Unidos, as GCs são documentos desenvolvidos de forma sistemática para aconselhar os prestadores de cuidados e os pacientes acerca dos cuidados de saúde apropriados em circunstâncias clínicas específicas (Mead, 2000). A definição do Dutch Institute for Healthcare Improvement CBO acrescenta à anterior que as GCs são baseadas nos resultados de investigação científica, seguida de discussão e expressão da opinião de especialistas, de forma a tornar a prática médica mais eficaz, eficiente e transparente (Rosenbrand, Croonenborg, & Wittenberg, 2008). Por sua vez, o NZGG apresenta uma definição mais alargada de GC, encarando-a como um documento que fornece orientação em diferentes níveis de interacção.

Para além de uma definição alargada de GC, o NZGG define dois tipos diferentes de guia consoante o seu processo de desenvolvimento (Rosenbrand, Croonenborg, & Wittenberg, 2008). Os diferentes tipos de guia definidos são:

- Guia Baseada em Consenso: é a forma mais comum de guias, resultante da concordância de opiniões entre especialistas;
- Guia Baseada na Evidência: trata-se de uma guia desenvolvida após a extracção e revisão de informação científica a partir da bibliografia, distinguindo o que é prova científica do que é opinião. Para além de aconselhar qual de dois tratamentos é o mais indicado, quantifica em termos absolutos os benefícios e malefícios que advêm da escolha de um ou outro.

As Guias Baseadas na Evidência são mais valiosas para os prestadores de cuidados de saúde e para os pacientes, pois as suas recomendações são baseadas em provas científicas e suportam decisões passíveis de produzir melhores resultados.

2.3. Características

A designação GC é frequentemente atribuída de forma errada a documentos como protocolos ou artigos de opinião. Contudo existe um conjunto de características que distinguem as GCs dos restantes documentos do género, e cuja presença é utilizada para avaliação da qualidade de uma guia (Rosenfeld & Shiffman, 2006). De seguida refere-se os elementos que caracterizam as guias e que são utilizados como parâmetros de qualidade:

- Definição clara do âmbito e objectivos da guia: trata-se da especificação dos objectivos gerais da guia, das questões clínicas que esta abrange e dos pacientes aos quais deve

ser aplicada. Sem a definição dos objectivos e âmbito da guia é impossível avaliar a sua eficácia;

- Envolvimento de todas as partes interessadas e intervenientes: o grupo de desenvolvimento das guias é heterogéneo, reunindo indivíduos de todas as áreas profissionais relevantes, inclusive pacientes. A guia é criada especificamente para os utilizadores, de forma a responder a todas as suas necessidades;
- Rigor no desenvolvimento: a procura e selecção de evidências científicas são realizadas segundo métodos sistemáticos, e os métodos de formulação de recomendações são descritos de forma clara. Todas as recomendações são baseadas em provas científicas e consideram os benefícios, efeitos colaterais e riscos que podem provocar. A guia é revista por um grupo de especialistas, externo à organização que as desenvolve, antes da sua publicação e os procedimentos para a sua actualização são definidos previamente;
- Clareza na apresentação: as recomendações são específicas e não possuem ambiguidade e as diferentes opções são apresentadas de forma clara. As recomendações essenciais são facilmente identificáveis e a guia é suportada por ferramentas que viabilizam a sua aplicação;
- Aplicabilidade: as barreiras organizacionais à aplicação da guia são discutidas e são considerados os custos da sua implementação. Também são definidos critérios de auditoria e revisão;
- Independência editorial: as GCs, que forem financiadas por dinheiro externo, devem frisar que as suas recomendações são totalmente independentes da organização financiadora e, no caso de existirem conflitos de interesses da parte dos membros do grupo de desenvolvimento, estes devem ser esclarecidos e incluídos no documento da guia.

Como resultado da Conference on Guideline Standardization (COGS) (Shiffman, Shekelle, Overhage, Slutsky, Grimshaw, & Deshpande, 2003), realizada em 2002, surgiu a COGS Checklist. A COGS Checklist é uma ferramenta desenvolvida para identificar características que permitam verificar a qualidade de uma GC. As guias que satisfazem os padrões de qualidade definidos na COGS Checklist são incluídas na base de dados da National Guideline Clearinghouse (NGC), uma iniciativa da AHRQ. As características que permitem a inclusão de uma GC nesta base de dados, vão de encontro às referidas acima, acrescentando que o

documento da guia deve estar disponível em língua inglesa e deve haver documentação disponível que ateste que a guia foi revista nos últimos 5 anos.

2.4. Objectivos das Guias Clínicas

As GCs são desenvolvidas com o objectivo de resumir e sintetizar o conhecimento e inovações na área da medicina. Desta forma, as GCs têm como função descrever os cuidados de saúde mais apropriados a uma situação clínica, com base nas melhores evidências científicas e no consenso de profissionais de saúde (Carnett, 1999).

Também são objectivos das GCs a redução das variações injustificadas na prática clínica, a promoção da medicina baseada na evidência e a satisfação da necessidade de transparência e responsabilização, de forma a melhorar a qualidade dos cuidados de saúde. As GCs devem servir de referência, de forma actuarem como ferramentas educacionais na área da saúde. Entre o conjunto de tarefas das GCs, encontra-se a promoção da utilização eficiente de recursos, a definição de mecanismos de avaliação da qualidade e auditoria, o destaque de vazios de conhecimento na bibliografia científica actual e a sugestão de campos de investigação futura (Miller & Kearney, 2004).

2.5. Benefícios e Limitações

2.5.1. Para os Profissionais de Saúde

O principal benefício das GCs para os profissionais de saúde é a melhoria da qualidade dos cuidados prestados (Woolf, Grol, Hutchinson, Eccles, & Grimshaw, 1999). É difícil para os profissionais de saúde ter tempo para procurar, recolher, avaliar e interpretar resultados de investigação, ao mesmo tempo que desenvolvem e aperfeiçoam a sua prática clínica. Como tal, as GCs funcionam como um veículo que faz chegar conhecimento actualizado aos profissionais de saúde, mantendo-os ao corrente de novos desenvolvimentos (Duff, Kitson, Seers, & Humphris, 1996).

No entanto, GCs fracas podem apresentar um impacto negativo, ao fornecerem indicações ineficazes ou prejudiciais (Woolf, Grol, Hutchinson, Eccles, & Grimshaw, 1999). Estas guias normalmente resultam de uma construção pobre e uma implementação inadequada. A concepção e utilização adequadas de GCs consomem recursos significativos, pelo que uma

gestão ineficiente de todo o processo de desenvolvimento e implementação terá consequências económicas, que por sua vez afectarão o orçamento da organização responsável. A interpretação do conteúdo de uma guia pode ser problemático, pois o significado exacto dos seus termos nem sempre está definido e as recomendações nem sempre se encontram bem articuladas.

As guias desempenham um papel importante no suporte à qualidade, à auditoria e à investigação, através de um conjunto de especificações e requisitos utilizados na avaliação dos cuidados de saúde (Woolf, Grol, Hutchinson, Eccles, & Grimshaw, 1999). Isto permite afirmar que as GCs possibilitam a medição dos cuidados de saúde, de forma a avaliar o seu grau de adequação.

Existe também o risco de as GCs levarem a uma prática médica inflexível. A liberdade da prática quotidiana de um profissional de saúde pode ficar restringida por uma confiança excessiva em guias com um carácter mais autoritário (Siriwardena, 1995). Pode haver uma pressão no sentido de moldar rigorosamente o comportamento dos profissionais de saúde à semelhança das directrizes das guias, suprimindo assim a liberdade clínica.

Através da observação é possível constatar que a maior parte das guias encontra-se escrita sob a forma de documentos longos em formato de texto. Estes documentos, por vezes são de difícil leitura, representando um obstáculo para os profissionais de saúde, dada a forma desestruturada como a informação é apresentada (Mancia, et al., 2007). O formato em papel também não favorece a consulta da guia no momento do acto clínico e torna difícil a integração e aplicação das guias no processo de consulta do paciente. Ao nível da manutenção, as operações de actualização e revisão são problemáticas, pois podem obrigar a reescrever documentos inteiros.

2.5.2. Para os Pacientes

As GCs, por vezes, são acompanhadas de versões para os pacientes, na forma de folhetos ou vídeos, que os informam sobre os serviços que devem esperar que os profissionais de saúde executem e sobre os benefícios e limitações das diferentes opções de tratamento. Este aspecto é extremamente positivo, pois o processo clínico torna-se mais transparente para os pacientes, e estes passam a ter um papel activo na tomada de decisão (Mead, 2000). Contudo, se as guias apresentarem falhas na concepção, fornecem informações desajustadas, o que pode dificultar a comunicação entre os pacientes e os profissionais de saúde (Woolf, Grol, Hutchinson, Eccles, & Grimshaw, 1999).

Por outro lado, as GCs podem beneficiar os pacientes através de intervenções na política de saúde pública, por exemplo, ao chamarem a atenção para populações carentes e problemas de saúde que normalmente são negligenciados. Deste modo, o sistema de saúde evolui no sentido de disponibilizar novos e melhores serviços para os mais necessitados (Woolf, Grol, Hutchinson, Eccles, & Grimshaw, 1999).

2.5.3. Para as Organizações de Saúde

As GCs aumentam a eficiência e otimizam a relação benefício/custo da prática clínica (Duff, Kitson, Seers, & Humphris, 1996). A sua implementação ajuda a reduzir custos hospitalares, por exemplo, em relação a despesas com fármacos, cirurgias ou outros procedimentos, evitando o desperdício de recursos. Contudo, seguir guias com falhas pode comprometer a eficiência e resultar em intervenções financeiras dispendiosas. Todavia, publicitar a adesão a GCs tem o potencial de melhorar a imagem pública de hospitais, o que se traduz num maior apoio político da parte de órgãos oficiais.

2.6. Metodologia de Desenvolvimento

Cada um dos países membros da G-I-N possui a sua própria metodologia para o desenvolvimento de guias baseadas na evidência. No entanto, a maioria das metodologias de desenvolvimento rege-se por princípios semelhantes e não apresenta divergências significativas (Rosenbrand, Croonenborg, & Wittenberg, 2008; Rosenfeld & Shiffman, 2006). A Figura 1 representa o ciclo de vida geral de uma GC, desde a concepção até à implementação. De seguida apresenta-se os principais passos da metodologia de desenvolvimento de uma guia. Em seguida aborda-se cada uma das fases de desenvolvimento de guias.

2.6.1. Definição do Tema

Nesta fase propõe-se um tema para a guia, com base nos problemas que motivam o seu desenvolvimento (Rosenfeld & Shiffman, 2006). Os temas para o desenvolvimento de guias devem ser importantes e concretizáveis.

Os temas importantes apresentam um carácter multidisciplinar e estão relacionados com procedimentos clínicos muito utilizados e condições de saúde vinculadas a doenças e

custos económicos. As GCs podem ser desenvolvidas para uma grande variedade de assuntos, incluindo condições de saúde (e.g., sinusite, dor de garganta, perda de audição, entre outras) e procedimentos clínicos (e.g., amigdalectomia, septoplastia, timpanoplastia, entre outros). Um comité de direcção, que inclua representantes da organização responsável pelo desenvolvimento da guia e representantes das partes interessadas, deve identificar o tema da guia.

Os temas concretizáveis têm como base evidências científicas de qualidade, assentes em ensaios clínicos controlados. De preferência deve existir uma revisão sistemática sobre o tema, de forma a fornecer definições claras da condição de saúde ou procedimento em questão.

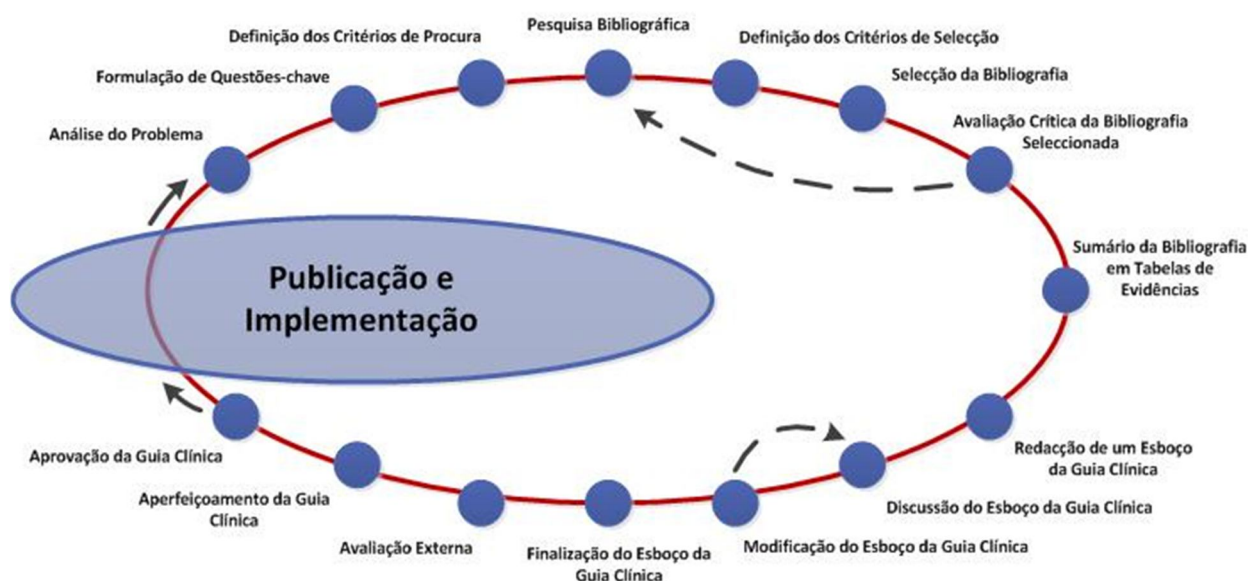


Figura 1 -- Representação do ciclo de vida de uma GC (adaptado de (Rosenbrand, Croonenborg, & Wittenberg, 2008)).

2.6.2. Composição do Grupo de Desenvolvimento de Guias Clínicas

O processo de desenvolvimento de uma guia deve incluir a participação de representantes dos grupos e das disciplinas afectadas pela guia (Rosenbrand, Croonenborg, & Wittenberg, 2008; Rosenfeld & Shiffman, 2006). Com base nos resultados da fase anterior, constitui-se um grupo de trabalho com representantes das áreas englobadas pelo tema. As GCs devem ser desenvolvidas por médicos, em colaboração com os representantes daqueles que serão afectados pelas intervenções que nelas constam, incluindo pacientes, membros das

administrações das organizações de saúde e outros profissionais de saúde. O estabelecimento de um grupo multidisciplinar de desenvolvimento de guias requer:

- Todos os grupos relevantes representados, de modo a fornecer o conhecimento e as capacidades necessários a todas as fases do processo clínico;
- Todas as evidências científicas localizadas e avaliadas de forma crítica;
- Os problemas da aplicação prática das guias identificados e tratados;
- A confirmação da credibilidade da guia pelas partes interessadas e a sua cooperação na implementação.

Os grupos de desenvolvimento de guias variam em tamanho, dependendo da área de abrangência do tema, mas geralmente apresentam entre 15 a 25 membros (Rosenbrand, Croonenborg, & Wittenberg, 2008). Deve haver uma preocupação em incluir membros de uma área geográfica distribuída, com representantes dos diversos locais onde a guia será implementada.

2.6.3. Análise do Problema

Após a constituição do grupo de desenvolvimento deve discutir-se quais os assuntos, no âmbito do tema, que a guia deve abordar (Rosenbrand, Croonenborg, & Wittenberg, 2008). Esta análise consiste num conjunto de entrevistas e questionários às partes interessadas e grupos de pacientes. Nesta fase de análise do problema é necessário procurar outras guias que possam existir acerca do tema em questão. A existência de outras guias com o mesmo tema não invalida a criação de uma nova, uma vez que as existentes podem encontrar-se desactualizadas. As guias encontradas podem servir como ponto de partida para a discussão em grupo, podendo encontrar-se vazios de conhecimento que devem ser corrigidos na guia em criação.

Desta fase deve resultar um conjunto de tópicos de carácter clínico e organizacional a abordar.

2.6.4. Formulação de Questões-chave

O grupo de desenvolvimento deve formular um conjunto de questões-chave estruturadas a partir dos tópicos obtidos na fase de análise do problema, que identificam de forma inequívoca a população em investigação (população alvo do diagnóstico ou intervenção), o tipo de controlo utilizado e as medidas utilizadas para avaliar a eficácia das intervenções (Rosenbrand,

Croonenborg, & Wittenberg, 2008) (Rosenfeld & Shiffman, 2006). Estas questões serão os critérios utilizados na pesquisa bibliográfica.

2.6.5. Pesquisa Bibliográfica

A pesquisa bibliográfica deve focar-se nas melhores evidências disponíveis para abordar cada uma das questões-chave e assegurar uma ampla cobertura de estudos, de preferência dos mais relevantes (Rosenbrand, Croonenborg, & Wittenberg, 2008). Os tipos de estudos científicos incluídos nesta fase são os seguintes:

- Revisões sistemáticas;
- Ensaio clínico aleatório;
- Estudos observacionais;
- Estudos de diagnóstico;
- Estudos económicos;
- Estudos de qualidade;
- Guias Clínicas.

O grupo de desenvolvimento deve definir algumas restrições na procura inicial, como por exemplo se deve ou não privilegiar trabalhos publicados em detrimento de trabalhos não publicados ou data mínima para as publicações.

De modo a minimizar tendências e assegurar uma cobertura adequada da bibliografia relevante, a pesquisa bibliográfica deve incluir recursos de pesquisa diversificados, como bases de dados automáticas (e.g., MEDLINE), fontes bibliográficas (e.g., Science Citation Index) e artigos de conferências.

2.6.6. Avaliação Crítica da Bibliografia Seleccionada

Uma vez seleccionadas as potenciais fontes de evidência para o desenvolvimento de uma guia, avalia-se a metodologia utilizada em cada um, de forma a assegurar a sua validade (Rosenbrand, Croonenborg, & Wittenberg, 2008). A avaliação da metodologia utilizada em estudos baseia-se num conjunto de questões que se focam nos aspectos de concepção do estudo, e que influenciam significativamente os resultados obtidos e respectivas conclusões. Estas questões variam de acordo com o tipo de estudo. É utilizado um conjunto de listas de verificação com as questões, de forma a conferir consistência a todo o processo.

Normalmente, após a avaliação da bibliografia seleccionada, o grupo de desenvolvimento aperfeiçoa os critérios de procura e procede a nova procura bibliográfica.

2.6.7. Resumo da Bibliografia em Tabelas de Evidências

As tabelas de evidências devem resumir todos os estudos identificados na procura bibliográfica e que foram considerados válidos na avaliação crítica, sendo apresentados num formato *standard* para facilitar a comparação de resultados entre estudos (Rosenbrand, Croonenborg, & Wittenberg, 2008; Rosenfeld & Shiffman, 2006). As tabelas apresentam as evidências que suportam cada um dos resultados dos estudos e são um componente essencial no registo de desenvolvimento de guias, assegurando que a base das recomendações do grupo de desenvolvimento é transparente.

2.6.8. Redacção de um Esboço da Guia Clínica

Nesta fase, o grupo de desenvolvimento deve redigir um esboço da guia (Rosenbrand, Croonenborg, & Wittenberg, 2008) (Rosenfeld & Shiffman, 2006). Tipicamente o formato de uma guia inclui os seguintes elementos:

- Questões-chave às quais a guia deve responder;
- Tabelas de evidências utilizadas no desenvolvimento;
- Conclusões, incluindo a qualidade das evidências;
- Relevância clínica da guia;
- Perspectiva do paciente em relação à guia;
- Consequências da sua implementação ao nível das organizações de saúde;
- Recomendações;
- Referências.

2.6.9. Classificação das Evidências

A *qualidade* de uma guia depende da qualidade das evidências e opiniões que lhe servem de base. Nem todas as evidências que são utilizadas numa guia apresentam a mesma solidez e qualidade (Rosenbrand, Croonenborg, & Wittenberg, 2008) (Rosenfeld & Shiffman, 2006). Para haver transparência na criação de guias, as evidências são classificadas. Desde os anos 1970,

foram empregues vários sistemas de classificação da qualidade das evidências e da fiabilidade das recomendações, por diversas organizações. Não existe um sistema unificado, o que leva a que organizações diferentes utilizem diferentes sistemas para classificar a qualidade das evidências e a fiabilidade das recomendações.

No ano 2000, iniciou-se uma colaboração informal de indivíduos com interesse em abordar as limitações existentes dos sistemas de classificação da altura. Este projecto denomina-se Grading of Recommendations Assessment, Development and Evaluation (GRADE) (Guyatt, et al., 2011) e o seu objectivo é o desenvolvimento de uma abordagem à classificação da qualidade das evidências, que seja sensível e comum a todas as organizações. Para alcançar transparência e simplicidade, o sistema GRADE propõe quatro níveis de classificação da qualidade das evidências: *high*, *moderate*, *low* e *very low*. Algumas das organizações que utilizam o sistema GRADE optaram por combinar os níveis *low* e *very low*, num só. Evidências baseadas em estudos aleatórios controlados começam o processo de classificação com o nível de qualidade *high*, contudo a confiança na evidência pode baixar devido a diferentes motivos, como por exemplo:

- Limitações do estudo;
- Inconsistência dos resultados;
- Falta de precisão;
- Parcialidade do estudo.

Os estudos observacionais (e.g., estudos de caso controlo) iniciam o processo de classificação no nível *low*, podendo subir caso a magnitude do efeito do tratamento seja muito elevada ou existam indícios de uma forte correlação entre a dose administrada e resposta ao tratamento.

A classificação das recomendações da guia resulta de um conjunto de factores como o nível de qualidade das evidências, certeza no balanço entre efeitos desejáveis e indesejáveis, certeza da utilização eficiente dos recursos, entre outros.

O sistema GRADE tem sido implementado por diversas organizações, apesar de ainda se encontrar em desenvolvimento.

2.6.10. Outras Considerações

É raro as evidências mostrarem de forma clara e objectiva quais as acções que devem ser recomendadas para todas as situações. Consequentemente, nem sempre é claro para os indivíduos que não estiveram envolvidos no processo de desenvolvimento da guia, como foram alcançadas as recomendações. De forma a resolver este problema, o grupo de desenvolvimento elabora um resumo sobre a sua perspectiva em relação às evidências e como estas originaram as recomendações da guia (Rosenbrand, Croonenborg, & Wittenberg, 2008). Isto pode levar a constantes modificações no esboço da guia, até chegar a um esboço final.

2.6.11. Avaliação Externa

O esboço da guia deve ser avaliado por entidades independentes da organização responsável pelo seu desenvolvimento (Rosenbrand, Croonenborg, & Wittenberg, 2008; Rosenfeld & Shiffman, 2006). Estas avaliações externas podem ser realizadas através de conferências nacionais e internacionais e associações médicas.

As revisões ocorrem durante o processo de desenvolvimento da guia e proporcionam ao grupo de desenvolvimento a oportunidade de apresentar as suas conclusões preliminares e esboço da guia a uma audiência mais vasta. Deste modo, o grupo de desenvolvimento obtém um parecer do seu trabalho, na forma de sugestões sobre novas evidências a acrescentar à guia ou interpretações alternativas das evidências apresentadas. O facto de entidades de uma área geográfica mais alargada reverem a guia, contribui para uma maior aceitação e disseminação da mesma.

2.6.12. Aprovação e Publicação da Guia Clínica

Após a fase de avaliação externa, a guia está finalizada. A versão final da guia é enviada a todas as sociedades científicas que participaram no seu desenvolvimento para aprovação oficial (Rosenbrand, Croonenborg, & Wittenberg, 2008; Rosenfeld & Shiffman, 2006).

Por fim, a guia é publicada. Normalmente utiliza-se o formato de documento electrónico, disponibilizado no sítio *web* da organização responsável pelo desenvolvimento da guia.

2.7. *Living Guidelines*

O cenário actual de desenvolvimento, disseminação e implementação de GCs depara-se com o problema da sua desactualização. A maioria das guias é actualizada apenas uma vez a cada 5 anos, o que contrasta com a rapidez do crescimento do conhecimento científico. Neste momento, as GCs são documentos estáticos, cuja modificação é difícil. Este problema coloca um novo desafio à comunidade científica. Da necessidade de resposta a este desafio, surgiu o conceito de *living guideline* (Kaiser & Miksch, 2009; Seyfang, et al., 2007).

Living guidelines são GCs que são actualizadas de forma contínua, mais flexíveis e com maior adaptabilidade. São guias que fornecem conhecimento actualizado aos profissionais de saúde. Para tornar isto possível, as guias devem apresentar uma estrutura modular, para que apenas uma parte da guia possa ser ajustada, evitando a revisão de todo o documento. O processo de desenvolvimento também deve ser alvo de alterações, pois algumas das guias são criadas numa narrativa desestruturada. A utilização de apoio informático pode trazer benefícios ao processo de desenvolvimento, visto que as ferramentas informáticas dependem de representações formais e estruturadas e podem ser utilizadas para responder aos desafios colocados em todas as fases do ciclo de vida de uma guia. Em certos casos, as GCs podem ter múltiplas interpretações, dado o seu elevado grau de complexidade. Consequentemente, podem apresentar um carácter ambíguo, incompleto e inconsistente. Para corrigir este problema, é necessário um modelo de representação de GCs, que capte todas as necessidades de informação da guia e ao mesmo tempo permita a sua modelação de uma forma estruturada.

3. Guias Interpretáveis por Computador

3.1. Introdução

Como resposta aos desafios colocados pelo formato em papel de GCs surgiram as CIGs (Kaiser & Miksch, 2009; Sonnenberg & Hagerty, 2006). As CIGs são representações em formato informático de GCs, que são integradas em sistemas informáticos de apoio à decisão e visam facilitar a implementação de guias na prática clínica diária, uma vez que se apresentam disponíveis no momento do acto clínico. A implementação de guias em sistemas de apoio à decisão pode levar a uma melhoria da aceitação e aplicação de guias na prática diária, pois estes sistemas são capazes de monitorizar as acções e observações dos profissionais de saúde.

O capítulo começa por referir os obstáculos à implementação de CIGs e em seguida aponta os principais benefícios da sua utilização. Depois resume os principais aspectos do desenvolvimento de CIGs. De seguida aborda-se as principais abordagens neste domínio. Por fim realiza-se uma reflexão acerca dos modelos de representação das diferentes abordagens.

3.2. Obstáculos à Implementação de CIGs

O desenvolvimento de SADCs com base em formatos informáticos de guias tem sido proposto como estratégia de promoção da implementação de GCs. Contudo, existem alguns obstáculos à implementação de sistemas informáticos baseados em guias.

A conversão de GCs em algoritmos informáticos a partir das suas versões em texto não é uma tarefa fácil, pois estas versões não foram originalmente concebidas para serem interpretáveis por computadores e em alguns casos contêm instruções complexas, que manipulam demasiadas variáveis e que dificilmente se traduziriam em algoritmos eficientes (Chim, Cheung, Fung, & Wong, 2003; Tierney, et al., 1995).

Por vezes, o vocabulário utilizado nos documentos possui palavras evasivas para quantificar medidas, em lugar de limites numéricos, e os critérios nos pontos de decisão nem sempre são explícitos e indicam o que fazer. A falta de precisão dos conceitos dá lugar a ambiguidade e vazios de conhecimento, com os quais os computadores não conseguem lidar

(Chim, Cheung, Fung, & Wong, 2003; Tierney, et al., 1995). Quanto maior a simplicidade e assertividade de uma guia, mais fácil será a sua adaptação ao formato de CIGs.

A integração de sistemas baseados em guias com Registos de Saúde Electrónicos (RSEs) também é uma tarefa complexa, pois cada RSE possui a sua própria forma de organização da informação, que raramente coincide com a forma como a guia organiza a informação (Peleg, Keren, & Denekamp, 2008). O RSE pode ter uma orientação:

- Cronológica: os dados clínicos do paciente e as observações são registados de forma cronológica;
- Por origem: os dados clínicos são orientados consoante a sua origem, ou seja a proveniência da informação determina a sua catalogação e conseqüente registo;
- Orientação por problema: organização da informação por problema de saúde;

Deve haver um mapeamento dos conceitos do registo clínico para os conceitos da CIG, o que implica uma grande capacidade de adaptação do formato informático de guias que se pretende utilizar.

3.3. Benefícios da Implementação de CIGs

Uma possível solução para os problemas acima mencionados é o desenvolvimento de um modelo *standard* de representação de GCs. Um modelo formal para representação de guias fornece um conhecimento aprofundado dos processos clínicos que constam em GCs. A utilização de CIGs traz benefícios (Elkin, et al., 2000), tais como:

- Um modelo de representação de guias pode ser utilizado para identificar os diferentes requisitos que se devem verificar para que possa haver uma decisão, auxiliando os profissionais de saúde neste momento crítico do processo clínico;
- As CIGs trazem também benefícios ao nível da verificação e validação de GCs, uma vez que possibilitam o desenvolvimento de mecanismos automáticos para realizar estas tarefas;
- A aplicação de um modelo formal de representação ajuda a assegurar a validade sintáctica, lógica e clínica de uma guia;
- A reutilização do conhecimento também é uma mais-valia das CIGs. Se o modelo utilizado organizar uma GC em módulos, cada um contendo fragmentos de conhecimento, torna-se fácil incluir estes fragmentos de conhecimento em outras guias

e até referenciar uma guia mais específica no contexto de uma guia com maior abrangência;

- A disseminação de guias também fica facilitada com um formato informático, assim como a sua actualização e modificação, através de mecanismos automáticos que realizam este tipo de operações.

3.4. Principais Aspectos do Desenvolvimento de CIGs

Identificaram-se quatro áreas de grande importância na concepção de sistemas informáticos de apoio à decisão baseados em guias (Clercq, Blom, Korsten, & Hasman, 2004), que devem ser consideradas no desenvolvimento do modelo de CIGs a utilizar. As áreas a considerar são as seguintes:

- Modelação e representação de guias;
- Aquisição de guias;
- Verificação e teste de guias;
- Execução de guias.

Cada uma destas áreas possui um conjunto de aspectos que servem de parâmetros de análise das diferentes abordagens à modelação de CIGs. De seguida abordar-se-á cada um destes aspectos.

3.4.1. Modelação e Representação de Guias Clínicas

O modelo é a característica central de cada abordagem (Clercq, Blom, Korsten, & Hasman, 2004; Peleg, et al., 2003). A questão de como representar uma guia é crucial no desenvolvimento de um modelo de CIGs, o qual deve fornecer uma compreensão aprofundada dos procedimentos clínicos que constam numa guia, uma descrição precisa e inequívoca da guia e meios para que *parsers* a possam analisar, possibilitando a sua execução por motores de execução automáticos. Para alcançar estes objectivos, há um conjunto de aspectos que devem ser abordados, os quais são referidos em seguida:

- Primitivas: são unidades de construção que são utilizadas para montar uma guia (e.g., tarefas, regras, nodos, entre outros). A maioria das abordagens modela as guias segundo um Task Network Model (TNM), modelação hierárquica do fluxo de trabalho

segundo uma rede de tarefas, podendo adquirir diferentes formas, nomeadamente a de fluxograma. Os TNMs normalmente baseiam-se num conjunto de tarefas genéricas, tais como decisões e acções;

- Complexidade: a representação deve ser capaz de representar diferentes tipos de guias que possam diferir entre si em termos de complexidade e níveis de abstracção, por exemplo, através da nidificação de guias ou decomposição das mesmas em módulos;
- Tipos de conhecimento: as guias possuem diferentes tipos de conhecimento, nomeadamente o conhecimento declarativo (e.g., o conhecimento específico do domínio) e o conhecimento processual (e.g., o método de inferência ou o método de apoio à decisão). O modelo deve apresentar expressividade suficiente para suportar estes tipos de conhecimento, de forma que as guias possam ser utilizadas em múltiplos domínios clínicos.
- Linguagem: a representação deve ser suportada por uma linguagem formal (vocabulário, sintaxe e semântica). A linguagem deve ser compacta e objectiva, de forma a possibilitar a motores de execução automáticos a execução atempada da guia no momento do processo clínico.
- Adaptação local: é importante que o modelo seja suficientemente genérico, que permita a partilha de guias entre diferentes instituições. Para o efeito, é necessária a existência de mecanismos de actualização de guias que disseminem as alterações até às suas adaptações locais.

3.4.2. Aquisição de Guias Clínicas

O processo de aquisição do conhecimento contido em GCs é fundamental no desenvolvimento de CIGs (Clercq, Blom, Korsten, & Hasman, 2004). Este processo deve basear-se no modelo de representação adoptado e deve ser auxiliado por ferramentas informáticas de aquisição de conhecimento. Estas ferramentas auxiliam os especialistas do domínio de conhecimento da guia a formularem e estruturarem o conhecimento segundo o modelo subjacente. Os mecanismos de actualização necessários para a criação de novas versões de guias também são da competência das ferramentas de aquisição.

3.4.3. Verificação de Guias Clínicas

A precisão, a correcção sintáctica e a correcção semântica de uma CIG são de extrema importância na sua aceitação pelos profissionais de saúde e na sua integração na prática clínica (Clercq, Blom, Korsten, & Hasman, 2004). Por conseguinte, é necessário realizar testes de verificação com o objectivo de assegurar a integridade de uma guia.

Estes testes garantem que não existem incorrecções numa guia, socorrendo-se para o efeito a ambientes de simulação, nos quais a guia é aplicada a diferentes casos clínicos, fornecidos por registos existentes de pacientes.

3.4.4. Execução de Guias Clínicas

A criação de SADCs, baseados em guias, assenta na premissa de que estas são codificadas num formato interpretável por *parsers*, integrados em motores de execução automáticos (Clercq, Blom, Korsten, & Hasman, 2004; Isern & Moreno, 2008).

O tempo de execução de uma guia deve ser tão reduzido quanto possível, de forma a responder às necessidades dos profissionais de saúde, no momento do acto clínico. A arquitectura do motor de execução não deve depender do sistema e da aplicação utilizados, pois isso impossibilitaria a sua utilização em diferentes domínios clínicos.

3.5. Abordagens à Modelação de CIGs

Pode considerar-se que a primeira abordagem à modelação de CIGs, embora rudimentar, foi a implementada no sistema HELP (Gardner, Pryor, & Warner, 1999). Este sistema foi desenvolvido no LDS Hospital, na cidade de Salt Lake, nos Estados Unidos. O sistema tornou-se operacional em 1967, mantendo-se até aos dias de hoje. A componente de CIGs do sistema consistia num conjunto de módulos, denominados *help sectors*, que continham o conhecimento clínico na forma de regras lógicas. Deste então, o desenvolvimento de abordagens de CIGs proliferou, sendo predominante nas duas últimas décadas, como é possível observar na Figura 2. Contudo, nenhuma das abordagens desenvolvidas se afirmou como um *standard* a utilizar em SADCs.

De seguida serão discutidas diferentes abordagens à modelação de CIGs, dando especial atenção ao modelo de representação de cada uma, visto ser este o aspecto que influencia todos os outros acima mencionados e que faz parte dos objectivos da presente dissertação. A selecção

das abordagens seguiu dois critérios: consenso dos autores sobre quais as mais relevantes e a disponibilidade de literatura sobre as abordagens. As abordagens seleccionadas foram: Arden Syntax (Clayton, Pryor, Wigertz, & Hripcsak, 1989), Guideline Interchange Format (GLIF) (Peleg, Boxwala, Bernstam, Tu, Greenes, & Shortliffe, 2001), PROforma (Fox, Johns, & Rahmzadeh, 1998), Asbru (Miksch, Shahar, & Johnson, 1997), EON (Musen, Tu, Das, & Shahar, 1996) e Standards-Based Sharable Active Guideline (SAGE) (Tu, et al., 2007).

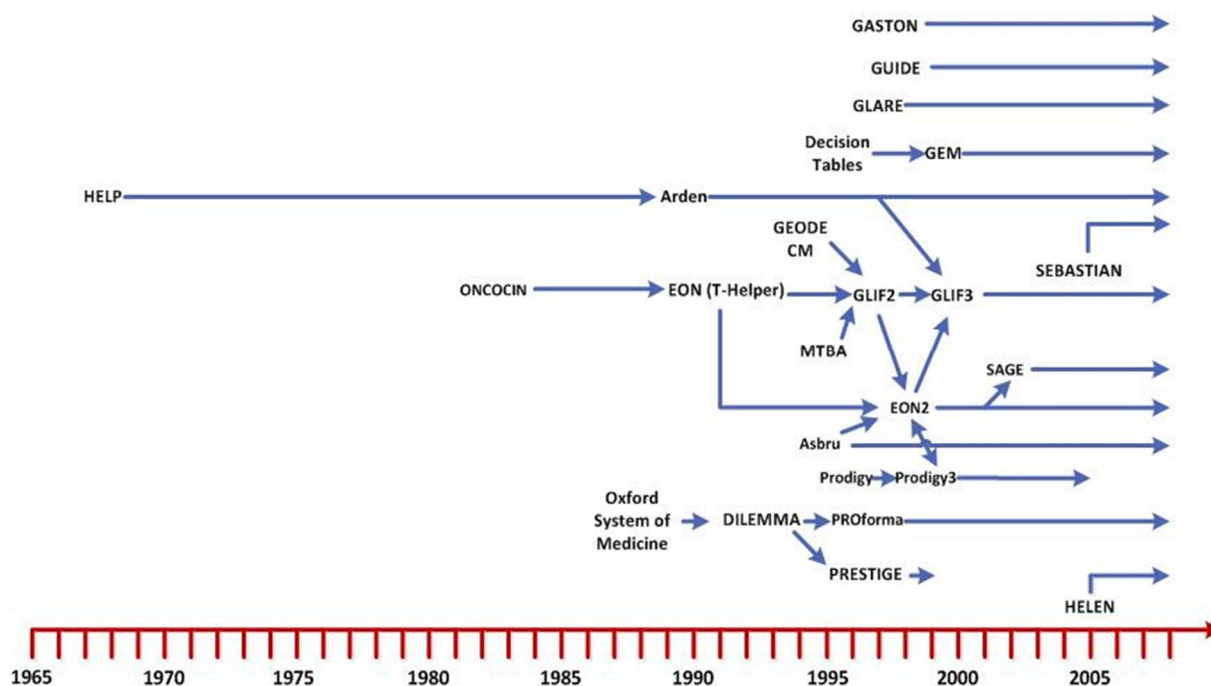


Figura 2 – Representação das diferentes abordagens à modelação de CIGs, posicionadas sobre um eixo temporal de acordo com a altura em que se iniciou o seu desenvolvimento. Uma seta entre duas abordagens significa que a abordagem antecedente influenciou a abordagem consequente (adaptado de (Elkin, et al., 2000)).

3.5.1. Arden Syntax

3.5.1.1. Introdução

Um dos formalismos mais conhecidos na representação de CIGs e suporte à decisão é Arden Syntax, baseado no sistema HELP. Este formalismo foi desenvolvido em 1989, e o seu nome resulta de uma conferência que reuniu instituições de saúde e académicos, com o objectivo de criar um modelo que permitisse a partilha de guias entre diferentes instituições, com diferentes sistemas de informação (Clayton, Pryor, Wigertz, & Hripcsak, 1989). Arden Syntax foi

reconhecido como um *standard* em 1992 pela American Society for Testing and Materials (ASTM) (Hripcsak, Ludemann, Pryor, Wigertz, & Clayton, 1994). Actualmente a versão de Arden Syntax em desenvolvimento é 3.0 e trata-se de um formalismo publicado pelo grupo Health Level Seven (HL7) (HL7, 2011). Esta abordagem foca-se na partilha de guias simples e independentes sob a forma de módulos. Não é um formato apropriado para guias complexas, sendo utilizado para representar regras simples.

3.5.1.2. Modelação e Representação de Guias

Em Arden Syntax, cada guia é modelada como um Medical Logic Module (MLM) que toma uma única decisão. Cada MLM é um ficheiro ASCII, que contém componentes agrupados em três categorias: *maintenance*, *library* e *knowledge* (Hripcsak, 1994).

Os compartimentos *maintenance* e *library* contêm informações descritivas acerca da guia (e.g. título, versão, palavras-chave), que são necessárias para a sua partilha. Como se pode verificar na Figura 3, um dos componentes da categoria *maintenance* é *title*, contudo existem outros, como *(file)name*, *author*, *version*, *institution*, *date of last modification* e *validation status* (Hripcsak, 1994). Em *validation status* especifica-se se a guia foi aprovada numa dada instituição local, através dos seguintes valores: *testing* (em fase de testes), *research* (aprovada para investigação clínica), *production* (aprovada para a prática clínica) e *expired* (já não se encontra em utilização) (Clercq, Blom, Korsten, & Hasman, 2004). Quando uma instituição recebe um dado MLM, o seu *validation status* está no valor *testing*. A passagem de *testing* para *production* implica uma transferência de responsabilidade da instituição que desenvolveu o MLM para a instituição local de acolhimento. O compartimento *library* é utilizado para uma descrição mais detalhada da guia, como se pode verificar pelo atributo *purpose*, que descreve o objectivo do MLM.

O conhecimento médico é armazenado na categoria *knowledge*, através dos seus componentes: *type*, *data*, *evoke*, *logic* e *action* (Hripcsak, 1994). Os componentes *data*, *evoke*, *logic* e *action* assumem um papel preponderante nesta categoria. O componente *data* é utilizado para obter os valores dos conceitos referidos no MLM, a partir do sistema de informação da instituição, como se pode observar na Figura 3 em que, por exemplo, se especifica que o valor do hematócrito (percentagem de glóbulos vermelhos no sangue) é o verificado na última contagem que consta no RSE do paciente (Clercq, Blom, Korsten, & Hasman, 2004). Por sua vez, o componente *evoke* especifica os eventos que despoletam a execução do MLM. No

exemplo em análise, *evoke* apresenta o valor *blood_count_storage*, o que significa que a cada atualização dos resultados das análises sanguíneas na base de dados, o MLM será executado (Clercq, Blom, Korsten, & Hasman, 2004). O compartimento *logic* contém critérios de decisão, que podem levar a uma determinada acção sob a forma de regras de produção, em conjunção com operadores lógicos (e.g., *or*, *and*), operadores de listas (e.g., *merge*, *sort*), operadores de cálculo (e.g., *sum*, *average*), e operadores temporais (e.g., *before*, *after*, *ago*). Os operadores lógicos podem apresentar três valores possíveis: *true*, *false* e *null* (Hripcsak, 1994). O valor *null* é encarado como desconhecido. Quando uma regra é avaliada em *true* realiza-se uma determinada acção especificada no compartimento *action*. Se as regras forem avaliadas em *false* ou *null*, a execução do MLM é abortada. As acções que constam no componente *action* podem incluir o envio de mensagens aos prestadores de cuidados de saúde (como no caso dos sistemas de alerta) ou invocação de outros MLMs, possível através da declaração *call*. Apesar de ser possível a invocação de um MLM por outro, a Arden Syntax não apresenta uma estrutura que permita controlar este processo.

Em termos de linguagem, Arden Syntax é definida em Backus-Naur Form (BNF), uma técnica de notação utilizada para descrever a sintaxe de linguagens utilizadas em computação. Os MLMs são baseados em texto, apresentando sempre o formato representado na Figura 3, em que os critérios de decisão são expressos em regras de produção textuais (Teije, Miksch, & Lucas, 2008).

```

maintenance:
  title: Alert on low hematocrit;;
library:
  purpose: Warn provider of new or worsening anemia.;;
knowledge:
  type: data-driven;;
  data:
    blood_count_storage := event {'complete blood count'};
    hematocrit := read last {'hematocrit'};
    previous_hct := read last ({'hematocrit'} where it occurred before
                             the time of hematocrit);;
  evoke: blood_count_storage;;
  logic:
    if hematocrit is not number then conclude false; endif;
    if hematocrit <= previous_hct-5 or hematocrit<30 then conclude true;
    endif;;
  action:
    write " The patient's hematocrit ("|| hematocrit ||") is low or
          falling rapidly.;;";
end:

```

Figura 3 -- Exemplo de um MLM (reproduzido de (Teije, Miksch, & Lucas, 2008)).

3.5.1.3. Aquisição e Verificação de Guias

Existem várias ferramentas que podem ser utilizadas para o desenvolvimento de MLMs. Entre elas existem editores de texto com verificação de sintaxe nos quais são introduzidos os MLMs como texto livre, assim como ferramentas que utilizam vocabulários controlados (Jenders & Dasgupta, 1996).

A versão 3.0 de Arden Syntax que se encontra actualmente em desenvolvimento pelo grupo HL7 é baseada em Extensible Markup Language (XML). Esta nova sintaxe denomina-se ArdenML (Arden Syntax Markup Language) (Kim, Haug, Rocha, & Choi, 2008) e estão a desenvolver-se esforços no sentido de criar ferramentas de aquisição de MLMs em ArdenML e ferramentas de conversão de MLMs no seu formato original para ArdenML, ao mesmo tempo que se realiza a verificação da sua integridade e validade.

3.5.1.4. Execução de Guias

De forma a executar os MLMs, é necessário efectuar a sua conversão para um formato que seja reconhecido por um motor de execução de guias. Existe um grande número de aplicações que foram desenvolvidas utilizando pseudocódigo (Ma, 1995), C++ (Kuhn & Reider, 1994), MUMPS (McCauley, Young, Clark, & Peters, 1996) e JAVA (Karadimas, Chailloleau, Hemery, Simonnet, & Lepage, 2002). Contudo, compilar Arden Syntax para um formato intermédio e posteriormente para um formato executável é uma tarefa difícil, pois é necessária a utilização de compiladores específicos para instituições que utilizem sistemas de informação em linguagens de programação diferentes (Kim, Haug, Rocha, & Choi, 2008).

3.5.2. Guideline Interchange Format (GLIF)

3.5.2.1. Introdução

GLIF (Ohno-Machado, et al., 1998) representa um esforço da organização Intermed Collaboratory (uma colaboração das Universidades de Harvard, Stanford e Columbia) no desenvolvimento de uma representação partilhável de CIGs, cuja primeira publicação data de 1998. Recebeu influências de outras abordagens existentes, como Arden Syntax (Clayton, Pryor, Wigertz, & Hripcsak, 1989), GEODE CM (Stoufflet, Ohno-Machado, Deibel, Lee, & Greenes, 1996), MBTA (Barnes & Barnett, 1995) e EON (Musen, Tu, Das, & Shahar, 1996).

Esta abordagem foi desenvolvida de forma a reflectir um fluxograma de passos estruturados e agendados, que representassem decisões e acções clínicas. As primeiras iterações de GLIF careciam de uma especificação formal dos passos de uma guia e de um mapeamento dos dados da guia para o registo de saúde electrónico. O principal objectivo é a partilha de guias e por isso estas são modeladas de forma a serem perceptíveis tanto por especialistas do domínio clínico como por *parsers* automáticos utilizados em diferentes sistemas de apoio à decisão.

As primeiras versões de GLIF permitiam a representação de guias mais complexas do que Arden Syntax. No entanto apresentavam também algumas limitações, tais como:

- Na primeira versão, atributos importantes do modelo eram representados como *strings* de texto;
- O número de construtores em GLIF2 era considerado muito limitado e não abrangia aspectos como: decisões alternativas, iterações e estados do paciente;
- Em GLIF e GLIF2 não existiam construtores que permitiam o mapeamento dos elementos do RSE do paciente para os elementos da guia.

Estas falhas foram corrigidas numa versão posterior, denominada GLIF3 (Peleg, et al., 2000), que é a versão de GLIF que se irá abordar.

3.5.2.2. Modelação e Representação de Guias

O modelo GLIF3 é orientado aos objectos e consiste num conjunto de classes que descrevem características típicas de uma guia (por exemplo decisões ou acções), atributos dessas classes e tipos para os valores dos atributos. Em GLIF3 todas as classes e atributos são descritos através de diagramas de Unified Modeling Language (UML) (Rumbaugh, Jacobson, & Booch, 1999).

Uma guia, em GLIF3, é um objecto, que pode conter *sub-guias*. Este objecto possui um conjunto de atributos administrativos, como nome e autor, mas também atributos que descrevem os *objectivos* da guia.

Este modelo consiste num conjunto de passos, que estão ligados num fluxograma. GLIF3 define cinco tipos de passos: *decision steps*, *patient state steps*, *branch steps*, *synchronization steps* e *action steps* (Figura 4). Cada guia em GLIF consiste num conjunto de pontos, que representam cada um dos cinco tipos de passos, ligados entre si num fluxograma. Desta forma GLIF alcança um dos seus principais objectivos, que se trata de conseguir uma maior facilidade na interpretação de guias pelos indivíduos.

Decision steps modelam pontos de decisão numa guia, direccionando o fluxo de trabalho de um passo para passos alternativos. Dentro de uma *decision* existem duas subclasses: *case step* e *choice step*. Em *case step* existe um conjunto de expressões lógicas que direccionam o fluxo de trabalho para um dos passos alternativos. As expressões lógicas correspondem a um excerto de Arden Syntax, que assume a designação *Guideline Expression Language* (GEL) (Peleg, et al., 2001). Um *choice step* aplica-se em situações em que a guia sugere múltiplas alternativas, mas deixa a escolha a cargo de um agente externo, no caso o utilizador.

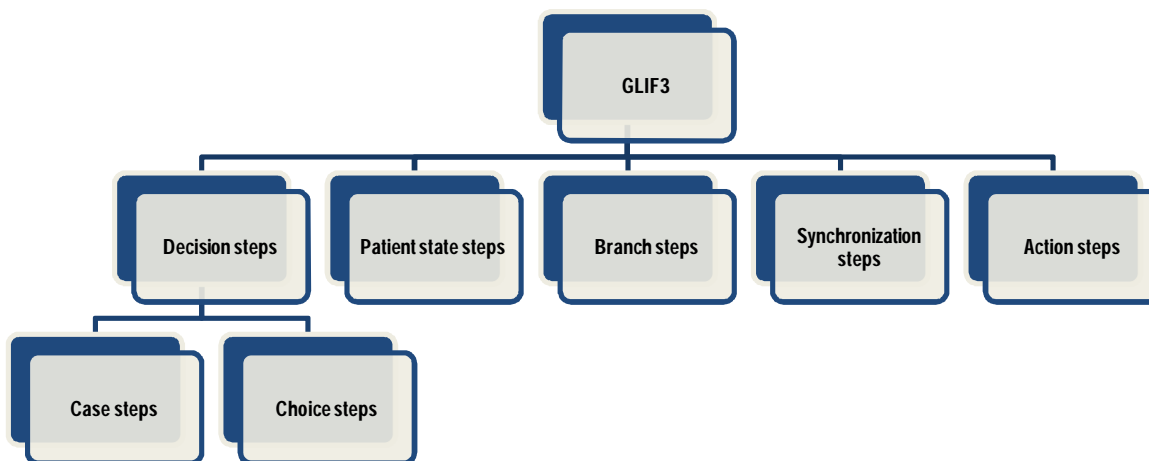


Figura 4 -- Representação esquemática dos tipos de passos do modelo GLIF3.

Patient state steps funcionam como etiquetas que contêm atributos utilizados para descrever o estado do paciente. Este tipo de passo pode ser utilizado como ponto de entrada de dados no sistema. Quando o estado do paciente é actualizado, a guia que possui o *patient state* correspondente é executada.

Os *branch steps* modelam um conjunto de passos concorrentes, direccionando o fluxo de trabalho para passos paralelos. São utilizados juntamente com *synchronization steps*. Os passos paralelos, que se seguem a *um branch step*, convergem para um passo comum através de um *synchronization step*. Quando uma das ramificações chega a um passo de sincronização, existe um atributo de continuação, que faz a verificação se os passos dos ramos paralelos já foram cumpridos, para assim proceder-se ao próximo passo. O atributo de continuação é expresso como uma expressão lógica.

Os *action steps* modelam tarefas que devem ser realizadas. São definidos três tipos de tarefas: acções médicas, acções orientadas a actividades (como por exemplo envio de

mensagens ou obtenção de dados do paciente) e acções de controlo (invocação de estruturas como *sub-guias* ou *macros* que permitem recursividade).

Para que as guias sejam partilháveis por diferentes instituições, GLIF3 recorre à utilização de *standards* para a estruturação da informação. GLIF3 define três camadas de informação:

- A camada nuclear: onde são definidas as variáveis e as relações existentes entre elas;
- A camada do Reference Information Model (RIM) (Schadow, Russler, Mead, & McDonald, 2000): onde é utilizado o *standard* HL7 para a estruturação da informação e definição da sintaxe das comunicações;
- A camada de conhecimento médico: que especifica os métodos necessários para realizar a interface com várias fontes de conhecimento médico e sistemas de informação, como por exemplo terminologias como Unified Medical Language System (UMLS) (Lindberg, 1990) ou sistemas de informação clínicos das instituições locais.

A partir da versão GLIF3, GLIF é expresso através de XML.

A linguagem GELLO (Sordo, Ogunyemi, Boxwala, & Greenes, 2003) também pode ser utilizada para especificar critérios de decisão no modelo GLIF, com a vantagem, em relação à linguagem GEL, de permitir a referência às variáveis da camada de informação central.

De modo a que as guias sejam perceptíveis por pessoas, interpretáveis por computadores e adaptáveis a diferentes instituições locais, GLIF3 define três níveis de abstracção: o nível conceptual, o nível computacional e o nível de implementação. O nível mais abstracto é o nível conceptual, no qual é representado o fluxograma da guia, que pode ser visualizado pelas pessoas, mas não pode ser interpretado pelos sistemas de apoio à decisão. O nível conceptual pode ser visualizado na Figura 5. As especificações e verificações que tornam a guia interpretável por uma máquina encontram-se no nível computacional. Por fim, o nível de implementação contém as especificações necessárias para adaptações do modelo GLIF3 aos sistemas de informação de instituições locais.

3.5.2.3. Aquisição e Verificação de Guias

Protégé (Grosso, Eriksson, Ferguson, Gennari, Tu, & Musen, 1999) e GEODE (Greenes, Boxwala, Sloan, Ohno-Machado, & Deibel, 1999) são utilizados como ferramentas de aquisição de conhecimento em GLIF3. Ambas as ferramentas implementam uma aquisição baseada em fluxogramas.

3.5.2.4. Execução de Guias

O motor de execução da versão 2 de GLIF denominava-se Partners Computerized Algorithm Processor and Editor (PCAPE) *tool* (Zielstorff, et al., 1998).

Contudo com o desenvolvimento de GLIF3 encontram-se também em desenvolvimento novos motores de execução, como é o caso do Guideline Execution Engine (GLEE) (Wang, et al., 2004). Este novo motor de execução permite uma melhor integração com os sistemas de informação clínicos.

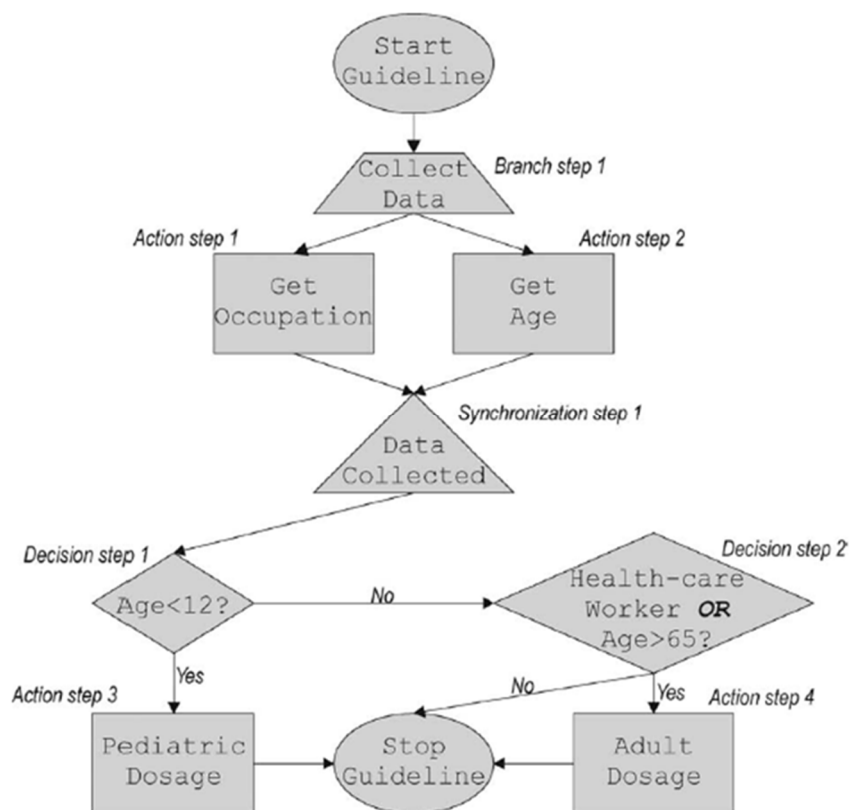


Figura 5 – Excerto de uma representação gráfica de uma guia para vacinação em GLIF3 (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).

3.5.3. PROforma

3.5.3.1. Introdução

No Reino Unido, o Advanced Computation Laboratory of Cancer Research iniciou em 1998 o desenvolvimento do formalismo PROforma (Fox, Johns, & Rahmanzadeh, 1998; Vollebregt, Teije, Harmelen, der, & Mosseveld, 1999). O objectivo deste formalismo é o desenvolvimento de

sistemas especialistas mais confiáveis que possam auxiliar os cuidados de saúde a um paciente de forma activa, através de suporte à decisão e gestão do fluxo de trabalho. PROforma resulta da concatenação das palavras “*proxy*” (que significa “permissão para actuar como outrem”) e “*formalize*” (que significa “dar forma definitiva a”).

3.5.3.2. Modelação e Representação de Guias

Tal como GLIF, PROforma também representa guias como um fluxograma, nos quais os nodos são instâncias de classes pré-definidas. A este modelo de representação dá-se o nome de ontologia de tarefas PROforma. O modelo define quatro classes de tarefas, representadas na Figura 6. *Plans*, *decisions*, *actions* e *enquiries* possuem os seus próprios atributos.

Todas as tarefas que constam no fluxograma derivam de uma tarefa inicial comum, *root task*, que contém atributos que são comuns a todas as quatro tarefas deriváveis. Estes atributos podem ser de cariz administrativo: *name* (identificador da tarefa), *caption* (nome da tarefa para efeitos de impressão) e *description* (descrição breve da guia). Outros atributos podem descrever aspectos relacionados com a abrangência da guia, como:

- *Goals*: definem o objectivo da tarefa (e.g., *achieve(normal_respiration)*);
- *Preconditions* e *postconditions*: condições que se devem verificar antes e depois, respectivamente, da execução de uma tarefa (e.g., *risk_level=severe*);
- *Trigger conditions*: podem ser eventos temporais ou estados do paciente, que, ocorrendo, despoletam a realização de uma tarefa (e.g., *peak_flow <30*);
- *Cycles*: definem condições e restrições segundo as quais uma tarefa deve ser repetida (e.g., *cycle(Integer,Interval)*, *cycle(until(State),Interval)*).

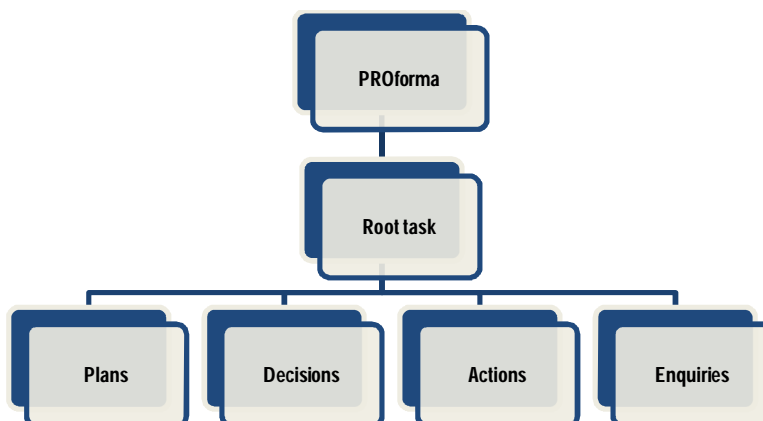


Figura 6 -- Representação esquemática das principais classes de tarefas do modelo PROforma.

A classe *plans* define *sub-guias* e determina uma sequência ordenada de tarefas, restrições lógicas, restrições temporais à sua execução e as circunstâncias em que um plano deve terminar ou abortar. Isto está explícito no tipo de atributos que a classe *plans* acrescenta aos já referidos: *components*, *scheduling constraints*, *temporal constraints*, *abort conditions* e *termination conditions*. O atributo *components* alberga um conjunto de referências a tarefas que constituem um plano (e.g., *history*, *diagnosis*, *therapy*, *follow-up*), cuja ordem é definida em *scheduling* e *temporal constraints*. É possível definir que uma tarefa ocorre especificamente após outra, ou que uma tarefa ocorre após um dado período de tempo. Esta restrição temporal à realização de tarefas é o que demarca PROforma de outras abordagens como GLIF. Outra forma de redireccionar o fluxo de trabalho é através de *termination conditions* (estado do paciente que provoca o fim do plano) e *abort conditions* (estado do paciente que provoca a interrupção do plano).

A classe *decisions* é representada como o conjunto de soluções candidatas. Cada solução candidata está associada a um esquema com um conjunto de expressões lógicas que suportam ou refutam cada resultado. Estas expressões lógicas consistem em regras, variáveis qualitativas, pesos e factores de certeza (Fox & Das, 2000) a favor ou contra soluções candidatas. Cada um dos argumentos, a suportar ou refutar uma opção, tem associados sinais positivos (marcados por um sinal +) ou sinais negativos (marcados por sinais -). O valor do peso positivo ou negativo depende da quantidade de sinais positivos/negativos que uma regra tem associada. Através de uma função matemática, os sinais + e - são transformados em valores numéricos e calcula-se o peso resultante do conjunto de regras, apresentando-se no final as opções por ordem decrescente de preferência.

Uma *action* em PROforma é uma tarefa cuja execução é requisitada pelo motor de execução de PROforma a um agente externo (como por exemplo o utilizador, *software* externo ou um componente de hardware). Geralmente este tipo de tarefas consiste na emissão de mensagens ou a chamada de um programa externo através de uma Application Programming Interface (API). Estas acções são sempre atómicas e não podem ser decompostas. Em *actions* encontram-se os atributos *method* (a forma como a execução deve ser realizada) e *confirmation* (especificação sobre se a acção necessita de confirmação do utilizador).

A classe *enquiries* define tarefas de obtenção de informação médica ou administrativa. Esta informação pode ser obtida através de perguntas ao utilizador ou a partir do registo de

saúde electrónico do paciente. Nesta classe, destaca-se o atributo *data definition*, que define a forma como os dados são introduzidos na base de conhecimento.

A representação em fluxograma de PROforma apresenta a particularidade de as tarefas serem representadas de forma diferente, em função da classe a que pertencem. *Plans* são representados por rectângulos com os vértices redondos, *decisions* através de círculos, *actions* através de rectângulos e *enquiries* por losangos.

As guias em PROforma são armazenadas utilizando Red Representation Language (R^2L), uma linguagem de representação orientada ao tempo. Para realizar a execução da guia é necessário converter a guia para outra linguagem, denominada Logic of R^2L (L_{R^2L}). L_{R^2L} é uma linguagem de representação do conhecimento baseada em predicados.

3.5.3.3. Aquisição e Verificação Guias

PROforma possui um conjunto de ferramentas para desenvolvimento de guias. A Figura 7 representa a *framework* do ambiente de criação de tarefas de PROforma (Clercq, Blom, Korsten, & Hasman, 2004), que permite aos autores de guias defini-las em termos das instâncias das classes de tarefas e respectivos atributos.

O painel esquerdo da Figura 7 mostra uma visão geral em forma de árvore de todos os planos, ao passo que o painel central mostra uma representação gráfica do plano seleccionado. Quando uma tarefa é seleccionada na árvore, o painel direito apresenta os seus atributos. A aquisição de guias em PROforma é realizada em duas fases. A primeira fase consiste em definir através de uma ferramenta gráfica instâncias das quatro classes. A segunda fase consiste em inserir valores específicos para os atributos de cada instância.

Após o armazenamento da guia em R^2L , esta é processada por uma ferramenta de verificação de forma a detectar erros declarativos e de sintaxe. Após a verificação dá-se a tradução para L_{R^2L} .

As aplicações Arezzo e HeCaSe2 também permitem a aquisição e verificação guias em PROforma (Isern & Moreno, 2008).

3.5.3.4. Execução de Guias

A *framework* PROforma (Clercq, Blom, Korsten, & Hasman, 2004) possui um motor de execução *standard* que executa as guias introduzidas através da linguagem L_{R^2L} . O motor de execução pode

ser ligado a várias interfaces através de APIs específicas. A execução também é possível utilizando Arezzo e HeCaSe2 (Isern & Moreno, 2008).

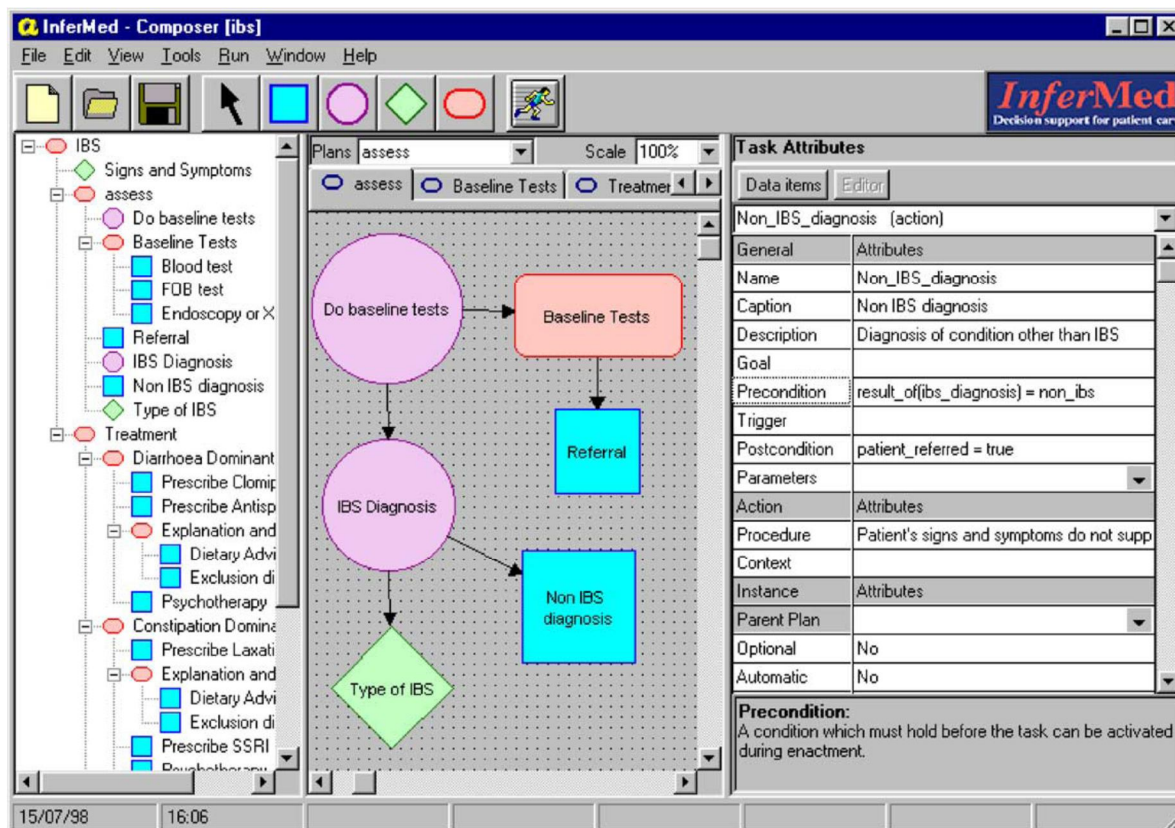


Figura 7 -- Visualização de uma guia no ambiente de criação de tarefas PROforma (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).

3.5.4. Asbru

3.5.4.1. Introdução

Tal como PROforma, o formalismo Asbru (Miksch, Shahar, & Johnson, 1997) permite expressar objectivos, que, no contexto deste modelo, adquirem particular relevância e são designados por intenções. Foi desenvolvido pela Universidade de Stanford e pela Universidade de Tecnologia de Viena e é particularmente avançado na modelação dos aspectos temporais de guias. Asbru faz parte do projecto Asgaard (Shahar, Miksch, & Johnson, 1998) e é um formalismo para representação de guias como planos, especializando-se na representação de padrões e anotações temporais e desenvolvendo um método de visualização de guias num eixo temporal.

3.5.4.2. Modelação e Representação de Guias

Asbru utiliza um modelo baseado em intenções de forma a representar GCs como planos. Este modelo utiliza uma noção de plano semelhante à de PROforma, na medida em que os planos são vistos como colecções de itens. Cada tarefa é realizada com base em métodos de resolução de problemas, que apresentam um conjunto de estratégias gerais, independentemente do domínio de aplicação. O conhecimento necessário para resolver uma dada tarefa é definido em *knowledge roles*, que providenciam uma descrição da função do conhecimento no método de resolução do problema. Asbru define os seguintes *knowledge roles*: *preferences*, *plan intentions*, *conditions*, *effects* e *plan body*.

O conteúdo de um *plan body* é sempre constituído por outros planos, até que um plano já não possa ser decomposto. Os planos que não podem ser decompostos denominam-se *actions*. Em Asbru, as guias consistem em *plans* e *actions*, e as suas funcionalidades são definidas pelos *knowledge roles*.

As *preferences* restringem a aplicação de um dado plano, de forma a atingir um dado objectivo. Exemplos de preferências são:

- *Select-method*: uma forma de determinar o método a utilizar em função da sua adequação (e.g., *exact fit*, *roughly fit*);
- *Resources*: uma especificação de recursos obrigatórios ou proibidos (e.g., em casos de infecção muito específicos, proibir a cirurgia e obrigar à utilização de antibióticos);
- *Strategy*: que define a estratégia a utilizar (e.g., *agressive* ou *normal*).

Um dos aspectos chave de Asbru é a representação dos objectivos de um plano em *intentions*. A definição de *intentions* auxilia a selecção do plano mais apropriado e é crucial no apoio à decisão. Um exemplo destas situações é um caso em que um profissional de saúde pretende tratar a hipertensão, para a qual um tratamento possível é a administração de beta-bloqueadores. Contudo, o médico deseja evitar a sua utilização e segue outro plano. Com base nas *intentions*, caso o objectivo de baixar a hipertensão não seja atingido, o sistema é capaz de fazer uma apreciação crítica do procedimento do profissional de saúde, aconselhando-o. Por outro lado, se o resultado do tratamento estiver de acordo com as *intentions*, não será gerada a apreciação crítica e o sistema aceitará o plano seguido como correcto. As *intentions* são

definidas como padrões temporais de acções dos profissionais de saúde e estados dos pacientes que devem ser mantidos, alcançados ou evitados. Existem quatro categorias de *intentions*:

- *Intermediate state*: os estados dos pacientes que devem ser mantidos, alcançados ou evitados, ou seja, manter um dado estado (e.g., níveis de substâncias no sangue) durante o período de tempo em que decorre o plano;
- *Intermediate action*: as acções dos prestadores de cuidados que devem ocorrer durante a execução do plano (e.g., verificar níveis de glicose uma vez por dia);
- *Overall state pattern*: o padrão global de estado do paciente que se deve verificar após completar o plano (e.g., o nível de glicose do paciente é apropriado);
- *Overall action pattern*: o padrão global das acções dos profissionais de saúde que se deve verificar após a conclusão do plano (e.g., o paciente visitou regularmente o nutricionista nos últimos três meses).

As *conditions* são também padrões temporais e são utilizadas para alterar o estado de um plano. Tal como PROforma, Asbru define estados para os planos, que passam por activo, suspenso, terminado ou abortado, e as categorias que definem estes estados. Estas categorias são:

- *Filter-preconditions* e *setup-preconditions*: que se devem verificar para que um plano seja considerado aplicável;
- *Suspend-conditions*: que determinam quando um plano deve ser suspenso;
- *Abort-conditions*: que determinam quando um plano deve ser abortado.

Effects descrevem a relação entre argumentos de planos e efeitos mensuráveis através de funções matemáticas, por exemplo, o estabelecimento da relação inversamente proporcional entre insulina e o nível de glicose no sangue. Também se pode incluir neste item probabilidades de ocorrência dos efeitos.

O *plan body* é um conjunto de planos e acções que devem ser realizados sempre que as condições se apresentem. Em Asbru, existem três tipos de planos: *sequential*, *concurrent* e *cyclical*.

Na representação de guias em Asbru são de extrema importância as anotações temporais (Shahar, et al., 2003), que especificam quatro pontos no tempo relativamente a um ponto de referência, o qual pode ser um ponto específico ou abstracto no tempo ou uma transição de estados de um plano. Estes quatro pontos são: *earliest starting shift* (ESS), *latest starting shift* (LSS), *earliest finishing shift* (EFS) e *latest finishing shift* (EFS). Podem ser

especificadas duas durações: *minimum duration* (MinDu) e *maximum duration* (MaxDu). Estas referências representam as restrições temporais para a execução dos planos ou verificação de condições.

A sintaxe formal da linguagem Asbru está definida em BNF e as guias são codificadas numa linguagem tipo LISP. Actualmente Asbru encontra-se disponível também em XML (Votruba, Miksch, & Kosara, 2004).

3.5.4.3. Aquisição e Verificação de Guias

Ao contrário de outras abordagens como GLIF e PROforma, a visualização de guias em Asbru não acontece na forma de um fluxograma, pois a visualização do tempo e intenções através de um fluxograma é difícil e complexa. Para o efeito foi desenvolvida uma ferramenta denominada AsbruView (Miksch, Kosara, Shahar, & Johnson, 1998), que utiliza gráficos para a visualização de guias. Em AsbruView, os planos são visualizados como pistas e os vários tipos de condições são vistos como sinais de tráfego, segundo uma vista topológica. A Figura 8 apresenta a vista topológica em AsbruView de uma guia para o tratamento de distúrbio respiratório em criança. O comprimento da pista representa o tempo e a espessura representa planos que se encontram no mesmo nível de decomposição. Um exemplo da vista de condições como sinais de tráfego é o semáforo que se pode verificar na Figura 8.

Em AsbruView os planos também podem ser visualizados na vista temporal, que se foca na sua dimensão temporal e respectivas condições. Esta vista faz uso das referências temporais acima mencionadas.

Ainda se encontram em desenvolvimento ferramentas que permitam a verificação e teste de guias em Asbru, de forma a detectar erros de declaração e sintaxe.

Para além de AsbruView, existe uma ferramenta de aquisição de guias denominada DeGeL (Shahar, et al., 2003).

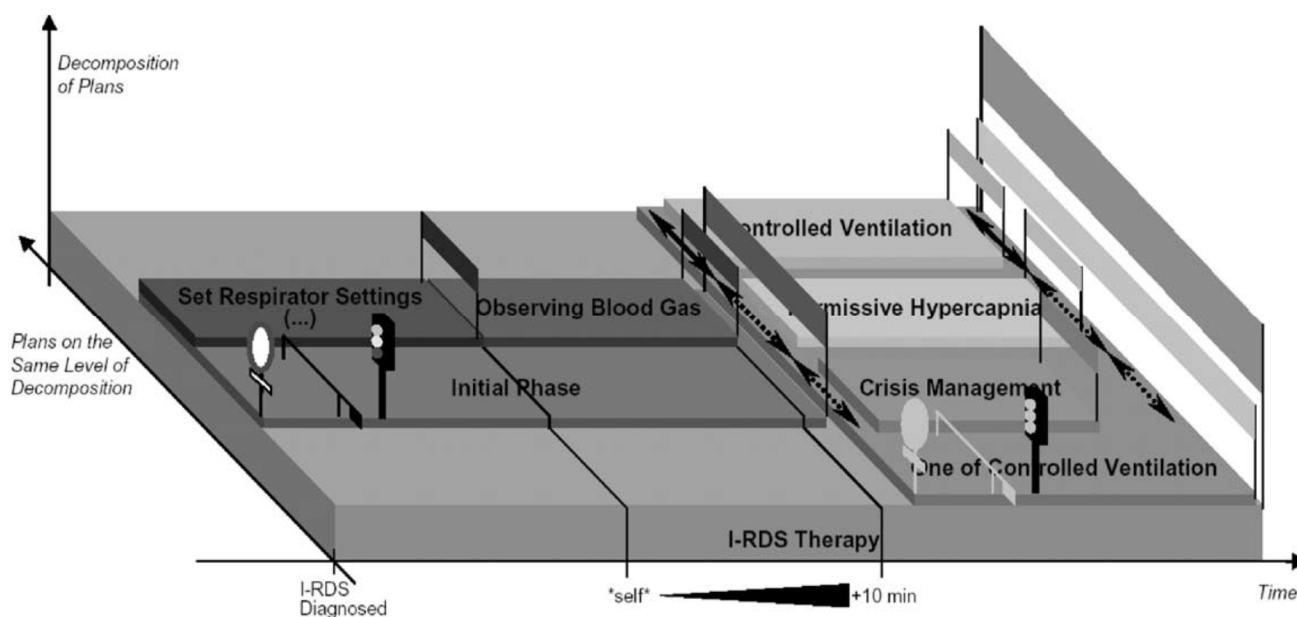


Figura 8 -- Visualização de uma guia para o tratamento de distúrbio respiratório em crianças em AsbruView (reproduzido de (Clercq, Blom, Korsten, & Hasman, 2004)).

3.5.4.4. Execução de Guias

Ferramentas de *software* que permitem a execução de guias em Asbru encontram-se em desenvolvimento. Contudo, também é possível a sua execução em DeGeL (Shahar, et al., 2003). DeGeL é uma aplicação *web* que possui um repositório de guias em Asbru, organizadas segundo uma rede hierárquica de planos e acções.

3.5.5. EON

3.5.5.1. Introdução

O formalismo EON foi um dos modelos de CIGs que realçou a importância da compreensão do utilizador do fluxo de tarefas representado. EON (Musen, Tu, Das, & Shahar, 1996) foi desenvolvido na Universidade de Stanford e é o precursor de formalismos como GLIF (Peleg, Boxwala, Bernstam, Tu, Greenes, & Shortliffe, 2001) e SAGE (Tu, et al., 2007), continuando a ser desenvolvido como um sistema de investigação. Tal como GLIF representa as guias na forma de fluxogramas.

A arquitectura EON consiste em vários componentes que facilitam a aquisição e execução de GCs. O modelo para GCs de EON denomina-se Dharma (Tu & Musen, A flexible

approach to guideline modeling., 1999) e é orientado aos objectos. Dharma consiste em classes que descrevem entidades de uma guia de passos temporalmente estruturados. O modelo Dharma não é estático, o que significa que o modelo pode ser expandido com classes adicionais que captem novos aspectos da guia.

3.5.5.2. Modelação e Representação de Guias

No modelo Dharma, as guias gerem o comportamento do paciente, consistindo em decisões e acções que podem levar a mudanças no estado do paciente ao longo do tempo. Neste modelo as decisões são tomadas em encontros dos profissionais de saúde com os pacientes. Acções, como escrever receitas ou a solicitação de um exame de laboratório, são executadas durante os encontros. Para a representação de restrições temporais, EON utiliza um excerto de Asbru.

As classes primitivas em EON encontram-se representadas na Figura 9 e são: *scenarios*, *decisions*, *actions* e *goals*. Estas classes formam a ontologia da guia.

Um *scenario* é uma caracterização parcial do estado de um paciente. Nesta classe existem condições de selecção, que especificam os critérios necessários para um paciente se encontrar no referido *scenario*. Os *scenarios* permitem que um médico sincronize a gestão de um paciente com as partes correspondentes de uma guia e são normalmente utilizados como pontos de entrada de informação numa guia. Um *scenario* é sempre seguido de *actions* ou *decisions*. A classe *scenarios* possui atributos que especificam os critérios de selecção e que identificam o passo seguinte a um *scenario*. Os *scenarios* também podem ser utilizados para modelar excepções que representam situações de saúde que raramente ocorrem. A ontologia Dharma define dois tipos de excepções:

- Excepções que são reparáveis: excepções que levam o paciente de volta a um *scenario* incluído na guia;
- Excepções que não são reparáveis: quando o estado do paciente implica o encaminhamento para outra guia.

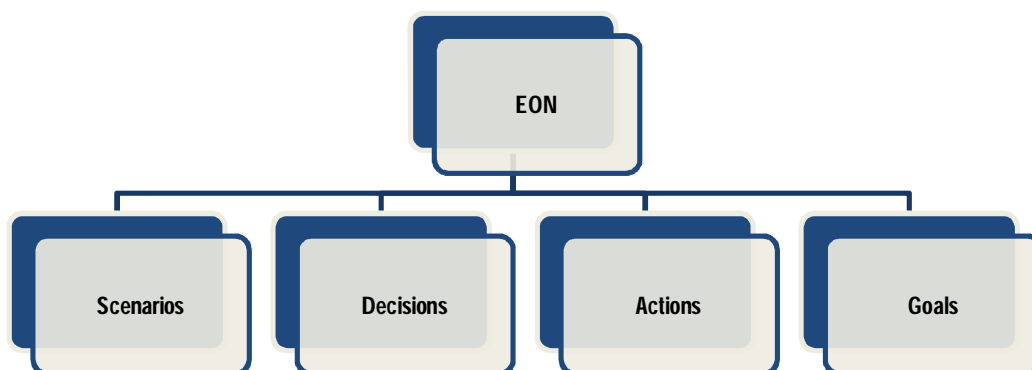


Figura 9 -- Representação esquemática das principais classes do modelo EON.

Na ontologia Dharma, existem dois tipos básicos de *decisions* na forma de subclasses: *decisions* que modelam escolhas na forma de regras “*if...then...else...*” e *decisions* que requerem a realização de uma escolha heurística de entre um número de alternativas. A escolha no segundo tipo de *decisions* é auxiliada por preferências determinadas por condições de inclusão ou exclusão que apoiam ou contrariam cada alternativa (semelhante a PROforma). Para que uma alternativa ganhe preferência sobre outra, é necessário que a condição de inclusão seja avaliada como verdadeira e a condição de exclusão seja avaliada como falsa. Caso nenhuma das condições sejam avaliadas em verdadeira, não sendo possível seleccionar uma ou outra alternativa, são atribuídos valores de preferência por defeito, que existem já no sistema.

As *actions* são definidas como actos instantâneos que conduzem a alterações no estado do mundo, como por exemplo recolher dados do paciente, mostrar uma mensagem ao utilizador ou começar um novo tratamento. Em EON também existe o conceito de *activity*. Ao passo que *actions* são instantâneas, as *activities* modelam processos que decorrem ao longo do tempo. As *activities* possuem estados que se podem alterar ao longo do tempo, como resultado de *actions*. EON também possui *actions* que referenciam um conjunto de outras acções ou uma *sub-guia*. Exemplos deste tipo de *actions* são aquelas que modelam passos de ramificação e sincronização.

Cada passo em EON pode ser associado a um objectivo, um *goal*. A noção de *goal* pode ser comparada à noção de *intention* em Asbru. Neste caso, os *goals* são representados como critérios booleanos (e.g., *blood pressure <130/85 = true*).

EON define também um modelo de dados do paciente onde existem classes e atributos apropriados para a representação desta informação. Por exemplo, é definida uma classe *patient*, cujas instâncias possuem informações demográficas sobre pacientes específicos. Também é definida uma classe *qualitative_entry*, que descreve observações qualitativas sobre o paciente, assim como uma *numeric_entry* para resultados de testes médicos quantitativos. O modelo de dados do paciente não pretende ser uma cópia do seu RSE, mas sim, modelar os aspectos do paciente que são relevantes para as guias e protocolos médicos.

À semelhança do modelo de dados do paciente, EON possui um modelo de especialidade médica, que consiste numa ontologia de domínio médico que estrutura os conceitos de um domínio (por exemplo medicamentos e tratamentos).

Em termos de linguagem, EON utiliza o Resource Description Framework (RDF) para a descrição do modelo e das guias. Para a representação de critérios mais complexos em *goals* é utilizada a Protégé Axiom Language (PAL) (Grosso, Eriksson, Ferguson, Gennari, Tu, & Musen, 1999), que se trata de um excerto da sintaxe do Knowledge Interchange Format (KIF) (Patil, et al., 1992).

3.5.5.3. Aquisição e Verificação Guias

A ferramenta Protégé (Grosso, Eriksson, Ferguson, Gennari, Tu, & Musen, 1999) também é utilizada como ferramenta de aquisição de conhecimento no modelo EON, na qual os utilizadores são capazes de introduzir e visualizar guias. .

Como já foi mencionado, a PAL é utilizada para descrever critérios complexos, utilizados nas decisões e o Protégé facilita aos criadores de guias a escrita destes critérios lógicos.

3.5.5.4. Execução de Guias

De forma a facilitar o desenvolvimento de motores de execução de guias, EON define uma arquitectura de decisão (Tu, Musen, & others, 2001) que contém componentes para a execução de guias e interfaces com outros sistemas de informação.

No centro da arquitectura está o Padda Guideline Execution Server, o qual aplica uma guia aos dados de um paciente, retirados da base de dados de um sistema de informação clínico. No Padda Server existe um gestor da base de conhecimento que utiliza uma API fornecida por Protégé. É o Padda Server que determina se uma dada guia é aplicável a um paciente e implementa o modo de interacção com o mundo exterior.

A interacção do Padda Server com outros sistemas de informação é possível através de uma especificação de interface escrita em Common Object Request Broker Architecture Interface Definition Language (CORBA IDL), que consiste em métodos de transferência de dados estruturados entre o servidor e o cliente.

De forma a analisar situações de pacientes, os termos existentes no seu RSE devem ser mapeados para os conceitos no modelo de dados do paciente, assim como a terminologia do domínio da especialidade médica deve ser mapeada para a terminologia da especialidade do sistema de informação. Isto é conseguido através de uma ferramenta denominada Tzolkin.

3.5.6. Standards-Based Sharable Active Guideline Environment (SAGE)

3.5.6.1. Introdução

O projecto SAGE (Tu, et al., 2007) é uma colaboração de seis grupos de investigação (IDX Systems, Centro Médico da Universidade do Nebraska, Intermountain Health Care, Apelon Inc., Stanford Medical Informatics e Mayo Clinic) e procura estabelecer uma infra-estrutura que permita a partilha de guias em sistemas de informação clínicos heterogéneos. Um aspecto chave de SAGE nos seus objectivos é o seu envolvimento com organizações que desenvolvem *standards* (e.g., HL7), de forma a encurtar a distância entre a lógica das guias e a sua implementação clínica. Representa uma evolução face a formalismos anteriores como EON (Musen, Tu, Das, & Shahar, 1996) e GLIF (Ohno-Machado, et al., 1998).

Existem três componentes nos modelos de informação SAGE: uma visão dos dados do paciente, um modelo de organização dos cuidados de saúde e um modelo do fluxo de trabalho. O objectivo destas especificações é a separação entre os passos genéricos nos cuidados de saúde e os passos específicos de um ambiente em particular.

3.5.6.2. Modelação e Representação de Guias

O modelo de representação de SAGE (Tu, et al., 2007) consiste em *recommendation sets* que simbolizam diferentes GCs. Um *recommendation set* é um conjunto de recomendações que são aplicáveis num dado contexto clínico, organizadas segundo um formalismo. Os *recommendation sets* são compostos por *context nodes*, *decision nodes*, *action nodes* e *routing nodes*, como se pode verificar na Figura 10. É possível referenciar outros *recommendations sets*, num dado

recommendation set, permitindo a nidificação de GCs. A representação dos diferentes nodos, num grafo, denomina-se *activity graph*.

O contexto é especificado num *context node*, que por sua vez é definido pela combinação das especificações do ambiente clínico (e.g., consultas externas de clínica geral), das especificações do profissional de saúde para o qual se dirigem as recomendações e de estados do paciente que sejam relevantes (e.g., paciente hipertenso). Os *action nodes* especificam actividades do sistema de informação executadas como suporte à guia, como por exemplo, a obtenção de parâmetros do estado do paciente. Os *decision nodes* contêm condições de inclusão e condições de exclusão, à semelhança de EON, sobre as recomendações, expressas na linguagem GELLO (Sordo, Ogunyemi, Boxwala, & Greenes, 2003). Quanto aos *routing nodes*, controlam a execução da guia.

SAGE faz uso de *standards*, como o RIM do grupo HL7 (Schadow, Russler, Mead, & McDonald, 2000), para estruturar a informação e tipos de dados. É implementado um modelo de informação denominado Virtual Medical Record (VMR), que foi expandido para a representação de dados de pacientes e acções da guia. SAGE utiliza *standards* de terminologias como SNOMED-CT e LOINC, que podem ser encontrados no Unified Medical Language System (UMLS) (Bodenreider, 2004).

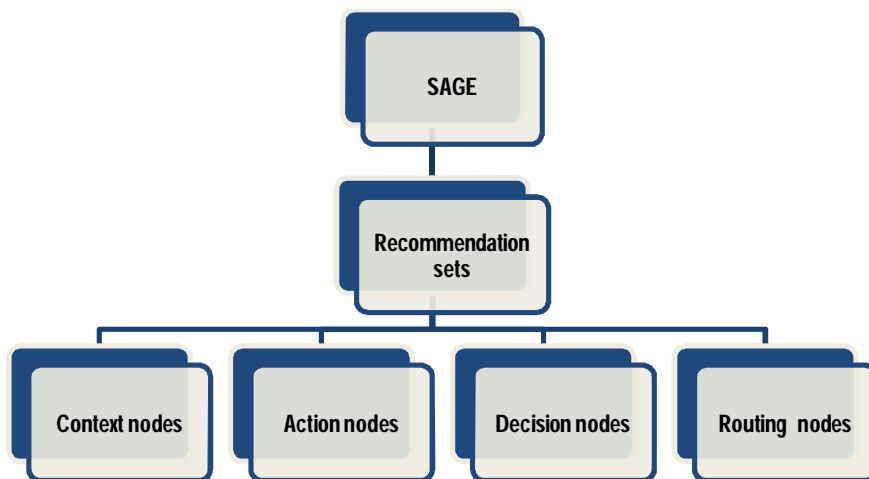


Figura 10 - Representação esquemática das principais classes do modelo SAGE.

3.5.6.3. Aquisição e Verificação Guias

Num ambiente SAGE (Ram, et al., 2004), assume-se que primeiramente a guia é importada para a instituição de cuidados de saúde. A instalação da guia envolve dois passos principais: a instituição edita a guia, para que esta corresponda às suas políticas organizacionais e clínicas (processo designado por *localization*) e o mapeamento dos conceitos dos *standards* da guia para os dados do sistema de informação clínico (processo designado por *binding*). Após a instalação, a guia está disponível para ser colocada no motor de execução.

Quando é detectada uma guia para execução, o motor de execução realiza verificações de forma a determinar se a guia está correctamente codificada.

3.5.6.4. Execução de Guias

O ambiente de execução de guias SAGE (Ram, et al., 2004) consiste num motor de execução, um auditor de eventos, um servidor de terminologia, um conjunto de interfaces denominadas *VMR/action services*, todos presentes na Figura 11. O motor de execução pode executar múltiplas guias.

O motor de execução interpreta o conteúdo dos *context*, *action* e *decision nodes* e gere o fluxo de trabalho. O *event listener* é o mecanismo pelo qual o motor de execução é notificado de mudanças de estado no sistema de informação clínico, através de um módulo JAVA que encaminha eventos do sistema de informação clínico para o *event listener*. Os *VMR/action services* são interfaces para os dados dos pacientes e para funcionalidades do sistema de informação clínico. Os *VMR services* obtêm dados do sistema de informação clínico, ao passo que os *action services* são utilizados para desencadear acções no sistema de informação clínica. Os *VMR/action services* utilizam *standards* de mensagens HL7 (Schadow, Russler, Mead, & McDonald, 2000) na comunicação com os sistemas de informação clínica.

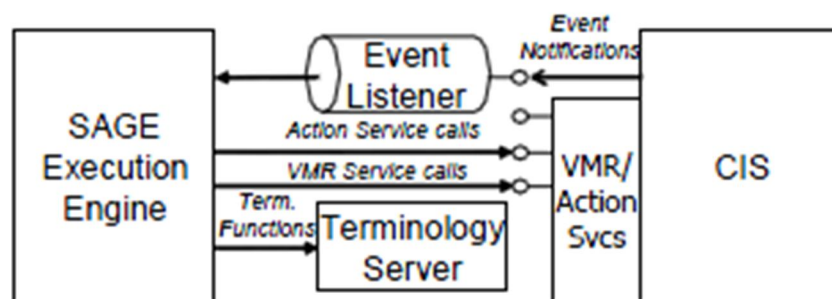


Figura 11 – Esquema da arquitectura de execução de SAGE (reproduzido de (Ram, et al., 2004)).

3.6. Análise

De forma a escolher a melhor forma de representação computacional de guias, é necessário definir alguns critérios a que os modelos devem obedecer, o que é possível através da análise das diferentes abordagens de CIGs.

Ao nível da codificação de GCs é importante considerar a facilidade e o tempo necessário para codificar guias, assim como o nível de qualificações necessário para o fazer. A codificação de guias deve ser possível mesmo a pessoal sem qualificações técnicas a nível informático. Por outro lado, é necessário que haja reprodutibilidade, ou seja, que múltiplas codificações da mesma guia produzam o mesmo resultado final. Para além de reprodutibilidade, a codificação deve produzir exactidão, na medida em que deve reproduzir fielmente o conteúdo da guia. A facilidade com que a representação de CIGs pode ser actualizada ou modificada é um aspecto importante, pelo que se deve ter em contas aspectos relativos à sua manutenção.

A introdução de melhorias também deve ser avaliada, por comparação com outras representações que não as possuam. Isto é particularmente importante no que toca ao apoio à decisão.

Ao nível da implementação em instituições de cuidados de saúde, deve ser avaliada a incorporação das guias nos sistemas de informação clínicos, nomeadamente a ligação entre os conceitos e variáveis representados nas CIGs com os conceitos e variáveis do sistema de informação clínica.

O *output* do sistema de CIGs é um dos pontos de discussão. Não existe um consenso sobre se o *output* deve ser apenas um conjunto de recomendações, um conjunto de recomendações juntamente com segmentos explicativos ou se deve haver uma ordem de preferências para as recomendações.

Cada um dos modelos de CIGs mencionados foca-se em diferentes aspectos da representação, desenvolvimento e implementação de guias. Ao passo que Arden Syntax e GLIF se concentram na padronização de GCs, PROforma foca-se em aspectos relacionados com a sua execução, Asbru concentra-se na representação temporal de GCs, e EON e SAGE dão especial importância ao estabelecimento de uma arquitectura que suporte o desenvolvimento, implementação e partilha de guias.

As características fundamentais de um modelo de representação de CIGs podem ser agrupadas da seguinte forma: primitivas de representação, linguagem (Wang, Peleg, Tu, Shortliffe, & Greenes, 2001) e modelação do fluxo do processo clínico (O'Connor, Shankar, Parrish, & Das, 2009). A Figura 12 fornece uma representação esquemática destas características.

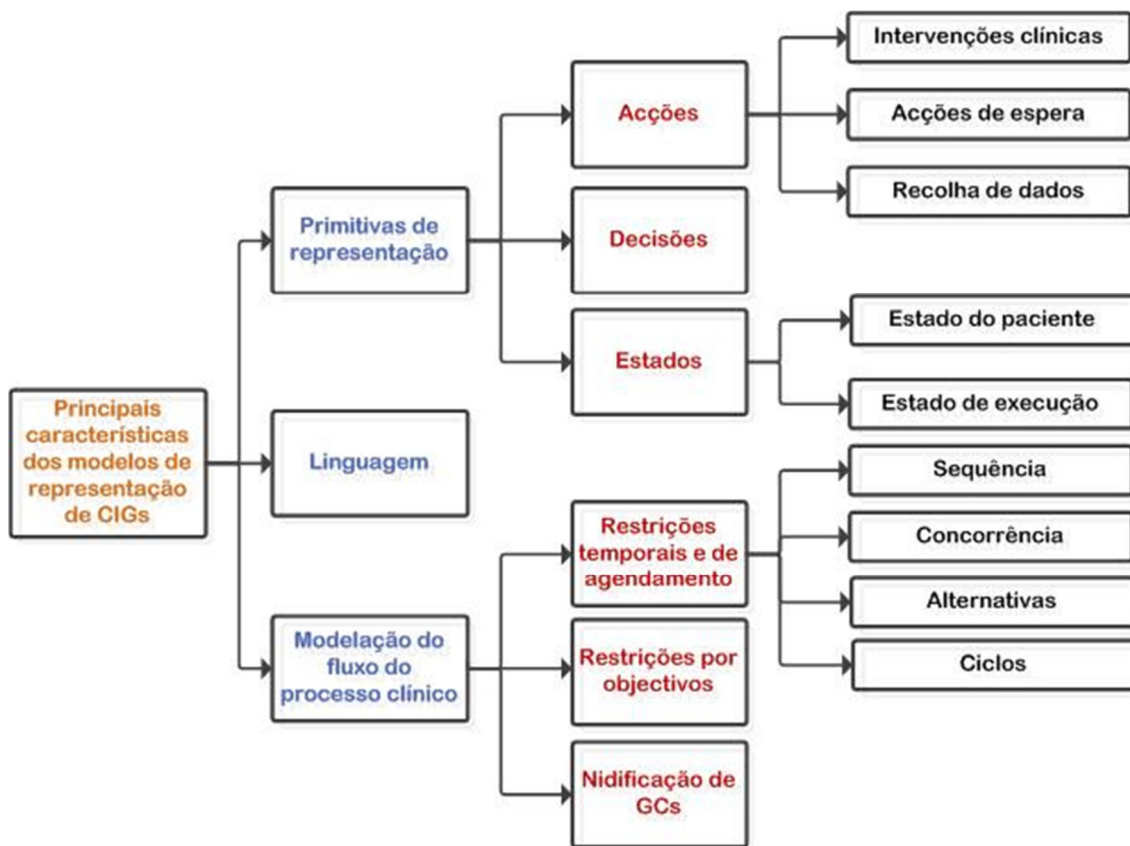


Figura 12 – Representação esquemática das principais características dos modelos de representação de CIGs.

As primitivas são um conjunto de blocos de informação, que devem captar todos os aspectos da informação de uma guia. São as classes base utilizadas na representação de guias.

Arden Syntax é a única das abordagens mencionadas que não modela guias com base num conjunto de primitivas, optando por considerar cada guia como um módulo independente, no qual está contido uma regra. Dada a simplicidade desta representação, a sua expressividade é reduzida, pelo que a sua utilização é mais adequada a sistemas de alerta e monitorização. GLIF, PROforma, Asbru e EON modelam guias de forma semelhante, na medida em que utilizam

primitivas para a sua construção sob a forma de passos, tarefas ou planos. GLIF e EON apresentam uma elevada semelhança, contudo GLIF apresenta um número fixo de primitivas (tal como PROforma, Asbru e SAGE), ao passo que é possível adicionar primitivas ao conjunto de primitivas básicas de EON. Neste aspecto, EON apresenta maior expressividade e uma melhor adequação às necessidades de informação do utilizador.

Existe um conjunto de primitivas básicas, que pela sua importância deve estar presente em cada um dos modelos. Estas primitivas básicas são acções, decisões e estados do paciente (pontos de entrada de dados). GLIF, EON e SAGE apresentam todas estas três primitivas, embora em SAGE os *patient states* existam sob a forma de uma integração com o VMR e as *actions*. Os MLMs de Arden Syntax também apresentam construtores semelhantes, sendo possível estabelecer uma correspondência entre os componentes *logic*, *action* e *evoke* e as primitivas decisão, acção e estado do paciente, respectivamente. Acções e decisões também podem ser encontradas em PROforma, contudo nesta abordagem não existe um suporte directo para o estado do paciente. Os *enquiries* de PROforma são vistos como acções tanto em EON como em GLIF. Em Asbru, a primitiva presente é *action* e cada um dos sub-planos consiste na sua essência numa *action*.

Todas as abordagens possuem uma forma de representação temporal. A estrutura mais complexa, e porventura mais completa, é a de Asbru. GLIF e EON adoptam um excerto de Asbru para representação temporal. De forma a ser compatível com Arden Syntax, GLIF define um conjunto de operadores comuns a Arden Syntax como *before*, *after* e *ago*, que apresentam paralelismos com os presentes em PROforma. PROforma define uma linguagem de expressão de restrições à execução de planos, que permite definir a sua duração e condições de repetição.

De forma a representar incerteza, Arden Syntax e GLIF apresentam três valores lógicos: *true*, *false* e *unknown*, mas não se realiza um tratamento desta informação. PROforma organiza um sistema de preferências, que considera a incerteza associada à informação na atribuição de pesos aos argumentos. Contudo, a representação de Informação Imperfeita não tem sido uma prioridade no desenvolvimento de CIGs.

Ao nível da complexidade dos modelos, todas as abordagens, excepto Arden Syntax, permitem a nidificação de sub-guias dentro das suas primitivas. GLIF, EON, SAGE possuem primitivas que podem conter uma referência a uma sub-guia ou sub-plano. Já em Asbru, cada *plan body* contém sub-planos que são percorridos até que se encontre um sub-plano que já não possa ser decomposto. PROforma permite a inclusão de planos em outros planos.

EON, PROforma e Asbru servem-se de *goals* e *intentions* para definir guias num nível de abstracção superior e que impõem restrições por objectivos á execução de GCs.

A Tabela 1 resume as principais características das abordagens em questão, quanto às primitivas de representação, modelação do fluxo do processo clínico e modelação dos dados do paciente.

Tabela 1 – Comparação das características de diferentes abordagens de CIGs (adaptado de (Wang, Peleg, Tu, Shortliffe, & Greenes, 2001)).

Modelo	Primitivas de representação		Modelação do fluxo do processo clínico				Modelação de dados do paciente
	Decisões	Acções	Estado do paciente	Estado de execução	Restrições temporais	Nidificação de guias	
Arden Syntax	<i>logic slot</i>	<i>action slot</i>	-	-	invocação de um módulo Asbru	-	-
EON	<i>decision</i>	<i>action, activity</i>	<i>scenario, activity state</i>	<i>activity state</i>	fluxograma (tarefas sequenciais)	sub-guia em <i>activities</i>	<i>Electronic Patient Record ontology</i>
PROforma	<i>decision</i>	<i>action, enquiry</i>	-	<i>task state</i>	linguagem de restrições temporais	<i>plans, subplans</i>	-
GLIF	<i>decision step</i>	<i>action step</i>	<i>patient state step</i>	-	invocação de um módulo Asbru	<i>sub-guia</i>	<i>patient state steps</i> e camada de conhecimento médico representação orientada a padrões temporais
Asbru	<i>condition, preference</i>	<i>plan</i>	<i>padrões temporais</i>	<i>plan state</i>	<i>plan body</i>	<i>plans, subplans</i>	orientada a padrões temporais
SAGE	<i>decision node</i>	<i>action node</i>	VMR	-	fluxograma (tarefas sequenciais)	sub-guias em <i>recommendation sets</i>	<i>Virtual Medical Record</i>

Em termos de linguagem, todas as abordagens definem uma linguagem que descreve a representação através de uma sintaxe formal. Arden Syntax, PROforma e Asbru utilizam BNF,

GLIF utiliza UML e EON baseia-se na sintaxe do Protégé. No que às guias em si diz respeito, Arden Syntax utiliza ficheiros ASCII e XML, GLIF utiliza XML, Asbru utiliza uma sintaxe semelhante a LISP (embora também se encontre disponível em XML) e PROforma utiliza R^L.

A linguagem GELLO é utilizada para expressar os critérios de decisão em GLIF e SAGE. O mesmo papel é desempenhado pela linguagem GEL em Asbru.

4. Representação do Conhecimento

4.1. Introdução

Neste capítulo aborda-se essencialmente a questão da representação do conhecimento, nomeadamente a representação do conhecimento a partir de Informação Imperfeita, que é de extrema importância no desenvolvimento de um modelo de representação de CIGs.

Em Inteligência Artificial a representação do conhecimento é um meio de comunicação, através do qual se transmite à máquina o conhecimento sobre o mundo real. É a base de qualquer sistema, pois se a representação não for suficientemente expressiva, não permite uma comunicação precisa. A representação deve também ser perceptível tanto para pessoas como para máquinas, de forma a ser fácil de utilizar. Quando se utiliza uma representação formal para descrever o mundo real, o conjunto de objectos do mundo real que é possível representar designa-se por universo de discurso. O formalismo de representação fornece um vocabulário para representar o universo de discurso e definições que associam o nome das entidades no universo do discurso ao seu significado. Entre as representações de conhecimento mais utilizadas encontra-se a lógica.

Neste capítulo começa-se por fazer uma distinção entre dados, informação e conhecimento. Em seguida aborda-se a Informação Imperfeita e métodos que permitem lidar com este aspecto da representação do conhecimento. Por fim aborda-se a Programação em Lógica (PL) como formalismo de representação do conhecimento em Inteligência Artificial.

4.2. Dados, Informação e Conhecimento

Existe uma interdependência entre dados, informação e conhecimento e cada um destes conceitos desempenha papéis no processo cognitivo (Cooper, 2010).

Dados são representações de actos, que simplesmente existem e não apresentam qualquer significado para além da sua existência, são apenas entidades sintácticas. Quando é atribuído significado aos dados, através do estabelecimento de relações entre os diferentes factos, obtém-se informação. O conhecimento resulta da estruturação e organização da informação como resultado de um processo cognitivo.

Do ponto de vista do processo de tomada de decisão (Aamodt & Nygård, 1995), ilustrado na Figura 13, os dados são a entrada de um processo de interpretação, o primeiro passo da tomada de decisão. A capacidade de um sistema de interpretar informação deriva do conhecimento que este possui. A saída do processo de interpretação é a informação, que por sua vez é a entrada e saída da segunda parte do processo de tomada de decisão baseado em conhecimento. Informação gera nova informação, através de um processo de elaboração. O conhecimento é o resultado da aprendizagem, é informação incorporada nas capacidades de raciocínio de um agente, que pode ser utilizada num processo de tomada de decisão.



Figura 13 -- O modelo dados-informação-conhecimento, segundo o processo de tomada de decisão (adaptado de (Aamodt & Nygård, 1995)).

4.3. Informação Imperfeita

Dado que, num sistema de apoio à decisão, se está a lidar com representações do mundo real, com toda a complexidade que este conceito acarreta, o conhecimento que se tem do mundo real não é completo. Tal como as pessoas, os sistemas de informação tomam decisões com base em informação incompleta e imprecisa, pelo que, este é um factor a ter em consideração na concepção dos modelos de representação.

A Informação Imperfeita pode ter diferentes origens (Morgan & Henrion, 1992), como por exemplo: informação incompleta, discordância entre fontes de informação, incerteza em relação ao modelo utilizado para aproximação à realidade, entre outras. Em última análise, a

introdução da informação num sistema é realizada por um ser humano, logo estará sempre sujeita a uma percepção subjectiva da realidade, que é inerente à condição humana, o que se pode verificar pela imprecisão da linguagem utilizada, por vezes, em observações para recolha de dados. A análise de dados quantitativos, que requerem medições, está sujeita a erros aleatórios provenientes dos aparelhos de medição ou da técnica de medida, que caso se propaguem em medições posteriores, dão origem a um erro sistemático, provocando um afastamento entre o valor real e a média das medições efectuadas. Muitas das quantidades a serem medidas variam ao longo do tempo e espaço, podendo apenas estimar-se o seu valor, ao passo que outras são totalmente imprevisíveis, pois resultam de fenómenos aleatórios.

A Informação Imperfeita pode assumir diferentes formas: incerteza, incompletude, inexactidão e incoerência (Morgan & Henrion, 1992). A incerteza ocorre quando falta informação acerca do mundo real e um agente apresenta uma percepção subjectiva sobre a veracidade ou falsidade de um facto, patente, por exemplo, na utilização da expressão “talvez”. Por sua vez, a incompletude significa que não existe informação relevante acerca de um dado atributo, ocorrendo em casos em que não se especifica um valor, mas sabe-se que ele existe, ou em casos em que o atributo se refere a uma característica que não é aplicável ao objecto em análise. A inexactidão refere-se aos casos em que o valor exacto de um atributo não é conhecido, mas sabe-se que se encontra num dado intervalo de valores ou conjunto de valores. Quanto à incoerência, refere-se à obtenção de informações contraditórias para um mesmo atributo.

A Informação Imperfeita é importante no domínio do conhecimento médico. Um sintoma é uma indicação, de algum modo incerta, de uma condição de saúde, uma vez que pode ou não ocorrer juntamente com a doença. Deste modo, deve ser utilizada uma medida de incerteza que permita estimar o risco de doença quando o sintoma é observado. No caso de imprecisões linguísticas, expressões como “febre elevada” ou “excesso de peso”, que são utilizadas diariamente na prática clínica para descrever sintomas, a aplicação de uma medida de imprecisão pode ser vantajosa. No domínio clínico, a incerteza está associada à relação entre sintomas e doenças, ao passo que a imprecisão está associada à representação de sintomas (Straszecka, 2006). Uma forma comum de ponderar sobre a incerteza é através de índices que se encontram em guias e que são utilizados para estimar o risco de doença, com base em pontuações atribuídas aos parâmetros de saúde. O risco é calculado através da soma algébrica destas pontuações (Schaffer, Levitt, Hershkop, Oh, MacDonald, & Lanctot, 2002).

4.4. Técnicas Numéricas para lidar com Informação Imperfeita

No ponto anterior abordou-se as diferentes formas que a Informação Imperfeita pode assumir. Neste ponto realiza-se a identificação das principais técnicas numéricas que permitem lidar com alguns casos de Informação Imperfeita. Estas técnicas derivam da teoria das probabilidades. Contudo, em alguns casos de Informação Imperfeita, não possuem expressividade suficiente e implicam exigências ao nível de informação e processamento que são pouco práticas (Sheridan, 1991).

4.4.1. Redes Bayesianas

As redes Bayesianas derivam da teoria das probabilidades e surgiram pela primeira vez em meados dos anos 80, através de um algoritmo que calculava de forma eficiente as probabilidades numa rede Bayesiana (Pearl, 1986). No início dos anos 90 as redes Bayesianas tornaram-se bastante populares, quando surgiu o primeiro *software* de modelação baseado neste modelo (Andersen, Olesen, Jensen, & Jensen, 1989). Nos anos que se seguiram, as redes Bayesianas foram aplicadas em diversas áreas, nomeadamente como ferramentas de reconhecimento de padrões e de classificação (Friedman, Geiger, & Goldszmidt, 1997).

Uma rede Bayesiana é um modelo estatístico que é definido em termos de dois componentes:

- Um componente qualitativo: um grafo orientado acíclico, no qual cada vértice representa uma variável aleatória, que pode assumir um dado conjunto de valores e tem associada a respectiva distribuição de probabilidades. A existência de um vértice que liga duas outras variáveis indica a existência de uma dependência estatística entre essas variáveis.
- Um componente quantitativo: uma distribuição condicional de probabilidades $p((x_i)|pa(x_i))$ para cada variável X_i , $i = 1, \dots, n$, dados os nodos pais, representados por $pa(x_i)$.

Uma vez definida a estrutura, é necessário determinar as relações entre as variáveis do grafo. Para o efeito recorre-se à distribuição conjunta de probabilidades, que é definida como o produto das distribuições das probabilidades condicionais de cada nodo, na forma:

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p((x_i)|pa(x_i)) \quad \forall x_1, \dots, x_n \in \Omega_{X_1, \dots, X_n} \quad (1)$$

em que Ω_{X_i} representa o conjunto de todos os valores possíveis de X_i .

As redes Bayesianas são modeladas de acordo com distribuições de probabilidades, o que permite obter estimativas de risco e incerteza de forma mais rigorosa que em modelos que apenas consideram valores médios. A utilização de probabilidades condicionais permite estabelecer relações de causa e efeito entre as variáveis do modelo, nomeadamente nos casos em que existe Informação Imperfeita acerca de alguma das variáveis. As redes Bayesianas permitem a modelação de sistemas complexos, com muitas variáveis de forma relativamente rápida.

Contudo, à medida que o número de variáveis aumenta, o processo de construção da rede necessita de mais dados para manter a precisão das estimativas (Uusitalo, 2007).

No domínio de implementações de CIGs, as redes Bayesianas são alvo de alguma investigação, nomeadamente na aplicação de redes Bayesianas dinâmicas na previsão de prognósticos, através da conjunção dos aspectos temporais e causais do domínio clínico (van, Taal, & Lucas, 2008). As redes Bayesianas também podem ser utilizadas na construção de GCs (Mani & Aliferis, 2007), utilizando técnicas de aprendizagem automática baseadas em redes Bayesianas para descobrir relações de causalidade entre sintomas e doenças.

4.4.2. Teoria de Dempster-Shafer

A teoria de Dempster-Shafer teve origem a partir da investigação sobre teoria das probabilidades com limite superior e inferior (Dempster, 1967), sendo desde então aumentada pela introdução do conceito de funções de crença (Shenoy & Shafer, 1986). As funções de crença permitem derivar graus de crença para uma questão com base em questões relacionadas.

A teoria de Dempster-Shafer postula que partindo de um conjunto de proposições, designado por Θ , finito e não vazio, de forma a conter todas as possíveis soluções para um problema, cada subconjunto de Θ é interpretado como uma hipótese para a resolução do problema, dando origem a 2^{Θ} hipóteses. Com base numa evidência relacionada com a questão, é atribuída uma probabilidade a cada subconjunto, através de uma função de massa, m . Os valores de crença situam-se no intervalo $[0,1]$, sendo que 0 representa a ausência de crença e 1 representa a certeza, de forma que a soma de todos os valores atribuídos, incluindo ao conjunto Θ , é 1.

A probabilidade que sobra após a atribuição de probabilidades aos subconjuntos é chamada de crença não atribuída, representada por $m(\theta)$. Se $m(A) = x$ e não é atribuída qualquer crença a outro subconjunto de Θ , então $m(\theta) = 1 - x$. A probabilidade que sobra é atribuída a Θ e não à negação de A , como acontece no modelo Bayesiano. Isto permite à teoria de Dempster-Shafer lidar com situações de falta de informação acerca do mundo real, na medida em que a falta de crença numa hipótese não significa a sua negação.

No entanto, a teoria assume que as evidências são independentes entre si, o que nem sempre se verifica. A teoria de Dempster-Shafer apresenta uma complexidade computacional elevada, o que torna difícil a sua implementação. Outro obstáculo à implementação é o facto de apenas lidar com conjuntos exclusivos e exaustivos de hipóteses (Liu, Yang, Wang, & Sii, 2002).

4.4.3. Conjuntos Fuzzy e Lógica Fuzzy

A teoria dos conjuntos *fuzzy* foi proposta em meados dos anos sessenta (Zadeh, 1965), da qual foi derivada a lógica *fuzzy* (Zadeh, 1975) (Zadeh, 1996).

Segundo a teoria clássica dos conjuntos, a pertença de um elemento a um conjunto é definida em termos binários, ou seja, um elemento pertence ou não pertence a um conjunto. Por outro lado, a teoria dos conjuntos fuzzy permite que um elemento pertença parcialmente a um conjunto, através de medidas de pertença. Seja X um espaço de objectos e A um conjunto *fuzzy*, a medida de pertença de X a A é calculada através de uma função de pertença $f_A(x)$, que associa a cada objecto em X um valor no intervalo $[0,1]$, em que 0 significa que o objecto não pertence ao conjunto e 1 significa que o objecto pertence inteiramente ao conjunto.

Exemplos de conjuntos fuzzy podem ser os conjuntos *altura* = {*baixo, mediano, alto*}, definidos por conceitos vagos. Neste caso quanto maior for o valor numérico da altura de um elemento, maior será a sua medida de pertença ao conjunto *alto* e menor será a sua medida de pertença ao conjunto *baixo*, e vice-versa.

A lógica *fuzzy* assenta na noção de grau de verdade de uma proposição, que, como foi acima mencionado, é um valor no intervalo $[0,1]$. A lógica fuzzy define operadores que expressam a disjunção e conjunção de proposições e apenas se foca no grau de verdade de uma proposição, independentemente do seu significado (Sheridan, 1991). Um exemplo é o grau de verdade de uma disjunção $P \vee Q$, em que o grau de verdade de P é 0.5 e o grau de verdade de

Q é 0.3. Neste caso o grau de verdade é 0.5 e no caso de $P \wedge Q$ seria 0.3, mesmo que P seja $\neg Q$.

Os conjuntos *fuzzy* têm sido alvo de um estudo intenso também na área de CIGs. É possível descrever uma forma de operacionalizar guias com recurso a lógica *fuzzy*, de forma a gerar diferentes conclusões com diferentes graus de verdade (Liu & Shiffman, 1997). Por outro lado, também se aposta em introduzir esta técnica de quantificação de Informação Imperfeita em formalismos existentes, como é o caso de Arden Syntax, descrevendo-se uma extensão a este formalismo, denominada Fuzzy Arden Syntax, de forma a lidar com informação vaga e com incerteza (Vetterlein, Mandl, & Adlassnig, 2010).

4.4.4. Lógica Probabilística

A lógica probabilística combina a capacidade da teoria das probabilidades para lidar com incerteza, com a capacidade da lógica para explorar a estrutura do conhecimento (Nilsson, 1986). A lógica é utilizada para deduzir novas proposições a partir de proposições existentes. Cada proposição tem associada uma probabilidade no intervalo $[0,1]$. A probabilidade das novas proposições é obtida a partir das probabilidades das proposições que lhes deram origem.

A probabilidade de uma proposição é a soma das probabilidades dos mundos em que a proposição é verdadeira (Sheridan, 1991). O ponto de partida é a probabilidade das proposições, a partir das quais se calcula a probabilidade dos mundos onde as proposições se verificam, para obter a probabilidade de outras proposições. No fundo, trata-se de considerar os diferentes cenários possíveis.

A lógica probabilística tem aplicações no domínio das CIGs, como é possível verificar no trabalho desenvolvido em (Lindgren & Eklund, 2005), onde se apresenta um sistema de apoio à decisão baseado nesta técnica, para colmatar falhas na detecção de condições raras no diagnóstico de doenças cognitivas.

4.5. Verificação da Validade de Guias através de Lógica Temporal

A utilização de lógica clássica na descrição dos mecanismos de raciocínio implícitos em GCs tem sido o objecto de estudo em (Hommersom, Lucas, & Bommel, 2008; Lucas, 2003), em que se

propõe a utilização de uma extensão à lógica clássica, na forma de lógica temporal para realizar a verificação da qualidade de GCs. Esta verificação de qualidade realiza-se através da comparação com indicadores de qualidade propostos por profissionais de saúde.

Em lugar de uma abordagem orientada aos objectos (e.g., GLIF, Asbru), na qual uma guia é traduzida para uma linguagem formal e depois utiliza-se métodos lógicos para analisar a representação e verificar se determinadas propriedades do domínio se verificam, neste caso utiliza-se uma meta-abordagem, na medida em que se identificam requisitos gerais de uma guia e depois se investiga se existe conformidade.

Para o efeito são definidos quatro operadores temporais que se encontram na Tabela 2. As relações entre as diferentes fórmulas lógicas que representam o conhecimento de uma guia estabelecem-se como uma ordem temporal, segundo a estrutura temporal $F \models (T, <, I)$, em que T é o conjunto de intervalos ou pontos temporais, $<$ representa a ordenação temporal e I é uma interpretação dos elementos da linguagem em relação a T e $<$.

Tabela 2 -- Operadores temporais, utilizados na lógica temporal para avaliação da qualidade de GCs (adaptado de (Lucas, 2003)).

Notação	Interpretação
$H\varphi$	φ foi sempre verdade no passado
$G\varphi$	φ será sempre verdade no futuro
$P\varphi$	φ foi verdade algures no passado
$F\varphi$	φ será verdade algures no futuro

Este método assume que o conhecimento de GCs divide-se em dois tipos:

- Conhecimento relativo aos mecanismos (pato)fisiológicos subjacentes a uma doença e a forma como o tratamento influencia estes mecanismos. O conhecimento apresenta uma natureza causal;
- Conhecimento relativo a boas práticas na selecção de tratamentos, meta-conhecimento.

No caso do conhecimento sobre prescrição de medicamentos, este poderia ser formalizado da seguinte forma:

$$(Gd \wedge r) \rightarrow G(m_1 \wedge \dots \wedge m_n) \quad (2)$$

em que d é o nome de um medicamento ou conjunto de medicamentos, r corresponde a um requisito para que o medicamento faça efeito e m_k é um modo de actuação do medicamento, como por exemplo, redução da libertação de glicose pelo fígado, caso o medicamento fosse para controlo de diabetes. Os modos de acção podem ser combinados com uma intenção n (e.g., alcançar um nível normal de glicemia), uma condição do paciente c e requisitos r_j para os modos de acção serem eficazes, da seguinte forma:

$$(Gm_{i_1} \wedge \dots \wedge Gm_{i_m} \wedge r_1 \wedge \dots \wedge r_p \wedge Hc) \rightarrow Gn \quad (3)$$

É possível formalizar as boas práticas clínicas através do conhecimento do *background* do paciente (B), de um conjunto de medicamentos ($T = \{d_1, \dots, d_p\}$), um conjunto de condições de pacientes (C), um conjunto de requisitos (R) e um conjunto de intenções (N) que o profissional de saúde pretende cumprir. Um conjunto de medicamentos T corresponde a um tratamento se:

- $B \cup GT \cup C \cup R \not\equiv \perp$ (os medicamentos não têm efeitos opostos);
- $B \cup GT \cup C \cup R \models N$ (os medicamentos respondem a todos os problemas do paciente que fazem parte das intenções).

A verificação da qualidade das guias é realizada através de um demonstrador de teoremas, denominado KIV (Fensel & Schnogge, 1997).

4.6. Programação em Lógica

As origens da Programação em Lógica (PL) remontam a (McCarthy, 1959). De um modo geral, consiste na utilização da lógica matemática para resolver problemas de programação. A programação em lógica foi evoluindo através do desenvolvimento de formalismos de Raciocínio Não Monótono (Ginsberg, 1991), como o Pressuposto do Mundo Fechado (McCarthy, 1959). O Pressuposto do Mundo Fechado postula que toda a informação que não está mencionada na base de conhecimento é considerada falsa, introduzindo assim a negação por falha na prova.

A definição de PL como lógica de predicados surgiu mais tarde (Emden & Kowalski, 1976), conjuntamente com a primeira linguagem de PL, o Prolog. O Prolog (Colmerauer, Kanoui, Pasero, & Roussel, 1972) é uma linguagem declarativa, ou seja, um programa em Prolog é

expresso em termos de relações, representados como factos e regras. A computação inicia-se quando se realiza uma interrogação sobre estas relações.

Uma linguagem de PL é constituída por constantes, funções e predicados. As constantes são introduzidas através de átomos. Os átomos apresentam-se na forma:

$$p(t_1, \dots, t_m) \quad (4)$$

em que p é um predicado de cardinalidade m , e t_i são termos. Um átomo, por si, corresponde a um literal positivo e a negação de um átomo corresponde a um literal negativo. As regras apresentam-se na forma:

$$A_0 \leftarrow A_1, \dots, A_m, \text{not } A_{m+1}, \dots, A_n \quad (5)$$

com $m, n \geq 0$, em que A_i são literais e *not* representa a negação por falha na prova. A parte esquerda da regra corresponde à conclusão e designa-se por cabeça, ao passo que a parte direita corresponde às premissas, designando-se por corpo. A Equação (5) representa uma disjunção de literais, designada por cláusula. Em Prolog as cláusulas utilizadas são cláusulas de Horn.

Num programa em lógica, a resposta a uma questão pode ser apenas de dois tipos: verdadeiro ou falso. Isto deve-se às limitações que um programa em lógica apresenta no que diz respeito à representação do conhecimento, ao não permitir a representação explícita de informação negativa. No mundo real, no qual se inclui o domínio clínico, a ausência de informação não significa a sua falsidade. A PL não permite inferir sobre Informação Imperfeita, pois não considera os casos de ausência de informação, de informação contraditória ou imprecisa.

4.7. Extensão à Programação em Lógica

A Extensão à Programação em Lógica (EPL) (Analide, Novais, Machado, & Neves, 2006; Neves, Machado, Analide, Abelha, & Brito, 2007), utiliza a negação forte (Neves, 1984), também designada por negação clássica ou explícita, para representar explicitamente informação negativa. Isto permite criar uma distinção entre uma questão que é falsa porque não pode ser

provada ou uma questão que é falsa porque pode ser provada a sua negação. A negação forte representa-se por \neg e a forma de um programa em lógica estendida é a seguinte:

$$A_0 \leftarrow A_1, \dots, A_m, \text{not } A_{m+1}, \dots, A_n \quad (6)$$

com $m, n \geq 0$, em que A_i são literais do tipo p ou $\neg p$.

A EPL utiliza a representação de valores nulos (Analide, Novais, Machado, & Neves, 2006) como estratégia de enumeração de casos de informação imperfeita, permitindo a distinção entre diferentes situações em que as respostas são conhecidas (verdadeiras ou falsas) ou desconhecidas. Os valores nulos podem ser de dois tipos: o primeiro permite a representação de valores desconhecidos, não necessariamente de um conjunto de valores, e o segundo permite a representação de valores desconhecidos, a partir de um conjunto de valores. De seguida formular-se-ão exemplos de como estes valores podem ser utilizados.

Cada programa está associado a um conjunto de hipóteses ou explicações para as interrogações. Estas hipóteses podem ser representadas na forma de excepções às extensões dos predicados que compõem o programa. Esta característica denota a utilização da abdução (Kakas, Kowalski, & Toni, 1998) como forma de Raciocínio Não Monótono pela EPL.

Considere-se a representação do parâmetro temperatura corporal da seguinte forma:

```
temperature: value
```

em que o argumento do predicado `temperature` representa o valor em graus *celsius* (°C) da temperatura de um paciente.

Para realizar um diagnóstico de gripe, um profissional de saúde precisa de saber se o paciente apresenta febre, ou seja, se este apresenta uma temperatura corporal superior a 37.5°C. Se o profissional de saúde não possuir meios para medir a temperatura, desconhece o seu valor. A temperatura pode ser superior ou inferior ao valor 37.5°C, logo, é permitido ao profissional de saúde assumir que o paciente tem febre, mas ele não pode afirmar com certeza tal facto. Esta situação retrata um valor nulo do tipo desconhecido e a correspondente extensão do predicado `temperature` encontra-se representada no Programa 1.

```
-temperature(X) ← not temperature(X), not abducible(temperature(X)).  
abducible(temperature(X)) ← temperature(⊥).  
temperature(⊥).
```

Programa 1 -- Extensão do predicado `temperature` com representação de um valor nulo do tipo desconhecido.

A primeira cláusula denota a oclusão do predicado `temperature`, ou seja, considera-se que há negação forte de `temperature` quando este predicado não é representado com informação positiva na base de conhecimento e não existe nenhuma hipótese, na forma do predicado `abducible`, para o valor da temperatura. A segunda cláusula representa que `temperature` pode tomar qualquer valor se for representado com o símbolo \perp , que representa o valor nulo do tipo desconhecido. A terceira cláusula corresponde à representação do valor nulo desconhecido, em que qualquer valor é uma possível solução para a temperatura, mas não há uma hipótese sobre qual é esse valor. Desta forma, descarta-se a possibilidade de assumir como falsa qualquer interrogação sobre o predicado `temperature`.

Se o profissional de saúde efectuar várias medições de temperatura e estas medições fornecerem valores diferentes, não há como ter a certeza sobre qual deles o correcto. Neste caso está-se perante um valor nulo do tipo desconhecido de um conjunto de valores. Considere-se que o profissional de saúde efectuou três medições e que os valores obtidos são: 36.5°C, 37°C e 37.6°C. A extensão do predicado `temperature` para esta situação encontra-se no Programa 2.

```
-temperature(X) ← not temperature(X), not abducible(temperature(X)).  
abducible(temperature(36.5)).  
abducible(temperature(37)).  
abducible(temperature(37.6)).
```

Programa 2 -- Extensão do predicado `temperature` com representação de um valor nulo do tipo desconhecido de um conjunto de valores.

As três últimas cláusulas do Programa 2 representam o facto de que o valor de temperatura do paciente é desconhecido, mas está no conjunto de valores {36.5,37,37.6}.

O valor nulo do tipo desconhecido de um conjunto de valores pode apresentar uma variação, que é a representada pelo Programa 3.

```
-temperature(X) ← not temperature(X), not abducible(temperature(X)).
abducible(temperature(35)).
abducible(temperature(37.7)).
? ((abducible(temperature(X1)) ∨ abducible(temperature(X2)) ∧
¬(abducible(temperature(X1)) ∧ abducible(temperature(X2))).
```

Programa 3 -- Extensão do predicado `temperature` com representação de um valor nulo do tipo desconhecido de um conjunto de valores, com invariante.

No caso do Programa 3, a última cláusula representa um invariante, definido segundo um operador XOR. Este invariante afirma que a extensão do predicado `temperature` permite os termos `temperature(35)` ou `temperature(37.7)`, mas não ambos.

A inferência sobre o corpo de conhecimento em EPL pode traduzir-se na extensão de um predicado `demo`. Este meta-predicado pode ser definido nos termos $demo:T, V \rightarrow \{true, false, unknown\}$ (Analide, Novais, Machado, & Neves, 2006), segundo as cláusulas que se seguem:

```
demo(T, true) ← T.
demo(T, false) ← ¬ T.
demo(T, unknown) ← not T, not ¬ T.
```

Programa 4 -- Extensão do meta-predicado `demo` (reproduzido de (Analide, Novais, Machado, & Neves, 2006)).

A primeira cláusula estabelece que quando se pretende provar um teorema, este é exposto a uma base de conhecimento de informação positiva, retornando o valor de verdade `true`. A segunda cláusula denota que o teorema recorre a informação negativa na base de conhecimento, retornando o valor de verdade `false`. A terceira cláusula atribui o valor de verdade `unknown` ao teorema que não pode ser provado recorrendo a informação positiva nem informação negativa.

4.8. Qualidade da Informação

A Qol (Neves, Machado, Analide, Abelha, & Brito, 2007; Novais, Salazar, Ribeiro, Analide, & Neves, 2010) é um modelo de quantificação da informação que foi desenvolvido a partir da EPL. A Qol da extensão de um predicado p é dada em termos de valores de verdade no intervalo $[0,1]$, isto é, se a informação é conhecida (positiva ou negativa), a Qol é 1. Para casos em que a informação é desconhecida, a Qol é dada pela expressão:

$$QoI_p = \lim_{N \rightarrow +\infty} \frac{1}{N} = 0 \quad (N \gg 0) \quad (7)$$

em que N é o número de termos candidatos a solução. Em casos como o Programa 1, em que $N \gg 0$, a Qol da extensão do predicado `temperature` é 0. Para situações em que a extensão de um predicado i é desconhecida, mas pode ser obtida de um conjunto de valores, a Qol é dada por:

$$QoI_p = \frac{1}{card} \quad (8)$$

em que $Card$ corresponde ao número de hipóteses de p , caso o conjunto de hipóteses seja disjunto, como no caso do Programa 3. No caso do conjunto de hipóteses não ser disjunto, a Qol é dada por:

$$QoI_p = \frac{1}{C_1^{Card} + \dots + C_{Card}^{Card}} = \frac{1}{\sum_{j=1}^{Card} C_j^{Card-1}} = \frac{1}{2^{Card-1}} \quad (9)$$

em que C_i^{Card} , $i = 1, \dots, Card$, representa um conjunto de combinações com $Card$ elementos.

O modelo de Qol apresenta outro aspecto que é a importância relativa que um predicado atribui a cada um dos seus atributos. A importância relativa é representada por w_i^k , que significa a relevância do atributo k na extensão do predicado i . A Qol assume que todos os pesos estão normalizados, tal que:

$$\sum_{1 \leq k \leq n} w_i^k = 1, \forall i \quad (10)$$

Através de (7) torna-se possível definir uma função de pontuação de predicados, ou seja, para um valor $x = (x_1, \dots, x_n)$ num espaço multidimensional tem-se:

$$V_i(x) = (\sum_{1 \leq k \leq n} w_i^k \times QoI_i(x))/n \quad (11)$$

em que $V_i(x)$ representa a pontuação atribuída à extensão do predicado i .

O método de QoI permite a construção de cenários, nos quais se montam todos os diferentes casos possíveis que podem ocorrer num dado sistema, constituído por y predicados, cada um deles com n atributos (Neves, Machado, Analide, Abelha, & Brito, 2007). Em cada cenário, os predicados apresentam uma dada pontuação, o que torna possível selecção do cenário que maximize as pontuações dos seus predicados, a melhor opção possível.

4.9. Análise

A questão da representação do conhecimento é de extrema importância em qualquer SADC. A forma como se representa o conhecimento dita a forma como se obtém nova informação a partir de informação existente. Sendo uma GC, um conjunto de recomendações sobre boas práticas clínicas com uma base científica sólida, que visa guiar os profissionais de saúde ao longo do processo clínico, pode dizer-se que desempenha a função do profissional de saúde ideal, colmatando eventuais vazios de conhecimento do profissional de saúde real. Deste modo, a programação subjacente a um SADC, baseado em guias, deve simular o pensamento de um profissional de saúde, percebendo o estado do mundo real e deixando a cargo do mecanismo de raciocínio a derivação de nova informação. A utilização de métodos formais baseados em lógica na estruturação de conhecimento tem-se revelado promissora, como se pode verificar através da verificação da qualidade de guias através de lógica temporal. Neste aspecto, a PL assume-se como o formalismo mais relevante, permitindo a concepção de mecanismos de inferência especializados.

A decisão consiste na escolha entre duas ou mais opções de acção, e nesta escolha identifica-se fontes de informação, elabora-se argumentos a favor e contra uma opção e realiza-se a acção. A PL ocupa-se desta estruturação do conhecimento. No entanto, no mundo real a informação disponível é imperfeita, o que obriga ao desenvolvimento de mecanismos que permitam lidar com este tipo de informação.

A quantificação da Informação Imperfeita pode fornecer um meio de estabelecer relações de dependência entre parâmetros, como acontece, por exemplo, nas redes Bayesianas. Contudo, grande parte dos métodos numéricos, que permitem lidar com Informação Imperfeita, baseia-se em probabilidades, o que se pode revelar desvantajoso, na medida em que é necessário realizar estimativas de distribuição de probabilidades, que por vezes não descrevem o mundo real. Outros métodos como os conjuntos fuzzy trabalham com conjuntos exaustivos de soluções, o que nem sempre é possível, pois pode não se conhecer todas as soluções para um problema. A grande maioria dos métodos não aborda todos os aspectos da Informação Imperfeita, nomeadamente os casos de incompletude da informação.

A solução passa pela combinação de uma representação de conhecimento com elevada expressividade, como é o caso da ELP, que permite a representação de informação incerta, incompleta, inexacta e incoerente, com um método de quantificação da informação que preveja métodos de cálculo para estes casos de Informação Imperfeita, como é o caso da QoI.

A integração de ELP e QoI num modelo de representação de CIGs pode revelar-se vantajosa, permitindo a representação de conhecimento que as abordagens actuais de CIGs não são capazes de representar.

5. Proposta de um Modelo de CIGs

5.1. Introdução

Com base na análise dos diferentes modelos de representação de CIGs, apresenta-se de seguida uma proposta de um modelo capaz de representar o conhecimento contido em GCs e ao mesmo tempo lidar com Informação Imperfeita.

O formalismo que se pretende propor para a representação de guias baseia-se em PROforma (Fox & Das, 2000; Vollebregt, Teije, Harmelen, der, & Mosseveld, 1999), procurando tirar partido das suas potencialidades.

As razões que elevaram à escolha de PROforma, como base para este modelo, prendem-se com o conjunto de características que este formalismo reúne e que vão de encontro aos objectivos deste trabalho de dissertação. É importante a forma de apresentação da guia ao utilizador, e neste aspecto, a apresentação sob a forma de fluxograma é a mais adequada. Os fluxogramas ilustram de forma simples a transição de informação entre os elementos que os constituem, o que permite a quem o está a visualizar uma rápida compreensão geral do processo em questão. PROforma, de forma semelhante a GLIF, representa guias como um grafo orientado e foca-se sobretudo na execução de guias e na tomada de decisão. PROforma apresenta uma forma de representação temporal que não é demasiado complexa e possui atributos nas tarefas que permitem expressar objectivos ou intenções. Em PROforma utiliza-se uma linguagem lógica de predicados (L_{R^2}), o que torna mais fácil uma adaptação a Prolog. Ao contrário de muitas das abordagens mencionadas, PROforma não pretende modelar um registo clínico do paciente, mas sim os aspectos mais relevantes na tomada de decisão, o que também vai de encontro ao pretendido. Embora este modelo não apresente um suporte explícito para o estado do paciente, é possível modelá-lo recorrendo a *triggers* e outros atributos do mesmo tipo.

De seguida apresenta-se o modelo desenvolvido para a representação de guias, através da abordagem de três tópicos essenciais: primitivas de representação, representação do conhecimento e formato de partilha e validação.

5.2. Primitivas de Representação

O modelo de representação proposto foi concebido de forma a suportar operações que ocorrem num processo clínico (Vollebregt, Teije, Harmelen, der, & Mosseveld, 1999). Estas operações são:

- *Triggers*: ocorrência de estados clínicos ou outros eventos que provocam o reconhecimento de problemas clínicos que requerem solução;
- *Resolução de problemas*: identificação das possíveis soluções para um problema. Se o problema é um diagnóstico, então as soluções alternativas serão explicações para o estado do paciente; se o problema é o tratamento de uma condição de saúde, então as soluções são medicamentos ou tratamentos alternativos. Esta operação é formalizada através de uma tarefa de decisão, lógica dedutiva e abductiva;
- *Argumentação*: avaliação da solidez das alternativas para um problema clínico. Esta operação é possível através da utilização de QoI, que fornece uma base qualitativa e quantitativa para a tomada de decisão;
- *Adopção de estratégias*: os profissionais de saúde deparam-se com dois tipos de decisão: decisões acerca do que acreditar e decisões acerca do que fazer. Em ambos os tipos de decisões avalia-se os argumentos a favor e contra, estabelecendo-se uma ordem de preferência entre as diferentes soluções;
- *Agendamento*: definição da sequência das tarefas clínicas que constituem o processo clínico;
- *Aquisição de dados*: operações de recolha de dados do estado do paciente.

Este modelo é orientado aos objectos e tem como unidade fundamental a tarefa. Neste contexto, uma guia é modelada como um conjunto de tarefas, dando origem a um plano. São suportadas quatro classes básicas de tarefas:

- *Decision*: uma tarefa que envolve uma escolha de algum tipo, uma decisão de diagnóstico, tratamento ou investigação. Numa tarefa de decisão define-se as diferentes opções disponíveis, informação relevante e as regras que ditam a escolha;
- *Action*: um procedimento clínico que só pode ser executado fora do sistema, como por exemplo a administração de uma injeção;
- *Enquiry*: uma acção cujo objectivo é a obtenção de informação, onde se especifica o tipo de parâmetro a obter e, caso seja necessário, o método pelo qual este é obtido;

- *Plan*: um conjunto de tarefas que devem ser executadas de forma a atingir um objectivo clínico. Um plano é constituído por um número variável de tarefas, ou mesmo sub-planos. É possível definir a sequência pela qual as tarefas são executadas, a duração do plano e se plano se deve repetir ou não;

Cada tarefa é vista como uma instância de uma classe mais geral de tarefas. A relação entre as diferentes classes de tarefas está ilustrada na Figura 14. Cada GC (plano) está inserida numa classe de tarefas designada por *root task*, consoante o seu objectivo seja diagnóstico, tratamento ou monitorização. De seguida aborda-se as diferentes classes e os atributos que as constituem, referindo exemplos da sua utilização.

De seguida aborda-se cada uma das classes de tarefas através de exemplos, retirados de uma GC para detecção de doenças respiratórias em crianças e adultos (ICSI, 2011).

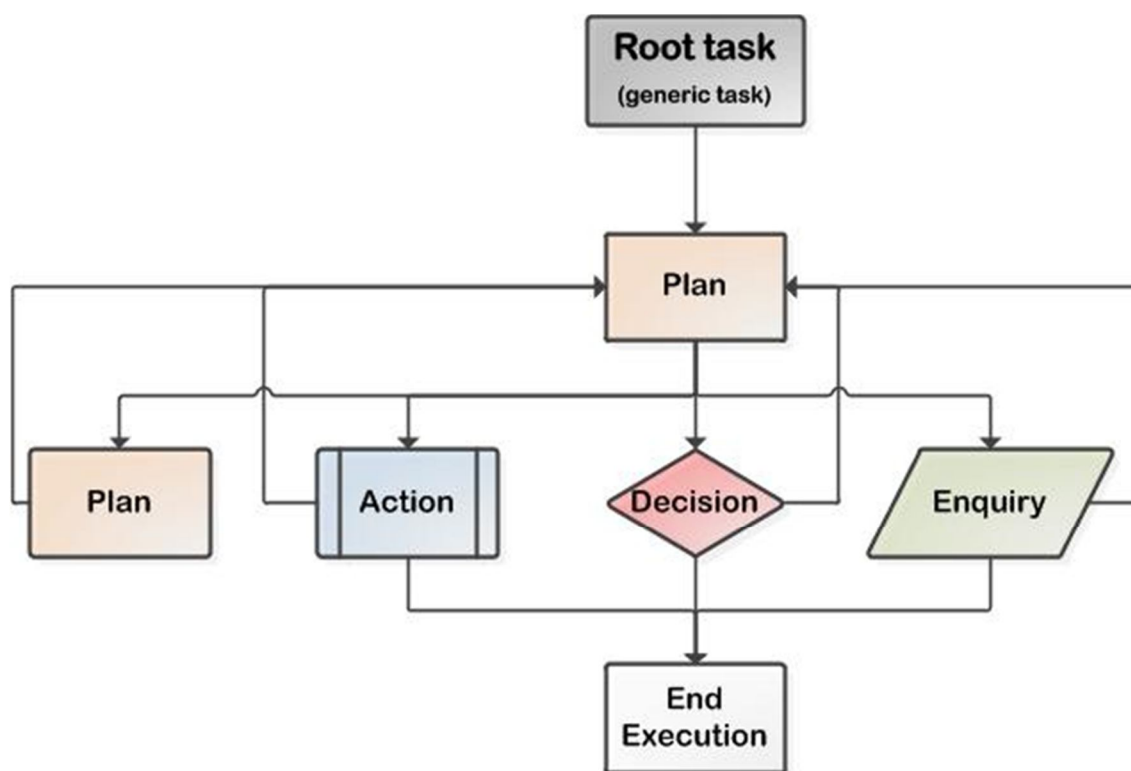


Figura 14 -- Relação entre as diferentes classes de tarefas. *Actions*, *decisions* e *enquiries* são tarefas atómicas, ao passo que *plans* são tarefas compostas, na medida em que podem ser constituídos por um número variável de tarefas atómicas e sub-planos.

5.2.1. *Root task*

Como já foi dito, a *root task* representa uma classe geral de tarefas, albergando os planos que possuem orientação comum, por exemplo, guias para diagnóstico ou tratamento. A Tabela 3 apresenta os atributos que uma *root task* possui e exemplos para o seu preenchimento.

Tabela 3 – Atributos de uma *root task* e respectivos exemplos.

Atributo	Obrigatoriedade	Exemplo
Name	Sim	RT1
Caption	Sim	Guidelines for diagnosis
Description	Sim	Procedures to determine the cause of a health condition.
Components	Sim	P1,P2,P3,....
Goal	Não	diagnosis(yes).

O atributo *name* é do tipo texto, obrigatório, e consiste num identificador único da tarefa. Convencionou-se que este identificador é constituído pelas iniciais do tipo de tarefa (e.g., *root task*- RT) seguidas de um valor numérico, que no caso do exemplo da tabela é 1. Duas tarefas diferentes possuem identificadores diferentes, uma vez que *name* é utilizado para fazer referência à tarefa ao longo dos planos.

De seguida, no atributo *caption*, também do tipo texto, é colocado o nome da guia a ser utilizado quando o nome deve ser impresso. O atributo *caption* é obrigatório.

O atributo *description* consiste num texto descritivo da tarefa, fornecendo um maior detalhe, como por exemplo a sua finalidade ou uma explicação alargada daquilo em que consiste a tarefa, no caso do exemplo, especifica-se que as guias para diagnóstico são procedimentos para determinar a causa de uma condição de saúde. Este atributo apresenta um carácter obrigatório.

O atributo *components* corresponde a um atributo obrigatório que possui a colecção de planos que a *root task* possui. Não existe um número máximo para o número de planos que uma *root task* pode possuir. Este atributo apenas pode conter *plans*, que se encontram representados pelo seu *name*.

Por sua vez, o atributo *goal* contém o objectivo da *root task*, na forma de um predicado na linguagem de programação em lógica Prolog. No final da execução do plano pretendido, a aplicação subjacente ao modelo deve verificar se o objectivo se encontra na base de conhecimento. No caso da interrogação sobre o objectivo resultar em verdadeiro, a execução deverá terminar, pois o problema foi resolvido. Por outro lado, se o resultado da interrogação for falso, o sistema deverá sugerir outro plano. No caso do exemplo, o objectivo é alcançar um diagnóstico. Este atributo possui um carácter opcional.

5.2.2. *Plan*

Um *plan* define um conjunto de tarefas, que podem ser necessárias num dado procedimento clínico. *Plan* partilha com *root task* os atributos *name*, *caption*, *description*, *components* e *goal*. A tabela 4 apresenta os atributos de um *plan* e exemplos da sua utilização.

Tabela 4 – Atributos de um *plan* e respectivos exemplos.

Atributo	Obrigatoriedade	Exemplo
Name	Sim	Exemplo 1: P1 Exemplo 2: P6 Exemplo 3: P7
Caption	Sim	Exemplo 1: Diagnosis and treatment of respiratory illness Exemplo 2: Drug treatment for allergic rhinitis Exemplo 3: Test for streptococcus pharyngitis
Description	Sim	Exemplo 1: Procedures to determine if an individual is suffering from respiratory illness. Exemplo 2: Set of drugs to mitigate the symptoms of allergic rhinitis. Exemplo 3: Procedures for conducting

			the test for streptococcus pharyngitis.
Author		Não	Exemplo 1: Institute for Clinical Systems Improvement Exemplo 2: --- Exemplo 3: ---
Date		Não	Exemplo 1: 01-01-2011 Exemplo 2: --- Exemplo 3: ---
Version		Não	Exemplo 1: 3.0 Exemplo 2: --- Exemplo 3: ---
Components		Sim	Exemplo 1: P4,P5,D1,A1,E1 Exemplo 2: A4,A5,A6,E2 Exemplo 3: A7,A8,A9,E3
Scheduling constraints		Sim	Exemplo 1: P4-E1-D1-(P5 OR A1) Exemplo 2: (A4 AND A5 AND A6)-E2 Exemplo 3: A7-A8-A9-E3
Trigger condition		Não	Exemplo 1: --- Exemplo 2: allergic_rhynitis(yes). Exemplo 3: rheumatic_fever(yes).
Duration	Value	Não	Exemplo 1: ---
	Unit		---
	Value		Exemplo 2: 2
	Unit		month
	Value		Exemplo 3: ---
	Unit		---
Periodicity	Value	Não	Exemplo 1: ---
	Unit		---
	Value		Exemplo 2: 1
	Unit		day
	Value		Exemplo 3: ---

Unit		---
Repetitions	Não	<p>Exemplo 1: ---</p> <p>Exemplo 2: ---</p> <p>Exemplo 3: 2</p>
Goal	Não	<p>Exemplo 1: ---</p> <p>Exemplo 2: allergic_symptoms(no).</p> <p>Exemplo 3: ---</p>

Os atributos *name*, *caption* e *description* desempenham em *plan*, as mesmas funções e são do mesmo tipo que em *root task*, como é possível verificar através dos valores dos atributos dos exemplos.

Os atributos *author*, *date* e *version* desempenham uma função administrativa. Uma vez que as guias são representadas como planos e podem ser partilhadas, é importante representar informação de autoria, a data em que foram criadas e a sua versão actual. O atributo *author* é do tipo texto e contém o autor da guia. Já o atributo *date* é uma data no formato DD-MM-AA (DD – dia, MM – mês, AA – ano) e representa a data de criação da guia. O atributo *version* é um valor numérico que representa a versão actual da guia. Estes três atributos são opcionais, pois um plano pode representar uma guia completa ou um subconjunto de tarefas que compõe uma guia.

A função de *components*, neste contexto, é igual à desempenhada em *root task*, com a diferença de permitir a inclusão de todo o tipo de tarefas e não apenas de planos.

Em *scheduling constraints* define-se a ordem relativa pela qual as tarefas se realizam. Trata-se de um atributo do tipo texto, obrigatório, no qual se coloca a sequência de tarefas, segundo a seguinte notação:

- Uma tarefa que ocorre após outra está separada da tarefa anterior pelo operador “-”;
- Duas tarefas que ocorrem simultaneamente estão interligadas pelo operador “AND”;
- Duas tarefas mutuamente exclusivas estão interligadas pelo operador “OR”, o que significa que apenas se realiza uma das tarefas. A utilização do operador “OR” ocorre em tarefas que sucedem uma tarefa de decisão. É o resultado da decisão que determina qual das tarefas, interligadas por “OR”, se irá realizar;
- Utiliza-se parênteses de forma a agrupar tarefas interligadas por “OR” ou “AND”.

Segundo esta notação, no caso do Exemplo 1 da Tabela 4, as tarefas P4, E1 e D1 são executadas sequencialmente. De seguida surge uma bifurcação no fluxograma do processo clínico, com duas tarefas alternativas. A escolha de qual das tarefas será executada é ditada pela tarefa D1. Por sua vez, no Exemplo 2, há uma ramificação do fluxograma do processo clínico que indica a execução simultânea das tarefas A4,A5 e A6, seguida da sincronização do fluxo de trabalho para a execução da tarefa E2. O Exemplo 3 é um caso normal de execução sequencial das tarefas A7, A8,A9 e E3. O *parser*, de uma eventual aplicação que executa guias neste modelo, deve ser capaz de interpretar as *scheduling constraints* e construir o fluxograma que representa a guia.

No atributo *trigger condition* coloca-se a condição que despoleta a execução de uma tarefa, que neste caso é um plano. Este atributo corresponde a um predicado em linguagem de programação Prolog. As tarefas que sucedem um passo de decisão contêm *trigger conditions* que correspondem às diferentes escolhas que se podem realizar na decisão. Após a decisão, o motor de execução verifica as *trigger conditions* das tarefas seguintes, de forma a encontrar a tarefa cuja *trigger condition* corresponde à alternativa seleccionada. No caso do Exemplo 2, o plano de tratamento à base de fármacos para rinite alérgica, apenas se iniciará se verificar-se que o paciente possui rinite alérgica. Já no Exemplo 3, só se executará o plano que contém as tarefas para a realização de um teste à presença da bactéria *streptococcus*, caso se verifique que o paciente possui febre reumática, que é um indicio de faringite provocada por este microorganismo. *Trigger condition* é um atributo opcional, visto nem sempre ser necessário. Não há limite para o número de *trigger conditions* que uma tarefa pode apresentar.

Os atributos *duration*, *repetitions* e *periodicity* permitem aplicar restrições temporais à execução de tarefas.

O atributo *duration* é opcional e corresponde à duração que uma tarefa deve ter. Esta duração define-se através de dois sub-atributos: *value* e *unit*. *Value* corresponde a um valor numérico para a duração e *unit* à unidade utilizada para definir a duração. *Unit* apenas pode tomar cinco valores: *minute*, *hour*, *day*, *week*, *month*. O motor de execução deve verificar se o plano tem duração e apenas passa à tarefa seguinte quando o tempo definido na duração terminar.

Repetitions corresponde ao número de vezes que uma tarefa deve ser executada. Corresponde a um valor numérico inteiro e também é um atributo opcional. No Exemplo 3, o

plano P7 está definido para ser executado duas vezes, de forma a confirmar o resultado do teste à presença de *streptococcus*.

O atributo *periodicity* permite definir ciclos de tarefas com base num valor numérico (*value*) e numa unidade (*unit*). Tal como no atributo *duration*, *unit* apenas pode tomar os valores: *minute*, *hour*, *day*, *week*, *month*. Este é um atributo opcional, que pode ser utilizado em conjugação com *duration*. No Exemplo 2 declara-se que o plano P6 apresenta a duração de dois meses e uma periodicidade de um em um dia. O plano P6 corresponde à toma de um conjunto de fármacos que visam mitigar os sintomas de alergia e, tal como está definido, estes fármacos devem ser tomados com o espaço de um dia entre cada toma, durante dois meses. No final dos dois meses executa-se a tarefa que se segue ao plano P6.

Por fim, o atributo *goal* apresenta a mesma função em *plan* que em *root task*. As tarefas que se encontram dentro do plano estão condicionadas pelo *goal* do plano. No caso do Exemplo 2, no final da execução das tarefas A4, A5, A6 e E2 do plano P6, o motor de execução deve verificar se o *goal* se verifica, ou seja, se o paciente já não demonstra sintomas de alergia. Caso o *goal* se verifique, o plano será dado como concluído e executar-se-á a tarefa seguinte, caso contrário, as tarefas do plano serão novamente executadas de forma a atingir o objectivo.

5.2.3. *Decision*

A tarefa *decision* é crucial no processo de tomada de decisão. As *decisions* são definidas em termos de regras que levam à selecção de opções.

A tarefa possui alguns dos atributos já abordados como: *name*, *caption*, *description* e *trigger condition*. A Tabela 5 apresenta os atributos característicos de uma *decision*.

Tabela 5 – Atributos de uma *decision* e respectivos exemplos.

Atributo	Obrigatoriedade	Exemplo
Name	Sim	D2
Caption	Sim	Determine if the patient has viral respiratory infection (VRI)
Description	Sim	Based on the presence of rhinorrhea, cough, fever and sore throat, assess if

		the patient has VRI.
Trigger condition	Não	---
Candidate	Sim	vri(yes).
		vri(no).
Argument rule	Sim	if rhinorrhea(yes) and sore_throat(yes) and temperature(X) > temperature(37.5) and cough(3) then vri(yes).
		if rhinorrhea(no) or sore_throat(no) or temperature(X) =< temperature(37.5) or cough(X) < cough(3) then vri(no).

Os atributos específicos de uma *decision* são *candidate* e *argument rule*, de carácter obrigatório.

O atributo *candidate* possui uma opção para escolha na decisão na forma de um predicado na linguagem de programação em lógica Prolog. O número de *candidates* de uma *decision* representa o número de opções disponíveis e não pode ser inferior a dois. Não há um limite superior para o número de *candidates*. No caso do exemplo da Tabela 5, existem dois *candidates*, um que é a existência de infecção respiratória viral e outro que é a inexistência de infecção respiratória viral.

Cada atributo *argument rule* possui uma regra de produção relativa a uma opção que se encontra em *candidate*. A regra de produção estabelece uma relação entre os parâmetros do estado do paciente, a partir dos quais se podem inferir conclusões. Na próxima secção deste capítulo serão abordadas as regras com maior detalhe, assim como o motor de inferência utilizado para derivar conclusões. Tal como *candidate*, é necessário que hajam pelo menos duas *argument rules*, relativas a cada um dos *candidates*.

5.2.4. *Action*

Uma *action* é uma tarefa que só pode ser executada por um agente externo à aplicação. Uma *action* pode ser utilizada para representar uma grande variedade de operações no âmbito de um *plan*, desde a administração de medicação à prescrição de receitas, pedidos de exames ou

execução de procedimentos cirúrgicos. As *actions* possuem atributos comuns às tarefas anteriores, como *name*, *caption*, *description*, *trigger condition*, *duration*, *periodicity* e *repetitions*.

Na Tabela 6 figuram os atributos característicos de uma *action*.

Tabela 6 – Atributos de uma *action* e respectivos exemplos

Atributo	Obrigatoriedade	Exemplo
Name	Sim	A10
Caption	Sim	Gargle with salt water
Description	Sim	Treat sore throat with salt water.
Trigger condition	Não	---
Method	Não	1/4 teaspoon dissolved in 8 ounces of warm water.
Confirmation	Sim	Yes
Duration	Value	2
	Unit	day
Periodicity	Value	12
	Unit	hour
Repetitions	Não	---

O atributo *method* corresponde a uma descrição textual, tão detalhada quanto possível, do método a utilizar para realizar a acção. No caso do exemplo da Tabela 5, *method* especifica como se deve misturar a solução salina. Este atributo é opcional.

Quanto a *confirmation*, é um atributo do tipo texto, que visa especificar se a acção a realizar exige a permissão do utilizador, apresentando assim dois valores possíveis: *Yes* e *No*. Este é um atributo obrigatório.

Na Tabela 6 fornece-se um exemplo sobre a utilização das restrições temporais em *actions*. Neste caso especifica-se que se deve gargarejar com água salgada de doze em doze horas, durante dois dias, para tratar a dor de garganta.

5.2.5. *Enquiry*

Enquiry é uma tarefa cujo objectivo é retornar informação. Esta informação pode ser clínica ou administrativa. Consiste num ponto de entrada de dados e possui atributos comuns a outras tarefas como *name*, *caption*, *description*, *trigger condition* e *method*.

A Tabela 7 apresenta os atributos característicos de um *enquiry*.

Tabela 7 – Atributos de um *enquiry* e respectivos exemplos

Atributo	Obrigatoriedade	Exemplo
Name	Sim	E3
Caption	Sim	What is the body temperature of the patient?
Description	Sim	Obtain the body temperature (°C) of the patient.
Trigger condition	Não	---
Method	Não	Measure the axial temperature of the patient.
Enquiry name	Sim	Temperature
Data definition	Sim	temperature(X).

O atributo *method*, no contexto de um *enquiry*, indica o método pelo qual deve ser obtida a informação. No exemplo da Tabela 7, a temperatura corporal do paciente deve obter-se através da medição da temperatura axial.

O atributo *enquiry name* é do tipo texto, e consiste na identificação do parâmetro que se pretende obter. Este atributo é obrigatório.

Por fim, *data definition* permite definir como o parâmetro será introduzido na base de conhecimento, na forma de um predicado em linguagem de programação em lógica Prolog. *Data definition* também é um atributo obrigatório.

5.3. Nível Computacional

Neste tópico aborda-se o método adoptado para a representação do conhecimento, a implementação da Qol no modelo proposto e o motor de inferência desenvolvido.

5.3.1. Representação do Conhecimento

Na representação do conhecimento, nomeadamente dos parâmetros de saúde que constituem o estado do paciente, utiliza-se a EPL (Analide, Novais, Machado, & Neves, 2006; Neves, Machado, Analide, Abelha, & Brito, 2007) em linguagem Prolog (Colmerauer, Kanoui, Pasero, & Roussel, 1972), tal como foi definida no capítulo quatro, nos Programas 1,2,3 e 4. Como já foi referido, a EPL engloba dois tipos de negação: a negação forte (representada pelo símbolo \neg) e a negação por falha na prova (representada por *not*). A negação por falha na prova assume que *not p* é verdadeiro quando não existe prova da existência de *p*, ao passo que a negação forte permite a representação explícita de informação negativa, o que é útil quando a informação disponível do predicado *p* é incompleta e a sua ausência não significa a sua falsidade.

A representação de um parâmetro de saúde consiste num predicado alusivo ao parâmetro, com um atributo que representa o seu valor. Considerando os parâmetros que constituem as regras na tarefa de decisão D2, na Tabela 5, os predicados para os sintomas são os seguintes:

```
rhinorrhoea: valor
sore_throat: valor
temperature: valor
cough: valor
```

Introduz-se uma pequena alteração às definições das extensões dos predicados em EPL, que é a representação de conhecimento positivo acerca de um predicado, através do predicado *fact*. Esta alteração permite uma maior facilidade na manipulação de predicados no motor de inferência. Para ilustrar esta alteração, considere-se uma situação em que há certeza da presença de rinorreia. A extensão do predicado *rhinorrhoea* é dada pelo Programa 5.

```
-rhinorrhea(X) ← not fact(rhinorrhea(X)), not  
abducible(rhinorrhea(X)).  
fact(rhinorrhea(yes)).
```

Programa 5 -- Extensão do predicado `rhinorrhea`.

A representação de um valor nulo do tipo desconhecido também sofrerá alterações. Considere-se o caso em que se desconhece qualquer informação sobre se o paciente tem a garganta inflamada, ou não. A extensão do predicado `sore_throat` é dada pelo Programa 6.

```
-sore_throat(X) ← not fact(sore_throat(X)), not  
abducible(sore_throat(X)).  
abducible(sore_throat(X)) ← fact(sore_throat(+)).  
fact(sore_throat(+)).
```

Programa 6 -- Extensão do predicado `sore_throat` com representação de um valor nulo do tipo desconhecido.

No caso do parâmetro temperatura, considere-se que temperatura do paciente é desconhecida, mas está no conjunto de valores {36.5,37,37.6}, representando um valor nulo do tipo desconhecido de um conjunto de valores. Já no caso do sintoma tosse, considere-se a classificação em fraca (valor 1), moderada (valor 2) e forte (valor 3). O valor de tosse é desconhecido, mas encontra-se no conjunto de valores {2,3}. O predicado `cough` permite `cough(2)` ou `cough(3)`, mas não ambos. As extensões dos predicados `temperature` e `cough` são dadas pelos Programas 7 e 8 respectivamente.

```
-temperature(X) ← not fact(temperature(X)), not  
abducible(temperature(X)).  
abducible(temperature(36.5)).  
abducible(temperature(37)).  
abducible(temperature(37.6)).
```

Programa 7 -- Extensão do predicado `temperature` com representação de um valor nulo do tipo desconhecido de um conjunto de valores.

```
-cough(X) ← not fact(cough(X)), not abducible(cough(X)).  
Abducible(cough(2)).  
Abducible(cough(3)).  
? (abducible(temperature(X1)) ∨ abducible(temperature(X2))) ∧  
¬(abducible(temperature(X1)) ∧ abducible(temperature(X2))).
```

Programa 8 -- Extensão do predicado `cough` com representação de um valor nulo do tipo desconhecido de um conjunto de valores, com invariante.

A inferência de conclusões, a partir dos parâmetros do estado do paciente, obriga à representação de teoremas que a tornem possível. Neste caso, os teoremas encontram-se na forma de regras de produção, que se podem visualizar em *argument rules*, na Tabela 5. Estas regras apresentam a seguinte estrutura:

```
if condition then conclusion.
```

As condições designam-se por antecedentes e a conclusão designa-se por consequente. As regras de produção (Bratko, 2001) são uma forma natural de representar o conhecimento e possuem algumas vantagens, tais como:

- Modularidade: cada regra define uma porção reduzida e independente de conhecimento;
- Incrementabilidade: podem ser adicionadas novas regras à base de conhecimento sem afectar as regras existentes;
- Modificabilidade: é possível alterar regras sem afectar as restantes;
- Transparência: dada a sua proximidade à linguagem natural, o conteúdo das regras é perceptível pelas pessoas.

Contudo, é necessário aumentar a expressividade das regras de produção. Para o efeito, definiram-se os operadores:

- "*and*": que expressa a conjunção de condições;
- "*or*": que expressa a disjunção de condições;
- "<", ">", "=" e ">=": que permitem a comparação de valores numéricos dos parâmetros do estado do paciente.

Com auxílio destes operadores e com base no exemplo da Tabela 5 define-se o conjunto de regras que compõem a decisão D2 no Programa 9.

```
if rhinorrhea(yes) and sore_throat(yes) and temperature(X) >
temperature(37.5) and cough(3) then vri(yes).
```

```
if rhinorrhea(no) or sore_throat(no) or temperature(X) =<
temperature(37.5) or cough(X) < cough(3) then vri(no).
```

Programa 9 – Regras de produção para diagnóstico de infecção respiratória viral.

O Programa 9 tem como objectivo o diagnóstico de infecção respiratória viral e os antecedentes de ambas as regras são relativos aos parâmetros rinorreia, garganta inflamada, temperatura corporal e tosse. A primeira regra declara que o paciente possui uma infecção viral respiratória se apresentar os sintomas febre, garganta inflamada, temperatura superior a 37.5°C e tosse no valor 3. Estas condições têm de estar todas reunidas para um diagnóstico positivo de infecção viral respiratória. Por outro lado, a segunda regra declara que se o paciente não apresentar pelo menos uma das condições acima mencionadas, não possui infecção viral respiratória.

5.3.2. Qualidade da Informação e Pontuação de uma Conclusão

As Qols das condições das regras de produção são calculadas de acordo com o método abordado no capítulo anterior. Após a obtenção da Qols das condições, calcula-se a pontuação da conclusão. Esta pontuação de uma conclusão corresponde a uma adaptação da noção de pontuação da Qol. Em lugar de calcular a pontuação $V_i(x)$ para um predicado, através da importância relativa w_i^k que cada um dos seus atributos possui, transpõe-se esta ideia para as conclusões das regras de produção. Deste modo, cada conclusão terá uma pontuação, que corresponde à soma da Qols dos antecedentes (condições) que levam à verificação da conclusão. O peso relativo de uma condição, c , numa conclusão d , é representado pelo símbolo w_d^c . Estes pesos encontram-se normalizados, de forma a:

$$\sum_{1 \leq c \leq n} w_d^c = 1, \forall d \quad (13)$$

em que n representa o número de antecedentes de uma conclusão. Define-se assim uma função de pontuação de conclusões na forma:

$$V_d(x) = (\sum_{1 \leq c \leq n} w_d^c \times QoI_c(x)) / n \quad (14)$$

Em que $V_d(x)$ representa a pontuação da conclusão d . Neste esquema de cálculo cada condição apresenta um peso igual numa conclusão.

Este esquema de cálculo vai de encontro ao que acontece em algumas guias, que apresentam tabelas de pontuação para os parâmetros de saúde que avaliam.

5.3.3. Motor de Inferência para a Tomada de Decisão

A tarefa de decisão é a mais importante no contexto de um processo clínico e, segundo a teoria da decisão, corresponde à actualização de uma opinião com base em Informação Imperfeita, na forma de evidências clínicas. A decisão representa a escolha entre duas ou mais opções com base no estado do paciente. No modelo proposto, uma decisão é representada pela tarefa *decision*.

A aplicação do meta-predica *demo*, representado no Programa 4, para determinar o valor de verdade dos predicados *sore_throat*, *temperature* e *cough* daria a resposta *unknown* (desconhecido) para os três. Isto acontece devido à Informação Imperfeita que caracteriza a extensão destes predicados. No entanto, não se pode dizer que as conclusões que se apoiam nestes predicados são falsas, uma vez que eles podem tomar valores que correspondem às condições necessárias às conclusões.

Torna-se necessário um motor de inferência que analise as regras de produção do corpo de conhecimento, que verifique o tipo de conhecimento que caracteriza cada uma das condições e as distinga relativamente a este aspecto. O meta-predicado desenvolvido para este efeito denomina-se *demo_forward*. Este meta-predicado encontra-se representado no Programa 10. O *demo_forward* verifica todas as regras e procura provar cada uma das conclusões, verificando as suas condições. Conclusões que se baseiem em condições com Informação Imperfeita podem ser válidas. Caso uma conclusão seja válida, o *demo_forward* fornece uma lista com as condições e respectivas Qols, assim como a pontuação global da conclusão, que é uma noção que se abordará mais à frente.


```
demo_forward(CONCLUSION,LIST1,SCORE) ← (if CONDITION then
CONCLUSION),composition(CONDITION,CONCLUSION,LIST1),
score(LIST1,SCORE),assert(conclusion(CONCLUSION,SCORE)).
```

Programa 10 – Meta-predicado demo_forward.

O componente fundamental de `demo_forward` é o predicado auxiliar `composition`, que avalia o valor de verdade de cada uma das condições. O predicado `composition` retorna a condição, a conclusão e a Qol da condição. O Programa 17, do Anexo A, contém as cláusulas do predicado `composition` que lidam com casos de condições compostas, ou seja, ligadas pelos operadores "and" ou "or". No caso de condições ligadas pelo operador "and", verifica-se o valor de verdade de cada uma das condições, adicionando-se a Qol de cada condição à lista de Qols. Por outro lado, no caso de condições ligadas pelo operador "or", verifica-se o valor de verdade de cada uma, mas só a maior Qol de entre as duas condições é adicionada à lista de Qols. As cláusulas de `composition` que lidam com os operadores "<",">","=<" e ">=" encontram-se no Programa 18 do Anexo A.

Quanto a condições simples, existem quatro casos a considerar: há informação positiva sobre a condição, a condição existe como um valor nulo do tipo desconhecido, a condição existe como um valor nulo do tipo desconhecido de um conjunto de valores e a condição existe como um valor nulo desconhecido de um conjunto de valores, com invariante.

Quando há informação positiva relativamente à condição, como é o caso do predicado `rhinorrhea` representado no Programa 5, o predicado `composition` verifica se a condição está representada no predicado `fact` e se não está representada como uma hipótese no predicado `abducible`. A cláusula que realiza a operação encontra-se no Programa 11. Uma vez realizada esta verificação, é-lhe atribuído o valor de Qol 1.

```
composition(CONDITION,CONCLUSION,LIST1) ←
fact(CONDITION),not(abducible(CONDITION)),
LIST1=[qoi(CONDITION,1)].
```

Programa 11 – Cláusula do predicado composition que lida com casos em que há informação positiva sobre as condições.

Se a condição existe como um valor nulo do tipo desconhecido, o predicado `composition` verifica se o predicado está representado com o predicado `fact` e com `abducible`. Isto só acontece em casos de valores nulos do tipo desconhecido como o do Programa 6. Neste caso, utiliza-se a Equação 7 para o cálculo da QoI, e como N é um valor muito superior a 0, o valor de QoI a atribuir é 0. A cláusula de `composition` que lida com este caso encontra-se representada no Programa 12.

```
composition(CONDITION,CONCLUSION,LIST1) ←
current_predicate(A,CONDITION),current_predicate(A,X),
current_predicate(A,Y),fact(X),abducible(Y),
LIST1=[qoi(CONDITION,0)].
```

Programa 12 – Cláusula do predicado `composition` que lida com casos em que a condição é um valor nulo do tipo desconhecido.

Relativamente aos valores nulos do tipo desconhecido de um conjunto de valores, com e sem invariante, o predicado `composition` procede de forma semelhante para os dois casos, excepto na verificação dos invariantes e no método de cálculo da QoI. As cláusulas de `composition` que lidam com estes casos encontram-se no Programa 13. Recorrendo aos exemplos dos Programas 7 e 8, as Equações a utilizar para calcular as QoIs são a 9 e 8 respectivamente. Deste modo, o valor de QoI para `temperature`, arredondado às centésimas, será aproximadamente 0.14 e o valor de QoI para `cough` será 0.5.

```
composition(CONDITION,CONCLUSION,LIST1) ←
abducible(CONDITION),current_predicate(A,CONDITION),
current_predicate(A,X),not(fact(X)),current_predicate(A,TERM),
findall(TERM,(+TERM::INVARIANT),Z),length(Z,N),N=0,
qoi_calculation_2(V,A),LIST1=[qoi(CONDITION,V)].
```

```
composition(CONDITION,CONCLUSION,LIST1) ←  
abducible(CONDITION),current_predicate(A,CONDITION),  
current_predicate(A,X),not(fact(X)),current_predicate(A,TERM),  
findall(TERM,(+TERM::INVARIANT),Z),length(Z,N),N=1,  
qoi_calculation_1(V,A),LIST1=[qoi(CONDITION,V)].
```

Programa 13 – Cláusulas do predicado `composition` que lidam com valores nulos do tipo desconhecido de um conjunto de valores. A primeira cláusula refere-se a casos sem invariante e a segunda cláusula refere-se a casos com invariante.

Obtidas as Qols das condições, o Programa 10 recorre ao predicado auxiliar `score`, que se encontra representado no Programa 19 do Anexo A, para o cálculo da pontuação da conclusão, tal como é definido na Equação 14. Em seguida, o predicado `conclusions`, também representado no Anexo A, pelo Programa 20, permite ordenar as conclusões por ordem decrescente de pontuação, fornecendo uma ordem de preferência para as diferentes conclusões ao utilizador. A conclusão seleccionada é a que se encontra em primeiro lugar.

Este motor de inferência é utilizado em cada tarefa *decision*. No próximo capítulo fornece-se um exemplo na forma de caso de estudo de uma tarefa de *decision*, em que se aplica este motor de inferência.

É possível utilizar o predicado `composition` para a verificação de *goals* em planos que os tenham definido. Fornecendo uma medida da Qol do *goal* ao utilizador. Em última análise cabe sempre ao utilizador a escolha se o plano deve ser ou não executado novamente. Os mecanismos previstos no modelo apenas sugerem a repetição do plano caso o *goal* não se verifique.

5.3.4. Partilha e Validação de Guias Clínicas

Para haver partilha e disseminação de CIGs é necessário que estas apresentem um formato que suporte às guias e um mecanismo de descrição da sintaxe do modelo de representação. Nesta área, o formato XML (W3C, 2011) é a escolha preferencial para a representação dos modelos das principais abordagens de CIGs, como Arden Syntax (Kim, Haug, Rocha, & Choi, 2008), GLIF3 (Peleg, et al., 2000) e Asbru (Votruba, Miksch, & Kosara, 2004).

Para a representação do modelo proposto, propõe-se também o formato XML. XML consiste num formato flexível e simples, baseado em etiquetas, que para além de ser

interpretável por máquinas, é perceptível por pessoas. Este formato é genérico, o que permite descrever um número variado de estruturas diferentes. A flexibilidade do formato provém do facto de ser possível definir um elevado número de etiquetas diferentes. XML permite a descrição de relações em bases de dados relacionais.

Uma das maiores vantagens do formato XML é a existência de uma linguagem para definição de regras de validação em documentos no formato XML, XML Schema Definition (XSD) (W3C, 2011). XSD baseia-se na descrição de elementos e tipos de dados. Deste modo, é possível descrever as diferentes classes de tarefas como elementos e os respectivos atributos. No Anexo B encontra-se o Programa 21, que corresponde a uma proposta de XML *schema* que descreve a sintaxe do modelo proposto.

6. Caso de Estudo

6.1. Introdução

Após a apresentação do modelo no capítulo anterior, procede-se agora ao desenvolvimento de um pequeno caso de estudo, no qual o modelo é utilizado para representar uma guia utilizada na prática clínica.

Optou-se pela representação de uma guia para detecção e tratamento de níveis elevados de colesterol. A guia designa-se por *Detection and Treatment of High Blood Cholesterol in Adults* (NHLBI, 2002), e foi desenvolvida pelo National Heart Lung and Blood Institute (NHLBI), nos Estados Unidos. Os níveis elevados de colesterol são um problema grave nos países ocidentais desenvolvidos. Em Portugal, cerca de 56% da população possui colesterol acima do nível recomendado (cerca de 100 mg/dL) (OMS, 2008), e estão associados a uma variedade de doenças cardíacas, nomeadamente à doença cardíaca coronária (Antonopoulos, 2002). A doença cardíaca coronária é uma das principais causas de morte prematura nos países desenvolvidos.

De seguida fornece-se uma explicação sobre o que é o colesterol e a sua relação com a doença cardíaca coronária. Também se realiza uma descrição breve da guia a representar e da guia no modelo proposto. Por fim, analisa-se a expressividade e funcionalidade da representação.

6.2. Colesterol e Doença Cardíaca Coronária

O colesterol é um lípido (gordura) presente nas membranas celulares e é um precursor dos ácidos biliares e das hormonas esteróides. O colesterol viaja na corrente sanguínea na forma de partículas constituídas por lípidos e proteínas, designadas por lipoproteínas. Existem três classes principais de lipoproteínas transportadoras (Antonopoulos, 2002): Low Density Lipoprotein (LDL), High Density Lipoprotein (HDL) e Very Low Density Lipoprotein (VLDL). A LDL é responsável pelo transporte de 70% do colesterol que circula no organismo. A LDL, quando ligada ao colesterol, adquire a designação de LDL-Cholesterol (LDL-C), que pela sua correlação com o risco de doença cardíaca coronária, é designada por "mau colesterol".

A LDL-C é a principal responsável pela aterosclerose (Graham, James, & Cowan, 2000). Esta é uma doença inflamatória crónica na qual ocorre formação de ateromas nos vasos sanguíneos. Os ateromas são placas compostas essencialmente por lípidos e tecido fibroso, que levam à diminuição progressiva do diâmetro dos vasos sanguíneos, podendo levar à obstrução dos mesmos. A formação de ateroma ocorre quando a LDL-C penetra a parede do vaso sanguíneo, atravessa o endotélio e aloja-se na camada endotelial interna. A oxidação da LDL-C provoca lesões nas células endoteliais, o que resulta na libertação de substâncias nocivas do conteúdo celular. Como resposta, ocorre uma acumulação de fagócitos (glóbulos brancos com funções de digestão de partículas nocivas). Os produtos da fagocitose acumulam-se juntamente com tecido cicatrizado sobre os lípidos, levando à formação de placas fibrosas. Estas placas são instáveis, por acção da LDL, e podem fragmentar-se, o que leva à formação de obstruções responsáveis por trombozes, como ilustra a Figura 15. A ruptura de placas é responsável pelas síndromes cardíacas mais agudas (e.g., enfarte do miocárdio, angina de peito) e a sua ocorrência nas artérias coronárias pode causar doença cardíaca coronária, privando as células do miocárdio da recepção de sangue oxigenado, o que provoca a sua morte.

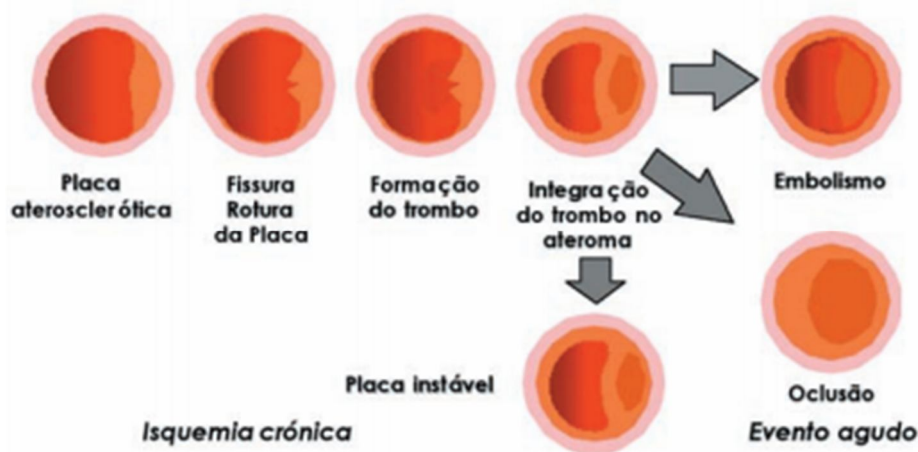


Figura 15 – Formação de um trombo a partir de uma placa instável (reproduzido de (Silva, 2004)).

Níveis elevados de LDL-C contribuem para a formação de mais placa fibrosa e para o aumento da sua instabilidade. A acção da VLDL é análoga à acção da LDL e existe em pequena quantidade, pelo que é contabilizada juntamente com a LDL. Quanto à HDL, crê-se que esta

contribua para a redução dos efeitos da LDL, embora os mecanismos pelos quais isto acontece não sejam, ainda, completamente conhecidos.

6.3. Descrição da Guia para Detecção e Tratamento de Níveis Elevados de Colesterol

A guia Detection and Treatment of High Blood Cholesterol in Adults (NHLBI, 2002) tem como principal objectivo o controlo dos níveis de lipoproteínas no organismo de um indivíduo.

A guia está representada nas Figuras 34 a 39, no Anexo C, e encontra-se dividida em nove passos:

- O passo 1 da guia consiste em traçar o perfil lipoproteico do paciente. Este passo deve demorar entre nove a doze horas. Neste passo obtém-se: o nível de LDL-C (mg/dL), o nível total de colesterol (mg/dL) (corresponde a soma dos níveis das três classes de lipoproteínas) e o nível de HDL Cholesterol (HDL-C) (mg/dL). Este passo também informa sobre os níveis óptimos de cada um dos parâmetros;
- No passo 2 determina-se se o paciente apresenta risco de sofrer de doença cardíaca coronária. Para o efeito, verifica-se se o paciente já sofre de doença cardíaca coronária ou se possui aterosclerose;
- No passo 3 verifica-se se o paciente possui factores de risco, para além de LDL elevada. Os factores de risco avaliados são tabagismo, hipertensão, baixa HDL, historial de doença cardíaca coronária na família e a idade;
- O passo 4 consiste numa avaliação sobre se é necessário estimar o risco a dez anos de doença cardíaca coronária. Se o paciente não apresentar risco de doença cardíaca coronária e apresentar um número de factores de risco superior ou igual a dois, é necessário determinar o risco a dez anos de doença cardíaca coronária. O valor do risco a dez anos calcula-se através das tabelas de Framingham que se encontram na Figura 39, do Anexo C;
- No passo 5, com base no risco de doença cardíaca coronária, no risco a dez anos de doença cardíaca coronária e no número de factores de risco, define-se o nível de LDL que se pretende alcançar, a necessidade de alterar os hábitos de vida e a necessidade de realizar tratamento com fármacos;

- O passo 6 contém os procedimentos para alteração dos hábitos de vida do paciente. Neste passo especifica-se recomendações de alimentação e exercício físico, entre outras;
- Por sua vez, o passo 7 possui o conjunto de fármacos que é possível receitar ao paciente, caso seja necessário. Estes fármacos alteram o metabolismo das lipoproteínas;
- O passo 8 especifica os sintomas para a detecção de síndrome metabólica, após três meses de alteração dos hábitos de vida, e as recomendações para o tratamento desta condição de saúde;
- Por fim, o passo 9 possui as opções de tratamento para níveis elevados de triglicédeos e níveis baixos de HDL. Neste passo também se definem níveis de não-HDL a alcançar. O nível não-HDL corresponde à subtração do nível de HDL ao nível total de colesterol.

6.4. Representação da Guia Clínica no Modelo Proposto

A representação da guia Detection and Treatment of High Blood Cholesterol in Adults, no modelo proposto, obrigou a algumas adaptações que serão convenientemente explicadas. Embora a guia esteja estruturada em passos, existem alguns aspectos que se optou por simplificar, sem, no entanto, desvirtuar o seu conteúdo.

A representação será ilustrada através de figuras, que representam o fluxograma da guia, juntamente com textos explicativos. Começa-se pela *root task*, em que se insere a guia, e avança-se para os *plans* e o seu conteúdo. Esta representação possui um total de dezassete *plans*.

6.4.1. *Root Task* RT1

Segundo o modelo proposto, a guia para detecção e tratamento de níveis elevados de colesterol é representada como o *plan* P1, inserido na *root task* RT1, que contém guias para tratamento. RT1 e P1 encontram-se representados na Figura 16.

Nos atributos do *plan* P1 especifica-se o nome da guia, a sua descrição, o seu autor, a data de criação e a sua versão actual. O *plan* P1 é composto pelos *plans* P2, P3, P4, P5 e P6, pelas *decisions* D1 e D2 e pela *action* A1.

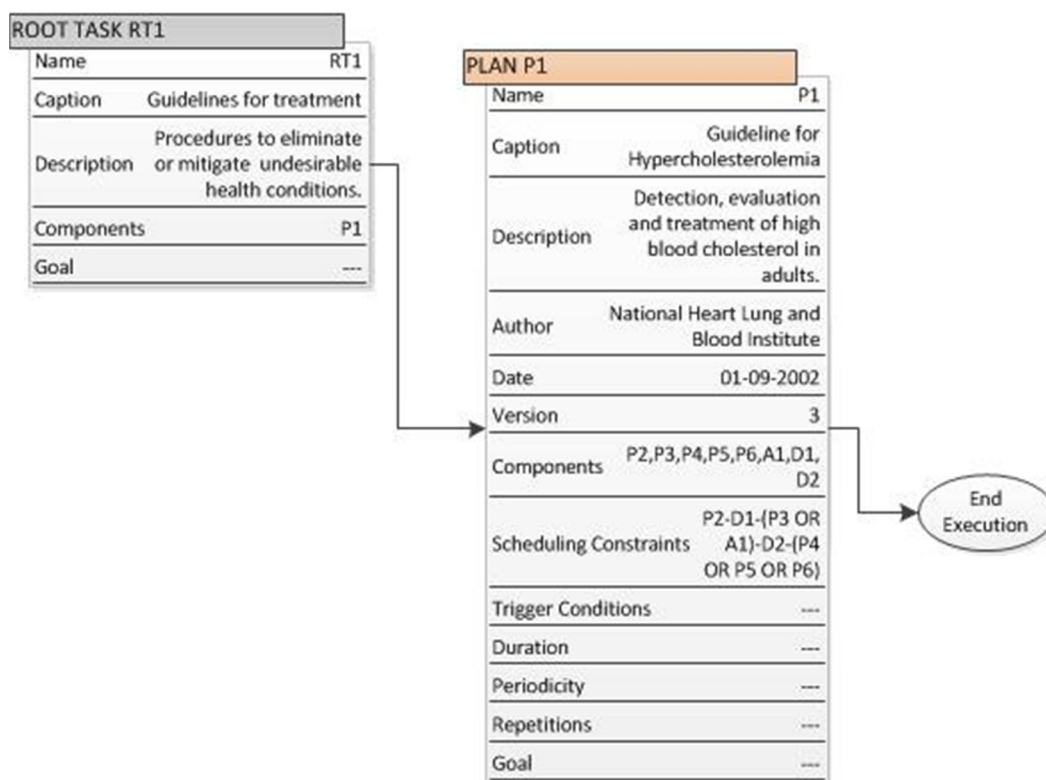


Figura 16 – Representação da *root task* RT1 e da guia como *plan* P1.

6.4.2. *Plan*P1

A representação de P1 até D2 encontra-se na Figura 17. Inicia-se com o *plan* P2, que corresponde a um conjunto de *enquiries* que visa obter informação do paciente. Os parâmetros obtidos neste plano correspondem ao perfil lipoproteico do paciente (nível de LDL-C, nível de HDL-C e nível total de colesterol), à identificação de doença cardíaca coronária, ou risco de doença cardíaca coronária, e aos factores de risco que condicionam o estabelecimento de uma meta para o nível de LDL-C. Este plano possui uma duração de 12 horas, pelo que a execução só seguirá para D1 após este período de tempo. P1 corresponde a uma junção dos passos 1, 2 e 3 da guia, visto que estes passos consistem em entradas de dados.

De seguida surge a *decision* D1. Este momento de decisão no processo clínico corresponde ao passo 4 da guia. D1 define que, se o paciente apresentar dois ou mais factores de risco que alteram o nível de LDL-C a alcançar e não apresentar risco de doença cardíaca coronária, deve-se estimar o risco a dez anos de doença cardíaca coronária. Isto significa que o

processo clínico será direccionado para P3, que apresenta a *trigger condition* condizente. Caso não seja necessária a avaliação do risco a dez anos, o processo clínico será direccionado para A1, que é uma recomendação para prosseguir para a determinação do nível de LDL-C a alcançar.

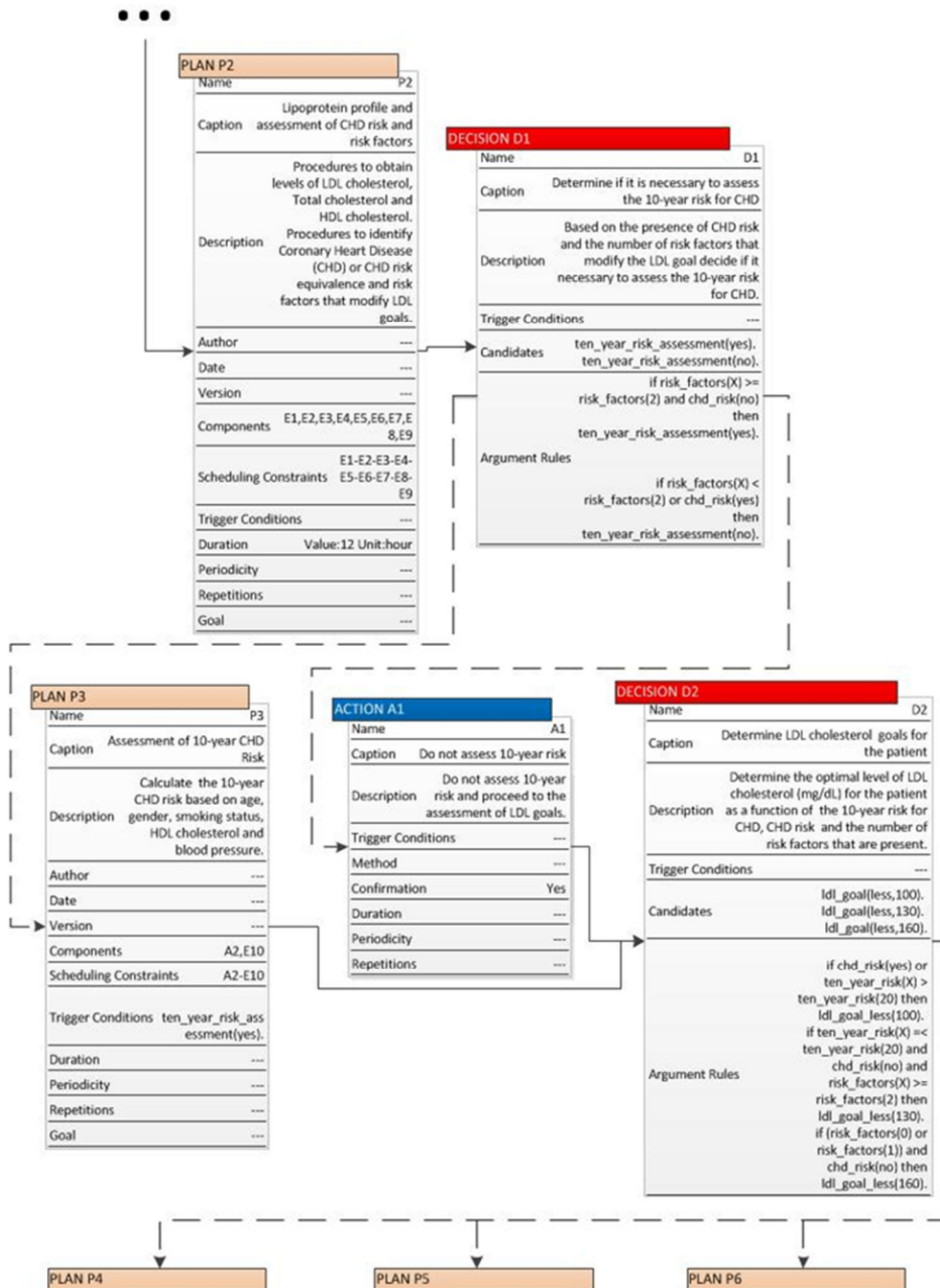


Figura 17 – Representação do *plan* P1 (1ª parte).

Na *decision* D2, com base nos parâmetros do estado do paciente, especifica-se as metas para os níveis de LDL-C. Existem três situações possíveis:

- O nível de LDL-C deve ser inferior a 100 mg/dL e o processo clínico é reencaminhado para P4;
- O nível de LDL-C deve ser inferior a 130 mg/dL e o processo clínico é reencaminhado para P5;
- O nível de LDL-C deve ser inferior a 160 mg/dL e o processo clínico é reencaminhado para P6.

A Figura 18 ilustra a porção de P1 em que estão representados P4, P5,P6.

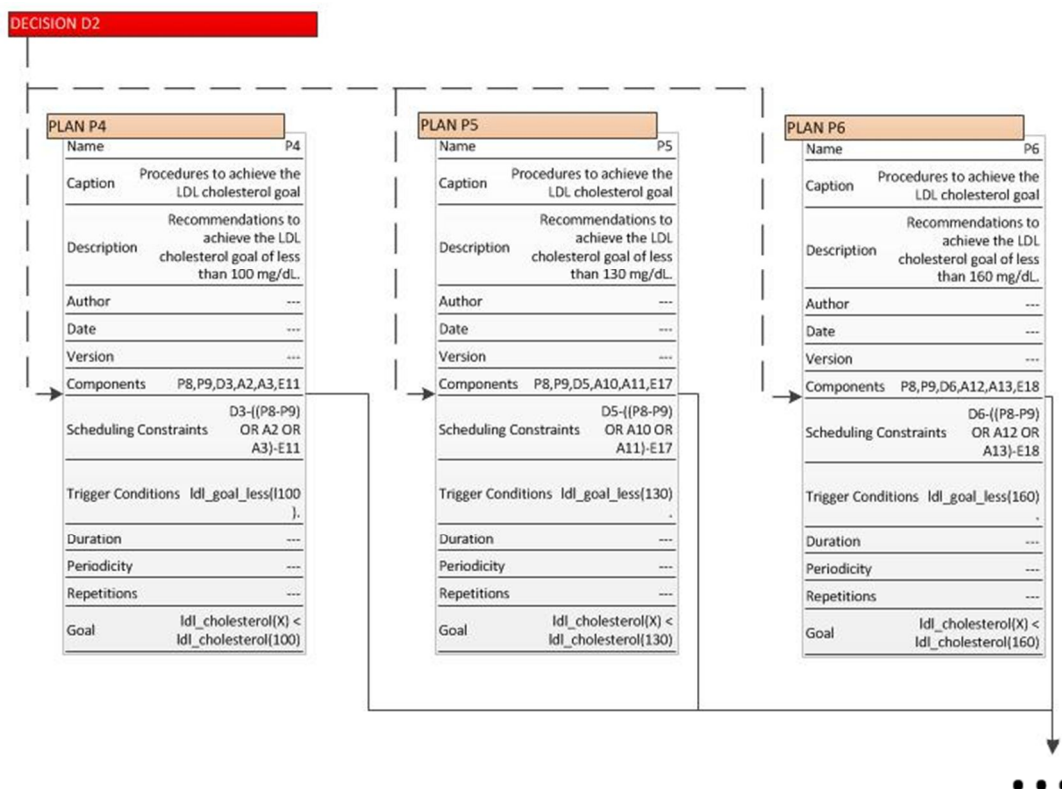


Figura 18 – Representação do *plan* P1 com os planos P4, P5 e P6 (2ª parte).

Os plans P4, P5 e P6 encontram-se condicionados por *goals*, que especificam os níveis de LDL a obter. O plano só terminará de executar se o nível de LDL do *goal* for atingido. D2, juntamente com estes três planos, correspondem a uma condensação dos passos 5, 6, 7 e 8 da guia. O passo 9 encontra-se representado no *plan* P7, que se irá abordar mais à frente.

6.4.3. *PlanP2*

O *plan P2* encontra-se representado na Figura 19. Trata-se de um plano constituído apenas por *enquiries*, em que se obtêm diferentes parâmetros do estado do paciente.

Os *enquiries* E1 a E3 realizam questões sobre o perfil lipoproteico do paciente. Através destes *enquiries* obtêm-se o nível de LDL-C, o nível total de colesterol e o nível de HDL-C. A definição do predicado que representa estes parâmetros encontra-se nos atributos *data definition* dos respectivos *enquiries*.

O *enquiry* E4 corresponde a uma questão sobre o risco de doença cardíaca coronária. Caso o paciente já possua doença cardíaca ou alguma das condições de saúde relacionadas com a aterosclerose, que são especificadas, considera-se que possui risco de doença cardíaca coronária.

No caso dos *enquiries* E5 a E8, correspondem à avaliação de factores de risco de doença cardíaca coronária. Os parâmetros avaliados são: sexo, idade, pressão sanguínea e tabagismo. A obtenção destes parâmetros é importante, pois são necessários para a utilização das tabelas de Framingham.

Por fim, o *enquiry* E9 corresponde à determinação do número de factores de risco que o paciente possui. Esta informação é relevante para averiguar se é necessário o cálculo do risco a dez anos de doença cardíaca coronária.

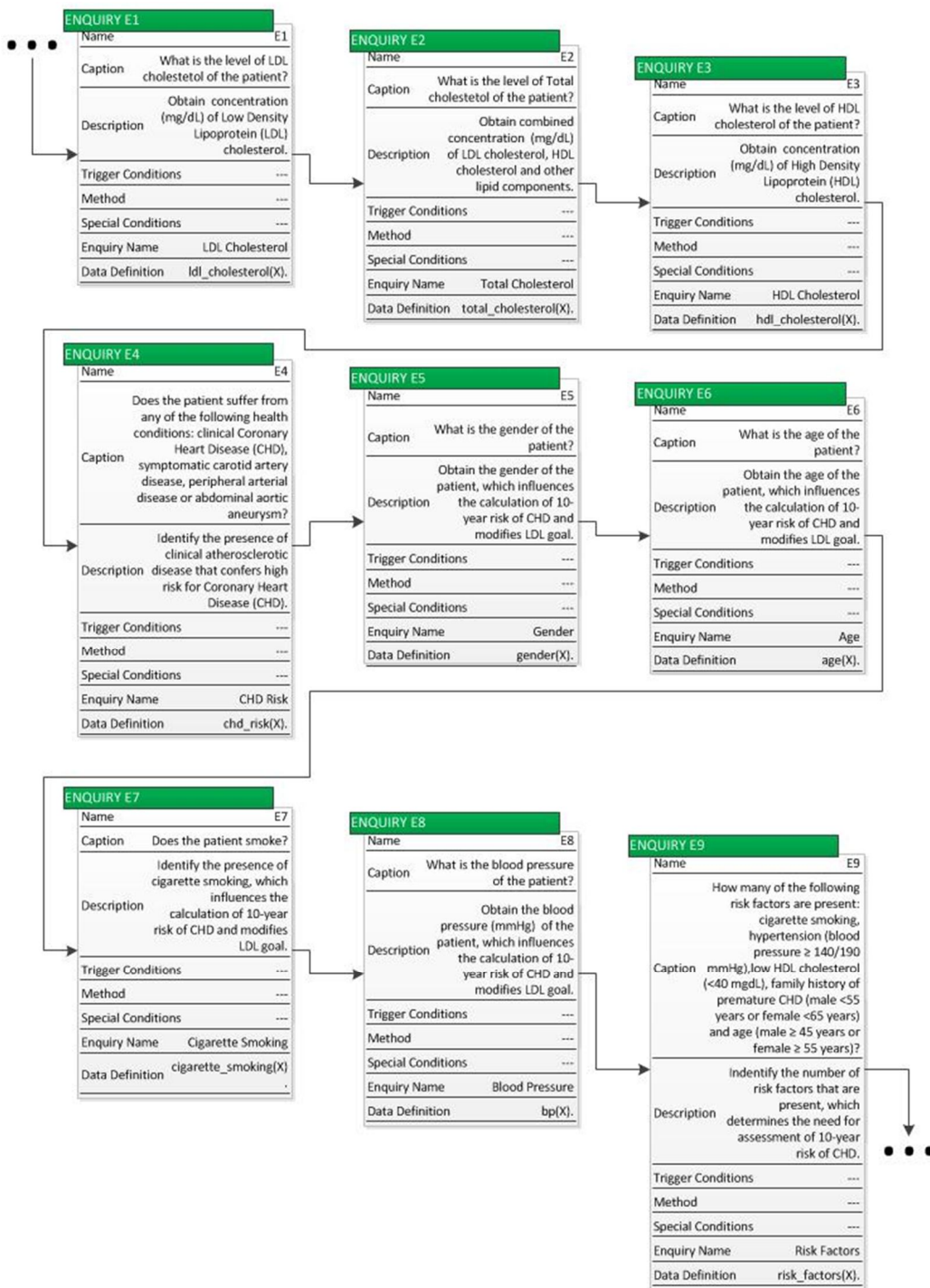


Figura 19 – Representação do *plan* P2.

6.4.4. *Plan* P3

O *plan* P3 corresponde ao conjunto de recomendações para o cálculo do risco a dez anos de doença cardíaca coronária. A Figura 20 ilustra os componentes de P3.

O primeiro componente de P3 é uma *action* que transmite o que se deve fazer para calcular o risco a dez anos. O atributo *method* especifica como o cálculo se deve realizar. O risco a dez anos de doença cardíaca coronária obtém-se através da soma das pontuações de cada factor de risco nas tabelas de Framingham. O valor obtido deve ser introduzido na tarefa seguinte, o *enquiry* E10.

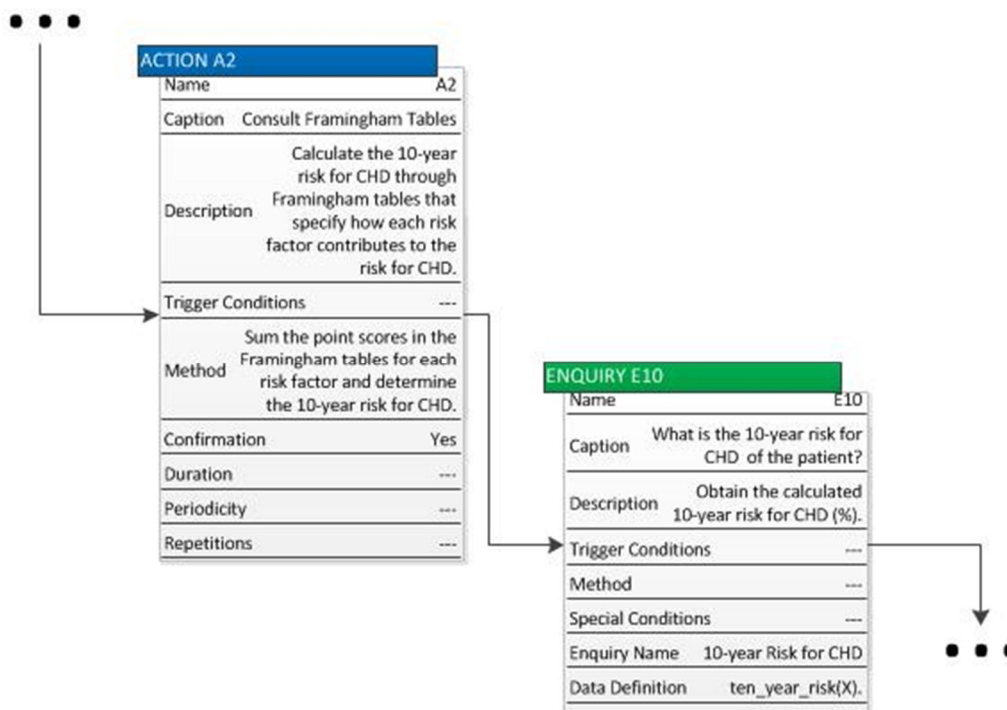


Figura 20 – Representação do *plan* P3.

6.4.5. *Plans* P4, P5 e P6

As estruturas dos *plans* P4, P5 e P6 são semelhantes, com a diferença de que estão condicionados por diferentes *goals* no que diz respeito aos níveis de LDL-C a alcançar. Deste modo, procede-se a uma explicação do *plan* P4 como referência para os outros *plans*.

O *plan* P4, que se encontra representado na Figura 21, inicia-se com a tarefa D3, que, com base nos níveis de LDL-C, determina se o paciente deve alterar os seus hábitos de vida (Therapeutic Lifestyle Changes – TLC) e se deve realizar tratamento com fármacos que alteram o metabolismo das lipoproteínas. Se o paciente apresentar níveis de LDL-C superiores a 100 mg/dL deve realizar TLC, sendo encaminhado para o *plan* P8. Se os níveis forem superiores a 130 mg/dL, o paciente também deve realizar tratamento com fármacos (*action* A2). No atributo

method de A2 especifica-se o conjunto de fármacos que podem ser utilizados. No caso de nenhum dos dois ser necessário, o paciente é encaminhado para a *action* A3.

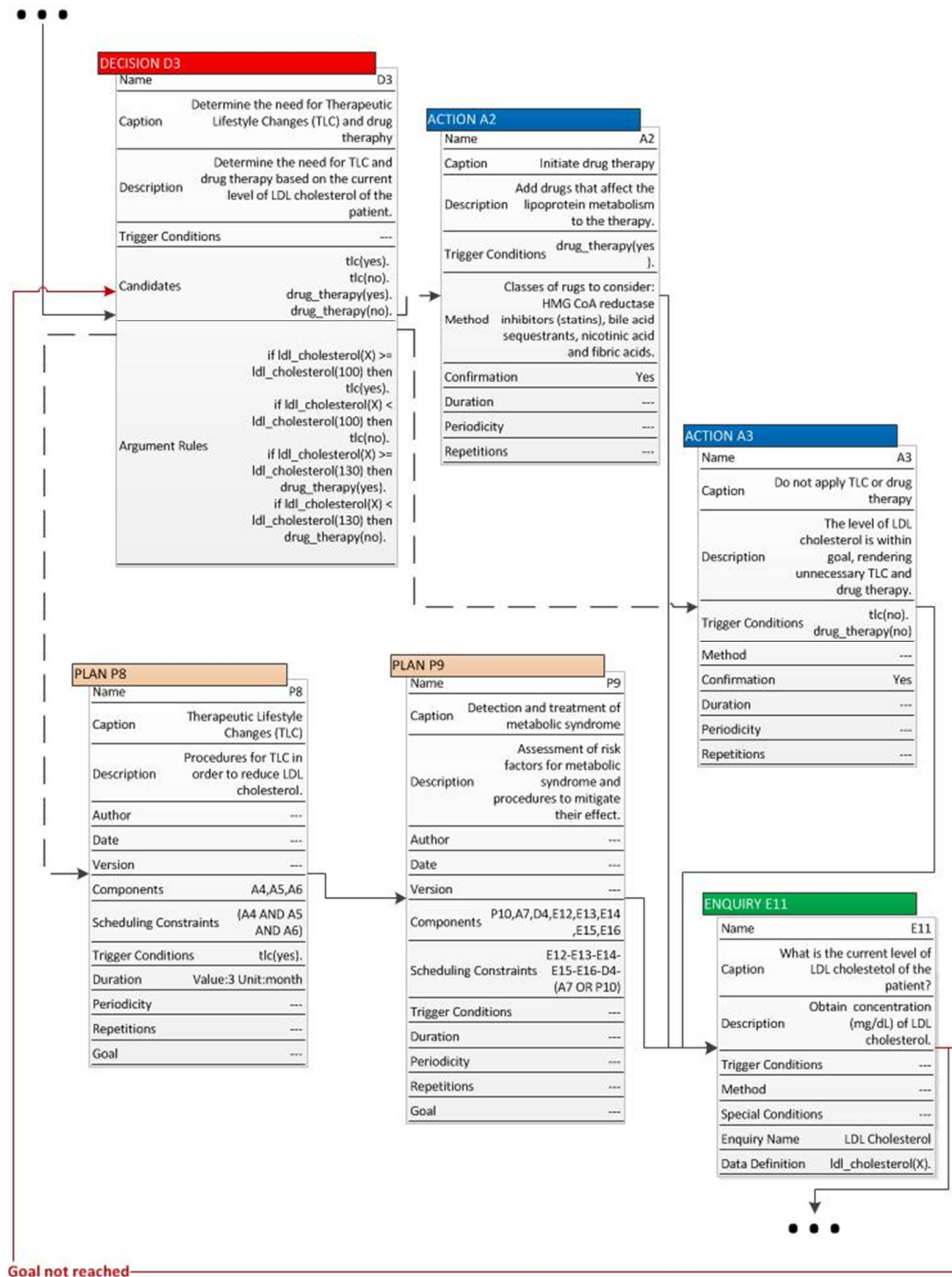


Figura 21 – Representação do *plan* P4.

O *plan* P8 apresenta uma duração de três meses. Após os três meses realiza-se uma avaliação da existência de síndrome metabólica, através do *plan* P9.

No final de P4 sincronizam-se as diferentes opções de tratamento para baixar o nível de LDL-C no *enquiry* E11. Trata-se de um ponto de entrada para o valor de LDL-C após o tratamento. Caso o *goal* para o nível de LDL-C seja alcançado, o plano será dado como terminado, caso contrário o plano será novamente executado, até produzir os resultados pretendidos.

As representações dos plans P5 e P6 podem ser vistas nas Figuras 40 e 41 do Anexo D.

6.4.6. Plan P8

O *plan* P8 contém as recomendações para alterar os hábitos de vida, no sentido de reduzir os níveis de LDL-C. Este plano encontra-se representado na Figura 22.

P8 é constituído por três *actions*: A4, A5 e A6. A4 recomenda a dieta alimentar que o paciente deve seguir. A gestão do peso e do índice de massa corporal é especificada em A5. A6 corresponde a recomendações de exercício físico. De acordo com a representação da Figura 22, estas *actions* devem ser executadas em simultâneo.

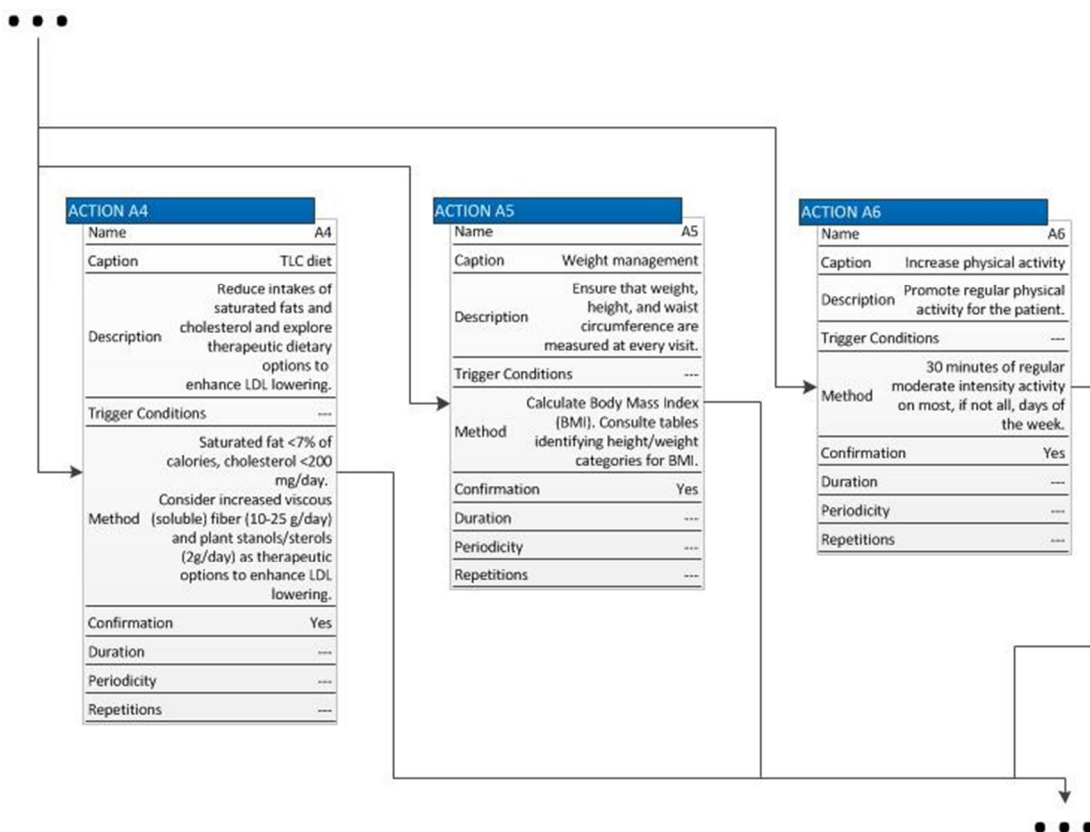


Figura 22 – Representação do *plan* P8.

6.4.7. *Plan P9*

O *plan* P9, representado na Figura 23, tem como objectivo o diagnóstico e tratamento de síndrome metabólica. A síndrome metabólica é uma condição de saúde que se caracteriza por obesidade, níveis elevados de triglicédeos, baixos níveis de HDL-C, hipertensão e elevados níveis de glicose.

Os *enquiries* E12 a E16 obtêm os parâmetros necessários para o diagnóstico de síndrome metabólica. Na *decision* D4 realiza-se esse diagnóstico e, com base no resultado, reencaminha-se o processo clínico para P10, que contém os procedimentos para tratar a síndrome metabólica, ou para A7, que afirma que o paciente não possui esta condição de saúde.

Considerou-se que um paciente possui síndrome metabólica caso apresente:

- Um perímetro abdominal superior a 102 cm no caso de ser do sexo masculino, superior a 88 cm no caso de ser do sexo feminino;
- Nível de triglicédeos superior ou igual a 150 mg/dL;
- Nível de HDL-C inferior a 40 mg/dL no caso de ser do sexo masculino, inferior a 40 dLmg/no caso de ser do sexo feminino;
- Pressão sanguínea superior ou igual a 130/85 mmHg;
- Nível de glicose superior a 110 mg/dL.

No caso de não apresentar pelo menos uma destas condições, o paciente não possui síndrome metabólica.

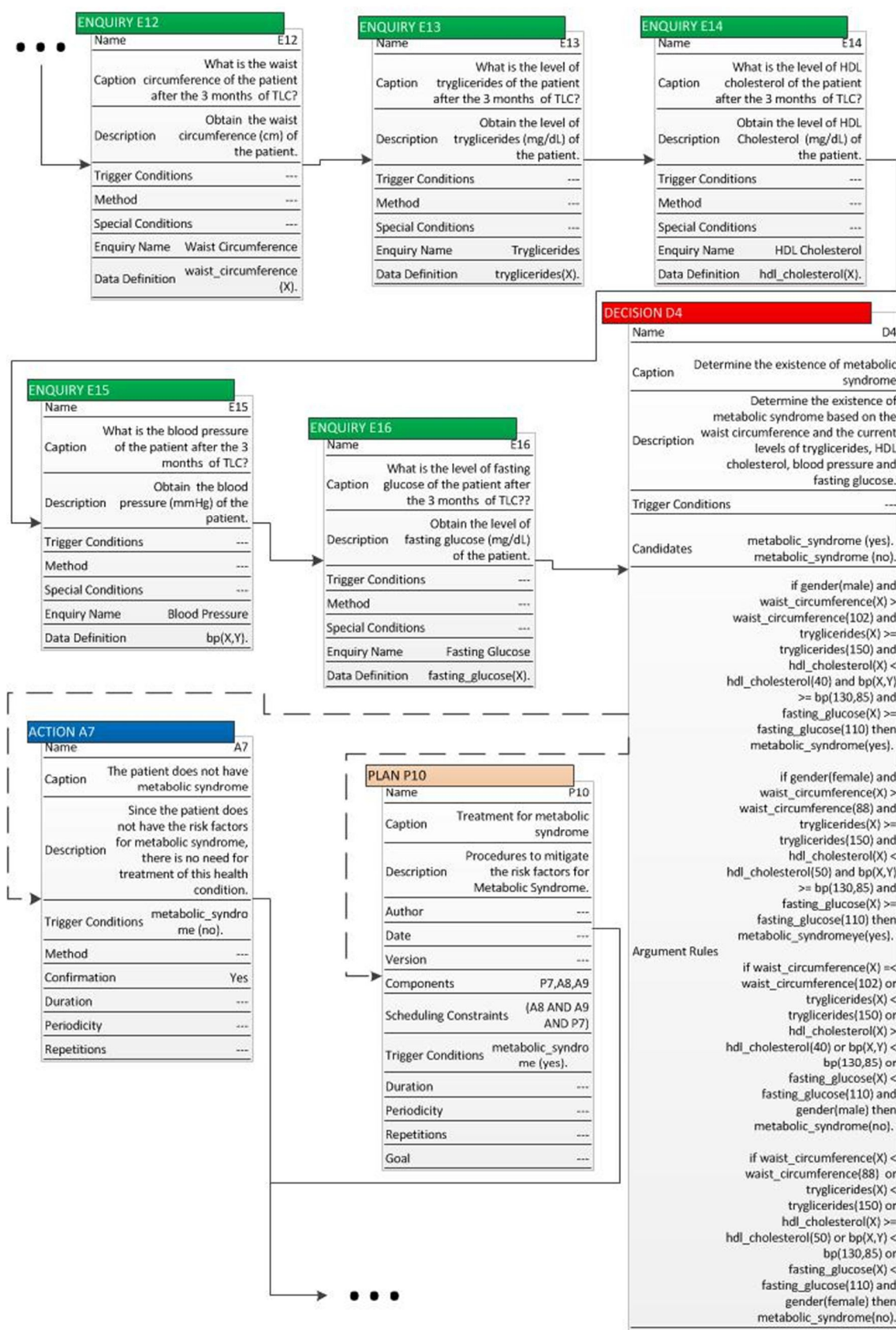


Figura 23 – Representação do plan P9.

6.4.8. Plan P10

O *plan* P10 contém os procedimentos para tratar a síndrome metabólica, na forma de duas *actions* (A8 e A9) e um *plan* (P7). O *plan* P10 encontra-se representado na Figura 24.

A8 recomenda o tratamento de obesidade através da intensificação da gestão de peso e de exercício físico. A9 sugere o tratamento de hipertensão através de aspirina. Por fim, o *plan* P7 contém os procedimentos para detecção e tratamento de níveis elevados de triglicérideos e baixos níveis de HDL-C.

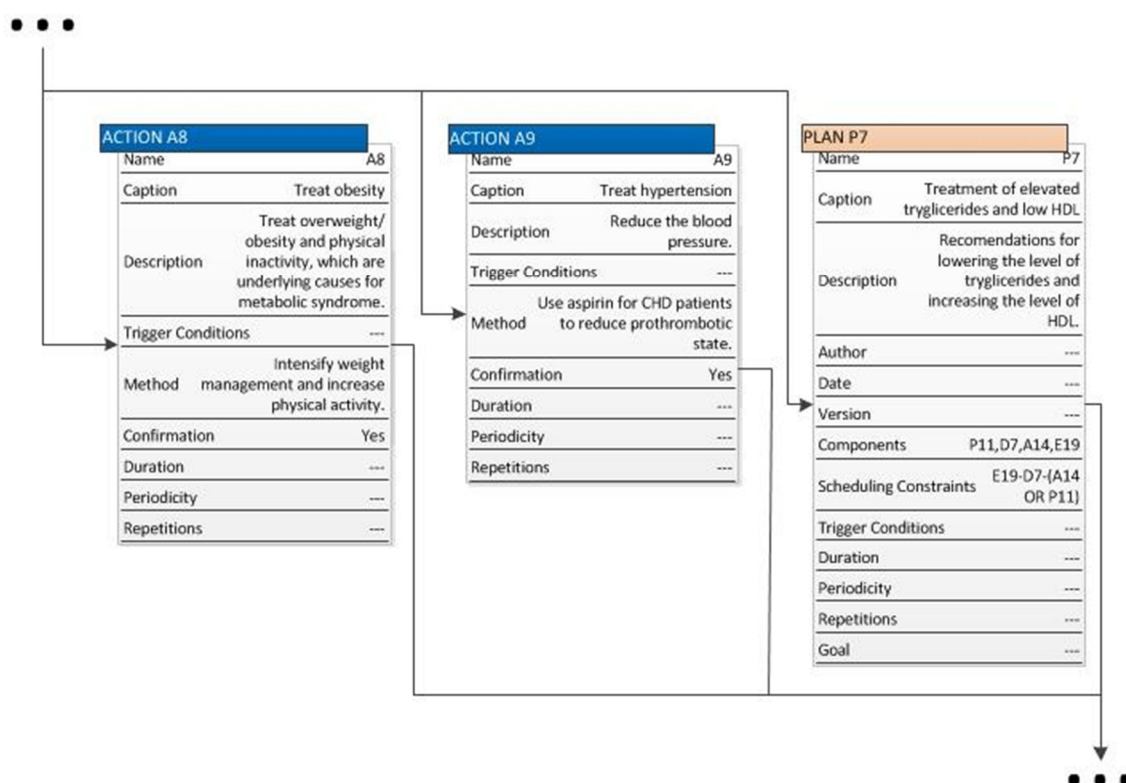


Figura 24 – Representação do *plan* P10.

6.4.9. Plan P7

O *plan* P7 corresponde uma avaliação da necessidade de tratamento de níveis elevados de triglicérideos.

Como se pode verificar na Figura 25, o plano inicia-se com o *enquiry* E19, cuja finalidade é obter o nível actual de triglicérideos do paciente. Actualiza-se o nível de triglicérideos, pois este pode ter baixado como resultado dos restantes tratamentos de síndrome metabólica.

Com base no valor obtido em E19, a *decision* D7 reencaminha o processo clínico para as diferentes opções possíveis. Se o nível de triglicérides for superior a 150 mg/dL, deve haver tratamento para este nível elevado de triglicérides. A tarefa que implementa o tratamento de níveis elevados de triglicérides é o *plan* P11. Se o nível de triglicérides for inferior a 150 mg/dL, surge a *action* A14, que sugere que o paciente não precisa de tratamento.

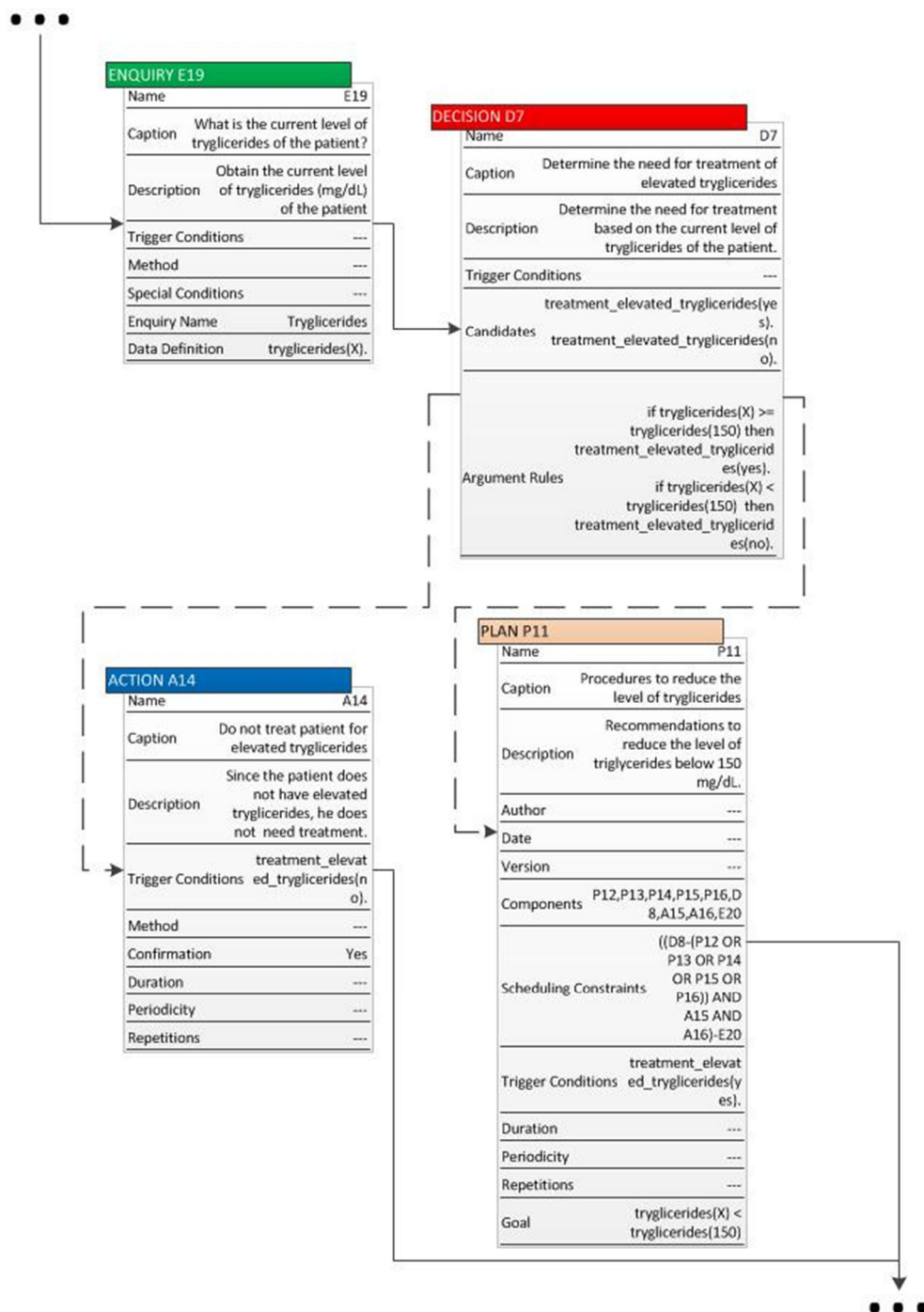


Figura 25 – Representação do *plan* P7.

6.4.10. Plan P11

O *plan* P11, cuja primeira parte se encontra representada na Figura 26, inicia-se com três tarefas simultâneas D8, A15 e A16.

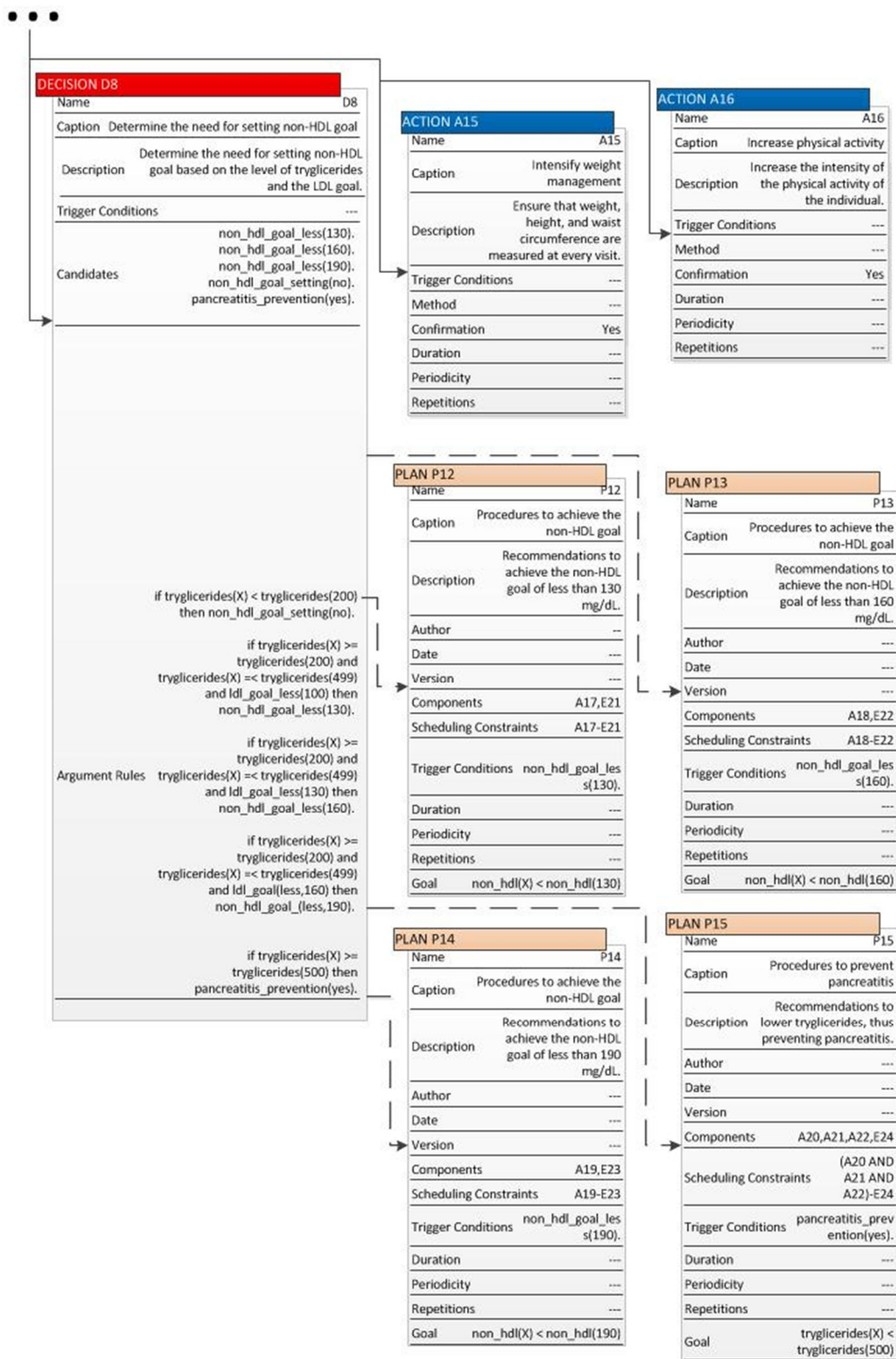


Figura 26 – Representação do *plan* P11 (1ª parte).

A15 corresponde a uma recomendação para intensificar a gestão de peso do paciente e A16 aconselha a um aumento da intensidade de exercício físico.

Já D8 é uma tarefa de decisão que determina se é necessário estabelecer um nível não-HDL-C como objectivo do tratamento. O nível não-HDL-C corresponde à subtracção do nível de HDL-C ao nível total de colesterol. A meta não-HDL-C é calculada através da adição de 30 mg/dL ao nível de LDL-C que foi estabelecido como objectivo do tratamento em tarefas anteriores. Só é necessário definir níveis não-HDL-C como objectivos, se o nível de triglicérideos for superior ou igual a 200 mg/dL e inferior ou igual a 499 mg/dL. Consoante o nível de não-HDL-C a atingir seja inferior a 130 mg/dL, 160 mg/dL ou 190 mg/dL, o processo clínico será direccionado para os planos P12, P13 e P14, respectivamente. No caso do nível de triglicérideos ser igual ou superior a 500 mg/dL deve-se proceder à prevenção de pancreatite, cujas instruções encontram-se em P15. Por outro lado, se o nível de triglicérideos for inferior a 200 mg/dL, o processo clínico é reencaminhado para o *plan* P16, representado na Figura 27, para detecção e tratamento de baixos níveis de HDL-C.

Ocorre sincronização do processo clínico no *enquiry* E20, que visa obter o nível actual de triglicérideos do paciente. Caso este nível atinja o objectivo de P11 (inferior a 150 mg/dL), o plano será dado como terminado, caso contrário será novamente executado.

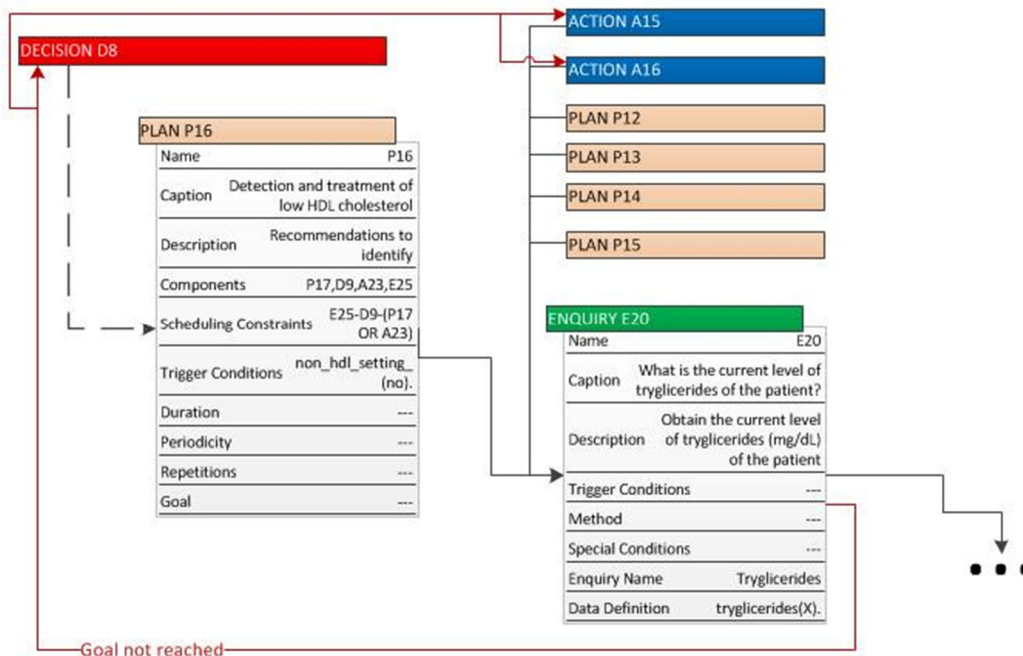


Figura 27 – Representação do *plan* P11 (2ª parte).

6.4.11. *Plans* P12, P13 e P14

Os plans P12, P13 e P14 são semelhantes, com a exceção dos níveis não-HDL-C que possuem como *goals*.

O *plan* P12 encontra-se representado na Figura 28 e é constituído pela *action* A19 e pelo *enquiry* E23. A19 corresponde ao procedimento para atingir o nível não-HDL-C que se traçou como meta. O procedimento passa pela intensificação do tratamento com fármacos que reduzem os níveis de LDL-C e VLDL-C. De seguida, no *enquiry* E23 pretende-se saber qual o nível de não-HDL-C após o tratamento. Caso o nível de não-HDL-C seja o definido no *goal* do plano, o plano será dado como terminado. Se tal não se verificar, o plano executará novamente.

Os plans 13 e 14 estão representados nas Figuras 42 e 43 do Anexo D.

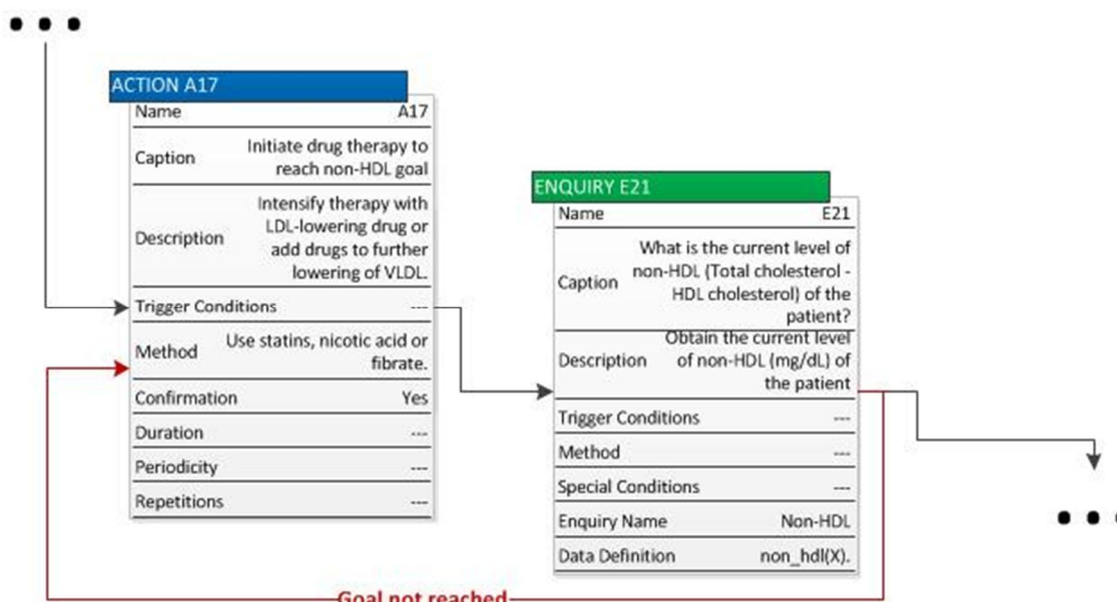


Figura 28 – Representação do *plan* P12.

6.4.12. *Plan* P15

O *plan* P15 reúne o conjunto de recomendações para prevenção da ocorrência de pancreatite, como resultado de níveis elevados de triglicéridos. A sua representação encontra-se na Figura 29.

P15 inicia-se com três *actions* simultâneas: A20, A21 e A22. A *action* A20 recomenda uma dieta alimentar com baixo teor em lípidos, os quais não devem constituir mais de 15% das

calorias ingeridas. Esta medida visa reduzir efeitos de uma eventual hiperlipoproteinemia, uma condição de saúde em que o organismo não é capaz de digerir lípidos. A21 sugere um maior rigor na gestão do peso e a intensificação da actividade física. A *action* A22 sugere a ingestão de fármacos que reduzem os níveis de triglicéridos, tais como o fibrato e o ácido nicotínico.

Em seguida surge o *enquiry* E24, cujo objectivo é a obtenção do nível de triglicéridos do paciente após o tratamento. No final do plano realiza-se a verificação do *goal*.

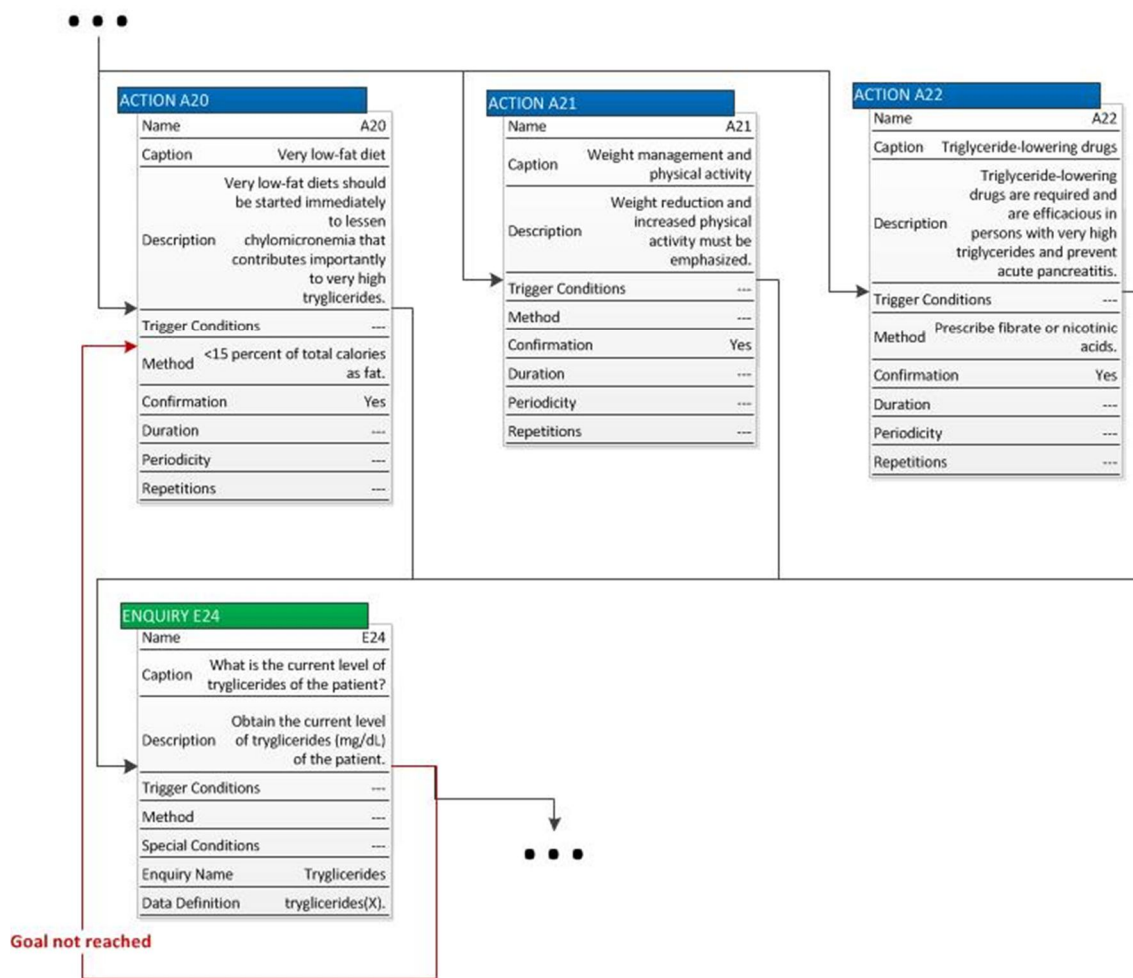


Figura 29 – Representação do *plan* P15.

6.4.13. *Plan* P16

O *plan* P16 possui as tarefas necessárias para a detecção e tratamento de baixos níveis de HDL-C.

P16 encontra-se representado na Figura 30 e, como se pode verificar, inicia-se com o *enquiry* E25, cujo objectivo é a obtenção do nível actual de HDL-C do paciente. Com base no

valor obtido e no risco de doença cardíaca coronária do paciente, determina-se em D9 a necessidade de tratamento do paciente para níveis baixos de HDL-C. O *plan* P17 contém os procedimentos para aumentar os níveis de HDL-C, ao passo que a *action* A23 não recomenda o paciente para tratamento.

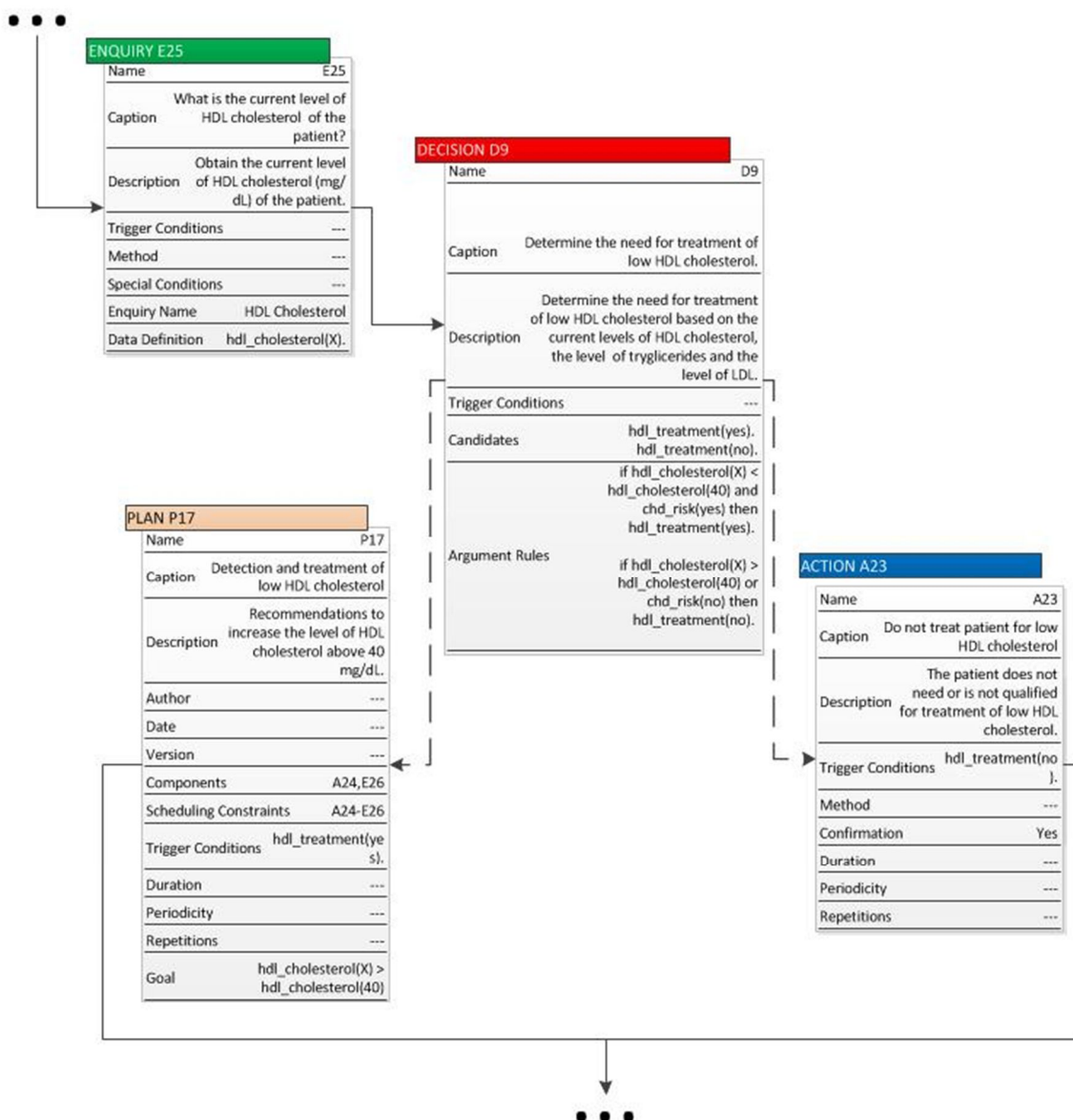


Figura 30 – Representação do *plan* P16.

6.4.14. *Plan* P17

Por fim, o *plan* P17 é constituído pela *action* A2 e pelo *enquiry* E26, como se pode verificar na Figura 31.

O *plan* P17 recomenda a utilização de ácido nicotínico e fibrato, de forma a aumentar os níveis de HDL-C. Após a utilização destes fármacos, o *enquiry* E26 tem como objectivo obter o nível de HDL-C para comparação com o *goal* definido em P17.

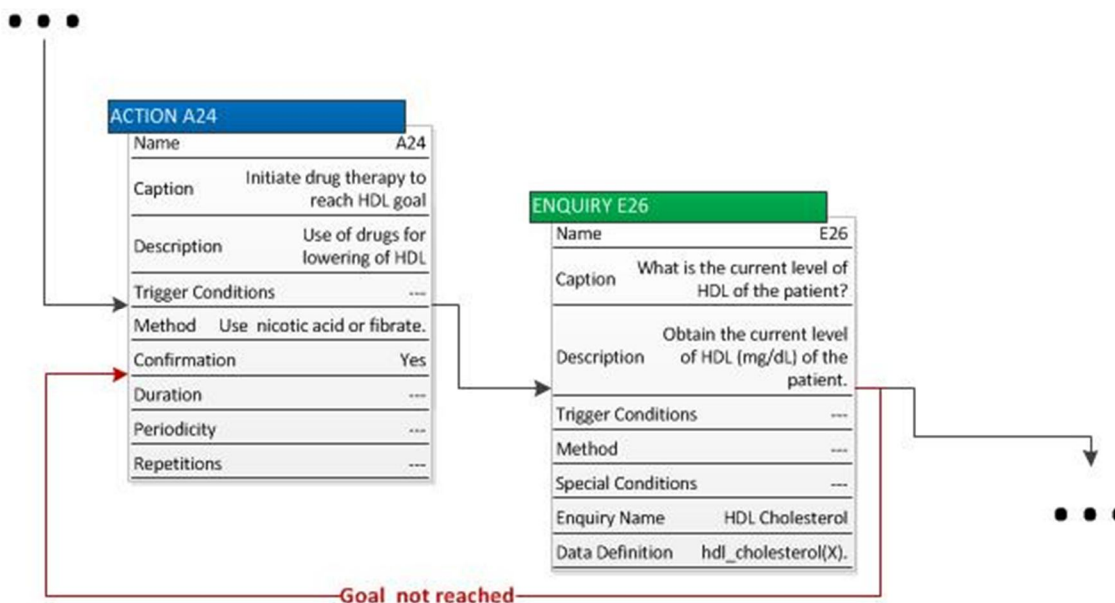


Figura 31 – Representação do *plan* P17.

6.5. Tomada de Decisão

A tomada de decisão é um dos focos principais desta dissertação. Como tal, procede-se de seguida a um exemplo de uma tarefa de decisão, em que se utilizará o motor de inferência referido no capítulo anterior. A tarefa seleccionada foi D2, do *plan* P1, representada na Figura 17.

A tarefa D2 possui três soluções candidatas, que correspondem às três metas que se podem estabelecer para o nível de LDL-C:

- Nível de LDL-C inferior a 100 mg/dL;
- Nível de LDL-C inferior a 130 mg/dL;
- Nível de LDL-C inferior a 160 mg/dL.

Estas soluções encontram-se na forma de conclusões, nas regras do Programa 14, que ditam qual destes níveis se deve aplicar.

```
if chd_risk(yes) or ten_year_risk(X) > ten_year_risk(20) then  
  ldl_goal_less(100).
```

```
if ten_year_risk(X) =< ten_year_risk(20) and chd_risk(no) and  
risk_factors(X) >= risk_factors(2) then ldl_goal_less(130).
```

```
if (risk_factors(0) or risk_factors(1)) and chd_risk(no) then  
  ldl_goal_less(160).
```

Programa 14 – Regras de produção da tarefa D2 para determinação dos níveis LDL-C a atingir.

Através das regras do Programa 14 é possível perceber que os parâmetros do estado do paciente que são utilizados na tarefa de decisão são: risco de doença cardíaca coronária, risco a dez anos de doença cardíaca coronária e o número de factores de risco presentes.

Considere-se agora um cenário 1 em que se determina com certeza que o paciente não possui risco de doença cardíaca coronária, com base na ausência de aterosclerose. Contudo, não é possível afirmar com certeza, o número de factores de risco que o paciente possui, mas há indícios de que o número de factores de risco presentes faz parte do conjunto {1,2,3}. Quanto ao risco a dez anos de doença cardíaca, dada a incerteza quanto aos factores de risco, não é possível calcular um valor exacto, mas sabe-se que é 19% ou 21%. O Programa 15 traduz o cenário 1 em EPL.

```
-chd_risk(X) ← not(fact(chd_risk(X))),not(abducible(chd_risk(X))).  
fact(chd_risk(no)).
```

```
-ten_year_risk(X) ← not(fact(ten_year_risk(X))),  
not(abducible(ten_year_risk(X))).  
abducible(ten_year_risk(19)).  
abducible(ten_year_risk(21)).  
+ten_year_risk(X)::(findall(X,fact(ten_year_risk(X)),Z),  
length(Z,N),N=<1).
```

```
-risk_factors(X) ←
not(fact(risk_factors(X))),not(abducible(risk_factors(X))).
abducible(risk_factors(1)).
abducible(risk_factors(2)).
abducible(risk_factors(3)).
```

Programa 15 – Extensões dos predicados do cenário 1.

Num cenário 2, não é possível averiguar a presença de aterosclerose, desconhecendo-se qualquer informação sobre a existência de risco de doença cardíaca coronária. No entanto, sabe-se que o número de factores de risco presentes é 2. O risco de doença cardíaca coronária foi avaliado em 15%. O Programa 16 traduz este cenário de informação.

```
-chd_risk(X) ← not(fact(chd_risk(X))),not(abducible(chd_risk(X))).
abducible(chd_risk(X)) :- fact(chd_risk(+)).
fact(chd_risk(+)).
```

```
-ten_year_risk(X) ← not(fact(ten_year_risk(X))),
not(abducible(ten_year_risk(X))).
fact(ten_year_risk(15)).
```

```
-risk_factors(X) ←
not(fact(risk_factors(X))),not(abducible(risk_factors(X))).
fact(risk_factors(2)).
```

Programa 16 – Extensões dos predicados do cenário 2.

Através da utilização do predicado `demo_forward`, verifica-se quais as conclusões que são válidas e calcula-se as pontuações para essas conclusões. As cláusulas do predicado `composition` que permitem lidar com os casos de `ten_year_risk` e `risk_factors` encontram-se no programa 18 do Anexo A. Embora se represente apenas as cláusulas para `ten_year_risk`, as de `risk_factors` são análogas. No caso do cenário 1, as QoIs, com aproximação às centésimas dos predicados, calculam-se segundo:

$$QoI_{chd_risk} = 1 \quad (15)$$

$$QoI_{ten_year_risk} = \frac{1}{2} = 0.50 \quad (16)$$

$$QoI_{risk_factors} = \frac{1}{2^3-1} = 0.14 \quad (17)$$

As conclusões possíveis a partir do cenário 1 são: nível de LDL-C inferior a 100 mg/dL, nível de LDL-C inferior a 130 mg/dL e nível de LDL-C inferior a 160 mg/dL. As pontuações das conclusões calculam-se segundo:

$$V_{ldl_goal_less(100)} = QoI_{ten_year_risk} = 0.50 \quad (18)$$

$$V_{ldl_goal_less(130)} = \frac{QoI_{ten_year_risk} + QoI_{chd_risk} + QoI_{risk_factors}}{3} = \frac{0.50 + 1 + 0.14}{3} = 0.55 \quad (19)$$

$$V_{ldl_goal_less(160)} = \frac{QoI_{chd_risk} + QoI_{risk_factors}}{2} = \frac{1 + 0.14}{2} = 0.57 \quad (20)$$

Através do predicado `conclusions` é possível ordenar as conclusões por ordem decrescente de pontuação, seleccionando a conclusão que se encontra no topo da lista, que no caso, é um nível de LDL-C inferior a 160 mg/dL. O processo clínico é direccionado para a tarefa que apresenta a *trigger condition* correspondente à conclusão.

No caso do cenário 2, os cálculos, com aproximação às centésimas, das QoIs e das pontuações das conclusões são:

$$QoI_{chd_risk} = 0 \quad (15)$$

$$QoI_{ten_year_risk} = 1 \quad (16)$$

$$QoI_{risk_factors} = 1 \quad (17)$$

$$V_{ldl_goal_less(100)} = QoI_{ten_year_risk} = 0 \quad (15)$$

$$V_{ldl_goal_less(130)} = \frac{QoI_{ten_year_risk} + QoI_{chd_risk} + QoI_{risk_factors}}{3} = \frac{1 + 0 + 1}{3} = 0.67 \quad (16)$$

A conclusão nível de LDL-C inferior a 160 mg/dL não é válida, pois não se satisfazem as suas condições. Apesar da pontuação para o nível de LDL-C inferior a 100 mg/dL para o risco de doença cardíaca coronária ser 0, não quer dizer que a conclusão não seja válida, apenas fornece uma medida da solidez da informação em que se baseia. No cenário 2, a conclusão seleccionada seria o nível de LDL-C inferior a 130 mg/dL.

Deste modo, torna-se possível fornecer ao utilizador, que está a executar uma guia no modelo proposto, uma espécie de painel de bordo em cada tarefa de decisão, que resume a informação relevante naquele momento e fornece uma medida da qualidade dessa informação. O painel de bordo pode ter o aspecto de gráficos circulares com os valores dos pesos relativos de cada condição. Os pesos relativos obtêm-se através do quociente da QoI de cada condição pelo número de condições de uma conclusão. A pontuação de uma conclusão obtêm-se através da soma destes pesos relativos. A Figura 32 exemplifica um possível painel de bordo para o cenário 1.

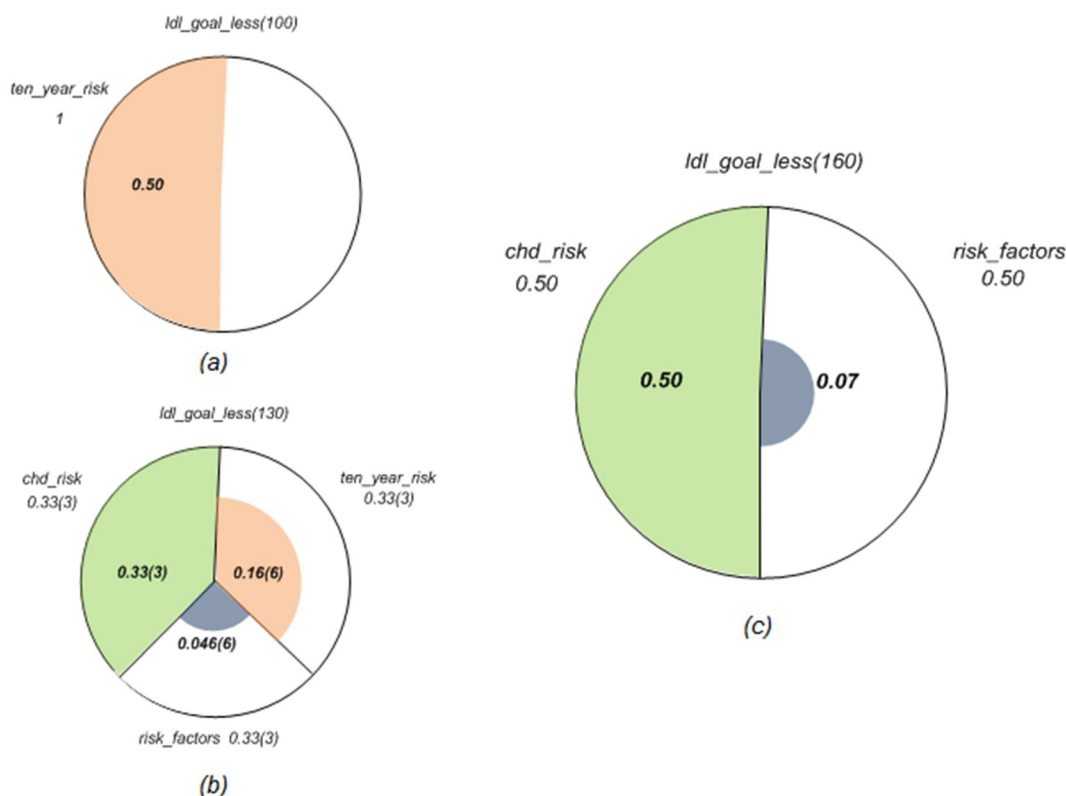


Figura 32 – Representação do painel de bordo do cenário 1 na decision D2: (a) pontuação do nível de LDL-C inferior a 100 mg/dL, (b) pontuação do nível de LDL-C inferior a 130 mg/dL e (c) pontuação do nível de LDL-C inferior a 160 mg/dL.

As decisões assumem a forma de uma escolha entre diferentes opções e o que se pretende é simular os processos cognitivos envolvidos no raciocínio em diagnósticos clínicos. Do ponto de vista da teoria de decisão, a realização de um diagnóstico é a actualização de uma opinião baseada em Informação Imperfeita, as evidências clínicas (Elstein & Schwarz, 2002).

O modelo de decisão que se propõe está patente em cada tarefa de decisão e é composto por cinco fases, ilustradas na Figura 33.

A primeira fase consiste na Formulação de Hipóteses, na qual se consideram todas as soluções candidatas presentes numa dada tarefa de decisão e as regras de produção que as sustentam em *argument rules*.

Segue-se a fase de Verificação, em que se verifica se as evidências (condições) que compõem as regras de produção, de cada solução candidata, existem na base de conhecimento como informação positiva ou na forma de valores nulos, como casos de Informação Imperfeita.

Se não se verificar a existência das evidências nestes moldes, passa-se à solução candidata seguinte. No caso de a informação ser positiva ou existir como valor nulo, procede-se à fase de Avaliação, em que se calcula a Qol para cada condição e se calcula a pontuação da solução candidata em questão, com base nos valores de Qol obtidos. No final desta fase, reinicia-se o processo até não existirem soluções candidatas disponíveis.

Em posse das pontuações das soluções candidatas, que correspondem às opções disponíveis, procede-se à Selecção, em que se elabora uma lista de preferências com as diferentes soluções e selecciona aquela que possui o valor mais elevado de pontuação.

Uma vez seleccionada a solução procede-se à fase de Atribuição de Tarefas. Nesta fase atribui-se a tarefa seguinte no processo clínico, de acordo com a conclusão da solução seleccionada. Esta fase é possibilitada pela existência das *trigger conditions*.

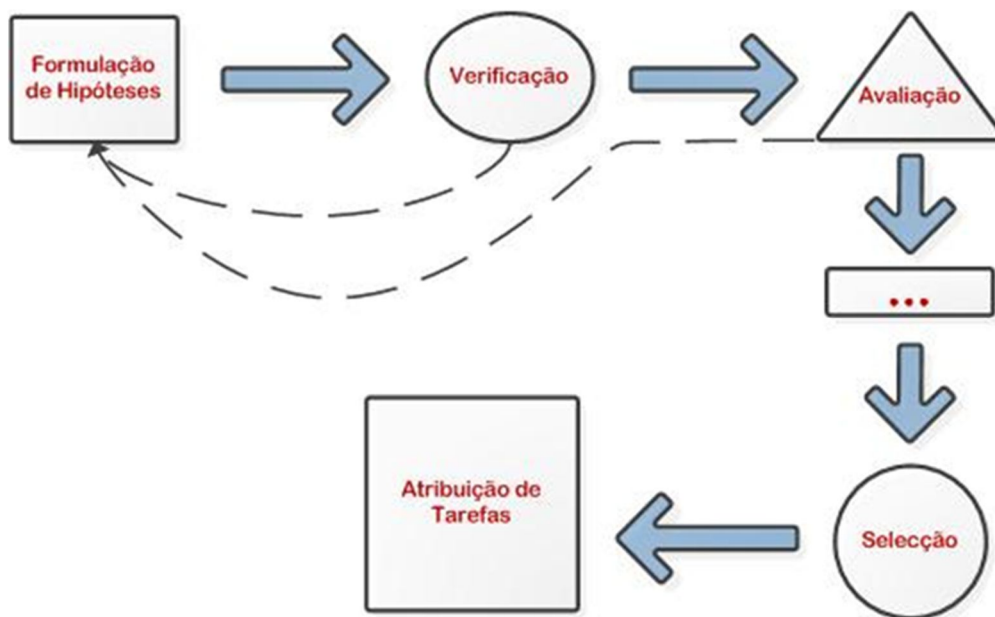


Figura 33 – Representação do modelo de tomada de decisão baseado em GCs.

6.6. Análise

A representação da guia Detection and Treatment of High Blood Cholesterol in Adults, no modelo proposto, revelou que este último possui uma boa capacidade de adaptação às diferentes necessidades de informação da guia. Foi possível distribuir a informação dos diferentes procedimentos que compõem a guia pelas diferentes primitivas de representação, sem, no entanto, desvirtuar o seu significado.

O modelo proposto é orientado aos objectos, ou seja, tem como base um conjunto de primitivas de representação que simulam os diferentes componentes de um processo clínico. Ao nível das primitivas de representação, o modelo baseia-se no conjunto de tarefas definidas por PROforma: *plans*, *actions*, *decisions* e *enquiries*. Esta abordagem permite ao utilizador perceber rapidamente em que consiste o conteúdo de cada primitiva. A representação na forma de fluxograma facilita ao utilizador a compreensão do processo clínico, pois confere uma maior percepção do papel desempenhado pelas instâncias de cada primitiva. A visualização do processo clínico como um fluxograma é implementada em PROforma, GLIF e EON.

A utilização de *plans* permite a definição de diferentes níveis de complexidade na guia. Neste contexto, um plano pode ser uma guia ou um conjunto de tarefas de uma guia. À

semelhança de EON, GLIF, Asbru e SAGE, o modelo permite a definição de *sub-guias*. Isto foi particularmente útil na divisão dos passos da guia em planos modulares, que podem ser referenciados no interior de outros planos. Tal como as demais abordagens, este modelo permite definir informação administrativa tal como autoria, data de criação e versão de uma guia.

O modelo apresenta uma noção de *action* que consiste numa recomendação médica ou numa intervenção clínica, não assumindo a mesma dimensão que as primitivas equivalentes em GLIF, EON e SAGE. Nestas abordagens, as *actions* também passam pelo envio de mensagens no seio do sistema de informação que as alberga e passos de aquisição de informação também são considerados acções. No entanto, no modelo proposto define-se um tipo de tarefa especialmente para a aquisição de dados, o *enquiry*.

As *decisions*, neste modelo, são semelhantes às de PROforma, possuindo os mesmos atributos. Contudo diferem na forma de representação do conhecimento, pois são utilizadas regras de produção. Também em Arden Syntax utilizam-se regras semelhantes e o objectivo da aproximação a esta abordagem, é aproveitar os seus pontos fortes, que passam pela modularidade dos seus MLMs. As *decisions* correspondem a pontos de escolha no fluxograma representativo da guia e o processo clínico é direccionado com recurso a *trigger conditions*.

Ao nível das restrições da execução das tarefas, definem-se nas tarefas *action* e *plan*, atributos que representam os casos básicos de restrições. Na guia que foi representada, recorreu-se com frequência ao atributo *goal*, para definir os objectivos de planos que visavam aumentar ou diminuir os níveis de lipoproteínas. Para além de PROforma, pode-se encontrar um paralelismo na definição de *intentions* em Asbru. O modelo também permite expressar duração de tarefas, tarefas periódicas e ciclos de tarefas. Quanto ao agendamento de tarefas, define-se uma sintaxe que permite expressar tarefas sequenciais, simultâneas e alternativas. Embora esta sintaxe não seja tão expressiva como a lógica temporal de Asbru, permitiu representar os diferentes passos da guia.

No que toca à estruturação da informação e definição de tipos de dados, o modelo fica um pouco a dever a abordagens como GLIF e SAGE, que utilizam *standards* como o RIM do grupo HL7 para a definição de tipos de dados. A representação de conhecimento específico de domínio médico também não foi contemplada no modelo proposto. Este aspecto não foi considerado essencial face aos objectivos do presente trabalho.

No que toca à linguagem utilizada para representar o conhecimento, utiliza-se a EPL. A EPL permite uma elevada expressividade em termos de Informação Imperfeita, permitindo definir

diferentes casos de ausência de conhecimento, como se pode ver pelos diferentes exemplos apresentados ao longo do documento. Nenhuma das abordagens mencionadas considera a Informação Imperfeita no contexto do processo clínico. A tomada de decisão passa a ser apoiada por uma medida da qualidade de informação, que permite ao utilizador optar com base na solidez das evidências.

O modelo proposto permite ao utilizador avaliar o estado do conhecimento em cada tarefa de decisão. Isto é possível através de um painel de bordo que fornece medidas do grau de confiança de cada um dos parâmetros de saúde do paciente, uma característica que as abordagens à modelação de CIGs actuais não possuem.

7. Conclusões e Trabalho Futuro

7.1. Conclusões

Desde a sua projecção até à sua finalização, este trabalho de dissertação focou-se no estudo de formas de resposta às necessidades de conhecimento dos profissionais de saúde durante o processo clínico. Do estudo dessas necessidades, verificou-se que os profissionais de saúde estão sujeitos a situações de stress, que são responsáveis por erros médicos, variações da prática clínica e prática de medicina defensiva. Concluiu-se que é necessário aproximar os profissionais de saúde das boas práticas clínicas e da medicina baseada na evidência. As GCs são o veículo ideal para este efeito.

As GCs são documentos desenvolvidos de forma sistemática, com base em evidências científicas, que aconselham os profissionais de saúde sobre boas práticas clínicas. Do estudo dos benefícios e limitações de GCs, concluiu-se que o seu formato em papel não é adequado para consulta rápida e que a solução passa pela implementação de formatos informáticos de guias integrados em sistemas de apoio à decisão, as CIGs.

Os principais modelos das diferentes abordagens de CIGs focam-se sobretudo nos aspectos temporais da execução de guias e na integração com sistemas de informação das instituições de saúde, deixando de lado a questão da Informação Imperfeita no processo clínico. Esta é uma questão que não deve ser ignorada, visto que é comum os profissionais de saúde tomarem decisões em cenários de incerteza.

Com o intuito de melhorar os resultados do processo clínico e lidar com situações de Informação Incompleta, desenvolveu-se um modelo de representação genérico de guias. Verificou-se que os diferentes modelos existentes recorrem a blocos de construção de guias semelhantes, que se designam por primitivas de representação. A construção do novo modelo baseou-se nas primitivas de representação mais comuns.

A Informação Imperfeita engloba casos de incerteza, incompletude, inexactidão e incoerência, e a EPL surge como a linguagem de programação de maior expressividade para estes casos. A EPL tem associado um método de quantificação de Informação Imperfeita, a QoI.

O modelo desenvolvido representa as guias como conjuntos de tarefas, que podem assumir a forma de *plans*, *actions*, *decisions* e *enquiries*. O modelo permite o agendamento de

tarefas e o estabelecimento de objectivos e restrições temporais à execução de planos. A ELP é utilizada como linguagem de representação do conhecimento e a Qol como método de quantificação da informação. Verificou-se que este modelo é capaz de representar uma guia utilizada na prática clínica diária através de um caso de estudo. Em cada *decision* utiliza-se a Qol como suporte à escolha das diferentes opções disponíveis. A informação surge como um painel de bordo com os valores das pontuações de cada uma das opções.

Embora o modelo apresentado possua algumas limitações ao nível da estruturação de dados e reconhecimento de conceitos específicos do domínio clínico, pode concluir-se que apresenta o potencial de melhorar os resultados do processo clínico, fornecendo um suporte mais sólido para a tomada de decisão na presença de incerteza. Este é um território pouco explorado no domínio de CIGs e o modelo proposto pode servir como um ponto de partida para o desenvolvimento de um SADC mais completo, capaz de auxiliar os profissionais de saúde a evitar comportamentos de medicina defensiva, variações na prática clínica e eventuais erros médicos.

7.2. Contribuições

Este trabalho abordou uma área que, embora possua projectos relevantes, ainda carece de desenvolvimento. Neste projecto optou-se por um modelo de representação de GCs que reúne as características fundamentais dos modelos de representação existentes, mas que surge como uma abordagem de CIGs mais genérica. Abordou-se dois problemas importantes: a modelação de guias e a incorporação de Informação Imperfeita em GCs. Nesta dissertação dá-se a conhecer os aspectos fundamentais sobre GCs, como: definição, benefícios, limitações e metodologia de desenvolvimento. Da análise das limitações, surge a necessidade de desenvolvimento de CIGs.

Em termos de trabalho realizado pode salientar-se:

- Levantamento dos benefícios de CIGs e obstáculos à sua implementação. Assim como a enumeração das principais abordagens de CIGs e definição de critérios de avaliação, que permitem apontar os aspectos em que diferem;
- Desenvolvimento de um modelo de representação de CIGs, baseado nas principais características das abordagens existentes, que através das suas primitivas de representação (*plans*, *actions*, *decisions* e *enquiries*) e restrições (temporais, de

agendamento e por objectivos) à execução de GCs capta as necessidades de informação do processo clínico;

- Incorporação de Informação Imperfeita no modelo de representação desenvolvido através de EPL;
- Quantificação da Informação Imperfeita através do método Qol no contexto de um processo clínico, associado às recomendações de uma GC;
- Desenvolvimento de um motor de inferência em linguagem de programação em lógica Prolog, capaz de reconhecer casos de Informação Imperfeita e calcular a Qol dos diferentes parâmetros de saúde de um paciente;
- Realização de um caso de estudo do modelo desenvolvido com uma guia para detecção e tratamento de elevados níveis de colesterol;
- Definição de um modelo de decisão baseado em GCs, constituído por cinco fases: Formulação de Hipóteses, Verificação, Avaliação, Selecção e Atribuição de Tarefas.

As reflexões e propostas realizadas neste trabalho de dissertação pretendem auxiliar os profissionais de saúde através da operacionalização das suas tarefas e de um suporte à tomada de decisão. Desta forma, pretende-se contribuir para aumentar a disponibilidade de GCs no momento do processo clínico, fornecendo um meio que permita prevenir casos de erro médico, variações na prática clínica e medicina defensiva.

7.3. Trabalho Relevante

Este trabalho nasceu como resposta às necessidades do panorama actual de desenvolvimento de CIGs. Este projecto foi submetido a concurso na Fundação de Ciência e Tecnologia (FCT) com o título: "Active Guidelines in a Clinical Setting".

Em termos de publicações, o trabalho originou um artigo em conferência, com a referência:

- Tiago Oliveira, Ângelo Costa, João Neves, Paulo Novais, "Digital Clinical Guidelines Modelling", 25th European Simulation and Modelling Conference (ESM'2011) Guimarães (a publicar).

7.4. Trabalho Futuro

Na última década, grande parte do desenvolvimento de CIGs focou-se na representação de guias e na linguagem subjacente a essa representação. Desta investigação resultaram vários modelos, cada um com as suas vantagens e limitações. Contudo, a comparação das diferentes abordagens sugere que os componentes mínimos para a representação de guias já estão definidos. Grande parte das abordagens apresenta primitivas de representação comuns.

O próximo passo será o desenvolvimento de um modelo de representação *standard* utilizando os componentes acima referidos. Este modelo de representação deve apoiar-se em *standards* de informação, como é o caso do RIM do grupo HL7, de forma a tornar mais fácil a implementação de guias a nível local, integradas nos sistemas de informação das instituições. SAGE é um exemplo da nova orientação que o desenvolvimento de CIGs está a seguir.

Contudo, há que realçar a necessidade futura de abordar a questão da Informação Imperfeita no contexto clínico. O ambiente clínico é prolífico neste tipo de situações e neste contexto as decisões são de extrema importância, podendo produzir resultados irreversíveis. Daí a necessidade de apresentar fundamentações sólidas para as decisões.

Quanto ao modelo de representação proposto, é possível abordar as suas limitações através do desenvolvimento de um suporte para o estado do paciente e da implementação de *standards* de informação que permitam uma melhor estruturação dos dados. Outro aspecto a considerar é a utilização de terminologias que permitam o reconhecimento de vocabulário específico do domínio clínico.

Referências Bibliográficas

- Aamodt, A., & Nygård, M. (1995). Different roles and mutual dependencies of data, information, and knowledge - An AI perspective on their integration. *Data & Knowledge Engineering*, 16(3), 191-222.
- Analide, C., Novais, P., Machado, J., & Neves, J. (2006). Quality of knowledge in virtual entities. *Encyclopedia of communities of practice in information and knowledge management*.
- Andersen, S., Olesen, K., Jensen, F., & Jensen, F. (1989). HUGIN—a shell for building Bayesian belief universes for expert systems. *Citeseer*, 2, pp. 1080-1085.
- Antonopoulos, S. (2002). Third report of the National Cholesterol Education Program (NCEP) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (Adult Treatment Panel III) final report. *Circulation*, 106(3143), 3421.
- Barnes, M., & Barnett, G. (1995). An architecture for a distributed guideline server. *American Medical Informatics Association*, (p. 233).
- Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1), D267.
- Bratko, I. (2001). *Prolog programming for artificial intelligence*. Addison-Wesley Longman Ltd.
- Carnett, W. (1999). Clinical practice guidelines: a tool to improve care. *Quality Management in Healthcare*, 8(1), 13.
- Chim, J., Cheung, N., Fung, H., & Wong, K. (2003). Electronic clinical practice guidelines: current status and future prospects in Hong Kong. *Hong Kong Medical Journal*, 9, 299-301.
- Clayton, P., Pryor, T., Wigertz, O., & Hripcsak, G. (1989). Issues and structures for sharing medical knowledge among decision-making systems: the 1989 Arden Homestead Retreat. *American Medical Informatics Association*, (p. 116).
- Clercq, P. d., Blom, J., Korsten, H., & Hasman, A. (2004). Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1), 1-27.
- Colmerauer, A., Kanoui, H., Pasero, R., & Roussel, P. (1972). Un système de communication homme-machine en français. *Groupe de Recherche en Intelligence Artificielle, Univ. de Aix-Marseille, Luminy*, 3, 99.

- Cooper, P. (2010). Data, information and knowledge. *Anaesthesia & Intensive Care Medicine*, 11(12), 505-506.
- Dempster, A. (1967). Upper and lower probability inferences based on a sample from a finite univariate population. *Biometrika*, 54(3-4), 515.
- Duff, L., Kitson, A., Seers, K., & Humphris, D. (1996). Clinical guidelines: an introduction to their development and implementation. *Journal of Advanced Nursing*, 23(5), 887-895.
- Elkin, P., Peleg, M., Lacson, R., Bernstam, E., Tu, S., Boxwala, A., et al. (2000). Toward standardization of electronic guideline representation. *MD Computing*, 17(6), 39-44.
- Elstein, A., & Schwarz, A. (2002). Clinical problem solving and diagnostic decision making: selective review of the cognitive literature. *Bmj*, 324(7339), 729.
- Emden, M. V., & Kowalski, R. (1976). The semantics of predicate logic as a programming language. *Journal of the ACM (JACM)*, 23(4), 733-742.
- Farquhar, C., Kofa, E., & Slutsky, J. (1992). Clinicians' attitudes to clinical practice guidelines: a systematic review. *methods*, 3, 5.
- Fensel, D., & Schnogge, A. (1997). Using KIV to specify and verify architectures of knowledge-based systems. *Published by the IEEE Computer Society*, (p. 71).
- Fink, A., Kosecoff, J., Chassin, M., & Brook, R. (1984). Consensus methods: characteristics and guidelines for use. *American Journal of Public Health*, 74(9), 979.
- Fox, J., & Das, S. (2000). Arguments about beliefs and actions: decision making in the real world. *Safe and sound: artificial intelligence in hazardous applications*, 55-76.
- Fox, J., Johns, N., & Rahmzadeh, A. (1998). Disseminating medical knowledge: the PROforma approach. *Artificial Intelligence in Medicine*, 14(1-2), 157-182.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29(2-3), 131-163.
- Gardner, R. M., Pryor, T., & Warner, H. R. (1999). The HELP hospital information system: update 1998. *International Journal of Medical Informatics*, 54(3), 169-182.
- Ginsberg, M. (1991). The computational value of nonmonotonic reasoning. *Citeseer*, (pp. 262-268).
- Graham, R., James, P., & Cowan, T. (2000). Are clinical practice guidelines valid for primary care? *Journal of clinical epidemiology*, 53(9), 949-954.

- Greenes, R., Boxwala, A., Sloan, W., Ohno-Machado, L., & Deibel, S. (1999). A framework and tools for authoring, editing, documenting, sharing, searching, navigating, and executing computer-based clinical guidelines. *American Medical Informatics Association*, (p. 261).
- Grosso, W., Eriksson, H., Fergerson, R., Gennari, J., Tu, S., & Musen, M. (1999). Knowledge modeling at the millennium. *Proc. KAW'99*.
- Guyatt, G., Oxman, A., Akl, E., Kunz, R., Vist, G., Brozek, J., et al. (2011). GRADE guidelines: 1. Introduction-GRADE evidence profiles and summary of findings tables. *Journal of clinical epidemiology*, 64(4), 383-394.
- Hampton, J. (1997). Evidence-based medicine, practice variations and clinical freedom. *Journal of evaluation in clinical practice*, 3(2), 123-131.
- Haux, R. (2010). Medical informatics: past, present, future. *International journal of medical informatics*, 79(9), 599-610.
- Hommersom, A., Lucas, P., & Bommel, P. V. (2008). Checking the quality of clinical guidelines using automated reasoning tools. *Theory and Practice of Logic Programming*, 8(5-6), 611-641.
- Hripcsak, G. (1994). Writing Arden Syntax medical logic modules. *Computers in Biology and Medicine*, 24(5), 331-363.
- Hripcsak, G., Ludemann, P., Pryor, T., Wigertz, O., & Clayton, P. (1994). Rationale for the Arden syntax. *Computers and Biomedical Research*, 27(4), 291-324.
- ICSI. (2011). Health Care Guideline:Diagnosis and Treatment of Respiratory Illness in Children and Adults.
- Isern, D., & Moreno, A. (2008). Computer-based execution of clinical guidelines: a review. *International journal of medical informatics*, 77(12), 787-808.
- Jenders, R., & Dasgupta, B. (1996). Assessment of a knowledge-acquisition tool for writing Medical Logic Modules in the Arden Syntax. *American Medical Informatics Association*, (p. 567).
- Kaiser, K., & Miksch, S. (2009). Versioning computer-interpretable guidelines: Semi-automatic modeling of Living Guidelines' using an information extraction method. *Artificial intelligence in medicine*, 46(1), 55-66.
- Kakas, A., Kowalski, R., & Toni, F. (1998). The Role of Abduction in Logic Programming. *Handbook of Logic in Artificial Intelligence and Logic Programming: Logic programming*, 5, 235.

- Karadimas, H., Chailloleau, C., Hemery, F., Simonnet, J., & Lepage, E. (2002). Arden: an architecture for MLM execution on the Java platform. *Journal of the American Medical Informatics Association*, 9(4), 359.
- Kim, S., Haug, P., Rocha, R., & Choi, I. (2008). Modeling the Arden Syntax for medical decisions in XML. *International journal of medical informatics*, 77(10), 650-656.
- Kuhn, R., & Reider, R. (1994). A C++ framework for developing Medical Logic Modules and an Arden Syntax compiler. *Computers in biology and medicine*, 24(5), 365-370.
- Lindberg, C. (1990). The Unified Medical Language System (UMLS) of the National Library of Medicine. *Journal (American Medical Record Association)*, 61(5), 40.
- Lindgren, H., & Eklund, P. (2005). Logic of Dementia Guidelines in a Probabilistic Argumentation Framework. In L. Godo (Ed.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (Vol. 3571, pp. 470-470). Springer Berlin / Heidelberg.
- Linzer, M. (2005). *Organizational climate, stress, and error in primary care: the MEMO Study*. DTIC Document.
- Liu, J., & Shiffman, R. (1997). Operationalization of clinical practice guidelines using fuzzy logic. *American Medical Informatics Association*, (p. 283).
- Liu, J., Yang, J., Wang, J., & Sii, H. (2002). Review of uncertainty reasoning approaches as guidance for maritime and offshore safety-based assessment. *Journal of UK Safety and Reliability Society*, 23(1), 63-80.
- Lohr, K., Eleazer, K., & Mauskopf, J. (1998). Health policy issues and applications for evidence-based medicine and clinical practice guidelines. *Health Policy*, 46(1), 1-19.
- Lucas, P. (2003). *Quality checking of medical guidelines through logical abduction*. Citeseer.
- Ma, H. (1995). Mapping clause of Arden Syntax with HL7 and ASTM E 1238-88 standard. *International journal of bio-medical computing*, 38(1), 9-21.
- Mancia, G., Backer, G. D., Dominiczak, A., Cifkova, R., Fagard, R., Germano, G., et al. (2007). 2007 Guidelines for the management of arterial hypertension. *European Heart Journal*, 28(12), 1462.
- Mani, S., & Aliferis, C. (2007). A Causal Modeling Framework for Generating Clinical Practice Guidelines from Data. In R. Bellazzi, A. Abu-Hanna, & J. Hunter (Edits.), *Artificial Intelligence in Medicine* (Vol. 4594, pp. 446-450). Springer Berlin / Heidelberg.
- Maslach, C., Schaufeli, W., & Leiter, M. (2001). Job burnout. *Annual review of psychology*, 52(1), 397-422.

- McCarthy, J. (1959). Programs with Common Sense. (pp. 75-91). Her Majesty's Stationary Office.
- McCauley, B., Young, I., Clark, I., & Peters, M. (1996). Incorporation of the Arden Syntax within the reimplementation of a closed-loop decision support system. *Computers and biomedical research*, 29(6), 507-518.
- Mead, P. (2000). Clinical guidelines: promoting clinical effectiveness or a professional minefield? *Journal of Advanced Nursing*, 31(1), 110-116.
- Melo, B., Gomes, A., & Cruz, J. (1997). Stress ocupacional em profissionais da saúde e do ensino.
- Miksch, S., Kosara, R., Shahar, Y., & Johnson, P. (1998). Asbruvew: Visualization of time-oriented, skeletal plans., (pp. 11-18).
- Miksch, S., Shahar, Y., & Johnson, P. (1997). Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. *Citeseer*, (p. 9).
- Miller, M., & Kearney, N. (2004). Guidelines for clinical practice: development, dissemination and implementation. *International journal of nursing studies*, 41(7), 813-821.
- Morgan, M., & Henrion, M. (1992). *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*. Cambridge Univ Pr.
- Musen, M., Shahar, Y., & Shortliffe, E. (2006). Clinical decision-support systems. *Biomedical Informatics*, 698-736.
- Musen, M., Tu, S., Das, A., & Shahar, Y. (1996). EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 3(6), 367.
- Neves, J. (1984). A logic interpreter to handle time and negation in logic data bases. *ACM*, (pp. 50-54).
- Neves, J., Machado, J., Analide, C., Abelha, A., & Brito, L. (2007). The halt condition in genetic programming. *Progress in Artificial Intelligence*, 160-169.
- NHLBI. (2002). *Guideline-At-A-Glance: Detection and Treatment of High Blood Cholesterol*.
- Nilsson, N. (1986). Probabilistic logic 1. *Artificial intelligence*, 28(1), 71-87.
- Novais, P., Salazar, M., Ribeiro, J., Analide, C., & Neves, J. (2010). Decision Making and Quality-of-Information. *Springer*, (pp. 187-195).

- O'Connor, M., Shankar, R., Parrish, D., & Das, A. (2009). Knowledge-data integration for temporal reasoning in a clinical trial system. *International journal of medical informatics*, 78, S77--S85.
- Ohno-Machado, L., Gennari, J., Murphy, S., Jain, N., Tu, S., Oliver, D., et al. (1998). The guideline interchange format. *Journal of the American Medical Informatics Association*, 5(4), 357.
- Ollenschlager, G., Marshall, C., Qureshi, S., Rosenbrand, K., Burgers, J., Makela, M., et al. (2004). Improving the quality of health care: using international collaboration to inform guideline programmes by founding the Guidelines International Network (GIN). *Quality and Safety in Health Care*, 13(6), 455.
- Patil, R., Fikes, R., Patel-Schneider, P., McKay, D., Finin, T., Gruber, T., et al. (1992). The DARPA knowledge sharing effort: Progress report. *Citeseer*.
- Pearl, J. (1986). Fusion, propagation, and structuring in belief networks* 1. *Artificial intelligence*, 29(3), 241-288.
- Pearson, K. (1998). Role of evidence-based medicine and clinical practice guidelines in treatment decisions. *Clinical therapeutics*, 20, C80--C85.
- Peleg, M., Boxwala, A., Bernstam, E., Tu, S., Greenes, R., & Shortliffe, E. (2001). Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. *Journal of Biomedical Informatics*, 34(3), 170-181.
- Peleg, M., Boxwala, A., Ogunyemi, O., Zeng, Q., Tu, S., Lacson, R., et al. (2000). GLIF3: the evolution of a guideline representation format. *American Medical Informatics Association*, (p. 645).
- Peleg, M., Keren, S., & Denekamp, Y. (2008). Mapping computerized clinical guidelines to electronic medical records: Knowledge-data ontological mapper (KDOM). *Journal of biomedical informatics*, 41(1), 180-201.
- Peleg, M., Ogunyemi, O., Tu, S., Boxwala, A., Zeng, Q., Greenes, R., et al. (2001). Using features of Arden Syntax with object-oriented medical data models for guideline modeling. *American Medical Informatics Association*, (p. 523).
- Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R., et al. (2003). Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, 10(1), 52.

- Ram, P., Berg, D., Tu, S., Mansfield, G., Ye, Q., Abarbanel, R., et al. (2004). Executing clinical practice guidelines using the SAGE execution engine. *Medinfo*, 11(Pt 1), 251-5.
- Ribeiro, J., Machado, J., Abelha, A., Fernandez-Delgado, M., & Neves, J. (2010). Integrating Incomplete Information into the Relational Data Model., 1.
- Rosenbrand, K., Croonenborg, J. V., & Wittenberg, J. (2008). *Guideline development*. IOS Press: Amsterdam.
- Rosenfeld, R., & Shiffman, R. (2006). Clinical practice guidelines: a manual for developing evidence-based guidelines to facilitate performance measurement and quality improvement. *Otolaryngology-Head and Neck Surgery*, 135(4 suppl), S1.
- Rumbaugh, J., Jacobson, R., & Booch, G. (1999). The unified modelling language reference manual.
- Schadow, G., Russler, D., Mead, C., & McDonald, C. (2000). Integrating medical information and knowledge in the HL7 RIM. *American Medical Informatics Association*, (p. 764).
- Schaffer, A., Levitt, A. J., Hershkop, S. K., Oh, P., MacDonald, C., & Lanctot, K. (2002). Utility scores of symptom profiles in major depression. *Psychiatry Research*, 110(2), 189-197.
- Seyfang, A., Martinez-Salvador, B., Serban, R., Wittenberg, J., Miksch, S., Marcos, M., et al. (2007). Maintaining formal models of living guidelines efficiently. *Artificial Intelligence in Medicine*, 441-445.
- Shahar, Y., Miksch, S., & Johnson, P. (1998). The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1-2), 29-51.
- Shahar, Y., Young, O., Shalom, E., Mayaffit, A., Moskovitch, R., Hessian, A., et al. (2003). DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. *Springer-Verlag New York Inc*, 2780, p. 122.
- Shenoy, P., & Shafer, G. (1986). Propagating belief functions with local computations. *IEEE Expert*, 1(3), 43-52.
- Sheridan, F. (1991). A survey of techniques for inference under uncertainty. *Artificial Intelligence Review*, 5(1), 89-119.
- Shiffman, R., Shekelle, P., Overhage, J., Slutsky, J., Grimshaw, J., & Deshpande, A. (2003). Standardized reporting of clinical practice guidelines: a proposal from the Conference on Guideline Standardization. *Annals of internal medicine*, 139(6), 493.

- Silva, M. (2004). *Fundação Portuguesa de Cardiologia*. Obtido em 22 de Agosto de 2011, de <http://www.fpcardiologia.pt/docs/N7.pdf>
- Siriwardena, A. (1995). Clinical guidelines in primary care: a survey of general practitioners' attitudes and behaviour. *The British Journal of General Practice*, 45(401), 643.
- Somekh, B. (2005). *Action Research: A Methodology for Change and Development*. McGraw-Hill.
- Sonnenberg, F., & Hagerty, C. (2006). Computer-Interpretable Clinical Practice Guidelines. *Methods Inf Med*, 45(1), S145-58.
- Sordo, M., Ogunyemi, O., Boxwala, A., & Greenes, R. (2003). GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support: AMIA 2003 Open Source Expo. *American Medical Informatics Association, 2003*, p. 1012.
- Stoufflet, P., Ohno-Machado, L., Deibel, S., Lee, D., & Greenes, R. (1996). GEODE-CM: a state-transition framework for clinical management. *American Medical Informatics Association*, (p. 924).
- Straszecka, E. (2006). Combining uncertainty and imprecision in models of medical diagnosis. *Information Sciences*, 176(20), 3026-3059.
- Studdert, D., Mello, M., Sage, W., DesRoches, C., Peugh, J., Zapert, K., et al. (2005). Defensive medicine among high-risk specialist physicians in a volatile malpractice environment. *JAMA: the journal of the American Medical Association*, 293(21), 2609.
- Teije, A. T., Miksch, S., & Lucas, P. (2008). *Computer-based medical guidelines and protocols: a primer and current trends* (Vol. 139). los Pr Inc.
- Tierney, W., Overhage, J., Takesue, B., Harris, L., Murray, M., Vargo, D., et al. (1995). Computerizing guidelines to improve care and patient outcomes: the example of heart failure. *Journal of the American Medical Informatics Association*, 2(5), 316.
- Toker, A., Shvarts, S., Perry, Z., Doron, Y., & Reuveni, H. (2004). Clinical guidelines, defensive medicine, and the physician between the two. *American journal of otolaryngology*, 25(4), 245-250.
- Tu, S., & Musen, M. (1999). A flexible approach to guideline modeling. *American Medical Informatics Association*, (p. 420).
- Tu, S., Campbell, J., Glasgow, J., Nyman, M., McClure, R., McClay, J., et al. (2007). The SAGE Guideline Model: achievements and overview. *Journal of the American Medical Informatics Association*, 14(5), 589.

- Tu, S., Musen, M., & others. (2001). Modeling data and knowledge in the EON guideline architecture. *Studies in health technology and informatics*, 280-284.
- Uusitalo, L. (2007). Advantages and challenges of Bayesian networks in environmental modelling. *Ecological modelling*, 203(3-4), 312-318.
- van, M. A., Taal, B. G., & Lucas, P. J. (2008). Dynamic Bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41(4), 515-529.
- Vetterlein, T., Mandl, H., & Adlassnig, K. (2010). Fuzzy Arden Syntax: A fuzzy programming language for medicine. *Artificial intelligence in medicine*, 49(1), 1-10.
- Vollebregt, A., Teije, A. t., Harmelen, F. v., der, J. v., & Mosseveld, M. (1999). A study of PROforma, a development methodology for clinical procedures. *Artificial Intelligence in Medicine*, 17(2), 195-221.
- Votruba, P., Miksch, S., & Kosara, R. (2004). Facilitating knowledge maintenance of clinical guidelines and protocols., (pp. 57-61).
- Wang, D., Peleg, M., Tu, S., Boxwala, A., Ogunyemi, O., Zeng, Q., et al. (2004). Design and implementation of the GLIF3 guideline execution engine. *Journal of biomedical informatics*, 37(5), 305-318.
- Wang, D., Peleg, M., Tu, S., Shortliffe, E., & Greenes, R. (2001). Representation of clinical practice guidelines for computer-based implementations. *Studies in health technology and informatics*, 285-289.
- Woolf, S., Grol, R., Hutchinson, A., Eccles, M., & Grimshaw, J. (1999). Potential benefits, limitations, and harms of clinical guidelines. *Bmj*, 318(7182), 527.
- Zadeh, L. (1965). Fuzzy sets. *Information and control*, 8(3), 338-353.
- Zadeh, L. (1975). Fuzzy logic and approximate reasoning. *Synthese*, 30(3), 407-428.
- Zadeh, L. (1996). Fuzzy logic= computing with words. *Fuzzy Systems, IEEE Transactions on*, 4(2), 103-111.
- Zielstorff, R., Teich, J., Paterno, M., Segal, M., Kuperman, G., Hiltz, F., et al. (1998). P-CAPE: a high-level tool for entering and processing clinical practice guidelines. Partners Computerized Algorithm and Editor. *American Medical Informatics Association*, (p. 478).

Referências Electrónicas

- G-I-N. (2011). *Membership*. Obtido em 13 de Julho de 2011, de Guidelines International Network: <http://www.g-i-n.net/membership>
- HL7. (2011). *Arden Syntax*. Obtido em 12 de 8 de 2011, de Health Level Seven International: <http://www.hl7.org/implement/standards/ardensyntax.cfm>
- OMS. (2008). *Blood Cholesterol*. Obtido em 22 de Agosto de 2011, de WHO Statistics: http://gamapserv.who.int/gho/interactive_charts/ncd/risk_factors/cholesterol_prevalence/atlas.html
- OMS. (2010). *Facts and Figures: Europe*. Obtido em 22 de Agosto de 2011, de World Health Organization.
- OMS. (2011). *National Health Accounts*. Obtido em 22 de Junho de 2011, de World Health Organization: http://www.who.int/whr/2010/whr10_pt.pdf
- W3C. (2011). *XML Essentials*. Obtido em 17 de Agosto de 2011, de W3C: <http://www.w3.org/standards/xml/core>

Anexos

Anexo A - Programas Auxiliares do Motor de Inferência

Os programas que se seguem correspondem a programas auxiliares, cuja inclusão no corpo do texto se considerou desnecessária, mas que são relevantes para a compreensão do funcionamento do motor de inferência.

O programa 17 corresponde às cláusulas do predicado `composition` que lidam com a conjunção e disjunção de condições.

```
composition(CONDITION1 and CONDITION2,CONCLUSION,LIST1) ←  
composition(CONDITION1,CONCLUSION,LIST2),composition(CONDITION2,CONCLU  
SION,LIST3),append(LIST2,LIST3,LIST1).
```

```
composition(CONDITION1 or CONDITION2,CONCLUSION,LIST1) ←  
composition(CONDITION1,CONCLUSION,LIST1),not(composition(CONDITION2,CO  
NCLUSION,LIST2)).
```

```
composition(CONDITION1 or CONDITION2,CONCLUSION,LIST2) ←  
composition(CONDITION2,CONCLUSION,LIST2),not(composition(CONDITION1,CO  
NCLUSION,LIST1)).
```

```
composition(CONDITION1 or CONDITION2,CONCLUSION,LIST1) ←  
composition(CONDITION1,CONCLUSION,LIST1),composition(CONDITION2,CONCLU  
SION,LIST2),sum(LIST1,R1),sum(LIST2,R2),R1>=R2.
```

```
composition(CONDITION1 or CONDITION2,CONCLUSION,LIST2) ←  
composition(CONDITION1,CONCLUSION,LIST1),composition(CONDITION2,CONCLU  
SION,LIST2),sum(LIST1,R1),sum(LIST2,R2),R1<R2.
```

```
composition((CONDITION1 or CONDITION2) and  
CONDITION3,CONCLUSION,LIST4) ←  
composition(CONDITION1,CONCLUSION,LIST1),composition(CONDITION2,CONCLU  
SION,LIST2),composition(CONDITION3,CONCLUSION,LIST3),sum(LIST1,R1),sum  
(LIST2,R2),R1>=R2,append(LIST1,LIST3,LIST4).
```

```
composition((CONDITION1 or CONDITION2) and  
CONDITION3, CONCLUSION, LIST4) ←  
composition(CONDITION1, CONCLUSION, LIST1), composition(CONDITION2, CONCLU  
SION, LIST2), composition(CONDITION3, CONCLUSION, LIST3), sum(LIST1, R1), sum  
(LIST2, R2), R1 < R2, append(LIST2, LIST3, LIST4).
```

Programa 17 – Cláusulas do predicado `composition` que lidam com condições compostas.

De seguida, coloca-se as cláusulas de `composition` que lidam com os operadores ">", "<", ">=", "<=". Estas cláusulas são específicas para cada parâmetro de saúde, sendo necessário defini-las para todas as condições que comparam valores numéricos. Neste caso, apresentam-se as cláusulas para o predicado `ten_year_risk`, como exemplo.

```
composition(ten_year_risk(X) < ten_year_risk(B), CONCLUSION, LIST1) ←  
fact(ten_year_risk(C)), C \== unknown, C < B,  
not(abducible(ten_year_risk(D))), assert(qoi(ten_year_risk(C), 1, CONCLUS  
ION)), LIST1=[qoi(ten_year_risk(C), 1)].
```

```
composition(ten_year_risk(X) <= ten_year_risk(B), CONCLUSION, LIST1) ←  
fact(ten_year_risk(C)), C \== unknown, C <= B,  
not(abducible(ten_year_risk(D))),  
assert(qoi(ten_year_risk(C), 1, CONCLUSION)),  
LIST1=[qoi(ten_year_risk(C), 1)].
```

```
composition(ten_year_risk(X) > ten_year_risk(B), CONCLUSION, LIST1) ←  
fact(ten_year_risk(C)), C \== unknown, C > B,  
not(abducible(ten_year_risk(D))),  
assert(qoi(ten_year_risk(C), 1, CONCLUSION)),  
LIST1=[qoi(ten_year_risk(C), 1)].
```

```
composition(ten_year_risk(X) >= ten_year_risk(B), CONCLUSION, LIST1) ←  
fact(ten_year_risk(C)), C \== unknown, C >= B,  
not(abducible(ten_year_risk(D))),  
assert(qoi(ten_year_risk(C), 1, CONCLUSION)),  
LIST1=[qoi(ten_year_risk(C), 1)].
```

```
composition(ten_year_risk(X) < ten_year_risk(B),CONCLUSION,LIST1)
←fact(ten_year_risk(D)),abducible(ten_year_risk(E)),
assert(qoi(ten_year_risk(B),0,CONCLUSION)),
LIST1=[qoi(ten_year_risk(B),0)].
```

```
composition(ten_year_risk(X) =< ten_year_risk(B),CONCLUSION,LIST1) ←
fact(ten_year_risk(D)),abducible(ten_year_risk(E)),
assert(qoi(ten_year_risk(B),0,CONCLUSION)),
LIST1=[qoi(ten_year_risk(B),0)].
```

```
composition(ten_year_risk(X) > ten_year_risk(B),CONCLUSION,LIST1) ←
fact(ten_year_risk(D)),abducible(ten_year_risk(E)),
assert(qoi(ten_year_risk(B),0,CONCLUSION)),
LIST1=[qoi(ten_year_risk(B),0)].
```

```
composition(ten_year_risk(X) >= ten_year_risk(B),CONCLUSION,LIST1) ←
fact(ten_year_risk(D)),abducible(ten_year_risk(E)),
assert(qoi(ten_year_risk(B),0,CONCLUSION)),
LIST1=[qoi(ten_year_risk(B),0)].
```

```
composition(ten_year_risk(X) < ten_year_risk(B),CONCLUSION,LIST1) ←
abducible(ten_year_risk(C)),C<B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=0,qoi_calculation_2(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) =< ten_year_risk(B),CONCLUSION,LIST1) ←
abducible(ten_year_risk(C)),C=<B,
not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=0,qoi_calculation_2(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) > ten_year_risk(B),CONCLUSION,LIST1)←
abducible(ten_year_risk(C)),C>B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=0,qoi_calculation_2(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) >= ten_year_risk(B),CONCLUSION,LIST1) ←
←abducible(ten_year_risk(C)),C>=B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=0,qoi_calculation_2(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) < ten_year_risk(B),CONCLUSION,LIST1) ←
abducible(ten_year_risk(C)),C<B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=1,qoi_calculation_1(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) =< ten_year_risk(B),CONCLUSION,LIST1) ←
abducible(ten_year_risk(C)),C=<B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=1,qoi_calculation_1(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) > ten_year_risk(B),CONCLUSION,LIST1) ←
abducible(ten_year_risk(C)),C>B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=1,qoi_calculation_1(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

```
composition(ten_year_risk(X) >= ten_year_risk(B),CONCLUSION,LIST1)
←abducible(ten_year_risk(C)),C>=B,not(fact(ten_year_risk(D))),
findall(ten_year_risk(E),(+ten_year_risk(E)::INVARIANT),Z),length(Z,N)
,N=1,qoi_calculation_1(V,ten_year_risk),
assert(qoi(ten_year_risk(C),V,CONCLUSION)),
LIST1=[qoi(ten_year_risk(C),V)].
```

Programa 18 – Cláusulas do predicado `composition` que lidam com os operadores ">", "<", ">=", "<=" para o predicado `ten_year_risk`.

O cálculo da pontuação de uma conclusão realiza-se através do predicado `score`, representado no Programa 19.

```
score(LIST,SCORE) ← sum(LIST,R),length(LIST,N),  
SCORE is R/N.
```

Programa 19 – Predicado `score` que calcula a pontuação de uma conclusão.

A ordenação por ordem decrescente de pontuação realiza-se através do predicado `conclusions`.

```
conclusions(L2) ←  
findall(conclusion(A,B),conclusion(A,B),Z),mergesort(Z,D),  
reverse(D,L1),remove_dups(L1,L2).
```

Programa 20 – Predicado `conclusions` que cria uma lista de conclusões ordenadas por ordem decrescente de pontuação.

Anexo B - XML Schema do Modelo de CIGs Proposto

O Programa 21 apresenta uma proposta de um XML *schema* para o modelo de CIGs proposto, em que se define as diferentes classes de tarefas e respectivos atributos.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Duration">
    <xs:sequence>
      <xs:element name="Value" type="xs:double" />
      <xs:element name="Unit">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="minute" />
            <xs:enumeration value="hour" />
            <xs:enumeration value="day" />
            <xs:enumeration value="week" />
            <xs:enumeration value="month" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Periodicity">
    <xs:sequence>
      <xs:element name="Value" type="xs:double" />
      <xs:element name="Unit">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="minute" />
            <xs:enumeration value="hour" />
            <xs:enumeration value="day" />
            <xs:enumeration value="week" />
            <xs:enumeration value="month" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Action">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" />
      <xs:element name="Caption" type="xs:string" />
      <xs:element name="Description" type="xs:string" />
      <xs:element name="Triggercondition" type="xs:string"
minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="Method" type="xs:string" minOccurs="0"
/>
      <xs:element name="Confirmation">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Yes" />
            <xs:enumeration value="No" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        </xs:element>
        <xs:element name="Duration" type="Duration" minOccurs="0"
/>
        <xs:element name="Periodicity" type="Periodicity"
minOccurs="0" />
        <xs:element name="Repetitions" type="xs:integer"
minOccurs="0" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Decision">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Caption" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="Triggercondition" type="xs:string"
minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="Candidate" type="xs:string"
minOccurs="2" maxOccurs="unbounded" />
        <xs:element name="Argumentrule" type="xs:string"
minOccurs="2" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Enquiry">
    <xs:sequence>
        <xs:element name="Name" type="xs:string" />
        <xs:element name="Caption" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="Triggercondition" type="xs:string"
minOccurs="0" maxOccurs="unbounded" />
        <xs:element name="Method" type="xs:string" minOccurs="0"
/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Components">
    <xs:sequence>
        <xs:element name="Plan" type="Plan" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="Decision" type="Decision" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="Action" type="Action" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="Enquiry" type="Enquiry" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Plans">
    <xs:sequence>
        <xs:element name="Plan" type="Plan" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Roottask">
    <xs:sequence>
        <xs:element name="Name" />
        <xs:element name="Caption" type="xs:string" />
        <xs:element name="Description" type="xs:string" />
        <xs:element name="Plans" type="Plans" />
        <xs:element name="Goal" type="xs:string" minOccurs="0" />
    </xs:sequence>

```



```

</xs:complexType>
<xs:complexType name="Plan">
  <xs:sequence>
    <xs:element name="Name" type="xs:string" />
    <xs:element name="Caption" type="xs:string" />
    <xs:element name="Description" type="xs:string" />
    <xs:element name="Author" type="xs:string" />
    <xs:element name="Date" type="xs:date" />
    <xs:element name="Version" type="xs:double" />
    <xs:element name="Components" type="Components" />
    <xs:element name="Schedulingconstraints" type="xs:string"
  />
    <xs:element name="Triggercondition" type="xs:string"
minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="Duration" type="Duration" minOccurs="0"
  />
    <xs:element name="Periodicity" type="Periodicity"
minOccurs="0" />
    <xs:element name="Repetitions" type="xs:integer"
minOccurs="0" />
    <xs:element name="Goal" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

Programa 21 – XML *schema* do modelo de CIGs proposto.

Anexo C – Guia para Detecção e Tratamento de Níveis Elevados de Colesterol

As Figuras 34 a 39 representam a guia que foi utilizada para elaborar o caso de estudo. Esta guia foi desenvolvida pelo NHLBI e possui a designação de Detection and Treatment of High Blood Cholesterol in Adults (NHLBI, 2002).

National Cholesterol Education Program

ATP III Guidelines At-A-Glance Quick Desk Reference

1

Step 1 Determine lipoprotein levels—obtain complete lipoprotein profile after 9- to 12-hour fast.

ATP III Classification of LDL, Total, and HDL Cholesterol (mg/dL)

LDL Cholesterol – Primary Target of Therapy	
<100	Optimal
100-129	Near optimal/above optimal
130-159	Borderline high
160-189	High
≥190	Very high
Total Cholesterol	
<200	Desirable
200-239	Borderline high
≥240	High
HDL Cholesterol	
<40	Low
≥60	High

2

Step 2 Identify presence of clinical atherosclerotic disease that confers high risk for coronary heart disease (CHD) events (CHD risk equivalent):

- Clinical CHD
- Symptomatic carotid artery disease
- Peripheral arterial disease
- Abdominal aortic aneurysm.

3

Step 3 Determine presence of major risk factors (other than LDL):

Major Risk Factors (Exclusive of LDL Cholesterol) That Modify LDL Goals

- Cigarette smoking
- Hypertension (BP ≥140/90 mmHg or on antihypertensive medication)
- Low HDL cholesterol (<40 mg/dL)*
- Family history of premature CHD (CHD in male first degree relative <55 years; CHD in female first degree relative <65 years)
- Age (men ≥45 years; women ≥55 years)

* HDL cholesterol ≥60 mg/dL counts as a "negative" risk factor; its presence removes one risk factor from the total count.

- Note: in ATP III, diabetes is regarded as a CHD risk equivalent.

NATIONAL INSTITUTES OF HEALTH
NATIONAL HEART, LUNG, AND BLOOD INSTITUTE




Figura 34 – Primeira página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).

4

Step 4 If 2+ risk factors (other than LDL) are present without CHD or CHD risk equivalent, assess 10-year (short-term) CHD risk (see Framingham tables).
Three levels of 10-year risk:

- >20% — CHD risk equivalent
- 10-20%
- <10%

5

Step 5 Determine risk category:

- Establish LDL goal of therapy
- Determine need for therapeutic lifestyle changes (TLC)
- Determine level for drug consideration

LDL Cholesterol Goals and Cutpoints for Therapeutic Lifestyle Changes (TLC) and Drug Therapy in Different Risk Categories.

Risk Category	LDL Goal	LDL Level at Which to Initiate Therapeutic Lifestyle Changes (TLC)	LDL Level at Which to Consider Drug Therapy
CHD or CHD Risk Equivalents (10-year risk >20%)	<100 mg/dL	≥100 mg/dL	≥130 mg/dL (100-129 mg/dL: drug optional)*
2+ Risk Factors (10-year risk ≤20%)	<130 mg/dL	≥130 mg/dL	10-year risk 10-20%: ≥130 mg/dL 10-year risk <10%: ≥160 mg/dL
0-1 Risk Factor†	<160 mg/dL	≥160 mg/dL	≥190 mg/dL (160-189 mg/dL: LDL-lowering drug optional)

* Some authorities recommend use of LDL-lowering drugs in this category if an LDL cholesterol <100 mg/dL cannot be achieved by therapeutic lifestyle changes. Others prefer use of drugs that primarily modify triglycerides and HDL, e.g., nicotinic acid or fibrates. Clinical judgment also may call for differing drug therapy in this subcategory.

† Almost all people with 0-1 risk factor or have a 10-year risk <10%, thus 10-year risk assessment in people with 0-1 risk factor is not necessary.

6

Step 6 Initiate therapeutic lifestyle changes (TLC) if LDL is above goal.

TLC Features

- TLC Diet:
 - Saturated fat <7% of calories, cholesterol <200 mg/day
 - Consider increased viscous (soluble) fiber (10-25 g/day) and plant stanols/sterols (2g/day) as therapeutic options to enhance LDL lowering
- Weight management
- Increased physical activity

Figura 35 – Segunda página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).

7

Step 7

Consider adding drug therapy if LDL exceeds levels shown in Step 5 table:

- Consider drug simultaneously with TLC for CHD and CHD equivalents
- Consider adding drug to TLC after 3 months for other risk categories.

Drugs Affecting Lipoprotein Metabolism

Drug Class	Agents and Daily Doses	Lipid/Lipoprotein Effects	Side Effects	Contraindications
HMG CoA reductase inhibitors (statins)	Lovastatin (20-80 mg)	LDL ↓18-55%	Myopathy Increased liver enzymes	Absolute: • Active or chronic liver disease Relative: • Concomitant use of certain drugs*
	Pravastatin (20-40 mg)	HDL ↑5-15%		
	Simvastatin (20-80 mg)	TC ↓7-30%		
	Fluvastatin (20-80 mg)			
	Atorvastatin (10-80 mg) Rosuvastatin (0.4-0.8 mg)			
Bile acid sequestrants	Cholestyramine (4-16 g)	LDL ↓15-30%	Gastrointestinal distress Constipation Decreased absorption of other drugs	Absolute: • dysbacteriosis • lipoproteinemia • TG >400 mg/dl Relative: • TG >200 mg/dl
	Colestipol (5-20 g)	HDL ↑3-5%		
	Coesevelam (2.6-3.8 g)	TC No change or increase		
Nicotinic acid	Immediate release (crystalline) nicotinic acid (1.5-3 gm), extended release nicotinic acid (Niaspan®) (1-2 g), sustained release nicotinic acid (1-2 g)	LDL ↓5-25% HDL ↑15-35% TC ↓20-50%	Flushing Hyperglycemia Hyperuricemia (or gout) Upper GI distress Hepatotoxicity	Absolute: • Chronic liver disease • Severe gout Relative: • Diabetes • Hyperuricemia • Peptic ulcer disease
	Fibric acids	Gemfibrozil (600 mg BID) Fenofibrate (200 mg) Clofibrate (1000 mg BID)		

* Cyclosporine, macrolide antibiotics, various anti-fungal agents, and cytochrome P-450 inhibitors (fibrates and niacin) should be used with appropriate caution.

Figura 36 – Terceira página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).



Step 8

Identify metabolic syndrome and treat, if present, after 3 months of TLC.

Clinical Identification of the Metabolic Syndrome – Any 3 of the Following:

Risk Factor	Defining Level
Abdominal obesity*	Waist circumference [†]
Men	>102 cm (>40 in)
Women	>88 cm (>35 in)
Triglycerides	≥150 mg/dL
HDL cholesterol	
Men	<40 mg/dL
Women	<50 mg/dL
Blood pressure	≥130/≥85 mmHg
Fasting glucose	≥110 mg/dL

* Overweight and obesity are associated with insulin resistance and the metabolic syndrome. However, the presence of abdominal obesity is more highly correlated with the metabolic risk factors than is an elevated body mass index (BMI). Therefore, the simple measure of waist circumference is recommended to identify the bodyweight component of the metabolic syndrome.
 † Some male patients can develop multiple metabolic risk factors when the waist circumference is only marginally increased, e.g., 94–102 cm (37–39 in). Such patients may have a strong genetic contribution to insulin resistance. They should benefit from changes in life habits, similarly to men with categorical increases in waist circumference.

Treatment of the metabolic syndrome

- Treat underlying causes (overweight/obesity and physical inactivity):
 - Intensify weight management
 - Increase physical activity.
- Treat lipid and non-lipid risk factors if they persist despite these lifestyle therapies:
 - Treat hypertension
 - Use aspirin for CHD patients to reduce prothrombotic state
 - Treat elevated triglycerides and/or low HDL (as shown in Step 9).

Figura 37 – Quarta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).

9

Step 9 **Treat elevated triglycerides.**

ATP III Classification of Serum Triglycerides (mg/dL)

<150	Normal
150-199	Borderline high
200-499	High
≥500	Very high

Treatment of elevated triglycerides (≥150 mg/dL)

- Primary aim of therapy is to reach LDL goal
- Intensify weight management
- Increase physical activity
- If triglycerides are ≥200 mg/dL after LDL goal is reached, set secondary goal for non-HDL cholesterol (total - HDL) 30 mg/dL higher than LDL goal.

Comparison of LDL Cholesterol and Non-HDL Cholesterol Goals for Three Risk Categories

Risk Category	LDL Goal (mg/dL)	Non-HDL Goal (mg/dL)
CHD and CHD Risk Equivalent (10-year risk for CHD >20%)	<100	<130
Multiple (2+) Risk Factors and 10-year risk ≤20%	<130	<160
0-1 Risk Factor	<160	<190

If triglycerides 200-499 mg/dL after LDL goal is reached, consider adding drug if needed to reach non-HDL goal:

- intensify therapy with LDL-lowering drug, or
- add nicotinic acid or fibrate to further lower VLDL.

If triglycerides ≥500 mg/dL, first lower triglycerides to prevent pancreatitis:

- very low-fat diet (≤15% of calories from fat)
- weight management and physical activity
- fibrate or nicotinic acid
- when triglycerides <500 mg/dL, turn to LDL-lowering therapy.

Treatment of low HDL cholesterol (<40 mg/dL)

- First reach LDL goal, then:
- Intensify weight management and increase physical activity
- If triglycerides 200-499 mg/dL, achieve non-HDL goal
- If triglycerides <200 mg/dL (isolated low HDL) in CHD or CHD equivalent consider nicotinic acid or fibrate.

Figura 38 – Quinta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).

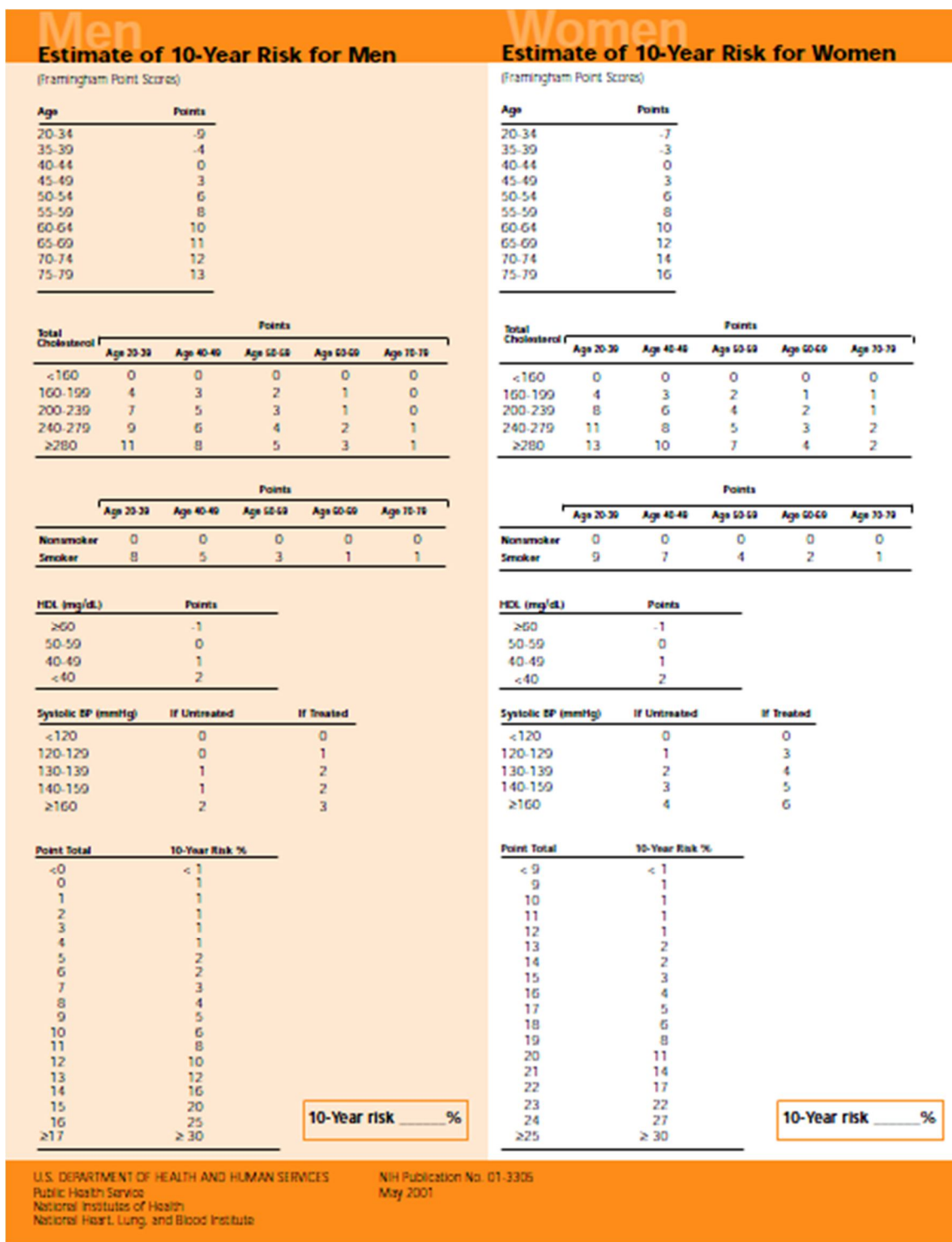


Figura 39 – Sexta página da guia Detection and Treatment of High Blood Cholesterol in Adults (reproduzido de (NHLBI, 2002)).

Anexo D – *Plans* da Guia para Detecção e Tratamento de Elevados Níveis de Colesterol

No Anexo D representam-se os planos da guia para detecção e tratamento de elevados níveis de colesterol que, por apresentarem semelhanças com outros planos, não foram incluídos no corpo do texto.

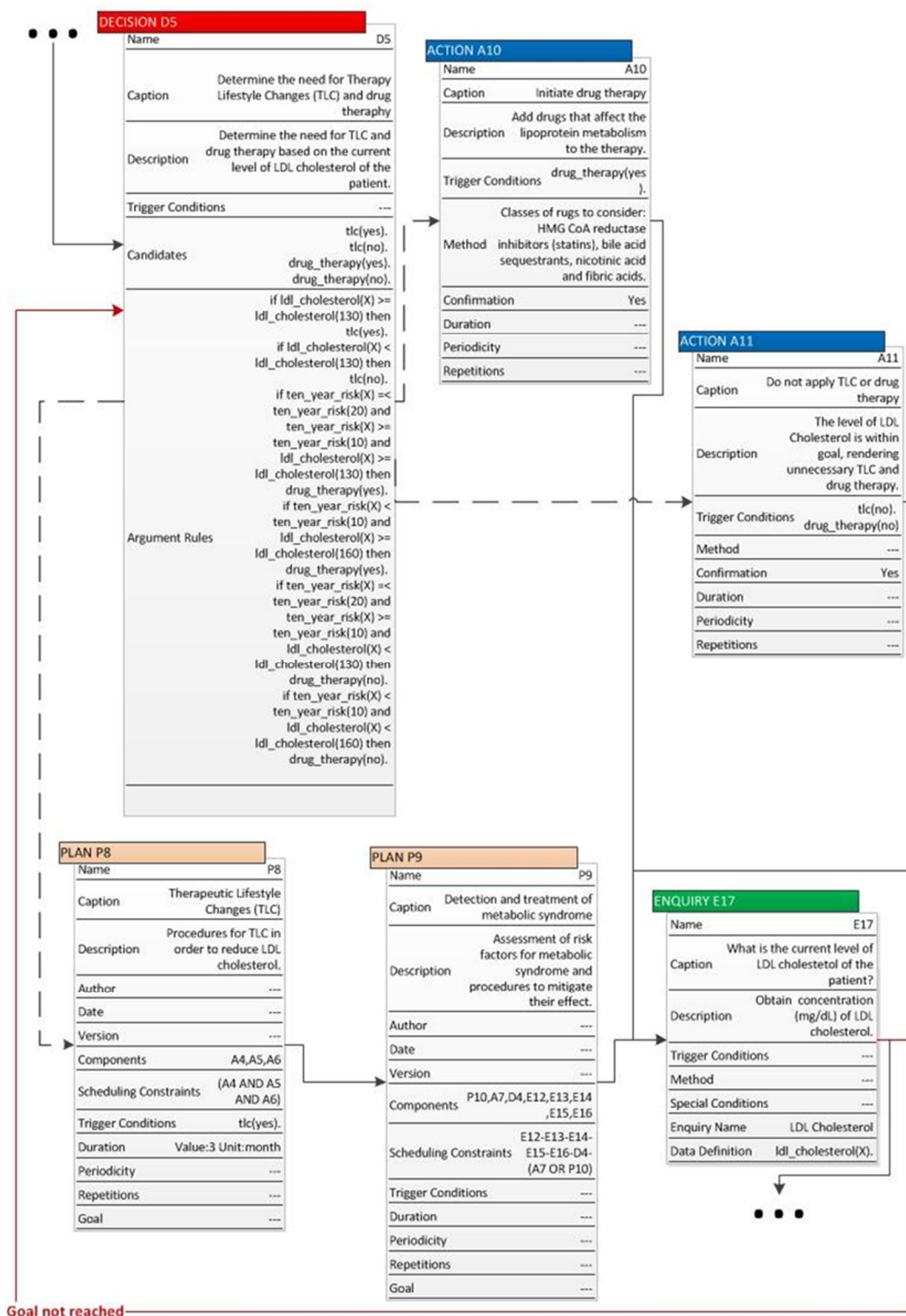


Figura 40 – Representação do *plan* P5.

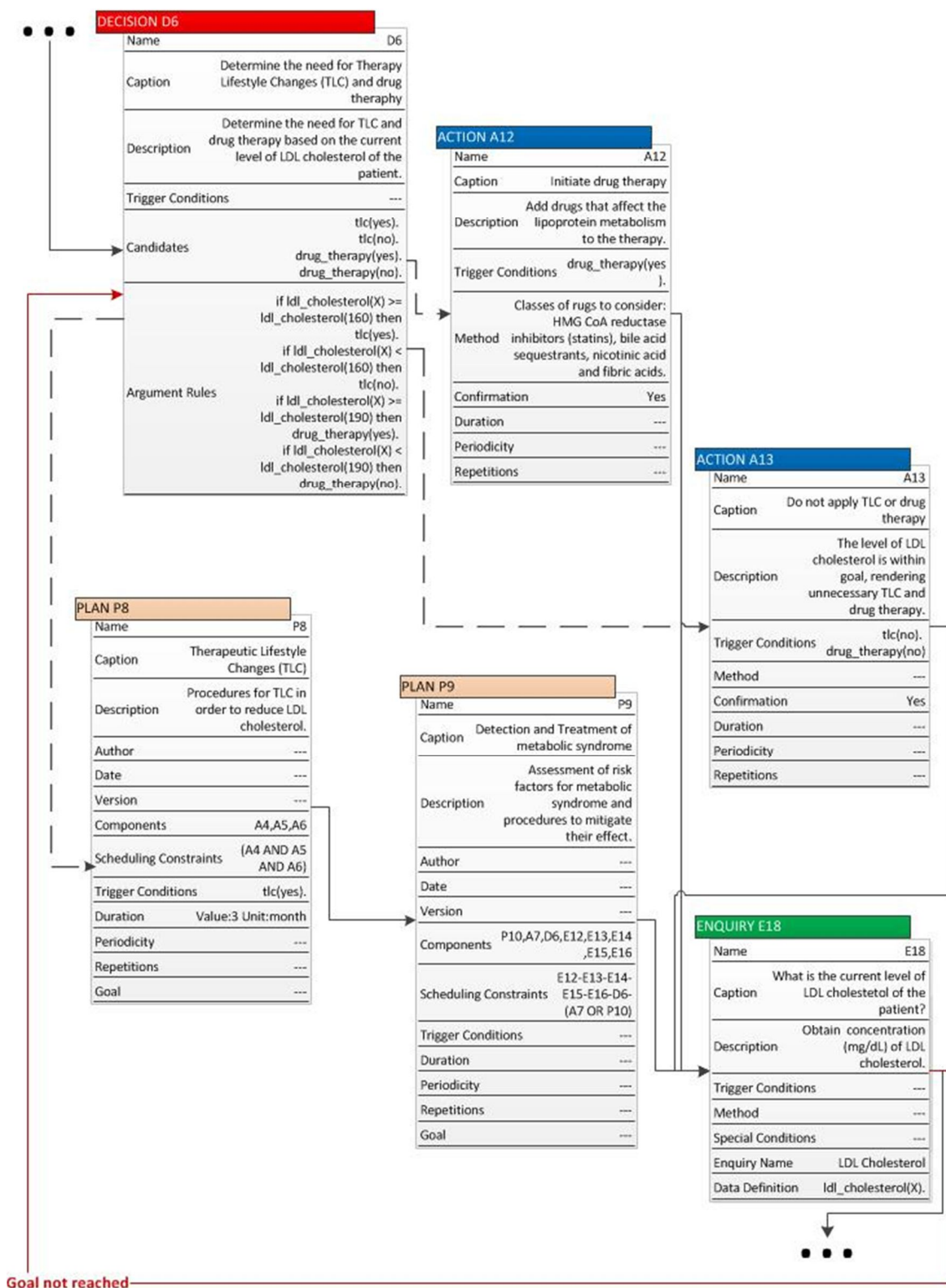


Figura 41 – Representação do *plan* P6.

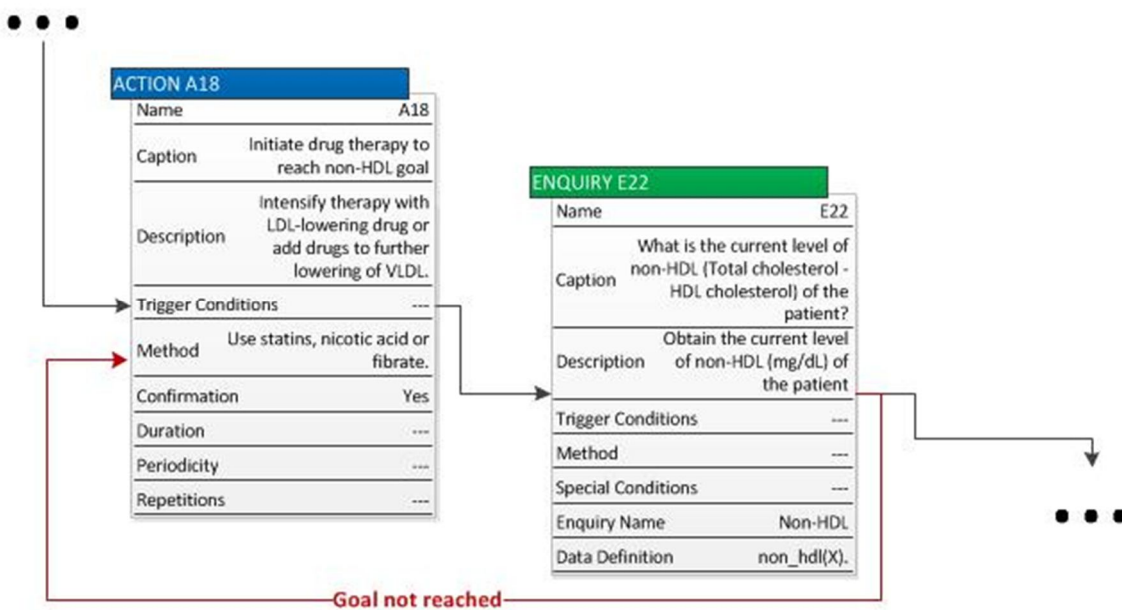


Figura 42 – Representação do *plan* P13.

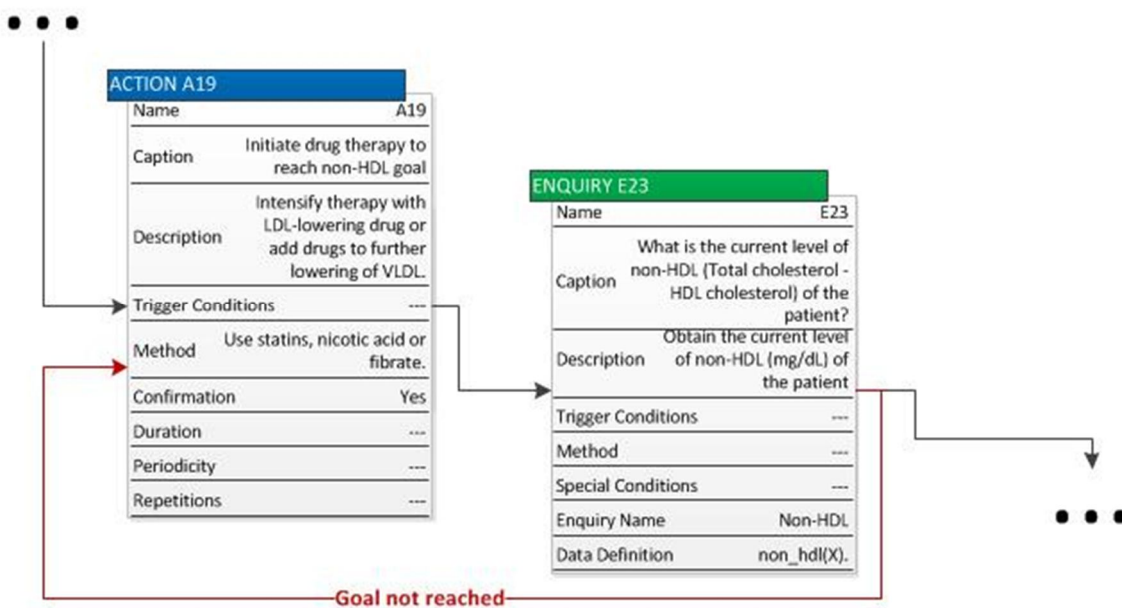


Figura 43 – Representação do *plan* P14.