

ST-Modeler:
Um Ambiente de Desenvolvimento e Verificação de
Guias de Conduta Clínica

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Márcio Paixão Dantas e aprovada pela Banca Examinadora.

Campinas, 30 de junho de 2008.



Jacques Wainer (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Dantas, Márcio Paixão

D235s ST-Modeler: um ambiente de desenvolvimento e verificação de guias de conduta clínica /Márcio Paixão Dantas -- Campinas, [S.P. :s.n.], 2008.

Orientador : Jacques Wainer

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Inteligência artificial – Aplicações médicas. 2. Representação do conhecimento (Sistemas especialistas). 3. Modelagem. 4. Verificação (Lógica). I. Wainer, Jacques. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: ST-Modeler: a clinical guidelines development and verification environment

Palavras-chave em inglês (Keywords): 1. Artificial intelligence – Medical applications. 2. Knowledge representation (Expert systems). 3. Modelling. 4. Logical verification.

Área de concentração: Sistemas de Informação
Inteligência artificial em medicina

Titulação: Mestre em Computação

Banca examinadora: Prof. Dr. Jacques Wainer (IC-Unicamp)
Prof. Dr. Ivan Torres Pisa (Unifesp)
Profª. Dra. Ariadne Maria Brito Rizzoni Carvalho (IC-Unicamp)

Data da defesa: 30/06/2008

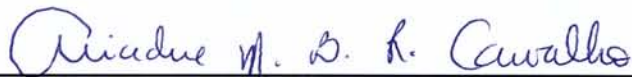
Programa de Pós-Graduação: Mestrado em Computação

TERMO DE APROVAÇÃO

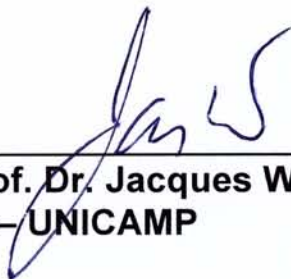
Dissertação Defendida e Aprovada em 30 de junho de 2008, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Ivan Torres Pisa
DIS - UNIFESP



Prof^a. Dr^a. Ariadne Maria Brito Rizzoni Carvalho
IC – UNICAMP



Prof. Dr. Jacques Wainer
IC – UNICAMP

ST-Modeler: Um Ambiente de Desenvolvimento e Verificação de Guias de Conduta Clínica

Márcio Paixão Dantas¹

Agosto de 2008

Banca Examinadora:

- Jacques Wainer (Orientador)
- Ivan Torres Pisa
DIS - UNIFESP
- Ariadne Maria Brito Rizzoni Carvalho
IC-UNICAMP
- Siome Klein Goldenstein (Suplente)
IC-UNICAMP

¹Suporte financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Em memória de meu avô José Alves Paixão, vovó Tosca, e tias Geralda, Zuleika e Lourdes D'Elorenzo.

Resumo

Um guia de conduta médica é um documento formado por um conjunto de recomendações sistematicamente desenvolvidas para auxiliar nas decisões de praticantes e pacientes sobre cuidados de saúde em circunstâncias específicas. Desde que sejam bem elaborados, comprovadamente melhoram a qualidade média do atendimento médico.

Frequentemente médicos não estão acostumados com guias práticos escritos em papel e não os aplicam adequadamente. Implementados em sistemas computacionais podem melhorar significativamente a qualidade do atendimento médico diretamente no local da assistência.

Este trabalho está ligado ao projeto ST-Guide, uma iniciativa surgida no Instituto de Computação da Universidade Estadual de Campinas (IC-UNICAMP) para resolver o problema de formalização e implementação de guias clínicos práticos. As contribuições realizadas foram: (a) reestruturação do projeto em outras tecnologias; (b) estabelecimento e implementação de uma arquitetura extensível para suportar modelagem e implementação de guias; (c) realização de alterações conceituais para facilitar o processo de modelagem; (d) criação de novo algoritmo para verificação lógica de incompletude e ambiguidade; e (e) criação de um ambiente gráfico de desenvolvimento de guias associado a um verificador lógico que atua em tempo real, isto é, à medida em que ocorre a modelagem.

Experimentos baseados em guias clínicos para assistência pré-natal e hipertensão são apresentados e o texto é concluído com uma discussão do que foi realizado e sugestão de trabalhos futuros.

Abstract

Clinical guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances. Since well elaborated, they provenly improve medical assistance quality.

Oftenly clinicians aren't used to paper written guidelines and don't apply them adequately. Implemented in computer systems guidelines can significantly improve medical care quality directly at the point-of-care.

This work is related to ST-Guide project, an initiative arisen at the Institute of Computing of State University of Campinas (IC-UNICAMP) to solve the practice guideline formalization and implementation problems. The contributions done were: (a) project restructuring on other technologies; (b) extensible architecture specification and implementation to support guideline modelling and implementation; (c) conceptual changes to ease modelling; (d) new incompletude and ambiguity logical verification algorithm; and (e) guideline graphical development environment coupled with a real time logical verifier.

Experiments based on pre-natal assistance and hypertension clinical guidelines are presented and the text is concluded with a discussion of what was done and future work suggestions.

Agradecimentos

Senhor, obrigado por ter me concedido mais esta oportunidade de aprendizado e superação.

Pai e mãe, novamente sem vocês não teria sido possível.

Professores Antonio Tavares da Costa Jr e Iraziet Charret, a seriedade, dedicação e amor de vocês pelo ensino e ciência constituem minha referência. Sem vocês eu não teria acreditado.

Professor Jacques Wainer, sempre que precisei tive o seu total apoio e confiança, além do bom humor.

Professor Julio López, ser seu monitor foi uma grande honra. Nossas longas conversas sobre ensino de programação, criptografia e assuntos aleatórios já deixam muitas saudades.

Flávio, Celso, Richard, Renato (Tchê), Mário, Rafael (Mamão), Renato (Chenca), Raoni, Anibal, Fernando, Carlos e Gustavo, Campinas foi minha casa graças a vocês.

Colegas do IC, em especial ao Cléo Billa, que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

Gostaria também de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), que desde a graduação vem me incentivando com bolsas de pesquisa.

Obrigado a todos!!

Sumário

Resumo	vi
Abstract	vii
Agradecimentos	viii
1 Introdução	1
2 Conceitos Básicos	3
2.1 Guias Informais e Formais	3
2.2 Formalização	4
2.3 Qualidade em guias clínicos práticos	5
2.4 Projetos centrados em documento	7
2.4.1 Medical Guideline Technology (MGT)	7
2.4.2 Guideline Element Model (GEM)	8
2.5 Projetos centrados em modelo	8
2.5.1 The Arden Syntax	9
2.5.2 GuideLine Interchange Format(GLIF)	10
2.5.3 PROforma	11
2.5.4 Asbru	12
2.5.5 EON	13
2.5.6 Prodigy	13
3 ST-Guide	15
3.1 Fundamentos	15
3.2 ST-Linguagem	18
3.2.1 Mudanças	18
3.3 ST-Sistema	21
3.3.1 Representação interna do paciente	22
3.3.2 Arquitetura	23

4	Verificação Lógica	34
4.1	Revisão	34
4.2	Verificação no ST-Guide	35
5	ST-Modeler	37
5.1	Linguagem Visual	37
5.2	Usando o ST-Modeler	39
5.2.1	Estado Inicial	39
5.2.2	Criar/Copiar/Remover um estado	39
5.2.3	Editar ST-State e Visualizar sua Situação	39
5.2.4	Modelando Literais	41
5.2.5	Configurando Transições e Relações de Uso	50
5.3	Experimentos	53
5.3.1	Assistência Pré-Natal	53
5.3.2	Hipertensão Arterial	57
5.4	Considerações sobre Usabilidade	61
6	Discussão, Conclusões e Futuros Trabalhos	63
	Bibliografia	65
A	Guia para Assistência Pré-Natal	70
B	Gramática da ST-Linguagem em XML	79
C	Modelo Conceitual da Representação Interna de Paciente	90
D	Gravidez de Alto Risco	94
E	Exercícios do experimento de usabilidade	97

Lista de Tabelas

2.1	Níveis de especificação no GLIF e características do modelo gerado	10
2.2	Segundo nível da hierarquia de tarefas do <i>PROforma</i>	12
3.1	Exemplo de operadores implementados no ST-Sistema. “Op.” usado como diminutivo de operando.	27
3.2	Operandos implementados no ST-Sistema.	28
3.3	Descrição dos métodos da interface <i>Operand</i>	29
5.1	Palavras-chave usadas nas condições de transições e relações de uso.	50
5.2	Rotina laboratorial pré-natal.	53
5.3	Procedimentos médicos nas consultas pré-natal.	54
C.1	Modelo conceitual da representação interna de paciente do ST-Sistema.	90
C.2	Estado atual.	91
C.3	Medicação.	91
C.4	Exame Físico ou Laboratorial.	92
C.5	Pergunta e Resposta.	92
C.6	Lembrete.	92
C.7	Literal.	93
C.8	Operador.	93
C.9	Operando.	93
C.10	Transição e Relação de Uso.	93

Lista de Figuras

3.1	Exemplo de máquina de estados.	17
3.2	Modelo de consultas.	17
3.3	Fragmentos do esquema <i>XML</i> da ST-Linguagem: elementos <i>Guide</i> , <i>State</i> , <i>HelperState</i> e <i>Declarations</i>	19
3.4	Diagrama principal de classes do ST-Sistema.	25
3.5	Representação interna de paciente.	26
3.6	Interface <i>Operator</i> e classes que a implementam.	26
3.7	Interface <i>Operand</i> e classes que a implementam.	27
3.8	Processo de carregamento de guia por uma aplicação externa.	30
3.9	Detalhe do processo de carregamento de guia, “dentro” do método <i>MedicalGuideline.setStates(Element)</i> . Instanciação de operando por reflexão e configuração à partir de dados do guia.	31
3.10	Implementação de guia por uma aplicação externa. <i>EMR Mapper</i> representa o componente do sistema externo responsável por mapear o prontuário eletrônico médico do sistema externo para o interno do ST-Sistema.	32
3.11	Detalhamento do processo de implementação, método <i>process</i> de <i>MedicalGuideline</i>	33
4.1	Transições de um estado qualquer, S_0 , em função das condições c_i , $1 \leq i \leq n$	36
5.1	ST-Modeler: componentes do <i>plug-in</i> no Eclipse.	38
5.2	Linguagem Visual do ST-Modeler.	38
5.3	ST-Modeler: estado inicial e suas telas de configuração.	40
5.4	Menu ST-Modeler: Criando novo ST-State e Helper-State.	40
5.5	Interface de usuário para edição de ST-State.	42
5.6	Edição de ST-State: adicionar pergunta. Em destaque, circulados estão os comandos para inserir e remover elementos da lista de perguntas.	43
5.7	ST-Modeler: tela de adição de literal para um estado.	44
5.8	ST-Modeler: modelando o literal equivalente a <i>pressão diastólica do paciente maior que 90 mmHg</i>	46

5.9	ST-Modeler: modelando o literal equivalente a <i>Existe histórico pessoal ou familiar de hemofilia</i>	48
5.10	ST-Modeler: usando o operando <i>Age</i> em um literal.	49
5.11	Criando uma nova transição ou relação de uso no menu do ST-Modeler. . .	50
5.12	Tela de edição de transição ou relação de uso no ST-Modeler.	51
5.13	Exemplo de configuração de transição.	52
5.14	Modelo do tratamento de pré-natal: parte um.	55
5.15	Modelo do tratamento de pré-natal: parte dois.	56
5.16	Esboço de modelagem do estado <i>Ate 28a</i> de assistência pré-natal sem Helper-States.	58
5.17	Modelo do experimento de hipertensão: parte um.	60
5.18	Modelo do experimento de hipertensão: parte dois.	61

Capítulo 1

Introdução

Atualmente se escuta dizer com frequência que informação é o bem mais valioso que uma pessoa ou organização pode ter. Se a pessoa é um médico ou trabalha na área de saúde, fica ainda mais evidente a importância de possuir informação atualizada de forma rápida e fácil.

Segundo o *Institute of Medicine* (IOM), organismo importante de saúde norte-americano, um guia de conduta médica é um documento formado por um conjunto de recomendações sistematicamente desenvolvidas para auxiliar nas decisões de praticantes e pacientes sobre cuidados de saúde em circunstâncias específicas [1]. Um guia real para assistência pré-natal[2] está disponível para consulta no Apêndice A.

Existem muitos estudos que avaliam o efeito do uso de guias de conduta em tratamentos médicos, entre eles: [3], [4] e [5]. A avaliação, em geral, é bastante positiva. A grande maioria dos trabalhos atesta o ganho de qualidade média em tratamentos de saúde devido à adoção de guias práticos. Contudo, a razão decisiva para a implementação dos guias em hospitais e organizações de saúde é a economia substancial de recursos financeiros [5].

Um problema abordado em [6] é que se tem dado mais atenção ao desenvolvimento de guias de conduta do que à implementação destes na rotina diária de tratamento médico. [7] mostra um outro fator agravante: frequentemente médicos não estão acostumados com guias práticos escritos em papel e não os aplicam adequadamente. [8] sugere um crescimento exponencial no número de guias publicados na Inglaterra à partir de 1989, criticando duramente o grande volume de papel gerado e suas consequências: acesso moroso e dificuldade de uso das informações. Segundo [9], guias de conduta implementados em sistemas de suporte à tomada de decisão baseados em computador contribuem significativamente para melhorar a qualidade da prática clínica diária, resolvendo eficientemente o problema de uso inadequado e difícil dos guias de conduta em papel.

Muitos sistemas computacionais foram desenvolvidos tanto para modelagem como para execução de guias de conduta médica. Como ainda não há técnicas de processamento de

linguagem natural (PLN) capazes de processar e “entender” um texto com total confiabilidade, a maioria destes sistemas possui uma fase de aquisição manual de conhecimento. Isto significa que é preciso que uma pessoa transcreva as informações contidas no guia para uma linguagem interpretável por computador. Este processo também é conhecido como *formalização*. Outro fator que impossibilita a formalização automática de guias clínicos é a presença comum de erros textuais e lógicos no próprio texto.

Este trabalho está ligado ao projeto ST-Guide [10], uma iniciativa surgida no Instituto de Computação da Universidade Estadual de Campinas (IC-UNICAMP) para resolver o problema de formalização e implementação de guias clínicos práticos. As contribuições realizadas foram: (a) reestruturação do projeto em outras tecnologias; (b) estabelecimento e implementação de uma arquitetura extensível para suportar modelagem e implementação de guias; (c) realização de alterações conceituais para facilitar o processo de modelagem; (d) criação de novo algoritmo para verificação lógica de incompletude e ambiguidade; e (e) criação de um ambiente gráfico de desenvolvimento de guias associado a um verificador lógico que atua em tempo real, isto é, à medida em que ocorre a modelagem.

O texto está organizado como segue:

- **Capítulo 2: Conceitos Básicos.** Revisão bibliográfica sobre formalização, qualidade de guias e projetos já existentes.
- **Capítulo 3: ST-Guide.** Apresentação do ST-Guide, sua nova linguagem, representação interna de paciente, arquitetura e facilidades de modelagem.
- **Capítulo 4: Verificação Lógica.** Revisão sobre técnicas de verificação lógica aplicadas em guias clínicos na literatura e apresentação do algoritmo de verificação criado para o ST-Guide.
- **Capítulo 5: ST-Modeler.** Apresentação e introdução prática ao ST-Modeler, o ambiente de desenvolvimento e verificação de guias clínicos interpretáveis por computador do ST-Guide. Dois experimentos de modelagem referentes a um guia de assistência pré-natal[2] e a um de hipertensão[11] são comentados e, então, considerações sobre a usabilidade da ferramenta encerram o capítulo.
- **Capítulo 6: Discussão, Conclusões e Futuros Trabalhos.** Conclusão do trabalho discutindo seus resultados e limitações. Sugestão de novos trabalhos.

Capítulo 2

Conceitos Básicos

Neste capítulo o leitor será introduzido aos principais conceitos referentes a guias de conduta médica.

2.1 Guias Informais e Formais

Um guia de conduta médica em seu formato textual original costuma apresentar vários problemas ou **inconformidades**. Abaixo, estas são listadas:

1. *Ambiguidade*: condições do paciente podem levar à execução de mais de uma recomendação, sendo que não está claro no guia se era essa realmente a intenção dos seus autores.
2. *Incompletude*: o paciente apresenta condições para as quais o guia não especifica nenhuma recomendação.
3. *Inconsistência*: o guia recomenda ações conflitantes entre si ou com o estado do paciente. Duas ações conflitantes entre si podem ocorrer ou simultaneamente ou em instantes diferentes do tratamento.
4. *Abstração*: uso de termos cujo significado ou entendimento não pode ser encontrado no guia. É assumido pelos autores que o leitor conhece a terminologia usada. Muitas vezes, este tipo de inconformidade produz outros tipos, como ambiguidade.

Estas inconformidades podem dificultar bastante a implementação ou uso do guia. Diz-se que um guia de conduta em seu estado original é *informal*. Naturalmente, guias que não contêm orientações bem definidas e específicas têm a sua probabilidade de adesão por clínicos bastante diminuída [12].

O processo de transformar um guia em formato original para uma linguagem formal ou interpretável por computador é chamado de *formalização*. Formalizar um guia é o primeiro passo em ordem para implementá-lo computacionalmente. Esta nova representação ou modelo do guia original é dita *formal*. Observe que, devido a possíveis inconformidades no documento original, pode ser que um guia de conduta possua mais de uma representação formal. Além disso um guia, apesar de formal, ainda pode conter inconformidades.

Estudando a relação entre guias de conduta médica informais e formais, [13] faz uma análise baseada em dois casos de estudo. Nos dois casos de estudo, um mapeamento entre o documento original e o modelo formal são feitos. Entre as principais anomalias identificadas estão:

1. não-mapeamento de partes do modelo original;
2. não-mapeamento maior ainda de partes do modelo formal.

(1) significa que muitas vezes o modelo formal não contempla todas as informações contidas no guia original. (2) implica que para conseguir construir uma especificação formal, às vezes, é preciso inserir muita informação que não está presente no documento original.

Estas anomalias podem aparecer em menor ou maior grau, dependendo do guia original e da técnica de formalização adotada.

2.2 Formalização

Como alternativas para formalização de guias de conduta médica em linguagens interpretáveis por computador, surgiram basicamente duas abordagens [14]:

1. *Centrada em documento*: formalização é ascendente, isto é, usa o documento original como fonte direta (autêntica) de conhecimento;
2. *Centrada em modelo*: formalização é descendente. Um **modelo compacto** do guia é especificado por um especialista primeiro e, somente depois, fragmentos textuais do guia são associados com o modelo para fins de documentação.

O modelo formal resultante da abordagem (1) é um modelo que pode ser diretamente mapeado para o modelo informal. Já em (2), o modelo formal é decorrente da modelagem de um especialista e a relação entre os modelos informal e formal é indireta, isto é, o modelo formal é uma interpretação do informal.

A abordagem (1) oferece algumas vantagens [14]: (a) capaz de cobrir informações que não se encaixam bem no modelo compacto; (b) menor risco de deixar informações

importantes de fora do modelo formal. Porém, sua maior desvantagem é que apesar de resultar em um modelo formal, este é significativamente mais difícil de ser executado por uma aplicação [14]. Isso ocorre, em grande parte, porque como o modelo formal é extraído diretamente do texto original na abordagem (1), ele torna-se bastante vulnerável às inconformidades do guia informal. Já na abordagem (2), diante de um problema no texto, um especialista pode usar o seu conhecimento para inferir a melhor interpretação possível e impedir que o processo de implementação computacional falhe.

2.3 Qualidade em guias clínicos práticos

No final da seção 2.2, começou-se uma discussão sobre como problemas na elaboração do guia afetam a sua implementação e adesão. Sabe-se que guias mal escritos têm sua taxa de adesão por clínicos bastante diminuída [12] e que a variação de qualidade entre os guias publicados é alta [15], apesar da existência de guias metodológicos com o propósito de orientar os desenvolvedores ([1] e [16], entre outros).

Segundo o IOM, um guia clínico prático deve possuir oito atributos desejáveis:

- validade;
- confiabilidade e reprodutibilidade;
- aplicabilidade clínica;
- flexibilidade clínica;
- clareza;
- documentação;
- desenvolvimento por um processo multidisciplinar;
- e planos de revisão.

Estes atributos são características gerais desejáveis em um guia. [17], continuando o trabalho de [12], faz um trabalho mais específico e investiga dentro de guias que atributos têm as recomendações mais seguidas por clínicos e, ao mesmo tempo, que atributos têm as recomendações menos seguidas. Os possíveis atributos de uma recomendação considerados foram:

1. suportado por evidência científica*;
2. suportado por discussão de benefícios;

3. suportado por discussão de malefícios;
4. fácil de seguir*;
5. efeitos podem ser vistos rapidamente;
6. compatível com normas e valores*;
7. parte de árvore de decisão complexa*;
8. requer conhecimento novo;
9. requer novas habilidades*;
10. demanda mudanças na organização;
11. demanda mudança nas rotinas;
12. evoca reações negativas nos pacientes.

Os resultados de [17] caracterizam estatisticamente os atributos 1 à 6 como facilitadores para adoção de uma recomendação e os demais como barreiras. Os atributos marcados com * são os que apresentaram maior correlação positiva ou negativa com adoção.

Na prática, sabe-se que guias metodológicos para desenvolvimento de guias práticos não têm sido seguidos em geral. [18] faz uma revisão estruturada de todos os guias identificados pela ferramenta de busca MEDLINE publicados entre 1985 e 1997. Os resultados obtidos, na maioria dos quesitos, mostram que mais da metade dos guias analisados não seguem as diretrizes de desenvolvimento que deveriam. Ainda em [18], uma análise temporal mostrou melhora tímida na porcentagem de guias dentro das diretrizes de desenvolvimento em um período de mais de 10 anos.

Para lidar com este problema, alguns trabalhos, como [15], têm surgido visando a um ganho real maior na qualidade e facilidade de implementação computacional de guias práticos. A estratégia usada é a padronização do processo usando especificações mais objetivas e menos abstratas.

Cada país, geralmente, regulamenta os seus próprios padrões de desenvolvimento, distribuição e implementação. No entanto, existe uma tendência para que estes padrões convirjam ou que ao menos haja uma troca maior de experiências na comunidade internacional. Por exemplo, em dezembro de 2002, foi criado o *Guidelines International Network* (G-I-N), uma associação internacional sem fins lucrativos formada por organizações e indivíduos envolvidos no desenvolvimento de guias clínicos práticos. Atualmente o G-I-N possui como membros 63 organizações de 33 diferentes países espalhados por quatro continentes.

Melhorar os padrões de desenvolvimento e fazer com que os comitês desenvolvedores realmente os sigam é fundamental para a minimização de inconformidades nos guias. Ao mesmo tempo, guias com recomendações fáceis de seguir, baseadas em evidência científica forte e compatíveis com as normas e valores clínicos atuais terão maior adesão por clínicos.

2.4 Projetos centrados em documento

O número de projetos que usam formalização centrada em documento é bem menor que o de centrados em modelo. A principal razão, mencionada anteriormente, é a dificuldade de se utilizar, de fato, o modelo formal obtido.

A estratégia básica dos projetos que têm usado esta abordagem é mapear o documento original para um modelo formal hierárquico composto por um conjunto pré-definido de tipos hierárquicos. Em geral, modelos formais são arquivos *XML* (*eXtensible Markup Language*) e os tipos hierárquicos (*tags*) permitidos estão definidos juntamente com as regras hierárquicas em um arquivo *DTD* (*Document Type Definition*). A definição da *DTD* e o processo de marcação (mapeamento) do texto original para o arquivo *XML* é o que basicamente diferencia os projetos.

Os projetos que mais se destacam são: *Guideline Element Model* [19] e *Medical Guideline Technology* [14].

2.4.1 Medical Guideline Technology (MGT)

O *Medical Guideline Technology* (MGT) é um projeto europeu que tem como principais marcas o *Guide-X* [20], metodologia de formalização de guias em arquivos *XML*, e o *Stepper* [14], ferramenta de marcação de texto.

Como descrito em [20], o *Guide-X* foca no processo de marcação de texto algumas partes específicas do guia:

- sentenças de definição;
- relacionamentos causais entre vários parâmetros clínicos, eventos e etc;
- objetivos ou intenções de uma ação, atividade ou do guia como um todo;
- sentenças procedurais (o que deve ser feito e sob quais condições).

Os autores clamam que estes elementos cobrem aproximadamente qualquer informação importante contida em um guia.

O *Guide-X* é uma abordagem que envolve múltiplos passos nos quais o guia formal vai sofrendo transformações até que o processo de formalização resulte em um modelo

executável por computador. No entanto, as ferramentas necessárias para a concretização dos últimos passos não estão prontas. Assim, ainda não é possível implementar um guia usando as ferramentas do projeto GMT.

2.4.2 Guideline Element Model (GEM)

O projeto GEM [19] talvez seja o mais ativo do gênero. O seu *DTD* foi adotado pela *American Society of Testing and Materials (ASTM)* como a representação padrão para guias de conduta médica em *XML*. Possui uma hierarquia contendo mais de 100 tipos hierárquicos distribuídos em nove grupos principais:

- identificação;
- desenvolvedor;
- propósito;
- audiência alvo;
- população alvo;
- método de desenvolvimento;
- testes;
- plano de revisão;
- componentes de conhecimento.

Este é um projeto ambicioso que, além de buscar a implementação computacional de guias clínicos, preocupa-se com os aspectos de qualidade e manutenção destes.

Atualmente, duas ferramentas estão sendo desenvolvidas paralelamente neste projeto: *GEM Cutter* e *GEM-Q* [21]. A primeira, já operacional, tem como objetivo ajudar no processo de marcação de texto. A segunda, ainda um protótipo, recebe como entrada guias no formato *GEM* e retorna medidas de qualidade metodológica.

É possível implementar guias formalizados no formato GEM. Para exemplos de implementação, consulte: [22] e [23].

2.5 Projetos centrados em modelo

À seguir são apresentados e caracterizados os principais projetos centrados em modelo.

2.5.1 The Arden Syntax

The Arden Syntax[24] define uma sintaxe para representação de módulos de lógica médica (MLMs). Cada módulo é armazenado como arquivo texto no formato ASCII e encapsula lógica e conhecimento necessários para uma decisão ou ação médica. As ações possíveis são: exibição ou envio de mensagens, alertas para um destinatário; execução de outros módulos MLMs; e retorno de dados para o MLM que invocou o atual.

Um MLM é constituído por três partes ou espaços: *maintenance*(manutenção), *library*(biblioteca) e *knowledge*(conhecimento). Os dois primeiros guardam metadados do MLM como título, nome do autor, versão do Arden, propósito, explanação, palavras-chave e citações. O último espaço, subdivide-se em mais 6 partes, as quais encapsulam:

1. *type*: tipo de conhecimento codificado;
2. *data*: dados usados na lógica condicional e invocação;
3. *evoke*: quando MLM deve ser invocado;
4. *logic*: lógica condicional para execução;
5. *action*: ação a ser executada;
6. e *urgency*: urgência ou prioridade do módulo.

O único tipo de conhecimento definido até então é o orientado a dados (*data-driven*). A sintaxe para definição dos dados e especificação da lógica condicional é muito parecida com a das linguagens Pascal e Visual Basic. A seção *data* de *knowledge* pode usar outras MLMs, consultas a uma base dados, e programas externos para definir os dados. Como muitos MLMs dependem da ocorrência de eventos pré-definidos para serem invocados, é preciso que um processo monitor esteja em execução para invocá-los.

A sintaxe Arden oferece uma série de operadores para implementação dos MLMs. No entanto, muitos operadores são meras repetições de operações primitivas existentes em diversas outras linguagens de programação. Isto é um reflexo claro de mistura entre modelagem de conhecimento médico e rotinas de programação em si. As “palavras” necessárias para representar um conhecimento qualquer são aquelas que pertencem ao próprio contexto deste e que ao mesmo tempo o sintetizam. Usar palavras diferentes destas apenas tornam o modelo mais complicado e difícil de entender.

Outro ponto onde a mesma crítica é válida está na especificação de consultas à base de dados diretamente do próprio MLM. Consultas à base de dados são extremamente dependentes do sistema alvo. Portanto, novamente temos o conhecimento médico misturado indevidamente com uma lógica de programação específica para um sistema alvo.

[25] mostra os problemas que podem surgir ao compartilhar um MLM entre instituições diferentes.

A abordagem de acesso ao EMR (*Electronic Medical Record*) usada pela sintaxe Arden se baseia em uma forte tendência de padronização do mesmo na indústria, principalmente norte-americana. Como será visto adiante, esta não é a abordagem escolhida pelo ST-Guide. Padronização externa não faz necessariamente parte do escopo do problema de representar conhecimento e interpretá-lo. Basta que o projeto adote uma representação interna de EMR para cumprir os seus objetivos com a vantagem de que o guia obtido é compartilhável.

Apesar das críticas, a sintaxe Arden é bastante útil e ocupa uma posição histórica importantíssima. Ela foi adotada pela *American Society for Testing and Materials* em 1992 como padrão para definição e compartilhamento de bases de conhecimento de saúde modulares. E, à partir de 1999, foi adotada também pela *Health Level Seven* (HL7). Além disso, hospitais norte americanos de grande porte como o *JFK Medical Center* de Edison, New Jersey, e *Sarasota Memorial Hospital* de Sarasota, Florida, usam sistemas especialistas baseados em sintaxe Arden[26].

2.5.2 GuideLine Interchange Format(GLIF)

GuideLine Interchange Format[27] ou GLIF até a versão GILF2 era uma linguagem para especificação de guias clínicos em fluxogramas ordenados temporalmente por passos de decisão clínica, ação, desvio e/ou sincronização.

Até a versão GILF2 os modelos não podiam ser usados para a implementação do guia por computador, pois a especificação dos passos era uma descrição textual desestruturada informada pelo usuário. À partir da versão GILF3 uma série de modificações foi realizada e a implementação se tornou possível.

É possível modelar um guia em três diferentes níveis de abstração, apresentados na Tabela 2.1 junto com as características do modelo gerado.

Tabela 2.1: Níveis de especificação no GLIF e características do modelo gerado

Nível	Características
1. Conceitual	fluxograma não executável, usado apenas para navegação visual.
2. Computável	soma controle de fluxo, acesso ao EMR. Passível de verificações lógicas.
3. Implementável	pronto para uso por sistema específico. Contém elementos não compatilháveis entre sistemas-alvo.

Outras informações sobre o GLIF3:

- permite o aninhamento de guias em passos de ação;
- decisões podem ser ou automatizadas (*care steps*) ou tomadas obrigatoriamente por um usuário (*choice step*);
- sintaxe da linguagem é baseada em XML;
- suas expressões condicionais e de ação são especificadas por uma gramática que é extensão da sintaxe Arden.
- possui uma camada de conhecimento médico que contém um dicionário de termos e pode prover acesso a bases de conhecimento, possibilitando checagens semânticas nas ações.

Em [28] estratégias de versionamento de guias são exploradas.

2.5.3 PROforma

PROforma[29] foi desenvolvido no Laboratório de Computação Avançada do *Cancer Research UK*, Reino Unido. Engloba uma linguagem lógica híbrida de composição de conhecimento e suas respectivas tecnologias de modelagem, verificação e implementação. O projeto tem como foco aplicações médicas críticas, onde incertezas aparecem naturalmente devido à frequente ausência de informação.

O conhecimento é representado através de uma rede de tarefas (*task network*). Arestas direcionadas entre tarefas representam restrições temporais.

Orientação à objetos é usado para estruturar a ontologia de tarefas, as quais todas herdam de uma tarefa raiz, *Root task*, que possui os atributos genéricos de uma tarefa segundo o modelo conceitual do PROforma (*domino*). À seguir são listados alguns dos principais atributos:

- *Goal* - objetivo da tarefa. Se satisfeito, tarefa pára e pós-condições são assumidas como verdadeiras.
- *Precondition* - lista de pré-condições para execução.
- *Postcondition* - fatos lógicos assumidos ao se finalizar com sucesso.
- *Trigger condition* - gatilhos que podem disparar a tarefa, mediante a validação das pré-condições.
- *Cycle* - condições sob as quais a tarefa deve ser repetida.

A Tabela 2.2 exibe as classes de segundo nível da hierarquia *PROforma*, especializações diretas de *Root task*. Essas são as classes que servem de base para toda modelagem. A ferramenta gráfica de desenho permite inclusão em um diagrama e subsequente configuração das tarefas por preenchimento de *templates*

Tabela 2.2: Segundo nível da hierarquia de tarefas do *PROforma*.

Tarefa	Descrição
<i>decision</i>	envolve a avaliação de opções e a escolha em si.
<i>plan</i>	conjunto de tarefas, incluindo sub-planos, e suas restrições de interação.
<i>action</i>	recomendações textuais ou chamadas a procedimentos externos.
<i>enquiry</i>	uma ação que gera informação. Inclui a descrição da informação desejada. e um método para obtenção dos dados, seja consultando o EMR local do paciente ou a base de dados de um laboratório distante.

A *engine* de implementação de guias *PROforma* pode funcionar em modo *stand-alone*, interagindo com um usuário e exibindo recomendações e questionários baseados no guia em execução. Entretanto, a maneira ideal de utilizá-la é integrando-a com um sistema externo através de comunicação por uma API padrão. Em uma recente reimplementação da *engine* na linguagem Java denominada *Tallis*, os guias podem ser publicados e implementados diretamente pela internet.

Entre os sistemas especialistas de saúde, *PROforma* é o que tem maior sucesso comercial. Ele serve como base para um amplo conjunto de aplicações médicas que têm como objetivo gerenciar desde a dor de pacientes com câncer até um tratamento de asma[26]. É um projeto que propõe soluções para todas as fases de desenvolvimento de um guia médico interpretável por computador. A forma como lida com incertezas e a ontologia de tarefas são características bastante interessantes. Um ponto negativo (recorrente em vários projetos) é a mistura em algumas tarefas de conhecimento médico com rotinas de programação, que indicam por exemplo, como algumas informações devem ser entradas no sistema. Este tipo de rotina é irrelevante para o conhecimento médico em si e por isso não deviam estar misturados.

2.5.4 Asbru

Asbru [30] é fruto de uma parceria entre a Universidade de Tecnologia de Viena (Alemanha) e a Universidade de Stanford (Estados Unidos). Atualmente conta com a colaboração de várias universidades espalhadas pela Europa e oriente médio. O projeto contempla modelagem, verificação e execução de guias.

Um modelo em Asbru é composto por uma hierarquia de **planos-esqueleto**, os quais capturam apenas a essência dos procedimentos, deixando espaço para flexibilizar a execução em si. A linguagem é baseada em intenções e orientada a tempo.

Intenções são definidas como padrões temporais de estados ou ações a serem mantidos, alcançados ou evitados. Usadas para representar o sentido de um plano em várias etapas do mesmo, permitem comparação e crítica melhores entre guia e ações de um usuário. Para um ótimo exemplo, consultar Seção 3.2.1 de [30].

Anotações temporais são formadas por tempo inicial, tempo final, duração e referência. Para comportar incertezas, tempo e duração são definidos por intervalos da forma: [mínimo unidade, máximo unidade]. Referência especifica a origem do tempo (instante inicial, $t = 0$) e pode ser configurada por variáveis.

Asbru oferece um formalismo poderoso para denotar guias. Devido à sua expressividade temporal tem sido muito usado para verificação formal de guias, incluindo identificação de inconsistência lógica (rever definição na Seção 2.1).

2.5.5 EON

EON [31] foi desenvolvido pela Universidade de Stanford e, assim como o Prodigy (Seção 2.5.6), é construído sobre o ambiente de engenharia de conhecimento Protégé 2000 [32].

EON constitui-se de um conjunto de modelos e componentes de software para criação de aplicações baseadas em guias médicos. Contém um conjunto de servidores intermediários que realizam computação necessária ao suporte de tarefas específicas do cuidado médico baseado em guias. Entre os servidores estão: *Padda*, servidor de execução de guias que recebe como entrada dados do paciente e guia, e retorna as recomendações específicas do guia para o paciente como saída; *Tzolkin*, mediador de dados temporais, capaz de interpretar e retornar o resultado de consultas com complexos relacionamentos e abstrações temporais; e *WOZ*, provedor de serviços de explanação para outros componentes [31].

Toda aplicação EON e servidores possuem acesso aos modelos e bases de conhecimento criados no Protégé 2000. Entre os modelos estão disponíveis: representação do paciente no EON; base extensível de conhecimento médico específico; e representação do guia médico EON. Guias EON, como Asbru, também são planos-esqueleto refinados em tempo de execução até que o plano final seja gerado [33].

2.5.6 Prodigy

Prodigy [34] é uma iniciativa do *Sowerby Centre for Health Informatics at Newcastle* (SKIN), Reino Unido, e propõe solução para modelagem e implementação de guias para atendimento primário. Antes vinculado a Universidade de Newcastle, o SCHIN agora é uma empresa privada sem fins lucrativos [26]. O Prodigy encontra-se descontinuado

desde sua terceira versão, porém está integrado com vários sistemas médicos comerciais e continua a ser amplamente usado por clínicos gerais na Inglaterra.

Um guia Prodigy 3 se parece muito com uma máquina de estados e é organizado como um grafo orientado de *Scenarios* (cenários), *Action steps* (passos de ação) e *Subguidelines* (sub-guias). Arestas no modelo indicam que um estado (destino) é alcançável a partir do outro (origem). Transições ocorrem dependendo da avaliação das *pré-condições*, se estado for um cenário, ou *regras de inclusão e exclusão*, se for um passo de ação ou sub-guia.

Scenarios representam estados facilmente identificáveis do paciente e servem como pontos de entrada no guia.

Actions Steps são estados que refinam as ações do guia a partir de cenários. Podem iniciar, modificar ou parar *atividades* que persistem pelo tempo ou executar *ações* instantâneas.

Subguidelines são usados para representar um conjunto de decisões e atividades/ações que fazem parte de um mesmo procedimento. Encapsulam uma série de *Scenarios*, *Actions Steps* e suas interações.

Diferente de sua segunda versão, que relegava ao usuário a tomada de decisão para transição de estados, Prodigy 3 implementa guias automaticamente. O clínico tem liberdade para seguir ou não as recomendações e o cenário atual do paciente é sempre reestimado. Se necessário, uma lista dos prováveis cenários é mostrada para seleção.

Conceitualmente Prodigy 3 e ST-Guide (Seção 3) são muito parecidos. Ambos se propõem a resolver o mesmo tipo de problema, modelando encontros entre paciente e clínico. No entanto, apesar de se equivalerem em uma série de aspectos (principalmente em relação a expressividade das linguagens), diferenças em objetivos secundários impactaram tanto o desenho dos projetos quanto as funcionalidades disponíveis. Por exemplo, como as transições do Prodigy apenas servem para mostrar possíveis saídas de um estado, a lógica das transições estão no estado de destino e por isso é fácil ver um guia como um componente ou sub-guia. No ST-Guide, transições dependem de informações disponíveis no estado de origem, o que traz complicações a componentização. Por outro lado, devido às mesmas características, é extremamente mais fácil corrigir inconformidades em um modelo ST-Guide do que em um Prodigy 3.

Sem querer fazer uma comparação injusta e tendo apenas intenção de explicitar diferenças entre os projetos, alguns quesitos presentes no ST-Guide e não encontrados no Prodigy 3 são: verificação lógica interativa do modelo; diferentes modos de interação; e independência de plataforma (Prodigy 3 se integra com outros sistemas via biblioteca de tecnologia proprietária *ActiveX* [35] conforme [34]). Quesitos presentes no Prodigy não encontrados no ST-Guide: hierarquização de guias (sub-guias); e ações específicas sobre administração de drogas (incremento e decremento de dose, por exemplo).

Capítulo 3

ST-Guide

O ST-Guide [10] é um projeto centrado em modelo, ativo e em desenvolvimento na Universidade Estadual de Campinas (Unicamp), Brasil. O foco deste projeto é a modelagem e implementação de guias clínicos para *atendimento primário*¹ e secundário, sobretudo para doenças crônicas. A modelagem de guias baseada em padrões de atendimento (obtidos empiricamente) e as diferentes possibilidades de interação [36, 37] com o guia implementado são características que diferenciam este projeto dos demais presentes na literatura contemporânea.

3.1 Fundamentos

A estratégia de formalização de guias usada pelo ST-Guide tem origem na observação de como ocorrem consultas de atendimento primário. Neste tipo de consulta, após o diagnóstico, o médico geralmente segue uma linha sistemática de atendimento, que pode ser abstraída em duas fases:

1. Fase A
 - requisitar novos exames laboratoriais;
 - fazer recomendações médicas ao paciente;
 - prescrever medicamentos;
 - dar alta ao paciente ou encaminhá-lo para especialista;
 - marcar próxima consulta;

¹*Atendimento primário* é a fase inicial de um tratamento de saúde, geralmente conduzida por um clínico geral, pediatra ou ginecologista. Nesta fase, o paciente é tratado com os métodos mais difundidos e aceitos pela comunidade médica. Se os cuidados primários não forem suficientes para restabelecer e/ou manter a saúde do paciente, este é encaminhado para um especialista, profissional que caracteriza a fase de *atendimento secundário*.

2. Fase B

- realizar exames físicos;
- perguntar ao paciente questões relevantes para o tratamento;
- analisar exames (laboratoriais e físicos) e perguntas do paciente ;
- identificar, manter ou alterar linha de tratamento;

Assumindo que o procedimento clínico durante uma consulta primária pode ser descrito pelo modelo acima, é bastante natural que o formalismo escolhido para representar um tratamento (ou guia clínico prático) seja uma máquina de estados. Para se instanciar uma máquina de estados, é preciso especificar alfabeto, conjunto de estados, estados iniciais, estados finais, função de transição e ações de cada estado. Considere que:

1. cada estado do guia encapsula uma Fase A e uma Fase B;
2. dependendo do estado de saúde do paciente e perguntas e exames realizados, este é conduzido para outros estados do guia;
3. pode ser atribuído um estado inicial ao paciente;
4. podem existir nenhum, um ou mais estados finais de tratamento (alta ou encaminhamento para especialista).

(1) caracteriza o que é um estado e as ações que devem ser executadas no mesmo. Baseando-se nisso, pode-se definir um conjunto de estados. (2) indica que cada letra do alfabeto está associada com o resultado de uma expressão condicional e, simultaneamente, descreve a forma da função de transição; (3) garante que a máquina sempre poderá ser inicializada e (4) apenas chama atenção para a possibilidade de uma modelagem onde a máquina não deva parar (tratamento de doenças crônicas, por exemplo). Temos, portanto, uma caracterização informal de uma máquina de estados capaz de modelar o processo de atendimento médico primário e, conseqüentemente, guias práticos primários.

A Figura 3.1 traz um exemplo visual de máquina de estados. Estados são representados por elipses e transições por arestas direcionadas. Cada aresta está associada com uma entrada (letras minúsculas no diagrama), que no ST-Guide corresponde à uma expressão condicional em lógica Booleana. Arestas sem origem indicam que o estado incidido é um estado inicial da máquina (no exemplo, S1). S2 é um sorvedouro, uma vez que não é origem de nenhuma transição. Estados sorvedouros não são necessariamente estados finais, os quais precisam ser explicitamente listados.

Extraída de [10], a Figura 3.2 ilustra o modelo de consultas no ST-Guide e mostra que as transições de estado ocorrem durante as consultas. Transições ocorrem da Fase B do estado de origem para a Fase A do de destino.

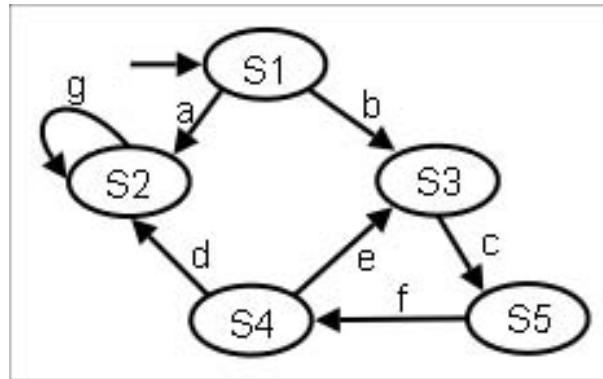


Figura 3.1: Exemplo de máquina de estados.

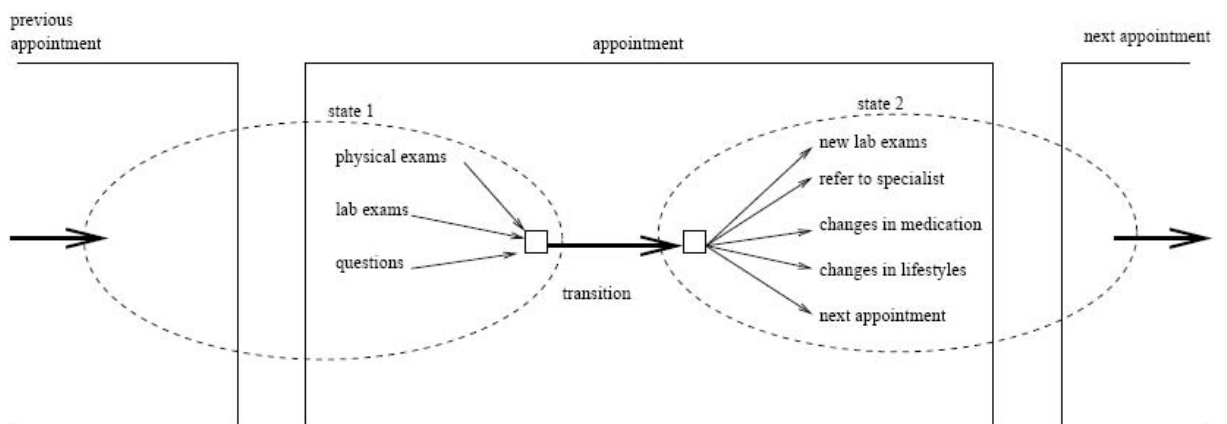


Figura 3.2: Modelo de consultas.

Na prática, o formalismo empregado pelo ST-Guide tem se mostrado realmente útil, possibilitando a modelagem de guias para diagnóstico e tratamento de tuberculose, hepatite C, hipertensão e câncer em células germinativas de crianças [10].

Por ser uma abordagem que formaliza guias clínicos práticos em máquinas de estados, o sistema foi batizado, em inglês, de *State/Transitions Guide*, ou, simplesmente, ST-Guide. À seguir, são descritos os dois principais componentes deste sistema.

3.2 ST-Linguagem

ST-Linguagem é o padrão de codificação de texto usado para representar os guias práticos através de máquinas de estado.

Até o início de 2006, baseava-se em sintaxe de programação *Prolog*. Desde então, decidiu-se por basear a ST-Linguagem em uma linguagem que a tornasse mais independente da tecnologia de *software* que interpreta e implementa os guias. Ao mesmo tempo, seria interessante que a linguagem-base escolhida fosse mais conhecida pela comunidade de desenvolvedores de *software*. Uma excelente candidata, que cumpre estes requisitos, é a *XML*. Por isso, no início de 2006 trabalhou-se na especificação de uma nova ST-Linguagem em *XML*.

A gramática completa da ST-Linguagem em *XML* encontra-se no Apêndice B. A Figura 3.3 mostra um trecho do esquema *XML* da linguagem, dando uma visão geral sobre a sua gramática. Sua expressividade foi colocada em prova a cada guia formalizado.

Guias clínicos para tratamento de tumores em células germinativas (TCG) e hipertensão foram formalizados baseando-se na modelagem feita por [36]. Outros guias, como: assistência pré-natal [2]; tratamento dos transtornos de ansiedade [38]; e icterícia em recém-nascidos saudáveis [39] foram formalizados durante este trabalho.

3.2.1 Mudanças

Nesta Seção, são descritas e comentadas as alterações mais significativas feitas na ST-Linguagem durante este trabalho.

Avaliação das transições entre estados

A versão anterior do ST-Guide implementava uma máquina de estados através do paradigma de programação funcional, utilizando funções lógicas. Dessa forma, as transições da máquina de estados também eram mapeadas em funções lógicas. Se usamos o interpretador da linguagem funcional correspondente, obviamente o processo de interpretação do guia é facilitado.

```

<!-- Guide -->
- <xs:element name="guide">
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="author" />
  <xs:element ref="name" />
  <xs:element ref="version" />
  <xs:element ref="declarations" />
  <xs:element ref="states" />
  <xs:element ref="helperStates" />
</xs:sequence>
</xs:complexType>
</xs:element>

```

```

<!-- declarations -->
- <xs:element name="declarations">
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="questionsDec" minOccurs="0" maxOccurs="unbounded" />
  <xs:element ref="remindersDec" minOccurs="0" maxOccurs="unbounded" />
  <xs:element ref="medicationsDec" minOccurs="0" maxOccurs="unbounded" />
  <xs:element ref="phyExamsDec" minOccurs="0" maxOccurs="unbounded" />
  <xs:element ref="labExamsDec" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>

```

```

<!-- State -->
- <xs:element name="state">
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="posX" />
  <xs:element ref="posY" />
  <xs:element ref="width" />
  <xs:element ref="height" />
  <xs:element ref="description" />
  <xs:element ref="phyExams" />
  <xs:element ref="labExams" />
  <xs:element ref="questions" />
  <xs:element ref="medications" />
  <xs:element ref="reminders" />
  <xs:element ref="nextSession" />
  <xs:element ref="literals" />
  <xs:element ref="uses" />
  <xs:element ref="transitions" />
</xs:sequence>
  <xs:attribute ref="id" use="required" />
</xs:complexType>
</xs:element>

```

```

<!-- Helper State -->
- <xs:element name="helperState">
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="posX" />
  <xs:element ref="posY" />
  <xs:element ref="width" />
  <xs:element ref="height" />
  <xs:element ref="description" />
  <xs:element ref="phyExams" />
  <xs:element ref="labExams" />
  <xs:element ref="questions" />
  <xs:element ref="medications" />
  <xs:element ref="reminders" />
  <xs:element ref="literals" />
  <xs:element ref="uses" />
</xs:sequence>
  <xs:attribute ref="id" use="required" />
</xs:complexType>
</xs:element>

```

Figura 3.3: Fragmentos do esquema XML da ST-Linguagem: elementos *Guide*, *State*, *HelperState* e *Declarations*.

A nova versão da ST-Linguagem não oferece a mesma facilidade de interpretação. As transições são representadas de forma direta utilizando-se modelagem orientada à objetos. Agora, transições pertencem ao seu estado de origem e cada transição possui um identificador, um estado de destino e uma condição, que é uma expressão em lógica Booleana. Os literais usados nesta expressão devem estar definidos no estado de origem da transição.

A forma como literais são modelados também merece atenção. Cada literal possui identificador, descrição, operador e operandos. Um operador é composto por identificador (nome da classe que o representa) e número de entradas ou operandos que ele requer. Um operando é especificado por identificador, tipo (nome da classe que o representa) e valor.

A forma como estes elementos interagem em tempo de execução será explicada em detalhes, adiante, na Seção 3.3.2. Por ora, pode-se ter uma boa idéia do funcionamento do sistema sabendo que literais podem assumir dois valores, verdadeiro ou falso, e que o valor assumido depende da ação do seu operador sobre os operandos em tempo de execução. Operandos e operador são instanciados quando o guia é processado de acordo com seu identificador e tipo, respectivamente. Estes campos guardam o nome das classes que devem ser instanciadas. Quando há uma consulta ao guia, os operandos acessam a representação interna do paciente (Seção 3.3.1) e configuram a si mesmos. Em seguida, para saber o valor atual do literal, basta executar o operador sobre os operandos configurados.

Helper-States

Durante os experimentos de formalização de guias usando a ferramenta de desenvolvimento a ser apresentada no Capítulo 5 sentiu-se a necessidade de criar mecanismos de modelagem mais eficientes. O problema encontrado foi que em muitos guias é preciso tratar variáveis independentes de forma simultânea. Nestas situações, usando o formalismo anterior [10], é preciso modelar um a um todos os possíveis caminhos do tratamento. Como será mostrado no exemplo à seguir, em algumas situações isto é extremamente indesejável.

Um exemplo: considere um tratamento simples constituído até então por um único caminho na máquina de estados. Suponha que existe um estado deste tratamento, S , que deve variar um pouco dependendo de algumas condições, c_1, c_2, \dots, c_n (onde n é número inteiro), que podem ser verdadeiras ou falsas e são independentes entre si. Além disso, se c_i , $1 \leq i \leq n$ for verdadeira, implica em uma mudança, δS_i , em S tal que surge um novo estado $S_i = S + \delta S_i$. Para que o tratamento seja completo, isto é, especifique o que fazer para todas as possíveis combinações de condições antes de S , é preciso que sejam criados x novos estados, onde

$$x = 1 + \sum_{i=1}^{n-1} C_{n,i} \quad (3.1)$$

e $C_{n,i} = \frac{n!}{i!(n-i)!}$ é o número de possíveis combinações de n elementos, i a i . Assim, para números pequenos como $n = 3$, $n = 4$ ou $n = 5$, respectivamente, temos $x = 7$, $x = 15$ ou $x = 31$ novos estados e, por consequência, 7, 15 ou 31 novos caminhos criados no tratamento. Se existirem k estados como S vizinhos no caminho do tratamento, então teremos que inserir kx novos estados e x^k novos caminhos.

Fica claro que fazer estas combinações manualmente é uma tarefa repetitiva, entediante e, perigosamente, propensa à erros. Para mitigar este problema, na nova *ST-Linguagem* usamos um tipo especial de estado denominado, ***Helper-State***. Como o nome indica, este tipo de estado serve como um ajudante para os estados convencionais (*ST-States*) da máquina. A relação entre eles é de **uso**. Isto é, um *ST-State* usa um ou mais *Helper-States* conforme as condições da relação de uso são verdadeiras ou não. Usar um *Helper-State*, em termos práticos, é o mesmo que unir-se a ele. Para facilitar mais ainda a modelagem, *Helper-States* podem usar outros *Helper-States*.

Do ponto de vista de atributos possuídos, *Helper-States* diferem de *St-States* apenas quanto aos campos “Próxima sessão” e “Transições”, os quais não estão presentes nos *Helper-States*.

3.3 ST-Sistema

ST-Sistema refere-se à parte do sistema encarregada de processar, interpretar e implementar os guias representados em *ST-Linguagem*. Para isso, assume uma representação eletrônica interna do paciente (dados pessoais e físicos, exames e histórico). O mapeamento entre a representação de paciente interna do *ST-Sistema* e a de outro sistema (de prontuário eletrônico, por exemplo) é feita em uma camada de controle da aplicação principal.

O *ST-Sistema* pode atuar em diferentes modos [36, 37]:

1. aprendiz: avalia e critica ações do usuário consulta-a-consulta;
2. autoritário: usuário deve seguir estritamente recomendações do sistema;
3. especialista: usuário não é obrigado a seguir recomendações do sistema;
4. auditor: recebe um tratamento como entrada e o avalia em relação às recomendações de um guia.

O que diferencia os modos autoritário e especialista não é a ação do ST-Sistema em si, mas sim o poder que aplicação externa concede ao usuário. É importante frisar que independente do modo de interação, este poder é sempre relativo, uma vez que a decisão final de seguir uma recomendação ou não é responsabilidade exclusiva do profissional de saúde. Se ele discorda da recomendação ou acha que ela pode ser prejudicial ao paciente, então não deve segui-la.

O ST-Sistema passa por uma fase de reestruturação. Sua versão completa está implementada em *Prolog*. Desde o início de 2006, tem-se trabalhado no desenvolvimento de uma nova versão na plataforma *Java*. Esta decisão foi tomada devido a vários fatores, entre eles: possibilidade de uma modelagem mais robusta e extensível, baseada no paradigma orientado à objetos; componentes construídos podem ser facilmente integráveis a sistemas *desktop* ou *web*; tecnologia bastante conhecida no meio acadêmico e na indústria; e, com grande peso, conhecimentos técnicos da nova equipe de desenvolvimento do projeto.

Será apresentada nesta Seção a nova versão do ST-Sistema, capaz de processar, interpretar e implementar guias no novo formato de ST-Linguagem. Por enquanto, somente os modos especialista e autoritário estão implementados. À seguir, os temas abordados são:

- representação interna do paciente (Seção 3.3.1);
- e arquitetura do ST-Sistema (Seção 3.3.2).

3.3.1 Representação interna do paciente

Para que se possa aplicar um guia para um dado paciente, é preciso, primeiro, saber como estão organizados e como acessar os dados sobre este paciente.

A abordagem elaborada neste projeto visa a tornar o ST-Sistema independente das aplicações (sistemas computacionais) que o usam, inclusive quanto à representação eletrônica ou prontuário eletrônico do paciente. Assim diferentes sistemas com bases de dados e prontuários eletrônicos heterogêneos podem usar o mesmo ST-Sistema para implementar computacionalmente guias médicos, sem que seja preciso nenhuma alteração no ST-Sistema em si.

O modelo conceitual de prontuário eletrônico de paciente usado pelo ST-Sistema é mostrado no Apêndice C, Tabela C.1. Em seguida, nas Tabelas C.2 a C.10, são especificados conceitualmente todos os atributos direta ou indiretamente referenciados na Tabela C.1, o que acaba fornecendo uma visão estrutural e conceitual quase completa do sistema. O modelo completo do sistema em *UML* [40] é mostrado na Seção 3.3.2, que apresenta a arquitetura e arcabouço do ST-Sistema.

Integração com o ST-Sistema

A integração entre aplicações externas e o ST-Sistema se dá através de uma etapa de mapeamento entre a representação eletrônica de paciente usada na aplicação externa e a representação eletrônica interna, usada no ST-Sistema. Este mapeamento deve ser realizado na aplicação externa. Para o ST-Sistema é transparente a forma como os dados estão guardados no sistema da aplicação externa. O contrário nem sempre é verdadeiro. Na maioria dos sistemas de prontuário eletrônico há registro sobre medicações, e exames físicos e laboratoriais do paciente. No entanto, não costuma haver registro de perguntas realizadas. Portanto, na maioria das vezes a integração com o ST-Sistema exige que o sistema externo guarde dados que ele não guardava antes. Além das perguntas, outro dado que o sistema externo deverá guardar é o identificador do estado atual do paciente no tratamento.

3.3.2 Arquitetura

Nesta seção será apresentada a arquitetura e o arcabouço implementado para o ST-Sistema. Para isso serão utilizados diagramas UML [40]. O objetivo é fornecer um modelo orientado a objetos que resuma conceitualmente o que é o ST-Sistema, como ele funciona e como pode ser estendido. Arquitetura e arcabouço não serão tratados separadamente para evitar redundância. Não utilizamos nenhum processo de engenharia de *software* para descrever o ST-Sistema porque o modelo resultante seria muito mais rico do que julgamos necessário apresentar aqui.

Modelo Estrutural

O diagrama de classes da Figura 3.4 mostra como se relacionam classes e interfaces que compõem a arquitetura do ST-Sistema. O diagrama da Figura 3.5 complementa o último, mostrando a modelagem em *UML* da representação interna do paciente (classe *Patient*). Em cada classe, são mostrados apenas os atributos e métodos conceitualmente importantes. A classe abstrata *GuidelineRunner*, presente na Figura 3.4, e as interfaces *Operand* e *Operator*, apresentadas em detalhes nas Figuras 3.6 e 3.7, constituem os pontos de extensão da arquitetura.

Os atributos das classes e o relacionamento entre as mesmas desenhado nos diagramas das Figuras 3.4 e 3.5 têm origem na modelagem conceitual apresentada anteriormente em forma de tabelas no Apêndice C. As Tabelas C.1 a C.10 servem como dicionário de dados para as classes desenhadas, mediante a tradução direta dos termos em inglês dos diagramas para o português. Note que por possuírem características idênticas, exames físicos (*PhyExam*) e exames laboratoriais (*LabExam*) foram modelados como subclasses

de exame médico (*MedicalExam*).

A classe *State* possui dois atributos, posição (*position*) e dimensão (*dimension*), e vários métodos que não foram apresentados na Tabela C.2. Os atributos são necessários para guardar a posição cartesiana (x,y) e as dimensões (largura,altura) do estado no modelo gráfico do tratamento desenvolvido usando o ST-Modeler (Seção 5). Já os métodos permitem responder questões práticas importantes, como:

- quais exames possuem resultados válidos e quais estão faltando?
- quais perguntas têm respostas válidas e quais ainda precisam ser respondidas?

As únicas classes que não foram previamente explicadas são: *MedicalGuideline*, *GuidelineRunner* e *SpecialistModeRunner*.

MedicalGuideline é a classe que encapsula na forma de objeto o guia formal em ST-Linguagem. O processamento e encapsulamento do guia em objetos é feito através de uma chamada ao método *load*, que recebe como entrada o nome do arquivo que contém o guia em ST-Linguagem. Quando *load* termina de executar, a instância de *MedicalGuideline* fica totalmente configurada com os dados do guia e está pronta para processá-lo em função de pacientes. Isto ocorre através de dois métodos: *getState* e *process*.

1. *getState*: dado um paciente (objeto *Patient*) com o identificador do estado atual (*currentState*) configurado, este método retorna o mesmo estado atual, porém com todas as recomendações do guia configuradas;
2. *process*: recebe como entrada um paciente, carrega totalmente o estado atual deste, configura e avalia os literais em função do paciente e retorna uma tabela *Hash* (facilmente transformada em lista em Java) contendo as possíveis transições de estado.

Os atributos de *MedicalGuideline* são: *nome do guia*, *autor*, *versão* e *declarações*. Este último atributo guarda todas as declarações de exames físicos e laboratoriais, medicamentos, lembretes e perguntas que são usadas ao longo do tratamento formalizado. Assim, vários estados podem usar a mesma pergunta, exame ou outro item qualquer fazendo apenas referências aos itens anteriormente declarados. Como a estrutura de dados destinada a armazenar as declarações deve oferecer acesso rápido aos seus itens, a estrutura escolhida para *declarations* foi do tipo tabela *Hash*.

Além de oferecer acesso para funcionalidades importantes (*load* e *getState*) implementadas em *MedicalGuideline*, a classe abstrata *GuidelineRunner* tem como principal função ser o ponto de extensão para as implementações dos modos de interação[37] de usuário com o guia. Classes não-abstratas que herdam *GuidelineRunner* devem no mínimo implementar o método *process*, no qual a transição automática entre estados deve ser validada

e implementada. Por enquanto só os modos especialista e autoritário estão disponíveis através da mesma classe, *SpecialistModeRunner*.

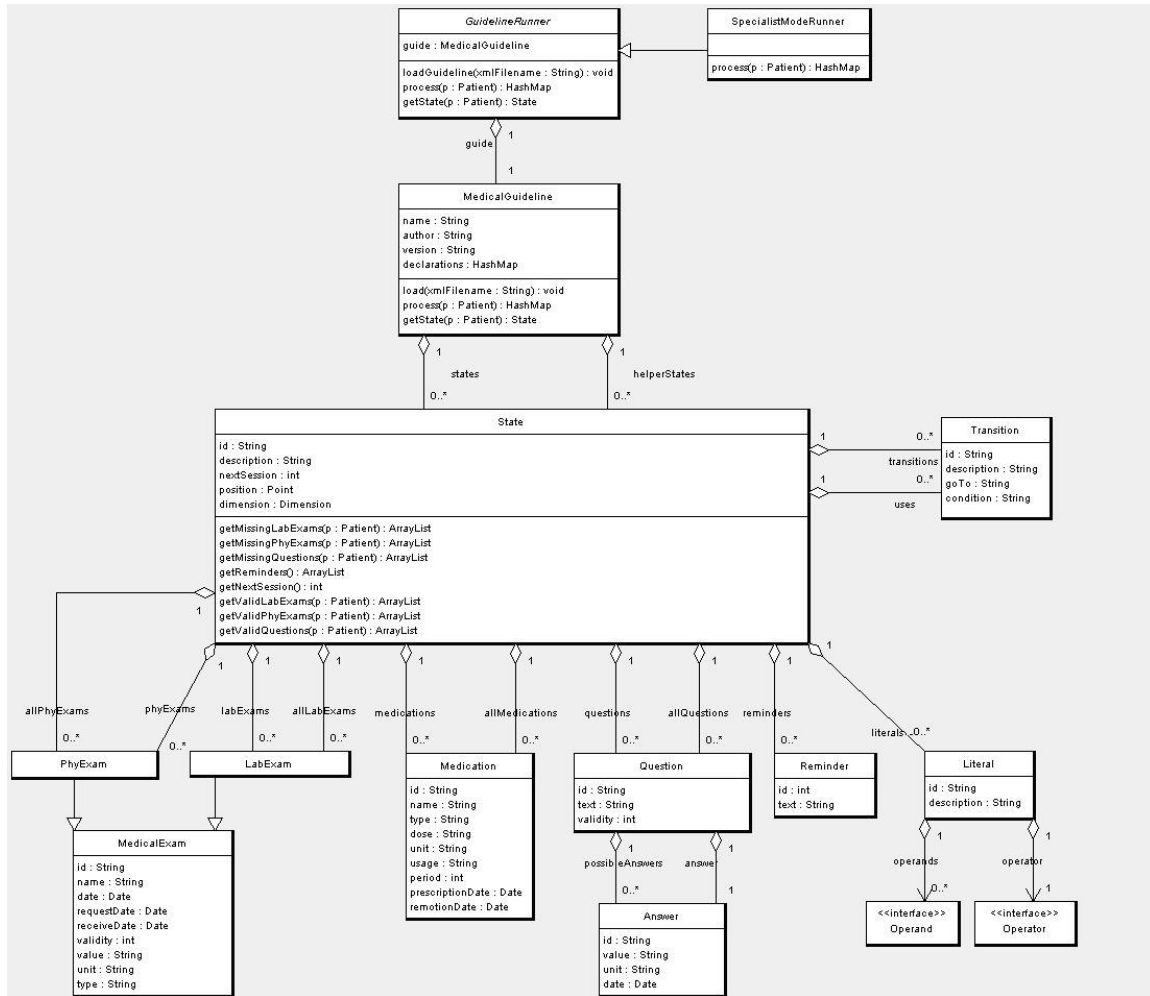


Figura 3.4: Diagrama principal de classes do ST-Sistema.

Pontos de Extensão

As interfaces *Operand* e *Operator* e a classe *GuidelineRunner*, conforme afirmado anteriormente, são os pontos de extensão do ST-Sistema.

Para uma classe implementar *Operator*, vide o diagrama da Figura 3.6, deverá implementar os métodos *getNuOperands* e *perform*. O primeiro método é usado para saber quantos operandos o operador usa. O segundo encapsula a lógica ou algoritmo da operação em si. *perform* recebe como parâmetro um vetor de operandos, e tem como funções: verificar se os operandos estão configurados apropriadamente; e processar estes operandos

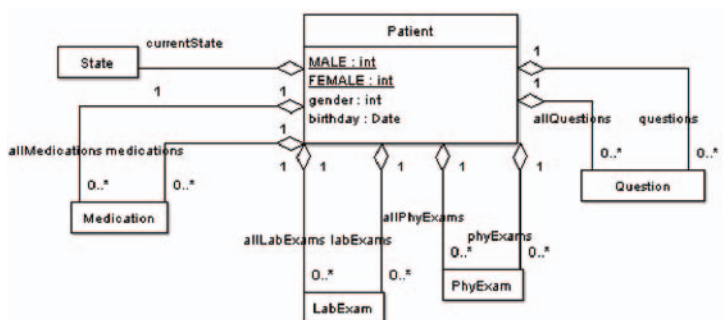


Figura 3.5: Representação interna de paciente.

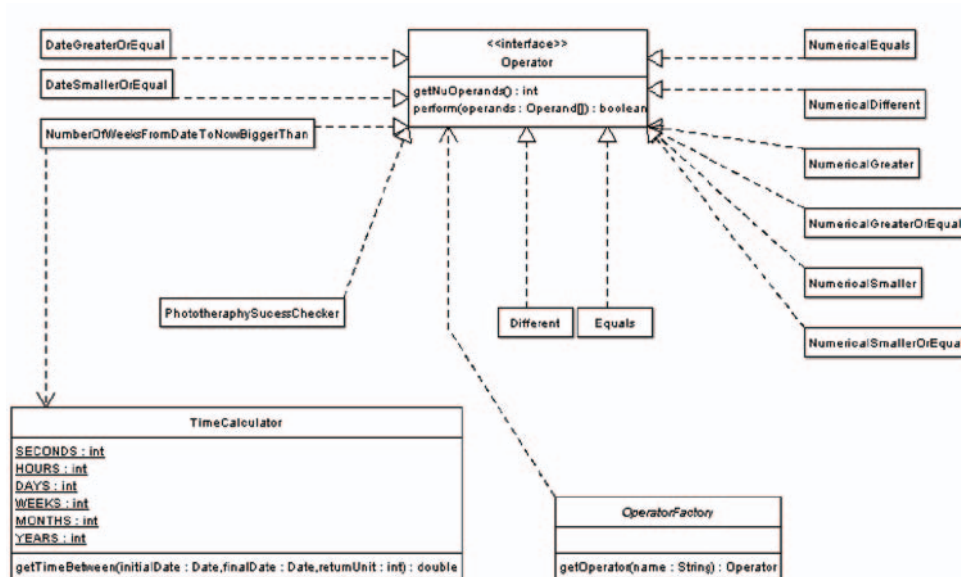


Figura 3.6: Interface *Operator* e classes que a implementam.

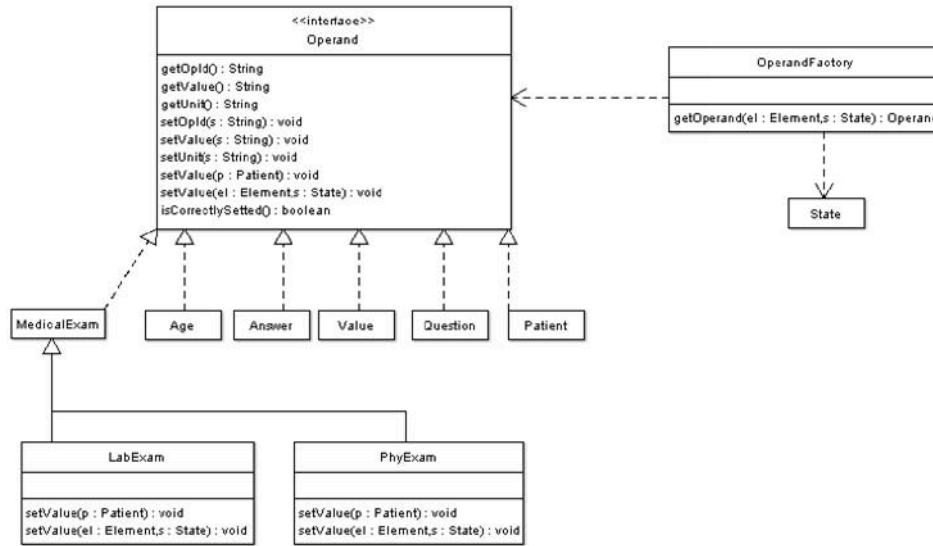


Figura 3.7: Interface *Operand* e classes que a implementam.

segundo a sua lógica interna retornando verdadeiro ou falso. Na Tabela 3.1 tem-se exemplos de classes que implementam *Operator* já existentes no arcabouço do ST-Sistema.

Tabela 3.1: Exemplo de operadores implementados no ST-Sistema. “Op.” usado como diminutivo de operando.

Operador	Ops.	Descrição
<i>Equals</i>	2	verifica se valor dos ops. é igual textualmente
<i>Different</i>	2	verifica se valor dos ops. é diferente textualmente
<i>NumericalEquals</i>	2	verifica se valor dos ops. é igual numericamente
<i>NumericalDifferent</i>	2	verifica se valor dos ops. é diferente numericamente
<i>NumericalGreater</i>	2	verifica se 1 ^o op. é numericamente maior que 2 ^o
<i>NumericalSmaller</i>	2	verifica se 1 ^o op. é numericamente menor que 2 ^o

Apesar de sua simplicidade, os operadores da Tabela 3.1 são de longe os mais usados nos tratamentos formalizados durante este trabalho. Isso é bastante natural, uma vez que muitas decisões na rotina médica estão baseadas em comparações numéricas (por exemplo, “pressão sistólica é maior que 15 mmHg?”) e/ou em confirmações ou não de informações (por exemplo, “há histórico de hipertensão na família?”).

Para que operadores sejam realmente úteis é preciso que eles tenham acesso aos dados do prontuário eletrônico do paciente e do guia formal. Este acesso ocorre através dos

operandos, classes que implementam a interface *Operand*. Operandos diferentes encapsulam diferentes tipos de dados, tanto do prontuário eletrônico como do guia formal. Como operandos constituem a única entrada de dados para os operadores, é vital que eles sejam capazes de encapsular qualquer informação usada durante o tratamento.

Na Figura 3.7, a interface *Operand* é detalhada e um conjunto de operandos já estabelecidos no arcabouço do ST-Sistema é mostrado. As classes que implementam *Operand* são listadas e descritas na Tabela 3.2. À seguir, na Tabela 3.3, os métodos de *Operand* são descritos.

Tabela 3.2: Operandos implementados no ST-Sistema.

Operando	Descrição
<i>PhyExam</i>	exame físico
<i>LabExam</i>	exame laboratorial
<i>Value</i>	um valor qualquer, podendo ter uma unidade associada
<i>Question</i>	pergunta realizada para paciente
<i>Answer</i>	resposta de uma pergunta
<i>Age</i>	idade do paciente
<i>Patient</i>	prontuário eletrônico do paciente

Das classes que implementam *Operand*, apenas duas são novas para o leitor: *Age* e *Value*. *Age* é um operando cujo valor é a idade de um paciente em horas, dias, semanas ou anos, dependendo da unidade configurada no guia. *Value* encapsula um valor (numérico ou não) e sua unidade, quando aplicável. *Value* é usado principalmente para encapsular resultados de exames médicos.

Para garantir que um operador pudesse ter acesso a todos os dados disponíveis no EMR, *Patient* também é um operando.

A classe *OperandFactory* (Figura 3.7) é responsável por instanciar novos operandos durante o carregamento de um guia. Como a classe do objeto a ser instanciado só é conhecida em tempo de execução é utilizado o mecanismo de reflexão em Java [41, 42] para elegantemente criar o objeto. Além de instanciar um novo operando à partir de especificações do guia em processamento, *OperandFactory* também permite que o novo operando possa se configurar à partir de informações do guia. Isso ocorre com uma chamada ao método *setValue(el:Element,s:State):void* do operando recém-instanciado. Este processo é mostrado usando um diagrama de sequência *UML* na Figura 3.9.

Tabela 3.3: Descrição dos métodos da interface *Operand*

Método	Descrição
<i>getOpId():String</i>	retorna identificador do operando
<i>getValue():String</i>	retorna valor do operando
<i>getUnit():String</i>	retorna unidade do valor
<i>setOpId(s:String):void</i>	configura identificador do operando
<i>setUnit(s:String):void</i>	configura unidade
<i>setValue(p:Patient):void</i>	configura valor com base no paciente <i>p</i>
<i>setValue(el:Element,s:State):void</i>	configura valor com base no guia formal
<i>isCorrectlySetted():boolean</i>	retorna <i>true</i> se o operando está corretamente configurado

Interações

Nesta seção é mostrado através de diagramas de sequência *UML* como os objetos do ST-Sistema interagem durante o *carregamento*, Figura 3.8, e *implementação computacional*, Figura 3.10, de um guia por uma aplicação externa ou de usuário.

Detalhes sobre como os operandos se configuram à partir do guia durante o carregamento são modelados no diagrama da Figura 3.9.

O diagrama da Figura 3.11 especifica como os objetos interagem durante a avaliação das transições de estado e, neste contexto, mostra como um operando se configura à partir dos dados do paciente antes de ser usado pelo operador do literal correspondente. O método *performModelLogic()* de *GuidelineRunner* representa a execução da lógica do modo de interação com o usuário antes que *process* termine.

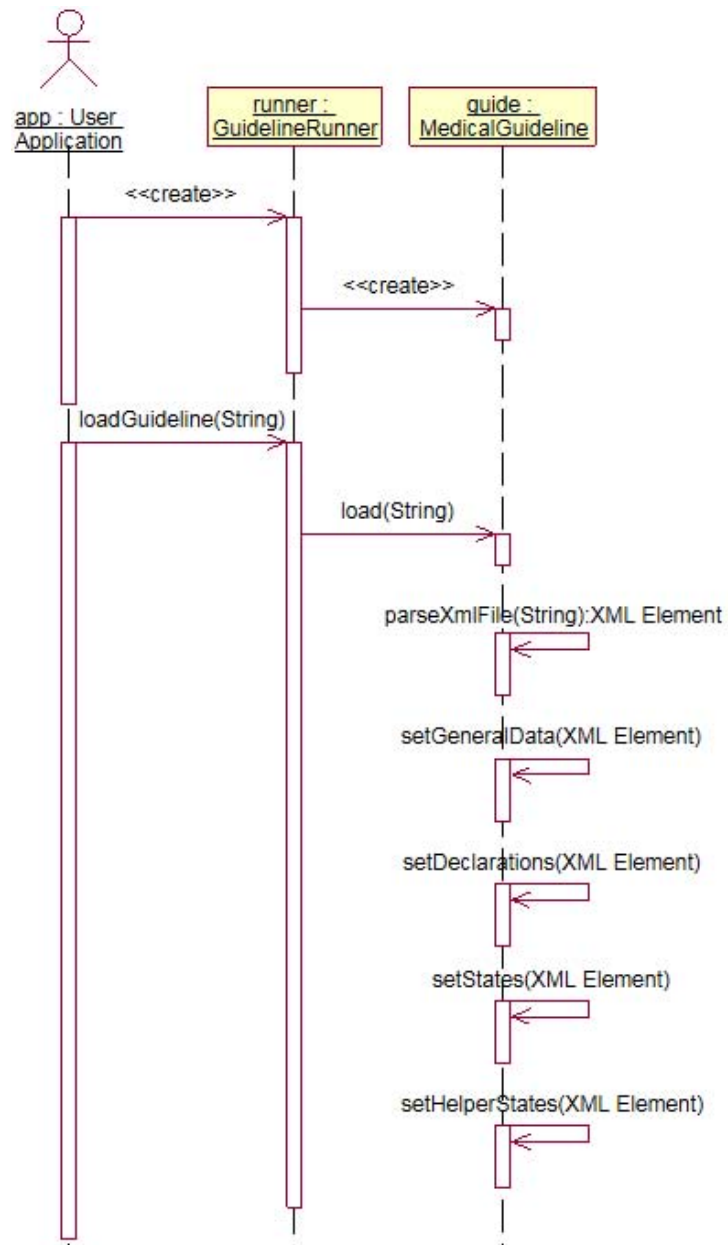


Figura 3.8: Processo de carregamento de guia por uma aplicação externa.

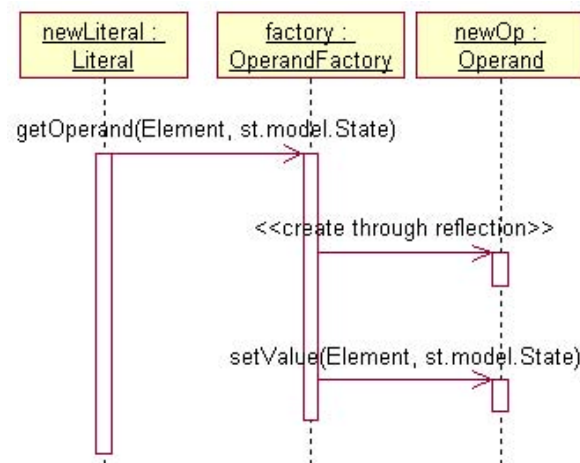


Figura 3.9: Detalhe do processo de carregamento de guia, “dentro” do método *MedicalGuideline.setStates(Element)*. Instanciação de operando por reflexão e configuração à partir de dados do guia.

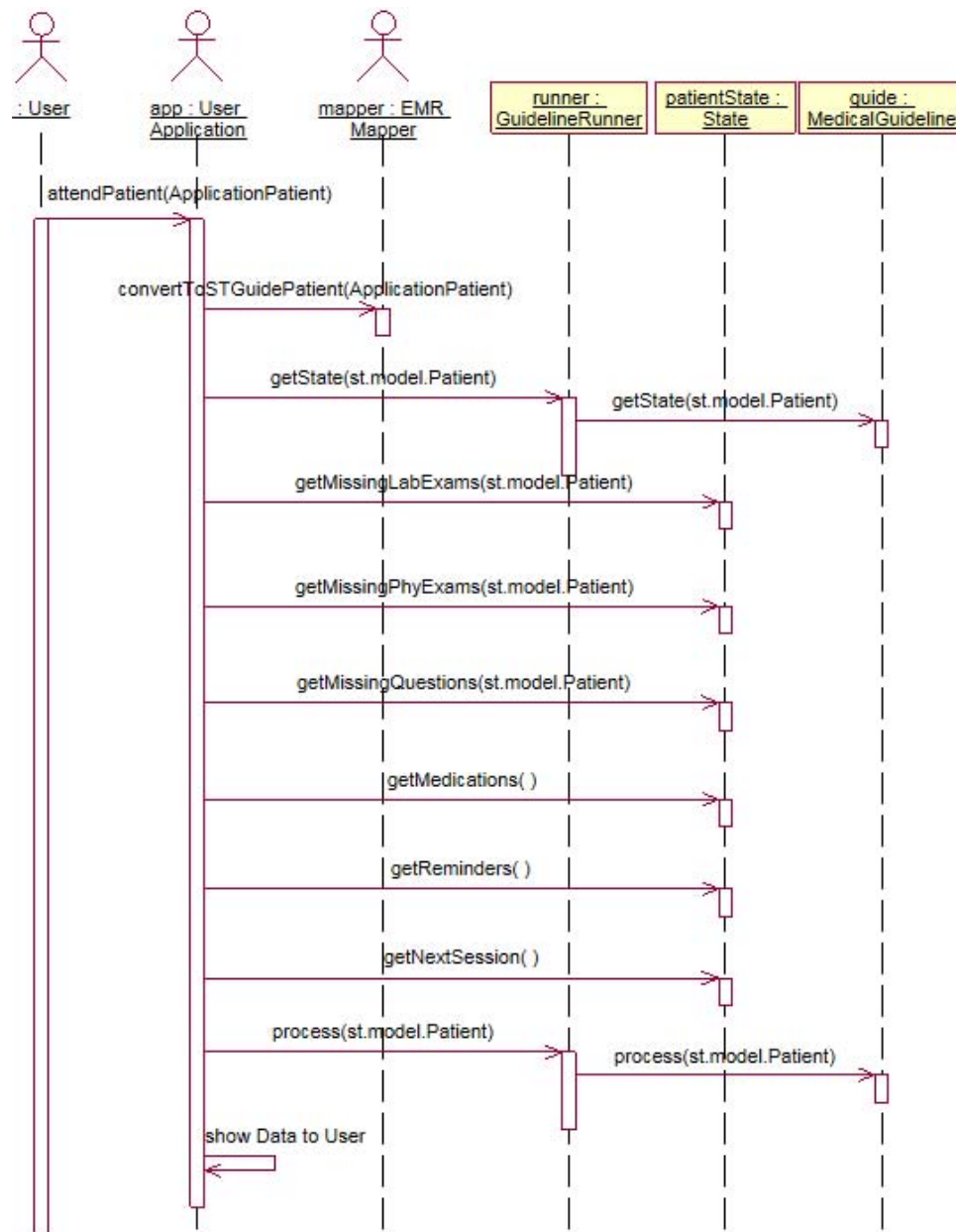


Figura 3.10: Implementação de guia por uma aplicação externa. *EMR Mapper* representa o componente do sistema externo responsável por mapear o prontuário eletrônico médico do sistema externo para o interno do ST-Sistema.

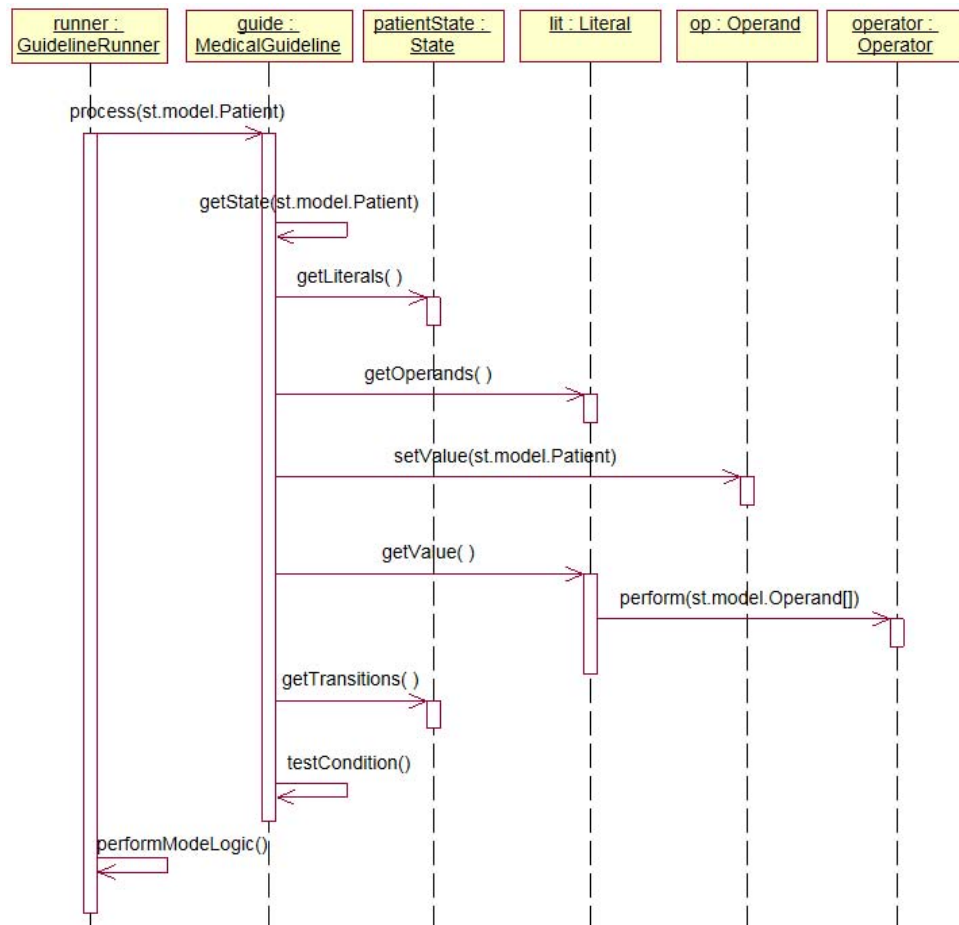


Figura 3.11: Detalhamento do processo de implementação, método *process* de *Medical-Guideline*.

Capítulo 4

Verificação Lógica

Segundo a definição de formalidade na Seção 2.1, ainda que um tratamento seja traduzido para uma linguagem interpretada por computador, ele ainda não pode ser considerado livre de erros. Para tal é preciso que não contenha inconformidades. Como explicado ao longo do Capítulo 2, um guia sem inconformidades é essencial tanto para a sua aplicação eficaz como para a sua aceitação no meio clínico.

Quando o processo ou modo pelo qual o tratamento é modelado não implica (por construção) que o mesmo é livre de erros, torna-se necessária uma etapa final de verificação do tratamento para se poder avaliar ou atestar sua qualidade. Este é o caso do ST-Guide, assim como todos os outros projetos conhecidos na literatura.

Neste Capítulo inicialmente é feita uma revisão de métodos de verificação de guias clínicos encontrados na literatura. Em sequência, apresenta-se o método utilizado pelo ST-Guide.

4.1 Revisão

Existem diferentes trabalhos publicados sobre verificação de guias. [43] verifica incompletude e ambiguidade através de análise lógica e uso de tabelas de decisão. [44] aprimora o trabalho anterior ligando as tabelas de decisão a camadas de meta-dados que podem conter, por exemplo, fontes das informações, custo e qualidade.

[45] fornece uma pequena revisão sobre o assunto e apresenta um método que mapeia guias *Asbru* para regras lógicas e, em seguida, usa as regras junto com uma base de conhecimento para buscar anomalias considerando tanto propriedades dependentes do tratamento como independentes. [45] tem como grande revés o fato de buscar heurística-mente as anomalias, não garantindo a não existência destas no tratamento.

[46] foca sua atenção no aspecto temporal. Falhas nas restrições temporais decorrentes do controle de fluxo e estrutura hierárquica do tratamento são detectadas. No entanto,

se o guia apresenta planos que não podem ser ordenados temporalmente, então sua verificação temporal não é completa. O problema de verificar totalmente guias com atividades sequenciais desordenadas é NP-Difícil[46].

[47], à partir de guias em *Asbru*, usa um provador de teoremas interativo KIV[48] para verificar se o guia satisfaz uma série de propriedades. [49] soma ao trabalho anterior o uso de um verificador simbólico de modelo, o SMV[50].

4.2 Verificação no ST-Guide

Esta primeira verificação de guias implementada para o ST-Guide é capaz de detectar incompletude e ambiguidade. Como este projeto é centrado em modelo, não se espera que os guias resultantes contenham erros graves de abstração. Detectar inconsistência exige metodologias robustas e mais complexas, capazes de analisar o guia como um todo levando em conta aspectos temporais e conhecimento médico específico e independente do tratamento. Por falta de tempo hábil, a verificação completa do sistema será abordada em um trabalho posterior.

No ST-Guide as condições para mudança de estado são expressas em lógica Booleana. Isto, aliado ao fato de ambiguidade e incompletude serem inconformidades locais, ou seja, que ocorrem em um dado instante do tratamento, facilita bastante a verificação lógica. Ambas inconformidades podem ser vistas como um problema de satisfatibilidade (SAT). Devido à maneira como a ST-Linguagem foi modelada, é bastante direta a forma como as expressões booleanas de ambiguidade e incompletude são montadas e avaliadas por um resolvidor SAT.

Na Figura 4.1 é ilustrado um estado qualquer S_0 e suas transições para os estados S_i , mediante as condições c_i , onde $1 \leq i \leq n$ e n é um número inteiro. Considerando este exemplo geral, seguem as seguintes definições:

$$\begin{aligned} incompleto(S_0) &= \text{SAT}(\text{not}(c_1 \text{ or } c_2 \text{ or } \dots \text{ or } c_n)) \\ ambiguo(S_0) &= \text{SAT}(OR_{i=1}^{n-1} OR_{j=i+1}^n (c_i \text{ and } c_j)) \end{aligned}$$

, onde: $\text{SAT}(expr)$ é uma função que retorna se a expressão lógica $expr$ é satisfatível; OR funciona como um somatório, porém denota o uso de um operador lógico *or* ao final da expressão subjacente, quando necessário; *and* representa conjunção lógica; *or* representa disjunção lógica; e *not* é negação lógica.

Quando a função *incompleto* assume valor verdadeiro, então significa que existe uma configuração de literais que torna a expressão definida no termo direito da igualdade satisfatível. Ou seja, existe pelo menos uma configuração de literais da expressão que não ativa nenhuma das transições de S_0 . Portanto, assim fica caracterizado que S_0 está incompleto.

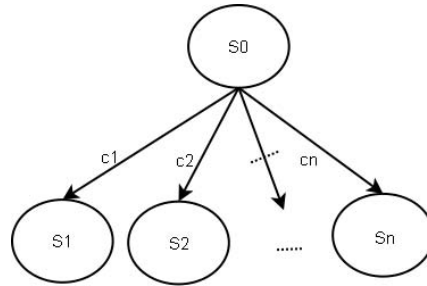


Figura 4.1: Transições de um estado qualquer, S_0 , em função das condições c_i , $1 \leq i \leq n$.

Se a função *ambiguo* assume valor verdadeiro, então existe pelo menos uma configuração de literais que ativa duas transições de S_0 simultaneamente. Assim, fica caracterizado ambiguidade em S_0 .

Na versão atual do *ST-Modeler*, descrito na Seção 5, estas verificações foram implementadas usando objetos da biblioteca de código aberto COBA especificada em [51]. As verificações são executadas automaticamente de forma interativa, assim que o usuário faz uma mudança significativa em um estado.

É oportuno enfatizar que relações de uso não entram na verificação de um estado. Relações de uso refletem uma ambiguidade intencional necessária para uma modelagem mais fácil do guia. Elas permitem que *ST-States* se unam a *Helper-States* evitando uma explosão no número de estados do tratamento.

Capítulo 5

ST-Modeler

O ST-Modeler é a ferramenta de desenvolvimento de guias clínicas construída durante este trabalho para permitir a modelagem gráfica de guias em ST-Linguagem e, ao mesmo tempo, fazer a verificação formal dos mesmos. Esta ferramenta é um *plug-in* para o *Eclipse*, um ambiente de desenvolvimento integrado (IDE) de código aberto desenvolvido pela *International Business Machines* (IBM) e bastante conhecido na comunidade de desenvolvimento de software.

Uma vez instalado o *plug-in*, para modelar um novo guia basta criar um arquivo com extensão *.st*. Este tipo de arquivo fica associado ao ST-Modeler dentro do Eclipse. A Figura 5.1 exemplifica o que acontece quando abrimos um arquivo *.st* e, simultaneamente, destaca e apresenta os componentes do ST-Modeler:

1. *Menu ST-Modeler*: ferramentas de criação e seleção de elementos de modelagem;
2. *Área de Modelagem*: área onde elementos são inseridos e editados;
3. *Controle de Zoom*: permite aproximar ou afastar visualização do modelo.

Com um duplo-clique sobre qualquer elemento, o formulário próprio para sua edição é apresentado. Facilidades como cópia e colagem foram implementados tanto para estados inteiros quanto para seus atributos.

Nas próximas seções deste Capítulo serão cobertos: linguagem visual do ST-Modeler; uma introdução prática à ferramenta; alguns dos experimentos de formalização realizados; e considerações sobre sua usabilidade.

5.1 Linguagem Visual

A linguagem visual usada para a modelagem é bastante simples e está resumida na Figura 5.2.

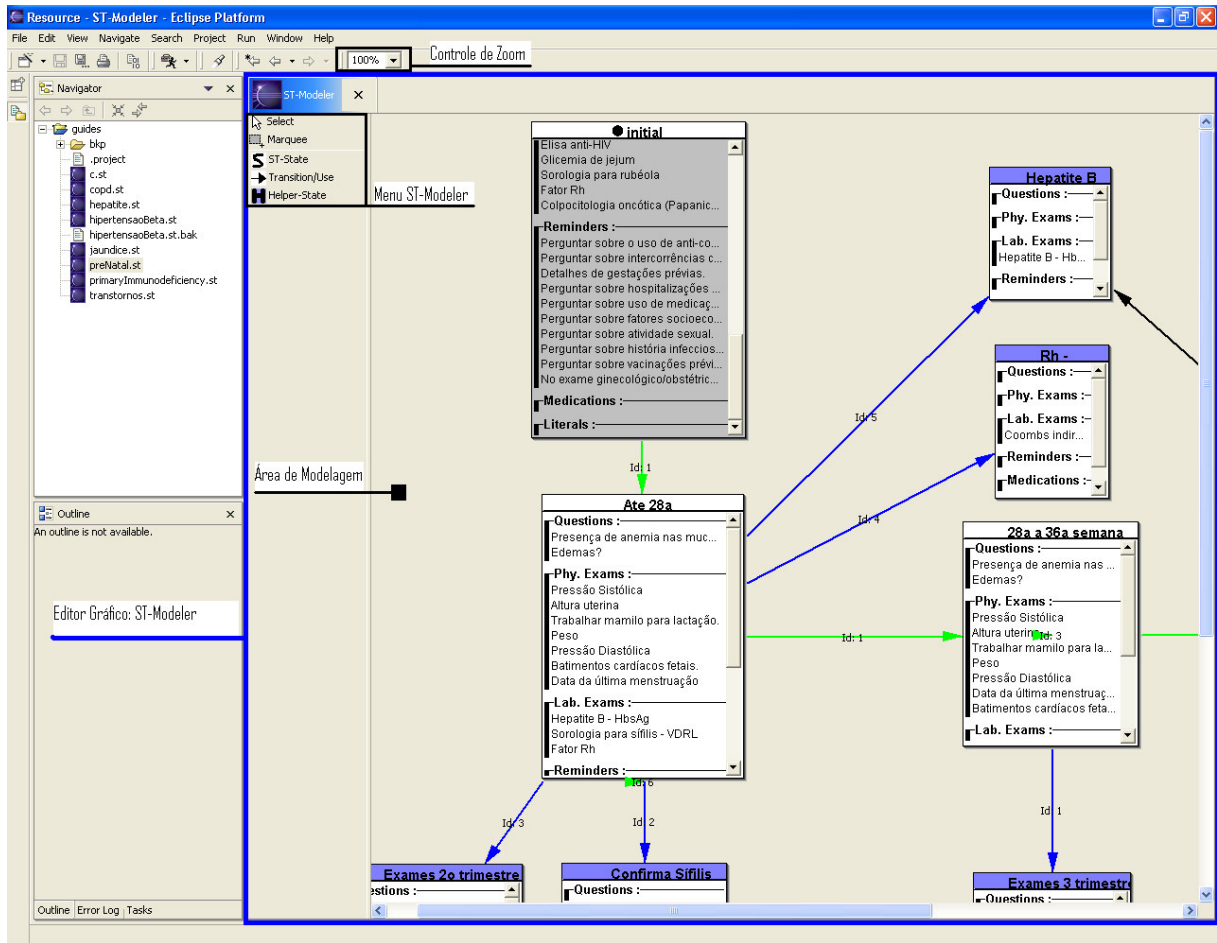


Figura 5.1: ST-Modeler: componentes do *plug-in* no Eclipse.

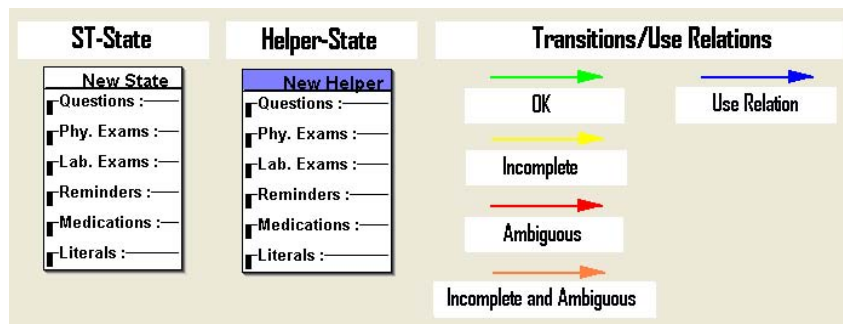


Figura 5.2: Linguagem Visual do ST-Modeler.

5.2 Usando o ST-Modeler

Nesta seção, introduzimos o leitor ao ST-Modeler de forma prática, mostrando o que se pode fazer com a ferramenta.

5.2.1 Estado Inicial

Toda máquina de estados deve possuir ao menos um estado inicial, à partir do qual todos os demais estados podem (ou deveriam poder) ser alcançados. No ST-Guide, trabalha-se com um único estado inicial, o qual obviamente é obrigatório. Por isso ao criar um diagrama no ST-Modeler, automaticamente este já apresenta um estado inicial, *initial*, o qual não pode ser apagado e que também serve para configurar informações gerais do guia, como nome, autor e prefixo dos arquivos de configuração usados.

A Figura 5.3 mostra um diagrama com o estado inicial gerado automaticamente com suas telas de edição de estado e propriedades do guia abertas. Entre as propriedades do guia, configurar o prefixo dos arquivos de configuração permite manter listas de dados usados (exames, perguntas, medicamentos e lembretes) diferentes para cada guia modelado. Isso se tornará mais claro com a leitura das próximas seções.

Na prática, o que diferencia o estado *initial* dos demais é que ele permite configurar as propriedades gerais do guia e não pode ser apagado. Além disso, marca o início do tratamento modelado.

5.2.2 Criar/Copiar/Remover um estado

Para criar um ST-State ou um Helper-State, deve-se clicar em um dos elementos do ST-Modeler *menu* em destaque na Figura 5.4 e, em seguida, clicar na posição da área de modelagem na qual se quer posicionar o estado.

Uma outra forma de criar um estado é copiando outro já existente. Para tal, deve-se selecionar um estado do modelo e apertar a tecla *c* do teclado. Imediatamente, a cópia do estado selecionado aparece transladada a uma certa distância do original. Agora, pode-se mover a cópia para qualquer lugar do diagrama e mudar as características desejadas.

Para remover um estado do diagrama, é preciso selecioná-lo e em seguida apertar a tecla *Delete*.

5.2.3 Editar ST-State e Visualizar sua Situação

Para poder editar um ST-State deve-se dar um duplo-clique sobre este no modelo. A Figura 5.5 mostra a interface de edição que surge na tela. Nesta interface podemos alte-

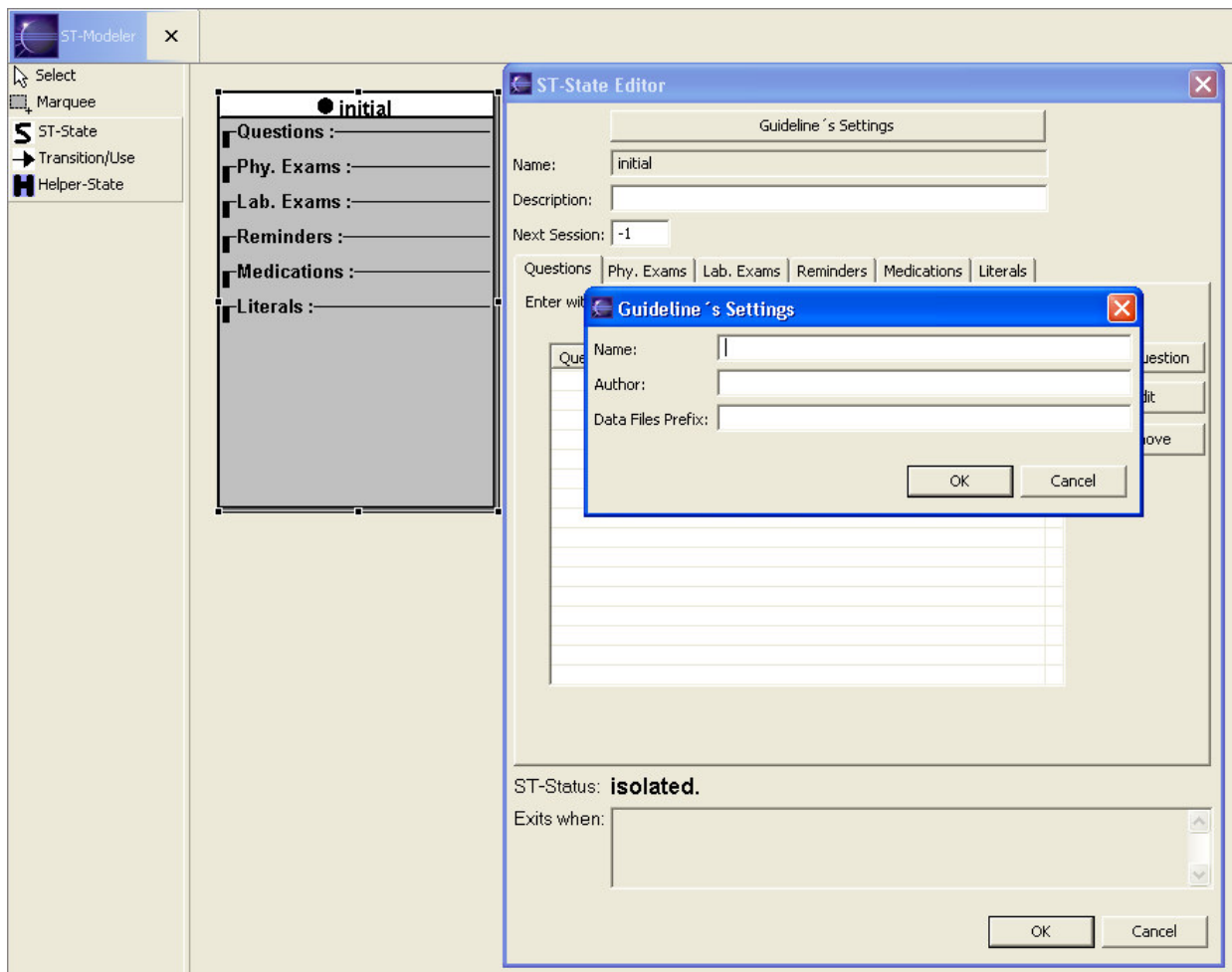


Figura 5.3: ST-Modeler: estado inicial e suas telas de configuração.

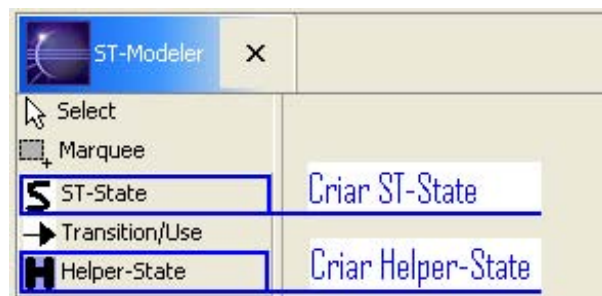


Figura 5.4: Menu ST-Modeler: Criando novo ST-State e Helper-State.

rar qualquer característica do estado: próxima sessão, perguntas, lembretes, medicações, exames físicos e laboratorias, e literais do estado.

Também se pode ver nesta tela qual é a situação atual do estado selecionado em relação à verificação lógica das suas transições. Um estado pode ter situação:

- **Ok**: não apresenta problemas;
- **Incomplete**: incompleto, existe ao menos uma configuração de literais para a qual não há transição especificada neste estado;
- **Ambiguous**: ambíguo, duas ou mais transições podem ser ativadas simultaneamente.

Ainda na janela da Figura 5.5, a aba *Questions* está selecionada e as perguntas a serem feitas no estado podem ser visualizadas em uma lista. Ao lado direito desta lista há três botões (*Add*, *Edit*, e *Remove*) que permitem respectivamente: adicionar perguntas, editar propriedades da pergunta selecionada ou removê-la. Ao selecionar adicionar uma pergunta, uma lista com as perguntas cadastradas é exibida, vide Figura 5.6. Se a pergunta desejada não está disponível, pode-se inserir uma nova pergunta na lista. A validade de uma resposta é configurada através do campo validade (*Validity*). No caso do exemplo da Figura 5.6, respostas para a pergunta dadas a até 30 dias antes da consulta serão consideradas válidas.

Em todas as outras abas (*Phy. Exams*, *Lab. Exams*, *Reminders*, *Medications* e *Literals*), o acesso às informações e funcionalidades segue o mesmo padrão.

A edição de Helper-States ocorre de forma idêntica com a diferença destes não possuem situação, uma vez que não são verificados logicamente.

5.2.4 Modelando Literais

Literais são os elementos básicos da lógica Booleana, variáveis que podem assumir valor verdadeiro ou falso. No ST-Guide essa atribuição de valores ocorre em tempo de execução, mediante a ação de um operador sobre um conjunto de operandos. Em cada estado, devemos definir os literais que serão, por sua vez, usados nas expressões condicionais de transições e relações de uso.

Para adicionar um novo literal a um estado, deve-se clicar em *Add Literal* na aba *Literals* da tela de edição do estado (vide Figura 5.5). Na Figura 5.7 temos a tela de adição de literais. A definição de um literal nesta tela consiste em:

1. selecionar um operador da lista do painel *Operator*;
2. selecionar operandos e, se preciso, configurar seus valores e unidades;

ST-State Editor

Name:

Description:

Next Session:

Questions | Phy. Exams | Lab. Exams | Reminders | Medications | Literals

Enter with the questions that must be asked to the patient at this state.

Question
<input type="checkbox"/> 52 : Presença de anemia nas mucosas? (1 days) 0 : Sim () 1 : Não ()
<input type="checkbox"/> 40 : Edemas? (30 days)

Add Question

Edit

Remove

ST-Status: **OK.**

Exits when:

OK Cancel

Figura 5.5: Interface de usuário para edição de ST-State.

Question ID: 48

Question Text:

- 41 - História pessoal ou familiar de doenças hereditárias/malformações?
- 43 - Avaliação das mamas?
- 44 - Avaliação da tireóide?
- 45 - Avaliação de mucosas?
- 46 - Avaliação dos pulmões?
- 47 - Avaliação do coração?
- 48 - Avaliação do abdome?
- 49 - Avaliação das extremidades?
- 50 - Exposição ambiental ou ocupacional de risco?
- 51 - Generalidade Anterior?
- 52 - Presença de anemia nas mucosas?
- 53 - Uso de drogas lícitas ou ilícitas?
- 54 - Dor abdominal?
- 55 - Constipação?

Validity: 30

Possible Answers:

	Answer 's Text or Value	Unit (optional)
01 -	Bom	NA
02 -	Ruim	NA
03 -		
04 -		
05 -		
06 -		
07 -		
08 -		
09 -		
10 -		

OK Cancel

Figura 5.6: Edição de ST-State: adicionar pergunta. Em destaque, circulados estão os comandos para inserir e remover elementos da lista de perguntas.

3. configurar identificador do literal;
4. descrever sucintamente o que o literal significa.

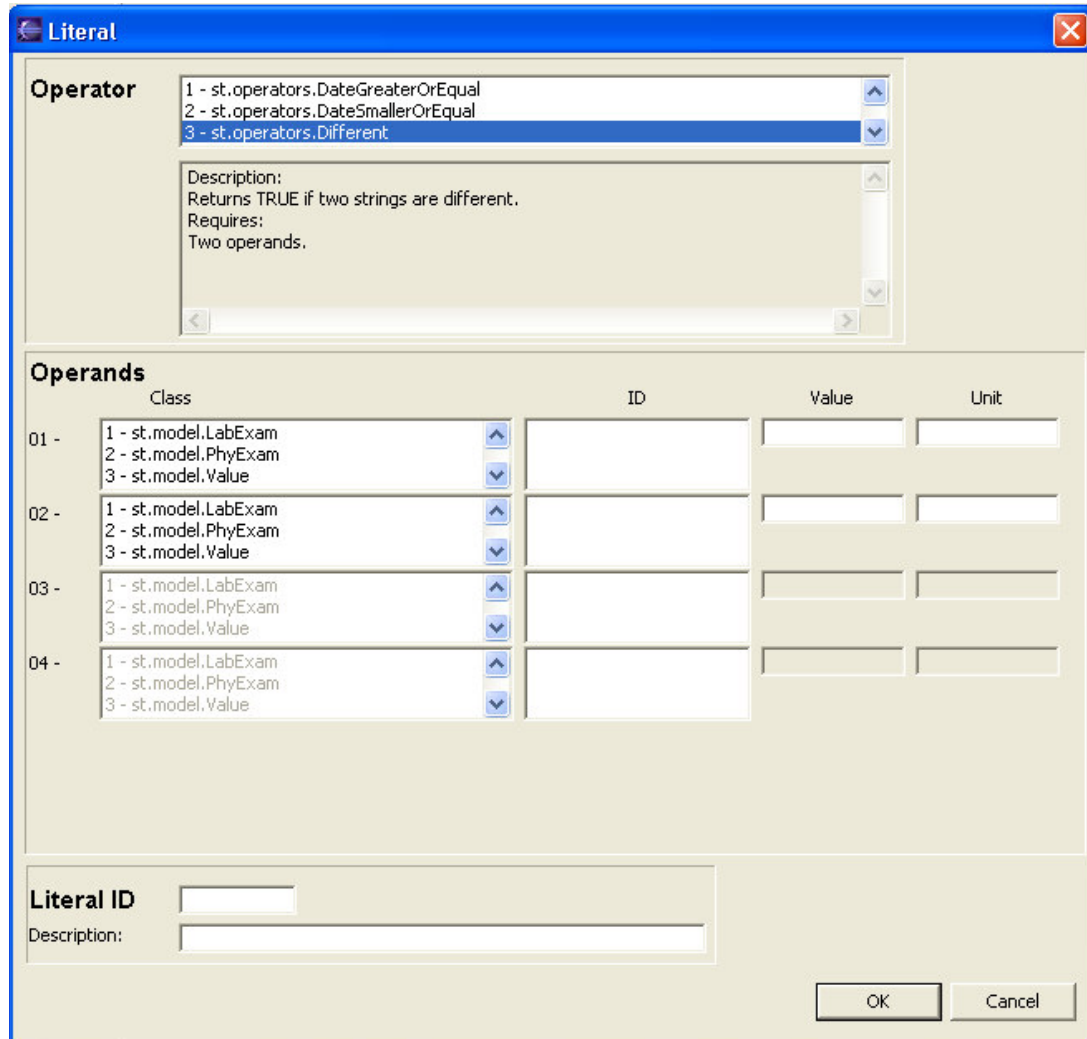


Figura 5.7: ST-Modeler: tela de adição de literal para um estado.

Repare que na seção *Operator* da tela há uma área que descreve como o operador selecionado funciona. Tanto a listagem de operadores e operandos como suas documentações são feitas em arquivos de configuração do ST-Modeler, permitindo fácil adição de operandos e operadores para uso no ST-Modeler.

Visando a tornar mais claro como funciona o processo, em seguida, mostrar-se-ão exemplos práticos de modelagem.

Exemplo 1:

Durante o tratamento de hipertensão, verificar se a pressão do paciente está alta é tarefa fundamental e rotineira. Normalmente a pressão é considerada alta quando (1) a pressão sanguínea diastólica (dbp) está maior que 90 mmHg e (2) a sistólica (sbp) maior que 140 mmHg.

Portanto, saber se o paciente tem pressão alta depende do resultado de dois testes. Como exemplo, modelar-se-á usando o ST-Modeler apenas o primeiro teste necessário:

$$dbp > 90mmHg. \quad (5.1)$$

Para que isso seja possível deve-se pensar na expressão acima em termos de operadores e operandos do ST-Guide. Uma referência rápida são as tabelas 3.1 e 3.2, que, respectivamente, listam os operadores e operandos mais usados.

Entre os operadores, o operador *NumericalGreater* é o adequado para o exemplo. Agora só falta caracterizar os operandos: *pressão diastólica do paciente* e *90 mmHg*. Pressão diastólica do paciente é um exame físico (*PhyExam*) e 90 mmHg é um valor (*Value*).

Assim, em uma notação conceitual bastante parecida com as sentenças atômicas usadas em lógica de primeira ordem[52], temos a seguinte expressão equivalente a 5.1:

$$NumericalGreater(PhyExam(dbp), Value(90mmHg)). \quad (5.2)$$

Configurar um literal que funcione como 5.2 é bastante fácil. Na tela de adição de literais do estado (Figura 5.7), basta selecionar o operador *NumericalGreater* da lista de operadores. Em seguida, seleciona-se como classe do primeiro operando *PhyExam*. Na coluna *ID*, aparece uma lista de todos exames físicos pedidos no estado atual e, então, seleciona-se o exame “Pressão Diastólica”. Caso o exame “Pressão Diastólica” não apareça, então este deverá ser adicionado à lista de exames do estado antes de continuar. Para o segundo operando, deve-se escolher a classe *Value*. ID não estará habilitado. O próximo passo é preencher os campos *Value* (Valor) e *Unit* (Unidade) com 90 e mmHg, respectivamente.

A Figura 5.8 mostra a tela de adição de literal configurada para o exemplo, atribuindo também um identificador e uma descrição para o literal.

Exemplo 2:

Em tratamentos de diversas enfermidades existem informações que o médico precisa saber e que não podem ser respondidas com um exame. O médico precisa perguntar ao paciente para obter a informação. Toda vez que não é possível obter uma informação pelo prontuário do paciente ou por exames médicos, recorre-se a uma pergunta.

Operator

7 - st.operators.NumericalEquals
8 - st.operators.NumericalGreater
9 - st.operators.NumericalGreaterOrEqual

Description:
Returns TRUE if the first operand is numerically greater than the second one.
Requires:
Two operands.

Operands

	Class	ID	Value	Unit
01 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value	1 - Pressão Diastólica		mmHg
02 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value		90	mmHg
03 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value			
04 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value			

Literal ID

Description:

OK Cancel

Figura 5.8: ST-Modeler: modelando o literal equivalente a *pressão diastólica do paciente maior que 90 mmHg*.

Durante um acompanhamento pré-natal, por exemplo, é preciso saber sobre o histórico pessoal ou familiar de doenças hereditárias dos pais. Isso é feito através de perguntas. Entre os quesitos que implicam em um aconselhamento genético estão: hemofilia, distrofia muscular, fibrose cística, coréia de Huntington, acondroplasia, síndrome de Marfan, deficiência de G6PD, fenilcetonúria, retardo mental ou desenvolvimento anormal.

Como modelar isto em termos de ST-Guide? Vamos tratar apenas o caso de hemofilia, já que os outros poderiam ser modelados identicamente ou até mesmo em uma mesma pergunta. Assim precisamos modelar um literal que expresse:

Existe histórico pessoal ou familiar de hemofilia.

Considere que no estado de interesse existe a pergunta “*Há histórico pessoal ou familiar de hemofilia?*” com possíveis respostas “*Sim*” ou “*Não*”. Novamente temos que caracterizar operadores e operandos. Usando a mesma notação conceitual do exemplo anterior, uma solução seria:

Equals(Question(“Há histórico pessoal ou familiar de hemofilia?”), Answer(“Sim”)).

Modelando a expressão acima no ST-Modeler, na tela de adição de literal, seleciona-se o operador *Equals*. Em seguida, como primeiro operando *Question*. Na coluna *ID*, aparecem as perguntas do estado atual e, então, seleciona-se a pergunta “*Há histórico pessoal ou familiar de hemofilia?*”. Como segundo operando, deve-se escolher a classe *Answer*. As possíveis respostas da pergunta selecionada são listadas na coluna *ID*. Então, a resposta “*Sim*” deve ser escolhida. Em seguida atribui-se um identificador e uma descrição para o literal. O resultado pode ser observado na Figura 5.9.

Exemplo 3:

Para tratar icterícia em recém-nascidos tem-se dois fatores decisivos para tomada de decisão: idade do bebê e total de bilirrubina sérica medidos, respectivamente, em horas e mg/dL. Neste exemplo destacamos o uso do operando *Age*, que retorna a idade do paciente em horas, dias, semanas ou anos, dependendo da unidade configurada. Modelaremos um literal equivalente a expressão:

idade do paciente \geq 25 horas.

Com base no Exemplo 1, facilmente percebemos que o operador *NumericalGreater* tendo como segundo operando um *Value* com valor 25 e unidade hora são os candidatos certos para a tarefa. O que muda neste caso é que o primeiro operando é da classe *Age* com unidade hora.

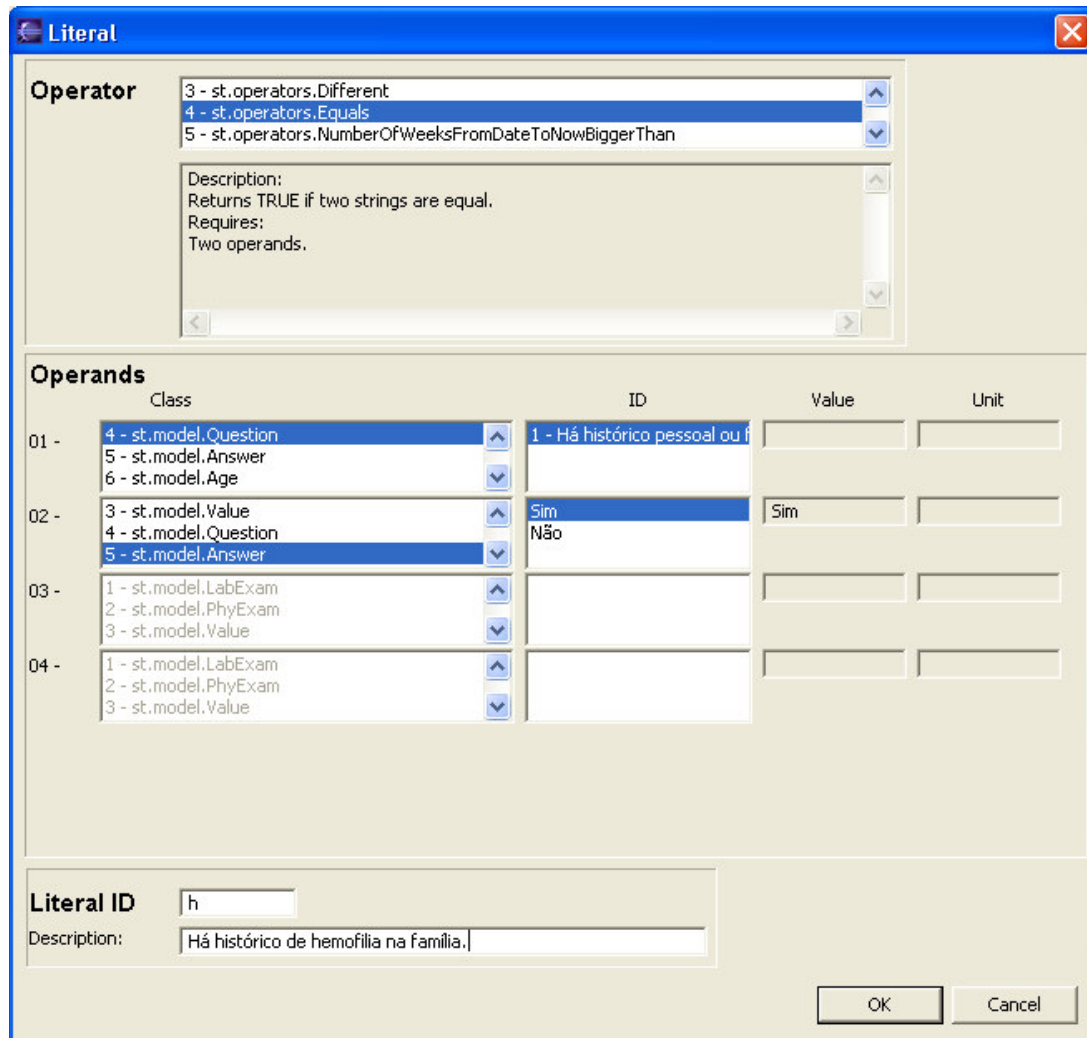


Figura 5.9: ST-Modeler: modelando o literal equivalente a *Existe histórico pessoal ou familiar de hemofilia*.

A configuração do campo *Value* do operando *Age* é irrelevante. O importante é configurar a sua unidade, campo *Unit*, com um dos valores: *hour* (hora), *day* (dia), *week* (semana) ou *year* (ano). Valores diferentes são ignorados, sendo ano a unidade padrão.

A Figura 5.10 mostra o exemplo implementado no ST-Modeler.

Literal

Operator

6 - st.operators.NumericalDifferent
 7 - st.operators.NumericalEquals
 8 - st.operators.NumericalGreater

Description:
 Returns TRUE if the first operand is numerically greater than the second one.
 Requires:
 Two operands.

Operands

	Class	ID	Value	Unit
01 -	4 - st.model.Question 5 - st.model.Answer 6 - st.model.Age		0.0	hour
02 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value		25	hour
03 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value			
04 -	1 - st.model.LabExam 2 - st.model.PhyExam 3 - st.model.Value			

Literal ID

Description:

OK Cancel

Figura 5.10: ST-Modeler: usando o operando *Age* em um literal.

5.2.5 Configurando Transições e Relações de Uso

Para estabelecer uma transição ou uma relação de uso entre estados, deve-se clicar no item do ST-Modeler Menu em destaque na Figura 5.11 e, em seguida, clicar nos estados de origem e destino desejados. Uma aresta preta direcionada da origem para o destino é imediatamente criada. A cor da aresta continua preta até que a sua condição seja configurada. A partir de então, a linguagem visual da Figura 5.2 explica as cores das arestas.



Figura 5.11: Criando uma nova transição ou relação de uso no menu do ST-Modeler.

Visando a configurar a condição com que uma transição ou relação de uso ocorre, deve-se clicar duas vezes sobre a mesma. A tela da Figura 5.12 se abre. Nela aparecem os campos condição (*Condition*), origem (*Source*), destino (*Goes To*), descrição (*Description*) e a aba literais disponíveis (*Available Literals*), onde são listados os literais que podem ser usados na condição da aresta e, simultaneamente, permite adição de novos literais ao estado de origem.

Configurar o campo *Condition* da aresta consiste em escrever uma expressão Booleana usando: (a) literais disponíveis, (b) palavras-chave da Tabela 5.1 e (c) parênteses para indicar precedência de subexpressões, quando necessário.

Tabela 5.1: Palavras-chave usadas nas condições de transições e relações de uso.

Palavra-Chave	Descrição
and	conjunção lógica
or	disjunção lógica
not	negação lógica
true	verdadeiro
false	falso

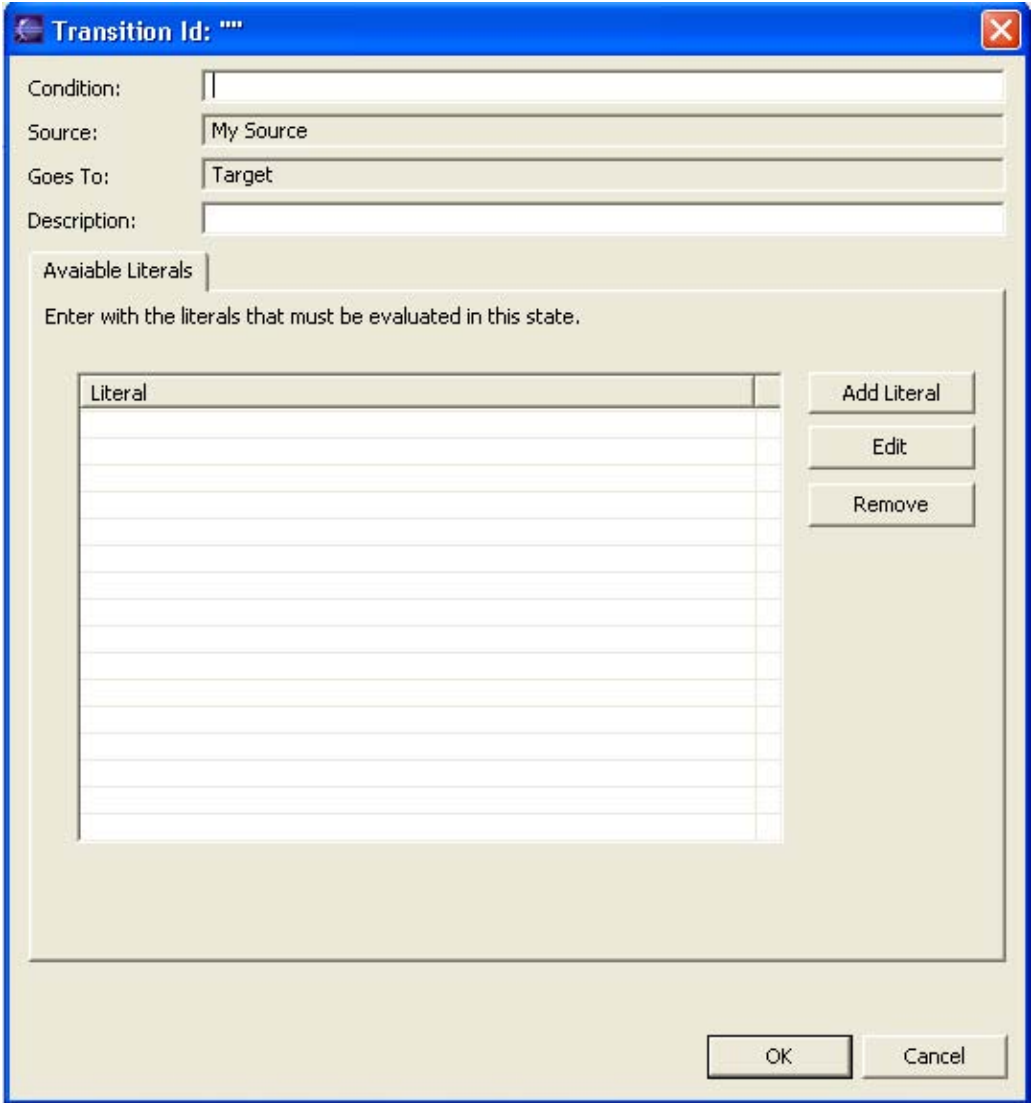


Figura 5.12: Tela de edição de transição ou relação de uso no ST-Modeler.

A tela da Figura 5.13 foi retirada do tratamento de hipertensão baseado em [11] e é um exemplo de configuração de transição. A transição ocorre se o paciente tiver apresentado pressão alta moderada sem outros agravantes, como hemorragia retinal ou diabetes. Caso há agravantes, o paciente é encaminhado para um especialista através de outra transição.

Transition Id: "2"

Condition: j and k and not (b or (s and d and (p or r)) or x)

Source: initial

Goes To: Pressure Measure 1

Description:

Available Literals

Enter with the literals that must be evaluated in this state.

Literal
d : dbp > 100 mmHg
s : sbp > 180 mmHg
b : has diabetes
r : sigs of retinal haemorrhage
p : signs of papilloedema
j : sbp > 140 mmHg
k : dbp > 90 mmHg
x : suspected phaeochromocytoma

Add Literal

Edit

Remove

OK Cancel

Figura 5.13: Exemplo de configuração de transição.

5.3 Experimentos

Aqui são apresentados e comentados guias modelados utilizando o ST-Modeler. A finalidade desses experimentos foi testar a sua usabilidade e, ao mesmo tempo, a expressividade da ST-Linguagem com guias reais.

5.3.1 Assistência Pré-Natal

Neste experimento foi formalizado o guia de assistência pré-natal publicado pelo Projeto Diretrizes [2], uma iniciativa conjunta da Associação Médica Brasileira e Conselho Federal de Medicina. O guia está disponibilizado de forma integral no Apêndice A.

Pode-se perceber uma forte correlação entre a variável tempo, no caso tempo de gravidez, e a assistência prestada. Um roteiro para consultas pre-natais é recomendado: “As consultas deverão ser mensais até a 28^a semana, quinzenais entre 28 e 36 semanas e semanais no termo...”. O tratamento em si tem padrões temporais um pouco diferentes, os quais podem ser divididos em dois grupos: rotina laboratorial e procedimentos médicos nas consultas, apresentados nas tabelas 5.2 e 5.3, respectivamente.

Tabela 5.2: Rotina laboratorial pré-natal.

Período	Exames (* pacientes de alto risco)
1 ^a consulta	papanicolau; tipagem sanguínea e fator RH; sorologia para sífilis, rubéola e toxoplasmose; elisa anti-HIV; antígeno HbsAG (hepatite B); cultura cervical para N. Gonorrhea e C. trachomatous*; hematócrito e hemoglobina; glicemia de jejum; sumário de urina e urocultura; e análise cromossômica fetal por biópsia de vilos corial.
2 ^o trimestre (12 ^a a 24 ^a semana)	teste sérico triplice materno; análise cromossômica fetal por amniocentese; glicemia de jejum; teste oral de tolerância à glicose; e curva glicêmica.
3 ^o trimestre doravante (24 ^a semana em diante)	hematócrito e hemoglobina; e cultura cervical para estreptococos do grupo B*.

As tabelas oferecem um bom resumo sobre o fluxo principal do tratamento, porém, não revelam vários detalhes presentes no guia e que tiveram que ser modelados. Outras recomendações que não estão contempladas nas tabelas referem-se à suplementação de vi-

taminas e nutrientes na gestação e recomendações gerais quanto a rotina ultra-sonográfica e avaliação da vitalidade fetal no pré-natal.

Tabela 5.3: Procedimentos médicos nas consultas pré-natal.

Período	Procedimentos
1 ^a consulta	questionar história médica da paciente e família (extensa lista de componentes); pesquisar sintomas relacionados à gravidez; aconselhamento genético; medir peso, altura, pressão arterial; avaliar mucosas, tireóide, mamas, pulmões, coração, abdome e extremidades da gestante; no exame ginecológico, avaliar a genitália externa, vagina, colo uterino; no toque bidigital, quando necessário, o útero e anexos.
Demais	mensurar pressão arterial; à partir da 12 ^a semana medir altura uterina; verificar a presença de edemas, anemia nas mucosas; auscultar os batimentos cardíacos fetais; e trabalhar mamilo para lactação.

Em vista da lógica temporal exigida pelo guia, o operador *NumberOfWeeksToDateBiggerThan* foi implementado. Ele recebe dois operandos, conta o número de semanas entre a data do primeiro operando e a data atual e, então, verifica se este número é maior ou igual ao segundo operando. Com este operador foi possível implementar todo o fluxo temporal do tratamento. A data da última menstruação é usada como estimativa do início da gravidez, podendo ser posteriormente atualizada pela data sugerida no exame de ultrassom.

O modelo desenvolvido aparece nas Figuras 5.14 e 5.15, sendo que o ponto de junção entre as figuras é marcado por reticências. Foram usados ST-States para marcar o roteiro de consultas, que é o padrão temporal primário do tratamento. Devidamente acoplados, o resto do tratamento é representado com Helper-States, o que facilitou imensamente a sua expressão.

Vale comentar que Helper-States podem ser inclusive usados para agrupar logicamente uma parte do modelo. Com o de pré-natal, isso ocorre na avaliação de risco da gravidez. Repare que o estado inicial está ligado com um Helper-State chamado *Avalia Alto-Risco*, o qual está ligado a outros quatro *helpers* que formam grupos lógicos de fatores de alto risco para gravidez. A união entre os estados mencionados sempre ocorre, pois a condição das relações de uso é uma tautologia (expressão: *true*). Dessa forma, sempre que uma paciente estiver no estado inicial ela será interrogada a respeito de todos os fatores que envolvem avaliação de alto risco para gravidez. Ao mesmo tempo, os fatores estão organizados e é

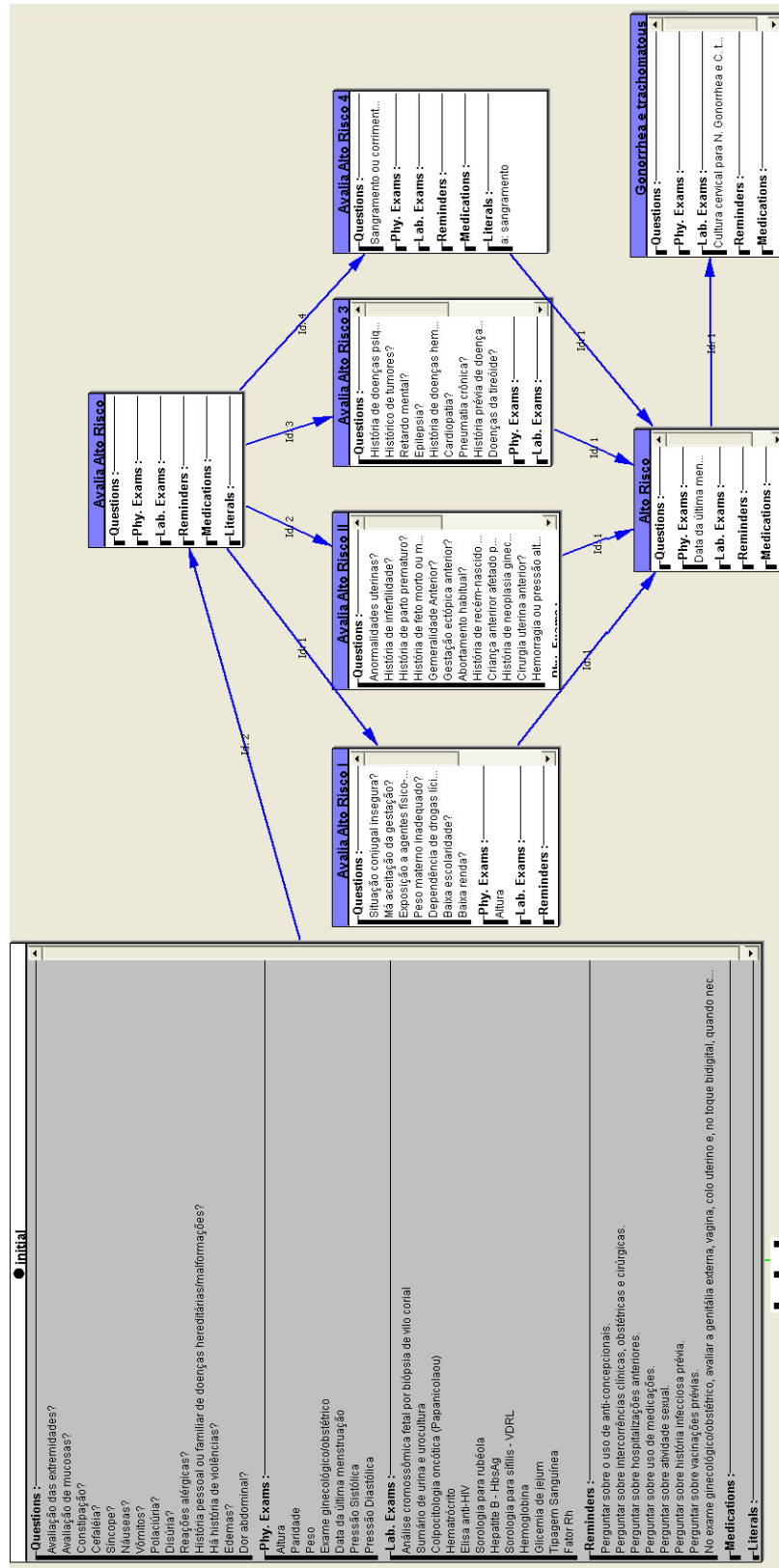


Figura 5.14: Modelo do tratamento de pré-natal: parte um.

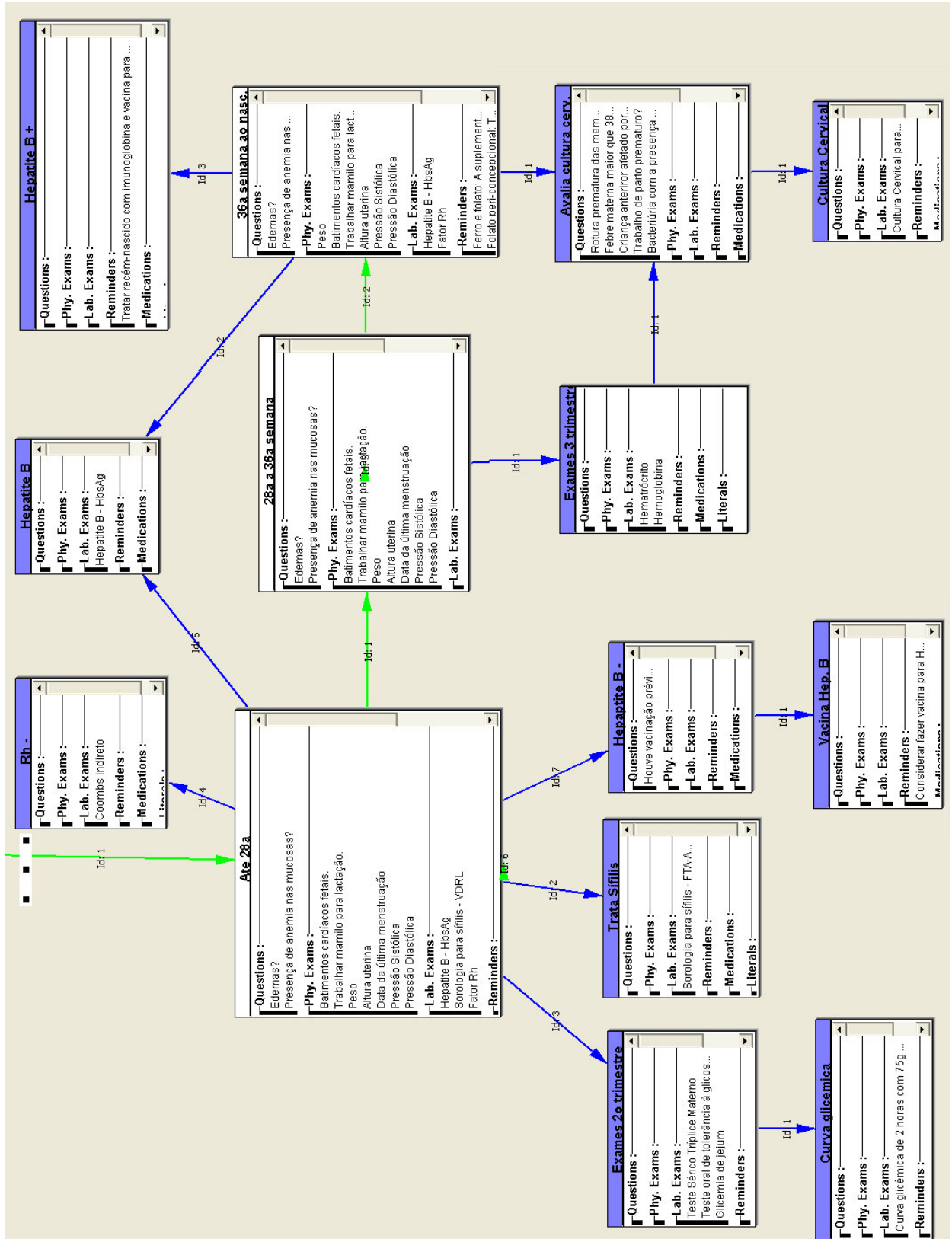


Figura 5.15: Modelo do tratamento de pré-natal: parte dois.

possível fazer uma fácil manutenção do modelo.

Entre as dificuldades encontradas, pode-se citar a falta de definição (ou abstração) para *gravidez de alto risco*. Pesquisando na internet foi possível obter uma definição, vide apêndice D, e adaptar o tratamento com a inserção de novas perguntas. No entanto, o ideal era que uma definição formal estivesse disponível no guia ou em alguma referência do mesmo. Como o objetivo aqui é usabilidade e expressividade da ferramenta, a definição pesquisada é suficiente.

Abstração é o problema mais grave deste guia. Na primeira consulta por exemplo, há uma longa lista de componentes da história médica da paciente que precisam ser interrogados. Além destes, questões sobre a atual gravidez também são recomendadas. O problema é que não há uma relação direta no texto entre as perguntas e as ações que devem ser tomadas. Apenas ao pesquisar uma definição para gravidez de alto risco, muitas perguntas tiveram seu propósito revelado. Isso acontece também com alguns exames da rotina laboratorial, cuja análise e posterior tomada de decisões ficam implícitas ao leitor.

Origem dos Helper-States

Durante a formalização deste tratamento o conceito de Helper-States se solidificou e foi implementado. O problema revelou-se ao modelar o estado *Ate 28a*, vide Figura 5.15, que dá sequência à fase inicial do tratamento encapsulando os exames do segundo semestre em consultas mensais. Neste estado, deve-se analisar os resultados dos exames de fator Rhesus (Rh), sífilis e hepatite B. Para cada um dos exames, independentemente, há procedimentos específicos adotados em função dos resultados.

Caracteriza-se um exemplo claro de estado bem definido que muda em função de variáveis independentes. Modelar este relacionamento sem usar Helper-States implica criar um novo estado para cada uma das possíveis combinações entre as variáveis (rever Seção 3.2.1). A Figura 5.16 é um rascunho de como ficaria a modelagem desta parte do tratamento sem Helper-States. Por uma questão de simplicidade e até mesmo legibilidade do diagrama, o rascunho não contempla os exames do segundo semestre e nem a vacinação para hepatite B.

5.3.2 Hipertensão Arterial

Hipertensão arterial é uma das doenças mais frequentes nas sociedades modernas e se caracteriza por uma elevação persistente da pressão arterial do paciente:

- pressão diastólica(ou mínima) > 90 mmHg e
- pressão sistólica (ou máxima) > 140 mmHg.

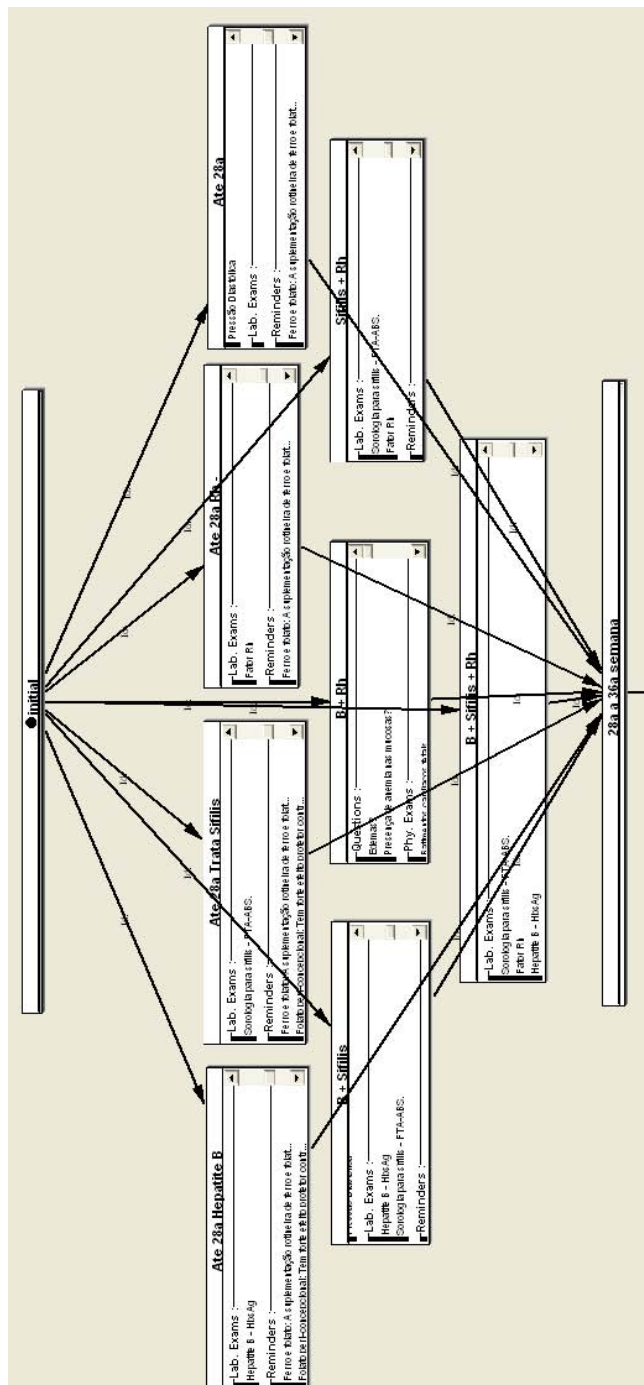


Figura 5.16: Esboço de modelagem do estado *Ate 28a* de assistência pré-natal sem Helper-States.

Além de ser uma doença que atinge porcentagens significativas da população, a hipertensão causa complicações como doença cerebrovascular, doença arterial coronária, insuficiência cardíaca, insuficiência renal crônica e doença vascular de extremidades [53]. O custo da doença e suas complicações são bastante altos também. Segundo [53], em 1998 o Ministério da Saúde teve um gasto de 475 milhões de reais com internações relativas à doença.

Neste experimento, modelamos um guia de referência rápida para controle de hipertensão em adultos recebendo tratamento primário [11]. O guia fornece de forma rápida e sucinta uma série de recomendações sobre medição de pressão, mudanças no estilo de vida do paciente, quando encaminhar o paciente para especialista, e intervenção farmacêutica, entre outras. Um fluxograma especifica a sequência temporal em que recomendações se aplicam e um algoritmo para o tratamento medicamentoso de novos pacientes diagnosticados é apresentado. Ambos foram essenciais para a modelagem.

[11] é um guia sucinto, organizado e fácil de entender. As dúvidas que surgiram sobre diagnóstico de hipertensão e encaminhamento para especialista puderam ser sanadas com o fluxograma e suas anotações laterais (página 5). No entanto, alguns pontos não ficaram claros, provavelmente por abstração de conhecimento médico. Por exemplo, para diagnosticar hipertensão é preciso observar (medir) pressão alta no paciente ao menos em três consultas consecutivas, ou seja, uma rotina cuidadosa. Já para considerar que o paciente não é hipertenso, basta uma única medição de pressão normal. Essa medição poderia ter sido influenciada por algum fator desconhecido e casual, como a ingestão de um chá na casa de um amigo, por exemplo.

O fluxo de consultas de medição e diagnóstico foram modelados usando ST-States e transições. Recomendações reusadas em vários ST-States, como as de medição de pressão, foram modularizadas nos Helper-States *Standardize Environment*, *Postural Hypotension* e *Post. Hyp. Reminder*, vide figuras 5.17 e 5.18, que mostram o modelo. Recomendações de mudanças no estilo de vida e dieta também foram agrupadas usando o mesmo conceito.

O ST-State *Drug Therapy* é o elemento da modelagem que ao mesmo tempo concentra maior elegância e complexidade. O paciente fica neste estado até que sua pressão atinja a meta do tratamento: abaixo de 90/140 mmHg. Enquanto o mesmo se encontra hipertenso, em função da sua idade, descendência racial, remédios em uso e pressão arterial corrente Helper-States se unem de forma a moldar o próximo passo do tratamento medicamentoso, que pode ser a adição de um novo medicamento ou em última instância a recomendação: *Consider specialist advice* (Considere ajuda de especialista). Uma vez que o paciente atinge a meta do tratamento, ele é encaminhado para o estado *Annual Review*, que representa um acompanhamento anual.

O que tornou possível esta modelagem bem sucedida do tratamento medicamentoso foi a implementação do operador *TakingMedicine*. Com as respectivas classes entre pa-

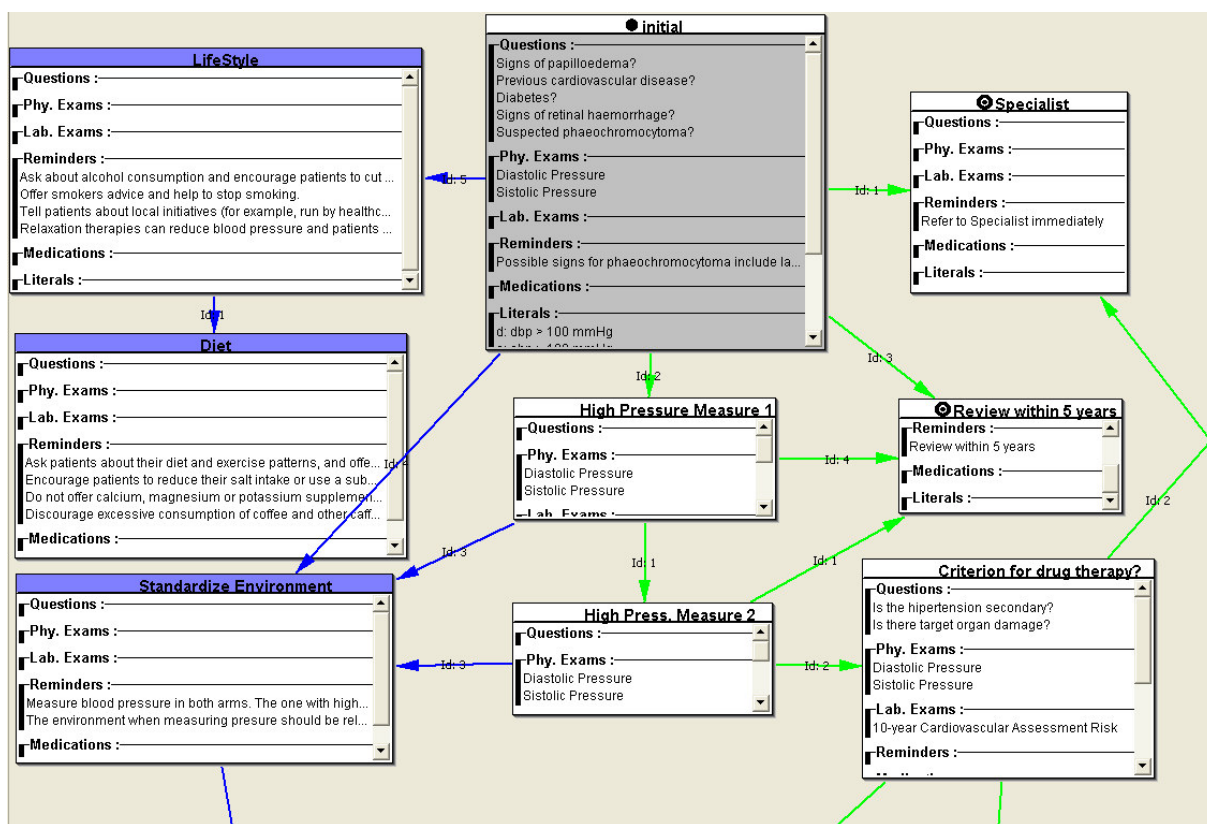


Figura 5.17: Modelo do experimento de hipertensão: parte um.

rentêses, este operador possui três operandos: paciente (Patient), remédio (Medication) e quantidade de dias (Value). Seu retorno é verdadeiro, se o paciente está tomando a medicação especificada no mínimo pela quantidade de dias configurada e, falso, caso contrário. *TakingMedicine* com certeza será bastante reusado em outras modelagens.

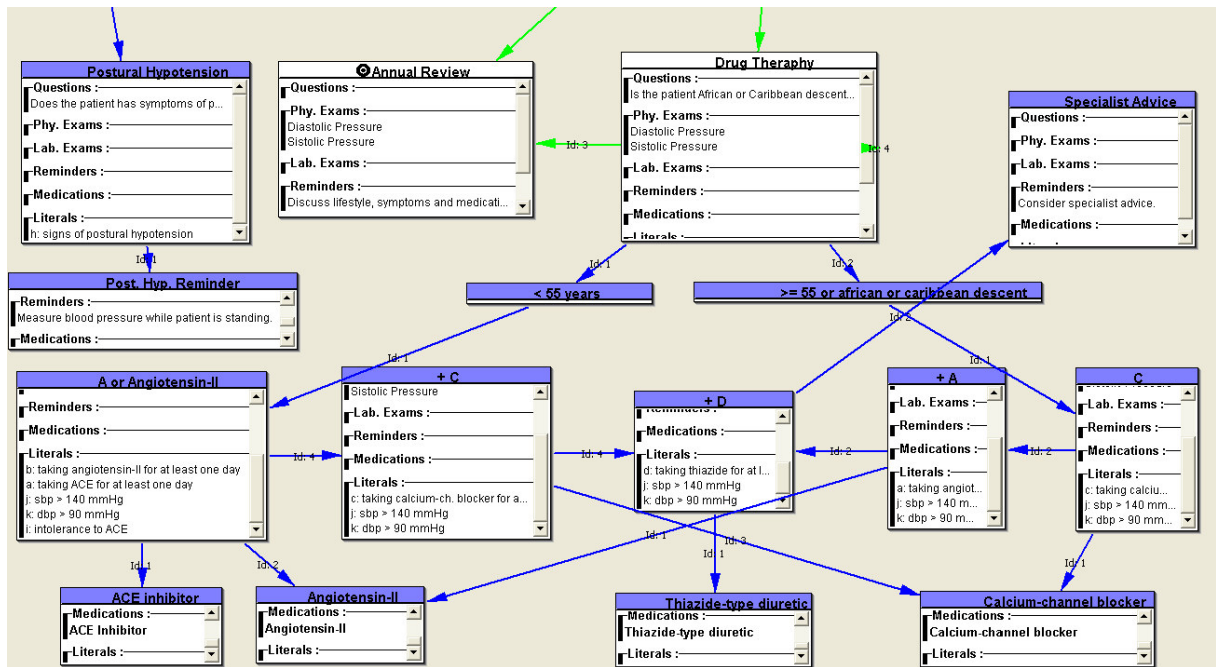


Figura 5.18: Modelo do experimento de hipertensão: parte dois.

5.4 Considerações sobre Usabilidade

Modelar com o ST-Modeler se mostrou uma tarefa bastante fácil e intuitiva para o autor e colegas com formação em ciência da computação. As dificuldades em si foram sempre impostas pelo guia modelado e não pela ferramenta. Noções sobre máquina de estados, lógica Booleana e funções são considerados pré-requisitos para modelar com o ST-Modeler.

Para ter noção se um usuário da área médica seria capaz de usar a ferramenta, um experimento de uma hora consistindo de uma aula introdutória (30 min) seguida de dois exercícios práticos (30 min) foi realizado com um colega médico psiquiatra. Ele foi escolhido porque além de ser médico tinha grande desenvoltura com computadores e era também estudante e pesquisador de inteligência artificial aplicada em medicina. Os exercícios estão disponíveis no Apêndice E.

No tempo planejado, com a ajuda do autor, ele conseguiu concluir os dois exercícios. Infelizmente, não houve disponibilidade de tempo para fazermos mais experimentos ou tentarmos modelar um guia em conjunto.

A experiência adquirida não permite fazer generalizações sobre a curva de aprendizagem do ST-Modeler ou mesmo avaliar sua usabilidade. Portanto, apenas serão transmitidas noções claras que surgiram ao observar e interagir com o usuário de teste:

1. usuário não demonstrou dificuldade em interagir com o diagrama de modelagem ou interface gráfica da ferramenta;
2. abstração do funcionamento de uma máquina de estados foi assimilado. Contudo, transformar o texto em um modelo formal (máquina de estados) foi a maior dificuldade observada;
3. processo de modelagem de literais e expressão de condições com lógica Booleana foi absorvido;
4. usuário entendeu os problemas de incompletude e ambiguidade que surgiram nos exercícios e com ajuda foi capaz de saná-los.

Capítulo 6

Discussão, Conclusões e Futuros Trabalhos

Guias de conduta médica, desde que sejam bem elaborados, comprovadamente melhoram a qualidade média do atendimento médico. Frequentemente médicos não estão acostumados com guias no formato textual e por isso não os aplicam adequadamente. O grande aumento na publicação de guias acaba piorando a situação, tornando o volume de conhecimento a ser assimilado cada vez maior e mais inacessível.

A implementação de guias usando sistemas computacionais surgiu como solução para o problema, tornando o acesso a recomendações rápido e transparente para o clínico, que pode usufruir do sistema diretamente no local de atendimento.

Antes de implementar um guia em um sistema computacional é necessário formalizá-lo, isto é, transformá-lo de linguagem textual e informal para uma linguagem formal. Neste contexto, surgiram duas grandes vertentes de formalização: centrada em modelo (formalização descendente) e centrada em documento (ascendente). Ambas formalizações podem produzir modelos executáveis por um programa, todavia, é mais difícil executar modelos formalizados ascendentemente.

ST-Guide e os projetos de implementação computacional de guias considerados mais importantes na literatura foram apresentados. Entre eles: MGT, GEM, sintaxe Arden, *PROforma*, Asbru, EON e Prodigy.

Dentre os projetos centrados em modelo, sintaxe Arden, *PROforma*, EON e Asbru vêem o problema de implementação de guias como especialização do problema clássico de planejamento em inteligência artificial. EON e Asbru usam planos-esqueleto para refinar planos em tempo de execução acessando bases de conhecimento médico.

Prodigy e ST-Guide assumem que um paciente sempre pode ser classificado em um estado e modelam encontros de atendimento primário entre paciente e clínico, o que acaba levando à adoção de um formalismo baseado em máquina de estados para representar

guias. Apesar de conceitualmente parecidos, possuem diferenças significativas em itens diversos como: modelo de consulta ou encontro; hierarquização de guias; verificação lógica; possíveis plataformas-base de execução; e modos de interação.

Durante este trabalho a ST-Linguagem tornou-se independente da tecnologia de execução e o ST-Sistema um arcabouço ou *framework*. Novos operadores e operandos podem ser implementados à medida em que for preciso e usados em guias sem qualquer impacto no ST-Sistema. Operações sobre imagens, sons e vídeos já são possíveis com a infra-estrutura atual. Basta implementar operadores com os algoritmos desejados.

Dado que um operando pode funcionar como uma função (*Age*, por exemplo), a expressividade máxima das expressões condicionais de transições e relações de uso equivale-se no mínimo à de lógica de segunda ordem. No mínimo, porque operadores podem receber vários parâmetros e, por consequência, encapsular várias funções. Por motivo de simplicidade, o ideal é que um operador encapsule apenas uma função. É preciso mais pesquisa exploratória para saber se funções de ordem três ou superior aparecem no domínio clínico de atendimento primário. Até agora, entre os vários experimentos realizados, estes casos jamais ocorreram.

Depois que o ST-Sistema foi implementado e começaram os experimentos de formalização, ST-Modeler e verificação lógica surgiram como passos naturais do projeto. O ambiente de desenvolvimento trouxe visibilidade sobre que estava sendo modelado, enquanto a verificação lógica prevenia erros imperdoáveis em um modelo (como incompletude e ambiguidade).

Considerando que a grande maioria dos problemas em um guia são locais, a técnica de verificação do ST-Guide oferece uma abordagem simples e poderosa que encontra a maioria dos erros possivelmente presentes. Em nenhum outro trabalho encontramos uma abordagem para checagem de ambiguidade e incompletude tão simples, que pudesse ser resumida em duas linhas. No entanto, a verificação completa é um objetivo e uma necessidade que deverá ser alcançada em um trabalho futuro.

O ST-Modeler cumpre competentemente a tarefa para a qual foi criado: modelagem gráfica de guias em ST-Linguagem com verificação lógica do modelo. Era desejável que ele pudesse ser usado por clínicos especialistas, porém este não era um objetivo. Sua usabilidade foi analisada informalmente via experimentos de modelagem realizados pela equipe do ST-Guide e por um médico pesquisador na área de inteligência artificial em medicina. Por enquanto, a maneira mais eficiente de produzir um guia usando a ferramenta é através do trabalho conjunto entre especialista médico e modelador/programador.

Como futuros trabalhos, dando continuidade ao projeto, sugerimos: hierarquização de guias; implementação dos modos de interação aprendiz e auditor; novos operadores e operandos para o ST-Sistema; verificação lógica completa; atribuição de nível de força da evidência para cada recomendação do guia; e versionamento de guias.

Referências Bibliográficas

- [1] Institute of Medicine. From development to use. *Washington D.C., National Academy Press*, 1992.
- [2] C.A. Alencar Jr. Assistência pré-natal. *Projeto Diretrizes, Federação Brasileira das Sociedades de Ginecologia e Obstetrícia, Associação Médica Brasileira e Conselho Federal de Medicina*, 2001.
- [3] J.M. Grimshaw and I.I. Russell. Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. *Lancet*, 342(8883):1317–22, 1993.
- [4] L.H. Thomas, E. Mccoll, N. Cullum, N. Rousseau, and Soutter J. Clinical guidelines in nursing, midwifery and the therapies: a systematic review. *Journal of Advanced Nursing*, 30(1):40–50, 1999.
- [5] S.H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. Clinical guidelines: Potential benefits, limitations, and harms of clinical guidelines. *BMJ*, 318:527–530, 1997.
- [6] A. Audet, S. Greenfield, and M. Field. Medical practice guidelines: current activities and future directions. *Ann Intern Med*, 113(9):709–14, 1990.
- [7] M.C. Vissers, A. Hasman, and C.J. Linden. Impact of a protocol processing system (protoview) on clinical behaviour of residents and treatment. *Int J Biomed Comput*, 42:143–50, 1996.
- [8] A. Hibble, D. Kanka, D. Pencheon, and F. Pooles. Guidelines in general practice: the new tower of babel? *BMJ*, 317(7162):862–3, 1998.
- [9] K. Kawamoto, C.A. Houlihan, E.A. Balas, and D.F. Lobach. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ*, 330(7494):765, 2005.
- [10] J. Wainer, A.M. Monteiro, M.S. Anção, and D. Sigulem. St-guide: a state/transition representation model for clinical guidelines. *Technical report, IC-Unicamp*, 2002.

- [11] Newcastle Guideline Development, Research Unit in collaboration with British Hypertension Society, and the National Collaborating Centre for Chronic Conditions. Hypertension: management of hypertension in adults in primary care. *NICE clinical guideline*, 34, 2006.
- [12] R. Grol, J. Dahluisen, S. Thomas, C. Veld, G. Rutten, and H. Mokkink. Attributes of clinical guidelines that influence use of guidelines in general practice: observational study. *BMJ*, 317(7162):858–61, 1998.
- [13] M. Geldof, A. Teije, F. Harmelen, M. Marcos, and P. Votruba. Informal and formal medical guidelines: Bridging the gap. *Artificial Intelligence in Medicine, Proceedings Lecture Notes in Artificial Intelligence*, (2780):173–178, 2003.
- [14] M. Ruzicka and V. Svatek. Mark-up based analysis of narrative guidelines with the stepper tool. *Symposium on Computerized Guidelines and Protocols*, 101:132–6, 2004.
- [15] R.N. Shiffman, P. Shekelle, J.M. Overhage, J. Slutsky, J. Grimshaw, and A.M. Deshpande. Standardized reporting of clinical practice guidelines: a proposal from the conference on guideline standardization. *Ann Intern Med*, 139:493–498, 2003.
- [16] National Health and Medical Research Council. A guide to the development, implementation and evaluation of clinical practice guidelines. *NHMRC*, www.nhmrc.gov.au/publications/synopses/cp65syn.htm (acessado 6 Fev 2006), 1998.
- [17] J.S. Burgers, R.P.T.M. Grol, J.O.M. Zaat, T.H. Spies, A.K. Bij, and H.G.A. Mokkink. Characteristics of effective clinical guidelines for general practice. *British Journal of General Practice*, 53:15–19, 2003.
- [18] T.M. Shaneyfelt, M.F. Mayo-Smith, and J. Rothwangl. Are guidelines following guidelines? the methodological quality of clinical practice guidelines in the peer-reviewed medical literature. *JAMA*, 281(20):1900–5, 1999.
- [19] R.N. Shiffman, B.T. Karras, A. Agrawal, R. Chen, L. Marenco, and S. Nath. Gem: A proposal for a more comprehensive guideline document model using xml. *J Am Med Inform Assoc*, (7):488–998, 2000.
- [20] V. Svatek, M. Ruzicka, and T. Kroupa. Guide-x - a step-by-step, markup-based approach to guideline formalisation. *First European Workshop on Computer-based Support for Clinical Guidelines and Protocols*, 139, 2001.
- [21] R.N. Shiffman and A. Agrawal. Evaluation of guideline quality using gem-q. *Proceedings MEDINFO*, pages 1097–1101, 2001.

- [22] P. Gershkovi and R.N. Shiffman. An implementation framework for gem encoded guidelines. *Proc AMIA Symp*, pages 204–8, 2001.
- [23] R.N. Shiffman, A. Agrawal, A.M. Deshpande, and P. Gershkovich. An approach to guideline implementation with gem. *Proceedings MEDINFO*, pages 271–5, 2001.
- [24] T.A. Pryor and G. Hripcsak. The arden syntax for medical logic modules. *Int J Clin Monit Comput*, 10(4):215–24, 1993.
- [25] T.A. Pryor and G. Hripcsak. Sharing mlm’s: an experiment between columbia-presbyterian and lds hospital. *Proc Annu Symp Comput Appl Med Care*, pages 399–403, 1993.
- [26] Guideline modelling methods and technologies. *Open Clinical*, <http://www.openclinical.org> (acessado 05 Fev 2008).
- [27] M. Peleg, A.A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E.H. Shortliffe, and R.A. Greenes. Glif3: the evolution of a guideline representation format. *Proceedings of American Medical Informatics Association Symposium*, pages 645–9, 2000.
- [28] M. Peleg and R. Kantor. Approaches for guideline versioning using glif. *AMIA Annu Symp Proc*, pages 509–513, 2003.
- [29] J. Fox, N. Johns, and A. Rahmansadeh. Disseminating medical knowledge: the proforma approach. *Artificial Intelligence in Medicine*, 14(1-2):157–81, 1998.
- [30] S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. *Proceedings of the 7th Workshop on Knowledge Engineering: Methods and Languages (KEMML-97)*, Milton Keynes, UK, 1997.
- [31] S.W. Tu and M.A. Musen. Modeling data and knowledge in the eon guideline architecture. *MEDINFO*, Pt 1(10):280–4, 2001.
- [32] J.H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.
- [33] M.A. Musen, S.W. Tu, A.K. Das, and Y. Shahar. Eon: A component-based approach to automation of protocol-directed therapy. *J Am Med Inform Assoc*, 3(6):367–88, 1996.

- [34] P.D. Johnson, S. Tu, N. Booth, B. Sugden, and I.N. Purves. Using scenarios in chronic disease management guidelines for primary care. *Proc AMIA Symp*, pages 389–93, 2000.
- [35] David Chappell. *Understanding Activex and Ole*. Little, Brown & Company, 1996.
- [36] C. Billa. Diferentes modos de interação num sistema automático de condutas clínicas para o atendimento primário. *Dissertação de Mestrado, IC-Unicamp*, 2004.
- [37] C. Billa and J. Wainer. Diferentes modos de interação num sistema automático de condutas clínicas para o atendimento primário. *CBIS - Congresso Brasileiro de Informática em Saúde*, 2004.
- [38] M. Versiani. Diagnóstico e tratamento de transtornos de ansiedade. *Projeto Diretrizes, Associação Brasileira de Psiquiatria, Associação Médica Brasileira e Conselho Federal de Medicina*, 2001.
- [39] American Academy of Pediatrics, Provisional Committee for Quality Improvement, and Subcommittee on Hyperbilirubinemia. Practice parameter: Management of hyperbilirubinemia in the healthy term newborn. *Pediatrics*, 94(4):558–565, 1994.
- [40] G. Booch, J. Rumbaugh, and I. Jacobson. The unified modeling language user guide. *Addison Wesley, First Edition October 20*, 1998.
- [41] J.M. Sobel and D.P. Friedman. An introduction to reflection-oriented programming. *Proceedings of Reflection'96*, San Francisco, CA, USA, Apr. 1996.
- [42] Sun Microsystems Inc. Trail: The reflection api. *The Java Tutorials*, <http://java.sun.com/docs/books/tutorial/reflect/index.html> (acessado 22 Mar 2008).
- [43] R.N. Shiffman and R.A. Greenes. Improving clinical guidelines with logic and decision-table techniques. *Medical Decision Making*, 14(3):245–1101, 1994.
- [44] R.N. Shiffman. Representation of clinical practice guidelines in conventional and augmented decision tables. *Journal of the American Medical Informatics Association*, 4:382–393, 1997.
- [45] G. Duftschmid and S. Miksch. Knowledge-based verification of clinical guidelines by detection of anomalies. *Artificial Intelligence in Medicine*, 22(1):23–41, 2001.
- [46] G. Duftschmid, S. Miksch, and W. Gall. Verification of temporal scheduling constraints in clinical practice guidelines. *Artificial Intelligence in Medicine*, 25(2):93–121, 2002.

- [47] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P.J.F. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36(3):193–209, 2006.
- [48] M. Balser, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums. Formal system development with kiv. In *T.Maibaum, editor, Fundamental Approaches to Software Engineering*, 1783 of LNCS:219–233, 2000.
- [49] S. Bäumlér, M. Balser, A. Dunets, W. Reif, and J. Schmitt. Verification of medical guidelines by model checking: a case study. *Proceedings of 13th International SPIN Workshop on Model Checking of Software*, 3925 of LNCS:219–233, 2006.
- [50] K.L. McMillan. *Symbolic Model Checking: An Approach to the State Explosion Problem*. Kluwer Academic, 1993.
- [51] J. Delgrande and T. Schaub. A consistency-based approach for belief change. *Artificial Intelligence Journal*, 151(1-2):1–41, 2003.
- [52] S. Russell and P. Norvig. *Artificial intelligence : a modern approach*. Prentice-Hall Inc., 1995.
- [53] Sociedades Brasileiras de Nefrologia Cardiologia e Hipertensão. Iv diretrizes brasileiras de hipertensão arterial. *Technical report*, 2002.
- [54] ABC da Saúde e Prevenção Ltda. Gestaçãõ de alto risco. *ABC da Saúde*, <http://www.abcdasaude.com.br/artigo.php?208> (acessado 18 Nov 2007), 2001.

Apêndice A

Guia para Assistência Pré-Natal

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

Assistência Pré-Natal

Federação Brasileira das
Sociedades de Ginecologia e Obstetria

Elaboração Final: 28 de maio de 2001

Autoria: Alencar Jr. CA

O Projeto Diretrizes, iniciativa conjunta da Associação Médica Brasileira e Conselho Federal de Medicina, tem por objetivo conciliar informações da área médica a fim de padronizar condutas que auxiliem o raciocínio e a tomada de decisão do médico. As informações contidas neste projeto devem ser submetidas à avaliação e à crítica do médico, responsável pela conduta a ser seguida, frente à realidade e ao estado clínico de cada paciente.

1

DESCRIÇÃO DO MÉTODO DE COLETA DE EVIDÊNCIAS:

Revisão de literatura e consensos.

GRAU DE RECOMENDAÇÃO E FORÇA DE EVIDÊNCIA:

A: Grandes ensaios clínicos aleatorizados e meta-análises.

B: Estudos clínicos e observacionais bem desenhados.

C: Relatos e séries de casos clínicos.

D: Publicações baseadas em consensos e opiniões de especialistas.

OBJETIVOS:

Oferecer informações atualizadas para realização da assistência pré-natal em gestantes normais, buscando que cada gestação culmine no parto de um recém-nascido saudável, sem prejuízos à saúde da mãe, através de:

a) Prevenção, identificação e/ou correção dos distúrbios maternos ou fetais que afetam a gravidez, incluindo os fatores sociais, econômicos, psicológicos e emocionais, bem como os médicos e/ou obstétricos;

b) Instrução do paciente sobre a gravidez, trabalho de parto, parto, atendimento ao recém-nascido, bem como aos meios de que ela pode se valer para melhorar sua saúde.

PROCEDIMENTOS:

1. Diagnóstico da gravidez;
2. Roteiro de consultas pré-natal;
3. Rotina laboratorial pré-natal;
4. Rotina ultra-sonográfica e avaliação da vitalidade fetal no pré-natal.

EVOLUÇÃO E DESFECHOS:

Redução da morbidade e mortalidade materna e perinatal.

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

RECOMENDAÇÕES GERAIS

QUANTO AO DIAGNÓSTICO DA GRAVIDEZ

O diagnóstico da gravidez pode ser efetuado em 90% das pacientes através dos sinais clínicos, sintomas e exame físico, em gestações mais avançadas. As queixas principais incluem o atraso menstrual, fadiga, mastalgia, aumento da frequência urinária e enjômos matinais¹(D).

Os testes urinários têm baixa taxa de resultados falso-positivos, mas elevada taxa de falso-negativos, podendo atrasar o início do pré-natal. A pesquisa através de radioimunoensaio da fração beta da gonadotrofina coriônica humana é o método mais sensível e confiável, embora também o mais caro²(D).

RECOMENDAÇÕES GERAIS

QUANTO AO ROTEIRO DE CONSULTAS PRÉ-NATAIS

Talvez o principal indicador do prognóstico ao nascimento seja o acesso à assistência pré-natal³(B). Os cuidados assistenciais no primeiro trimestre são utilizados como um indicador maior da qualidade dos cuidados maternos⁴(D). Se o início precoce do pré-natal é essencial para a adequada assistência, o número ideal de consultas permanece controverso. Segundo a Organização Mundial de Saúde, o número adequado seria igual ou superior a seis. Pode ser que mesmo com um número mais reduzido de consultas, por fim com maior ênfase para o conteúdo de cada uma das consultas, em pacientes de baixo risco, não haja aumento de resultados perinatais adversos⁵(A).

Atenção especial deve ser dispensada às grávidas com maiores riscos⁶(A).

As consultas devem ser mensais até a 28ª semana, quinzenais entre 28 e 36 semanas e semanais no termo⁷(D).

Os componentes da história que precisam ser interrogados na primeira consulta pré-natal são²(D):

Data precisa da última menstruação; regularidade dos ciclos; uso de anticoncepcionais; paridade; intercorrências clínicas, obstétricas e cirúrgicas; detalhes de gestações prévias; hospita-

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

lizações anteriores; uso de medicações; história prévia de doença sexualmente transmissível; exposição ambiental ou ocupacional de risco; reações alérgicas; história pessoal ou familiar de doenças hereditárias/malformações; gemelaridade anterior; fatores socioeconômicos; atividade sexual; uso de tabaco, álcool ou outras drogas ilícitas ou ilícitas; história infecciosa prévia; vacinações prévias; história de violências.

A pesquisa de sintomas relacionados gravidez também deve ser questionada²(D):

Núcleas, vômitos, dor abdominal, constipação, cefaleia, síncope, sangramento ou corrimento vaginal, disúria, polaciúria e edemas.

O aconselhamento genético deve ser efetuado para as que têm²(D):

Idade materna maior que 35 anos no parto; história pessoal ou familiar de criança com defeito aberto do tubo neural; risco tríplice para condições genéticas detectáveis; história pessoal ou familiar de doença cardíaca congênita; anomalias anatômicas fetais detectáveis; ultra-sonografia; resultados anormais do teste triplicado; história pessoal ou familiar de anormalidades cromossômicas; diabetes prévio; gestação abortamento habitual; exposição a drogas teratogênicas na gestação; feto com risco para desordem ligada ao cromossomo X; história de uso de quimicos ou radiação; história pessoal ou familiar de hemofilia, distrofia muscular, fibrose cística, coreia de Huntington, acondroplasia, síndrome de Marfan, deficiência de G6PD, fenilcetonúria, retardo mental ou desenvolvimento anormal.

No exame físico os mais importantes componentes que precisam ser incluídos na

primeira visita pré-natal são: peso, altura, pressão arterial, avaliação de mucosas, da tireoide, das mamas, pulmões, coração, abdome e extremidades. No exame ginecológico/obstétrico, avaliar a genitália externa, vagina, colo uterino e, no toque bidual, quando necessário, o terço e anexos. Após a 12ª semana medir a altura do fundo uterino no abdome. A ausculta fetal ser possível após a 10/12ª semana com o sonar-doppler²(D).

Nas visitas subsequentes, torna-se obrigatório medir a altura uterina, pesar a paciente, mensurar a pressão arterial, verificar a presença de anemia de mucosas, existência de edemas e auscultar os batimentos cardíacos fetais. É obrigatório trabalhar o mamilo para lactação. A definição da apresentação fetal deve ser determinada por volta da 36ª semana²(D).

Em relação à suplementação de vitaminas e nutrientes na gestação:

- Ferro e folato: A suplementação rotineira de ferro e folato parecem prevenir a instalação de baixos níveis de hemoglobina no parto e puerpério. Existem poucas informações em relação a outros parâmetros de avaliação da mãe e seu recém-nascido⁸(A);

- Folato peri-concepcional: Tem forte efeito protetor contra defeitos abertos do tubo neural. Deve ser usado rotineiramente pelo menos dois meses antes e nos dois primeiros meses da gestação. Esta informação deve ser difundida através de programas educacionais de saúde. Mulheres que tiveram fetos ou neonatos com defeitos abertos do tubo neural, têm que usar folato continuamente se ainda desejam engravidar⁹(A);

- Piridoxina (vitamina B6): Não existem

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

evidências para avaliar sua suplementação na gravidez, embora os resultados de um trabalho sugeriram que seu uso pode ter efeito protetor em relação ao esmalte dentário^{10(A)};

- **Cálcio:** Seu uso parece ser benéfico em mulheres que apresentam alto risco de desenvolver hipertensão na gestação e naquelas que têm baixa ingestão de cálcio. Novas investigações são necessárias para definição da melhor dose a ser utilizada na gestação^{11(A)};

- **Zinco:** Não existem evidências suficientes para avaliar os efeitos do seu uso na gravidez. Os possíveis efeitos benéficos sobre a prevenção do trabalho de parto prematuro necessitam de novas avaliações^{12(A)};

- **Proteínas:** A suplementação balanceada parece melhorar o crescimento fetal e reduzir os riscos de morte fetal e neonatal. As evidências são insuficientes para afirmar se existem outras vantagens potenciais^{13(A)}.

RECOMENDAÇÕES

GERAIS QUANTO À ROTINA LABORATORIAL PRÉ-NATAL

A rotina laboratorial na primeira consulta, de preferência no primeiro trimestre, baseada em evidências científicas, ^{14(D)}:

- Colpocitologia oncológica;

- Cultura cervical para *N. Gonorrhea* e *C. trachomatis* para grávidas de alto risco, com diminuição dos custos decorrentes da menor incidência de trabalho de parto prematuro, parto prematuro, endometrite pós-parto e conjuntivite neonatal;

- Hematócrito e hemoglobina;

- Tipagem sanguínea e fator Rh;

- Coombs indireto em pacientes Rh negativas com fator de risco para doença hemolítica perinatal. Repetir o teste entre 24 e 28 semanas se anteriormente negativo;

- Glicemia de jejum;

- Sorologia para sífilis - VDRL. Se positivo, necessita de confirmação com o FTA-ABS. Repetir com 36 semanas para pacientes de alto risco;

- Sorologia para rubéola. Se negativa, cuidados para evitar exposição desnecessária ao vírus. Vacinar imediatamente no pós-parto;

- HbsAg. Se positivo, novos testes estão indicados. Se negativo, e vacinação prévia não tiver sido efetuada, considerar fazer vacina. Fazer novo rastreio com 36 semanas. Se a mãe for portadora do antígeno, a criança tem 70% a 90% de chance de adquirir hepatite B e 85% a 90% de ser portadora crônica. O tratamento do recém-nascido com imunoglobulina e vacina para hepatite B reduz o risco em 85% a 90%;

- Elisa anti-HIV. As pacientes negativas, de risco, deverão repetir o teste durante os próximos 6 meses. O teste positivo necessita de confirmação com outro teste Elisa e com um teste específico (Western-blot);

- Sorologia para toxoplasmose. Não existe consenso em relação ao rastreio da toxoplasmose. Quando a IgM positiva, pode indicar infecção nos últimos dois anos. A titulação não auxilia a determinar quando a infecção da paciente ocorreu. Como o rastreio é caro e as vantagens do tratamento ainda não estão claramente definidas, novas pesquisas são necessárias(D);

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

- Sorologia para citomegalovírus. Infelizmente, não existe imunidade após a primoinfecção. Em virtude disso, e da impossibilidade de adoção de condutas específicas, caso a sorologia demonstre infecção, sua utilização na rotina pré-natal vem sendo extremamente questionável;

- Sumário de urina e urocultura. Redução do risco de infecções urinárias baixas sintomáticas e de pielonefrite;

- Análise cromossômica fetal por biópsia de vilos coriônicas. Risco de perda da gestação deve ser comparado aos riscos do procedimento. Em pacientes de baixo risco, desejável a avaliação ultrassonográfica dos marcadores biofísicos de cromossomopatias, especialmente translucência nucal, antes da decisão de realização do procedimento (D).

Os exames sugeridos no segundo trimestre são¹⁴ (D):

- Teste sérico triplice materno. A mensuração da alfa-fetoproteína, estriol não conjugado e gonadotrofina coriônica humana pode ser utilizada para rastreamento dos defeitos abertos do tubo neural e trissomias, especialmente a trissomia do cromossomo 21. Pode detectar 56% a 91% dos defeitos abertos do tubo neural e evitar 75% das amniocenteses, se oferecido entre 15 e 18 semanas para mulheres com menos de 35 anos. Para mulheres acima dos 35 anos, a recomendação é a realização da amniocentese, independente do resultado do teste;

- Análise cromossômica fetal por amniocentese. Risco de perda da gestação (0,5 a 1%) deve ser comparado aos riscos do procedimento;

- Teste oral de tolerância à glicose com 50 gramas de glicose entre a 24ª e 28ª semanas de gestação;

- Glicemia de jejum, segundo recomendação do Estudo Brasileiro do Diabetes Gestacional¹⁵ (D);

- Curva glicêmica de 2 horas com 75 gramas de glicose, segundo recomendação do Estudo Brasileiro do Diabetes Gestacional (dever ser realizado se TOTG maior que 140 mg/dl ou glicemia de jejum maior que 85 mg/dl)¹⁵ (D).

Os exames sugeridos no terceiro trimestre são¹⁴ (D):

- Hematcrito e hemoglobina;

- Cultura cervical para estreptococos do grupo B em pacientes de risco (rotura prematura das membranas com mais de 18 horas, trabalho de parto prematuro, criança anterior afetada pelo estreptococo grupo B, febre materna maior que 38°C e bacteriúria com a presença do estreptococo grupo B).

RECOMENDAÇÕES GERAIS QUANTO À
ROTINA ULTRA-SONOGRÁFICA E
AVALIAÇÃO DA VITALIDADE
FETAL NO PRÉ-NATAL

As melhores evidências científicas definem sua realização rotineira no 2º trimestre para determinação da idade gestacional e pesquisa de anomalias congênitas. A avaliação no 1º trimestre permite melhor acompanhamento da idade gestacional, detecção precoce de gestações gemelares e de fetos portadores de anomalias¹⁶ (A).

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

A realização de exames rotineiros após a 24ª semana, em pacientes de baixo risco, não confere benefícios maternos ou ao feto^{17(A)}. Em relação ao diagnóstico do bem-estar fetal, um estudo realizado com mais de 15.000 pacientes de baixo risco não ofereceu subsídios para seu uso rotineiro^{18(A)}.

Na avaliação da vitalidade fetal podemos dispor de:

- Contagem diária dos movimentos fetais: muito simples e inócua de avaliação fetal. A mãe deve quantificar os movimentos fetais, definindo o tempo necessário para obter-se um determinado número de movimentos ou, por outro lado, quantas vezes o feto se movimenta em período de uma hora;

- Cardiocotografia anteparto: Apesar de ser muito seguro e inócua em relação à mãe, não existem evidências científicas que autorizem seu uso rotineiro em gestações de baixo risco. Os dados incluídos nas últimas revisões, entretanto, são do início da utilização da cardiocotografia e podem não se correlacionar com a prática obstétrica atual^{19(A)};

- Dopplervelocimetria: Sua utilização em gestações de risco totalmente justificada, existindo melhora no cuidado obstétrico, parecendo ser promissora em reduzir mortes perinatais^{20(A)}.

Em pacientes de baixo risco não existem evidências que justifiquem sua utilização rotineira, não adicionando benefícios maternos ou ao feto^{21(A)}.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Chard T. Pregnancy tests: a review. *Hum Reprod* 1992; 7: 701-10.
2. Andolsek KM, Kelton GM. Risk assessment. *Prim Care* 2000; 27: 71-103.
3. Amini SB, Catalano PM, Mann LI. Effect of prenatal care on obstetrical outcome. *J Matern Fetal Med* 1996; 5: 142-50.
4. National Committee for Quality Assurance: Medicaid HEDIS: an adaptation of NCQA's Health Plan. Washington: Employer Data and Information, Set. 1998.
5. Villar J, Khan-Neelofur D. Patterns of routine antenatal care for low-risk pregnancy (Cochrane Review). In: *The Cochrane Library*, Issue 3, 2000. Oxford: Update Software.
6. McDuffie RS, Beck A, Bischoff K, et al. Effect of frequency of prenatal care visits on perinatal outcome among low risk women. *JAMA* 1998; 275: 847-51.
7. Cunningham FG, MacDonald PC, Gant NF, et al. Prenatal care. In: Cunningham FG, Williams J, Whitridge J, editors. *Williams obstetrics*. 20th ed., Connecticut: Appleton & Lange, 1997; 227-50.
8. Mahomed K. Iron and folate supplementation in pregnancy (Cochrane Review). In: *The Cochrane Library*, Issue 3, 2000. Oxford: Update Software.
9. Lumley J, Watson L, Watson M, et al. Periconceptional supplementation with folate and/or multivitamins for preventing neural tube defects (Cochrane Review). In: *The*

Projeto Diretrizes

Associação Médica Brasileira e Conselho Federal de Medicina

- Cochrane Library, Issue 3, 2000. Oxford: Update Software.
10. Mahomed K, Gulmezoglu AM. Pyridoxine (vitamin B6) supplementation in pregnancy (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
11. Atallah AN, Hofmeyr GJ, Duley L. Calcium supplementation during pregnancy for preventing hypertensive disorders and related problems (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
12. Mahomed K. Zinc supplementation in pregnancy (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
13. Kramer MS. Balanced protein/energy supplementation in pregnancy (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
14. US Preventive Services Task Force. Guide to clinical preventive services. 2nd ed. Baltimore: Williams and Wilkins; 1996.
15. Schimdt MI, Reichelt AJ. Consenso sobre diabetes gestacional e diabetes pré-gestacional. Arq Bras Endocrinol Metab 1999; 43:14-20.
16. Neilson JP. Ultrasound for fetal assessment in early pregnancy (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
17. Bricker L, Neilson JP. Routine doppler ultrasound in late pregnancy (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
18. Ewigman BG, Crane JP, Figoletto FD, et al. Effect of prenatal ultrasound screening on perinatal outcome: RADIUS Study Group. N Engl J Med 1993; 329:821-7.
19. Pattison N, McCowan L. Cardiotocography for antepartum fetal assessment (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
20. Neilson JP, Alfirevic Z. Doppler ultrasound for fetal assessment in high risk pregnancies (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.
21. Bricker L, Neilson JP. Routine doppler ultrasound in pregnancy (after 24 weeks gestation) (Cochrane Review). In: The Cochrane Library, Issue 3, 2000. Oxford: Update Software.

Apêndice B

Gramática da ST-Linguagem em XML

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- definition of simple elements -->
  <xs:element name="author" type="xs:string"/>
  <xs:element name="version" type="xs:string"/>
  <xs:element name="name" type="xs:string"/>

  <xs:element name="description" type="xs:string"/>
  <xs:element name="type" type="xs:string"/>
  <xs:element name="id" type="xs:string"/>
  <xs:element name="text" type="xs:string"/>
  <xs:element name="value" type="xs:string"/>
  <xs:element name="validity" type="xs:positiveInteger"/>
  <xs:element name="nextSession" type="xs:positiveInteger"/>
  <xs:element name="next" type="xs:string"/>
  <xs:element name="if" type="xs:string"/>

  <xs:element name="posX" type="xs:integer"/>
  <xs:element name="posY" type="xs:integer"/>
  <xs:element name="width" type="xs:integer"/>
  <xs:element name="height" type="xs:integer"/>
```

```
<xs:element name="dose" type="xs:string"/>
<xs:element name="unit" type="xs:string"/>
<xs:element name="usage" type="xs:string"/>
<xs:element name="period" type="xs:positiveInteger"/>

<!-- definition of attributes -->
<xs:attribute name="id" type="xs:string"/>
<xs:attribute name="inputs" type="xs:positiveInteger"/>

<!-- definition of complex types -->
<!-- simple reference to declared elements-->
<xs:complexType name="simpleRef">
  <xs:sequence>
    <xs:element ref="id"/>
  </xs:sequence>
</xs:complexType>

<!-- reference with validity -->
<xs:complexType name="refWithValidity">
  <xs:sequence>
    <xs:element ref="id"/>
    <xs:element ref="validity"/>
  </xs:sequence>
</xs:complexType>

<!-- label of physical and medical exams. Used to declare them. -->
<xs:complexType name="examLabel">
  <xs:sequence>
    <xs:element ref="id"/>
    <xs:element ref="name"/>
  </xs:sequence>
</xs:complexType>

<!-- **** definition of complex elements -->

<!-- answers -->
```

```

<xs:element name="answer">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="value"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="answers">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="answer" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!-- questions -->
<xs:element name="questionDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="text"/>
      <xs:element ref="answers"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="questionsDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="questionDec" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!-- reminders -->

```

```

<xs:element name="reminderDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="text"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="remindersDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="reminderDec" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!-- lab exams -->
<xs:element name="labExamDec" type="examLabel"/>

```

```

<xs:element name="labExamsDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="labExamDec" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<!-- phy exams -->
<xs:element name="phyExamDec" type="examLabel"/>

```

```

<xs:element name="phyExamsDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="phyExamDec" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>

<!-- medications -->
<xs:element name="medicationDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="name"/>
      <xs:element ref="type"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="medicationsDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="medicationDec" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- declarations -->
<xs:element name="declarations">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="questionsDec" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="remindersDec" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="medicationsDec" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="phyExamsDec" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="labExamsDec" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- *** State's complex elements -->

<!-- reference elements to the elements in "declarations" -->

```

```

<xs:element name="reminder" type="simpleRef"/>
<xs:element name="question" type="refWithValidity"/>
<xs:element name="phyExam" type="refWithValidity"/>
<xs:element name="labExam" type="refWithValidity"/>

<!-- Literal related -->
<xs:element name="var">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="type"/>
      <xs:element ref="id"/>
      <xs:element ref="value" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="op">
  <xs:complexType>
    <xs:attribute ref="id" use="required"/>
    <xs:attribute ref="inputs" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="literal">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="description"/>
      <xs:element ref="op"/>
      <xs:element ref="var" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="literals">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="literal" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

</xs:sequence>
  </xs:complexType>
</xs:element>

<!-- transitions -->
<xs:element name="transition">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="next"/>
      <xs:element ref="if"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="transitions">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="transition" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- uses -->
<xs:element name="helper">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="next"/>
      <xs:element ref="if"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="uses">
  <xs:complexType>
    <xs:sequence>

```

```

    <xs:element ref="helper" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- medications, exams and questions-->
<xs:element name="medication">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="dose"/>
      <xs:element ref="unit"/>
      <xs:element ref="usage"/>
      <xs:element ref="period"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="medications">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="medication" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="labExams">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="labExam" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="phyExams">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element ref="phyExam" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="reminders">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="reminder" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="questions">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="question" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<!-- State -->
<xs:element name="state">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="posX"/>
            <xs:element ref="posY"/>
            <xs:element ref="width"/>
            <xs:element ref="height"/>
            <xs:element ref="description"/>
            <xs:element ref="phyExams"/>
            <xs:element ref="labExams"/>
            <xs:element ref="questions"/>
            <xs:element ref="medications"/>
            <xs:element ref="reminders"/>
            <xs:element ref="nextSession"/>
            <xs:element ref="literals"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

    <xs:element ref="transitions"/>
    <xs:element ref="uses"/>
  </xs:sequence>
  <xs:attribute ref="id" use="required"/>
</xs:complexType>
</xs:element>

<xs:element name="states">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="state" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Helper State -->
<xs:element name="helperState">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="posX"/>
      <xs:element ref="posY"/>
      <xs:element ref="width"/>
      <xs:element ref="height"/>
      <xs:element ref="description"/>
      <xs:element ref="phyExams"/>
      <xs:element ref="labExams"/>
      <xs:element ref="questions"/>
      <xs:element ref="medications"/>
      <xs:element ref="reminders"/>
      <xs:element ref="literals"/>
      <xs:element ref="uses"/>
    </xs:sequence>
    <xs:attribute ref="id" use="required"/>
  </xs:complexType>
</xs:element>

```

```
<xs:element name="helperStates">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="helperState" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<!-- Guide -->
<xs:element name="guide">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="author"/>
      <xs:element ref="name"/>
      <xs:element ref="version"/>
      <xs:element ref="declarations"/>
      <xs:element ref="states"/>
      <xs:element ref="helperStates"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Apêndice C

Modelo Conceitual da Representação Interna de Paciente

A Tabela C.1 mostra a estrutura geral de como um paciente é representado no ST-Guide. As demais tabelas definem conceitualmente os elementos ligados diretamente ou indiretamente à esta tabela.

Tabela C.1: Modelo conceitual da representação interna de paciente do ST-Sistema.

Paciente	
Atributo	Descrição
Gênero	gênero do paciente (masculino ou feminino)
Data de Nascimento	dia, mês, ano e horário de nascimento
Estado atual	estado atual do paciente dentro do tratamento
Medicações	medicações em uso
Exames físicos	lista com último exame físico de cada tipo
Exames laboratoriais	lista com último exame laboratorial de cada tipo
Perguntas	lista com última pergunta de cada tipo respondida
Todas medicações	histórico de medicações
Todos exames físicos	histórico de exames físicos
Todos exames laboratoriais	histórico de exames laboratoriais
Todas perguntas	histórico de perguntas

Tabela C.2: Estado atual.

Estado atual	
Atributo	Descrição
Identificador	identificador único de estado
Descrição	breve descrição
Medicações	lista de medicações recomendadas
Exames físicos	lista de exames físicos recomendados
Exames laboratoriais	lista de exames laboratoriais recomendados
Perguntas	lista de perguntas a serem realizadas
Lembretes	lista de recomendações
Próxima consulta	número de dias até próxima consulta
Literais	possíveis literais a serem usados nas funções de transição ou uso
Transições	possíveis transições
Usa	possíveis relações de uso

Tabela C.3: Medicação.

Medicação	
Atributo	Descrição
Identificador	identificador único de medicação
Nome	nome do medicamento
Tipo	grupo de medicamentos ao qual pertence
Período	número de dias em que o medicamento deve ser usado
Data prescrição	data em que foi prescrito
Data remoção	data em que foi recomendada a interrupção de uso
Dose	dose recomendada
Unidade	unidade da dose
Uso	detalhes de uso (número de vezes por dia, tomar com leite ou água, etc..)

Tabela C.4: Exame Físico ou Laboratorial.

Exame Físico ou Laboratorial	
Atributo	Descrição
Identificador	identificador único de exame
Nome	nome do exame
Data	data e horário de realização
Validade	tempo em número de dias que o exame continua válido
Valor	resultado do exame (numérico ou não)
Unidade	unidade do resultado (quando houver)
Tipo	tipo de exame

Tabela C.5: Pergunta e Resposta.

Pergunta	
Atributo	Descrição
Identificador	identificador único de pergunta
Texto	texto da pergunta
Possíveis respostas	lista com possíveis respostas
Resposta	resposta selecionada para a pergunta
Validade	número de dias em que uma resposta continua válida
Resposta	
Atributo	Descrição
Identificador	identificador de resposta
Valor	valor da resposta
Unidade	unidade da resposta (se houver)
Data	data da resposta

Tabela C.6: Lembrete.

Lembrete	
Atributo	Descrição
Identificador	identificador único
Texto	texto da recomendação

Tabela C.7: Literal.

Literal	
Atributo	Descrição
Identificador	identificador único
Operador	elemento que atua sobre operandos retornando <i>true</i> ou <i>false</i> .
Operandos	lista de elementos (entradas) a serem usados pelo operador

Tabela C.8: Operador.

Operador	
Atributo	Descrição
Identificador	identificador de operador
Número de Operandos	número de operandos ou entradas requeridas

Tabela C.9: Operando.

Operando	
Atributo	Descrição
Identificador	identificador de operando
Tipo	diferencia grupos diferentes de operandos
Valor	valor do operando

Tabela C.10: Transição e Relação de Uso.

Transição e Relação de Uso	
Atributo	Descrição
Identificador	identificador de transição ou uso
Descrição	significado da transição/uso no tratamento
Alvo	identificador do estado ou <i>Helper-State</i> destino
Condição	expressão Booleana que define ocorrência da transição/uso

Apêndice D

Gravidez de Alto Risco

Aqui listamos vários fatores, divididos em grupos, que servem como indicadores de gestação de alto risco conforme [54].

1- Fatores individuais e sócio-econômicos:

- idade materna menor do que 17 anos ou maior do que 35 anos;
- altura materna menor do que 1,45m;
- exposição a agentes físico-químicos nocivos e estresse;
- má aceitação da gestação;
- situação conjugal insegura;
- baixa escolaridade;
- baixa renda;
- peso materno inadequado;
- dependência de drogas lícitas ou ilícitas.

2- História ginecológica e obstétrica anterior:

- gestação ectópica;
- abortamento habitual;
- infertilidade;

- anormalidades uterinas;
- feto morto ou morte neonatal não explicada;
- trabalho de parto prematuro;
- recém-nascido de baixo peso;
- neoplasia ginecológica;
- cirurgia uterina anterior;
- hemorragia ou pressão alta em gestação anterior.

3- Doenças maternas prévias ou concomitantes:

- cardiopatia (doença do coração);
- pneumopatia crônica (doença dos pulmões);
- doenças da tireóide;
- retardo mental;
- doenças sexualmente transmissíveis;
- tumores;
- doenças psiquiátricas;
- epilepsia;
- doenças hematológicas (do sangue);
- infecções.

4- Doenças da gestação atual:

- crescimento uterino maior ou menor do que o esperado;
- gestação gemelar ou múltipla;
- não realização de pré-natal ou pré-natal insuficiente;
- hipertensão associada a gestação;
- diabete associada a gestação;

- ruptura prematura de membranas (ruptura da bolsa antes de 37 semanas);
- isoimunização (doença do RH);
- ganho de peso excessivo.

Apêndice E

Exercícios do experimento de usabilidade

Introdução Prática ao ST-Modeler

PARTE 1

Conhecendo o ST-Modeler

Leia atentamente os textos 1 e 2. Apesar de não possuírem nenhuma fundamentação médica, considere que eles são fragmentos de guias clínicos reais e os modele usando o ST-Modeler.

1. Se o paciente já tiver tido caxumba e estiver com febre acima de 38 graus, então se recomenda Novalgina 2,5 ml de seis em seis horas por uma semana. Na semana seguinte, se a temperatura do paciente continuar acima de 38 graus celsius, então a dose de Novalgina deverá ser elevada para 5 ml e a densidade de leucócitos no sangue avaliada. Se o nível de leucócitos for menor que 10 unidades por mm³, então o paciente deverá ser encaminhado para um especialista. Caso contrário, pode ser dado alta para ele.
2. Neste instante do tratamento o paciente está estável. As consultas devem ser mensais e se deve sempre verificar a pressão sistólica (sbp) e diastólica (dbp). Se $sbp < 100$ mmHg, então Adalat Retard 20 mg/dia mostrou-se bastante eficaz. Se $sbp \geq 100$ mmHg, então Atenolol 50 mg/dia é a escolha mais recomendada. Se $dbp < 90$ mmHg então Xantinon B12 10mg/dia é recomendado. Caso contrário, Homeoprazol 12 mg/dia.

Após modelar cada um dos fragmentos, analise suas propriedades lógicas via ST-Modeler. Entenda os problemas que surgiram e tente corrigí-los. Depois, execute os fragmentos no ambiente de testes usando pacientes fictícios.