

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Programa de Pós-Graduação em Ciência da Computação

Uma Abordagem Baseada em Modelos para Suporte  
à Validação de Sistemas Médicos Físico-Cibernéticos

Lenardo Chaves e Silva

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Sistemas de Computação

Hyggo Oliveira de Almeida, D. Sc.

Angelo Perkusich, D. Sc.

(Orientadores)

Campina Grande, Paraíba, Brasil

©Lenardo Chaves e Silva, 12 de Novembro de 2015

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S586a Silva, Lenardo Chaves e.  
Uma abordagem baseada em modelos para suporte à validação de sistemas médicos físico-cibernéticos / Lenardo Chaves e Silva. – Campina Grande, 2015.  
145 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

"Orientação: Prof. Dr. Hyggo Oliveira de Almeida , Prof. Dr. Angelo Perkusich".

Referências.

1. Sistemas Médicos Físico-Cibernético. 2. Validação. 3. Simulação. 4. Análise Formal. 5. Modelos. I. Almeida, Hyggo Oliveira de. II. Perkusich, Angelo. III. Título.

CDU 004.2(043)

## Resumo

Sistemas Médicos Físico-Cibernéticos (SMFC) são sistemas críticos cientes de contexto que têm a segurança do paciente como principal requisito, demandando processos rigorosos de validação para garantir a conformidade com os requisitos do usuário e a correte orientada à especificação. Neste trabalho é proposta uma arquitetura baseada em modelos para validação de SMFC, focando em promover a reúso e a produtividade. Tal abordagem permite que desenvolvedores de sistemas construam modelos formais de SMFC baseados em uma biblioteca de modelos de pacientes e dispositivos médicos, bem como simular o SMFC para identificar comportamentos indesejados em tempo de projeto. A abordagem proposta foi aplicada a três diferentes cenários clínicos para avaliar seu potencial de reúso para diferentes contextos. A abordagem foi também validada por meio de uma avaliação empírica com desenvolvedores para avaliar o reúso e a produtividade. Finalmente, os modelos foram formalmente verificados considerando os requisitos funcionais e de segurança, além da cobertura dos modelos.

## **Abstract**

Medical Cyber-Physical Systems (MCPS) are context-aware, life-critical systems with patient safety as the main concern, demanding rigorous processes for validation to guarantee user requirement compliance and specification-oriented correctness. In this article, we propose a model-based approach for early validation of MCPS, focusing on promoting reusability and productivity. It enables system developers to build MCPS formal models based on a library of patient and medical device models, and simulate the MCPS to identify undesirable behaviors at design time. Our approach has been applied to three different clinical scenarios to evaluate its reusability potential for different contexts. We have also validated our approach through an empirical evaluation with developers to assess productivity and reusability. Finally, our models have been formally verified considering functional and safety requirements and model coverage.

Dedico esta tese à minha esposa Monalisa Lima e família,  
aos meus pais José Neuton e Maria Marlene e  
à minha irmã Patrícia Chaves.

## Agradecimentos

Esta Tese é parte de um sonho que sempre busquei alcançá-lo, superando as adversidades e os desafios encontrados ao longo dessa trajetória que se iniciou em 2005 quando ingressei na universidade.

Meus agradecimentos, em especial, é para minha esposa Monalisa Lima, que, mesmo distante fisicamente, sempre esteve ao meu lado dando todo o apoio necessário, tanto mental quanto físico quando nas horas oportunas. Obrigado minha princesa... espero que a partir de agora possamos desfrutar da nossa vida de casados e formar uma linda família. Te AMO Eternamente! Aos seus pais Sr. Jorge e D. Rita não tenho como agradecer tamanho acolhimento, que geralmente só se dá a um filho. Meu humilde obrigado.

Aos meus pais, Maria Marlene C. Freire e José Neuton P. e Silva, e minha irmã Patrícia Chaves, por sempre estarem ao meu lado nas minhas decisões, independente dos rumos que elas me levaram. Sou muito feliz com vocês e mesmo à distância podem contar comigo, estaremos sempre juntos e no mesmo caminho da felicidade. Amo todos vocês... obrigado!

Aos meus orientadores Prof. Dr. Hyggo Oliveira e Prof. Dr. Angelo Perkusich pelo intenso período de aprendizado e alta qualidade em suas orientações e aconselhamentos, que me fez a cada dia refletir que somos capazes e podemos ser sempre melhores naquilo que fazemos.

Aos membros das minhas bancas de Qualificação e Defesa de Tese de Doutorado, pelos enriquecedores comentários sobre o meu trabalho, Profs. Kyller C. Gorgônio (UFCEG), Saulo O. Dornellas Luiz (UFCEG), Augusto J. Venâncio Neto (UFRN), Pedro F. Ribeiro Neto (UERN) Leandro Dias (UFAL) e Evandro B. Costa (UFAL).

Ao meu amigo e Prof. Mirko Perkusich, também colega de programa de doutorado, pelo constante suporte na produção de artigos e outros conteúdos científicos em idioma Inglês. Mirko, você me fez aprender bastante entre uma conversa e outra durante esse período intenso de trabalho. Obrigado, meu amigo.

Às amigadas conquistadas no Embedded Lab. (UFCEG) desde o meu acolhimento, em março de 2012, como membro do laboratório... vou enumerar aqui uns poucos

nomes por categoria. Já Doutores: Frederico Bublitz, Kézia Vasconcelos, Daniel Bruno, Elthon Oliveira, Ivo Calado, Maurício Souza, Gustavo Soares, Thiago e Leandro Sales. Hoje Doutorandos: Danilo Freire, Leonardo Medeiros, Renata Saraiva, Álvaro Álvares, Michel Dias, Rodrigo Vilar, Jean Caminha, Felipe Barbosa, Alexandre Costa, Felipe Pontes, Walter Guerra e Mateus Lima. Demais Membros: Renata Fonseca, Rony Marcolino, Leonardo Santos, Aislan Lima e Ádrian Guedes. E, todos os membros do Projeto de Capacitação - Android Tech Center pela participação na avaliação experimental do meu trabalho.

Agradeço, em especial, à CAPES, pelo fomento a esta pesquisa, permitindo minha total dedicação ao programa de doutorado e possibilitando o desenvolvimento de uma pesquisa de qualidade.

Finalmente, mas certamente em primeiro lugar, agradeço ao meu DEUS todo poderoso! Em nome do Senhor, obrigado à todos!

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problemática . . . . .	3
1.2	Objetivos . . . . .	6
1.3	Relevância . . . . .	7
1.4	Estrutura do Documento . . . . .	10
<b>2</b>	<b>Fundamentação</b>	<b>11</b>
2.1	Sistemas Físico-Cibernéticos . . . . .	11
2.2	Ambiente Clínico Integrado . . . . .	13
2.3	Projeto Baseado em Modelos . . . . .	14
2.4	Paradigma de Projeto Orientado a Atores . . . . .	20
2.4.1	Ator . . . . .	20
2.4.2	Modelo de Atores . . . . .	22
2.4.3	Hierarquia de Modelos de Atores . . . . .	23
2.4.4	Modelos de Computação . . . . .	24
2.4.5	Modelos Heterogêneos de Atores . . . . .	35
2.4.6	Ferramentas de Apoio à Modelagem . . . . .	37
2.5	Considerações Finais . . . . .	39
<b>3</b>	<b>Biblioteca de Modelos: Construindo Modelos Reutilizáveis</b>	<b>41</b>
3.1	Construção dos Modelos de Pacientes . . . . .	42
3.1.1	Etapa 1 – Escolher uma Base de Dados Clínica . . . . .	43
3.1.2	Etapa 2 – Obter um Modelo Estatístico . . . . .	44
3.1.3	Etapa 3 – Desenvolver um Modelo AOD . . . . .	48

---

3.1.4	Etapa 4 – Validar um Modelo de Paciente . . . . .	59
3.2	Construção dos Modelos de Dispositivos Médicos . . . . .	60
3.2.1	Etapa 1 – Escolher um Dispositivo Médico Certificado . . . . .	61
3.2.2	Etapa 2 – Desenvolver um Modelo AOD . . . . .	62
3.2.3	Etapa 3 – Validar um Modelo de Dispositivo . . . . .	64
3.3	Biblioteca de Modelos: Prontos para Reúso . . . . .	66
3.4	Considerações Finais . . . . .	67
<b>4</b>	<b>Composição e Simulação: Validação de SMFC</b>	<b>69</b>
4.1	Projetando a Arquitetura Baseada em Modelo . . . . .	71
4.1.1	Selecionando um Modelo de Paciente . . . . .	71
4.1.2	Selecionando os Modelos de Dispositivos Médicos . . . . .	72
4.1.3	Compondo o Modelo de SMFC . . . . .	72
4.2	Definindo os Cenários Clínicos de Interesse . . . . .	74
4.3	Executando Atividades de Validação . . . . .	75
4.4	Considerações Finais . . . . .	75
<b>5</b>	<b>Validação da Abordagem Proposta</b>	<b>77</b>
5.1	Prova de Conceito em Diferentes Cenários Clínicos . . . . .	77
5.1.1	Contexto Clínico I . . . . .	77
5.1.2	Contexto Clínico II . . . . .	79
5.1.3	Contexto Clínico III . . . . .	81
5.2	Avaliação Empírica . . . . .	82
5.2.1	Escopo . . . . .	83
5.2.2	Objetos de Estudo . . . . .	83
5.2.3	Sujeitos . . . . .	84
5.2.4	Variáveis e Tratamento . . . . .	84
5.2.5	Procedimento . . . . .	85
5.2.6	Medidas . . . . .	87
5.2.7	Análise . . . . .	89
5.2.8	Ameaças à Validade . . . . .	92
5.3	Considerações Finais . . . . .	93

---

<b>6</b>	<b>Análise dos Modelos</b>	<b>94</b>
6.1	Considerações Finais . . . . .	100
<b>7</b>	<b>Conclusão</b>	<b>101</b>
	<b>Bibliografia</b>	<b>103</b>
	<b>Apêndice A Revisão Sistemática da Literatura</b>	<b>114</b>
A.1	<i>Background</i> . . . . .	114
A.1.1	Atividades de Verificação e Validação . . . . .	114
A.1.2	Teste Baseado em Modelo . . . . .	115
A.2	Materiais e Métodos . . . . .	116
A.3	Processo de Revisão Sistemática da Literatura . . . . .	117
A.3.1	Planejamento . . . . .	117
A.3.2	Execução . . . . .	118
A.3.3	Sumarização . . . . .	119
A.4	Resultados . . . . .	120
A.4.1	Identificação . . . . .	120
A.4.2	Seleção . . . . .	121
A.4.3	Extração . . . . .	122
A.4.4	Sumarização . . . . .	124
A.5	Discussão . . . . .	127
A.6	Considerações Finais do Capítulo . . . . .	128
	<b>Apêndice B Protocolo de Revisão Sistemática da Literatura (RSL)</b>	<b>130</b>
	<b>Apêndice C Consulta em <i>SQL</i> na Base de Dados Clínica <i>MIMIC II</i></b>	
	<b>para Caracterização da População de Interesse</b>	<b>136</b>
	<b>Apêndice D Questionário sobre o Perfil do Desenvolvedor</b>	<b>138</b>
	<b>Apêndice E Questionário de Avaliação de Reúso dos Modelos</b>	<b>142</b>

# Lista de Símbolos

- AOD** – *Actor-Oriented Design*
- ASTM** – *American Society for Testing and Materials*
- BMM** – *Bergman’s Minimal Model*
- BPMN** – *Business Process Model and Notation*
- CASE** – *Computer-Aided Software Engineering*
- CC** – *Condition Coverage*
- CPS** – *Cyber-Physical System*
- DC** – *Decision Coverage*
- DM** – *Dispositivo Médico*
- FDA** – *U.S. Food and Drug Administration*
- GL** – *Glucose Level*
- GQM** – *Goal-Question-Metric*
- HIL** – *Hardware In the Loop*
- HR** – *Heart Rate*
- ICE** – *Integrated Clinical Environment*
- MBT** – *Model-Based Testing*
- MCPS** – *Medical Cyber-Physical System*
- MCDC** – *Modified Condition/Decision Coverage*
- MDA** – *Model Driven Architecture*
- MDD** – *Model Driven Development*
- MDE** – *Model Driven Engineering*
- MEE** – *Máquina de Estados Estendida*
- MEF** – *Máquina de Estados Finito*
- MoC** – *Model of Computation*

---

**MLG** – Modelos Lineares Generalizados  
**MRM** – *Modelos de Regressão Multivariados*  
**NSF** – *National Science Foundation*  
**ODE** – *Ordinary Differential Equation*  
**OMG** – *Object Management Group* **PCA** – *Patient Controlled Analgesia*  
**PT** – *Peripheral Body Temperature*  
**RR** – *Respiratory Rate*  
**RSL** – Revisão Sistemática da Literatura  
**PD** – Processos Farmacodinâmicos  
**PID** – Proporcional-Integral-Derivativo  
**PK** – Processos Farmacocinéticos  
**RS** – Requisito de Segurança  
**SBD** – Sociedade Brasileira de Diabetes  
**SBP** – *Systolic Blood Pressure*  
**SLDV** – *Simulink Design Verifier*  
**SE** – Sistema Embarcado  
**SFC** – Sistema Físico-Cibernético  
**SMFC** – Sistema Médico Físico-Cibernético  
**StArt** – *State of the Art through Systematic Review*  
**TIC** – Tecnologia de Informação e Comunicação  
**UML** – *Unified Modeling Language*  
**UTI** – Unidade de Terapia Intensiva  
**V&V** – Verificação e Validação

# Lista de Figuras

2.1	Relacionamento entre os elementos do modelo funcional conceitual ICE.	14
2.2	Representação gráfica de um componente <i>ator</i> .	20
2.3	Representação gráfica de um componente <i>ator</i> com múltiplas portas.	22
2.4	Representação gráfica de um <i>modelo de atores</i> .	23
2.5	Exemplo de um modelo hierárquico de atores, usando a notação gráfica.	23
2.6	Relacionamentos entre os MoCs.	25
2.7	Exemplo de uma FSM em notação gráfica.	30
2.8	Exemplo de uma FSM representando o comportamento de um termostato.	31
2.9	Exemplo de uma ESM em notação gráfica.	34
2.10	Exemplo de uma ESM representando o comportamento de um semáforo.	35
2.11	Notação gráfica para um <i>Modelo Modal</i> .	37
3.1	Processo para Caracterizar a População de Interesse.	45
3.2	O Processo de Ajuste do Modelo de Regressão Multivariado para cada SV.	47
3.3	<i>MLG_RR</i> para o Modelo de Paciente.	51
3.4	Exemplo de um Modelo de Paciente.	52
3.5	Modelo de Paciente Baseado em Khan <i>et al.</i> (2013).	57
3.6	Modelo de Paciente Diabético.	58
3.7	Resultado Comparativo Entre os Sistemas de Controle de Glicose.	59
3.8	Modelo da Bomba de Insulina.	62
3.9	Controlador do Modelo da Bomba de Insulina.	63
3.10	Refinamento de Estado do Modelo da Bomba de Insulina.	65
3.11	Biblioteca do Usuário no <i>Ptolemy II</i> .	66

---

3.12	Modelos de Dispositivos na Biblioteca do Usuário do <i>Ptolemy II</i> . . . . .	67
4.1	Abordagem Baseada em Modelo para Validação de SMFC. . . . .	70
4.2	Uma Arquitetura Baseada em Modelos para SMFC. . . . .	70
4.3	Selecionando um Modelo de Paciente no <i>Ptolemy II</i> . . . . .	72
4.4	Selecionando um Modelo de Dispositivo no <i>Ptolemy II</i> . . . . .	73
4.5	Definindo um Modelo de SMFC no <i>Ptolemy II</i> . . . . .	74
5.1	Modelo de SMFC Usando o Modelo de Paciente em UTI. . . . .	78
5.2	Simulação de um Modelo SMFC Usando o Modelo de Paciente em UTI. . . . .	79
5.3	Modelo de SMFC Usando o Modelo da Bomba de Insulina. . . . .	80
5.4	Modelo de SMFC Usando um Modelo de Paciente Diabético. . . . .	81
5.5	Simulação de um Modelo SMFC Usando o Modelo de Paciente Diabético. . . . .	82
5.6	<i>Boxplots</i> dos Tempos Coletados para os Grupos de Controle e de Tratamento. . . . .	90
5.7	Distribuição das Respostas dos Desenvolvedores de acordo com os Fatores de Reúso. . . . .	91
6.1	Modelo de SMFC Usando o Modelo da Bomba de Insulina no <i>Simulink</i> . . . . .	95
6.2	Propriedade de Segurança 1. . . . .	95
6.3	Propriedade de Segurança 2. . . . .	96
6.4	Propriedade de Segurança 3. . . . .	96
6.5	Modelo do Software da Bomba de Insulina no <i>Simulink</i> . . . . .	97
6.6	Suposição Sobre o Estado do Cartucho da Bomba de Insulina. . . . .	98
A.1	Distribuição dos Estudos Primário por Fonte de Busca. . . . .	121
A.2	Análise Quantitativa na Etapa de Seleção. . . . .	122
A.3	Análise Quantitativa na Etapa de Extração. . . . .	122

# Lista de Tabelas

3.1	Medidas de Resumo da População de Interesse. . . . .	46
3.2	Mapeamento dos Modelos de Regressão para o Paradigma AOD. . . . .	49
3.3	Requisitos Técnicos Considerados na Modelagem da Bomba de Insulina. . . . .	61
5.1	Respostas ao Questionário para Identificar o Perfil do Desenvolvedor. . . . .	84
6.1	Resumo do Relatório de Cobertura para o Modelo de Software da Bomba de Insulina. . . . .	99
A.1	Ranqueamento dos estudos selecionados na etapa de Extração. . . . .	123
A.2	Synthesis of data from studies at the end of Step 3 of the RSL process. . . . .	125
A.3	Classificação das principais abordagens de projeto de um SMFC. . . . .	126
A.4	Classificação das ferramentas identificadas no processo RSL, e que são utilizadas no processo V&V de um SMFC. . . . .	127

# Capítulo 1

## Introdução

Os ambientes pessoais e corporativos estão cada vez mais repletos de recursos computacionais que interagem entre si utilizando mecanismos disponibilizados por diferentes meios de comunicação. Isso faz com que diversos elementos ao nosso redor se tornem uma fonte de informação para os seus usuários, proporcionando que entidades virtuais reajam da forma mais apropriada aos estímulos produzidos pelas entidades físicas. Cenários ricos em tecnologias, em que unidades computacionais embarcadas estão em constante interação com elementos do mundo real para monitorar e controlar processos físicos, caracterizam os denominados *Sistemas Físico-Cibernéticos* – SFC (do inglês, *Cyber-Physical Systems* – CPS) [16].

O tamanho e a complexidade de um SFC varia de acordo com o domínio de aplicação, processo físico associado, infraestrutura tecnológica necessária, nível de controle e segurança requerida. Cada um desses fatores agrega outro conjunto de particularidades e desafios que o sistema como um todo deve considerar. Ou seja, é preciso mitigar os riscos oriundos das próprias ações automáticas do sistema, considerando as diversas situações analisadas ao longo de sua execução. Isto é válido tanto para sistemas de escopo reduzido (e.g., pacientes diabéticos em tratamento com insulino-terapia usando uma bomba de infusão subcutânea) como para sistemas de escopo abrangente (e.g., pacientes em situação crítica de saúde que são monitorados e tratados continuamente em unidades de terapia intensiva – UTIs).

Os SFC têm sido apontados como uma solução poderosa para melhorar os serviços de saúde [5; 29]. De acordo com Lee *et al.* [51], os cenários clínicos tradicionais são

---

sistemas em malha-fechada em que os cuidadores são os “*controladores*”, dispositivos médicos funcionam como “*sensores*” e “*atuadores*”, e pacientes são as “*plantas*” físicas. Como uma classe distinta de SFC, Sistemas Médicos Físico-Cibernéticos (SMFC) modificam este cenário por introduzir entidades computacionais adicionais para ajudar o cuidador no controle da “*planta*”, isto é, no suporte à decisão [45]. Com isso, o suporte à decisão pode ser automatizado, possibilitando que os dados recebidos dos dispositivos de monitoramento sejam analisados por um elemento controlador para estimar o estado de saúde do paciente e prontamente iniciar um tratamento. Isto significa que comandos são emitidos para os dispositivos de atuação, delegados para executar determinadas ações. Esse ciclo de interação entre os elementos participantes do sistema caracteriza o que os engenheiros denominam de “fechamento da malha”.

A tomada de decisão automática, segura e inteligente realizada pelo próprio sistema médico com o auxílio de dispositivos médicos, e as diversas formas de obter/prover informações do contexto envolvido de/para os profissionais de saúde são os maiores diferenciais de um SFC aplicado à saúde. Neste sentido, em que o sistema é tipicamente projetado como uma rede de elementos interagindo com entradas e saídas físicas em vez de como dispositivos autônomos, esses sistemas são denominados de SFC Completos (do inglês, *full-fledge CPS*) [73]. No entanto, para se alcançar tal nível de eficiência em um sistema dessa natureza exige-se planejar, codificar e testar com bastante cautela todas as suas funcionalidades, pois em caso de falha do sistema o maior prejudicado é o paciente, com sérios riscos a sua saúde, inclusive de morte.

Portanto, no domínio da saúde as aplicações são vislumbradas a partir de sistemas de hardware e software específicos, e sua base é a aquisição e a análise de dados levando em consideração as informações do contexto em que o sistema está inserido, principalmente do ambiente, das atividades e do tempo [91]. Para isso, deve existir a colaboração de sensores de múltiplos domínios, com fluxo heterogêneo de informações e inteligente tomada de decisão e atuação [98].

A interação dos profissionais de saúde com os sistemas médicos caracteriza um dos principais aspectos para a tomada de decisão no tratamento de pacientes. Segundo Barbosa *et al.* [8], a maioria dos erros desses sistemas são oriundos de fatores humanos, da perda de precisão em cenários em que a alta acurácia é requerida para certas

operações e da suscetibilidade às condições ambientais hostis. Por exemplo, um dos seis problemas mais relatados em bombas de infusão são causados por fatores humanos, sendo que muitos desses problemas poderiam ser prevenidos [55]. Nestes casos, a interoperabilidade entre dispositivos médicos permitiria a execução automática de bloqueios de segurança que poderiam ajudar a reduzir erros médicos [90].

Além disso, aplicações de saúde não operam em um ambiente totalmente controlado, pois estas são centradas no paciente, cujo comportamento fisiológico não é completamente previsível. Consequentemente, SMFC são sistemas críticos e cientes de contexto por natureza, tendo suas decisões dirigidas pela mudança de contexto. Nestes sistemas, a segurança do paciente é um dos principais requisitos, demandando processos rigorosos de validação para garantir a conformidade com os requisitos do usuário e a correte orientada à especificação.

Logo, é fundamental que um SMFC possa tratar incertezas paramétricas e adaptar suas variáveis em virtude das mudanças no contexto atual do paciente. Entretanto, torna-se necessário o uso de modelos formais de pacientes que possuam a capacidade de fornecer informações que evidenciem a realidade do comportamento do ser humano mediante tratamentos clínicos. Além disso, deve-se possibilitar aos desenvolvedores a realização de testes quanto à efetividade de tais aplicações, com o objetivo de analisar se o sistema proposto está se comportando da forma esperada.

## 1.1 Problemática

Por diferentes razões, a validação de SMFC é inerentemente mais complexa do que a validação de sistemas tradicionais [51]. A partir da perspectiva *física*, os desenvolvedores de SMFC devem considerar a incerteza e a heterogeneidade dos pacientes. Por causa da compreensão insuficiente da dinâmica do corpo humano em resposta aos tratamentos clínicos, não é possível considerar um comportamento determinístico do paciente. Além disso, cada paciente é único, apresentando comportamentos diferentes de acordo com sua idade, histórico médico e aspectos ambientais [17; 75; 26]. Por outro lado, a validação desses sistemas com pacientes reais, sem comprometer a saúde dos pacientes, é um problema complexo [21; 51; 54], o que torna a valida-

ção uma abordagem promissora. No processo de desenvolvimento, a validação ocorre em tempo de projeto (i.e., *early validation*), permitindo analisar aspectos críticos de um sistema antes de sua implementação. Com esse propósito, simulação baseada em modelos é um método comumente aplicado [30].

A partir da perspectiva *cibernética*, os desenvolvedores de SMFC devem lidar com a heterogeneidade e a confiança no funcionamento dos dispositivos. A *US Food and Drug Administration* (FDA), agência regulatória americana, classificou aproximadamente 1.700 diferentes tipos de dispositivos, agrupados em 16 especialidades médicas. A FDA registrou um aumento no número de *recalls* causados por falhas de software em dispositivos médicos [22]. Essas falhas podem estar relacionadas ao projeto de software, à implementação e aos procedimentos de teste, bem como também ao crescimento da complexidade do ambiente de uso, que requerem uma maior conectividade e interoperabilidade desses dispositivos com os sistemas médicos.

Integrar as perspectivas *física* e *cibernética* também traz novos desafios. Como considerar diferentes comportamentos de pacientes e dispositivos para construir uma solução única e integrada de SMFC? Como validar cada aspecto comportamental do SMFC para diferentes tipos de pacientes e dispositivos? Como reutilizar dados de pacientes e dispositivos para simular diferentes SMFC, em diversos cenários clínicos? Como integrar novos conhecimentos sobre pacientes e dispositivos para melhorar a qualidade e a produtividade na validação futura de SMFC? Neste trabalho, o foco é abordar esses desafios.

Vários trabalhos relacionados têm sido propostos na área de SMFC. Por exemplo, os ambientes clínicos integrados (do inglês, *Integrated Clinical Environment – ICE*) propõem um modelo conceitual funcional e padrão que estabelece os requisitos para a integração segura de dispositivos médicos e outros equipamentos em um sistema médico [6]. Baseado neste modelo conceitual, pesquisadores constroem modelos que representam sistemas médicos para auxiliar no projeto e aumentar a resistência a erros nesses sistemas. Um ICE pode ser considerado como um arcabouço de alto nível para descrever um SMFC. Baseado em ICE, Pajic *et al.* [68] apresentam um cenário clínico para analgesia controlada do paciente, que pode se beneficiar da abordagem em malha-fechada para controlar a administração de medicamentos.

Jiang *et al.* [37; 35] publicaram uma série de trabalhos para estabelecer um procedimento para assegurar a operação correta de marca-passos cardíacos usando o conceito de um sistema em malha-fechada. Em Jiang *et al.* [37] os autores fornecem um ambiente para teste em malha-fechada, em que o modelo de paciente, especificamente um modelo formal do coração humano, é o centro de controle de um sistema de marca-passo cardíaco. O objetivo é avaliar a segurança operacional do dispositivo e a sua efetividade com base na condição do paciente.

Miller *et al.* [61] descrevem uma abordagem baseada em maquetes digitais como um método para teste de SMFC. Neste caso, os modelos são utilizados para testar dispositivos médicos por fornecer as entradas e analisar as saídas para permitir a avaliação das funcionalidades desses dispositivos. Os cenários de aplicação da abordagem envolveram um dispositivo de ventilação mecânica conectado a um modelo de pulmão artificial.

Há também trabalhos relacionados à modelagem de pacientes. Por exemplo, Van Heusden *et al.* [89] propõem um modelo de pâncreas artificial para pacientes com diabetes mellitus tipo 1 baseado na teoria de controle. Seu objetivo é melhorar o controle da glicose sanguínea do paciente. Uma das vantagens desse modelo é que ele incorpora a reação à medicação (i.e., insulina) do paciente. Outras abordagens utilizam modelos de contextos clínicos para validar e verificar o controle de software. Kang *et al.* [38], King *et al.* [43], Simalatsar e De Micheli [82] e Li *et al.* [52] focaram nas questões de projeto de SMFC. Eles acoplaram suas soluções para especificar formalismos e ferramentas associadas. Murugesan *et al.* [63] consideraram a verificação composicional de um sistema médico em múltiplos níveis de abstração, com diferentes formalismos usados em cada nível. A sua contribuição é no sentido de prover uma abordagem geral, escalável e prática para verificação em camadas de propriedades em sistemas físico-cibernéticos complexos.

Em geral, tais abordagens apresentam as seguintes limitações em relação aos desafios supracitados:

1. Os modelos para os cenários clínicos são restritos ao propósito específico do sistema, sem suporte para adaptar os respectivos modelos de pacientes e dispositivos para um novo contexto clínico de interesse, tornando o reuso dos modelos inviável;

2. A simulação dos processos físicos ignoram aspectos importantes das dinâmicas dos cenários clínicos, incluindo perturbações externas (e.g., intervenções do usuário) e reação particular de cada paciente aos mesmos estímulos (e.g., administração de medicamentos);
3. Os modelos de pacientes ou focam em variáveis de interesse para cenários clínicos específicos ou negligenciam o relacionamento entre os quatro principais sinais vitais: frequências cardíaca e respiratória, pressão arterial e temperatura corporal. A ausência deste último aspecto contraria o comportamento real do ser humano [17; 75]. Além disso, quando esses modelos são definidos formalmente em notação matemática eles não são representados computacionalmente e *vice-versa*. Logo, esses modelos possuem aplicabilidade limitada para outros cenários clínicos.

Propor uma solução para o problema abordado neste trabalho requer lidar com as diferentes características e dinâmicas dos elementos tanto do mundo físico como do mundo cibernético, conforme todo o contexto em que o sistema encontra-se inserido. No caso de um SMFC, um dos mais importantes desafios é garantir uma interação perfeita, ou seja, eficaz e segura, entre os dispositivos médicos, o sistema de controle, o paciente e os profissionais de saúde responsáveis pelo seu tratamento.

Para não comprometer a segurança do paciente durante a interação entre todos esses elementos, o projeto de um SMFC requer o uso de modelos como meio de solução e, principalmente, modelos de elementos tendo a concorrência como principal característica, dado que o ambiente em que esses sistemas operam não são totalmente controlados e muitas coisas ocorrem simultaneamente ao longo de sua execução. Por este motivo, a solução para abordar o problema discutido nesse trabalho deve levar em consideração todos esses aspectos dessa classe de sistemas aplicados à saúde.

## 1.2 **Objetivos**

Neste trabalho é proposta uma abordagem formal para validação de SMFC usando modelos de simulação baseados no paradigma de Projeto Orientado a Atores (do inglês,

*Actor-Oriented Design*) [2]. Uma biblioteca de modelos reutilizáveis é fornecida, sendo esta composta de dois tipos de modelos: pacientes e dispositivos. Modelos de pacientes são baseados em diretrizes médicas e bases de dados clínicas. Modelos de dispositivos são construídos a partir de especificações técnicas de dispositivos sensores e atuadores. Os modelos são construídos como componentes reutilizáveis, sendo personalizáveis o suficiente para ser reutilizados em diferentes SMFC, e podem também serem adaptados ou estendidos para se adequarem às especificidades dos projetos de SMFC.

Um processo de composição é proposto para apoiar os desenvolvedores a integrar componentes para construir seus modelos específicos de SMFC, usando modelos existentes de pacientes e dispositivos. Os modelos de SMFC integrados podem ser simulados e analisados para identificar comportamentos indesejados em tempo de projeto. Diretrizes de extensão e adaptação para cada tipo de modelo são também fornecidas, permitindo construir novos modelos reutilizáveis para enriquecer a biblioteca de modelos. Esse processo completo para melhorar o reúso e a produtividade na validação de SMFC é a principal contribuição deste trabalho.

A abordagem proposta foi aplicada em três diferentes cenários clínicos para avaliar seu potencial reúso para diferentes contextos. Adicionalmente, a abordagem foi validada com sucesso por meio de uma avaliação empírica com desenvolvedores, aplicando o método GQM (do inglês, *Goal-Question-Metric*) [9]. Finalmente, os modelos de simulação foram verificados com sucesso usando a ferramenta *Simulink Design Verifier* [57]. Os requisitos funcionais e de segurança foram verificados formalmente e a cobertura do modelo de um dos três cenários clínicos foi analisada.

### 1.3 Relevância

Nas últimas duas décadas, os sistemas computacionais destinados aos cuidados com a saúde humana têm desempenhado importantes papéis dentro do campo da medicina. Por exemplo, a telemedicina acompanhou a evolução das Tecnologias de Informação e Comunicação (TICs), tais como computadores, internet e, recentemente, os telefones inteligentes, utilizando-as para superar barreiras geográficas e expandir o acesso dos serviços de saúde a áreas remotas ou àquelas que necessitam melhorar a qualidade

da saúde da população. Sua finalidade é promover a troca de informações entre os profissionais de saúde e o suporte clínico à distância [94].

Recentemente, esse tipo de serviço tem evoluído ainda mais, principalmente com a possibilidade de interação direta das entidades computacionais com o paciente, intermediada pelos profissionais de saúde ou pelo próprio paciente. Essa interação conjunta entre o paciente, os profissionais de saúde e os computadores, com o objetivo de fornecer um serviço de saúde de melhor qualidade ao paciente em tratamento, caracteriza a classe de sistemas denominada de SMFC.

Uma vez que os diferentes cenários clínicos de um SMFC engloba uma heterogeneidade de dispositivos médicos integrados e controlados por sistemas computacionais de apoio à decisão, muitas vezes com intervenções automatizadas, a segurança do paciente deve ser garantida. Deste modo, torna-se fundamental que os desenvolvedores de SMFC utilizem-se das abordagens e ferramentas mais adequadas para verificar e validar tais sistemas. É, portanto, essa a principal motivação para a realização deste trabalho.

Outra motivação, que torna relevante este trabalho, está relacionada aos dispositivos médicos (DMs). Nos modernos sistemas de saúde, os dispositivos médicos, além de serem utilizados para auxiliar a tomada de decisão clínica, facilitam o cotidiano de seus usuários, ajudando as pessoas a viverem com saúde e, ainda mais importante, aos cuidadores a salvar vidas. Considerando a importância de minimizar as falhas nos projetos de software<sup>1</sup> de DMs, verifica-se a necessidade de melhorar a qualidade dos testes não somente dos sistemas médicos, mas dos próprios dispositivos que irão compor esses sistemas no sentido de garantir a segurança do paciente.

Para os desenvolvedores dessa classe de sistemas, a relevância do trabalho é evidenciada por meio dos seguintes aspectos inerentes à solução apresentada para o problema em questão, que são:

1. Minimizar o esforço do desenvolvedor no processo de validação de SMFC;
2. Dispensar, a priori, a necessidade de utilizar pacientes reais ou cobaias durante

---

<sup>1</sup>Nesse contexto, o termo *software* pode referir-se ao: 1) programa responsável pelo controle total ou parcial do dispositivo médico; 2) próprio dispositivo; ou, 3) programa utilizado para fabricar o dispositivo [22].

a execução dos testes iniciais de um SMFC, momento em que há maior probabilidade de detectar comportamentos indesejados com base no modelo do sistema.

Como evidência da relevância, contribuição e viabilidade prática deste trabalho, as seguintes publicações são enumeradas:

- SILVA, L. C.; PERKUSICH, M.; BUBLITZ, F. M.; ALMEIDA, H. O. and PERKUSICH, A. A model-based architecture for testing medical cyber-physical systems. In: Annual ACM Symposium (SAC '14), 2014, Gyeongju. *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York: ACM Press. p. 25–30.
- SILVA, L. C.; ALMEIDA, H. O. and PERKUSICH, A. Uma Arquitetura Baseada em Modelos para Sistemas Médicos Físico-Cibernéticos. In: Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP '14), 2014, Brasília. *Anais do VI Simpósio Brasileiro de Computação Ubíqua e Pervasiva*. Porto Alegre: SBC, 2014. p. 862–871.
- SILVA, L. C.; PERKUSICH, M.; ALMEIDA, H. O.; PERKUSICH, A.; LIMA, M. A. M. and GORGÔNIO, K. C. A Patient Baseline Model to Support Testing of Medical Cyber-Physical Systems. In: World Congress on Medical and Health Informatics (MEDINFO '15), 2015, São Paulo. *Proceedings of the 15th World Congress on Medical and Health Informatics*. São Paulo: IMIA and IOS Press, 2015. p. 549–553.
- SILVA, L. C.; ALMEIDA, H. O. and PERKUSICH, A. Verification and Validation of Medical Cyber-Physical Systems. In: Isabel Miranda; Rui Lopes Rijo; Maria Manuela Cruz-Cunha; Ricardo Martinho (Org.). *Encyclopedia of E-Health and Telemedicine*. 1ed. New York: IGI Global, 2015. (Approved).
- SILVA, L. C.; ALMEIDA, H. O.; PERKUSICH, A. and PERKUSICH, M. A Model-Based Approach to Support Validation of Medical Cyber-Physical Systems. *Sensors*, 15(11): 27625–27670, Oct. 2015.

## 1.4 Estrutura do Documento

O restante deste documento está organizado da seguinte forma:

- No Capítulo 2 são apresentados os conceitos fundamentais para o entendimento do trabalho.
- No Capítulo 3 é apresentada a biblioteca de modelos, fornecendo detalhes de como construir novos modelos para pacientes e dispositivos.
- No Capítulo 4 é detalhado o processo de composição para construir modelos de SMFC baseados nos modelos existentes de pacientes e dispositivos.
- No Capítulo 5 é discutida a validação da abordagem proposta, incluindo sua aplicação para três diferentes cenários e sua avaliação empírica.
- No Capítulo 6 é descrito o processo de análise formal dos modelos.
- No Capítulo 7 são apresentadas as conclusões e os trabalhos futuros.

# Capítulo 2

## Fundamentação

Neste capítulo são apresentados os conceitos fundamentais para o entendimento do restante deste trabalho.

### 2.1 Sistemas Físico-Cibernéticos

Muito Recentemente, a comunidade passou a entender que os desafios atuais em Sistemas Embarcados (SE) resultam de sua interação com processos físicos, e não dos recursos limitados dos dispositivos. Essa mudança de percepção, acarretada principalmente pela evolução tecnológica, levou a uma redefinição de conceitos na área de sistemas embarcados em tempo-real. Isto culminou no surgimento do termo *Cyber-Physical Systems* (CPS) – Sistema Físico Cibernético (SFC) – cunhado em 2006 por Helen Gill, membro pesquisadora da NSF (*National Science Foundation* – U.S.A.), que refere-se a integração da computação aos processos físicos.

Em SFC, computadores embarcados e redes monitoram e controlam os processos físicos, geralmente com ciclos de realimentação (i.e., *feedback loops*) em que os processos físicos afetam o modo como os sistemas computacionais devem agir e, consequentemente, essas ações interferem diretamente no comportamento desses processos. Portanto, o projeto de tais sistemas requer a compreensão conjunta da dinâmica de computadores, software, redes e processos físicos, sendo o tempo um fator crítico o suficiente para determinar o funcionamento correto do sistema. *Concorrência* é outro aspecto intrínseco nos SFC, uma vez que medir e controlar a dinâmica dos processos

físicos exige orquestrar as ações que os influenciam. Com isso, modelar a dinâmica temporal de seus componentes, bem como especificar propriedades de concorrência, torna-se uma tarefa indispensável para qualquer aplicação desses sistemas [49].

Segundo Li *et al.* [53] o principal uso de SFC é para o suporte à decisão em tempo real, sendo a qualidade de serviço altamente dependente de fatores como *segurança* e *previsibilidade*. O fator *segurança* refere-se à garantia que os impactos das operações do sistema permaneçam dentro dos limites desejados, enquanto a *previsibilidade* remete à capacidade do sistema reagir às condições inesperadas por meio de sua auto adaptação [46; 49].

Dentre os principais desafios de pesquisa em SFC elencados por Park, Zheng and Liu [70] estão questões de modelagem e controle de sistemas híbridos<sup>1</sup>, redes de sensores e atuadores, abstração computacional e arquitetura. Todas essas questões estão relacionadas ao modo de como lidar com os aspectos intrínsecos de sistemas centrados em processos físicos. São exemplos desses aspectos, a diversidade e a quantidade de informações a serem capturadas e tratadas, a heterogeneidade de elementos interagindo e a simultaneidade de eventos ocorrendo, seja do ambiente no qual o sistema encontra-se inserido ou do próprio sistema.

Os domínios de aplicação dos SFC são amplos e incluem, por exemplo, Automação Industrial (controle de processos), Robótica (telepresença), Transporte (sistemas automotivos avançados), Infraestrutura em grande escala (sistemas de comunicação), Militar (sistemas de defesa) e Saúde (sistemas e dispositivos médicos), este último domínio comumente conhecido como *Healthcare* [46].

Como discutido no Capítulo 1, SFCs aplicados à saúde são denominados de *Sistemas Médicos Físico-Cibernéticos* (SMFC). No caso de SMFCs, os processos físicos a serem controlados estão relacionados aos aspectos comportamentais do paciente, sendo este, portanto, o foco de atenção dessa subcategoria de SFCs.

---

<sup>1</sup>*Sistemas Híbridos* são àqueles que combinam as dinâmicas dirigidas a tempo e a evento [14]. Ou seja, integra a dinâmica contínua do mundo físico e a lógica discreta do mundo cibernético.

## 2.2 Ambiente Clínico Integrado

Um ambiente clínico integrado (do inglês, *Integrated Clinical Environment* – ICE), como definido no padrão **F2761-09(2013)** da *ASTM International*, é aquele que combina dispositivos médicos, heterogêneos e interoperáveis, e outros equipamentos integrados para criar um sistema médico para o cuidado de um único paciente de alta gravidade [6]. Trata-se, portanto, de um ambiente em que o monitoramento, o tratamento e o diagnóstico é realizado em um único paciente. Exemplos de ICE incluem salas operatórias, unidades de terapia intensiva, unidades de emergência, transportes (e.g., ambulância e aeronaves) e, até mesmo, residências.

De uma série de padrões, o padrão **F2761-09(2013)** especifica os requisitos gerais, um modelo e um arcabouço para a integração segura de dispositivos médicos e equipamentos, independente de fabricante. Sua ênfase é a criação de um ICE, utilizando projeto baseado em modelo. O objetivo é desenvolver sistemas médicos com uma maior resistência à erros, melhorar a segurança do paciente e a eficácia de seu tratamento.

Em se tratando do modelo proposto nesse padrão, denominado de *Modelo Conceitual Funcional ICE* (ver Figura 2.1), seus elementos essenciais são os *dispositivos médicos* (2), as *interfaces de equipamento* (4), o *controlador de rede* (5) e o *supervisor* (7). Os números entre parênteses correspondem aos respectivos identificadores na Figura 2.1, na qual estão representados os relacionamentos entre os elementos do modelo ICE. Os dois principais indivíduos participantes do modelo ICE são o paciente (1), quem encontra-se em tratamento, e o operador (9), quem manipula os equipamentos.

O papel de uma *interface de equipamento ICE* é fornecer um canal de comunicação de um equipamento (3) para o *controlador de rede ICE*, responsável por estabelecer tal comunicação desses equipamentos com os demais elementos do ICE. Já o *supervisor ICE* provê uma plataforma para integração funcional entre os equipamentos por intermédio do controlador de rede, além de fornecer a lógica da aplicação e uma interface para os operadores de equipamentos.

Vale lembrar que, o modelo ICE é um *modelo de referência* para criação de ambientes clínicos integrados centrados no paciente. Portanto, a configuração física apresentada na Figura 2.1 é apenas para demonstrar como os seus elementos se relacionam.

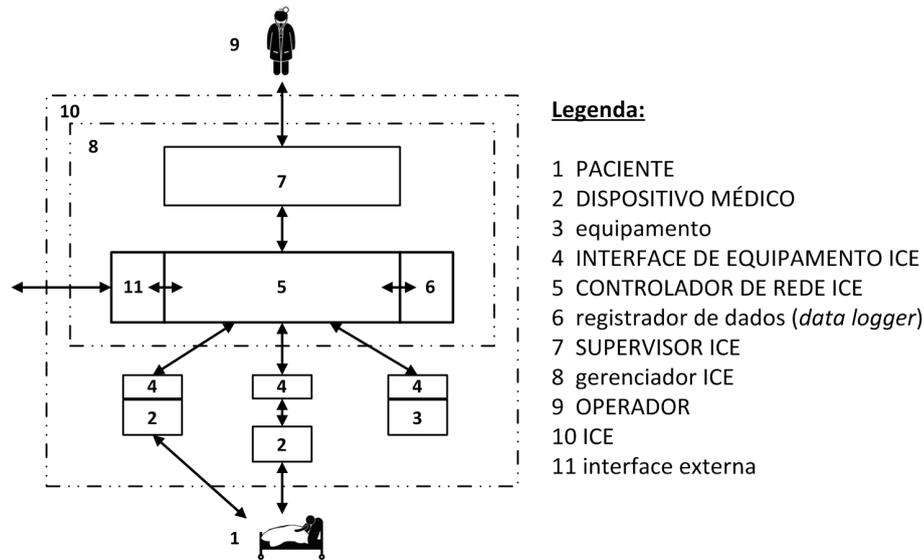


Figura 2.1: Relacionamento entre os elementos do modelo funcional conceitual ICE. Adaptado de [6].

Dessa forma, a implantação desse modelo pode estabelecer diferentes configurações, desde configurações mais simples como, por exemplo, usando um único dispositivo médico (2) com uma interface de equipamento ICE (4); ou de sistemas médicos complexos, que combinam vários ou todos os elementos do modelo ICE.

Dentre os benefícios clínicos de integrar dispositivos médicos independentes, enumeram-se: (a) a capacidade de combinar dados coletados por diferentes fontes para produzir novas informações; (b) o suporte à decisão clínica; e (c) a capacidade de implementar controle distribuído dos dispositivos para interligar a segurança e o controle em malha fechada.

## 2.3 Projeto Baseado em Modelos

A OMG<sup>2</sup> (*Object Management Group*) é um consórcio internacional, sem fins lucrativos, fundada em 1989 com o objetivo de reduzir a complexidade, minimizar os custos e acelerar a introdução de novas aplicações de software no mercado. Para isso, teve como iniciativa desenvolver padrões para tecnologias, incluindo padrões de modelagem como, por exemplo, a linguagem de modelagem unificada (do inglês, *Unified Modeling Language* – UML), a notação e modelo de processo de negócio (do inglês, *Business*

<sup>2</sup><http://www.omg.org/>

*Process Model and Notation* – BPMN), e a arquitetura dirigida a modelos (do inglês, *Model Driven Architecture* – MDA) [67].

Especificamente falando da MDA, cuja primeira especificação foi publicada em 2001, nela é enfatizado o papel dos modelos como artefatos primários no desenvolvimento de software, argumentando-se que os modelos deveriam ser precisos o bastante para suportar transformações automatizadas entre as fases do ciclo de vida [93].

Segundo a OMG, a MDA é uma abordagem que usa modelos no processo de desenvolvimento de software, separando a lógica de negócio e a aplicação a partir das capacidades tecnológicas das plataformas computacionais. Sua principal promessa é permitir a definição de aplicações e modelos de dados com flexibilidade a longo prazo no que diz respeito à implementação, integração, manutenção, teste e simulação. Portanto, com MDA almeja-se alcançar portabilidade, interoperabilidade e reuso por meio da separação arquitetural das preocupações a serem consideradas em um projeto de software [67].

Para Ameller [4], MDA é uma forma particular de desenvolvimento dirigido a modelos (do inglês, *Model-Driven Development* – MDD) que faz uso dos padrões da OMG. Sendo MDD um subconjunto da engenharia dirigida a modelos (do inglês, *Model-Driven Engineering* - MDE) com foco específico no processo de desenvolvimento, possibilitando a geração de código a partir dos modelos. Com um conceito mais geral, MDE engloba outros processos que utilizam modelos para apoiar outras atividades relacionadas à engenharia de software, tais como verificação e validação.

Como o objetivo geral deste trabalho está relacionado com as atividades da etapa de Verificação e Validação (V&V) de SMFC, o restante desta seção terá como ponto de exploração MDE. Dessa forma, é importante apresentar um breve histórico da evolução que tem tornado MDE uma tendência prática no projeto de software.

De acordo com Schmidt [76], inúmeros esforços passados criaram tecnologias que elevaram o nível de abstração usado para desenvolver software, como no caso da engenharia de software auxiliada por computador (do inglês, *Computer-Aided Software Engineering* – CASE) na década de 1980. Com ênfase em ferramentas que permitissem o desenvolvedor expressar seus projetos usando representações gráficas de propósito geral para programar software, a CASE não foi amplamente adotada na prática, tendo

pouco impacto na indústria de software comercial até o início dos anos de 1990. Dentre os problemas que prejudicaram sua adoção fala-se no restrito mapeamento de seus elementos para as plataformas existentes na época, na incapacidade de produção de sistemas em grande escala e na restrição aos domínios de aplicação devido a falta de personalização.

Apesar da maturidade das linguagens de programação da terceira geração, tais como C++, Java e C#, que proporcionaram ao desenvolvedor uma melhor experiência no projeto de software, os avanços ocorreram mais lentamente do que a complexidade das plataformas computacionais, como no caso de J2EE, .NET e CORBA. Como consequência, os desenvolvedores necessitam despende um maior esforço manual para garantir a portabilidade de suas aplicações para as novas plataformas ou versões posteriores de uma mesma plataforma [76].

Portanto, a abordagem MDE surgiu como uma promessa para auxiliar o desenvolvedor a lidar com esses problemas que tornam cada vez mais complexo o projeto de software. Conforme declara Schmidt [76], dentre as tecnologias que promovem a aplicação prática da MDE estão as linguagens de modelagem específicas de domínio e os motores de transformação de modelos. Por meio de metamodelos<sup>3</sup>, as linguagens de modelagem específicas de domínio possibilitam definir os relacionamentos entre os conceitos em um domínio de aplicação e especificar as semânticas e restrições associadas desse domínio. Usando os motores de transformação de modelos consegue-se analisar certos aspectos dos modelos e sintetizar diversos tipos de artefatos a partir dos modelos como, por exemplo, código fonte, entradas para simulação, representações alternativas do modelo e descrições que podem ser usadas para diferentes propósitos (e.g., teste).

Na busca de compreender a aplicação prática de MDE na indústria de software, Hutchinson *et al.* [32] analisaram os fatores técnicos, organizacionais e sociais que poderiam influenciar no sucesso e na falha de sua adoção. A pesquisa foi realizada durante um período de doze meses e o método aplicado para coleta dos dados foi por meio de questionários e entrevistas com profissionais de diferentes empresas praticantes de MDE, além de estudos observacionais nas próprias empresas. Os resultados dessa análise são bastante sugestivos, evidenciando, por exemplo, que as atividades relacionadas à MDE

---

<sup>3</sup>Um *metamodelo* descreve os conceitos que podem ser usados para construir um modelo [85].

têm um impacto positivo na produtividade ou manutenibilidade do software. Dentre essas atividades destacam-se o uso de modelos para comunicação entre os membros do projeto, compreensão de um problema em um mais alto nível de abstração e documentação de projetos, bem como para gerar código, testar e simular. Do ponto de vista de treinamento, há uma maioria significativa (74%) dos participantes da pesquisa que acham que usar MDE requer treinamento extra. Outro aspecto analisado são os custos das ferramentas oferecidas para a prática de MDE, que além de caras são imaturas, e muitas vezes são utilizadas de forma inadequada. A partir dos dados coletados do grupo de praticantes de MDE, e que foram analisados nessa pesquisa, outros resultados sugestivos são discutidos.

Apesar dos potenciais benefícios mencionados pelos especialistas de MDE participantes da pesquisa, os autores afirmam que não há evidências empíricas de que a implantação e o uso da abordagem contribuem para o sucesso de um projeto de software. Conseqüentemente, isso gera uma certa desconfiança da abordagem MDE e de suas ferramentas, dificultando sua ampla adoção pelas empresas [32].

No entanto, para Schmidt [76] ter uma notação visual (e.g., modelos) que agregue a semântica e a sintaxe de domínios específicos ajuda a achatar a curva de aprendizagem na aplicação prática de MDE por especialistas no assunto, tais como os engenheiros de sistemas e os arquitetos de software. Além disso, como as ferramentas de MDE impõem restrições para o domínio de interesse e realizam verificação dos modelos, torna-se possível detectar e prevenir erros nas etapas iniciais do ciclo de vida do desenvolvimento de software. Por conseguinte, consegue-se aumentar a confiabilidade do software desenvolvido e atender as necessidades de seus usuários.

Com o objetivo de identificar e discutir sobre as abordagens e ferramentas atualmente adotadas na prática de MDE no domínio de SMFC, em especial àquelas com foco nas atividades de V&V, Silva *et al.* [78; 79] fizeram uma Revisão Sistemática da Literatura (RSL). Tal revisão pode ser consultada no Apêndice A deste trabalho. Para guiar o processo de RSL, um protocolo de pesquisa foi definido, conforme detalhado no Apêndice B. As quatro principais fontes de busca de indexação de artigos científicos na área de computação foram pesquisadas, são elas: *IEEE Xplore DL*, *ACM DL*, *Science Direct* e *Portal de Periódicos CAPES* (que engloba *Scopus*, *Web of Science* e *Springer-*

*Link*). Outros artigos identificados na literatura, mas que não foram retornados nas pesquisas realizadas nas referidas fontes de busca, foram considerados no processo de revisão.

No estágio preliminar do processo de revisão, 43 estudos atenderam aos critérios iniciais do protocolo RSL. Ao levar em consideração os critérios de inclusão e exclusão especificados no protocolo RSL, 25 estudos foram aceitos para a etapa de extração de informações. Nessa etapa, todos os trabalhos foram lidos por completo e classificados conforme sua contribuição principal, formalismos utilizados, ferramentas de suporte adotadas e procedimentos de teste aplicados na avaliação dos sistemas desenvolvidos. Ao término dessa etapa, apenas 16 estudos foram considerados para a etapa de sumarização dos resultados, etapa em que foram feitas análises quantitativas e, principalmente, qualitativas acerca dos estudos.

Alguns dos resultados desta revisão serão brevemente apresentados e discutidos nessa seção. Por exemplo, neste levantamento 62,5% dos estudos utilizam alguma abordagem baseada em modelos como principal método formal para representar o comportamento de elementos em SMFC. Em geral, os modelos construídos são apoiados e/ou estão integrados com outros formalismos, tais como Autômatos e suas variantes (temporizados, híbridos e com tarefas) fato que ocorre em 60% dos casos.

No que diz respeito às ferramentas de apoio para V&V de SMFC, o *Simulink* da Mathworks<sup>®</sup> é a preferência dos projetistas e pesquisadores da área, sendo a ferramenta adotada em 50% dos estudos analisados. Destes estudos, metade também utilizam o *UPPAAL*<sup>4</sup> como ferramenta para verificação dos modelos propostos. Outros aspectos interessantes foram percebidos como, por exemplo, o uso da ferramenta *Stateflow*, também da Mathworks<sup>®</sup>, para representar como o sistema reage aos eventos, às condições baseadas em tempo e aos sinais de entrada externa. Além disso, *TIMES*<sup>5</sup> é vista como uma das principais ferramentas para geração de código a partir de modelos.

Quanto às estratégias utilizadas para testar SMFC, *model checking* e simulação representam, respectivamente, os processos de verificação e validação de SMFC em 62,5% e 50% dos estudos. Ambas as técnicas foram utilizadas conjuntamente em

---

<sup>4</sup><http://www.uppaal.org/>

<sup>5</sup><http://www.timestool.com/>

37,5% dos estudos analisados. Por fim, com 56,25% dos estudos propondo algum tipo de abordagem como contribuição, evidencia-se a necessidade de garantir a segurança do paciente em um SMFC.

Na etapa de interpretação dos resultados alcançados a partir desta revisão, pôde-se verificar que o principal foco dos estudos analisados é verificar propriedades de segurança dos dispositivos médicos ou sistemas propostos. Neste sentido, os autores apresentam cenários de uso específicos para apresentar provas de conceito para cada projeto. Isto demonstra a importância da realização de testes mais rigorosos para SMFC, devido aos fatores críticos e riscos diretamente associados aos seus projetos como, por exemplo, a forma de lidar com o tempo, a concorrência, a complexidade e a incerteza dos processos físicos controlados (e.g, pacientes), bem como a heterogeneidade e a necessidade de integração dos dispositivos médicos a um sistema maior de supervisão e controle para a tomada de decisão.

Ao confrontar os principais estudos identificados no processo RSL supracitado com este trabalho percebe-se que, enquanto os autores dos trabalhos relacionados focam em soluções para contextos específicos, o foco deste trabalho é fornecer uma solução geral para diferentes contextos clínicos. Outro aspecto está relacionado ao reuso dos modelos de pacientes (i.e., órgãos ou funções orgânicas específicas) construídos, pois esses modelos, por exemplo, não estabelecem um comportamento mínimo para caracterizar o comportamento de um paciente, tais como seus sinais vitais. Essa negligência ao desconsiderar os principais sinais vitais (i.e., frequência cardíaca, frequência respiratória, pressão arterial e temperatura corporal), que representam a condição básica de saúde de um indivíduo, tornam o reuso desses modelos inviável. Isto porque seria necessário um maior esforço na extensão desses modelos para incorporar novos comportamentos de um paciente para outros contextos clínicos. Diferentemente desses trabalhos, a solução proposta utiliza como ferramenta de modelagem o *Ptolemy II*, devido a alguns fatores, tais como sua capacidade simular diferentes semânticas de execução (requisito fundamental para SMFCs), o grande número de atores disponíveis em sua biblioteca e o alto nível abstração para construir modelos para o propósito de validação. No entanto, uma grande limitação do *Ptolemy II* é a ausência de mecanismos para verificação e análise de cobertura dos modelos, havendo a necessidade do desenvolvedor utilizar outras

ferramentas (e.g., *Simulink* e *UPPAAL*) para remodelar os comportamentos funcionais e propriedades do modelo que ele queira verificar e analisar do ponto de vista formal.

## 2.4 Paradigma de Projeto Orientado a Atores

O paradigma de Projeto Orientado a Atores (do inglês, *Actor-Oriented Design* – AOD) trata-se de uma metodologia de projeto baseada em componentes denominados *atores*, cuja ênfase é a concepção de modelos formalizados de concorrência [49].

Agha [2] foi um dos precursores do termo *ator*, evoluindo o conceito de agente de raciocínio autônomo descrito por Hewitt (1977) para, um agente computacional que possui um elemento de controle independente (*thread*), e se comunica por meio da troca de mensagens assíncronas.

### 2.4.1 Ator

*Atores* são componentes que executam concorrentemente e compartilham dados entre si por meio do envio de mensagens via portas, usadas como interfaces de comunicação [72]. Sendo esse conjunto de mensagens trocadas entre atores via uma única porta de cada ator, mencionado como um *signal*.

A notação gráfica de um ator ( $A$ ) em particular é representada por meio de um retângulo (*box*) com uma porta de entrada ( $x$ ) e uma porta de saída ( $y$ ), conforme ilustrado na Figura 2.2.

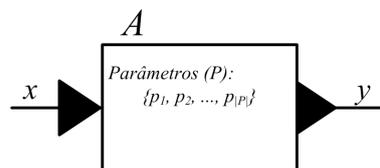


Figura 2.2: Representação gráfica de um componente *ator*. Adaptado de [49].

Os sinais de entrada e saída são funções da forma  $x : \mathbb{R} \rightarrow \mathbb{R}, y : \mathbb{R} \rightarrow \mathbb{R}$ , em que o domínio da função pode representar, por exemplo, em sistemas de tempo contínuo o *tempo*, e, o contradomínio o valor do sinal em um instante de tempo. Uma representação equivalente é dada pela função do tipo  $A : X \rightarrow Y$ , conjunto de funções que

mapeiam os reais para os reais, onde  $X = Y = \mathbb{R}^{\mathbb{R}}$  (domínio e contradomínio em  $\mathbb{R}$ ). Adicionalmente, um ator  $A$  pode depender de um conjunto de parâmetros ( $P$ ) do sistema que se deseja modelar, podendo estes serem opcionalmente incluídos na notação do ator.

Portanto, os mesmos conceitos formais especificados em Lee e Seshia [49] para definir um modelo de sistema físico, podem também ser utilizados na definição de um ator.

#### definição 2.4.1 – Ator

De maneira simples, um ator  $A$  pode ser definido como uma *4-tupla*

$$A = \langle X, P, Y, F \rangle$$

onde:

- ▣  $P$  é o conjunto finito de parâmetros internos;
- ▣  $X$  e  $Y$  é, respectivamente, o conjunto finito de portas de entrada e portas de saída, tal que  $X \cup Y \neq \emptyset$ . Isso significa que um ator deve possuir pelo menos uma porta de entrada ou de saída;
- ▣  $F$  é o conjunto finito de funções que mapeia sinais de entrada para sinais de saída do ator. Cada função é denotada por  $f : X \rightarrow Y$ , sendo  $X = Y = \mathbb{R}^{\mathbb{R}}$  (domínio e contradomínio em  $\mathbb{R}$ ).

□

No entanto, um ator pode possuir múltiplas *portas de entrada* ( $x$ ) e/ou *portas de saída* ( $y$ ), como se pode observar na Figura 2.3. Por exemplo, esse ator poderia representar uma função de um sistema físico para calcular a velocidade instantânea ( $y_1$ ) e a aceleração ( $y_2$ ), a partir da posição ( $x_1$ ) de um objeto com respeito ao tempo ( $p_1$ ), dado pela função  $s(t) = -2t^2 + 3t + 9$ . A função desse ator poderia utilizar os conceitos de derivada primeira e segunda, respectivamente, para calcular os valores de  $y_1$  e  $y_2$ .

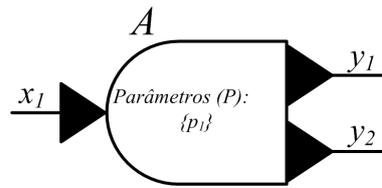


Figura 2.3: Representação gráfica de um componente *ator* com múltiplas portas.

Percebe-se nesse caso que, o *box* do ator possui uma representação visual personalizada, sendo tais personalizações conhecidas como *ícones*. Os *ícones* são blocos de construções úteis para instanciar atores na concepção de modelos de sistemas com comportamentos dinâmicos.

Vale ressaltar que, apesar de ter sido apresentada uma definição formal para um componente *ator*, não se deve restringir a forma de como esses componentes são especificados. Por exemplo, Tripakis *et al.* [86], definem um *ator* e os conceitos relacionados, de forma ainda mais formal e detalhada. O motivo dessa “abertura” na definição de um ator é que, em seu conceito genérico, pode representar, por exemplo, um fenômeno físico, um hardware, uma função ou programa completo, cuja estrutura, nível de complexidade e lógica embutida em seus modelos os tornam componentes bastante flexíveis.

## 2.4.2 Modelo de Atores

Um *box* composto por um conjunto de atores relacionados é denominado de *Modelo de Ator* (do inglês, *Actor Model*). Os modelos de atores podem ser agregados, formando uma *composição em cascata* como, por exemplo,

$$x_1 \rightarrow |A_1| \rightarrow y_1 \rightarrow x_2 \rightarrow |A_2| \rightarrow y_2,$$

onde  $y_1 = x_2$  ou  $\forall t \in \mathbb{R}, y_1(t) = x_2(t)$ . Na Figura 2.4 encontra-se representada graficamente essa definição para um modelo de ator.

Assim como um ator, um modelo de atores pode possuir múltiplos sinais de entrada e/ou saída, bem como utilizar-se de *ícones* para abstrair a estrutura interna e/ou a complexidade do modelo, evidenciando o conceito de *Hierarquia de Modelos de Atores*.

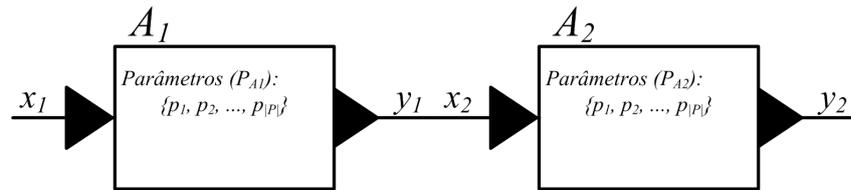


Figura 2.4: Representação gráfica de um *modelo de atores*. Adaptado de [49].

### 2.4.3 Hierarquia de Modelos de Atores

Tradicionalmente, sistemas que envolve a dinâmica de elementos do mundo físico são considerados de natureza complexa, o que torna também complexos os modelos contruídos para representar tal dinâmica [72]. Logo, uma representação mais simplificada de um sistema complexo, que facilite sua compreensão e análise, será de tal modo avaliada como um bom modelo. É com essa ideia chave que foi estabelecida a noção de *Modelo Hierárquico de Atores*.

Na hierarquia de modelos de atores, um ator que aparenta ser simples em uma visão mais abstrata do modelo de um sistema, pode ter em sua estrutura interna um outro modelo ou conjunto de modelos, como ilustrado na Figura 2.5.

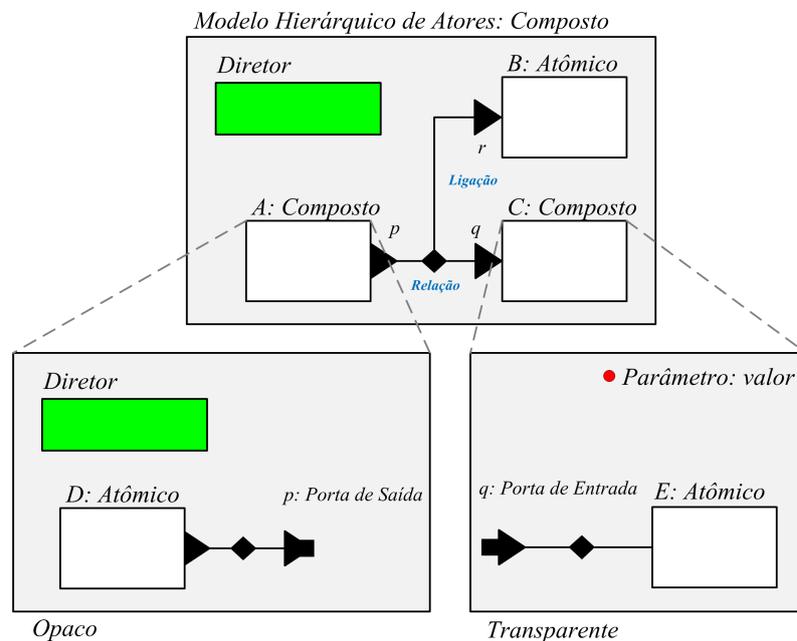


Figura 2.5: Exemplo de um modelo hierárquico de atores, usando a notação gráfica. Adaptado de [72].

Com isso, pode-se definir a seguinte taxonomia para componentes do tipo ator:

**definição 2.4.1.1** – *Ator Atômico* é aquele que não é internamente definido como um modelo de atores.

□

**definição 2.4.1.2** – *Ator Composto* é aquele constituído por meio da composição de outros atores.

□

No exemplo de modelo hierárquico de atores apresentado na Figura 2.5, as portas  $p$  e  $q$  une os níveis de hierarquia. Isso possibilita que o ator  $D$  se comunique com o ator  $E$ , mesmo que esses atores estejam localizados em diferentes níveis de abstração. Considerando a notação gráfica para um modelo de atores, o símbolo notacional representado por um diamante negro  $\blacklozenge$  significa uma *relação*), e possibilita que o sinal de um ator (e.g.,  $A$ ) seja compartilhado, simultaneamente, com mais de um ator (e.g.,  $B$  e  $C$ ) por meio de *ligações*. Enquanto que o círculo em vermelho  $\bullet$  indica um parâmetro do modelo.

Em relação aos atores compostos, esses podem ainda ser classificados como *opacos* ou *transparentes*. Um ator composto é dito ser *opaco* quando possui o componente ou bloco *Diretor*, que define a semântica do modelo. Ao contrário, um ator composto que não incorpora ao seu modelo o componente *Diretor* é dito ser *transparente*.

Desse modo, a semântica especificada por um *Diretor*, descreve o *Modelo de Computação*.

#### 2.4.4 Modelos de Computação

Um *Modelo de Computação* (do inglês, *Model of Computation* – MoC) é um elemento que especifica a semântica de composição e comunicação entre atores concorrentes de um modelo de sistema, por meio de três conjuntos de regras, a saber: 1 – Especificar o que constitui um componente; 2 – Especificar os mecanismos de concorrência; 3 – Especificar os mecanismos de comunicação [49].

Em geral, MoCs são *determinísticos*. Nesse contexto, isso significa dizer que, apesar da concorrência no modelo, os sinais enviados entre os atores não dependem das decisões arbitrárias escalonadas [72]. Caso se deseje comportamentos *não-determinísticos* em um modelo, esses devem ser explicitamente especificados dentro do próprio modelo de atores por quem o constrói.

Os relacionamentos entre MoCs podem ser visualizados na Figura 2.6. Devido a natureza híbrida de um SMFC em lidar com aspectos comportamentais do mundo físico e do mundo cibernético, exigindo modelos heterogêneos, neste trabalho serão discutidos os modelos de computação de *Tempo Contínuo*, de *Evento Discreto* e de *Máquinas de Estados*. Quanto aos demais MoCs, a maioria são descritos por Lee & Seshia [49] e Lee *et al.* [51], referências estas também utilizadas como base para discutir sobre os MoCs apresentados a seguir.

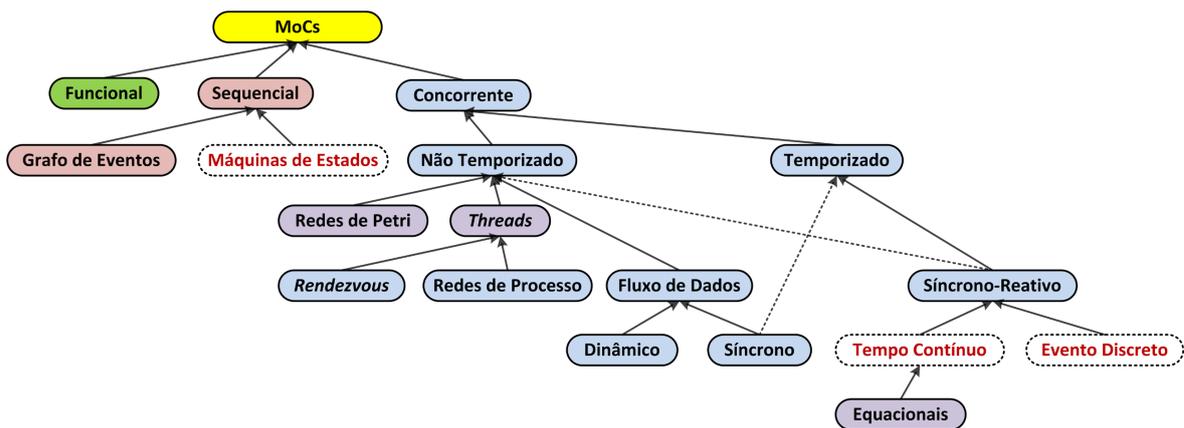


Figura 2.6: Relacionamentos entre os MoCs. Adaptado de [72].

## Tempo Contínuo

De forma geral, a dinâmica de sistemas físicos, tais como a dinâmica de movimentos da *Mecânica Clássica*, é dada por meio de equações diferenciais ordinárias (do inglês, *Ordinary Differential Equations* – ODEs). Por exemplo, para representar essas equações, modelam-se atores específicos que fornecem o comportamento de *integradores* matemáticos, e os conectam formando ciclos de realimentação, em que cada conexão representa uma *função de tempo contínuo* e os atores as relações entre essas funções. Desse modo, no MoC de *Tempo Contínuo* as soluções para ODEs são obtidas com o uso

de métodos numéricos, tais como Runge-Kutta e Euler [86], calculando-se um *ponto fixo* para todos os valores de sinais.

Um *ponto fixo* é definido como segue:

Dado qualquer função  $F : X \rightarrow X$  para qualquer conjunto  $X$ , se houver um  $x \in X$  tal que  $F(x) = x$ ,  $x$  é chamado de *ponto fixo*.

□

Em um modelo de atores determinístico, a função  $F$  especifica a semântica do modelo.

No MoC de tempo contínuo, o avanço do tempo é determinado pelo solucionador (i.e., *solver*) de ODE implementado em seu próprio diretor (*Continuous Director*). O *solucionador ODE* auxilia o diretor a escolher, a cada iteração, o rótulo de tempo que irá definir o tamanho do passo suficiente para a execução dos atores representados no modelo. Ao término da iteração, o tempo de simulação é incrementado.

Desse modo, o tempo avança mais suavemente a medida que os eventos vão sendo processados ao longo do tempo. Para uma maior precisão na execução de um modelo de tempo contínuo, certas propriedades como, *Linearidade* e *Invariância Temporal* devem ser aplicadas.

Definindo-se formalmente essas propriedades para um sistema  $S : X \rightarrow Y$ , onde  $X$  e  $Y$  são conjuntos de sinais, têm-se:

*Linearidade* – O sistema  $S$  é dito linear se

$$\forall x_1, x_2 \in X \text{ e } \forall a, b \in \mathbb{R}, S(ax_1 + bx_2) = aS(x_1) + bS(x_2).$$

□

*Invariância Temporal* – O sistema  $S$  é invariante no tempo se

$$\forall x \in X \text{ e } \forall \tau \in \mathbb{R}, S(D\tau(x)) = D\tau(S(x)),$$

onde  $\tau$  é o parâmetro de atraso e  $D\tau$  a função que calcula o atraso temporal. Sendo  $D\tau$  definido por

$$\forall x \in X \text{ e } \forall t \in \mathbb{R}, (D\tau(x))(t) = x(t - \tau).$$

□

No que diz respeito aos *ciclos de realimentação*, em que se estabelecem as conexões entre as funções representadas pelos atores, estes são evidenciados por ciclos direcionados de forma que uma saída de um ator é realimentada para interferir em uma entrada desse mesmo ator. Essa é uma importante característica dos sistemas de controle, que utilizam desse recurso para medir o erro<sup>6</sup> na saída do sistema, utilizando-o para fazer o ajuste de seu comportamento.

O MoC de *Tempo Contínuo* também interopera com outros modelos de computação temporizados, tal como o de *Evento Discreto*, apresentado a seguir.

### Evento Discreto

Diferentemente do que ocorre no MoC de *Tempo Contínuo*, no MoC de *Evento Discreto* um *evento* ocorre em um instante de tempo ao invés de ao longo do tempo, sendo esta a noção de *discreto*.

Por exemplo, um sinal de entrada para um ator pode ser definido por uma função do tipo  $u : \mathbb{R} \rightarrow \{ausente, presente\}$ . Isso significa que em qualquer tempo  $t \in \mathbb{R}$ , a entrada  $u(t)$  está *ausente* (i.e., não há evento) ou *presente* (i.e., há evento) naquele tempo. Um sinal desta forma é dito um *sinal puro*, pois não assume valor, mas fornece todas suas informações por estar ausente ou presente em qualquer instante de tempo. O sinal de saída pode ser modelado por uma função do tipo  $c : \mathbb{R} \rightarrow \{ausente\} \cup \mathbb{Z}$ . Com isso, quando todos os sinais de entrada estão ausentes  $c$  assume o valor *ausente*, no contrário, quando estão todas presentes a saída  $c$  assume um valor do tipo inteiro. Devido a possibilidade da saída assumir um valor específico, esse sinal torna-se *não puro*. Portanto, não há necessidade de um ator reagir quando as entradas estão ausentes, mas sim quando estiverem presentes, estabelecendo-se, assim, uma *dinâmica discreta*.

Formalmente, um *sinal discreto* pode ser definido como segue:

---

<sup>6</sup>O *erro* é calculado pela diferença entre o comportamento desejado (definido) e o comportamento atual medido na saída do sistema.

Seja  $e : \mathbb{R} \rightarrow \{\text{ausente}\} \cup X$ , onde  $X$  é qualquer conjunto de valores. Este sinal é um sinal discreto se, intuitivamente, está ausente na maioria das vezes, possibilitando contar, em ordem, as vezes em que está presente. Toda vez que este sinal está presente, caracteriza-se um *evento discreto*.

□

A dinâmica de um sistema discreto pode ser descrita como uma sequência de passos, os quais são denominados de *reações* e assume-se serem instantâneas. As reações de um sistema discreto são desencadeadas pelo ambiente em que opera. Dessa forma, em uma determinada reação observa-se os valores das entradas em um tempo  $t$  específico, e calcula-se os valores de saída para aquele mesmo instante de tempo  $t$ .

Em modelos de atores, a comunicação entre eles ocorre por meio de eventos. Todo evento possui um valor e um rótulo de tempo (*time stamp*), e os atores processam esses eventos seguindo uma ordem cronológica. Os eventos de saída produzidos por um ator são obrigatórios não serem temporalmente anteriores aos eventos de entrada consumidos por ele. Ou seja, atores no MoC de *Evento Discreto* são *causais*.

*Causalidade* é outra propriedade de sistemas, sendo caracterizada por sua saída depender somente das suas “entradas passadas e atuais”. Quando tais saídas dependem apenas das “entradas passadas”, o sistema é dito ser *estritamente causal*. Atores estritamente causais são úteis para criar ciclos de realimentação do sistema.

A definição formal da propriedade de *Causalidade* é dada por

Considere um sistema de tempo contínuo  $S : X \rightarrow Y$ , onde  $X = A^{\mathbb{R}}$  e  $Y = B^{\mathbb{R}}$  para alguns conjuntos  $A$  e  $B$ . Esse sistema é causal se  $\forall x_1, x_2 \in X$  e  $\tau \in \mathbb{R}$ ,

$$x_1|_{t \leq \tau} = x_2|_{t \leq \tau} \Rightarrow S(x_1)|_{t \leq \tau} = S(x_2)|_{t \leq \tau}$$

onde  $x|_{t \leq \tau}$  representa a *função de restrição temporal*, dando significado ao termo “entradas passadas e atuais”, devido incluir o tempo  $\tau$ .

□

Na execução desse modelo de computação, utiliza-se uma fila global de eventos. Quando um evento de saída é gerado por um ator, o evento é alocado na fila conforme

o seu rótulo de tempo. Em cada iteração, o diretor de evento discreto (*DE Director*) remove da fila o evento com menor rótulo de tempo, fazendo com que o ator responsável pelo evento produza o sinal de saída para o ator de destino.

Portanto, a dinâmica de evento discreto é adequada para modelar o comportamento de sistemas complexos, tais como hardware digital, sistemas financeiros e sistemas organizacionais humanos.

### Máquina de Estados

De natureza intrinsecamente sequencial, em uma *Máquina de Estados* os componentes, em vez de atores, são *estados*. Essa mudança altera o conceito de relacionamento entre os componentes, deixando de ser um caminho de comunicação entre atores, e passando a ser uma *transição* entre os estados.

O *estado* de um sistema é a sua condição em um instante de tempo específico. Em geral, este afeta como o sistema reage as entradas, sendo, portanto, um resumo do passado. Por conseguinte, uma máquina de estados é um modelo de um sistema com *dinâmica discreta*, em que cada reação mapeia avaliações das entradas para avaliações das saídas, podendo o mapeamento depender de seu *estado atual*. Quando tais modelos discretos possuem uma quantidade finita de estados, ou seja, o *espaço de estados*<sup>7</sup> é finito, esses são denominados de *Máquinas de Estados Finito* (do inglês, *Finite-State Machines* – FSM).

Se o número de estados é razoavelmente pequeno, então a FSM pode ser elaborada por meio do uso de uma notação gráfica, em que cada *estado* é representado por uma bolha/círculo, e as *transições* entre os estados por uma seta curva e direcionada. Por exemplo, para o espaço de estados  $SS = \{s_1, s_2, s_3\}$ , o modelo da FSM pode ser organizado conforme ilustrado na Figura 2.7.

O *estado inicial* indica o início de cada sequência de reações, sendo representado por uma seta apontada para o mesmo e que não possui estado de origem, nesse caso para  $s_1$ . Uma *transição* pode começar e terminar em um mesmo estado, como em  $s_3$ , caracterizando um laço (*loop*).

---

<sup>7</sup>O *espaço de estados* é definido como sendo o conjunto de estados alcançáveis pelo sistema a partir de seu estado inicial. É importante destacar que o espaço de estados de um sistema pode ser infinito.

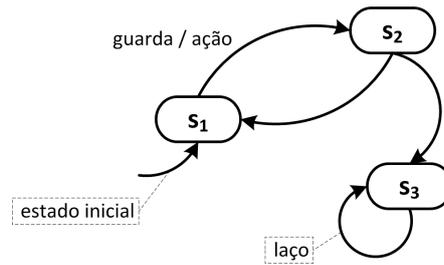


Figura 2.7: Exemplo de uma FSM em notação gráfica. Adaptado de [49].

As *transições* governam a dinâmica discreta da FSM por meio de rótulos do tipo *guarda / ação*. Uma *guarda* determina se uma transição pode ser tomada em uma dada reação, enquanto a *ação* especifica que saídas são produzidas em cada reação.

Assim, uma guarda funciona como um *predicado* (i.e., uma expressão booleana<sup>8</sup> valorada), avaliando a veracidade de sua expressão. Quando uma guarda é avaliada como *verdadeira*, dizemos que a transição está *habilitada*, ou seja, que a transição deve ocorrer. Isso fará com que o estado do sistema seja alterado, que antes se encontrava na origem da transição, passando ao estado de destino dessa transição. Desse modo, a ação da transição, caso definida, será executada, atribuindo-se valores para suas portas de saída. Qualquer porta de saída não mencionada na transição tomada, implicitamente assume o valor *ausente*, evidenciando-se o comportamento típico de processar uma saída em um modelo de evento discreto. No caso de uma FSM, os valores *presente* e *ausente* são sinônimos de “verdadeiro” e “falso”, respectivamente.

A notação para atribuição de valor a uma porta de saída é dada pelo padrão *nome := valor*, enquanto que para um predicado *nome = valor*. Se houver apenas uma saída no modelo, não é necessário mencionar o nome da porta na atribuição. Se nenhuma ação for determinada, então todas as saídas são implicitamente *ausentes*.

Uma FSM pode ser acionada por eventos (*event triggered*), em consequência da ocorrência destes, ou pelo tempo (*time triggered*), em intervalos regulares. Independente da forma de reação de uma FSM, isso não altera sua definição. Se nenhuma guarda em qualquer transição de saída do estado atual é avaliada como verdadeira, então a máquina permanecerá no mesmo estado.

Para explicar os conceitos apresentados até o momento, considere o modelo de um

<sup>8</sup>As expressões estabelecidas às guardas podem conter operadores lógicos, tais como negação ( $\neg$ ), conjunção ( $\wedge$ ) e disjunção ( $\vee$ ).

termostato ilustrado na Figura 2.8, cujo papel funcional é regular a temperatura de um sistema de ar-condicionado e mantê-la constante, conforme a necessidade do usuário.

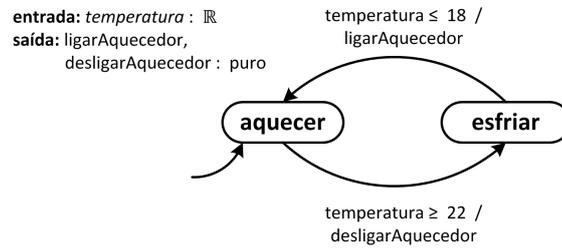


Figura 2.8: Exemplo de uma FSM representando o comportamento de um termostato com reações desencadeadas por eventos. Adaptado de [49].

O conjunto de estados para essa FSM é  $SS = \{aquecer, esfriar\}$ , sendo também definidos um único sinal de entrada ( $temperatura : \mathbb{R}$ ) e dois sinais de saída ( $ligarAquecedor, desligarAquecedor : puro$ , isto é, ou o sinal está *presente* ou está *ausente*). Suponha que o temperatura desejada pelo usuário seja  $20^{\circ}C$  (graus Celsius), ou seja, o valor de referência do sistema. Se o aquecedor estiver ligado ( $estado(t) = aquecer$ ) o termostato permitirá, nesse caso, que a temperatura se eleve até ultrapassar o valor de  $22^{\circ}C$ . Caso contrário, se o aquecedor estiver desligado ( $estado(t) = esfriar$ ) a temperatura cairá até ficar abaixo de  $18^{\circ}C$ . Portanto, as saídas do sistema somente serão observadas quando uma de suas transições disparar, em virtude da mudança no estado do aquecedor avaliado pelas condições de guarda, e a ação de saída de tal transição for executada. Vale ressaltar que, desse modo o sistema de ar-condicionado irá se comportar inadequadamente, pois não conseguirá manter a temperatura constante devido a tardia percepção no valor de sua entrada.

Em notação matemática, uma máquina de estados finito *determinística* é uma 5-tupla

$$MEF = \langle S, IN, OUT, f_t, st_{init} \rangle$$

onde:

- ▣  $S = \{s_1, s_2, \dots, s_{|S|}\}$  é um conjunto finito de estados;
- ▣  $IN = \{i_1, i_2, \dots, i_{|IN|}\}$  é um conjunto de valores/avaliações de entrada;

- ▣  $OUT = \{o_1, o_2, \dots, o_{|Out|}\}$  é um conjunto de valores/avaliações de saída;
- ▣  $f_t : S \times IN \rightarrow S \times OUT$  é uma **função de transição**, que mapeia um estado e um valor de entrada (e.g.,  $\langle s_1, i_2 \rangle$ ) para um *próximo* estado e um valor de saída  $\langle s_4, o_3 \rangle$ ;
- ▣  $s_{init}$  é o estado inicial da MEF, sendo  $s_{init} \in S$ .

□

A dinâmica de execução de uma FSM é simples, reagindo em sequência, a cada reação ela pode passar de um *estado atual* para um *estado seguinte*, que será o *estado atual* da próxima reação.

Por exemplo, sejam as seguintes funções  $s : \mathbb{N} \rightarrow S$ ,  $x : \mathbb{N} \rightarrow IN$ ,  $y : \mathbb{N} \rightarrow OUT$ , que, respectivamente, fornecem o estado de uma FSM em uma reação  $n \in \mathbb{N}$ , e as avaliações de entrada e saída em cada reação, então a equação que fornece a dinâmica dessa FSM é dada por

$$(s(n+1), y(n)) = f_t(s(n), x(n)).$$

□

Agora, considere a FSM referente ao termostato, mostrada na Figura 2.8. Sua representação formal é dada por:

- ▣  $S = \{aquecer, esfriar\}$ ;
- ▣  $IN = (temperatura \rightarrow \mathbb{R})$ ;
- ▣  $OUT = (\{ligarAquecedor, desligarAquecedor\} \rightarrow \{presente, ausente\})$ ;
- ▣  $s_{init} = aquecer$ ;
- ▣

$$f_t(s, i) = \begin{cases} (aquecer, ligarAquecedor), & \text{se } s = esfriar \wedge i(temperatura) \leq 18, \\ (esfriar, desligarAquecedor), & \text{se } s = aquecer \wedge i(temperatura) \geq 22, \\ (s, ausente), & \text{caso contrário.} \end{cases}$$

Em termos de definição formal, o que diferencia uma *FSM Determinística* de uma *FSM Não-Determinística* é que a *função de transição* passa a ser uma *relação de transição*. Assim, uma máquina de estados finito *não-determinística* é uma *5-tupla*

$$MEF = \langle S, IN, OUT, r_t, st_{init} \rangle$$

onde:

▣  $r_t : S \times IN \rightarrow 2^{S \times OUT}$  é uma **relação de transição**, que mapeia um estado e um valor de entrada (e.g.,  $\langle s_1, i_2 \rangle$ ) para um *conjunto de possíveis pares*  $\langle \text{próximo estado}, \text{valordesaída} \rangle$  (e.g.,  $\{ \langle s_4, o_3 \rangle, \langle s_5, o_6 \rangle \}$ ).

□

Quando o número de estados de uma FSM torna-se grande, fica complicado representar o seu modelo. Para resolver este problema criou-se o conceito de *Máquinas de Estados Estendida* (do inglês, *Extended State Machine* – ESM). Em uma ESM são adicionadas variáveis que podem ser lidas e escritas como parte da tomada de uma transição entre estados.

A notação de uma ESM difere da FSM convencional nos três seguintes aspectos, conforme ilustrado na Figura 2.9:

1. As declarações de variáveis são mostradas explicitamente para tornar mais fácil determinar se um identificador em uma guarda ou ação refere-se a uma variável, ou a uma entrada ou saída;
2. Após a inicialização, as variáveis que foram declaradas podem ser inicializadas. O valor inicial será mostrado na transição que indica o estado inicial;
3. Anotações de transição agora tem a forma:

*guarda / ação*  
*set ação.*

A *guarda* e a *ação* de saída possuem as mesmas definições para uma FSM, exceto que podem agora fazer referência à variáveis. No entanto, o conceito de *set ação(ões)*

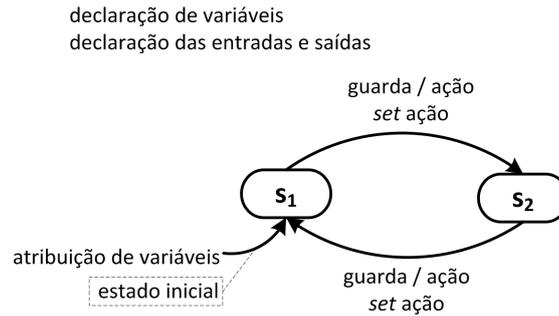


Figura 2.9: Exemplo de uma ESM em notação gráfica. Adaptado de [49].

foi acrescentado. Uma *set ação* especifica as atribuições para um conjunto de variáveis quando uma transição é tomada. Essas atribuições são feitas *após* a guarda ter sido avaliada e as saídas tenham sido produzidas. Assim, se a guarda ou as ações de saída referenciam uma variável, o valor da variável é aquele *antes* da atribuição especificada na *set ação*. Se houver mais do que uma atribuição, então essas atribuições serão executadas em ordem sequencial.

O modelo de um sistema de semáforo de trânsito, apresentado na Figura 2.10, esclarece esses novos conceitos oriundos de uma máquina de estados estendida. Nesse modelo, as reações ocorrem em intervalos de um segundo. Com início definido no estado *vermelho*, a variável *contador* contabiliza 60 segundos, sendo então habilitada e tomada a transição para o estado *verde*. O sistema irá permanecer no estado *verde* até que o sinal de entrada referente ao *pedestre* esteja *presente*, por exemplo, ao pressionar o botão para requisitar a passagem na faixa de pedestre. Se isso acontecer, e o tempo decorrido for inferior a 60 segundos, a máquina transita para o estado auxiliar *esperando*, onde permanecerá pelo tempo de 60 segundos, quando irá mudar para o estado *amarelo*. Por outro lado, essa mudança para o estado *amarelo* será imediata. Uma vez nesse estado, permanecerá por um tempo de 5 segundos antes de retornar ao estado *vermelho*. Quanto as saídas das transições nesse modelo, definiu-se os sinais *senalG*, *senalR* e *senalY*, respectivamente, para acender as luzes verde, vermelha e amarela do semáforo.

Máquinas de estados estendida podem fornecer uma maneira conveniente para acompanhar a passagem do tempo em um sistema. O seu estado inclui não somente a informação sobre qual estado discreto a máquina se encontra, mas também quais os

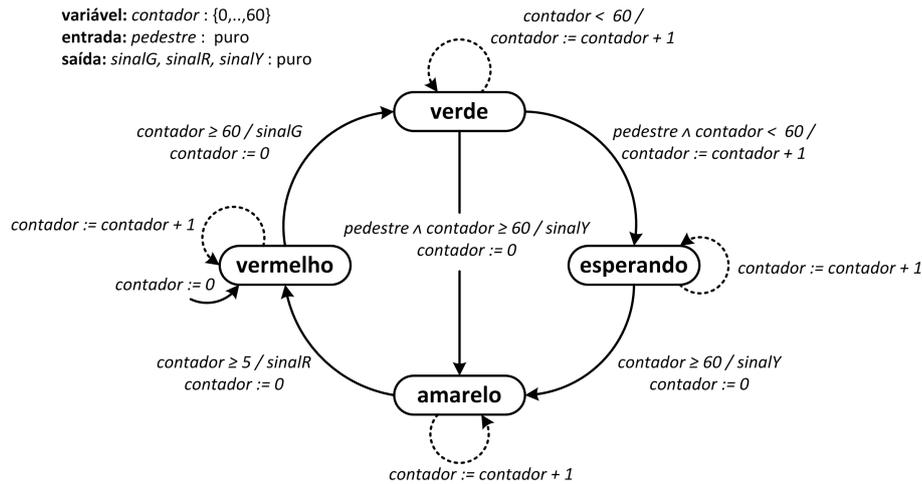


Figura 2.10: Exemplo de uma ESM representando o comportamento de um semáforo temporizado com reações em intervalos regulares de tempo. Adaptado de [49].

valores de todas suas variáveis. O número de estados possíveis pode, portanto, ser bastante grande, ou mesmo infinito. Ao mesmo tempo, nem todo estado de uma máquina de estado é alcançável.

Especificamente falando de uma das principais vantagens de utilizar máquinas de estados é que elas podem definir todos os comportamentos possíveis de um sistema. Como consequência, os seus modelos são passíveis de verificação formal, que determina se os comportamentos especificados são, de fato, os comportamentos desejáveis. Outros inúmeros detalhes sobre o MoC de Máquina de Estados podem ser encontrados em Lee *et al.* [49].

Por fim, em se tratando de modelos de atores, uma *Máquina de Estados* pode ser utilizada para definir o comportamento de um ator, dentro de outros modelos de computação, entre estes estão incluídos o MoC de *Tempo Contínuo* e de *Evento Discreto*.

### 2.4.5 Modelos Heterogêneos de Atores

Sistemas computacionais que se relacionam com processos físicos, normalmente, são intrinsecamente complexos, pois necessitam de diferentes técnicas de modelagem para representar o seu comportamento [47]. Por integrar as dinâmicas de *tempo contínuo* com a de *evento discreto*, tem-se um *Sistema Híbrido* (do inglês, *Hybrid Systems* –

HS) [14].

Nesse sentido, as necessidades de um Sistema Físico-Cibernético (SFC) [46] se assemelham com as de um HS, devido um SFC integrar elementos do mundo físico (e.g., processos mecânicos, químicos, físicos e biológicos), que possuem *dinâmica contínua*, com elementos do mundo cibernético (e.g., entidades computacionais e suas redes), caracterizados pela *dinâmica discreta*.

Dessa forma, as duas técnicas de modelagem comumente utilizadas para representar cada uma dessas dinâmicas embutidas em um HS, e que também podem ser utilizadas para a modelagem de SFC, são as equações diferenciais ordinárias (ODEs), para a dinâmica contínua, e as máquinas de estados finito (FSM), para a dinâmica discreta. Na modelagem de sistemas híbridos utilizando o paradigma AOD, esses modelos são denominados de *Modelos Modais* [49].

Em um *modelo modal* associa-se a cada estado de uma FSM um comportamento dinâmico, denominado de *modo de operação*. Quando o modelo do sistema encontra-se em um determinado *modo*, a dinâmica do modelo é dada pelo *refinamento de estado* referente àquele modo. Um *refinamento de estado* produz o comportamento dinâmico da saída em função da entrada como uma sequência de passos, que são as *reações*.

No *refinamento de estado*, além das saídas, podem ser produzidos valores para um conjunto adicional de variáveis contínuas de estado, evidenciando-se, assim, o conceito de máquina de estados estendida (ESM). Logo, sem assumir que as entradas e as saídas de um sistema necessariamente estão *ausentes* durante o tempo entre as *reações*, define-se a ocorrência de uma mudança no estado do sistema quando uma *transição de modo* é tomada. Isso ocorre quando a *guarda* que especifica a combinação de entradas e estados contínuos é satisfeita. A ação associada com uma *transição de modo*, por sua vez, define o estado contínuo no modo de destino. Resumidamente, *transições de modo* ocorrem na dinâmica discreta, enquanto *refinamentos de estados* em tempo contínuo.

Na Figura 2.11 é apresentada a notação que define a estrutura geral de um *Modelo Modal*. Nesse caso específico, a ESM possui dois estados ou modos de operação ( $s_1, s_2$ ), sendo cada modo rotulado por “*sistema baseado em tempo*”, que especifica o modelo do refinamento de estado.

Lembrando que, em um *modelo modal*, dois *modos de operação* podem ter o mesmo

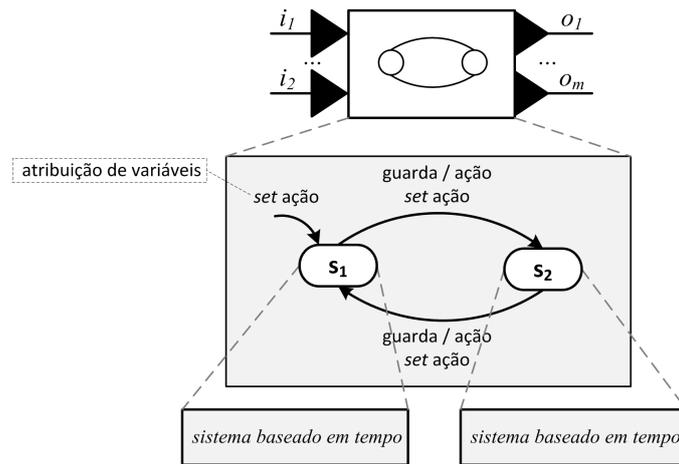


Figura 2.11: Notação gráfica para um *Modelo Modal*. Adaptado de [49].

*refinamento de estado*, sendo as saídas do sistemas originadas pelos mecanismos funcionais internamente implementados pelo modelo que representa tal refinamento.

Portanto, os modelos modais fornecem uma ponte entre modelos baseado no tempo e modelos de máquina de estados. A combinação desses modelos de computação fornece uma estrutura rica para descrever sistemas do mundo real. Em trabalho recente, Lee [48] apresenta um estudo sobre as semânticas utilizadas para a modelagem de fenômenos físicos. O autor demonstra a aplicação desses modelos de computação em diferentes contextos físicos, e também discute as dificuldades e os problemas associados à construção e simulação de modelos representativos desses sistemas.

Segundo Lee [47], é complicado lidar com a diversidade de modelos criados para representar as diversas e heterogêneas partes que irão interoperar para fornecer o comportamento de um SFC como um todo. A maior dificuldade é definir a semântica conjunta desses modelos, por causa das concorrentes interações entre os seus elementos funcionais. Surge daí a importância de se ter ferramentas e arcabouços de modelagem flexíveis, que proporcionem criar semânticas formais bem definidas, evitando dúbias interpretações e melhorando a compreensão e a análise dos modelos de SFC.

### 2.4.6 Ferramentas de Apoio à Modelagem

Os modelos de computação (MoCs) foram concebidos para prover uma solução para o *problema de composição de modelos homogêneos* [86]. Várias linguagens de modelagem

possibilitam tornar essa ideia uma realidade, adotando-se metodologias de *Projeto Baseado em Modelo* (do inglês, *Model-Based Design*). Exemplos dessas linguagens incluem UML<sup>9</sup> e SysML<sup>10</sup>, Simulink<sup>11</sup> e Stateflow<sup>12</sup>, AADL<sup>13</sup>, Modelica<sup>14</sup>, e LabVIEW<sup>15</sup>.

Segundo Tripakis *et al.* [86], esse tipo de linguagem eleva o nível de abstração no projeto de um sistema por meio de mecanismos que capturam concorrência, interação e comportamento temporal, aspectos esses considerados essenciais em sistemas modernos. Além disso, disponibilizam ferramentas para verificação e geração de código a partir dos modelos, facilitando o processo de implementação de sistemas.

Entretanto, essas linguagens oferecem pouco ou nenhum suporte à modelagem heterogênea [47], e quando tal suporte é oferecido, utiliza-se de uma *semântica fraca* que compromete o entendimento e a análise dos modelos [72]. Esse é o caso das linguagens que fazem uso da notação *diagrama de blocos*, tais como a UML e Simulink, não dando significado preciso às linhas que interligam os blocos de um modelo.

Embora não haja, hoje em dia, uma solução universalmente aceita para a modelagem heterogênea [86], o projeto *Ptolemy*<sup>16</sup> tem sido pioneiro na tentativa de prover uma solução para o *problema de composição de modelos heterogêneos*.

Tendo como objetivo prover uma *semântica forte* [72], o projeto *Ptolemy* estuda a modelagem, a simulação, e o projeto de sistemas embarcados concorrentes e de tempo-real. O princípio chave para alcançar tal objetivo é utilizar MoCs bem definidos que governem a interação entre componentes.

Como principais resultados pretendidos com o *Ptolemy*, busca-se desenvolver técnicas formais para suporte à modelagem heterogênea, e disponibilizar um arcabouço de software que aplique essas técnicas em modelos heterogêneos que seguem o paradigma AOD [87]. O arcabouço, que é escrito em Java e de código aberto (i.e., *open source*), foi nomeado de *Ptolemy II*.

Conforme definido no início da Seção 2.4, os componentes no paradigma AOD são

---

<sup>9</sup><http://www.uml.org>

<sup>10</sup><http://www.sysml.org>

<sup>11</sup><http://www.mathworks.com/products/simulink/>

<sup>12</sup><http://www.mathworks.com/products/stateflow/>

<sup>13</sup><http://www.aadl.info>

<sup>14</sup><http://www.modelica.org>

<sup>15</sup><http://www.ni.com/labview/>

<sup>16</sup><http://ptolemy.eecs.berkeley.edu/>

conhecidos como *atores*, que podem ser vistos como processos executando concorrentemente, e que utilizam-se de algum mecanismo para a comunicação entre eles. Sendo tal semântica de execução e comunicação definida por um MoC.

Na terminologia do *Ptolemy II*, um MoC é implementado por um bloco chamado de *Diretor*, e referido como um *Domínio*, por se tratar de um conjunto de regras que governam os modelos de atores. De acordo com Lee *et al.* [72], o termo “domínio” está relacionado com a noção em astrofísica, de que há regiões do universo com diferentes conjuntos de “*leis da física*”.

Apesar de estar ainda em desenvolvimento, o *Ptolemy II* já possui diversos MoCs implementados, com seus respectivos *diretores* disponíveis para experimentação de modelos heterogêneos de atores. Além dos MoCs já apresentados nessa seção, estão incluídos nessa lista os domínios *synchronous data flow* (SDF) e suas variantes, *synchronous-reactive* (SR) e *process networks* (PN).

## 2.5 Considerações Finais

Neste capítulo foi apresentada uma visão geral das teorias, conceitos e práticas fundamentais para o entendimento e a elaboração da solução a ser descrita nesta proposta de tese.

Inicialmente, foram discutidos os principais conceitos inerentes aos *Sistemas Físico-Cibernéticos* (SFC). A razão para isso é a natureza generalista desses sistemas em relação aos denominados *Sistemas Médicos Físico-Cibernéticos* (SMFC), que fazem uso dos fundamentos, das técnicas e das ferramentas de SFC no desenvolvimento de sistemas de controle e apoio à decisão para o domínio da saúde.

Em seguida, foi apresentado o conceito de *Ambiente Clínico Integrado*, cuja noção baseia-se na integração de dispositivos médicos e outros equipamentos com a finalidade de desenvolver sistemas médicos centrados no paciente. Assim, a partir de um modelo conceitual de arquitetura, é possível definir as mais variadas configurações para se criar cenários clínicos, usando diferentes instâncias de seus elementos.

Neste trabalho, a aplicação prática desses conceitos dar-se por meio de uma arquitetura baseada em modelos, sendo este o motivo pelo qual foram descritos e discutidos os

conceitos, as abordagens e as ferramentas adotadas na engenharia baseada em modelos. Adicionalmente, discutiu-se brevemente sobre os resultados de uma revisão sistemática da literatura acerca dos recentes trabalhos em SMFC que buscam aumentar a confiança de seus projetos por meio da execução de atividades relacionadas à Verificação e Validação (V&V) de software.

Por fim, como os modelos que compõem a solução desta proposta de tese utilizam componentes conhecidos como *atores* para formalizar suas abstrações, buscou-se esclarecer os principais conceitos definidos para o paradigma de *Projeto Orientado a Atores*. Dessa forma, foram detalhadas as noções e características de atores, modelos de atores e hierarquia desses modelos. Neste mesmo contexto, outro importante conceito descrito foi o de *Modelo de Computação*, componente responsável por especificar a semântica de execução do modelo geral do sistema e comunicação entre os atores concorrentes que compõem esse modelo.

Visando justificar a escolha da ferramenta de modelagem utilizada para conceber os modelos disponibilizados para a solução proposta, discutiu-se também sobre as particularidades das principais ferramentas que agregam os diversos conceitos apresentados neste capítulo.

# Capítulo 3

## Biblioteca de Modelos:

## Construindo Modelos Reutilizáveis

A biblioteca de modelos é um conjunto de modelos reutilizáveis de pacientes e dispositivos reutilizáveis visando aumentar o potencial de reuso e produtividade durante a validação de SMFC. Diretrizes foram definidas para permitir aos desenvolvedores de SMFC adaptarem os modelos existentes ou construírem seus próprios modelos de acordo com os requisitos do sistema, enriquecendo a biblioteca ao longo do tempo.

Neste trabalho, o processo de modelagem é baseado no paradigma de Projeto Orientado a Atores (do inglês, *Actor-Oriented Design* – AOD) [2], uma metodologia de projeto baseada em componentes chamados *atores*. Esta metodologia representa um modelo formal de concorrência, em que um ator é um agente computacional que possui uma *thread* de controle independente e se comunica por meio da troca de mensagens assíncronas. Para construir os modelos, utilizou-se a ferramenta de modelagem *Ptolemy II* [87], que é um arcabouço de software extensível baseado em AOD com suporte à experimentação. Sua ênfase está na composição de componentes concorrentes, usando modelos de computação bem definidos que governam as interações entre esses componentes.

Neste capítulo, detalha-se como construir os modelos de pacientes e dispositivos médicos para compor a biblioteca de modelos. Ao aplicar as diretrizes, o desenvolvedor pode construir modelos para dispositivos e pacientes a serem utilizados em diferentes contextos de SMFC. As mesmas diretrizes foram utilizadas para construir os modelos

existentes, que são também descritas neste capítulo.

### 3.1 Construção dos Modelos de Pacientes

Modelos de Pacientes são modelos formais construídos baseados em diretrizes médicas e bases de dados clínicas. Um modelo de paciente consiste de um conjunto de atributos/parâmetros (e.g., demográficos) que definem o perfil do paciente, e um conjunto de componentes para gerar dados sintéticos relacionados aos sinais vitais e parâmetros fisiológicos. Adicionalmente, um conjunto de modelos matemáticos é incorporado para representar as dinâmicas chave de processos biológicos, patológicos e farmacológicos no corpo de um paciente em tratamento por meio de medicamentos, como resposta aos diferentes estímulos. Em geral, esses modelos matemáticos são definidos pela *Farmacologia*.

Segundo Katzung [39], *Farmacologia* é a ciência que estuda substâncias que interagem com sistemas vivos por meio de processos químicos, ativando ou inibindo processos normais do corpo humano. As interações entre um medicamento e o corpo humano estão divididas em duas classes: (i) as ações do medicamento no corpo são denominadas de *Processos Farmacodinâmicos (PD)*; e (ii) as ações do corpo em resposta ao medicamento são denominadas de *Processos Farmacocinéticos (PK)*. Em outras palavras, os processos PK controlam a absorção, a distribuição e a eliminação de medicamentos, enquanto os processos PD estão preocupados com as ações químicas no organismo do paciente.

Um modelo de paciente pode representar o comportamento parcial ou total do corpo humano, incluindo órgãos específicos ou sistemas orgânicos completos, respectivamente. Tais representações podem incorporar aspectos simples, tais como um conjunto de sinais vitais e parâmetros fisiológicos, e aspectos complexos, tais como a resposta à administração de um medicamento específico. Modelar o comportamento de um paciente sempre levanta uma questão sobre como lidar com a sua natureza não-determinística. De acordo com Lee [45], SMFC naturalmente combinam modelos de tal forma que o determinismo não é preservado. Contudo, ele demonstra que modelos de SMFC determinísticos têm viabilidade prática. O grau em que o modelo imita a coisa a ser

modelada (ou vice-versa) define o seu valor, embora seja sempre aproximado.

Li *et al.* [54] destaca que não existem bons modelos para descrever o comportamento complexo de pacientes, geralmente descritos por equações diferenciais ordinárias (do inglês, *Ordinary Differential Equations* – ODE), e que esses modelos podem não se ajustar às ferramentas de verificação de modelos existentes para sistemas híbridos. Por outro lado, em um curto horizonte de tempo, técnicas tais como regressão linear e linearização podem descrever comportamentos básicos e complexos do paciente, respectivamente. Neste trabalho, este problema é abordado ao delinear os processos de forma compreensiva para descrever os comportamentos básicos de pacientes (e.g., sinais vitais) usando modelos de regressão multivariados (e.g., modelos lineares generalizados) e comportamentos complexos (e.g., cinéticas da insulina e da glicose) usando abordagens da *Teoria de Controle* (e.g., modelos de espaço de estados).

O procedimento para construir um modelo de paciente consiste de quatro etapas detalhadas nas subseções seguintes. A explicação para cada etapa é dada usando um exemplo de um Modelo de Paciente Diabético, que representa o comportamento de pacientes com diabetes mellitus.

### 3.1.1 Etapa 1 – Escolher uma Base de Dados Clínica

Para esta etapa inicial, escolheu-se uma base de dados de pacientes com dados pessoais e registros de sinais vitais e parâmetros fisiológicos. A base de dados clínica deve ter pelo menos 30 registros de pacientes para aplicar análise de regressão múltipla, que é a quantidade mínima requerida para alcançar um determinado nível de acurácia e confiança para os modelos de regressão. O tamanho da amostra é importante porque ela pode afetar o poder estatístico do modelo de regressão e a generalização do resultado [27]. Se a base de dados contém um grande volume de dados, é necessário caracterizar uma população de interesse. Este processo consiste em aplicar um conjunto de regras aos dados reais dos pacientes, que foram obtidos a partir da base de dados clínica. O objetivo é reduzir o grande volume de registros contidos na base de dados selecionada.

Por exemplo, para construir o Modelo de Paciente Diabético a base de dados selecionada foi a *MIMIC II Clinical Database v2.6* [71], cujo acesso foi autorizado pela

*PhysioNet.org*. Esta base de dados contém dados clínicos de pacientes em Unidades de Terapia Intensiva (UTI). Devido à grande quantidade de registros armazenados nesta base de dados clínica, um conjunto de regras foi aplicado sobre a mesma para caracterizar uma população de interesse.

A população obtida contém 38.141 observações referentes a 2.245 pacientes, sendo, aproximadamente, 37,4% do gênero feminino e 62,6% masculino. A partir dessa população, uma amostra foi obtida contendo uma observação de cada paciente. Tal observação é referente ao momento em que cada paciente foi admitido na UTI. Um resumo do processo para caracterizar a população de interesse e o procedimento de amostragem é apresentado na Figura 3.1, cujo tamanho da amostra representa apenas aproximadamente cerca de 0,001% da quantidade total de registros da base de dados clínica.

O próximo passo é selecionar o conjunto de variáveis preditoras para realizar a análise estatística. Uma vez que os sinais vitais e parâmetros fisiológicos fornecidos pelo modelo de paciente são baseados em limiares, estabeleceram-se os parâmetros que definem esses limiares em conformidade com *Diretrizes Clínicas*. Essas diretrizes clínicas também foram utilizadas para ajudar na identificação das potenciais variáveis preditoras para os modelos de regressão de cada sinal vital considerado no modelo de paciente.

Por exemplo, para selecionar as variáveis preditoras para a análise estatística para o Modelo de Paciente Diabético, utilizou-se as diretrizes clínicas descritas em [1; 12; 15; 28; 60; 65]. No total, oito potenciais variáveis preditoras foram selecionadas: *gênero*, *peso*, e *altura* como variáveis demográficas; *frequência cardíaca (hr)*, *frequência respiratória (rr)*, *pressão arterial sistólica (sbp)*, e *temperatura corporal (pt)*, como sinais vitais; e *nível de glicose sanguínea (gl)*, como parâmetro fisiológico. Na Tabela 3.1 são apresentadas as estatísticas descritivas acerca do conjunto de dados extraídos a partir desse processo.

### 3.1.2 Etapa 2 – Obter um Modelo Estatístico

Com a amostra extraída e as variáveis selecionadas para a análise, o objetivo desta etapa é obter um modelo de regressão para cada variável vital de interesse. Estatísticas

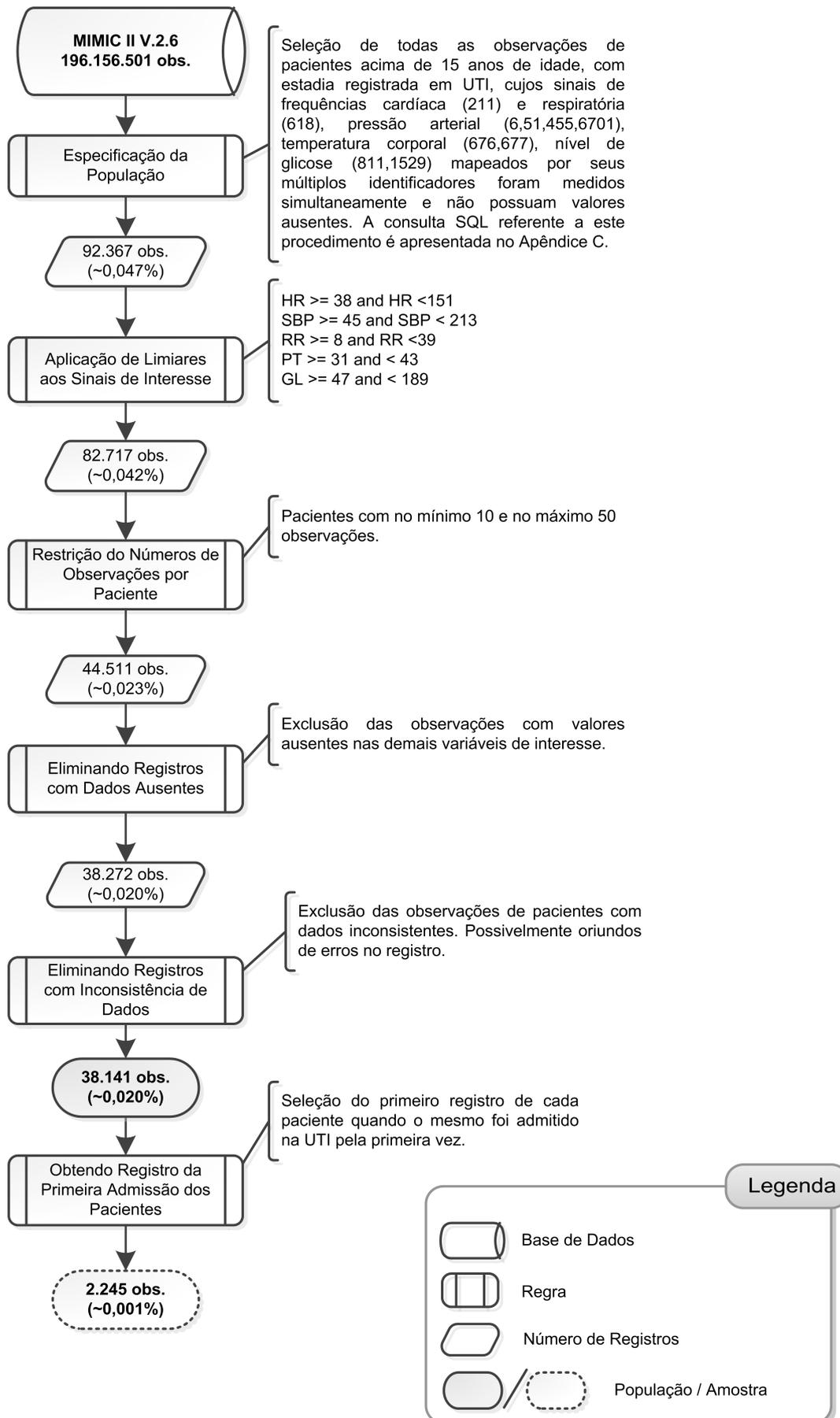


Figura 3.1: Processo para Caracterizar a População de Interesse.

Variável	Média	Std. Dev.	CV	Mínimo	Máximo
<i>hr_value</i>	87,315	14,263	0,163	40,00	150,00
<i>sbp_value</i>	116,518	20,049	0,172	48,00	212,00
<i>rr_value</i>	19,216	5,628	0,293	8,00	38,00
<i>pt_value</i>	37,251	0,739	0,020	31,70	41,44
<i>gl_value</i>	118,761	27,802	0,234	47,00	188,00
<i>weight</i>	84,300	20,730	0,246	33,00	200,00
<i>height</i>	169,804	10,379	0,061	124,50	231,10

Std. Dev.: Desvio Padrão; CV: Coeficiente de Variação.

Tabela 3.1: Medidas de Resumo da População de Interesse.

descritivas devem ser analisadas para escolher a técnica de regressão mais adequada. Neste sentido, comparam-se as medidas de resumo da amostra extraída com as da população de interesse para verificar a representatividade das variáveis em estudo. Além disso, analisa-se a distribuição dos dados usando *boxplots* e *histogramas*, bem como também as correlações lineares entre as variáveis selecionadas. Assim, é possível identificar o perfil dos dados contidos na amostra. Baseado no perfil dos dados, decidiu-se por utilizar Modelos de Regressão Multivariados (MRM) para gerar os sinais vitais. Esses modelos de regressão estatísticos serão integrados para prover a dinâmica básica do modelo de paciente.

O fluxo de trabalho para ajustar o MRM para cada sinal vital é ilustrado na Figura 3.2. Dada uma amostra extraída a partir da população de interesse, um MRM inicial é obtido. Em seguida, analisam-se as estatísticas descritivas desse modelo de regressão, incluindo o conjunto de variáveis preditoras. Se as estatísticas do modelo não fornecem evidências suficientes acerca da qualidade do modelo (i.e., as predições do modelo são INACEITÁVEIS), é necessário modificar o modelo de regressão inicial. Por exemplo, pode ser necessário incluir outras variáveis mais significantes ou remover aquelas menos significantes. Um exemplo de métrica de qualidade usado para avaliar o ajuste de um modelo de regressão é o coeficiente de determinação  $R^2$ . Por outro lado, se o modelo de regressão possui bons indicadores de qualidade (i.e., as predições do modelo de regressão são ACEITÁVEIS), gera-se um conjunto de gráficos de diagnósticos que permitem uma análise mais detalhada do modelo. Exemplos de gráficos de diagnósticos são os gráficos de envelope (*Normal Q-Q Plot*), de pontos de alavanca, de pontos influentes e o gráfico de resíduos *versus* os valores ajustados.

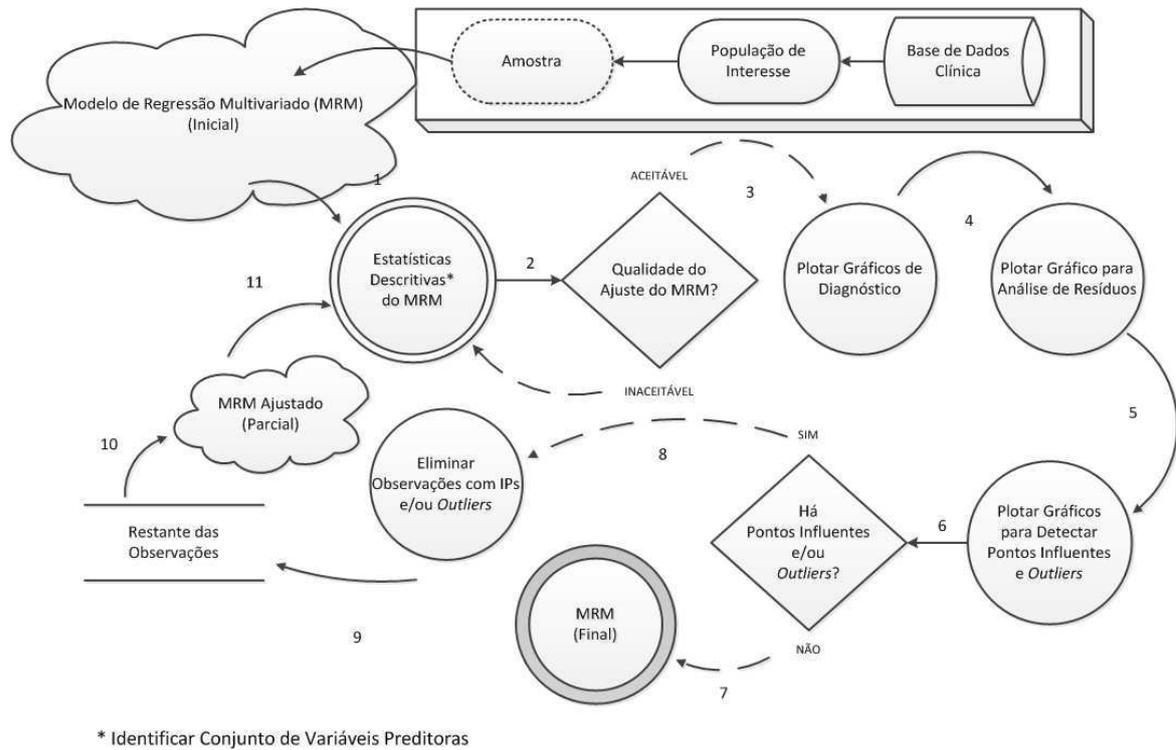


Figura 3.2: O Processo de Ajuste do Modelo de Regressão Multivariado para cada Sinal Vital de Interesse.

Gráficos de Envelope são úteis para verificar o ajuste dos modelos de regressão. Pontos de Alavanca podem interferir nos valores ajustados próximos a eles e na estimativa dos coeficientes de regressão. Pontos Influentes podem interferir nos valores estimados dos parâmetros do modelo. Por fim, o gráfico de resíduos *versus* valores ajustados é útil para avaliar suposições acerca do modelo de regressão (e.g., quaisquer tendências visíveis que mostrariam uma dependência dos erros nas variáveis predictoras) [33]. Após analisar os gráficos de diagnósticos de cada modelo de regressão, o desenvolvedor deve decidir qual desses modelos de regressão deve ser utilizado para gerar os sinais vitais que eles representam. Isto significa que os dados sintéticos gerados pelo modelo de paciente devem ser compatíveis com a amostra utilizada na análise estatística. O processo iterativo de ajuste ilustrado na Figura 3.2 deve ser aplicado a todos os modelos de regressão a serem incluídos no modelo de paciente.

Para o exemplo de Modelo de Paciente Diabético, uma classe de MRM, então chamados de Modelos Lineares Generalizados (MLG), foi utilizada para obter os modelos de regressão. Nos MLG, assume-se que a variável de resposta segue a família de dis-

tribuição exponencial e os valores preditos são calculados a partir de uma função de ligação [59]. Portanto, ajusta-se um modelo de regressão para cada sinal vital considerado (i.e.,  $hr$ ,  $rr$ ,  $sbp$ , e  $pt$ ). Por exemplo, considerando a variável de frequência respiratória ( $rr$ ), utilizou-se o *Modelo Normal Inversa* na forma de (3.1) com função de ligação canônica definida em (3.2) para prever o sinal de frequência respiratória ( $MLG\_RR$ ).

$$\hat{\mu} = \hat{\eta}^{-\frac{1}{2}} \quad (3.1)$$

Neste caso,  $\hat{\mu}$  é a média da frequência respiratória, e

$$\hat{\eta} = \hat{\beta}_0 + \sum_{i=1}^4 \hat{\beta}_i X_i + \sum_{j=2}^4 \hat{\beta}_{5j} X_{5j} + \sum_{k=6}^7 \hat{\beta}_k X_k \quad (3.2)$$

$\hat{\eta}$  é a componente sistemática (i.e., preditor linear), em que  $\beta_0$  corresponde ao intercepto e  $\beta_{1-7}$  aos coeficientes das variáveis  $hr$ ,  $sbp$ ,  $pt$ ,  $gl$  e  $group$ , bem como também as interações entre  $hr \times sbp$  e  $hr \times gl$ , respectivamente. A variável  $group$  é usada de acordo com a classificação do paciente, dados os valores das demais variáveis vitais. Portanto, a variável  $X_{5j}$  assume o valor 1 conforme o valor de  $j$  que especifica o grupo do paciente, e o valor 0 para todos os outros possíveis valores de  $j$ .

Para as demais variáveis vitais (i.e.,  $hr$ ,  $sbp$ , e  $pt$ ) foi utilizado o *Modelo de Regressão Linear Gamma*, cuja função de variância é mais restrita do que no *Modelo Normal Inversa*, dado por (3.3)

$$\hat{\mu} = \hat{\eta}^{-1} \quad (3.3)$$

onde  $\hat{\mu}$  é a média de cada sinal que o desenvolvedor deseja modelar; neste caso  $hr$ ,  $sbp$  e  $pt$ . Uma vez que o procedimento para mapear os modelos de regressão para todos os sinais vitais são similares, esses modelos de regressão foram omitidos.

### 3.1.3 Etapa 3 – Desenvolver um Modelo AOD

Nesta etapa, os modelos de regressão de cada sinal vital são mapeados para um modelo formal baseado no paradigma AOD. Para este propósito, é necessário escolher um

conjunto de parâmetros para definir o perfil do paciente. O modelo de paciente consiste da integração dos modelos de regressão para gerar os dados sintéticos inerentes aos sinais vitais de interesse, juntamente com os atributos demográficos que definem o perfil do paciente.

Os modelos de regressão mapeados para um modelo AOD devem permitir a interação entre eles e fornecer as dinâmicas comportamentais ao modelo de paciente. Assim, o desenvolvedor pode alterar o valor de um sinal vital específico durante uma simulação, e os valores dos demais sinais vitais serão automaticamente recalculados de acordo com seus respectivos modelos de regressão.

Na Tabela 3.2, é descrito como mapear os termos dos modelos de regressão para atores AOD, incluindo uma explicação sobre cada componente AOD.

Termo do Modelo de Regressão	Ator AOD	Explicação do Ator
<i>Intercepto</i>	<i>Constant</i>	Produz uma saída constante. O valor da saída é aquele do <i>token</i> presente no parâmetro <i>value</i> , que por padrão é um inteiro com valor 1.
<i>Coefficiente</i>	<i>Scale</i>	Multiplica a entrada por uma constante fornecida como parâmetro.
<i>Variável Preditora</i>	<i>Input port</i>	Uma “ <i>IOPort</i> ” que define o tipo de dado de uma entrada.
<i>Interação entre Variáveis Preditoras</i>	<i>Expression</i>	Em cada disparo, avalia uma expressão que pode incluir referências às entradas, tempo atual, e um contador de disparos. As portas são referenciadas pelos identificadores que possuem o mesmo nome da porta.
<i>Símbolo Matemático</i>	<i>AddSubtract</i>	Um adicionador/subtrator polimórfico. Este adicionador possui duas portas de entrada, ambas do tipo <i>multiport</i> , e uma porta de saída convencional. Os dados que chegam na porta de entrada chamada “ <i>plus</i> ” serão somados, e os dados que chegam na porta de entrada chamada “ <i>minus</i> ” serão subtraídos.
$\hat{\eta}$ (Preditor Linear)	<i>Expression</i>	-
$\hat{\mu}$ (Valor Médio)	<i>Output port</i>	Uma “ <i>IOPort</i> ” que define o tipo de dado de uma saída.

Tabela 3.2: Mapeamento dos Modelos de Regressão para o Paradigma AOD.

Todos os componentes AOD estão disponíveis como componentes gráficos na inter-

face do usuário no *Ptolemy II*.

Por exemplo, para mapear o modelo de regressão para a variável vital *rr* dado por (3.4) para o modelo AOD ilustrado na Figura 3.3, o intercepto é representado por um ator *Constant*; coeficientes por um ator *Scale*; variáveis preditoras por portas *Input*; e as interações entre variáveis (e.g.,  $X_6$  e  $X_7$ ) por atores *Expression*. A variável preditora *group*, que identifica o grupo ao qual o paciente pertence e cuja classificação é baseada em seus sinais vitais, é representada por um ator *Constant*. Este ator mapeia o vetor de coeficientes que pode ser usado na equação do modelo de regressão de acordo com a entrada *group* via porta de entrada, como apresentado em (3.2). Para representar os sinais antes de cada termo de (3.4), tal como  $-0,0001206X_3$ , a saída de cada termo é ligada à respectiva entrada do ator *AddSubtract*, que define cada sinal (i.e., “plus” ou “minus”). No caso do *MLG\_RR*, é necessário que o valor absoluto da saída do ator *AddSubtract* seja obtido usando o ator *AbsoluteValue* para fornecer o valor predito do sinal vital *rr*, como definido em (3.1). Para representar essa equação em AOD, utilizou-se também um ator *Expression*, que eleva o valor de  $\hat{\eta}$  a  $-0,5$ . A saída do modelo AOD representando o modelo de regressão para um sinal vital é arredondado para o valor inteiro mais próximo usando um ator *Round*. O resultado é o dado sintético gerado para o sinal vital modelado, neste caso, o sinal de frequência respiratória.

$$\begin{aligned} \hat{\eta} = & 0,002556 - 0,00005756X_1 + 0,00005381X_2 - 0,0001206X_3 - 0,00001715X_4 + \\ & -0,0009802X_{52} - 0,002567X_{53} + 0,001027X_{54} + \\ & -0,0000004994X_6 + 0,0000001579X_7 \end{aligned} \quad (3.4)$$

Esse mesmo processo de mapeamento pode ser usado para os outros sinais vitais. Na Figura 3.4, um modelo de paciente é ilustrado considerando todos os quatro sinais vitais, o perfil do paciente e os modelos de regressão já integrados.

Na Figura 3.4a, apresenta-se um modelo de paciente incorporando as características do paciente e os modelos de regressão dos quatro principais sinais vitais. Os seguintes elementos chave são destacados: (1) os parâmetros de configuração do modelo de paciente, incluindo o *Continuous Director*, que determina sua semântica de execução; (2) a subestrutura do modelo que especifica os valores iniciais dos sinais vitais e dos

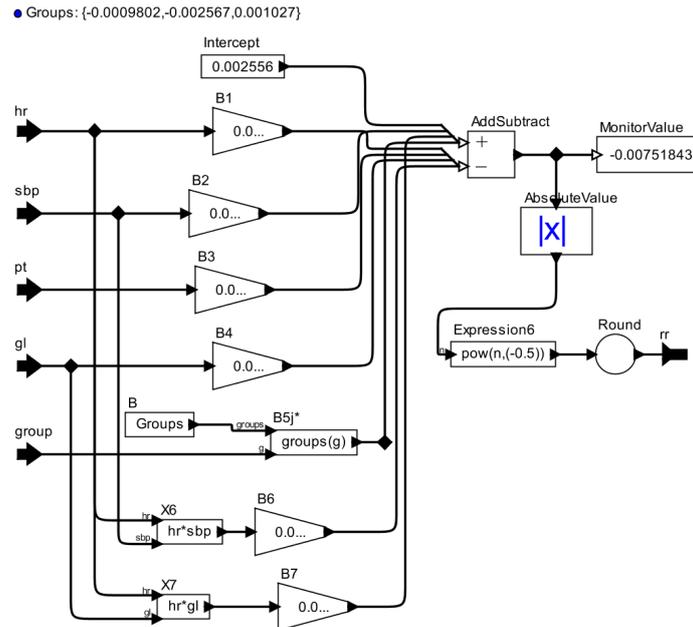
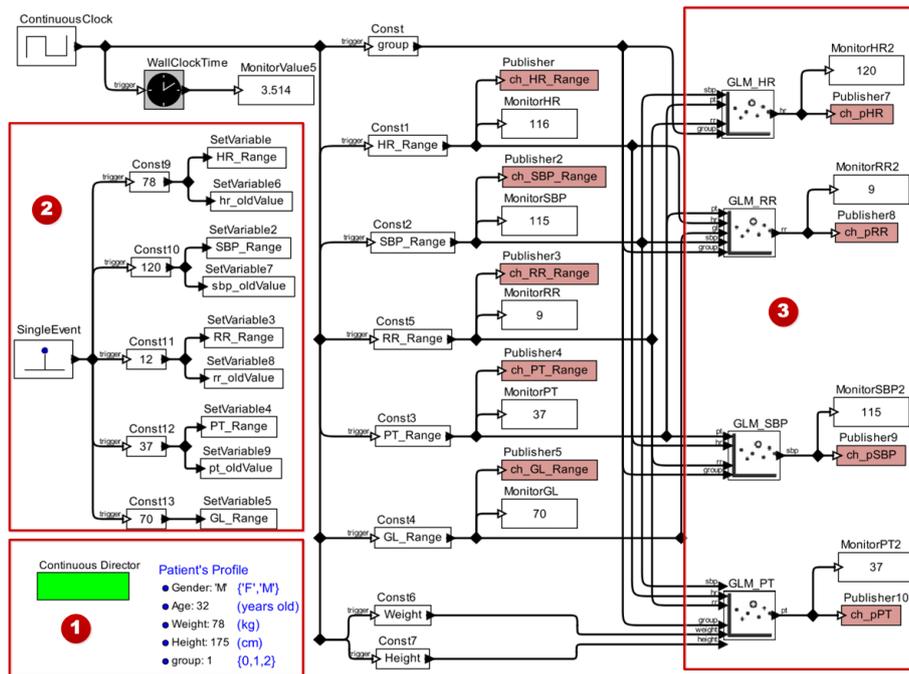


Figura 3.3: *MLG\_RR* para o Modelo de Paciente.

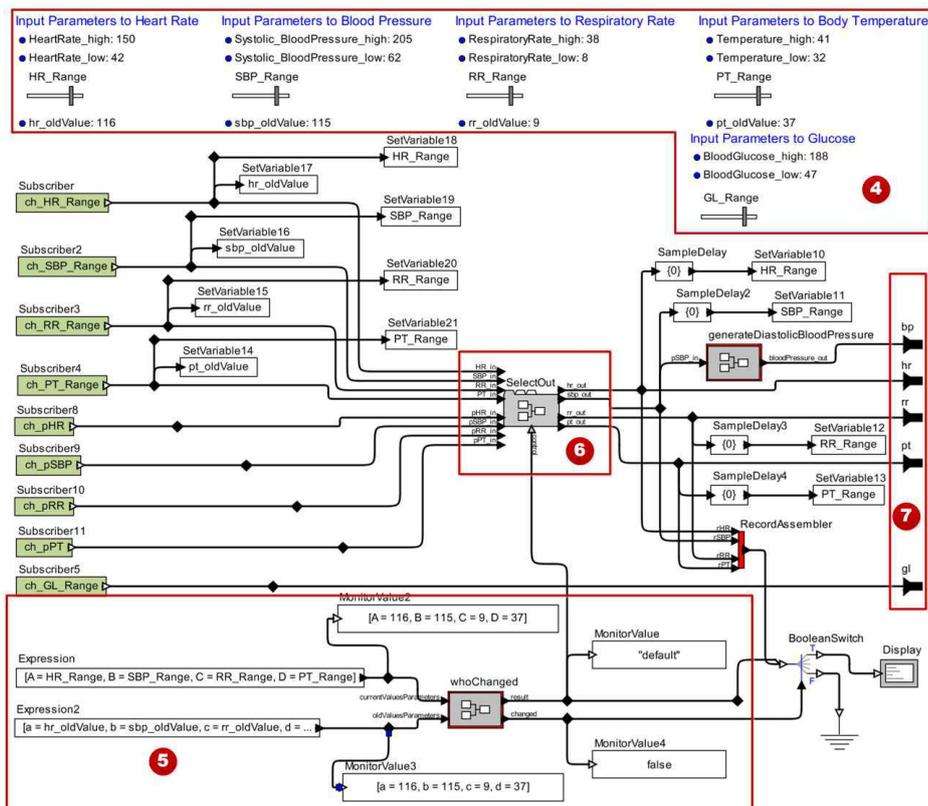
parâmetros fisiológicos representados no modelo; **(3)** os elementos rotulados pelo padrão *MLG\_<sigla\_do\_sinal\_vital>*, que denotam os modelos de regressão para cada sinal vital modelado. Os parâmetros do modelo são a base para gerar os valores para os sinais vitais fornecidos pelo modelo de paciente. Além do mais, as especificações dos limiares de cada sinal vital permitem ao desenvolvedor manipulá-los durante a simulação para representar diferentes condições de saúde para o modelo de paciente. Conseqüentemente, a análise de comportamento do SMFC pode ser realizada para várias situações.

A segunda parte de um modelo de paciente é ilustrada na Figura 3.4b. Os elementos destacados são: **(4)** os parâmetros que definem os limiares para cada sinal modelado para assegurar que esses sinais permaneçam dentro do intervalo de valores considerados na construção dos modelos de regressão; **(5)** a lógica empregada para identificar que sinal vital foi alterado pelo usuário em um dado instante de tempo da simulação do modelo; **(6)** o componente desenvolvido para selecionar a saída correta do modelo conforme intervenção do usuário; e **(7)** as portas que fornecem as interfaces de comunicação do modelo de paciente com os modelos de dispositivos médicos para a aquisição de dados.

Nota-se que o modelo de paciente atual possui apenas portas de saída para os sinais



(a) Modelos de regressão para estimar os dados sintéticos dos sinais vitais.



(b) Sub-modelo para prover informações para os dispositivos médicos.

Figura 3.4: Exemplo de um Modelo de Paciente.

vitais (i.e., *hr*, *rr*, *sbp*, e *pt*) e parâmetros fisiológicos (i.e., *gl*). Para receber as ações de controle de realimentação a partir dos modelos de atuadores, portas de entrada devem ser adicionadas ao modelo de paciente usando o editor gráfico do arcabouço *Ptolemy II*. Adicionalmente, o modelo de paciente deve ser adaptado para representar o comportamento dinâmico do processo farmacocinético correspondente a um tipo de medicamento a ser administrado. Em outras palavras, é necessário um modelo de paciente que não apenas simule a condição básica de saúde caracterizada pelos sinais vitais de um indivíduo, mas também clinicamente responda ao tratamento. Neste caso, os modelos de atuadores são utilizados para simular a administração do medicamento.

Em geral, os processos farmacocinéticos e farmacodinâmicos para um medicamento específico são matematicamente representados por uma ou mais ODE, tipicamente caracterizando modelos não lineares. Com o *Ptolemy II*, uma ODE é mapeada para o paradigma AOD por um ator *LinearStateSpace* ou *ContinuousTransferFunction*, cujo processo de mapeamento será explicado posteriormente usando o exemplo adotado neste trabalho. Apesar do comportamento humano ser não-determinístico, o corpo humano funciona como um sistema de controle, em que cada sinal de controle deve alcançar um comportamento desejado representado por um sinal de referência [14]. A resposta ao tratamento de um paciente por meio de um medicamento específico, bem como um sistema de controle automático com realimentação pode ser matematicamente representada por um controlador PID (proporcional-integral-derivativo) que age para levar o sinal controlado para seu valor de referência. Para um controlador PID paralelo de tempo contínuo<sup>1</sup>, a função de transferência é descrita por (3.5)

$$G(s) = K_p + K_i \left( \frac{1}{s} \right) + K_d \left( \frac{N \cdot s}{s + N} \right) \quad (3.5)$$

onde  $K_p$  é o ganho do termo *proporcional*,  $K_i$  é o ganho do termo *integral*,  $K_d$  é o ganho do termo *derivativo*, e o coeficiente de filtro  $N$  define a localização do pólo no filtro derivativo, que é  $-N$ . Retomando ao exemplo adotado, o termo *proporcional* representa a taxa de infusão de insulina, que é diretamente proporcional ao *erro*. Neste trabalho, o erro é a diferença entre o nível de glicose *desejado* (sinal de referência) e o nível de glicose *atual* (sinal controlado). O termo derivativo depende da taxa de

<sup>1</sup><http://www.mathworks.com/help/simulink/slref/pidcontroller.html>

mudança no nível de glicose do paciente. Quanto mais alta a taxa de aumento do nível de glicose, maior será a dosagem de insulina administrada ao paciente. Por fim, o termo *integral* avalia a taxa de infusão de insulina por meio da integral do *erro*, forçando o sinal do erro para zero a cada iteração.

Para projetar um controlador PID, uma prática bem conhecida é linearizar as ODE. Isto facilita o processo de ajuste do controlador PID em um determinado ponto de operação para o sinal controlado. A sintonia do controlador PID foi realizada de forma automática por meio do *Control System Toolbox*, um pacote de software do Matlab® que contém um conjunto de funções desenvolvidas para a Engenharia de Controle de Sistemas. Informações acerca do processo de sintonia do controlador PID, tais como o *overshoot*<sup>2</sup>, não foram fornecidas pelos autores. Ao término dessa etapa foi alcançada a *estabilidade* do processo em malha fechada, e encontrados os seguintes coeficientes:  $K_p = -36,32$ ,  $K_d = -587,47$  e  $K_i = -0,42$ , além de  $N = 9,0795$ . No caso do *Ptolemy II*, este também possui um ator PID para mapear a equação (3.5) para o paradigma AOD. Para maiores detalhes sobre questões relacionadas ao projeto de controladores PIDs, o leitor interessado pode consultar Bequette [10].

Para demonstrar esta etapa considerando o exemplo adotado, modelou-se o efeito da insulina para controlar o nível de glicose do paciente. Segundo Bergman, Phillips e Cobelli [11], o sistema de regulação da concentração de glicose é visualizado como sendo dividido em duas partes: (a) o segmento dependente da glicose, que determina a insulina no plasma; e (b) o segmento dependente da insulina, que determina a glicose no plasma. No segmento dependente da glicose (i.e., *Cinética da Glicose*) ocorre a degradação da insulina pelo pâncreas e tecidos. No segmento dependente da insulina (i.e., *Cinética da Insulina*) ocorre a produção da glicose e sua utilização pelos tecidos. Os autores propõem dois modelos matemáticos “mínimos” para descrever a dinâmica da glicose e a resposta à insulina durante um teste de tolerância à glicose intravenosa. Usando esses modelos, parâmetros característicos da sensibilidade à insulina e responsividade pancreática são gerados. Esse conjunto de modelos matemáticos foi nomeado de *Modelo Mínimo de Bergman* (do inglês, *Bergman’s Minimal Model – BMM*) [11].

<sup>2</sup>*Overshoot* ( $M_p$ ) é a diferença entre o valor máximo da saída do sistema e o valor final (valor em regime permanente), dividida pelo mesmo valor final, e geralmente referido em porcentagem [24]. Ou seja,  $M_p = \frac{\max y(t) - \lim_{t \rightarrow \infty} y(t)}{\lim_{t \rightarrow \infty} y(t)}$ .

O BMM assume um relacionamento em malha-fechada entre um número mínimo de três compartimentos:  $G(t)$  é a concentração de glicose instantânea no sangue ( $mg/dL$ );  $X(t)$  é a quantidade efetiva de insulina utilizada no processo de secreção da glicose no plasma ( $min^{-1}$ ); e  $I(t)$  é a concentração de insulina instantânea presente no plasma ( $\mu U/mL$ ). Cada compartimento é descrito por uma ODE, como mostrado em (3.6), (3.7), e (3.8).

$$\frac{dG(t)}{dt} = -p_1[G(t) - G_b] - X(t)G(t) + [D(t) + C(t)] \quad (3.6)$$

$$\frac{dX(t)}{dt} = -p_2X(t) + p_3[I(t) - I_b] \quad (3.7)$$

$$\frac{dI(t)}{dt} = -n[I(t) - I_b] + \gamma[G(t) - h]t + r(t) \quad (3.8)$$

onde  $D(t)$  e  $C(t)$  são parâmetros que representam perturbações ao paciente ( $mg/dL/min$ ). Por exemplo, a taxa de glicose absorvida após ingerir alimentos e a taxa de infusão de glucagon administrada para evitar episódios de hipoglicemia, respectivamente.  $G_b$  e  $I_b$  são os níveis basais de glicose e insulina no sangue.

Os demais parâmetros do BMM não linear são: o coeficiente  $p_1 = 0,0337$  ( $min^{-1}$ ), que define a efetividade de insulina quando a glicose permanece em  $I_b$ ; e  $-p_3/p_2$ , o efeito da insulina no nível da glicose para mantê-la dentro do intervalo normal de um indivíduo, equivalente à sensibilidade a insulina, em que  $p_2 = 0,0209$  ( $min^{-1}$ ) e  $p_3 = 7,5 \times 10^{-6}$  ( $min^{-2}$  ( $\mu U/mL$ ));  $n = 0,214$  ( $min^{-1}$ ) é uma constante de tempo que controla a taxa em que a insulina está sendo consumida no plasma;  $h$  ( $mg/dL$ ) é o limiar inferior da glicose sanguínea acima do qual a insulina endógena é secretada;  $\gamma$  ( $\mu U/mL$ ) é a taxa de entrega endógena de insulina quando a glicose é infundida de forma exógena e a concentração da glicose sanguínea está acima do limiar  $h$ , e, por fim,  $t = 5$  ( $min$ ) é o tempo de simulação.

Neste trabalho, utilizou-se uma forma linearizada do *Modelo Mínimo de Bergman* apresentado por Khan *et al.* [41], que é descrito por (3.9), (3.10), e (3.11)

$$\frac{dG(t)}{dt} = -p_1 [G(t) - G_b] - \bar{X} \cdot G(t) - \bar{G} \cdot X(t) + \bar{G} \cdot \bar{X} + [D(t) + C(t)] \quad (3.9)$$

$$\frac{dX(t)}{dt} = -p_2 X(t) + p_3 [I(t) - I_b] \quad (3.10)$$

$$\frac{dI(t)}{dt} = -n [I(t) - I_b] + \tau \cdot r(t) \quad (3.11)$$

onde  $\bar{G} = 0,81$  ( $mg/mL$ ) e  $\bar{X} = 0,0054$  ( $\mu U/mL$ ) são os valores médios de  $G(t)$  e  $X(t)$ . A saída do controlador  $r(t)$  ( $\mu U/mL/min$ ) age para manter o nível de glicose sanguínea igual a  $G_b = 0,811$  ( $mg/mL$ ) e  $\tau$  é um fator de conversão de unidade de  $r(t)$  para  $U/h$ .

O processo de *Linearização* resulta em um modelo linear que se aproxima de um sistema não linear, especificamente para um ponto de operação. De acordo com Franklin, Powell e Emami-Naeini [23], é possível obter um modelo linearizado que é válido próximo a um ponto de operação (i.e., ponto de equilíbrio). Sempre que o estado do sistema difere do ponto de operação, tanto o processo de linearização como o projeto do controlador deve ser realizado novamente.

Para o processo de um modelo linear em espaço de estados serão utilizadas as ODEs (3.6), (3.7) e (3.11). Assim, seja o vetor de estados  $x(t)$  e o vetor de entradas  $u(t)$ , onde  $x(t) = \begin{bmatrix} G(t) & X(t) & I(t) \end{bmatrix}^T$  e  $u(t) = \begin{bmatrix} D(t) & C(t) & r(t) \end{bmatrix}^T$ . O BMM não linear pode ser definido por  $\dot{x} = f(x, u) = \begin{bmatrix} f_0(x, u) & f_1(x, u) & f_2(x, u) \end{bmatrix}^T$  que representam as equações (1-3). Determina-se um modelo linearizado em torno do ponto de operação  $(x_0, u_0)$ , em que  $x_0 = \begin{bmatrix} G_0 & X_0 & I_0 \end{bmatrix}^T$  e  $u_0 = \begin{bmatrix} D_0 & C_0 & r_0 \end{bmatrix}^T$ , tal que (3.12) implica (3.13).

$$\dot{x}|_{\substack{x=x_0 \\ u=u_0}} = 0 \Rightarrow \begin{bmatrix} f_0(x_0, u_0) \\ f_1(x_0, u_0) \\ f_2(x_0, u_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \quad (3.12)$$

$$\Rightarrow \begin{bmatrix} -p_1(G_0 - G_b) - X_0 G_0 + D_0 + C_0 \\ -p_2 X_0 + p_3(I_0 - I_b) \\ -n(I_0 - I_b) + \tau \cdot r_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} G_0 = \frac{p_1 G_b + D_0 + C_0}{p_1 + \frac{p_3 \tau \cdot r_0}{p_2 \cdot n}} \\ X_0 = \frac{p_3 \tau \cdot r_0}{p_2 \cdot n} \\ I_0 = I_b + \frac{\tau \cdot r_0}{n} \end{cases} \quad (3.13)$$

Ao aplicar o processo de linearização demonstrado por Ogata [66], o BMM linearizado pode ser definido como  $\dot{x} - f(x_0, u_0) = A\Delta x + B\Delta u$ , como mostrado em (3.14) e representado por (3.15).

$$\Delta \dot{\mathbf{x}} = \underbrace{\mathbf{A}}_{\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}}} \Delta \mathbf{x} + \underbrace{\mathbf{b}}_{\left. \frac{\partial \mathbf{f}}{\partial u} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}}} \Delta u \quad (3.14)$$

$$\begin{bmatrix} \dot{G}(t) \\ \dot{X}(t) \\ \dot{I}(t) \end{bmatrix} = \begin{bmatrix} -p_1 - X_0 & -G_0 & 0 \\ 0 & -p_2 & p_3 \\ 0 & 0 & -n \end{bmatrix} \begin{bmatrix} \Delta G(t) \\ \Delta X(t) \\ \Delta I(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta D(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Delta C(t) + \begin{bmatrix} 0 \\ 0 \\ \tau \end{bmatrix} \Delta r(t) \quad (3.15)$$

O comportamento do modelo de paciente  $H(s)$  proposto por Khan *et al.* [41] é baseado na *Transformada de Laplace* das Equações (3.9)–(3.11) dado por (3.16),

$$H(s) = T_1(s) \cdot R(s) + T_2(s) [D(s) + C(s)] + T_3(s) \cdot I(s) \quad (3.16)$$

onde  $R(s)$  é a resposta à perturbação,  $I(s)$  é a *Transformada de Laplace* em resposta ao impulso e  $T_n(s)$  são as funções de transferências, em que  $n \in \{1, 2, 3\}$ . As funções de transferências relacionam as entradas para o modelo de paciente. Khan *et al.* [41] apresentam mais informações em relação ao uso do *Modelo Mínimo de Bergman*.

Uma vez compreendida a dinâmica a ser incorporada no modelo de paciente, tais dinâmicas são representadas usando o *Ptolemy II*, como mostrado na Figura 3.5.

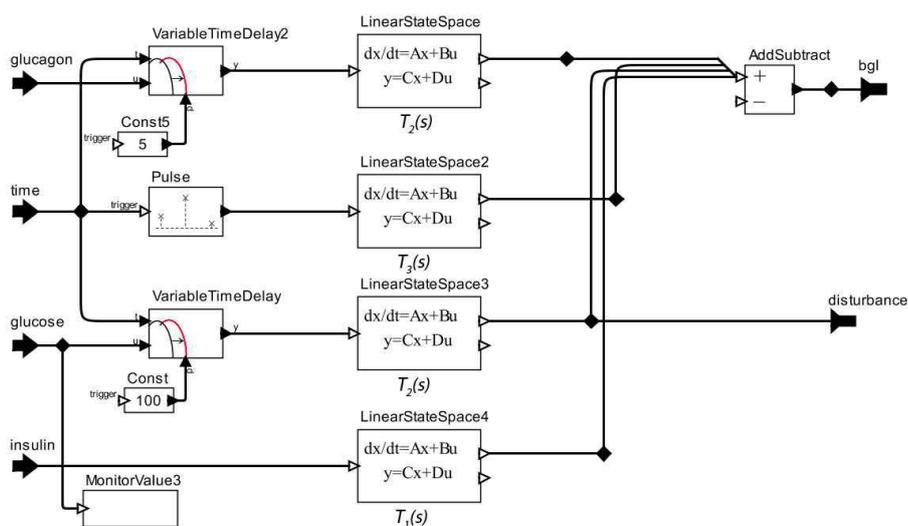


Figura 3.5: Modelo de Paciente Baseado em Khan *et al.* [41].

Dado que um sistema dinâmico linear pode ser representado por diferentes formas, tais como equações diferenciais, funções de transferência, espaço de estados, resposta em frequência e resposta ao impulso, é possível transformar a representação de um sistema de qualquer forma para outra [23]. Com o Matlab® é permitido realizar essas conversões automaticamente por meio de comandos específicos. Neste trabalho, o modelo de paciente proposto por Khan *et al.* [41] foi transformado da forma de *funções de transferência* para uma representação equivalente em *espaço de estados* usando o comando `tf2ss`. O relacionamento analítico entre a descrição em espaço de estados e a descrição em função de transferência é demonstrado por Franklin, Powell e Emami-Naeini [23]. A representação em espaço de estados foi utilizada porque o estado do modelo de paciente relacionado à glicose requer um *estado inicial*. Esse valor inicial para um estado específico não pode ser definido usando uma função de transferência.

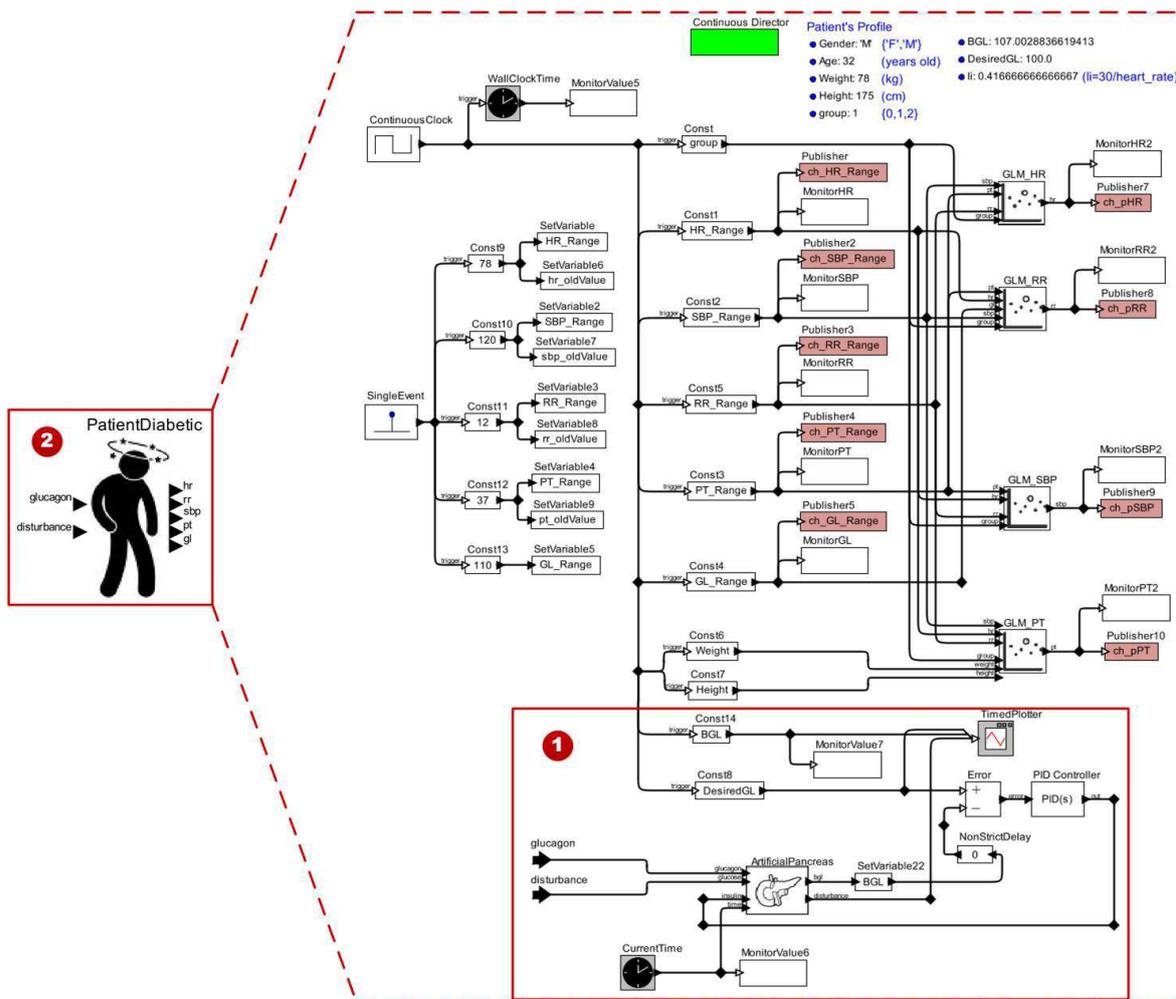


Figura 3.6: Modelo de Paciente Diabético.

O próximo passo desta etapa é transformar o modelo mostrado na Figura 3.5 para um modelo de ator e incluí-lo no modelo de paciente original. Para isso, portas de entradas devem ser adicionadas ao modelo de paciente para receber as ações dos dispositivos atuadores, bem como também de perturbações. Em seguida, é necessário estabelecer os relacionamentos entre os atores do Modelo de Paciente Diabético, visando representar a dinâmica de resposta à insulina no modelo de paciente. O resultado desse processo é mostrado na Figura 3.6, destacando as alterações chave realizadas no modelo de paciente: (1) os novos atores e portas de entrada adicionadas; e (2) a abstração do Modelo de Paciente Diabético como um ator a ser disponibilizado na biblioteca do usuário do *Ptolemy II*.

### 3.1.4 Etapa 4 – Validar um Modelo de Paciente

Nesta etapa, o comportamento do modelo de paciente deve ser validado para assegurar sua acurácia. Isto inclui analisar se os modelos de regressão calculam corretamente os dados gerados durante a simulação do modelo AOD. Essa etapa é finalizada após o desenvolvedor evidenciar o comportamento correto do modelo de paciente. Caso contrário, o desenvolvedor do SMFC deve realizar ajustes no modelo.

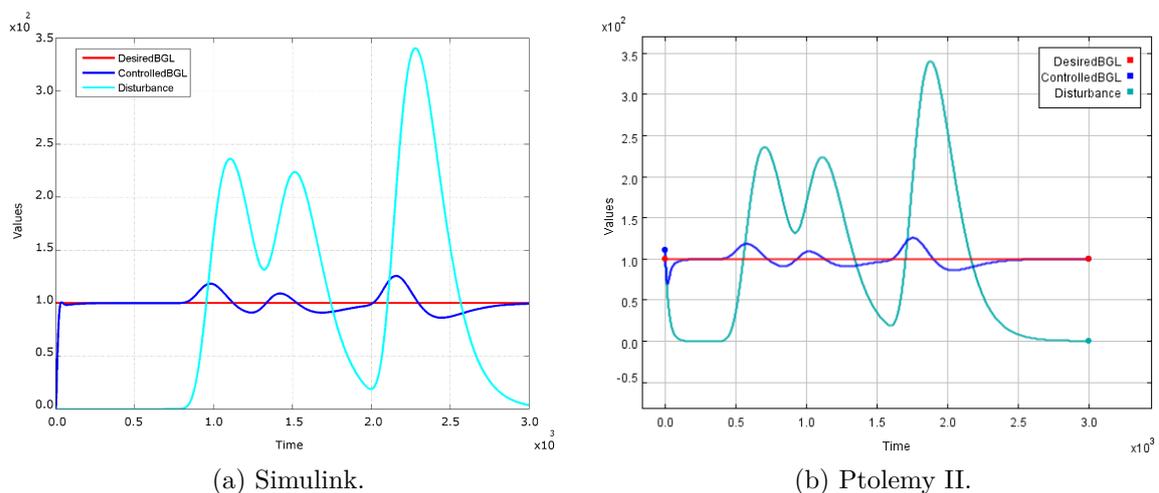


Figura 3.7: Resultado Comparativo Entre o Sistema de Controle do Nível de Glicose Sanguínea (BGL) Proposto por Khan *et al.* [41] Obtido Usando o *Simulink* e o Modelo Replicado no *Ptolemy II* após Simulação de 3000 Unidades de Tempo.

Para validar o Modelo de Paciente Diabético, o sistema de controle de glicose san-

guínea (BGL) foi replicado no *Ptolemy II* e comparado às saídas após simulação, como mostrado na Figura 3.7. Observa-se que ambos modelos de sistemas de controle BGL possuem comportamento semelhante. Assim, eles apresentam o mesmo formato de curva para o nível de glicose sanguínea controlada (*ControlledBGL*) gerado pelos modelos de pacientes. Isto permite também afirmar que os modelos de pacientes representados tanto no *Simulink* (i.e. como idealizado por Khan *et al.* [41]) quanto no *Ptolemy II* (i.e., modelo replicado) respondem igualmente às perturbações. Em outras palavras, nesses modelos de pacientes foi incorporada a mesma dinâmica de reação à insulina. Por outro lado, o modelo de paciente original, diferentemente do modelo replicado, não considera um estado inicial para o parâmetro da glicose. Esse comportamento pode ser notado durante as primeiras cem unidades de tempo de simulação. Outra mudança observada na Figuras 3.7a e 3.7b são as compensações (i.e., *offsets*) relacionadas à curva de perturbação, que não causam problemas para validar o comportamento do modelo de paciente.

## 3.2 Construção dos Modelos de Dispositivos Médicos

Modelos de dispositivos médicos são modelos formais construídos a partir da especificação técnica de dispositivos sensores e atuadores. Exemplos de tais dispositivos incluem medidores de pressão arterial, balanças, glicosímetros, e bombas de infusão. Modelos de sensores são capazes de capturar dados sobre a condição de saúde de pacientes. Modelos de atuadores administram medicamentos para manter o modelo de paciente em condição estável. Esses medicamentos são administrados para alcançar um efeito terapêutico benéfico em algum processo no organismo do paciente [39].

Em geral, modelos de sensores possuem uma estrutura mais simples se comparados aos modelos de atuadores. Essa característica está relacionada ao mais alto nível de segurança funcional requerido para esses dispositivos.

O processo para construção de um modelo de dispositivo médico consiste de três etapas. A explicação para cada passo é dada nas subseções seguintes, usando um exemplo de um modelo para uma bomba de insulina.

### 3.2.1 Etapa 1 – Escolher um Dispositivo Médico Certificado

Esta etapa visa identificar, analisar e selecionar uma especificação técnica que melhor descreve as características do dispositivo médico que o desenvolvedor deseja modelar. Os dispositivos médicos escolhidos devem ser certificados por uma agência regulatória tal como a FDA [20].

Para o exemplo utilizado, escolheu-se o guia do usuário de um sistema de bomba de insulina certificado [74] para elicitar os requisitos a serem implementados nesse modelo de atuador. Na Tabela 3.3 são apresentadas as funcionalidades chave para o funcionamento adequado da bomba de insulina. Uma vez que o objetivo é simular o dispositivo médico dado seu comportamento funcional, as características físicas e eletromagnéticas foram ignoradas.

Parâmetros de Configuração	Funcionalidades	Propriedades de Segurança
1. Tipo de Insulina: U100 <sup>(1)</sup> 2. Capacidade do Cartucho: 3,15ml <sup>(2)</sup> 3. Dose Basal (U/h): Min. = 0,1 Max. = 25,0 4. Perfil <i>Basal</i> : a. Padrão (# doses fixas/3min): 480 b. Personalizado (# doses ajustadas/h): 24 5. <i>Bolus</i> Padrão (U): Max. = 25,0 6. Taxa de Administração (dose/min) <sup>(3)</sup> : a. Dose <i>Basal</i> b. Dose <i>Bolus</i>	1. Modos Executar/Parar; 2. Selecionar perfil <i>basal</i> ; 3. Verificar nível do cartucho; 4. Programar dose de <i>bolus</i> padrão; 5. Prover informações ao usuário.	1. Erros(E)/Alertas(A): a. E: “Cartucho Vazio!” b. A: “Nível Baixo do Cartucho!” c. A: “Cartucho OK!” 2. Estado: a. “STOP” b. “EXECUTING”

<sup>(1)</sup> Insulina de ação curta/rápida;

<sup>(2)</sup> Equivalente à 315U de insulina U100;

<sup>(3)</sup> Simplificação, uma vez que na especificação adotada a taxa de administração é 0,2 U/s.

Tabela 3.3: Requisitos Técnicos Considerados na Modelagem da Bomba de Insulina.

O tratamento de um paciente diabético por terapia de insulina (i.e., insulino terapia) requer que múltiplas dosagens de insulina sejam administradas ao longo do dia. Um endocrinologista prescreve pelo menos dois tipos de dosagens de insulina ao paciente diabético; a insulina *basal* e a tipo *bolus* [18]. A insulina *basal* é usada para manter os níveis de glicose sanguínea dentro do intervalo normal de um indivíduo durante os períodos em jejum. Por ser de ação longa, ela é geralmente tomada uma ou duas vezes ao dia, dependendo da insulina. Por outro lado, a insulina tipo *bolus* é especificamente tomada em períodos específicos para manter os níveis de glicose sob controle, por exemplo, após uma refeição ou prática de exercícios físicos. Esta é uma insulina de ação rápida em que a administração segue as diretrizes médicas. Este é um cenário

clínico específico de uso da bomba de insulina.

### 3.2.2 Etapa 2 – Desenvolver um Modelo AOD

Nesta etapa, o modelo de dispositivo médico é construído em conformidade com a especificação técnica considerando as funcionalidades chave do dispositivo médico e as propriedades de segurança para o comportamento adequado do modelo. Uma vez que os requisitos para modelar a bomba de insulina são definidos, o próximo passo é construir o modelo. Uma visão geral do modelo da bomba de insulina é apresentada na Figura 3.8.

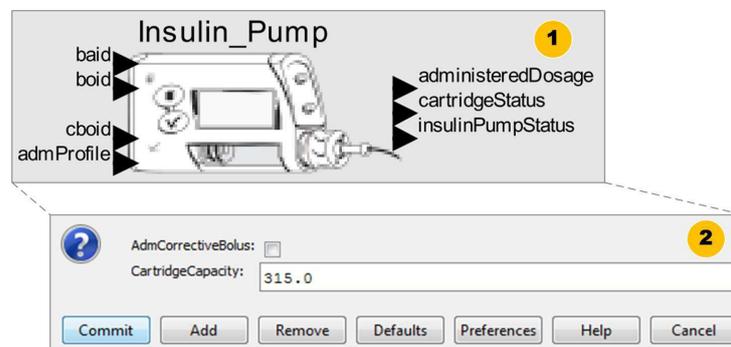


Figura 3.8: Modelo da Bomba de Insulina.

O componente *ator* usado para abstrair a lógica de funcionamento da bomba de insulina é apresentado na Figura 3.8 (1). Este componente instancia o modelo da bomba de insulina dentro de um modelo de SMFC. A bomba de insulina possui como entrada o identificador do perfil de administração de insulina (*admProfile*), e um conjunto de interfaces para receber os valores correspondentes às dosagens de insulina dos tipos *basal* (*baid*), *bolus* padrão (*boid*) e *bolus* corretivo (*cboid*). Usuários podem configurar esses parâmetros da bomba de insulina para ajustar às suas necessidades. Este modelo fornece como saída um conjunto de informações acerca do estado da bomba (*insulinPumpStatus*), o nível de insulina remanescente no cartucho (*cartridgeStatus*) e a quantidade de insulina administrada durante sua execução.

Como ilustrado na Figura 3.8 (2), o modelo da bomba de insulina possui outros parâmetros de configuração. Por exemplo, a capacidade do cartucho (*CartridgeCapacity*) acoplado à bomba de insulina e uma opção para administrar imediatamente uma

dosagem de *bolus* corretivo (*AdmCorrectiveBolus*) ao modelo de paciente. Usuários podem configurar esses parâmetros durante a simulação do modelo.

Além disso, na Figura 3.9, é ilustrado o componente de controle da bomba de insulina, que é responsável por controlar a sua dinâmica comportamental durante uma simulação. Para defini-lo, utilizou-se o conceito de *Modelo Modal*, que consiste de um conjunto finito de comportamentos e regras que governam as transições entre eles. As regras são capturadas por uma *Máquina de Estados Finito* (MEF). Cada comportamento representa um modo de operação para cada estado da MEF. As dinâmicas da MEF podem ser especificadas por múltiplos refinamentos de estado, em que cada refinamento define o comportamento do sistema naquele modo de operação [50]. Silva *et al.* [80] apresentam mais detalhes em relação a esse componente.

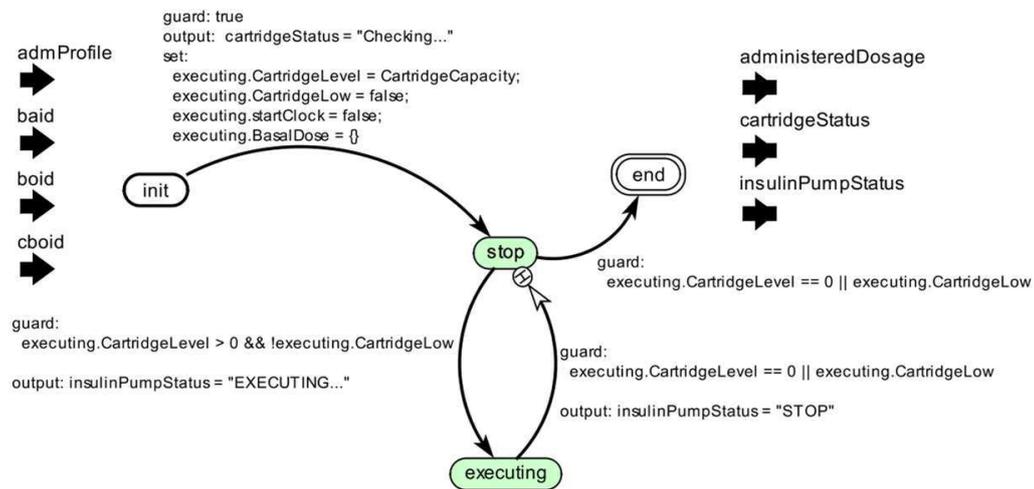


Figura 3.9: Controlador do Modelo da Bomba de Insulina.

Uma MEF pode ser generalizada para admitir entradas e saídas contínuas e combinar dinâmicas discreta e contínua. Esse modelo de computação é conhecido como uma *Máquina de Estados Estendida* (MEE) [50]. Uma MEE possui variáveis de estado numéricas, em que o número de estados depende do número de valores distintos que essas variáveis podem assumir. Portanto, para gerar os casos de simulação para os modelos de dispositivos médicos, pode-se combinar critérios de cobertura baseados em estrutura, tais como os critérios orientados ao controle de fluxo e estado/transição. Assim, é possível que modelos de atuadores interajam com modelos de pacientes, modelos de SMFC e outros modelos de dispositivos médicos.

O modelo de refinamento de estado para os modos de operação “*stop*” e “*executing*” é ilustrado na Figura 3.10. Tal refinamento possui um *Modelo de Eventos-Discreto*, em que interações entre atores ocorrem em momentos discretos no tempo. Cada interação representa um evento caracterizado pela troca de mensagens entre dois atores. A característica chave de um modelo de evento discreto é que cada ator reage aos eventos de entrada em ordem temporal [72].

Finalmente, na Figura 3.10, os resultados da simulação para o modelo da bomba de insulina são apresentados. No gráfico é mostrado o consumo histórico de insulina no cartucho como consequência das dosagens de insulina administradas (eixo Y) ao longo do período de simulação (eixo X).

### 3.2.3 Etapa 3 – Validar um Modelo de Dispositivo

O propósito desta etapa é avaliar as saídas (i.e., dados sintéticos) fornecidos pelo modelo de dispositivo médico. Essa etapa requer que os desenvolvedores instanciem o modelo via ferramenta de simulação provendo todos os parâmetros para sua execução. Assim, as entradas e saídas do modelo são comparadas para verificar se a corretude do comportamento do modelo e as propriedades de segurança foram alcançadas.

Como dito anteriormente, ao analisar os modelos de dispositivos médicos, observa-se que os modelos de sensores geralmente possuem uma estrutura mais simples que os modelos de atuadores. Essa característica está relacionada ao mais alto nível para operação segura de dispositivos atuadores, tal como a bomba de insulina. O software embarcado para os modelos de sensores são *sistemas de eventos discretos* e representam elementos do mundo cibernético. Eles possuem um *relógio local* que devem estar sincronizados com o *relógio global* do SMFC, instanciado em seu mais alto nível de abstração. Feito isto, é possível definir o intervalo de tempo em que o modelo coleta dados periodicamente a partir do modelo de paciente.

Para validar o modelo da bomba de insulina, diferentes condições de entrada são simuladas para testar todas as estratégias de administração de insulina (e.g., perfis *basal*, *bolus* padrão e *bolus* corretivo) e propriedades de segurança implementadas neste modelo. Um exemplo de tais propriedades de segurança é que a bomba de insulina deve parar de operar quando o cartucho está completamente vazio ou insuficiente para

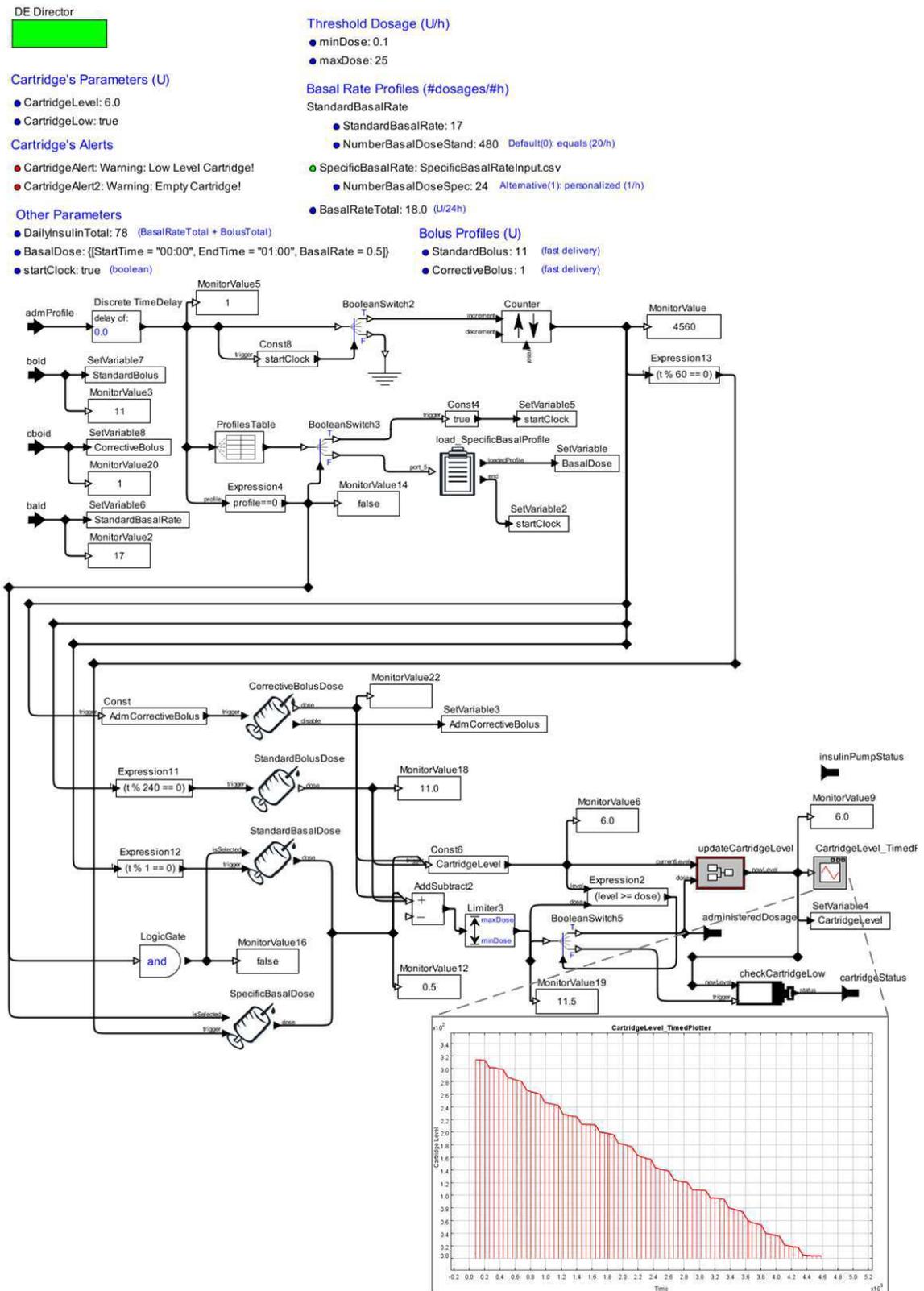


Figura 3.10: Refinamento de Estado do Modelo da Bomba de Insulina.

administrar a próxima dosagem de insulina programada. Portanto, essa etapa requer definir um cenário clínico para o uso intencionado da bomba de insulina. Na Seção 5.1 é detalhado o processo de validação do modelo da bomba de insulina.

### 3.3 Biblioteca de Modelos: Prontos para Reúso

Na prática, a biblioteca de modelos torna-se a biblioteca do usuário no *Ptolemy II*, como ilustrado na Figura 3.11.

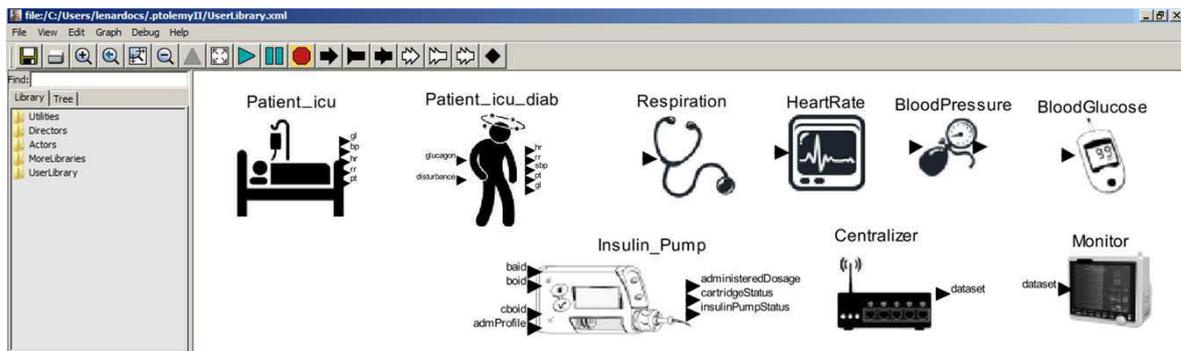


Figura 3.11: Biblioteca do Usuário no *Ptolemy II*.

O *Ptolemy II* fornece um editor gráfico com ações *drag and drop* para permitir aos desenvolvedores reutilizarem facilmente os atores e os modelos de atores. A biblioteca já possui dois modelos de pacientes (UTI e diabético), cinco modelos de sensores (monitores cardíacos e respiratório, medidor de pressão arterial, termômetro e glicosímetro), um atuador (bomba de insulina) e dois modelos de equipamentos (monitor multiparamétrico de sinais e centralizador de dados).

Silva *et al.* [81] disponibilizam o arquivo XML com todos os modelos reutilizáveis contidos nas bibliotecas de pacientes e dispositivos, como ilustrado na Figura 3.12. Tal arquivo pode ser aberto pela ferramenta *Ptolemy II* e os modelos instanciados para construir modelos de SMFC usando as diretrizes descritas no Capítulo 4.

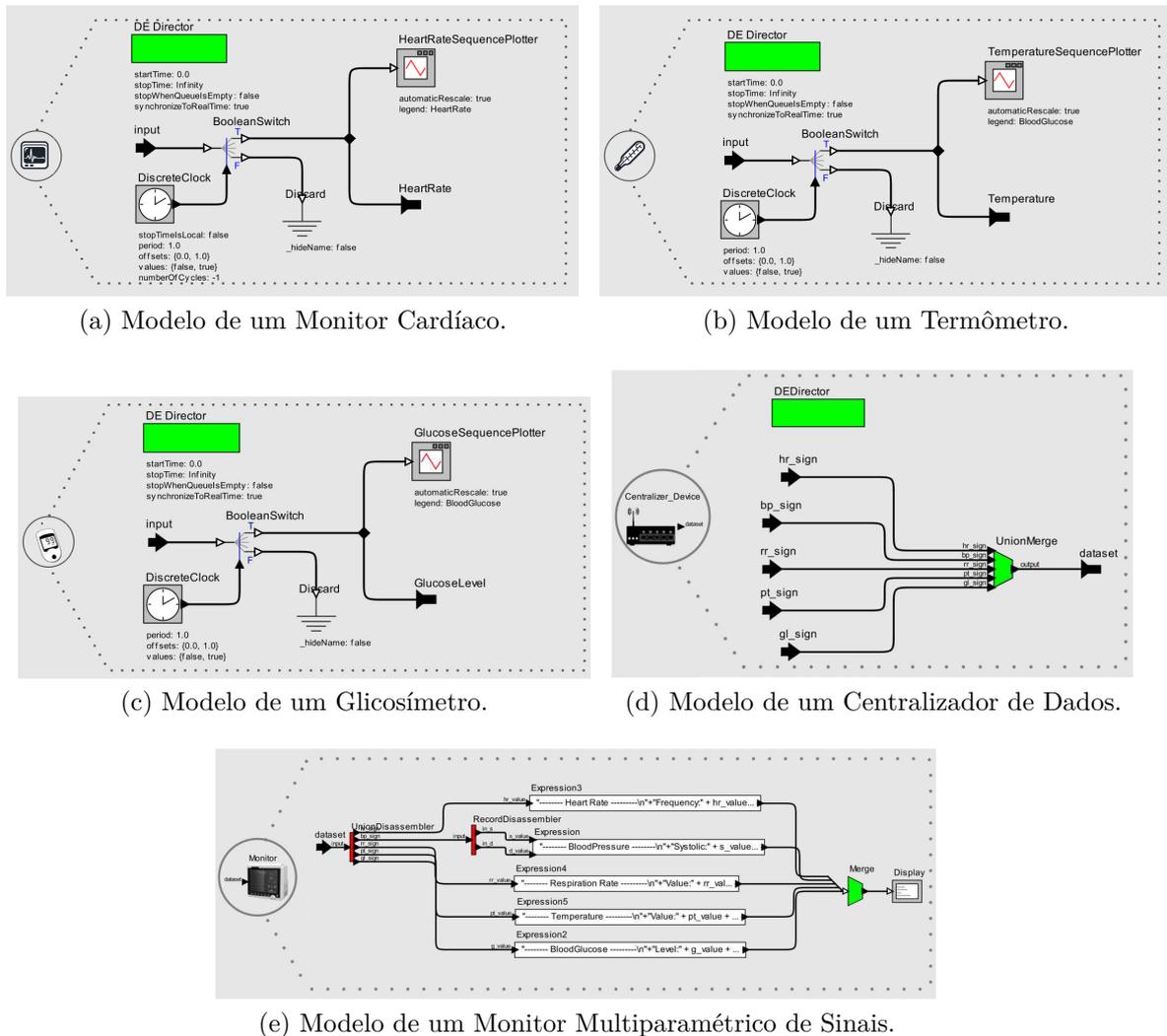


Figura 3.12: Modelos de Dispositivos na Biblioteca do Usuário do *Ptolemy II*.

### 3.4 Considerações Finais

A biblioteca de modelos é um conjunto de modelos reutilizáveis de pacientes e dispositivos reutilizáveis visando aumentar o potencial de reúso e a produtividade durante a validação de Sistemas Médicos Físico-Cibernéticos (SMFC). Neste capítulo foram descritas as diretrizes definidas para permitir aos desenvolvedores de SMFC adaptarem os modelos existentes na biblioteca ou construir seus próprios modelos de acordo com os requisitos do sistema, enriquecendo a biblioteca ao longo do tempo.

Os modelos contidos na biblioteca de dispositivos médicos são construídos em conformidade com as características funcionais e propriedades de segurança extraídas das

---

especificações técnicas de cada dispositivo de interesse. Tais modelos se comportam como *Sistemas de Eventos-Discretos*, representando parte dos elementos do mundo cibernético envolvidos no contexto de um SMFC. Diferentemente, para os modelos de pacientes, que são baseados em diretrizes médicas e bases de dados clínicas, foi adotado a semântica dos *Sistemas de Tempo Contínuo* para melhor representar o mundo físico na composição desses SMFCs, cujo processo será descrito no Capítulo 4.

Portanto, ao aplicar as diretrizes, o desenvolvedor pode construir modelos para dispositivos e pacientes a serem utilizados em diferentes contextos de SMFC. As mesmas diretrizes foram utilizadas para construir os modelos apresentados neste capítulo.

## Capítulo 4

# Composição e Simulação: Validação de SMFC

Neste capítulo é detalhado como utilizar a biblioteca de modelos formais de pacientes e dispositivos médicos para construir e simular modelos de SMFC de acordo com os requisitos do sistema. Modelos de SMFC são responsáveis por controlar os eventos e coordenar as ações de dispositivos médicos e pacientes.

Como ilustrado na Figura 4.1, a abordagem proposta consiste de: (1) projetar a arquitetura do SMFC, incluindo a seleção de modelos para pacientes (1.1) e dispositivos (1.2) e a construção do controlador do SMFC (1.3); (2) definir os cenários clínicos de interesse para simulação; e (3) realizar a simulação e a análise baseada em modelos. Nas subseções seguintes, é detalhada cada fase dessa abordagem.

Uma visão geral da arquitetura baseada em modelos de um SMFC é ilustrada na Figura 4.2. A arquitetura é composta dos seguintes componentes: (i) Modelos de Pacientes; (ii) Modelos de Dispositivos Médicos; e (iii) Modelo de SMFC (controlador). Esses modelos interagem entre si para estimular as reações de outros modelos. Essa interação caracteriza um ciclo de controle de realimentação (i.e., “*feedback control loop*”). Por exemplo, os modelos de sensores médicos monitoram o modelo de paciente para capturar os valores dos sinais vitais periodicamente e utilizá-los para tomada de decisão pelo modelo de SMFC. Em seguida, o modelo de SMFC delega as ações para os modelos de atuadores executarem mudanças no modelo de paciente.

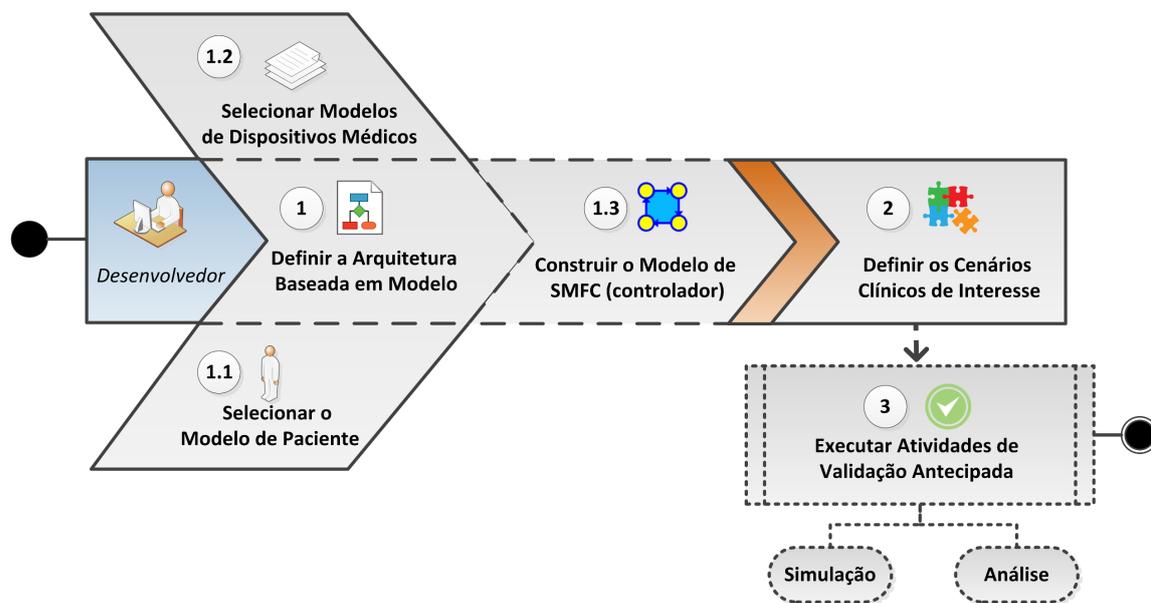


Figura 4.1: Abordagem Baseada em Modelo para Validação de SMFC.

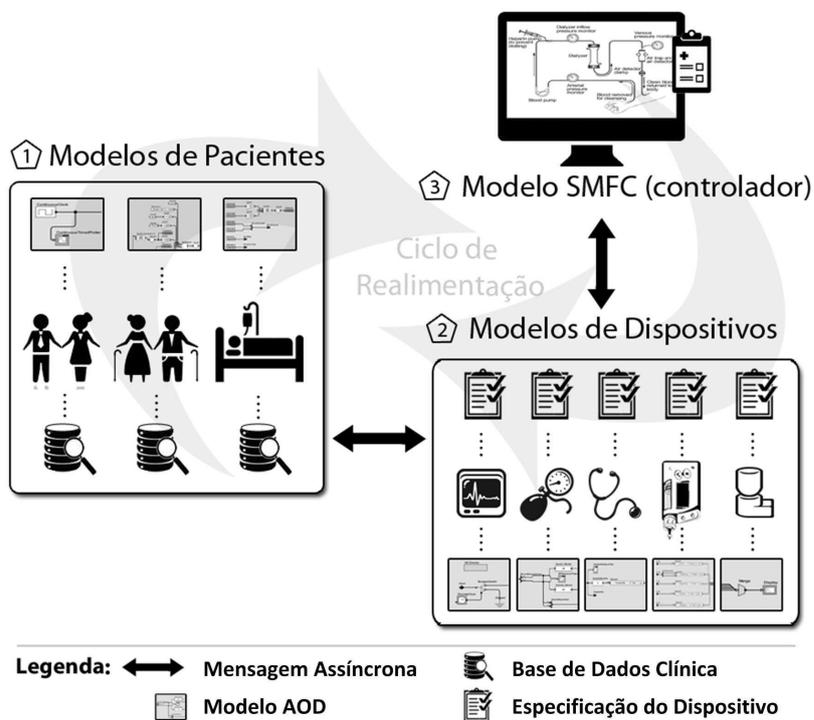


Figura 4.2: Uma Arquitetura Baseada em Modelos para SMFC [80].

## 4.1 **Projetando a Arquitetura Baseada em Modelo**

A arquitetura para o sistema a ser desenvolvido é uma composição de modelos de pacientes e dispositivos médicos, integrados pelo controlador do SMFC. O primeiro passo é selecionar os modelos de pacientes e dispositivos médicos a partir das bibliotecas disponíveis para serem integrados ao controlador do SMFC. Ambos os tipos de modelos são selecionados de acordo com os requisitos do sistema. Por exemplo, um desenvolvedor que quer construir um SMFC para ajudar especialistas de saúde a diagnosticar pessoas com Asma<sup>1</sup>, uma doença crônica pulmonar que inflama e estreita as vias aéreas, necessita identificar os sintomas primários da doença, tais como pieira (i.e., um som de assobio quando o paciente respira), aperto no peito, falta de ar e tosse. O diagnóstico de asma é baseado nos históricos médicos da família e do paciente, exames físicos e resultados de testes de laboratório. Um teste chamado espirometria é usado para verificar quanto ar o paciente pode inspirar e expirar e quão rápido ele pode soprar o ar para fora. Neste caso, o modelo de paciente deve incorporar o comportamento pulmonar e o modelo do dispositivo médico a ser selecionado no contexto desse sistema é o espirômetro.

Outras doenças levariam a outros comportamentos a serem representados pelo modelo de paciente, bem como também a outros modelos de dispositivos médicos, seja para o monitoramento ou para a atuação, combinados ao modelo de paciente. Se não há um modelo de paciente ou modelo de dispositivo disponível na biblioteca para um determinado conjunto de requisitos do SMFC, novos modelos deveriam ser construídos e adicionados à biblioteca, de acordo com o processo descrito no Capítulo 3.

Após selecionar os modelos de pacientes e dispositivos, eles devem estar integrados para compor o modelo de SMFC. Esse processo é detalhado nas próximas seções.

### 4.1.1 **Selecionando um Modelo de Paciente**

A primeira etapa é selecionar o modelo de paciente para o SMFC. A complexidade inerente ao modelo de paciente descrito no Capítulo 3 é escondida do desenvolvedor do SMFC.

---

<sup>1</sup><http://www.nhlbi.nih.gov/health/health-topics/topics/asthma>

Como ilustrado na Figura 4.3, o modelo de paciente é um modelo reutilizável na interface do usuário do *Ptolemy II*.

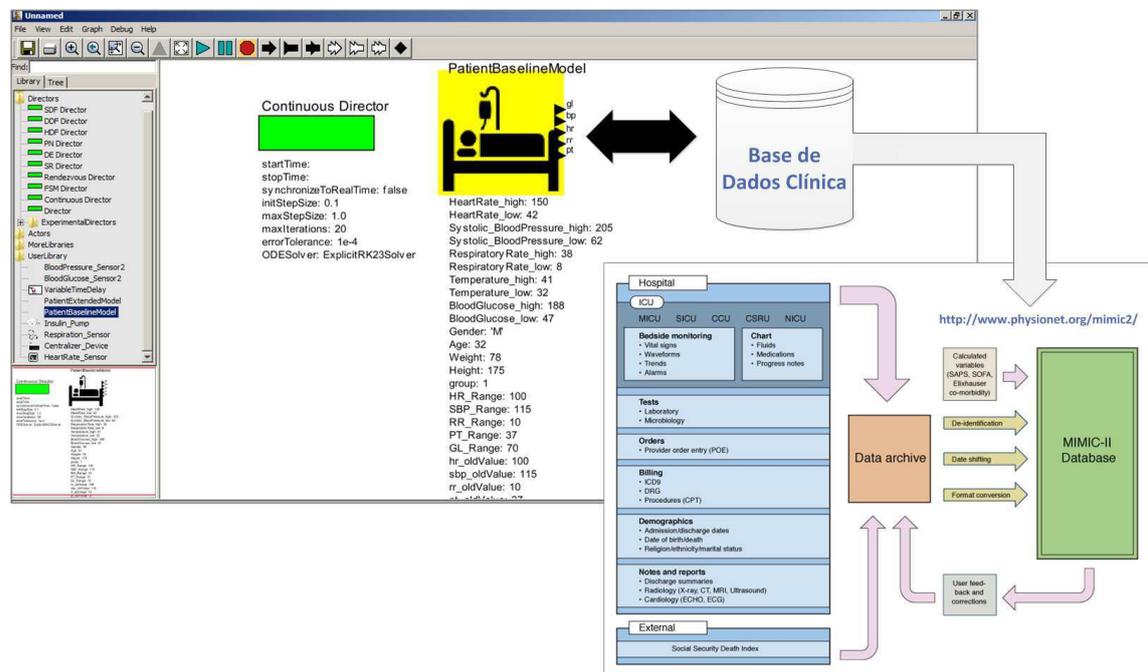


Figura 4.3: Selecionando um Modelo de Paciente no *Ptolemy II*.

A seleção do modelo de paciente depende das características da base de dados clínica requerida pelos cenários de simulação a serem avaliados pelo desenvolvedor. Nesse exemplo, o modelo de paciente abstrai o comportamento de pacientes em conformidade com a base de dados clínica *MIMIC II v2.6* [71]. Essa base de dados contém dados clínicos de pacientes admitidos em Unidades de Terapia Intensiva (UTIs).

#### 4.1.2 Selecionando os Modelos de Dispositivos Médicos

A seleção do modelo de dispositivo depende dos requisitos técnicos do dispositivo demandado pelos cenários de simulação a serem avaliados pelo desenvolvedor. No exemplo ilustrado na Figura 4.4, o modelo abstrai o comportamento de uma bomba de insulina.

#### 4.1.3 Compondo o Modelo de SMFC

O modelo de SMFC controla os eventos e coordena as ações dos modelos de dispositivos médicos e de pacientes para representar um SMFC real. Após analisar os dados



Figura 4.4: Selecionando um Modelo de Dispositivo no *Ptolemy II*.

oriundos dos modelos de sensores médicos, um modelo de SMFC delega as ações aos modelos de atuadores para administrar medicamentos de acordo com a condição de saúde do paciente. O estado atual do paciente é caracterizado pelos dados de seus sinais vitais gerados pelo modelo de paciente.

Essa ideia de *realimentação* em SMFC pode ser compreendida como o uso de qualquer informação disponível sobre o comportamento do sistema para ajustar continuamente a entrada para o sistema [14]. Essa entrada é frequentemente visualizada como um sinal de controle que visa alcançar um comportamento desejado (i.e., sinal de referência). Assim, a propriedade chave de realimentação é fazer correções, especialmente quando ocorrem perturbações inesperadas ao sistema. Um modelo de sistema controlado que usa a definição de realimentação permite ao desenvolvedor implementar duas formas de controle: (i) Controle em Malha-Aberta (do inglês, *Open-loop Control*), em que a entrada permanece fixa independente do efeito (bom ou ruim) que ela causa na saída observada; e (ii) Controle em Malha-Fechada (do inglês, *Closed-loop Control*), em que a entrada depende do efeito que ela causa na saída. Essa lógica de controle depende das características de cada SMFC, sendo o desenvolvedor responsável por modelar as regras de controle do sistema.

A integração de modelos de pacientes e dispositivos médicos ao modelo de SMFC pode também ser feita por meio da interface intuitiva do *Ptolemy II* (Figura 4.5).

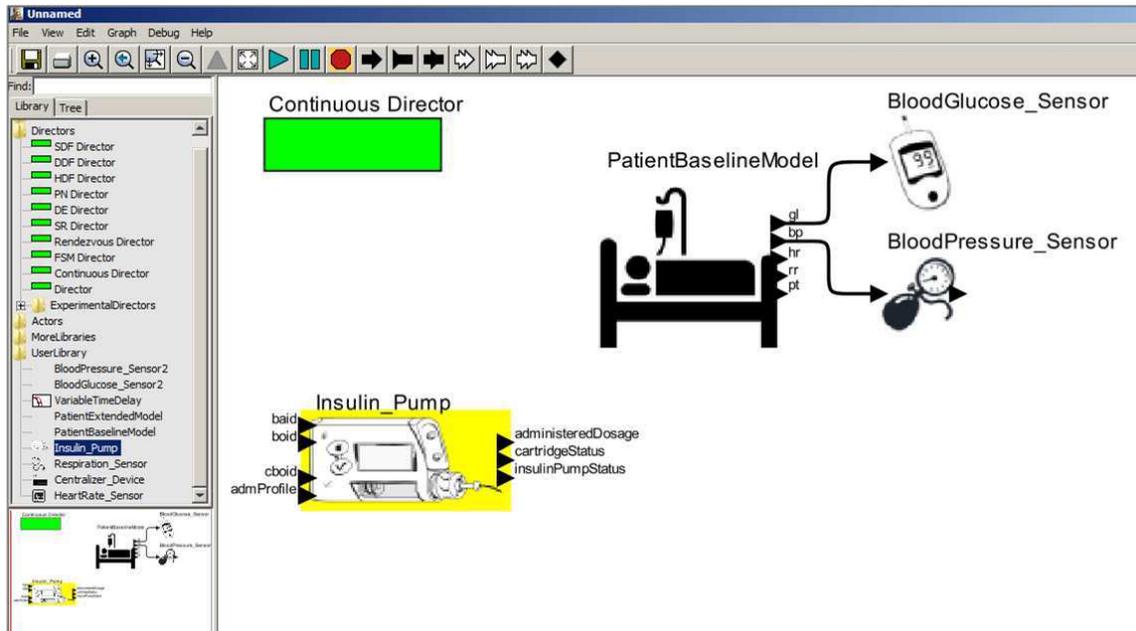


Figura 4.5: Definindo um Modelo de SMFC no *Ptolemy II*.

## 4.2 Definindo os Cenários Clínicos de Interesse

Os desenvolvedores devem definir os cenários clínicos, que funcionam como *casos de simulação* utilizados para analisar comportamentos do sistema [80]. Um *Cenário Clínico* pode ser usado para simular ambientes clínicos integrados (do inglês, *Integrated Clinical Environments* – ICE), onde monitoramento, diagnóstico e tratamento de um paciente são realizados. Um ICE combina dispositivos médicos heterogêneos interoperáveis e outros equipamentos para construir um sistema médico para tratar um paciente de alta acuidade [6]. Exemplos de ICE são salas operatórias, unidades de terapia intensiva, hospitais, ambulâncias e outros ambientes de cuidado agudo de pacientes.

Os benefícios de definir cenários clínicos independentemente dos dispositivos médicos incluem: (i) a capacidade de combinar os dados coletados a partir das diferentes fontes para produzir novas informações; (ii) o suporte à decisão clínica; e (iii) a capacidade de implementar controle distribuído de dispositivos médicos para travas de segurança e controle em malha-fechada. Dessa forma, três exemplos de modelos de cenários clínicos são apresentados na Seção 5.1.

### 4.3 Executando Atividades de Validação

As atividades de validação consistem na definição de cenários clínicos (i.e., casos de teste), que servirão para analisar o comportamento do modelo de SMFC (Seção 5.1), bem como também do mapeamento dos requisitos funcionais e de segurança do SMFC para uma notação formal (Capítulo 6). Tal mapeamento visa transformar esses requisitos em propriedades formais que devem ser mantidas para os diferentes cenários clínicos. Adicionalmente, suposições acerca do modelo de SMFC podem ser feitas para simplificar aspectos do modelo e restringir valores de entrada para o propósito de simulação.

Com a simulação de diferentes cenários clínicos é possível verificar como o sistema reage às várias condições de entrada, e quanto da estrutura do modelo é coberta a partir dessas entradas. Assim, técnicas de cobertura de modelos podem ser utilizadas para identificar caminhos de simulações inutilizados ou não executados no modelo.

### 4.4 Considerações Finais

Neste capítulo foi detalhado como fazer uso dos modelos formais de pacientes e dispositivos médicos disponíveis nas bibliotecas de atores para construir e simular modelos de *Sistemas Médicos Físico-Cibernéticos* (SMFC) de acordo com os requisitos do sistema a ser desenvolvido.

Uma visão geral da abordagem baseada em modelos para suporte à validação de SMFC foi apresentada. A abordagem consiste de: (1) projetar a arquitetura do SMFC, incluindo a seleção de modelos para pacientes (1.1) e dispositivos (1.2) e a construção do controlador do SMFC (1.3); (2) definir os cenários clínicos de interesse para simulação; e (3) realizar a simulação e a análise baseada em modelos.

O projeto da arquitetura descrita na abordagem foi idealizada a partir do Modelo Funcional Conceitual de Ambiente Clínico Integrado (ICE), descrito na Seção 2.2 do Capítulo 2. De acordo com o modelo ICE, um ambiente clínico integrado deve ser composto por elementos que desempenham diferentes funções em um ambiente clínico, mas quando integrados visam alcançar um objetivo único, que é melhorar a segurança de um paciente quando em tratamento por um sistema médico. Com requisitos bem definidos

para cada um de seus elementos, o modelo ICE permite que cada fabricante configure seu ambiente e implemente suas funções, conforme as necessidades do sistema médico. De forma semelhante, manteve-se essa mesma ideia na arquitetura proposta, possibilitando que os modelos de cenários clínicos virtuais fossem construídos com componentes atores flexíveis e com finalidades próprias, mas com interfaces bem definidas.

Com isso, conseguiu-se contemplar na solução proposta alguns dos principais aspectos inerentes a um SMFC, incluindo a concorrência entre os seus elementos, a separação de papéis funcionais para cada um desses elementos e as diferentes dinâmicas embutidas.

Por fim, com a possibilidade de se fazer reuso dos modelos pré-existentes nas bibliotecas apresentadas cria-se a capacidade de adaptá-los e/ou estendê-los para melhor satisfazer as necessidades do desenvolvedor, no momento da concepção do projeto do sistema ou dispositivo médico que se pretende desenvolver. Aliado a isso, permite-se definir os cenários clínicos para simular situações clínicas envolvendo comportamentos relacionados ao paciente, com o propósito de analisar como esses SMFC reagem mediante tais situações. Com esses cenários devidamente estruturados, permitir-se-á também, por exemplo, verificar propriedades e suposições formalizadas dos modelos desses sistemas durante a etapa de validação.

# Capítulo 5

## Validação da Abordagem Proposta

Com o objetivo de validar a abordagem proposta, um processo em dois passos foi realizado. Primeiro, a abordagem foi aplicada para três cenários clínicos diferentes para avaliar seu potencial de reuso em diferentes contextos. Segundo, uma avaliação empírica com desenvolvedores foi realizada, aplicando o método GQM (do inglês, *Goal-Question-Metric*) [9], para avaliar a capacidade de reuso e produtividade com o uso dos modelos construídos a partir da abordagem.

### 5.1 Prova de Conceito em Diferentes Cenários Clínicos

Três provas de conceito foram definidas para mostrar o uso dos modelos construídos seguindo a abordagem proposta. Cada prova de conceito representa um contexto clínico diferente, que poderia ser utilizado para apoiar as atividades de validação em um projeto de SMFC. Os modelos de SMFC construídos incluem modelos de dispositivos médicos e outros componentes funcionais, e também modelos de pacientes.

#### 5.1.1 Contexto Clínico I

Neste cenário, é descrita a situação em que pacientes são continuamente monitorados em uma unidade de terapia intensiva (UTI). Durante a estadia de um paciente na UTI, seus sinais vitais e parâmetros fisiológicos são registrados. Tais registros são utilizados

para avaliar as condições de saúde do paciente ao longo do período de tratamento.

O modelo de cenário clínico é ilustrado na Figura 5.1. Os modelos de sensores médicos (2) são conectados ao Modelo de Paciente em UTI (1) via portas. Isto permite ao desenvolvedor capturar os dados fornecidos pelo modelo de paciente. Um modelo de dispositivo centralizador (3) foi construído para permitir a comunicação entre os modelos de sensores médicos e um modelo de monitor multiparamétrico (4) de sinais. Para isso, um canal de comunicação foi definido no modelo do centralizador para cada modelo de sensor médico. O centralizador recebe os dados dos sensores médicos por meio de interfaces de comunicação sem fio, e torna-os disponíveis para o monitor multiparamétrico. No monitor multiparamétrico são exibidos os dados previamente formatados para os profissionais de saúde, auxiliando-os na tomada de decisão.

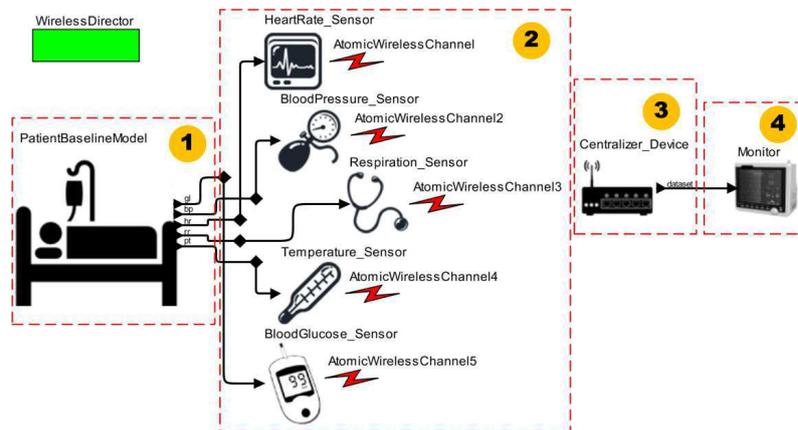


Figura 5.1: Modelo de SMFC Usando o Modelo de Paciente em UTI.

Na Figura 5.2, são ilustradas as capturas de telas com os dados da simulação após execução deste modelo de SMFC, e todos os parâmetros de configuração para o Modelo de Paciente em UTI e do próprio simulador do *Ptolemy II*. Com esse cenário foi possível demonstrar que o desenvolvedor tem a flexibilidade de simular várias condições clínicas em relação aos sinais vitais de um paciente, por utilizar o Modelo de Paciente em UTI. Quando esse modelo de paciente está acoplado aos modelos de sensores médicos, o desenvolvedor consegue monitorar os dados inerentes ao paciente e utilizá-los para estabelecer as estratégias de tratamento para o paciente. Assim, nessa prova de conceito é demonstrado como aplicar a abordagem proposta em cenários clínicos mais simples. Neste caso, os cenários não incluem modelos de atuadores.

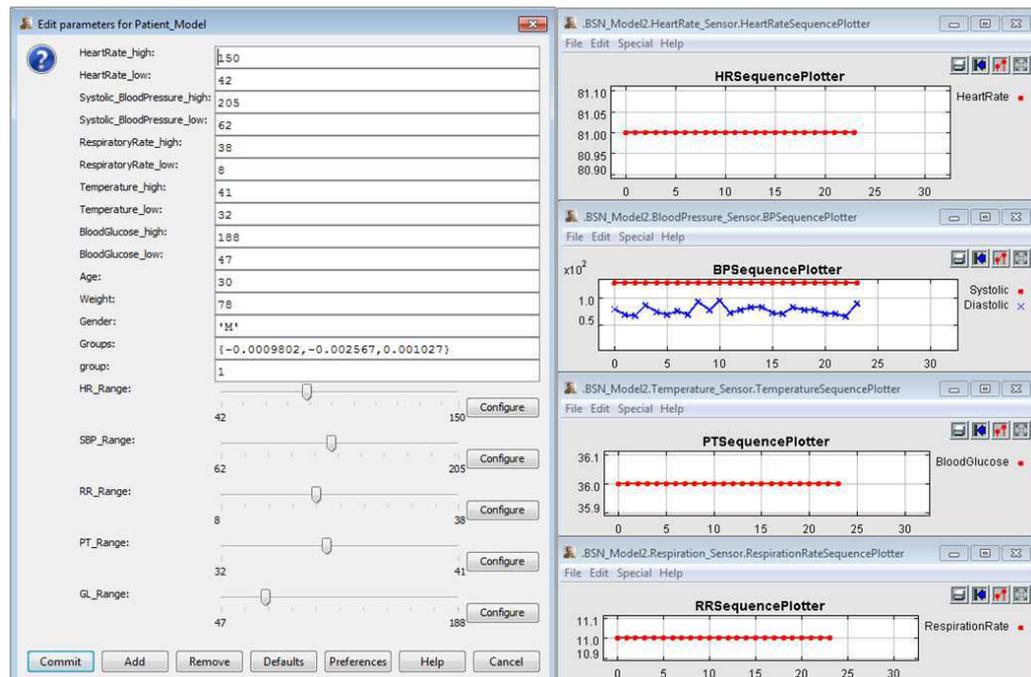


Figura 5.2: Simulação de um Modelo de SMFC Usando o Modelo de Paciente em UTI.

### 5.1.2 Contexto Clínico II

Este cenário clínico foi definido para analisar o comportamento do modelo da bomba de insulina quando utilizada por pacientes com *diabetes mellitus*. Tal cenário clínico foi escolhido por representar uma das principais preocupações da Organização Mundial de Saúde (OMS), uma vez que diabetes é uma doença crônica que afeta cerca de 347 milhões de pessoas da população mundial [96]. Ela aumenta em 22% o risco de morte por doenças cardiovasculares, sendo diretamente responsável por 3,5% de mortes neste grupo. Por fim, diabetes pode tornar-se a sétima causa líder de morte no mundo até o ano de 2030 [95].

No contexto deste cenário, o paciente necessitará configurar a bomba de insulina de acordo com a prescrição médica e seu regime de alimentação diária. A programação da bomba consiste em selecionar o perfil de administração de insulina *basal* e especificar as doses de insulina do tipo *bolus* padrão e *bolus* corretivo a serem administradas em tempos específicos do dia. Em se tratando do *bolus* corretivo, os pacientes acionam um botão na bomba de modo que um “tiro curto” de insulina é instantaneamente injetado em seu corpo.



### 5.1.3 Contexto Clínico III

Neste cenário, a farmacocinética da insulina foi utilizada para demonstrar como usar o Modelo de Paciente Diabético. Há duas estratégias distintas de uso desse modelo de paciente neste contexto clínico. A primeira é controlar o nível de glicose sanguínea como um sistema em malha-fechada, em que um controlador proporcional-integral-derivativo (PID) é projetado para esse propósito e embarcado no modelo de paciente (intrusiva). Esse processo é apresentado na Seção 3.1. A forma alternativa é controlar o nível de glicose por meio do modelo da bomba de insulina acoplada ao modelo de paciente (não-intrusiva), como um sistema em malha-aberta.

Na Figura 5.4, o modelo de SMFC é ilustrado para a primeira estratégia (i.e., sistema em malha-fechada). No modelo deste cenário observa-se apenas a presença de modelos de componentes funcionais (1) para gerar as perturbações para o modelo de paciente (2).

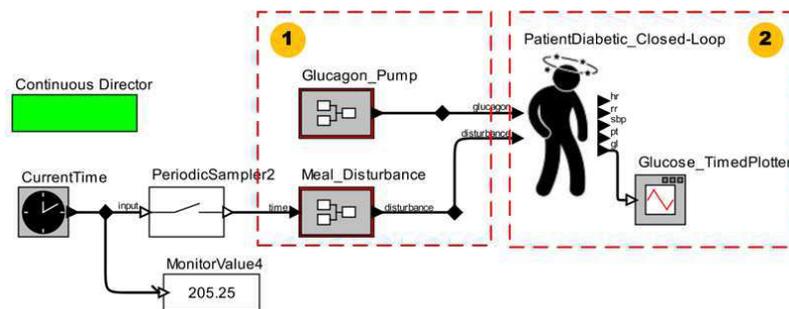


Figura 5.4: Modelo de SMFC Usando um Modelo de Paciente Diabético.

Os resultados da simulação deste modelo de cenário clínico é ilustrado na Figura 5.5. O comportamento do modelo de paciente representa a mesma resposta em relação à glicose se o modelo de paciente recebe perturbações. Portanto, esse modelo de paciente reage como demonstrado no término da construção do Modelo de Paciente Diabético. Assim, durante a simulação deste cenário notou-se que a estratégia de controle de glicose induz os modelos de regressão a ajustar os respectivos valores para cada sinal vital representado no modelo de paciente. Isto ocorre porque tais modelos de regressão utilizam a glicose (parâmetro fisiológico) como uma variável preditora.

Neste cenário, os valores para os sinais vitais podem ser alterados durante uma simulação, não apenas pelos desenvolvedores que querem simular situações anormais

em relação aos sinais vitais, mas também em consequência das ações do controlador PID, que age para controlar o nível de glicose em resposta às perturbações.



Figura 5.5: Simulação de um Modelo de SMFC Usando o Modelo de Paciente Diabético.

## 5.2 Avaliação Empírica

Para avaliar se a abordagem proposta promove o reúso e a produtividade, um estudo empírico foi realizado. O estudo considerou o ponto de vista dos desenvolvedores de SMFC. Para este propósito, a biblioteca de pacientes e dispositivos médicos foi avaliada por quatorze desenvolvedores. A avaliação foi dividida em duas fases: *treinamento* e *execução*. Durante a fase de *treinamento*, o pesquisador preparou os desenvolvedores para construir modelos de SMFC usando o *Ptolemy II*. Durante a fase de *execução*, os desenvolvedores foram divididos em dois grupos: *controle* e *tratamento*. O grupo de *controle* construiu dois modelos de SMFC com apenas um pequeno subconjunto de modelos disponíveis nas bibliotecas de pacientes e dispositivos médicos. O grupo de *tratamento* construiu os mesmos modelos de SMFC com todos os modelos disponíveis nas bibliotecas supracitadas.

A produtividade foi avaliada por aplicar o teste de hipótese *t-test* com o tempo médio que cada grupo levou para finalizar a construção do modelo referente a cada SMFC. Em seguida, os desenvolvedores responderam um questionário em relação aos

fatores de reúso dos modelos: *compreensibilidade, adaptabilidade e portabilidade*.

### 5.2.1 Escopo

Os objetivos da avaliação empírica, usando o modelo do método *Goal-Question-Metric* (GQM), apresentado por Basili e Rombach [9], são: (i) analisar *o uso das bibliotecas de modelos de pacientes e dispositivos médicos* para o propósito de *avaliação* com respeito à *produtividade* a partir do ponto de vista do *desenvolvedor* no contexto de *desenvolvedores construir modelos de SMFC*; e (ii) analisar *o uso das bibliotecas de modelos de pacientes e dispositivos médicos* para o propósito de *avaliação* com respeito ao *reúso* a partir do ponto de vista do *desenvolvedor* no contexto de *desenvolvedores construir os modelos de SMFC*.

As seguintes questões de pesquisa foram abordadas:

**RQ1:** Utilizar as bibliotecas de modelos de pacientes e dispositivos médicos aumentam a produtividade dos desenvolvedores?

**RQ2:** As bibliotecas de modelos são reutilizáveis?

Dadas as questões de pesquisa, as seguintes hipóteses foram definidas:

**H0-1:** A produtividade não é aumentada.

**HA-1:** A produtividade é aumentada.

**H0-2:** Os modelos não são reutilizáveis.

**HA-2:** Os modelos são reutilizáveis.

Em que, **H0-1** e **HA-1** estão relacionadas à **RQ1** e **H0-2** e **HA-2** à **RQ2**.

### 5.2.2 Objetos de Estudo

Segundo Kitchenham, Pickard e Pfleeger [44], objetos de estudo podem ser os programas, algoritmos ou problemas para os quais métodos ou ferramentas são aplicados em um determinado experimento. Nesta pesquisa, dois problemas foram definidos

para analisar o uso das bibliotecas de modelos de pacientes e dispositivos médicos: (i) construir o modelo de SMFC para o cenário clínico apresentado na Seção 5.1.1, em que pacientes são monitorados continuamente em uma unidade de terapia intensiva (UTI); e (ii) validar o comportamento do modelo da bomba de insulina apresentado na Seção 5.1.2.

### 5.2.3 Sujeitos

A técnica de amostragem por conveniência foi utilizada. A avaliação empírica foi executada em uma empresa brasileira que foi escolhida dada a relação universidade-empresa existente. Como sujeitos, a empresa forneceu um grupo de quatorze desenvolvedores. O perfil dos desenvolvedores é apresentado na Tabela 5.1.

Questões	Desenvolvedores													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>Idade</i>	21	25	20	21	30	21	24	27	21	21	20	25	22	23
<i>Conhecimento sobre Métodos Formais?<sup>b</sup></i>	S	S	N	N	S	S	N	S	N	S	S	S	S	S
<i>Opinião sobre a Etapa de Treinamento?<sup>a</sup></i>	ÓTIMO	REGULAR	BOM	ÓTIMO	BOM	ÓTIMO	ÓTIMO	ÓTIMO	ÓTIMO	ÓTIMO	BOM	BOM	BOM	ÓTIMO
<i>Resolveu a lista de exercícios?<sup>a</sup></i>	S	S	S	S	N	S	S	S	S	S	S	N	S	N
<i>Conhecimento sobre o Ptolemy II?<sup>a</sup></i>	BOM	BOM	BOM	REGULAR	POUCO	BOM	REGULAR	REGULAR	BOM	BOM	REGULAR	REGULAR	BOM	BOM
<i>Conhecimento sobre o trabalho?<sup>a</sup></i>	BOM	REGULAR	BOM	REGULAR	POUCO	BOM	REGULAR	REGULAR	BOM	BOM	REGULAR	BOM	BOM	BOM

<sup>a</sup>Após a Etapa de Treinamento; <sup>b</sup>Antes da Etapa de Treinamento; S: Sim; N: Não.

Tabela 5.1: Respostas ao Questionário para Identificar o Perfil do Desenvolvedor.

### 5.2.4 Variáveis e Tratamento

Dados os objetivos da pesquisa, duas variáveis dependentes (i.e., resposta) são definidas: *produtividade* do desenvolvedor e *reúso* dos modelos. Além disso, quatro variáveis independentes (i.e., fatores) são identificadas: as bibliotecas de modelos de pacientes e

dispositivos médicos, a experiência individual dos participantes (i.e., sujeitos) da pesquisa, a ferramenta de apoio à modelagem e o ambiente computacional. Os últimos três fatores são controlados em um nível fixo. Todos os sujeitos possuem experiências semelhantes e dispõem da mesma ferramenta de apoio à modelagem e ambiente computacional para realizarem o experimento. Portanto, o fator de tratamento são *as bibliotecas de modelos de pacientes e dispositivos médicos*.

### 5.2.5 Procedimento

Após o *Ptolemy II* ter sido instalado e o ambiente computacional para cada desenvolvedor ter sido configurado, o estudo foi realizado em duas fases: *treinamento* e *execução*. Essas fases são descritas a seguir.

#### Fase de Treinamento

Durante essa fase, os sujeitos foram preparados para construir os modelos de SMFC no *Ptolemy II* usando as bibliotecas de pacientes e de dispositivos médicos. As seguintes atividades foram planejadas:

- **Questionário Pessoal** - desenvolvedores respondem um questionário em relação às informações pessoais, experiência com métodos formais e com componentes reutilizáveis. Este questionário encontra-se disponível no Apêndice D.
- **Oficina de Aprendizagem** - o pesquisador realiza um curso introdutório em que os conceitos em relação aos SMFC, *Ptolemy II* e a abordagem proposta são apresentados. Além disso, exemplos práticos de modelos de pacientes e dispositivos no *Ptolemy II* são demonstrados.
- **Aprendizagem Prática Autônoma** - (i.e., aprender fazendo) com a ajuda assistida pelo pesquisador. Os sujeitos aplicaram as técnicas aprendidas para construir elementos simples no *Ptolemy II*, tais como um contador incremental e um sensor de sinal senoidal.

O material de treinamento (e.g., *slides* e arquivos dos exemplos do *Ptolemy II*) foram preparados pelo pesquisador. Durante o curso introdutório, apresentações áudio-visuais foram utilizadas. Para apoiar as atividades de aprendizagem prática que foram realizadas pelos sujeitos da pesquisa, o método de resolução de problemas foi adotado.

### Fase de Execução

Durante essa fase, os sujeitos utilizaram as bibliotecas de pacientes e dispositivos médicos para construir os modelos de SMFC no *Ptolemy II*. Ao término dessa fase, eles responderam a outro questionário em relação aos fatores de reuso dos modelos. As seguintes atividades foram planejadas:

- **Modelagem de SMFC** - os desenvolvedores foram divididos em dois grupos balanceados (i.e., mesmo tamanho): *controle* e *tratamento*. Cada grupo foi composto de sete sujeitos selecionados aleatoriamente. O grupo de controle resolveu os dois problemas apresentados na Seção 5.2.2, usando apenas um pequeno subconjunto de modelos das bibliotecas de pacientes (e.g., *Patient\_icu*) e dispositivos médicos (e.g., *HeartRate*, *Respiration*, *BloodGlucose*, *CalculatorID* e *Insulin\_Pump*). Para este grupo, os modelos *Patient\_icu* e *Insulin\_Pump* estavam parcialmente prontos para o reuso, enquanto outros modelos de dispositivos necessários para a resolução dos problemas precisavam ser totalmente construídos (e.g., *BloodPressure*, *Temperature*, *Centralizer* e *Bedside\_Monitor*). O grupo de tratamento resolveu os mesmos problemas, mas diferente do grupo de controle, este grupo utilizou todos os modelos das bibliotecas, com exceção do modelo do dispositivo centralizador. Vale ressaltar que para este grupo, todos os modelos disponíveis estavam totalmente prontos para reuso, sendo necessário apenas construir por completo o modelo *Centralizer*. Portanto, o experimento foi conduzido como um “estudo bloqueado sujeito-objeto” [97]. Para ambos os grupos, uma diretriz de execução foi fornecida para guiá-los na modelagem dos SMFC referentes aos problemas abordados.

- **Questionário de Reúso** - os desenvolvedores responderam ao questionário em relação aos fatores de reúso dos modelos: *compreensibilidade*, *adaptabilidade* e *portabilidade*. Este questionário encontra-se disponível no Apêndice E.

O material da execução (e.g., arquivos do *Ptolemy II* e diretrizes) foi também preparado pelo pesquisador. Durante a modelagem dos SMFC, o pesquisador estava disponível para auxiliar os desenvolvedores com quaisquer dificuldades de entendimento nas definições dos cenários clínicos e respostas do questionário.

### 5.2.6 Medidas

Para verificar as hipóteses e avaliar as questões e os objetivos da pesquisa, algumas métricas foram coletadas. Além disso, por definir as métricas, as hipóteses apresentadas na Seção 5.2.1 foram formalizadas para estatisticamente testá-las. Como declarado na Seção 5.2.4, as duas variáveis dependentes identificadas foram: a *produtividade* do desenvolvedor e o *reúso* dos modelos.

**Medição da Produtividade** - de acordo com Kitchenham, Pickard e Pfleeger [44], para medir a produtividade, é necessário tratar de esforço e tempo. Na atividade de *desenvolvimento de um SMFC*, os grupos de controle e tratamento construíram os modelos de SMFC para os mesmos cenários clínicos. A diferença foi que para o grupo de tratamento foram disponibilizados para reúso mais modelos das bibliotecas. Portanto, apenas foi medido o tempo requerido por cada grupo para finalizar os problemas. Sempre que cada sujeito finalizasse um modelo de SMFC, o tempo requerido para concluir a modelagem era coletado. Portanto, para cada problema, uma hipótese nula e uma alternativa foi definida formalmente:

**H0-1-1**  $\mu_{C_1} \leq \mu_{T_1}$ , em que  $\mu_{C_1}$  é, para o primeiro problema, o tempo médio requerido (em minutos) pelo grupo de controle para finalizá-lo, e  $\mu_{T_1}$ , pelo grupo de tratamento.

**HA-1-1**  $\mu_{C_1} > \mu_{T_1}$ .

**H0-1-2**  $\mu_{C_2} \leq \mu_{T_2}$ , em que  $\mu_{C_2}$  é, para o segundo problema, o tempo médio

requerido (em minutos) pelo grupo de controle para finalizá-lo, e  $\mu_{T_2}$ , pelo grupo de tratamento.

$$\mathbf{HA-1-2} \quad \mu_{C_2} > \mu_{T_2}.$$

Essas hipóteses foram verificadas usando o teste de hipótese *t-test* para avaliar **RQ1**.

**Medição do Reúso** - de acordo com Washizaki, Yamamoto e Fukazawa [92], o reúso de um componente pode ser decomposto em três fatores: *compreensibilidade*, *adaptabilidade* e *portabilidade*. *Compreensibilidade* é o esforço necessário por um usuário para reconhecer o conceito por trás de um componente e sua aplicabilidade. *Adaptabilidade* é a facilidade com que um componente pode ser adaptado para preencher um requisito que difere daquele para o qual ele foi originalmente desenvolvido. *Portabilidade* é a facilidade com que um software pode ser transferido de um ambiente para outro. Como em Washizaki, Yamamoto e Fukazawa [92], deseja-se neste experimento uma avaliação de caixa-preta dos componentes a partir do ponto de vista dos usuários. Portanto, os critérios de reúso bem conhecidos, tais como coesão e acoplamento, foram descartados, e focou-se nos três fatores previamente descritos.

Para coletar as métricas em relação a esses fatores, um questionário foi criado e, para cada questão, foi fornecida uma *escala Likert* de 5-pontos do pior (1) ao melhor (5). Por exemplo, para *compreensibilidade*, o questionário possui a questão “Como você avalia o esforço necessário para reconhecer o conceito por trás de um componente e sua aplicabilidade?”, em que a resposta 5 representa “Muito pouco esforço”. Ao término da atividade de *desenvolvimento de modelos de SMFC*, os sujeitos responderam o questionário. Portanto, para cada fator, uma hipótese nula e uma alternativa foi definida:

**H0-2-1**  $v_U \leq 3$ , em que  $v_U$  representa o ranque médio das respostas para a questão em relação à *compreensibilidade*.

$$\mathbf{HA-2-1} \quad v_U > 3.$$

**H0-2-2**  $v_A \leq 3$ , em que  $v_A$  representa o ranque médio das respostas para a questão

em relação à *adaptabilidade*.

**HA-2-2**  $v_A > 3$ .

**H0-2-3**  $v_P \leq 3$ , em que  $v_P$  representa o ranque médio das respostas para a questão em relação à *portabilidade*.

**HA-2-3**  $v_P > 3$ .

Essas hipóteses foram avaliadas usando o teste de hipótese *Wilcoxon* para avaliar **RQ2**.

### 5.2.7 Análise

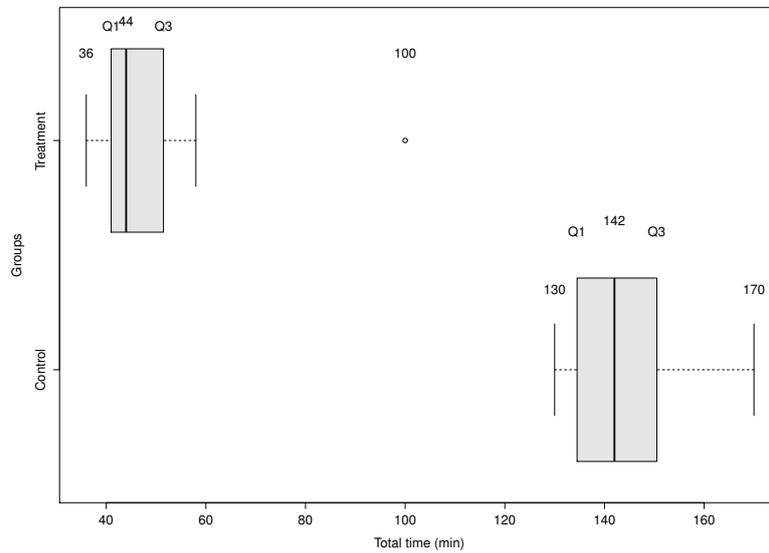
**RQ1** Utilizar as bibliotecas de modelos de pacientes e dispositivos médicos aumenta a produtividade dos desenvolvedores?

Para avaliar essa questão de pesquisa, **H0-1-1** e **H0-1-2** foram verificadas usando o teste de hipótese *t-test* com intervalo de confiança de 95% para a diferença entre a média dos dois grupos em estudo. Na Figura 5.6, para cada problema, o tempo coletado para o grupo de controle e o grupo de tratamento é mostrado.

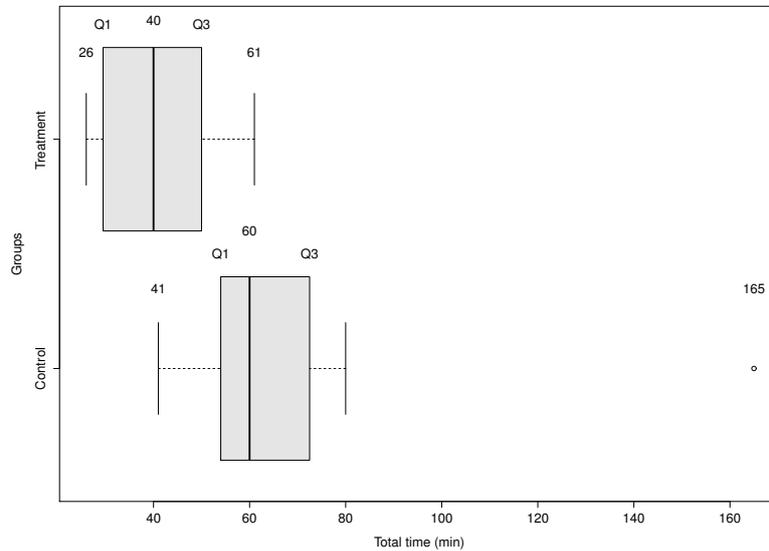
Para **H0-1-1**, o *p-value* calculado foi igual a  $1,41e-06$ , sendo, portanto, a hipótese nula refutada. Isso significa que o grupo de controle teve uma média estatisticamente maior do que a média do grupo de tratamento (no nível de significância de 0,05). Na prática, o grupo de tratamento economizou muito tempo (i.e., aproximadamente três vezes menos) para modelar o Cenário Clínico I em relação ao grupo de controle.

Para **H0-1-2**, calculou-se um *p-value* de 0,04181, refutando-se, assim, a hipótese nula. Neste caso, o grupo de tratamento economizou metade do tempo requerido para modelar o Cenário Clínico II comparado ao grupo de controle. Essa redução na diferença entre as médias dos dois grupos ocorreu devido ao número reduzido de passos necessários para modelar este segundo cenário.

Portanto, é possível concluir que o reuso dos modelos de pacientes e dispositivos disponíveis nas bibliotecas de modelos desenvolvidas a partir da abordagem proposta aumenta a produtividade dos desenvolvedores durante a construção de modelos de SMFC para validação desses sistemas.



(a) Primeiro Problema – Contexto Clínico I.



(b) Segundo Problema – Contexto Clínico II.

Figura 5.6: *Boxplots* dos Tempos Coletados para os Grupos de Controle e de Tratamento. (a) Primeiro Problema; (b) Segundo Problema.

**RQ2** Os modelos construídos a partir da abordagem proposta são reutilizáveis?

Para avaliar essa questão de pesquisa, **H0-2-1**, **H0-2-2** e **H0-2-3** foram verificadas usando o teste de hipótese *Wilcoxon* com intervalo de confiança de 95% para a diferença entre a resposta de todos os desenvolvedores e o parâmetro que especifica o valor usado para formar a hipótese nula, onde esse parâmetro é igual a três. Assim, tal parâmetro define o limiar para dizer se o modelo é reutilizável ou não. Na Figura 5.7, as respostas dos desenvolvedores para cada fator de reúso são apresentadas.

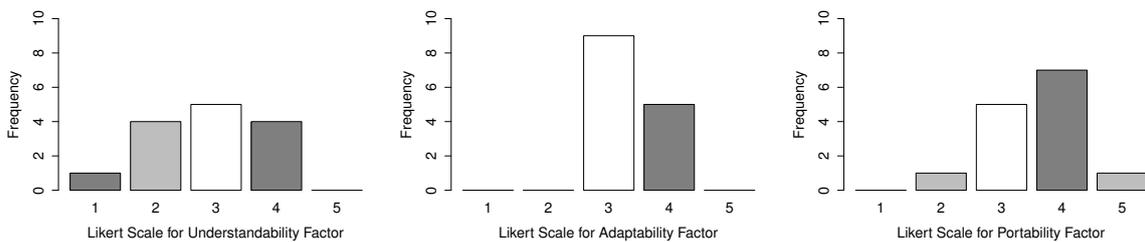


Figura 5.7: Distribuição das Respostas dos Desenvolvedores de Acordo com os Fatores de Reúso.

Para **H0-2-1**, o *p-value* obtido foi igual a 0,7394, sendo, neste caso, a hipótese nula aceita. Uma vez que a maioria dos desenvolvedores opinaram que a fase de treinamento foi boa ou ótima (i.e., cerca de 93% dos desenvolvedores) com um bom conhecimento adquirido sobre o trabalho (i.e., cerca de 60% dos desenvolvedores), a razão para explicar o resultado do teste dessa hipótese pode ser a pouca experiência dos desenvolvedores com a ferramenta *Ptolemy II*. Isto significa que a lista de exercícios praticada pelos desenvolvedores (i.e., cerca de 80%) durante a fase de treinamento foi insuficiente para os sujeitos da pesquisa modelarem os cenários clínicos solicitados na fase de execução.

Para **H0-2-2**, o *p-value* calculado foi de 0,01844. Logo, a hipótese nula foi refutada. Na prática, é possível dizer que os modelos de pacientes e dispositivos disponíveis nas bibliotecas de modelos são facilmente adaptáveis para preencher requisitos de outros contextos clínicos.

Para **H0-2-3**, um *p-value* de 0,01238 foi calculado, sendo a hipótese nula refutada. Assim, é possível afirmar que os modelos de pacientes e dispositivos disponíveis nas bibliotecas de modelos são portáveis de um cenário clínico específico para outro.

Apesar do problema relatado na verificação da hipótese **H0-2-1**, conclui-se, no

nível de significância de 0,05, que os modelos de pacientes e dispositivos disponíveis nas bibliotecas de modelos são reutilizáveis para apoiar os desenvolvedores na validação de SMFC.

### 5.2.8 Ameaças à Validade

A validade da conclusão destina-se em analisar o relacionamento entre o tratamento e os resultados. A maioria das ameaças à validade da conclusão foram mitigadas, tais como a heterogeneidade dos sujeitos participantes da pesquisa, a confiabilidade na implementação do experimento e a confiabilidade das medidas. Por outro lado, o tamanho da amostra de sujeitos (quatorze sujeitos divididos em dois grupos de sete) foi pequena. Por isso, utilizou-se o teste de hipótese *t-test* com 0,05 de significância.

A validade interna é motivo de preocupação quando os relacionamentos causais são examinados. No caso deste trabalho, as questões estão relacionadas ao *Ptolemy II*. A ferramenta possui algumas deficiências e defeitos relacionados ao salvamento dos modelos e execução das simulações que podem ter influenciados no tempo requerido pelos sujeitos para definir os modelos dos SMFC nos contextos dos cenários clínicos representados. Por exemplo, durante as ações de arrastar e soltar (i.e., *drag and drop*) de um modelo a partir da biblioteca de modelos do usuário para o editor gráfico do *Ptolemy II*, sua estrutura interna era embaralhada, impossibilitando os desenvolvedores de compreenderem os modelos. Neste caso, o desenvolvedor tinha que permanecer tentando realizar essa ação até que esse comportamento indesejado da ferramenta de modelagem não ocorresse.

Com a validade externa analisa-se até que ponto é possível generalizar os resultados e até que ponto os resultados são do interesse para outras pessoas de fora do contexto investigado. O estudo empírico foi realizado com sujeitos de uma empresa e com pouca ou nenhuma experiência na construção de SMFC. Dado que todos os sujeitos são de uma única empresa, os resultados não podem ser generalizados. Além disso, por um lado, ter os sujeitos do grupo de tratamento (reutilizando quase completamente a biblioteca de modelos) implementando com sucesso os cenários clínicos é um bom indicador do potencial de reuso das bibliotecas de modelos. Por outro lado, pode-se argumentar que o ganho na produtividade não seria significativo se desenvolvedores

experientes em SMFC e com o *Ptolemy II* participassem como sujeitos da pesquisa.

Por fim, na validade de construção foca-se em demonstrar evidências da relação entre a teoria e a observação. Para os objetos de estudos, como apenas dois cenários clínicos foram utilizados como problemática para os sujeitos da pesquisa executarem o experimento, a observação realizada pode ser insuficiente para representar a teoria apresentada.

### 5.3 Considerações Finais

Neste capítulo foram demonstradas algumas provas de conceitos que permitiu avaliar o potencial de reuso da abordagem proposta em diferentes contextos clínicos. Ainda no sentido de validar tal abordagem, realizou-se uma avaliação empírica com um grupo de desenvolvedores usando o método GQM (*Goal-Question-Metric*). Ainda neste capítulo foram discutidas as ameaças à validade da solução.

A partir da execução dos modelos de SMFC construídos seguindo a abordagem proposta, cada prova de conceito permitiu-se estruturar variadas configurações de SMFC usando os modelos de dispositivos médicos e simular diferentes condições de saúde de um paciente por meio de seus modelos. Todas os contextos clínicos simulados poderia ser utilizados para apoiar a etapa de validação de um SMFC em tempo de projeto.

Por meio da avaliação empírica com desenvolvedores, em que foram definidas questões de pesquisa e formalizadas hipóteses para analisar o potencial de reuso dos modelos existentes na biblioteca e a produtividade dos desenvolvedores na construção de modelos de SMFC para o propósito de validação, foi possível concluir por meio dos testes de hipóteses que a abordagem proposta reduz o esforço e o tempo requeridos pelos desenvolvedores na etapa de validação em tempo de projeto de SMFC. Isto significa afirmar que a abordagem proposta neste trabalho promove o reuso e a produtividade.

# Capítulo 6

## Análise dos Modelos

Para analisar os modelos que definem os cenários clínicos apresentados na Seção 5.1 utilizou-se o *Simulink Design Verifier* (SLDV)<sup>1</sup>. O SLDV é um componente opcional do Matlab<sup>®</sup> que permite analisar modelos dentro do ambiente *Simulink*. Ele utiliza métodos formais para identificar erros de projeto difíceis de encontrar nos modelos, incluindo lógica morta (*dead logic*), estouro de inteiro (*integer overflow*), divisão por zero e violações de propriedades de projeto e asserções. Como resultado, é possível realizar verificação e validação durante todo o processo de projeto. Assim, análise de modelo com o SLDV complementa a etapa de simulação por permitir aos resultados da simulação serem utilizados como entrada para análise com métodos formais [57].

Neste trabalho, demonstra-se o processo aplicado para verificar o modelo do software da bomba de insulina apresentado na Seção 3.2 e utilizado no cenário do segundo contexto clínico, como descrito na Seção 5.1.2. Dado que o ambiente *Ptolemy II* não suporta verificação formal, para realizar análise de segurança do modelo do software da bomba de insulina foi construído o modelo do SMFC no *Simulink* (ver Figura 6.1).

Em seguida, as propriedades básicas do sistema foram verificadas. Com esse propósito, os requisitos de segurança foram formalizados como propriedades de segurança. Propriedades de segurança são comportamentos indesejados que nunca deveriam ocorrer ou requisitos funcionais que devem ser satisfeitos.

---

<sup>1</sup><http://www.mathworks.com/products/sldesignverifier/>

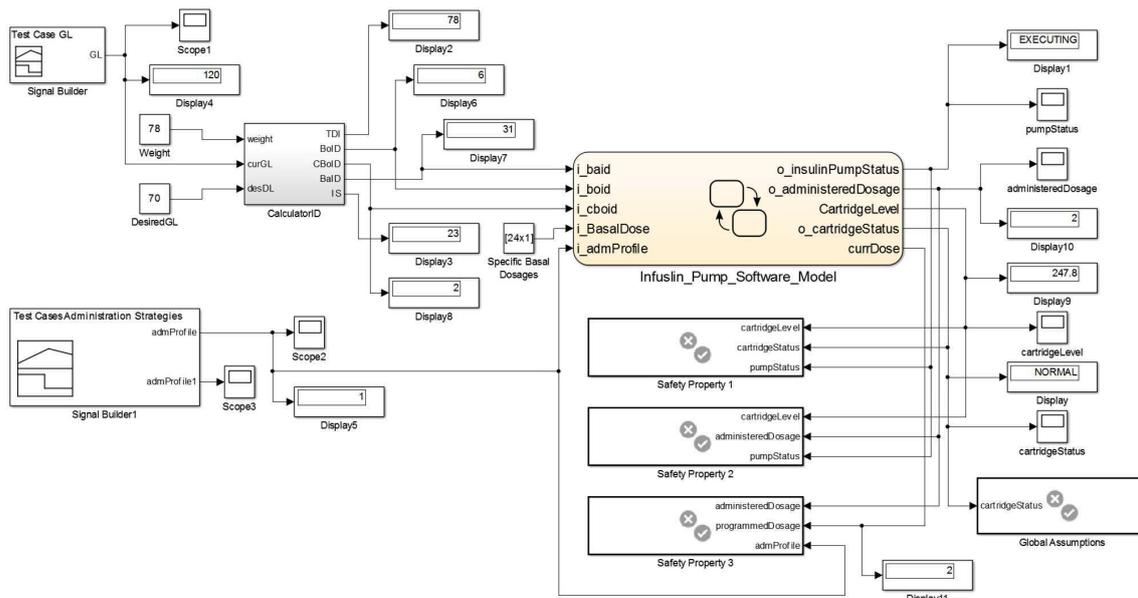


Figura 6.1: Modelo de SMFC Usando o Modelo da Bomba de Insulina no *Simulink*.

Por exemplo, para o modelo da bomba de insulina, os seguintes três requisitos de segurança foram definidos:

*Requisito de Segurança 1 (RS1):* se o nível do cartucho for igual a 0, então o estado do cartucho deve tornar-se "EMPTY" e após um atraso o estado da bomba deve ser "STOP". A formalização desta propriedade de vivacidade do sistema da bomba de insulina é ilustrada na Figura 6.2.

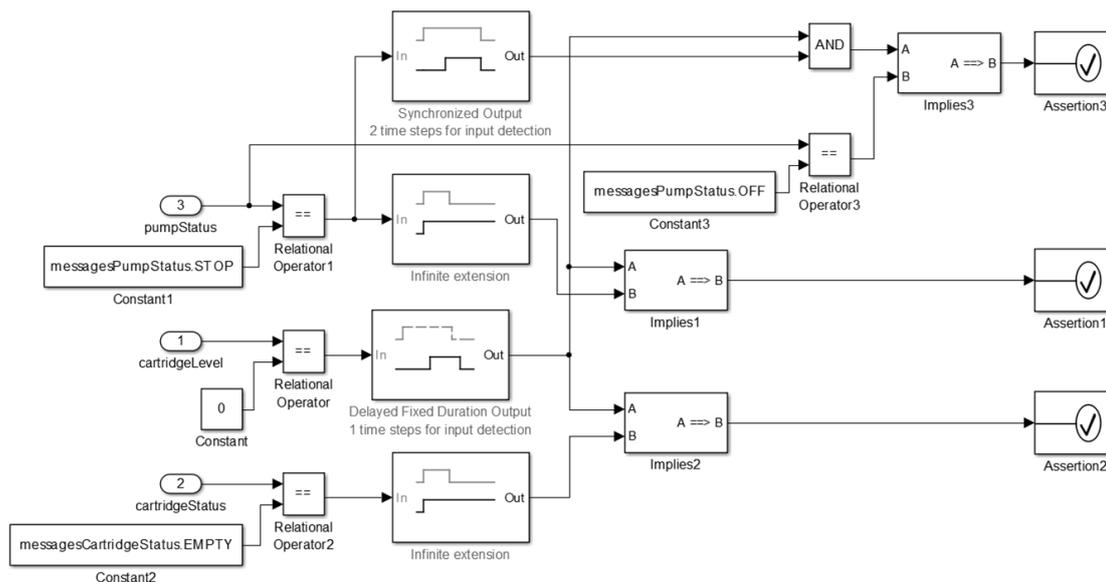


Figura 6.2: Propriedade de Segurança 1.

*Requisito de Segurança 2 (RS2):* se o nível do cartucho for menor do que a dosagem de insulina administrada, então o estado da bomba deve ser "STOP". A formalização desta propriedade é ilustrada na Figura 6.3.

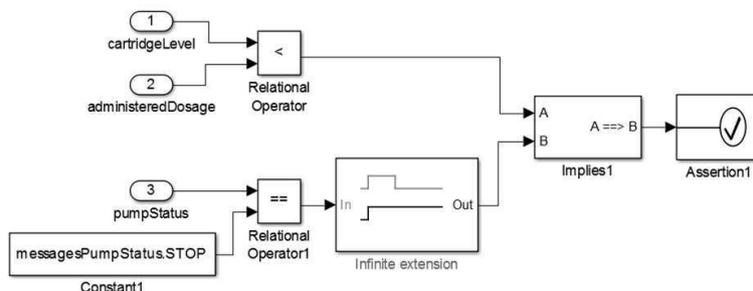


Figura 6.3: Propriedade de Segurança 2.

*Requisito de Segurança 3 (RS3):* sempre que o perfil de administração de insulina tornar-se 1 ("SPEC\_BASAL"), a dosagem de insulina administrada deve ser igual a dosagem programada. A formalização desta propriedade é ilustrada na Figura 6.4.

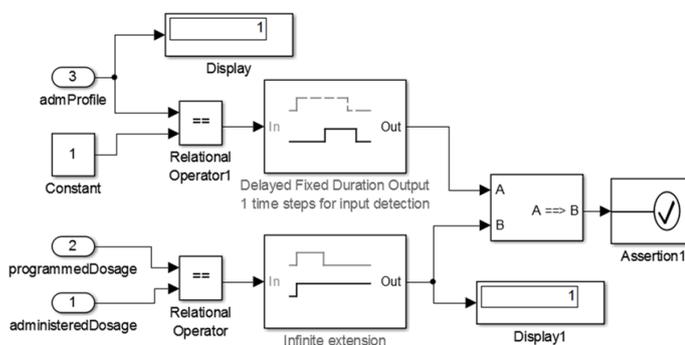


Figura 6.4: Propriedade de Segurança 3.

O comportamento do software da bomba de insulina foi representado com um diagrama do Stateflow<sup>2</sup> utilizado em conjunto com o *Simulink*, como ilustrado na Figura 6.5. O Stateflow é um ambiente para modelagem e simulação de lógica de decisão combinatória e sequencial baseado em máquinas de estado e diagramas de fluxos [58].

<sup>2</sup><http://www.mathworks.com/products/stateflow/>

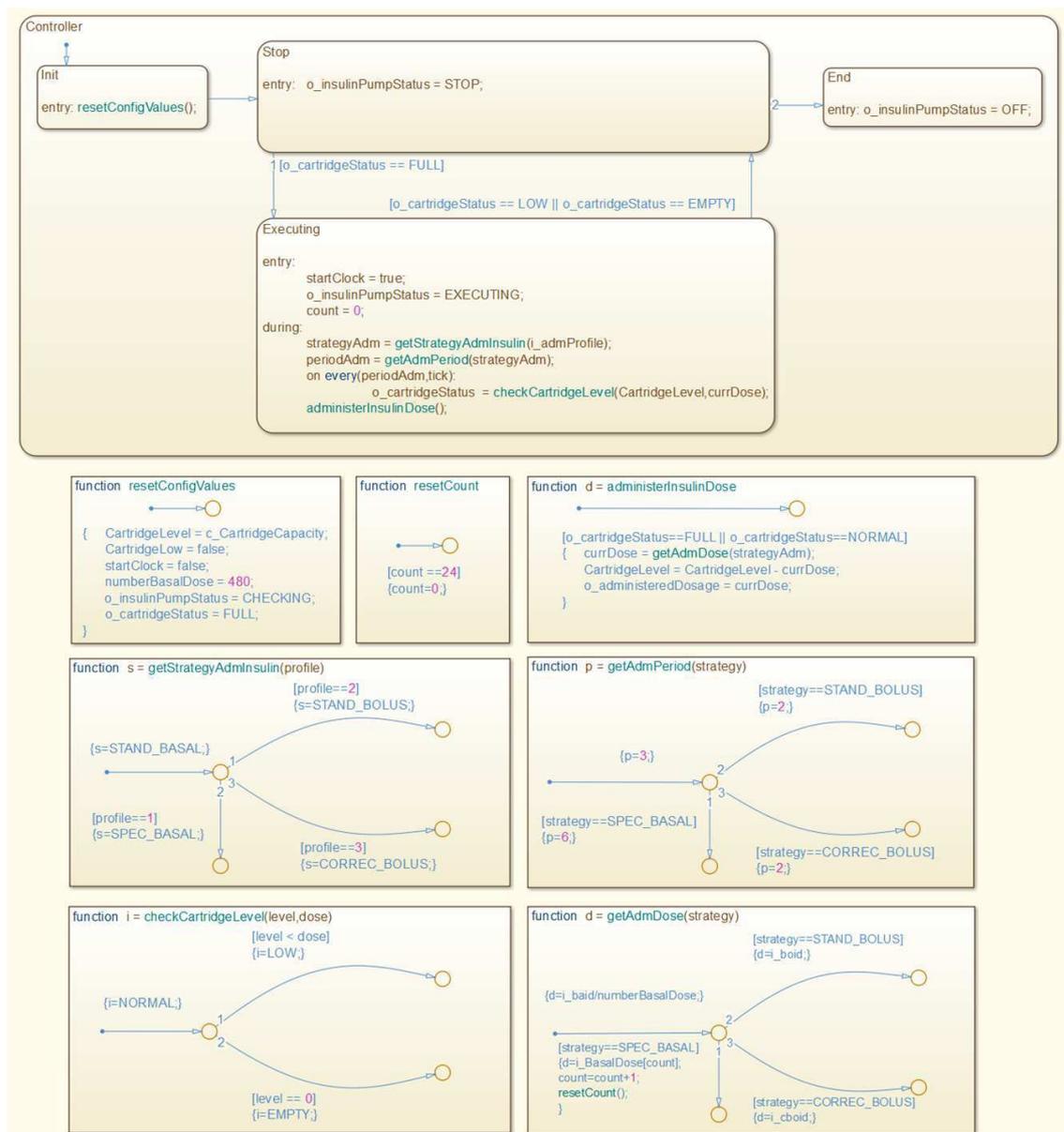


Figura 6.5: Modelo do Software da Bomba de Insulina no *Simulink*.

Uma restrição no modelo do software da bomba de insulina foi definida com a seguinte suposição (S1): o estado do cartucho nunca deve ser "EMPTY" e "LOW" simultaneamente. Essa suposição foi formalmente modelada, como ilustrado na Figura 6.6.

Para provar SP1, é necessário satisfazer as três provas objetivo que foram especificadas como asserções. Para provar SP2, SP3 e A1, é necessário satisfazer uma única prova objetivo. O SLDV foi utilizado para validar todos os requisitos e restrições. O tempo médio decorrido para a análise no MATLAB *R2013a* foi 160 segundos, usando um PC

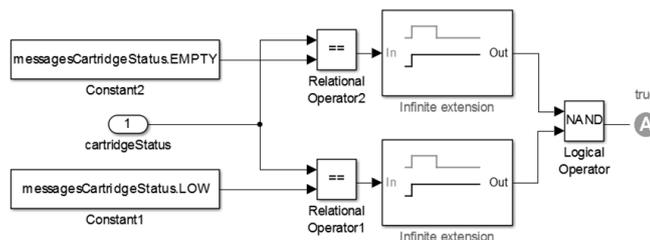


Figura 6.6: Suposição Sobre o Estado do Cartucho da Bomba de Insulina.

Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz com 6GB RAM e Sistema Operacional Windows 7.

A cobertura do modelo do software da bomba de insulina foi analisado usando o SLDV, que acumula informações acerca da ativação dos objetivos de verificação durante simulação e fornece resultados de cobertura [57]. A partir dos modelos *Simulink* e *Stateflow*, o SLDV gera casos de teste e parâmetros requeridos pelos padrões da indústria para desenvolver sistemas de alta integridade. Os critérios de cobertura estrutural utilizados nesta análise foram cobertura de decisão (DC), cobertura de condição (CC) e cobertura de decisão/condição modificado (MCDC) [88]. Para DC, o conjunto de testes (i.e., *test suite*) deve possuir testes que verifiquem cada decisão alcançável como verdadeiro e falso. Assim, o DC analisa os elementos que representam pontos de decisão em um modelo, tais como um bloco *switch* do *Simulink* ou estados do *Stateflow*. Para CC, o conjunto de testes deve ter testes que verifiquem cada condição no programa como verdadeiro e falso. Portanto, o CC analisa blocos cuja saída é a combinação lógica de suas entradas (e.g., o bloco de operador lógico) e transições do *Stateflow*. Por fim, o critério MCDC requer que a condição  $c$  afete independentemente o resultado da decisão  $d$ . Ela combina as capacidades de decisão e condição para determinar a extensão para que o caso de teste verifique a independência das entradas do bloco lógico e as condições de transição.

O resumo dos resultados da análise de cobertura do subsistema referente ao modelo do software da bomba de insulina é apresentado na Tabela 6.1.

Para os objetos *Controller*, *administerInsulinDose*, *getAdmPeriod*, *getStrategyAdminInsulin* e *resetCount*, alcançou-se 100% de cobertura de decisão (DC). Para os objetos *checkCartridgeLevel* e *getAdmDose* alcançou-se menos do que 100% de cobertura

Hierarquia do Modelo	Complexidade Ciclomática <sup>(1)</sup>	DC	CC	MDCC
Subsystem:Insulin Pump Software Model	22	92%	75%	50%
Chart:Insulin Pump Software Model	21	92%	75%	50%
State:Controller	7	100%	75%	50%
Function:administerInsulinDose	2	100%	75%	50%
Function:checkCartridgeLevel	2	75%	NA <sup>(2)</sup>	NA
Function:getAdmDose	3	67%	NA	NA
Function:getAdmPeriod	3	100%	NA	NA
Function:getStrategyAdmInsulin	3	100%	NA	NA
Function:resetCount	1	100%	NA	NA

<sup>(1)</sup>Medida da complexidade estrutural para cada objeto na hierarquia do modelo.

<sup>(2)</sup>NA: Não Aplicável.

Tabela 6.1: Resumo do Relatório de Cobertura para o Modelo de Software da Bomba de Insulina.

das decisões analisadas. O objeto *checkCartridgeLevel* alcançou 50% de cobertura de decisão porque a transição “[level == 0]{i=EMPTY;}” nunca foi avaliada como verdadeira durante a simulação com o caso de teste fornecido como entrada para o modelo do software da bomba de insulina. O mesmo ocorreu com as transições “[strategy==STAND\_BOLUS]{d=i\_boid;}” e “[strategy==CORREC\_BOLUS]{d=i\_cboid;}” de *getAdmDose*.

Considerando o critério de cobertura de condição (CC), alcançou-se 75% de cobertura por causa das seguintes condições nunca terem sido avaliadas como verdadeiras: (1) “o\_cartridgeStatus == EMPTY” da transição “[o\_cartridgeStatus == LOW || o\_cartridgeStatus == EMPTY]” do sub-estado “*Executing*” para o sub-estado “*Stop*”; (2) “o\_cartridgeStatus == FULL” da transição “[o\_cartridgeStatus == FULL || o\_cartridgeStatus == NORMAL]”. Ambas as transições também afetam a análise MCDC fazendo o modelo alcançar apenas 50% de cobertura. Neste caso, a razão é que a combinação especificada de entradas representadas em tais transições não ocorrem durante o caso de teste incluído no relatório. Por conta disso, o Modelo do Software da Bomba de Insulina alcançou 92% de cobertura em relação ao total de decisões presentes no modelo, 75% de cobertura em relação às condições e 50% de cobertura considerando o critério MDCC.

## 6.1 Considerações Finais

Neste capítulo foi descrito um procedimento que pode ser adotado para análise formal dos modelos de SMFC construídos a partir do reuso dos modelos disponíveis nas bibliotecas de modelos apresentada neste trabalho. Tal análise consiste de simular e formalizar propriedades do modelo do sistema a ser desenvolvido, por exemplo, do tipo funcional e de segurança, além de suposições acerca desses modelos. Dessa forma, os resultados da simulação são utilizados como entrada para a análise de cobertura dos modelos, de acordo com diferentes critérios de cobertura estrutural, tais como as possíveis decisões e condições presentes nos modelos.

Para a análise formal dos modelos foram utilizados os componentes de software *Simulink Design Verifier* (SLDV) e *Stateflow* do Matlab<sup>®</sup> como ferramentas de modelagem e verificação formal dos modelos.

# Capítulo 7

## Conclusão

Neste trabalho foi apresentada uma abordagem baseada em modelos para validação de Sistemas Médicos Físico-Cibernéticos (SMFC). Uma biblioteca de modelos reutilizáveis foi apresentada, sendo esta composta por dois tipos de modelos: pacientes e dispositivos. Modelos de pacientes são baseados em diretrizes médicas e base de dados clínicos. Modelos de dispositivos são construídos a partir da especificação técnica dos dispositivos sensores e atuadores. Os modelos são construídos como componentes reutilizáveis, com natureza personalizável o suficiente para ser reutilizada em diferentes SMFC.

A viabilidade prática da solução foi demonstrada por meio da construção de modelos para três diferentes contextos clínicos: (1) um ambiente de unidade de terapia intensiva; (2) situações de uso de uma bomba de infusão de insulina; e (3) um contexto de paciente diabético que necessita controlar o seu nível de glicose sanguínea. A abordagem foi também validada com sucesso por meio de uma avaliação empírica com desenvolvedores. Por meio da avaliação empírica foi possível concluir estatisticamente que a abordagem proposta promove o reúso e a produtividade dos desenvolvedores na etapa de validação em tempo de projeto de SMFC, reduzindo o esforço e o tempo requeridos para realização dessa etapa. Finalmente, foi realizada uma verificação formal dos modelos em relação aos requisitos funcionais e de segurança, bem como a análise de cobertura dos modelos que compõem um dos três cenários clínicos apresentados.

Portanto, as principais contribuições deste trabalho são: (i) a definição de uma abordagem baseada em modelos para apoiar o desenvolvedor na validação de SMFC;

(ii) a definição de uma arquitetura baseada em modelos para SMFC; (iii) a concepção de diretrizes para guiar o desenvolvedor na construção de modelos de pacientes e dispositivos para o contexto de SMFC; (iv) a disponibilização de bibliotecas de modelos de pacientes e dispositivos (i.e., médicos e equipamentos) prontos para o reúso em diferentes contextos clínicos de interesse do desenvolvedor; e (v) a demonstração de como fazer o reúso desses modelos no processo de validação de SMFC.

Como trabalhos futuros, planeja-se estender a abordagem proposta para validação final de SMFC. Em outras palavras, almeja-se integrar as diretrizes de composição e de simulação definidas neste trabalho com abordagens para simulação e testes de hardware e software dos dispositivos. Mais especificamente, será possível aplicar mecanismos *Hardware In the Loop* (HIL) [7] para permitir a integração da entrada/saída de dispositivos médicos reais de/para o modelo de AOD, em tempo de execução. Além disso, possibilitar a geração de casos de teste para certificação de sistemas de software e aplicações baseadas nos casos de simulação descritos neste trabalho.

# Bibliografia

- [1] ADA. Diagnosis and Classification of Diabetes Mellitus. *Diabetes Care*, 27(1):s5–s10, Jan. 2004. American Diabetes Association (ADA).
- [2] G. Agha. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, Cambridge, MA, USA, 1986.
- [3] R. Alur. Formal Verification of Hybrid Systems. In Proceedings of the Ninth ACM International Conference on Embedded Software. In *Proceedings of the Ninth ACM International Conference on Embedded Software*, pages 273–278, New York, NY, USA, 2011. ACM.
- [4] D. Ameller. Considering Non-Functional Requirements in Model-Driven Engineering. Master’s thesis, Llenguatges i Sistemes Informàtics (LSI), June 2009.
- [5] D. Arney, J. Plourde, R. Schrenker, P. Mattegunta, S. F. Whitehead, and J. M. Goldman. Design Pillars for Medical Cyber-Physical System Middleware. *OpenAccess Series in Informatics (OASICs)*, 36:124–132, 2014.
- [6] ASTM International. ASTM F2761-09(2013). Medical Devices and Medical Systems – Essential Safety Requirements for Equipment Comprising the Patient-Centric Integrated Clinical Environment (ICE), Part 1: General Requirements and Conceptual Model. [Online]. Disponível em: <<http://www.astm.org/Standards/F2761.htm>>. Acesso em: Mar. 2013., 2013.
- [7] M. Bacic. On Hardware-In-the-Loop Simulation. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, CDC-ECC ’05*, pages 3194–3198, Seville, Spain, 2005. IEEE.

- 
- [8] P.E.S. Barbosa, M. Morais, K. Galdino, M. Andrade, L. Gomes, F. Moutinho, and J.C.A. De Figueiredo. Towards medical device behavioural validation using Petri nets. In *Proceedings of the IEEE 26th International Symposium on Computer-Based Medical Systems, CBMS '13*, pages 4–10, Porto, Portugal, June 2013. IEEE.
- [9] V.R. Basili and H.D. Rombach. The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6):758–773, 1988.
- [10] B. W. Bequette. *Process Control: Modeling, Design, and Simulation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1 edition, 2002.
- [11] R. N. Bergman, L. S. Phillips, and C. Cobelli. Physiologic Evaluation of Factors Controlling Glucose Tolerance in Man: Measurement of Insulin Sensitivity and Beta-Cell Glucose Sensitivity from the Response to Intravenous Glucose. *Journal of Clinical Investigation*, 68(6):1456–1467, 1981.
- [12] A. E. Buxton, H. Calkins, D. J. Callans, J. P. DiMarco, et al. ACC/AHA/HRS 2006 Key Data Elements and Definitions for Electrophysiological Studies and Procedures: A Report of the American College of Cardiology/American Heart Association Task Force on Clinical Data Standards (ACC/AHA/HRS Writing Committee to Develop Data Standards on Electrophysiology). *Journal of the American College of Cardiology*, 48(11):2360–2396, Dec. 2006.
- [13] A. Canedo, E. Schwarzenbach, and M. A. Al Faruque. Context-sensitive synthesis of executable functional models of cyber-physical systems. In *Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems*, pages 99–108, New York, NY, USA, 2013. ACM.
- [14] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, New York, NY, USA, 2 edition, 2008.
- [15] A. V. Chobanian, G. L. Bakris, H. R. Black, W. C. Cushman, et al. Seventh Report of the Joint National Committee on Prevention, Detection, Evaluation, and Treatment of High Blood Pressure. *Hypertension*, 42(6):1206–1252, Dec. 2003.

- [16] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, et al. Looking Ahead in Pervasive Computing: Challenges and Opportunities in the era of Cyber-Physical Convergence. *Pervasive and Mobile Computing*, 8(1):2 – 21, Feb. 2012.
- [17] S. C. Delaune and P. K. Ladner. *Fundamentals of Nursing : Standards & Practice*. Cengage Learning, Clifton Park, NY, USA, 4 edition, 2011. 1504 p.
- [18] Diabetes.co.uk. Basal Bolus - Basal Bolus Injection Regimen. [Online]. Disponível em: <<http://www.diabetes.co.uk/insulin/basal-bolus.html>>. Acesso em: Mar. 2014.
- [19] S. Don, E. Choi, and D. Min. Event driven adaptive awareness system for Medical Cyber Physical Systems. In *Proceedings of the 4th International Conference on Awareness Science and Technology, iCAST '12*, pages 238–242, 2012.
- [20] FDA. Classify Your Medical Device. [Online]. Disponível em: <<http://www.webcitation.org/6ZdUv0YwI>>. Acesso em: Set. 2014., 2014a.
- [21] FDA. CFR - Code of Federal Regulations Title 21 – Part 820. [Online]. Disponível em: <<http://www.webcitation.org/6cFutBPFY>>. Acesso em: Set. 2014., 2014c.
- [22] FDA. Medical Device Recall Report - FY2003 to FY2012. [Online]. Disponível em: <<http://www.webcitation.org/6ZdUQoc4Q>>. Acesso em: Set. 2014., 2014d.
- [23] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, New Jersey, USA, 4 edition, 2002.
- [24] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson Prentice-Hall, Boston, 6 edition, 2009.
- [25] M. B. Gomes and A. C. Lerario. Gerenciamento Eletrônico do Diabetes. In M. B. GOMES and A. C. LERARIO, editors, *Diretrizes da Sociedade Brasileira de Diabetes*, SBD '09, pages 219–230. SBD, 2009.
- [26] J. V. González, O. A. V. Arenas, and V. V. González. Semiología de los Signos Vitales: Una Mirada Novedosa a un Problema Vigente. *Archivos de Medicina (Col)*, 12(2):221–240, 2012.

- 
- [27] J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, and R. L. Tatham. *Análise Multivariada de Dados*. Bookman, Porto Alegre, Brasil, 6 edition, 2009.
- [28] Y. Handelsman, J. Mechanick, L. Blonde, G. Grunberger, et al. American Association of Clinical Endocrinologists Medical Guidelines for Clinical Practice for Developing a Diabetes Mellitus Comprehensive Care Plan: Executive Summary. *Endocrine Practice*, 17(2):287–302, Mar. 2011. American Association of Clinical Endocrinologists (AACE).
- [29] S. A. Haque, S. M. Aziz, and M. Rahman. Review of Cyber-Physical System in Healthcare. *International Journal of Distributed Sensor Networks*, 2014:20 pages., 2014.
- [30] J. Hassine. Early modeling and validation of timed system requirements using Timed Use Case Maps. *Requirements Engineering*, 20(2):181–211, 2015.
- [31] E. M. Hernandez, A. Zamboni, S. Fabbri, and A. D. Thommazo. Using GQM and TAM to evaluate StArt - a tool that supports Systematic Review. *CLEI Electronic Journal*, 15(1):13 pages, 2012.
- [32] J. Hutchinson, J. Whittle, M. Rouncefield, and S. Kristoffersen. Empirical Assessment of MDE in Industry. In *Proceedings of the 33rd International Conference on Software Engineering, ICSE '11*, pages 471–480, New York, NY, USA, 2011. ACM.
- [33] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 1991.
- [34] R. Jetley, I. S. Purushothaman, and P.L. Jones. A formal methods approach to medical device review. *Computer*, 39(4):61–67, 2006.
- [35] Z. Jiang, M. Pajic, R. Alur, and R. Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2):191–213, 2014.

- 
- [36] Z. Jiang, M. Pajic, and R. Mangharam. Model-Based Closed-Loop Testing of Implantable Pacemakers. In *Proceedings of the IEEE/ACM International Conference on Cyber-Physical Systems*, pages 131–140, Chicago, IL, USA, 2011. IEEE.
- [37] Z. Jiang, M. Pajic, and R. Mangharam. Cyber-Physical Modeling of Implantable Cardiac Medical Devices. *Proceedings of the IEEE*, 100(1):122–137, 2012.
- [38] W. Kang, P. Wu, M. Rahmaniheris, L. Sha, R. B. Berlin, and J. M. Goldman. Towards organ-centric compositional development of safe networked supervisory medical systems. In *Proceedings of the IEEE 26th International Symposium on Computer-Based Medical Systems, CBMS '13*, pages 143–148, Porto, Portugal, 2013. IEEE.
- [39] B. G. Katzung. *Basic & Clinical Pharmacology*, chapter Introduction. McGraw-Hill Companies, Inc., 12 edition, 2012.
- [40] K. Khan, R. Kunz, J. Kleijnen, and G. Antes. *Systematic Reviews to Support Evidence-Based Medicine: How to Review and Apply findings of Health Care Research*. Hodder Arnold, London, UK, 2 edition, 2011.
- [41] S. H. Khan, A. H. Khan, and Z. H. Khan. Artificial Pancreas Coupled Vital Signs Monitoring for Improved Patient Safety. *Arabian Journal for Science and Engineering*, 38(11):3093–3102, 2013.
- [42] M.-J. Kim, S. Kang, W.-T. Kim, and I.-G. Chun. Human-interactive hardware-in-the-loop simulation framework for cyber-physical systems. In *Proceedings of the Second International Conference on Informatics and Applications*, pages 198–202, Lodz, Poland, 2013. IEEE.
- [43] A. L. King, L. Feng, O. Sokolsky, and I. Lee. Assuring the safety of on-demand medical cyber-physical systems. In *Proceedings of the IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications, CPSNA '13*, pages 1–6, Taipei, Taiwan, 2013. IEEE.
- [44] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *Software, IEEE*, 12(4):52–62, 1995.

- 
- [45] E. Lee. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors*, 15:4837–4869, 2015.
- [46] E. A. Lee. Cyber Physical Systems: Design Challenges. In *Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, ISORC '08*, pages 363–369, Washington, DC, USA, 2008. IEEE.
- [47] E. A. Lee. Heterogeneous Actor Modeling. In *Proceedings of the Ninth ACM International Conference on Embedded Software, EMSOFT '11*, pages 3–12, New York, NY, USA, 2011. ACM.
- [48] E. A. Lee. Constructive Models of Discrete and Continuous Physical Phenomena. *Access, IEEE*, 2:797–821, 2014.
- [49] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*. UC Berkeley, 2011. ISBN 978-0-557-70857-4.
- [50] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*. LeeSeshia.org, 1.5 edition, 2014.
- [51] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, et al. Challenges and Research Directions in Medical Cyber-Physical Systems. *Proceedings of the IEEE*, 100(1):75–90, Jan. 2012.
- [52] C. Li, A. Raghunathan, and N. K. Jha. Improving the Trustworthiness of Medical Device Software with Formal Verification Methods. *IEEE Embedded Systems Letters*, 5(3):50–53, 2013.
- [53] T. Li, F. Tan, Q. Wang, L. Bu, J-N. Cao, and X. Liu. From Offline toward Real-Time: A Hybrid Systems Model Checking and CPS Co-design Approach for Medical Device Plug-and-Play (MDPnP). In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, ICCPS '12*, pages 13–22, Washington, DC, USA,, 2012. IEEE.
- [54] T. Li, F. Tan, Q. Wang, L. Bu, J-N. Cao, and X. Liu. From Offline toward Real Time: A Hybrid Systems Model Checking and CPS Codesign Approach for

- Medical Device Plug-and-Play Collaborations. *IEEE Transactions on Parallel and Distributed Systems*, 25(3):642–652, 2014.
- [55] Y. Li, P. Oladimeji, C. Monroy, A. Cauchi, H. Thimbleby, D. Furniss, C. Vincent, and A. Blandford. Design of interactive medical devices: Feedback and its improvement. In *Proceedings of the 2011 International Symposium on IT in Medicine and Education*, volume 2 of *ITME '11*, pages 204–208, Guangzhou, China, Dec. 2011. IEEE.
- [56] J. J. Majikes, R. Pandita, and T. XIE. Literature Review of Testing Techniques for Medical Device Software. In *Proceedings of the 4th Medical Cyber-Physical Systems Workshop*, MCPS '13, page 8 pages., Philadelphia, PA, USA, Apr. 2013.
- [57] Mathworks. Simulink Design Verifier. [Online]. Disponível em: <[www.mathworks.com/products/sldesignverifier](http://www.mathworks.com/products/sldesignverifier)>. Acesso em: Jul. 2015., 2012.
- [58] Mathworks. Stateflow. [Online]. Disponível em: <[www.mathworks.com/products/stateflow](http://www.mathworks.com/products/stateflow)>. Acesso em: Jul. 2015., 2014.
- [59] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, London, 2 edition, 1989.
- [60] S. R. McGee. *Evidence-Based Physical Diagnosis*, chapter Respiratory Rate and Abnormal Breathing Patterns, pages 187–202. Saunders Elsevier, St. Louis, MO, USA, 2 edition, 2007.
- [61] B. Miller, F. Vahid, and T. Givargis. Digital Mockups for the Testing of a Medical Ventilator. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 859–862, Miami, FL, USA, 2012. ACM.
- [62] B. Miller, F. Vahid, and T. Givargis. MEDS: Mockup Electronic Data Sheets for Automated Testing of Cyber-physical Systems Using Digital Mockups. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '12*, pages 1417–1420, San Jose, CA, USA, 2012. IEEE.

- 
- [63] A. Murugesan, O. Sokolsky, S. Rayadurgam, M. Whalen, M. Heimdahl, and I. Lee. Linking abstract analysis to concrete design: A hierarchical approach to verify medical CPS safety. In *Proceedings of the ACM/IEEE 5th International Conference on Cyber-Physical Systems, ICCPS '14*, pages 139–150, Berlin, Germany, 2014. ACM/IEEE.
- [64] A. Murugesan, M. W. Whalen, S. Rayadurgam, and M. P.E. Heimdahl. Compositional Verification of a Medical Device System. In *Proceedings of the 2013 ACM SIGAda Annual Conference on High Integrity Language Technology*, pages 51–64, New York, NY, USA, 2013. ACM.
- [65] NHBPEP. The Fourth Report on the Diagnosis, Evaluation, and Treatment of High Blood Pressure in Children and Adolescents. *Pediatrics*, 114(Supplement 2):555–576, 2004.
- [66] K. Ogata. *Modern Control Engineering*. Prentice Hall, New Jersey, USA, 3 edition, 1997.
- [67] OMG. *MDA Guide Version 1.0.1*, miller, j. and mukerji, j. edition, Jun. 2003. Document Number: omg/2003-06-01.
- [68] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee. Model-Driven Safety Analysis of Closed-Loop Medical Systems. *IEEE Transactions on Industrial Informatics*, PP(99):1–1, 2012.
- [69] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee. Model-Driven Safety Analysis of Closed-Loop Medical Systems. *IEEE Transactions on Industrial Informatics*, 10(1):3–16, 2014.
- [70] K.-J. Park, R. Zheng, and X. Liu. Cyber-Physical Systems: Milestones and Research Challenges. *Computer Communications*, 36(1):1–7, 2012.
- [71] PHYSIONET. MIMIC II Databases. [Online]. National Institutes of Health (NIH). Disponível em: <<http://physionet.org/mimic2>>. Acesso em: Set. 2012., Jan. 2009.

- [72] C. Ptolemaeus, editor. *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org. [Online] Disponível em: <<http://ptolemy.org/systems>>. Acesso em: Out. 2014., Berkeley, CA, USA, 2014.
- [73] D. B. Rawat, J.J.P.C. Rodrigues, and I. Stojmenovic, editors. *Cyber-Physical Systems: From Theory to Practice*. CRC Press., 2015.
- [74] ROCHE. ACCU-CHEK Spirit Insulin Pump System: Pump User Guide. [Online]. Disponível em: <<http://www.webcitation.org/6ZdVHOgBx>>. Acesso em: Maio 2013, May. 2008.
- [75] C. B. Rosdahl and M. T. Kowalski. *Textbook of Basic Nursing*. Wolters Kluwer Health | Lippincott Williams & Wilkins, Philadelphia, PA, USA, 10 edition, 2012.
- [76] Douglas C. Schmidt. Guest Editor's Introduction: Model-Driven Engineering. *Computer*, 39(2):25–31, February 2006.
- [77] M. Shafique and Y. Labiche. A systematic review of state-based test tools. *International Journal on Software Tools for Technology Transfer*, 17(1):59–76, 2013.
- [78] L. C. Silva, H. O. Almeida, and A. Perkusich. Relatório de projeto de tese de doutorado. [Online]. Disponível em: <<http://1drv.ms/1qvnAxX>>. Acesso em: Ago. 2014., Mar. 2014.
- [79] L. C. Silva, H. O. Almeida, and A. Perkusich. Chapter: Verification and Validation of Medical Cyber-Physical Systems. Electronic Supplementary Material. [Online]. Disponível em: <<https://sites.google.com/site/vvmcps/>>. Acesso em: 24 Out. 2015., Set. 2014c.
- [80] L. C. Silva, M. Perkusich, F. M. Bublitz, H. O. Almeida, and A. Perkusich. A Model-Based Architecture for Testing Medical Cyber-Physical Systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC '14*, pages 25–30, New York, NY, USA, 2014a. ACM.
- [81] L.C. Silva, H.O. Almeida, A. Perkusich, and M. Perkusich. A Model-Based Approach to Support Validation of Medical Cyber-Physical Systems: Supplementary

- Material. [Online]. Disponível em: <<http://sites.google.com/site/mbatomcps/>>. Acesso em: 29 Out. 2015., Out. 2015.
- [82] A. Simalatsar and G. De Micheli. Medical guidelines reconciling medical software and electronic devices: Imatinib case-study. In *Proceedings of the IEEE 12th International Conference on Bioinformatics Bioengineering, BIBE' 12*, pages 19–24, Larnaca, Cyprus, 2012. IEEE.
- [83] O. Sokolsky, I. Lee, and M. Heimdahl. Challenges in the regulatory approval of medical cyber-physical systems. In *Proceedings of the 2011 International Conference on Embedded Software*, pages 227–232, Taipei, Taiwan, 2011. IEEE.
- [84] I. Sommerville. *Software Engineering*. Pearson Education, Inc., Boston, MA, USA, 9 edition, 2011.
- [85] T. Stahl, M. Voelter, and K. Czarnecki. *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons, 2006.
- [86] S. Tripakis, C. Stergiou, C. Shaver, and E. A. Lee. A Modular Formal Semantics for Ptolemy. *Mathematical Structures in Computer Science*, 23:834–881, Aug. 2013.
- [87] UC Berkeley. The Ptolemy Project: Heterogeneous, Modeling and Design. EECS Dept. [Online]. Disponível em: <<http://ptolemy.eecs.berkeley.edu/ptolemyII>>. Acesso em: Set. 2012., 2012.
- [88] M. Utting and B. Legeard. *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
- [89] K. Van Heusden, E. Dassau, H.C. Zisser, D.E. Seborg, and F.J. Doyle. Control-Relevant Models for Glucose Control Using A Priori Patient Characteristics. *IEEE Transactions on Biomedical Engineering*, 59(7):1839–1849, Jul. 2012.
- [90] K.K. Venkatasubramanian, S.K.S. Gupta, R.P. Jetley, and P.L. Jones. Interoperable Medical Devices. *Pulse, IEEE*, 1(2):16–27, Sept 2010.

- [91] S. Wang, L. Ji, A. Li, and J. Wu. Body Sensor Networks for Ubiquitous Healthcare. *Journal of Control Theory and Applications*, 9(1):3–9, Feb. 2011.
- [92] H. Washizaki, H. Yamamoto, and Y. Fukazawa. A Metrics Suite for Measuring Reusability of Software Components. In *Proceedings of the 9th IEEE International Symposium on Software Metrics*, pages 211–223, Sydney, Australia, 2003. IEEE.
- [93] J. Whittle, J. Hutchinson, and M. Rouncefield. The State of Practice in Model-Driven Engineering. *Software, IEEE*, 31(3):79–85, May 2014.
- [94] WHO. Telemedicine - Opportunities and developments in Member States: report on the second global survey on eHealth. In *Global Observatory for eHealth Series*, volume 2. World Health Organization, 2010.
- [95] WHO. World Health Statistics. Global Health Observatory (GHO), World Health Organization [Online]. Disponível em: <[http://www.who.int/gho/publications/world\\_health\\_statistics/2012/en/](http://www.who.int/gho/publications/world_health_statistics/2012/en/)>. Acesso em: Mar. 2014, 2012.
- [96] WHO. 10 Facts about diabetes. [Online]. Disponível em: <<http://www.who.int/features/factfiles/diabetes/facts/en>>. Acesso em: Jan. 2014., 2013.
- [97] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer, New York, NY, USA, 2012.
- [98] F-J. Wu, Y-F. Kao, and Y-C. Tseng. From Wireless Sensor Networks Towards Cyber-Physical Systems. *Pervasive and Mobile Computing*, 7(4):397–413, Aug. 2011.
- [99] A. B. Zamboni, A. D. Thommazo, E. C. M. Hernandez, and S. C. P. F. Fabbri. StArt - Uma Ferramenta Computacional de Apoio à Revisão Sistemática. In *Congresso Brasileiro de Software: Teoria e Prática*, pages 91–96, Salvador, BA, Brasil, 2010.

# Apêndice A

## Revisão Sistemática da Literatura

O objetivo deste capítulo é identificar as principais ferramentas e abordagens atualmente adotadas na Verificação e Validação (V&V) de SMFC, a fim de discutir o estado da arte relacionado à pesquisa. Desta forma, é realizada uma revisão da literatura para identificar e comparar as soluções propostas em relação aos diferentes aspectos, tais como contribuição, formalismos, ferramentas de apoio e estratégias de teste para SMFC. Neste sentido, apresenta-se também o processo de revisão sistemática utilizado nessa pesquisa.

### A.1 *Background*

Nesta seção, introduzem-se os conceitos fundamentais para testar um SMFC, tais como as atividades de V&V e *Model-Based Testing* (MBT). Adicionalmente, são apresentados alguns trabalhos relacionados.

#### A.1.1 **Atividades de Verificação e Validação**

O processo de verificação e validação de software é uma etapa importante do ciclo de desenvolvimento do sistema. Ele tem como objetivo mostrar que um sistema tanto encontra-se em conformidade com sua especificação como também ele satisfaz as expectativas do cliente do sistema [84].

Atividades de verificação e validação são atividades que se complementam para alcançar um propósito comum, que é assegurar que o sistema deve ser bom o suficiente para seu uso intencionado. A verificação está preocupada em checar se o desenvolvedor

está construindo corretamente o produto. A validação, no entanto, visa avaliar se o desenvolvedor está construindo o produto correto.

A principal atividade do processo V&V é o teste de sistema. Segundo Utting e Legeard [88], “*Testar é uma atividade realizada para avaliar a qualidade de um produto, e para melhorá-lo, por identificar defeitos e problemas*”. Resumindo, testar significa executar um sistema a fim de detectar comportamentos indesejados (i.e., falhas).

Jetley *et al.* [34] apresentam técnicas baseadas em métodos formais para garantir a operação segura e a confiabilidade de dispositivos médicos controlados por software. Neste sentido, os autores argumentam que o uso de uma abordagem integrada envolvendo modelos de uso, que são representações formais de software, juntamente com ferramentas e técnicas de análise estática, tal como análise baseada em *slicing*, podem tornar mais eficiente o processo de avaliação de software de dispositivos médicos, aumentando o seu rigor e reduzindo o esforço dos analistas forenses em caso de possíveis falhas detectadas nesses dispositivos na etapa de pós-venda.

Majikes *et al.* [56] fazem uma revisão das técnicas de teste que expõem falhas em dispositivos médicos controlados por software. O processo de revisão consiste em identificar e categorizar estudos de pesquisa de acordo com as falhas reportadas na classe I de *recalls* definida pela FDA e tipos de dispositivos médicos, como também em relação às técnicas de teste de software empregadas. Os resultados da pesquisa são apresentados de forma quantitativa.

Com o surgimento das abordagens de desenvolvimento baseado em modelos, artefatos como os próprios modelos, as propriedades formalizadas e os resultados da verificação e dos testes podem ser utilizados como evidência da qualidade de um SMFC [51]. Em SMFC, a segurança do paciente é a principal preocupação. Portanto, esses sistemas devem ser cuidadosamente e completamente testados.

### A.1.2 Teste Baseado em Modelo

O conceito de teste baseado em modelo (do inglês, *Model-Based Testing* – MBT) tem sido constantemente adotado para descrever uma variedade de técnicas utilizadas para gerar testes [88]. As quatro principais abordagens conhecidas são: 1) geração de dados de entrada de teste a partir do modelo de domínio; 2) geração de casos de teste a

partir do modelo de ambiente; 3) geração de casos de teste com informações sobre suas saídas (i.e., *oráculo*) a partir de um modelo de comportamento do sistema sob teste; e 4) geração de *scripts* de teste a partir de testes abstratos.

Shafique e Labiche [77] discutem sobre o estado atual das ferramentas de suporte a MBT com foco nas que dependem de modelos baseados em estado. Alguns critérios usados para comparar as ferramentas selecionadas foram definidos, incluindo aspectos como adequação do método e cobertura do teste relacionado ao código desenvolvido, além das atividades inerentes a MBT como, por exemplo, a criação e a verificação do modelo, depuração dos casos de teste, testes de regressão e rastreabilidade dos requisitos. Um total de 12 ferramentas foram selecionadas e brevemente apresentadas, sendo os resultados da revisão sistemática discutidos por meio do comparativo das ferramentas utilizando os critérios definidos no processo de revisão.

Uma vez que cada cenário clínico de um SMFC engloba uma heterogeneidade de dispositivos médicos integrados, todos controlados por sistemas computacionais para apoiar a tomada de decisão, muitas vezes com intervenções automatizadas, a segurança do paciente deve ser garantida por esses sistemas. Deste modo, torna-se fundamental que os desenvolvedores de SMFC utilizem-se das abordagens e ferramentas mais adequadas para verificar e validar tais sistemas. Portanto, essa é a principal motivação para a realização desta revisão sistemática.

## A.2 Materiais e Métodos

Nesta seção são descritos os passos realizados para identificar as principais abordagens e ferramentas, que representam o estado da arte em teste de projetos de SMFC.

O método de Revisão Sistemática da Literatura (RSL) usado nesta pesquisa é baseado em cinco passos descritos por Khan *et al.* [40], que são:

1. Definir as questões de pesquisa para o processo de revisão;
2. Identificar e selecionar os trabalhos relevantes;
3. Avaliar a qualidade dos trabalhos;

4. Sumarizar as evidências por meio da tabulação das características, qualidade e efeitos dos respectivos trabalhos;
5. Interpretar os resultados.

A ferramenta de auxílio para conduzir o processo de RSL foi a *StArt* v2.0 (acrônimo do termo em inglês “*State of the Art through Systematic Review*”) [99]. O objetivo da *StArt* é ajudar o pesquisador na aplicação do método de RSL por meio da gerenciamento das informações referentes à pesquisa, bem como na avaliação dos estudos identificados. Além de facilitar a análise qualitativa, a ferramenta *StArt* também fornece algumas funcionalidades que sintetizam os dados da pesquisa de forma quantitativa. Isto reduz o esforço do pesquisador na etapa de sumarização das evidências inerentes aos estudos analisados.

## A.3 Processo de Revisão Sistemática da Literatura

Com a finalidade de satisfazer os objetivos desta pesquisa, um protocolo de revisão sistemática foi definido e encontra-se disponível no Apêndice A. Alguns de seus elementos são descritos nesta seção, sendo estes agrupados de acordo com cada passo do método de RSL sugerido por Khan *et al.* [40]. Além disso, levou-se em consideração a forma como a ferramenta *StArt* [99] organiza tais informações, ou seja, seguindo as etapas de planejamento, execução e sumarização [31]. Pare obter informações adicionais sobre esta revisão sistemática, consultar Silva *et al.* [78; 79].

### A.3.1 Planejamento

Esta etapa é caracterizada pela definição do protocolo de pesquisa que irá apoiar os próximos passos no processo de revisão. O protocolo deve conter todas as informações e procedimentos necessários para a condução da pesquisa. Silva, Almeida e Perkusich [78] detalharam o protocolo definido para esta pesquisa. Nesta seção será apresentado apenas o objetivo, as questões de pesquisa usando o método de descrição livre, além

dos termos de busca que foram agrupados de acordo com a disciplina relacionada à pesquisa:

1. Definir as Questões de Pesquisa:

(a) *Objetivo da Pesquisa:*

Identificar as abordagens encontradas na literatura para definir e gerar casos de testes para validação de Sistemas Médicos Físico-Cibernéticos (do inglês, *Medical Cyber-Physical Systems* – MCPS). Especificamente, avaliar as abordagens e as ferramentas para melhorar as práticas de teste para estes sistemas, enquanto reduzem os custos (e.g., esforço e *time-to-market*) associados com a estratégia de teste.

(b) *Questões de Pesquisa (RQ):*

- i. **Questão de Pesquisa 1 (RQ1):** Quais técnicas de teste de software estão sendo empregadas nos projetos de SMFC?
- ii. **Questão de Pesquisa 2 (RQ2):** Quais as principais ferramentas adotadas para dar suporte à realização de teste de software em SMFC?

2. *Termos de Busca:*

- (a) Termos de Sistemas Médicos (MST) : “*medical cyber-physical systems*”;
- (b) Termos de Testes de Software (STT): “*testing*” or “*test*”;
- (c) Termos de Soluções (ST): “*technique*” or “*approach*” or “*methodology*” or “*method*”.

### A.3.2 Execução

Dado que o protocolo RSL foi estruturado e as questões de pesquisa foram definidas no *Passo 1* do processo de revisão, o pesquisador está apto a realizar a etapa de Execução, que consiste em:

1. Identificar os estudos relacionados às questões de pesquisa usando os motores de busca Web com uma lista de fontes de busca em que a revisão sistemática será realizada e seus respectivos refinamentos.

2. Selecionar ou rejeitar os estudos de acordo com os respectivos critérios de inclusão e exclusão especificados no protocolo de revisão, e, parcialmente, apresentados aqui:
  - (a) Critérios de Inclusão (I):
    - i. Estudos publicados em Conferências, Periódicos (*Journals*) e/ou Revistas (*Magazines*);
    - ii. Ano de Publicação entre 2006 e 21 de Jan. 2014;
    - iii. O texto completo do estudo deve estar disponível;
    - iv. Estudos que apresentam visões gerais sobre (*Medical*)/*Cyber-Physical Systems*, discutindo suas tendências e desafios;
    - v. Estudos que apresentam formalismos ou estratégias de teste para Sistemas Híbridos.
  - (b) Critérios de Exclusão (E):
    - i. Os termos de busca (*keywords*) encontrados estão presentes em: *Demo-abstract*; *Author index*; *Table of contents*; *Technical program*; *Index Proceedings*; *Title Page*; *References*.
    - ii. O estudo está indexado em mais de uma fonte de busca (duplicado);
    - iii. O estudo foi reestruturado e publicado novamente.
    - iv. O estudo não está em conformidade com os interesses da pesquisa;
    - v. O estudo não está associado ao domínio de sistemas médicos.
3. Extrair informações acerca dos estudos selecionados após uma leitura completa de seus textos, seguindo os campos do formulário de extração de informações elaborados para o protocolo de revisão.

### A.3.3 Sumarização

Dada a diversidade de informações dos resultados obtidos na etapa de Execução, a síntese dos dados realizada nesta pesquisa visa não apenas classificar os estudos de acordo com as estratégias de teste de software e as ferramentas de apoio ao processo V&V de um SMFC, mas também confrontá-las.

Complementando a etapa de Sumarização, esses estudos foram analisados quantitativamente e qualitativamente. Por fim, as evidências encontradas nos estudos foram caracterizadas e tabuladas, finalizando o processo de revisão com a interpretação dos resultados.

## A.4 Resultados

Os resultados da execução do método supracitado são apresentados nesta seção.

### A.4.1 Identificação

Nesta etapa preliminar da revisão foram encontrados 43 estudos. Isto significa que esses estudos satisfizeram os critérios iniciais do protocolo RSL, definido na etapa de Planejamento. Cinco fontes de busca de artigos científicos foram usadas: *IEEE Xplore DL*, *ACM DL*, *Science Direct*, Portal de Periódicos CAPES (*Scopus*, *Web of Science* e *SpringerLink*), e Manualmente. Leitores interessados em conhecer os refinamentos aplicados para cada fonte de busca podem consultar o protocolo RSL descrito no Apêndice B.

Alguns estudos foram manualmente incorporados ao processo de revisão. Portanto, essa fonte não possui um refinamento de busca explícito, mas seus estudos atendem os critérios de inclusão definidos no protocolo RSL desta pesquisa. Estes estudos são trabalhos relacionados à área de pesquisa de SMFC, que não foram retornados nas consultas realizadas pelos motores de busca Web elencados no protocolo de revisão.

Na Figura A.1 é ilustrado o número total de estudos encontrados após esse passo. A maioria desses estudos foram indexados pela *IEEE* (63%) e *ACM* (21%). Os demais motores de busca juntos contabilizam apenas 16% do total de estudos identificados nesta revisão.

Ao longo dos vários passos do processo de revisão sistemática, os estudos identificados nesta pesquisa são analisados e rotulados de acordo com o seu estado em cada etapa deste processo. Um rótulo pode assumir os seguintes valores: aceito (“*Accepted*”); rejeitado (“*Rejected*”); duplicado (“*Duplicated*”); ou não classificado (“*Unclassified*”). No

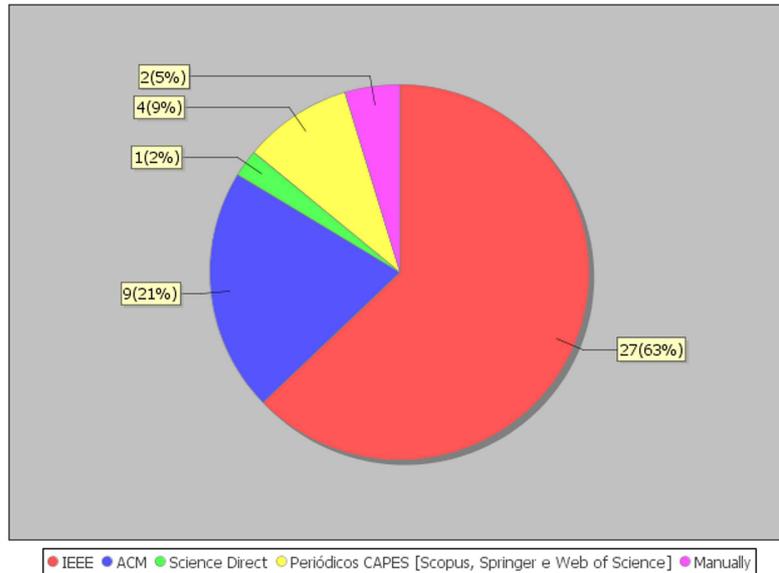


Figura A.1: Distribuição dos Estudos Primário por Fonte de Busca.

caso de estudos duplicados, a ferramenta *StArt* [99] permite identificá-los automaticamente por meio da análise comparativa das referências dos respectivos estudos.

#### A.4.2 Seleção

Levando em consideração os critérios de inclusão e exclusão apresentados no protocolo RSL, esta etapa teve como resultado 25 estudos aceitos para a etapa de Extração. Em relação aos demais estudos, 10 foram rejeitados e 8 considerados duplicados. Os critérios usados para atribuir o estado para cada estudo foram definidos no protocolo RSL descrito no Apêndice B.

Nesta etapa foi possível também determinar a ordem de prioridade de leitura dos estudos com base na avaliação dos seguintes metadados: título (“*title*”), resumo (“*abstract*”) e palavras-chave (“*keywords*”). Os dados desta análise quantitativa podem ser visualizados por meio dos gráficos referentes ao estado (“*Status*”) e prioridade de leitura (“*Reading Priority*”), como ilustrado na Figura A.2.

A ordem de prioridade de leitura pode ser classificada como muito alta (“*Very High*”), alta (“*High*”), baixa (“*Low*”) e muito baixa (“*Very Low*”). Vale ressaltar que esses dados são oriundos da execução do *Passo 2* do processo RSL, cujo objetivo é identificar e selecionar trabalhos relevantes para a pesquisa.

Como se pode observar na Figura A.2, apenas 58% dos estudos identificados em pesquisa inicial foram selecionados para a próxima etapa de análise, ou seja, etapa de Extração. Na etapa de Extração de informações, uma leitura completa do texto é realizada para obter dados relevantes sobre cada estudo.

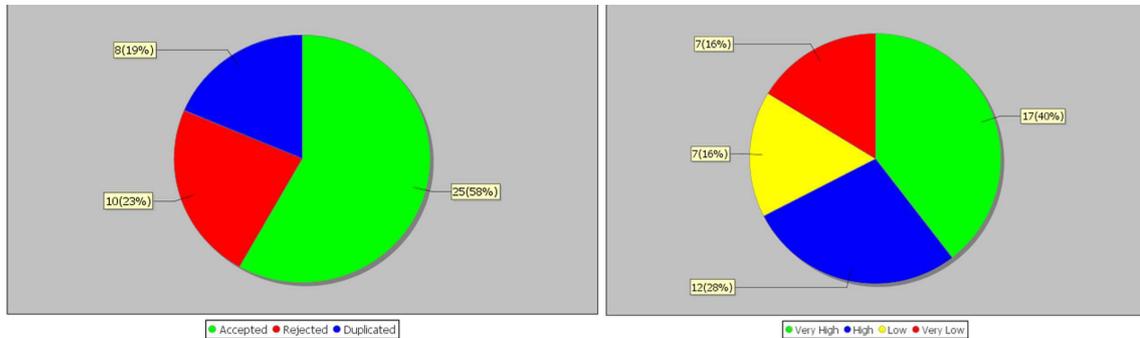


Figura A.2: Análise Quantitativa na Etapa de Seleção.

### A.4.3 Extração

Após a leitura completa dos 25 estudos selecionados na etapa anterior e extração das principais informações de cada estudo, somente 16 estudos foram considerados para a análise qualitativa que foi realizada no *Passo 3* do processo RSL. Outros 7 estudos foram rejeitados e 2 considerados duplicados, conforme ilustrado na Figura A.3.

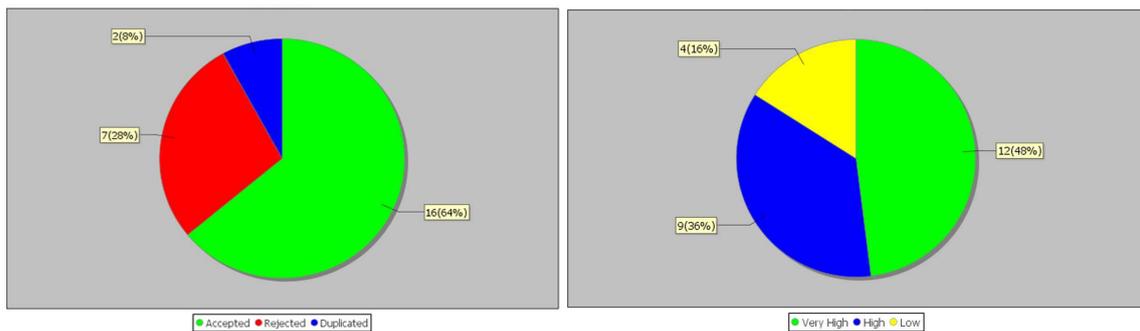


Figura A.3: Análise Quantitativa na Etapa de Extração.

A identificação dos estudos considerados duplicados nesta etapa foi possível apenas após a leitura completa dos mesmos, o que justifica a não exclusão de tais estudos na etapa anterior. Novamente, os critérios utilizados para selecionar os estudos a serem analisados nesse passo foram definidos no protocolo RSL descrito no Apêndice B.

Quatro categorias de informações foram criadas para extrair dados a partir da leitura completa dos 25 estudos selecionados a priori, que são: 1 - *Contribuição Principal*; 2 - *Formalismos*; 3 - *Ferramentas de Apoio*; 4 - *Estratégias de Teste*. A taxonomia foi definida para facilitar a categorização dos estudos, bem como a distinção entre as abordagens e ferramentas de testes identificadas na revisão. Lembrando que um mesmo estudo pode estar contido em mais de uma subcategoria para cada categoria apresentada, visto que um estudo pode utilizar diferentes formalismos, ferramentas e estratégias de teste nas diversas etapas do projeto de um SMFC. Os resultados da análise quantitativa são apresentados em Silva *et al.* [78; 79].

No *Passo 3* do processo RSL, o objetivo é avaliar a qualidade dos estudos após a leitura completa dos mesmos e extração de suas informações. Logo, os 16 estudos aceitos na etapa de Extração, considerados relevantes de acordo com o *Passo 2* do processo RSL, foram avaliados de forma mais detalhada com o propósito de explorar a heterogeneidade e obter outras informações que possibilitassem analisar a adequação dos estudos à pesquisa, além de verificar as evidências dos efeitos dos estudos.

Na Tabela A.1 são descritos os resultados da avaliação desses estudos.

ID do Estudo	Referência	Ranqueamento
8	Jiang <i>et al.</i> [36]	1
18	Li <i>et al.</i> [52]	2
26	Kang <i>et al.</i> [38]	3
6	Pajic <i>et al.</i> [69]	4
51	Murugesan <i>et al.</i> [64]	5
79	Khan <i>et al.</i> [41]	6
14	Li, Raghunathan and Jha [52]	7
31	King <i>et al.</i> [43]	8
27	Simalatsar and De Micheli [82]	9
9	Miller <i>et al.</i> [62]	10
13	Lee <i>et al.</i> [51]	11
19	Sokolsky <i>et al.</i> [83]	12
16	Alur [3]	13
12	Canedo <i>et al.</i> [13]	14
20	Kim <i>et al.</i> [42]	15
11	Don <i>et al.</i> [19]	16

Tabela A.1: Ranqueamento dos estudos selecionados na etapa de Extração.

A coluna *Ranqueamento* é uma informação derivada a partir da satisfação dos critérios de qualidade definidos no protocolo RSL, cuja função é medir a adequabilidade

de cada estudo ao objetivo da pesquisa. Os leitores interessados na forma de ranquear os estudos podem consultar Silva *et al.* [78; 79].

Com o ranqueamento dos 16 estudos, torna-se possível identificar aqueles que melhor respondem as questões de pesquisa abordadas nesta revisão, bem como executar o *Passo 4* do processo RSL (i.e., etapa de Sumarização). Na etapa de Sumarização, as evidências dos estudos são resumidas por meio de suas particularidades e seus efeitos.

#### A.4.4 Sumarização

Na Tabela A.2 são resumidos os dados obtidos após realização do *Passo 3* do processo RSL. Com isso, essa etapa consiste em classificar os estudos com relação às abordagens de teste e ferramentas de apoio aos processos V&V de um SMFC. Complementando essa etapa, os estudos foram analisados quantitativamente e qualitativamente.

De acordo com a Tabela A.2, 62,5% dos estudos utilizam a abordagem baseada em modelos como principal método formal para representar o comportamento de SMFC. Em geral, esses modelos são apoiados e/ou estão integrados com outros formalismos, tais como autômatos e suas variantes (e.g., temporizados, híbridos e com tarefas), o que ocorre em 60% dos casos.

No que diz respeito às ferramentas de apoio aos processos V&V de SMFC, o *Simulink* da Mathworks® é a ferramenta favorita dos projetistas e pesquisadores da área. Ela foi usada em 50% dos estudos analisados. Metade dos estudos que fizeram uso do Simulink também utilizaram a ferramenta *UPPAAL* para verificação dos modelos concebidos nos projetos de sistemas propostos. Outros aspectos interessantes foram percebidos como, por exemplo, o uso da ferramenta Stateflow, também da Mathworks®, para representar como esses sistemas reagem aos eventos, às condições baseadas em tempo e aos sinais de entrada externa. Além disso, a ferramenta *TIMES* é uma das principais ferramentas para geração de código a partir de modelos.

A classificação das abordagens levou em consideração os objetivos para os quais elas têm sido propostas dentro do contexto dos projetos de SMFC, como também em relação ao caso de uso para demonstrar sua aplicação. Deste modo, nesta classificação as abordagens são identificadas pelos respectivos estudos nos quais estas foram publicadas, conforme apresentado na Tabela A.3.

ID do Estudo	Características do Estudo			Estratégias de Teste
	Contribuição	Formalismos	Ferramentas de Apoio	
6	Abordagem	Baseada em Modelo; Autômatos.	UPPAAL; <i>Simulink</i> .	Simulação; <i>Model Checking</i> .
8	Sistema	Baseada em Modelo; Autômatos.	<i>Simulink</i> .	Monitores.
9	Abordagem	Maquete.	Outras.	Outras (ad-hoc).
11	Arcabouço	Dirigida a eventos.	Nenhum.	Outras (ad-hoc).
12	Abordagem	Baseada em Modelo; Outros.	<i>Modelica</i> .	Simulação.
13	Revisão da Literatura	Baseada em Modelo; Autômatos.	UPPAAL; <i>Simulink</i> ; Outras.	Simulação; <i>Model Checking</i> .
14	Abordagem	Outros (ad-hoc).	Outras.	<i>Model Checking</i> .
16	Revisão da Literatura	Baseada em Modelo; Autômatos; Outros.	<i>Simulink</i> ; <i>Modelica</i> ; Outros.	Simulação; <i>Model Checking</i> ; Outras.
18	Abordagem	Autômatos; ODE.	PHAVer.	<i>Model Checking</i> .
19	Revisão da Literatura	Baseada em Modelo; Autômatos; Outros.	UPPAAL; <i>Simulink</i> ; Outras.	Simulação; <i>Model Checking</i> .
20	Arcabouço	Outros.	Outras.	Simulação.
26	Abordagem	Baseada em Modelo; Outros.	<i>Simulink</i> ; Outros.	Simulação; <i>Model Checking</i> .
27	Abordagem	Autômatos.	Outras.	<i>Model Checking</i> ; Outros.
31	Abordagem	Baseada em Modelo; Outros.	Outras.	Outras.
51	Abordagem	Baseada em Modelo.	<i>Simulink</i> ; Outros.	<i>Model Checking</i> ; Outros.
79	Sistema	Baseada em Modelo; Autômatos; ODE.	UPPAAL; <i>Simulink</i> .	Simulação; <i>Model Checking</i> .

Tabela A.2: Synthesis of data from studies at the end of Step 3 of the RSL process.

ID do Estudo	Objetivo no Projeto SFC		Caso de Uso
	Sistema	Outros	
6	Verificar as propriedades de segurança	–	Sistema de bomba de infusão PCA ( <i>Patient-Controlled Analgesia</i> ) integrado ao modelo de paciente
9	–	Teste de software automatizado	Sistema de ventilação mecânica integrado ao modelo de pulmão
12 <sup>(1)</sup>	Gerar modelos funcionais executáveis para simulação	–	Máquina de Pipoca Eletromecânica
14	–	Verificar as propriedades de segurança	Software para marca-passo cardíaco
18	Verificar as propriedades de segurança	–	Procedimento cirúrgico à laser das vias aéreas com sistema de ventilação mecânica
26	Verificar as propriedades de segurança	–	Procedimento cirúrgico à laser das vias aéreas com sistema de ventilação mecânica
27	–	Validar as propriedades estruturais de protocolos médicos formalmente sentados; Verificar as propriedades de seus ciclos de vida e de tempo real	Protocolo para administração de medicamentos no tratamento de certos tipos de câncer
31	Verificar as propriedades de segurança no nível de requisitos	Verificar as propriedades de segurança no nível de requisitos	Sistema de Bomba de Infusão PCA, fornecendo uma interface para interação com o paciente
51	Verificar as propriedades de segurança no nível de requisitos	–	Sistema de Bomba de Infusão PCA Genérica

<sup>(1)</sup> Estudo está relacionado ao domínio de sistemas industriais, mas sua abordagem pode ser aplicada no domínio de sistemas médicos.

Tabela A.3: Classificação das principais abordagens de projeto de um SMFC.

Na Tabela A.4 é descrito o resultado da classificação das ferramentas. No processo de classificação das ferramentas notou-se a presença de dois importantes grupos: (1) as ferramentas que operam no nível de simulação do comportamento do modelo do sistema; e (2) aquelas com capacidade de prover mecanismos para a verificação de tais modelos. As demais ferramentas identificadas desempenham outros papéis no projeto de SMFC, tais como geração de código ou análise de requisitos e certas propriedades do sistema. Essas ferramentas foram categorizadas em um terceiro grupo denominado de “Ferramentas de Suporte”.

Ferramentas	Grupos		
	Simulação	Verificação	Suporte
<i>AGREE Framework</i>			●
<i>CBMC</i>			●
<i>MAP</i>			●
<i>Modelica</i>	●		
<i>PHAVer</i>		●	
<i>Simulink</i>	●		
<i>SLDV</i> <sup>1</sup>		●	
<i>Stateflow</i>	●		
<i>TIMES</i>			●
<i>UPPAAL</i>		●	

<sup>(1)</sup> *Simulink Design Verifier*.

Tabela A.4: Classificação das ferramentas identificadas no processo RSL, e que são utilizadas no processo V&V de um SMFC.

Finalmente, no *Passo 5* do processo de RSL, cujo objetivo é interpretar os resultados obtidos a partir da revisão, concluiu-se que o foco principal dos estudos é verificar as propriedades de segurança dos dispositivos ou sistemas médicos propostos. Nesse sentido, os autores apresentam cenários de uso específicos para demonstrar provas de conceito para cada projeto.

## A.5 Discussão

A heterogeneidade e a baixa qualidade de alguns estudos analisados dificultaram a interpretação dos resultados apresentados nesta revisão. Essa dificuldade está associada ao fato das abordagens identificadas não terem sido propostas especificamente para testar SMFC, mas dar suporte ao projeto desses sistemas como um todo.

Similarmente, contribuições específicas para o projeto de SMFC são propostas nesses estudos, dificultando comparar as abordagens identificadas considerando os tópicos inerentes ao teste dessa classe de sistemas, tais como tipos de teste, processos de teste e nível de cobertura. Outras dificuldades são devidas à diversidade das próprias abordagens identificadas, dado que elas não são soluções alternativas para um único problema, o que impossibilita realizar meta-análise sobre os estudos. No entanto, alguns estudos de alta qualidade entre os 16 estudos identificados na análise da revisão ajudam a superar esses problemas.

Os resultados apresentados nessa revisão sistemática da literatura não garantem a presença de todos os estudos relevantes para responder as questões de pesquisas desta revisão. Contudo, assegura-se que a maioria dos estudos disponíveis foi obtida e examinada, dado o esforço de pesquisa despendido e fontes de pesquisa incluídas nesta revisão.

Portanto, embora se tenha realizado uma pesquisa cuidadosa e criteriosa seguindo o protocolo de revisão definido para a mesma, sugere-se estender os termos de pesquisa relacionados aos sistemas médicos (e.g., “*medical applications*” e “*e-health systems*”), como também aos aspectos técnicos inerentes a esses sistemas (e.g., “*test case*”, “*systems verification*” e “*systems validation*”). Outras restrições que poderiam ser removidas em revisões futuras são a busca por termos apenas no idioma Inglês e em motores de busca Web, incluindo também materiais impressos, tais como livros e relatórios.

## A.6 Considerações Finais do Capítulo

Neste capítulo foi apresentada uma revisão sistemática dos recentes trabalhos relacionados à validação e verificação de Sistemas Médicos Físico-Cibernéticos (SMFC). A revisão visou identificar e avaliar as principais abordagens e ferramentas de suporte utilizadas para testes de SMFC. Para alcançar esse objetivo, definiu-se um Protocolo de Revisão Sistemática da Literatura.

Inicialmente foram identificados 43 estudos, mas apenas 16 foram selecionados para a análise qualitativa do processo de revisão. Após a leitura completa dos 16 estudos identificados nessa revisão, destacaram-se os trabalhos 6, 8, 18, 31 e 51, porque eles

descrevem em detalhes as questões relacionadas aos testes de SMFC.

Os resultados apresentados neste capítulo demonstram os esforços de pesquisa no contexto de SMFC, que no período de 2010 a 2013 tem se tornado uma área de pesquisa importante; especialmente na pesquisa de como fornecer mecanismos para assegurar a segurança do paciente e a ajuda de cuidadores na tomada de decisão correta de pacientes monitorados durante o tratamento, conforme cada cenário clínico em questão. Baseado nos resultados desta revisão, é possível afirmar que os objetivos desta pesquisa foram alcançados, uma vez que as principais abordagens e ferramentas de suporte foram identificadas com sucesso.

A revisão sistemática da literatura apresentada neste capítulo foi aprovada como capítulo de livro [79] na “*Encyclopedia of E-Health and Telemedicine*” a ser publicada pela *IGI Global*<sup>1</sup>.

---

<sup>1</sup><https://www.igi-global.com/>

# Apêndice B

## Protocolo de Revisão Sistemática da Literatura (RSL)

- Definir as questões de pesquisa para o processo de revisão:

1. *Objetivo da Pesquisa:*

Identificar na literatura as abordagens adotadas para definir e gerar casos de testes para validação de *Medical Cyber-Physical Systems* (MCPS). Especificamente, avaliar as abordagens e as ferramentas de testes que possam melhorar as práticas de teste adequadas para estes sistemas, enquanto reduzem os custos (e.g., esforço, financeiro e time-to-market) associados à realização desses testes.

1. *Descrição da Pesquisa:*

**Forma Livre:**

- **Questão de Pesquisa 1 (RQ1):** Quais técnicas de teste de software estão sendo empregadas nos projetos de MCPS?
- **Questão de Pesquisa 2 (RQ2):** Quais as principais ferramentas adotadas para dar suporte a realização de teste de software em MCPS?

**Forma Estruturada:**

- **População:** Estudos científicos que de algum modo utilizou alguma abordagem e/ou ferramenta de teste de software para execução de teste em sistemas médicos.
- **Intervenção:** Métodos formais/ferramentas para testes de sistemas médicos.

- 
- **Controle:** Fundamentação da área de pesquisa e alguns trabalhos relacionados.
  - **Resultados:** Conjunto das técnicas/ferramentas de teste utilizadas para minimização do esforço e dos custos necessários para realização de testes em MCPS.
  - **Aplicação:** *Medical Cyber-Physical Systems*.

- Identificar e selecionar os trabalhos relevantes para a pesquisa:

*Métodos de Fontes de Busca:*

- Pesquisa realizada por meio de motores de busca Web.

*Crítérios para Seleção das Fontes de Busca:*

- Principais bases de dados de indexação de estudos científicos nas áreas de Ciência da Computação.

*Fontes para Busca:*

- *IEEE Xplore Digital Library:* <http://ieeexplore.ieee.org>
- *ACM Digital Library:* <http://dl.acm.org>
- *Science Direct:* <http://www.sciencedirect.com>
- Portal de Periódicos da Capes: <http://www.periodicos.capes.gov.br>
  - \* *Scopus* (Elsevier);
  - \* *Web of Science* (Thomson Scientific);
  - \* *SpringerLink*.
- Outras (manualmente):
  - \* *Google Scholar*;
  - \* Publicações de Grupos de Pesquisas Acadêmicos.

- Idiomas dos Estudos:

- Inglês.

---

*Tipos de Estudos:*

- Revisões; Sistemas Computacionais Médicos; Formalismos e Ferramentas de apoio à Verificação e Validação de Sistemas de Segurança Crítica.

*Termos de Busca:*

- Termos de Sistemas Médicos (MST) : “*medical cyber-physical systems*”;
- Termos de Testes de Software (STT): “*testing*” ou “*test*”;
- Termos de Soluções (ST): “*technique*” ou “*approach*” ou “*methodology*” ou “*method*”.

*Refinamentos de Busca:*

- Encontrar termos em: Todos os campos (metadados e texto completo).

*Critérios de Inclusão:*

- (I) Estudos publicados em Conferências, Periódicos (*Journals*) e/ou Revistas (*Magazines*);
- (I) Ano de Publicação entre 2006 e 21 de Jan. 2014;
- (I) O texto completo do estudo deve possuir disponibilidade de acesso;
- (I) Estudos que apresentam visões gerais sobre (*Medical*)/*Cyber-Physical Systems*, incluindo tendências e desafios;
- (I) Estudos que apresentam formalismos ou abordagens de teste para Sistemas Híbridos.

*Critérios de Exclusão:*

- Quanto ao local da informação:
  - \* (E) Os termos de busca (*keywords*) encontrados estão presentes em:
    - *Demo-abstract*;
    - *Author index*;
    - *Table of contents*;

- *Technical program*;
  - *Index Proceedings*;
  - *Title Page*;
  - *References*.
- Quanto à redundância dos resultados:
    - \* (E) O estudo está indexado em mais de uma fonte de busca (duplicado);
    - \* (E) O estudo foi reestruturado e publicado novamente.
  - Quanto ao escopo do estudo:
    - \* (E) O estudo não está em conformidade com os interesses da pesquisa;
    - \* (E) O estudo não está associado ao domínio de sistemas médicos.

*Crítérios para Seleção Inicial dos Estudos:*

- Estudos que forneçam evidências de conformidade com o objetivo da pesquisa. Para isso serão selecionados os estudos...
  - \* ...cujos termos de busca definidos para a pesquisa estejam de forma explícita presentes (parcialmente ou totalmente) em seu título, resumo, palavras-chave e/ou termos indexados;
  - \* ...que forneçam algum tipo de revisão, formalismo, abordagem ou ferramenta de testes para *(Medical) Cyber-Physical Systems*.

*Campos para Extração de Dados dos Estudos:*

- Contribuição Principal = [Arquitetura, *Framework*, Abordagem/Metodologia, Protocolo, *Middleware*, Revisão Bibliográfica, Ferramenta, Sistema/Aplicação, Outras];
- Formalismo = [*Model-Based*, *Event-Driven*, *Automata*, ODE, *Mockup*, Outros, Nenhum];
- Ferramentas de Apoio = [*Ptolemy II*, *UPPAAL*, *Simulink*, *Modelica*, *SMV*, *PHAVer*, Outras, Nenhuma];
- Procedimentos de Teste = [Simulação, *Model Checking*, Monitores, Outros, Nenhum].

- Avaliar a qualidade dos trabalhos selecionados:

*Cr terios de Qualidade dos Estudos (CQ):*

- Adequa o aos termos de busca (metadados): Calcular o somat rio do produto entre os n meros de ocorr ncias e sua respectiva pontua o para cada termo em rela o aos campos t tulo (5 pontos), resumo (3 pontos) e palavras-chave (2 pontos);
  - Grau de conformidade com o interesse da pesquisa (conte do):
    - \* Alto; M dio; Baixo.
  - Intervalo de realiza o do estudo:
    - \* Maior que 3 anos;  ltimos 3 anos.
  - Detalhamento da solu o proposta para testes em MCPS:
    - \* Completo; Parcial; M nimo; Nenhum.
  - Suporte de Ferramentas:
    - \* Sim; N o (n o informado).
- Sumarizar as evid ncias por meio da tabula o das caracter sticas, qualidade e efeitos dos respectivos trabalhos:

*Sumariza o dos Estudos Selecionados:*

- Quanto  s abordagens de teste de software;
  - Quanto  s ferramentas de apoio ao teste de software;
  - An lise quantitativa e qualitativa dos estudos.
- Interpretar os resultados: Os resultados, e a sua interpreta o, est o limitados   abrang ncia da pesquisa e a heterogeneidade dos respectivos estudos identificados na literatura de MCPS.
    - Abrang ncia da Pesquisa:
      - \* Estudos publicados a partir do marco inicial que oficialmente caracterizou o surgimento do termo Cyber-Physical Systems (2006);

- \* Termos de busca especificamente evidenciados no idioma inglês;
  - \* Termos de busca da área médica restrito à “*Medical Cyber-Physical Systems*”, desconsiderando termos tais como “*medical systems*”, “*medical software*” e “*medical applications*”. Os resultados quantitativos em termos de números de estudos encontrados considerando todos os termos supracitados estão no arquivo *Search\_Results\_By\_Terms.xlsx* disponível em <https://goo.gl/PpquEz>.
- Heterogeneidade dos Estudos:
- \* Aspectos dos estudos avaliados podem influenciar na análise quantitativa dos resultados em virtude da diversidade de contribuições dos mesmos, que inclui, por exemplo, estudos de revisão do estado da arte, propostas de abordagens, arquiteturas, frameworks e aplicações.

# Apêndice C

## Consulta em *SQL* na Base de Dados Clínica *MIMIC II* para Caracterização da População de Interesse

*Selecionar todas as observações de pacientes acima de 15 anos de idade, com estadia registrada em UTI, cujos sinais de frequências cardíaca (211) e respiratória (618), pressão arterial (6,51,455,6701), temperatura corporal (676,677), além do nível de glicose (811,1529), mapeados por seus múltiplos identificadores, foram medidos simultaneamente e não possuam valores ausentes.*

---

```
1 SELECT
2     dp.subject_id , dp.sex , ic.icustay_admit_age as admit_age ,
3     ic.icustay_id , ce.charttime , height , weight_first as weight ,
4     ce.itemid as hr , ce.value1num as hr_value , ce2.itemid as bp ,
5     ce2.value1num as sbp_value , ce2.value2num as dbp_value ,
6     ce3.itemid as rr , ce3.value1num as rr_value , ce4.itemid as pt ,
7     ce4.value1num as pt_value , ce5.itemid as gl , ce5.value1num as
      gl_value
8 FROM
9     mimic2v26.d_patients dp , mimic2v26.icustay_detail ic ,
10    mimic2v26.chartevents ce , mimic2v26.chartevents ce2 ,
11    mimic2v26.chartevents ce3 , mimic2v26.chartevents ce4 ,
12    mimic2v26.chartevents ce5
13 WHERE
14    ce.itemid in (211) and ce2.itemid in (6,51,455,6701) and
15    ce3.itemid in (618) and ce4.itemid in (676,677) and
16    ce5.itemid in (811,1529) and
17    (ce2.charttime = ce.charttime) and
18    (ce3.charttime = ce2.charttime) and
```

---

```
19      (ce4.charttime = ce3.charttime) and
20      (ce5.charttime = ce4.charttime) and
21      ce.value1num is not null and ce2.value1num is not null and
22      ce2.value2num is not null and ce3.value1num is not null and
23      ce4.value1num is not null and ce5.value1num is not null and
24      (ce.subject_id = dp.subject_id) and
25      (ce2.subject_id = dp.subject_id) and
26      (ce3.subject_id = dp.subject_id) and
27      (ce4.subject_id = dp.subject_id) and
28      (ce5.subject_id = dp.subject_id) and
29      ce.icustay_id is not null and ce2.icustay_id is not null and
30      ce3.icustay_id is not null and ce4.icustay_id is not null and
31      ce5.icustay_id is not null and
32      (ce2.icustay_id = ce.icustay_id) and
33      (ce3.icustay_id = ce2.icustay_id) and
34      (ce4.icustay_id = ce3.icustay_id) and
35      (ce5.icustay_id = ce4.icustay_id) and
36      (ic.subject_id = dp.subject_id) and
37      (ic.icustay_id = ce.icustay_id) and
38      (extract(year from age(ce.charttime, dp.dob))*12 +
39      extract(month from age(ce.charttime, dp.dob)))/12 > 15 and
40      (extract(year from age(ce2.charttime, dp.dob))*12 +
41      extract(month from age(ce2.charttime, dp.dob)))/12 > 15 and
42      (extract(year from age(ce3.charttime, dp.dob))*12 +
43      extract(month from age(ce3.charttime, dp.dob)))/12 > 15 and
44      (extract(year from age(ce4.charttime, dp.dob))*12 +
45      extract(month from age(ce4.charttime, dp.dob)))/12 > 15 and
46      (extract(year from age(ce5.charttime, dp.dob))*12 +
47      extract(month from age(ce5.charttime, dp.dob)))/12 > 15
48 ORDER BY
49      dp.subject_id, ce.icustay_id, ce.charttime, ce.itemid
```

---

# Apêndice D

## Questionário sobre o Perfil do Desenvolvedor

*Este questionário visa coletar informações gerais acerca do nível de conhecimento dos desenvolvedores sobre Métodos Formais. Disponível online em: <https://goo.gl/b5JswO>.*

**\*Obrigatório**

**1. Qual o seu Nome?**

**2. Qual a sua idade?\***

Ex.: 20

**3. Qual o ano de ingresso no Curso de Graduação em Ciência da Computação (CC)?\***

Ex.: 2005

**4. Se você ainda está cursando CC, qual o seu atual período?**

Ex.: 2005.2

---

5. Se você concluiu o seu Curso de Graduação em CC, qual o ano de conclusão?

Ex.: 2009

6. Você saberia dizer do que se trata a disciplina de Métodos Formais?\*

Sim

Não

7. Se sua resposta para a questão anterior for “Sim”, você poderia explicar o conceito de Métodos Formais usando apenas uma única frase?

8. Se você cursou alguma disciplina envolvendo os conceitos de Métodos Formais, cite-as.

Ex.: Teoria da Computação; Tópicos em Formalismos de Computação; Métodos Formais.

9. Você já precisou formalizar algum componente de software para algum propósito específico, tal como compreensão componente ou análise de seu comportamento?\*

Sim

Não

10. Se sua resposta para a questão anterior for “Sim”, quais os tipos de formalismos utilizados?

Ex.: Autômatos; Redes de Petri; Atores, Diagrama de Blocos.

**11. Se sua resposta para a questão 9 for “Sim”, quais as ferramentas utilizadas?**

Ex.: UPPAAL; CPN Tools; Ptolemy II, Simulink.

**12. Em termos de reusabilidade, você costuma desenvolver componentes que podem ser reutilizados por você ou outros desenvolvedores?\***

- Nunca
- Às vezes
- Geralmente
- Quase sempre
- Sempre

**13. Quais os fatores que você considera relevante para reuso de componentes?\***

- Manutenibilidade
- Portabilidade
- Flexibilidade
- Compreensibilidade

Outro:

**14. Quais as métricas que você usaria para medir os fatores de reusabilidade mencionados na questão anterior?\***

---

15. Qual a importância de utilizar uma abordagem para a construção de componentes de reutilizáveis?\*

- Irrelevante
- Pouco Importante
- Importante
- Muito Importante
- Indispensável

16. Em se tratando da abstração de componentes utilizando abordagens baseadas em modelos, como modelos de componentes reutilizáveis podem auxiliá-lo na construção de um sistema a ser desenvolvido? \*

17. Para você o que significa um bom modelo para representar elementos como componentes/sistemas/hardware/software, etc.??\*

# Apêndice E

## Questionário de Avaliação de Reúso dos Modelos

*Este questionário visa coletar informações acerca do processo de validação de uso da abordagem e modelos disponíveis na construção de Modelos de Sistemas Médicos Físicos-Cibernéticos por um grupo de desenvolvedores. Disponível online em: <https://goo.gl/rOPLtc>.*

**\*Obrigatório**

**Qual o seu Nome?**

**Qual o grupo para o qual você foi selecionado?\***

- Controle
- Teste

**O que você achou do *Workshop* de Treinamento (Etapa 1)?\***

- Péssimo
- Regular
- Bom
- Ótimo
- Excelente

---

A partir dos conhecimentos transmitidos e material disponibilizado na Etapa 1, qual o seu nível de conhecimento adquirido sobre o trabalho?\*

- Nenhum
- Pouco
- Regular
- Bom
- Muito

Para você, quais os benefícios da abordagem proposta no trabalho para os desenvolvedores de Sistemas Médicos?

A partir dos conhecimentos transmitidos e material disponibilizado na Etapa 1, qual o seu nível de conhecimento adquirido sobre o Ptolemy II?\*

- Nenhum
- Pouco
- Regular
- Bom
- Muito

Você praticou no Ptolemy II a lista de exercícios solicitada ao término da Etapa 1?\*

- Sim
- Não

Se sua resposta para a questão anterior for “Não”, cite o motivo?

Em relação ao Cenário Clínico I, quanto tempo você demorou para modelar e alcançar o resultado da simulação?\*

**Dica:** Digite o valor do tempo em Minutos. Ex: 12.

Em relação ao Cenário Clínico II, quanto tempo você demorou para modelar e alcançar o resultado da simulação?\*

**Dica:** Digite o valor do tempo em Minutos. Ex: 18.

Em relação ao fator “Compreensibilidade”, como você avalia o esforço necessário para reconhecer o conceito por trás de um componente e sua aplicabilidade?\*

Responda de 1 a 5, considerando o nível de esforço crescente. Exemplo: 1 = pouco ou 5 para muito.

1   2   3   4   5  
○   ○   ○   ○   ○

Em relação ao fator “Adaptabilidade”, como você avalia a facilidade com que um modelo pode ser adaptado para cumprir uma exigência diferente daquela para a qual foi originalmente desenvolvido?\*

Responda de 1 a 5, considerando o nível de facilidade crescente. Exemplo: 1 = difícil ou 5 para fácil.

1   2   3   4   5  
○   ○   ○   ○   ○

---

**Em relação ao fator “Portabilidade”, como você avalia a facilidade com que o software pode ser transferido de um ambiente para outro?\***

Responda de 1 a 5, considerando o nível de facilidade crescente. Exemplo: 1 = difícil ou 5 para fácil.

1    2    3    4    5

**Quais as suas principais dificuldades na reutilização dos modelos disponíveis na biblioteca do usuário do Ptolemy II?**