Universidade Federal de Alagoas

Instituto de Computação

Coordenação de Pós-Graduação em Informática

Uma Abordagem Baseada em Modelos para Avaliação da Qualidade de Sistemas de Bomba de Infusão de Insulina

Mestrando

Tássio Fernandes Costa

Orientador

Álvaro Alvares de Carvalho César Sobrinho

Coorientador

Leandro Dias da Silva

Maceió, AL

Junho - 2022

Tássio Fernandes Costa

Orientador

Álvaro Alvares de Carvalho César Sobrinho

Coorientador

Leandro Dias da Silva

Maceió, AL

Junho - 2022

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

C837a Costa, Tássio Fernandes.

Uma abordagem baseada em modelos para avaliação da qualidade de sistemas de bomba de infusão de insulina / Tássio Fernandes Costa. – 2022.

71 f. : il.

Orientador: Álvaro Alvares de Carvalho César Sobrinho.

Co-orientador: Leandro Dias da Silva.

Dissertação (mestrado em informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 65-71.

1. Simulação. 2. Petri, Redes de. 3. Geometria e modelagem computacional. 4. insulina - Bomba de infusão. I. Título.

CDU: 004.414.23

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PPGI
Instituto de Computação/UFAL
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

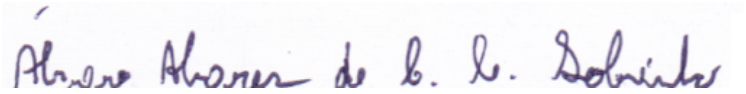
Folha de Aprovação

TÁSSIO FERNANDES COSTA

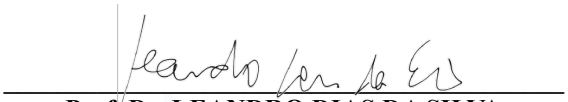
UMA ABORDAGEM BASEADA EM MODELOS PARA AVALIAÇÃO DA QUALIDADE DE SISTEMAS DE BOMBA DE INFUSÃO DE INSULINA

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 29 de junho de 2022.

Banca Examinadora:

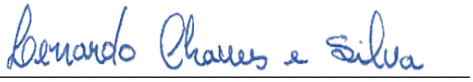


Prof. Dr. ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO
UFAPE-Universidade Federal do Agreste de Pernambuco
Orientador



Prof. Dr. LEANDRO DIAS DA SILVA
UFAL – Instituto de Computação
Examinador Interno e Coorientador

Prof. Dr. LEONARDO MELO DE MEDEIROS
Instituto Federal de Alagoas, IFAL-Maceió
Examinador Interno



Prof. Dr. LENARDO CHAVES E SILVA
Universidade Federal Rural do Semi-Árido, UFERSA-Mossoró
Examinador Externo

Prof. Dr. ANGELO PERKUSICH
Universidade Federal de Campina Grande, UFCG-Campina Grande
Examinador Externo

Agradecimentos

Inicialmente, quero agradecer ao meu orientador, professor Dr. Álvaro Alvares, por todo apoio e incentivo para que esse trabalho se tornasse realidade. Muito obrigado pela compreensão, amizade, paciência e motivação para seguirmos em frente mesmo diante das inúmeras dificuldades ao longo dessa jornada. Ao meu co-orientador, professor Dr. Leandro Dias, o meu muito obrigado pelas contribuições e apontamentos para a melhoria da qualidade da pesquisa. Aos professores, Dr. Angelo Perkusich, Dr. Leonardo Chaves e Dr. Leonardo Medeiros, gratidão por aceitarem o convite para fazer parte da banca avaliadora da dissertação.

Agradeço também aos meus pais, Geraldo Fernandes Costa e Auxiliadora Maria da Costa, que diante de suas limitações nunca mediram esforços para me ajudar a lutar pelos meus sonhos. Aos meus irmãos, Tiago, Gracinha, Talison, Talismar, Mateus e Débora, muito obrigado por estarem sempre juntos comigo. A minha noiva, Lilliane Domingos, sou grato por estar sempre ao meu lado compreendendo as minhas lutas e anseios. Amo todos vocês!

Por último, quero externalizar a minha gratidão ao meu tio Francisco Fernandes e às minhas tias Rosa, Fátima e Dorotéia (In Memoriam) por serem verdadeiros pilares para mim. Faltam palavras para descrever o quão foram e são determinantes em cada uma das minhas conquistas. Tenho a convicção que sem vocês não seria possível conquistar mais um título acadêmico. Muito obrigado por tudo e por tanto!

Resumo

Segurança e eficácia são atributos de qualidade cruciais para sistemas de bomba de infusão de insulina. Portanto, as agências reguladoras exigem a avaliação da qualidade e aprovação de tais sistemas antes do mercado para diminuir o risco de danos, motivando o uso de uma abordagem baseada em modelos (*Model-Based Approach - MBA*) formais para melhorar a qualidade. No entanto, usar uma *MBA* formal aumenta os custos e o tempo de desenvolvimento porque exige conhecimento especializado e análises minuciosas de comportamentos. O objetivo com este trabalho é auxiliar a avaliação de qualidade de tais sistemas de forma econômica e eficiente em termos de tempo, fornecendo artefatos de projeto reutilizáveis, aplicando a abordagem proposta (nomeada *MBA* com *CPN* - MBA/CPN). Foi definida uma abordagem baseada em redes de Petri coloridas (*Coloured Petri Nets - CPN*) e um estudo de caso sobre um sistema de bomba de infusão de insulina comercial para verificar e validar um modelo de referência (como um componente da MBA/CPN), descrevendo cenários de avaliação de qualidade. Também foi realizada uma avaliação empírica para verificar a produtividade e reutilização dos modeladores ao usar o modelo de referência. Tal modelo é relevante para raciocinar sobre comportamentos e avaliação de qualidade de tais sistemas concorrentes e complexos. Durante a avaliação empírica, utilizando o modelo de referência, 66,7% dos 12 modeladores entrevistados declararam nenhum esforço, enquanto 8,3%, esforço baixo, 16,7% esforço médio e 8,3% esforço considerável. Com base no conhecimento dos modeladores, foi implementado um aplicativo web para auxiliá-los na reutilização da abordagem proposta, possibilitando o treinamento baseado em simulação. Embora um número reduzido de modeladores tenha experimentado a abordagem, tal avaliação forneceu *insights* para melhorar a MBA/CPN. Dada a avaliação empírica e os resultados do estudo de caso, a MBA/CPN mostrou-se relevante para avaliar a qualidade de sistemas de bombas de infusão de insulina.

Palavras-chave: Simulação; Redes de Petri Coloridas; Modelagem; Bomba de Infusão de Insulina.

Abstract

Safety and effectiveness are crucial quality attributes for insulin infusion pump systems. Therefore, regulatory agencies require the quality evaluation and approval of such systems before the market to decrease the risk of harm, motivating the usage of a formal Model-Based Approach (MBA) to improve quality. Nevertheless, using a formal MBA increases costs and development time because it requires expert knowledge and thorough analyses of behaviors. We aim to assist the quality evaluation of such systems in a cost-effective and time-efficient manner, providing re-usable project artifacts by applying our proposed approach (named MBA with *CPN* - MBA/CPN). We defined a Coloured Petri nets (CPN) MBA and a case study on a commercial insulin infusion pump system to verify and validate a reference model (as a component of MBA/CPN), describing quality assessment scenarios. We also conducted an empirical evaluation to verify the productivity and reusability of modelers when using the reference model. Such a model is relevant to reason about behaviors and quality evaluation of such concurrent and complex systems. During the empirical evaluation, using the reference model, 66.7% of the 12 interviewed modelers stated no effort, while 8.3%, stated low effort, 16.7% medium effort, and 8.3% considerable effort. Based on the modelers‘ knowledge, we implemented a web application to assist them in re-using our proposed approach, enabling simulation-based training. Although a reduced number of modelers experimented with our approach, such an evaluation provided insights to improve the MBA/CPN. Given the empirical evaluation and the case study results, MBA/CPN showed to be relevant to assess the quality of insulin infusion pump systems.

Keywords: Simulation; Coloured Petri Nets; Modeling; Insulin Infusion Pump.

Lista de Figuras

2.1 Os principais componentes da *GSN* 11
2.2 Exemplo de um argumento de garantia GSN adaptado de [3] 12
2.3 Módulo de verificação de bateria do SBII. 14

4.1 Visão geral da MBA/CPN para avaliação da qualidade de SBII 24
4.2 Relacionamento entre fabricante e agência reguladora. 29
4.3 Caso de garantia *GSN* de nível mais alto de SBII. 30
4.4 Amostra da especificação *ACES* para o caso de garantia apresentado na Figura 4.3. 31
4.5 Módulo de *hardware* de SBII com duas baterias. 32
4.6 Módulo *Verify Battery* do primeiro refinamento do sistema. 33
4.7 Submódulo *Standard Infusion* do segundo refinamento do sistema. 37

5.1 Amostra de simulação do submódulo *Personalized Infusion*, representando o ACCU-CHECK Spirit. 40

6.1 Distribuição de respostas de acordo com os fatores de esforço e reutilização. 46
6.2 Modelo instanciado vs. instanciado corretamente. 47
6.3 Tempo coletado para o grupo controle e tratamento. 48
6.4 Distribuição de respostas considerando a complexidade das funcionalidades. 49

7.1 Aplicação Web e API implementadas como consumidores de componentes *Access/CPN*. 51
7.2 Diagrama de caso de uso. 52
7.3 Diagrama de pacotes. 53
7.4 Diagrama de classes. 54

7.5 Exemplo de interface gráfica do usuário da aplicação web, apresentando as principais funcionalidades. 57
7.6 Amostra da interface gráfica do usuário da aplicação web para simulação do sistema de bomba de infusão de insulina. 58

Lista de Tabelas

5.1 Descrição das simulações realizadas com base em testes abstratos. 41

6.1 Questionário identificando os perfis dos modeladores. 43

Conteúdo

1 Introdução **1**

1.1 Objetivos 4

1.2 Principais Contribuições 5

1.3 Metodologia 6

1.4 Organização do Documento 8

2 Embasamento Teórico **9**

2.1 Casos de Garantia 9

2.2 Notação Estruturada por Metas 10

2.3 Redes de Petri Coloridas 13

3 Trabalhos Relacionados **18**

3.1 Abordagens Baseadas em Modelos 18

3.2 Modularização, Composição e Reutilização de Modelos 20

3.3 Modelos de Bombas de Infusão de Insulina 21

4 MBA/CPN **23**

4.1 Decomposição de *hardware* e *software* 25

4.2 Primeiro Refinamento do Sistema 30

4.2.1 Módulo de *hardware* 30

4.2.2 Módulo de *software* 32

4.3 Segundo Refinamento do Sistema 36

5 Cenários de Avaliação de Qualidade **38**

5.1 Verificação do Primeiro Refinamento do Sistema 38

5.2 Validação do Segundo refinamento do Sistema 39

6 Avaliação Empírica **42**

6.1 Escopo, Modeladores e Variáveis 42

6.1.1 Procedimento e Medidas 44

6.2 Análise 46

7 Aplicação Web **50**

8 Discussão **59**

9 Conclusões e Trabalhos Futuros **63**

Capítulo 1

Introdução

O tratamento do diabetes geralmente requer o uso de Sistemas de Bomba de Infusão de Insulina (SBII). Os componentes de *hardware* compõem a bomba de infusão que emula o pâncreas humano, enquanto os componentes de *software* estão relacionados ao controle do comportamento da bomba (MERTZ, 2018 [22]). Como um sistema crítico de segurança, os fabricantes devem analisar os comportamentos dos SBII para fornecer, pelo menos, a garantia mínima exigida de correção (FRECKMANN *et. al.*, 2019 [11]).

De acordo com Woodcock *et. al.* (2009) [39] métodos formais desempenham um papel significativo na verificação dos requisitos de sistemas e na garantia da exatidão, confiabilidade e segurança dos sistemas desenvolvidos. No domínio médico, agências reguladoras, como, por exemplo, *Food and Drug Administration - FDA* dos Estados Unidos, precisam de meios eficazes para avaliar os dispositivos para certificar os sistemas desenvolvidos e garantir o comportamento seguro de cada sistema (CHEN *et. al.*, 2014 [4] e SIVAKUMAR *et. al.*, 2011 [32]) . As agências reguladoras carecem de técnicas e métodos rigorosos para fornecer garantia de segurança. Métodos formais podem ajudar a desenvolver sistemas confiáveis, seguros e protegidos. Além disso, podem fornecer evidências sólidas relacionadas com a confiança em funcionalidades, oferencendo suporte para a certificação de sistemas médicos confiáveis.

Nesta dissertação, foi utilizado o método formal chamado de redes de Petri coloridas (*Coloured Petri Nets - CPN*) (JENSEN; KRISTENSEN, 2015 [16]). O método foi usado para modelagem e validação de sistemas de bombas de infusão de insulina. *CPN* é uma linguagem gráfica formal para modelagem e validação de sistemas em que a simultaneidade

desempenha um papel importante. *CPN* têm sido aplicada com sucesso para a especificação formal, análise e verificação de diferentes protocolos de comunicação, sistemas embarcados e redes de dados, entre outros sistemas.

O manuseio de métodos formais geralmente requer conhecimento especializado e aumento de custos e tempo de desenvolvimento, o que é uma das principais motivações para realizar este estudo. Modelos de referência reutilizáveis têm o potencial de reduzir o impacto de tais custos. Em uma pesquisa anterior, Sobrinho *et. al.* (2017) [35] apresentaram um modelo de referência para auxiliar a certificação de sistemas biomédicos usando *CPN*, especificando componentes de *hardware* e *software* e aplicando simulações e a técnica de verificação de modelos. Em outra pesquisa anterior semelhante Costa *et. al.* (2019) [6] apresentaram um modelo de referência para auxiliar na certificação de SBII usando *CPN*. Nessas pesquisas, foi apresentada a relevância do uso de um modelo de referência *CPN* para gerar evidências para certificação. Uma limitação destes estudos anteriores é a falta de avaliação do modelo considerando a reutilização e produtividade, reduzindo a confiança na capacidade de diminuir custos e tempo de desenvolvimento na prática.

As agências governamentais reguladoras exigem que os fabricantes demonstrem que os SBII não colocam os usuários em situações de risco[1]. Em 2010, a *FDA* lançou a iniciativa de melhoria de bomba de infusão[2], referenciando problemas como defeitos de *software*, interfaces gráficas de usuário inadequadas e falhas mecânicas ou elétricas. Como resultado, a *FDA* divulgou em 2014 uma orientação para a indústria e a equipe da *FDA* seguirem durante o ciclo de vida de bombas de infusão de insulina[3]. De acordo com a mesma diretriz, os problemas mais comumente relatados de bombas de infusão incluem erro de *software*, fatores humanos, componentes quebrados, falha de bateria, falha de alarme e superinfusão e subinfusão. Alguns dos problemas relatados estão relacionados à atividade de projeto, enquanto outros são descritos pelos fabricantes como problemas desconhecidos, dificultando a obtenção de soluções.

Ainda existe um grande número de *recalls* relatados por agências governamentais regu-

[1]https://www.fda.gov/about-fda/economic-impact-analyses-fda-regulations/medical-device-classification-procedures-incorporating-fda-safety-and-innovation-act-procedures

[2]https://www.fda.gov/medical-devices/infusion-pumps/infusion-pump-improvement-initiative

[3]https://www.fda.gov/regulatory-information/search-fda-guidance-documents/infusion-pumps-total-product-life-cycle

ladoras em relação a SBII (RATHORE *et. al.*, 2018 [27]). Gao *et. al.* (2019) [12] afirmaram que dos 70 *recalls* de bombas de infusão lançados pela *FDA* entre 2001 e 2017, 17 *recals* foram causados por falhas de *software*. Para resolver esse problema, as agências reguladoras aumentaram o rigor da fiscalização quando os fabricantes submetem o sistema em desenvolvimento ao processo de certificação. Os fabricantes geralmente apresentam um conjunto de evidências de qualidade sobre sistemas baseados em padrões prescritivos (por exemplo, ISO 14971) e atributos de qualidade. Além das exigências exigidas pelos órgãos reguladores, um grande número de *recalls* também motiva a proposta e o uso de uma abordagem baseada em modelo (*Model-Based Approach - MBA*) formal para melhorar a qualidade. No entanto, sabe-se que o uso de tais abordagens geralmente aumenta os custos e o tempo de desenvolvimento.

Para abordar o problema do aumento de custos e tempo de desenvolvimento, neste trabalho é apresentado uma *MBA* de sistemas de bombas de infusão de insulina (nomeada *MBA* com *CPN* - MBA/CPN) com foco em modelos de referência *CPN* como artefatos de modelagem para aumentar a confiança nos comportamentos do sistema e fornecer avaliações de qualidade. A especificação do sistema de bombas de infusão de insulina está vinculada ao modelo *CPN* proposto pela definição de módulos *CPN* que representam partes críticas de tal sistema. Nesta dissertação, é descrito um estudo de caso em um sistema comercial, ou seja, o ACCU-CHEK Spirit[4], para avaliar um modelo de referência, como parte da MBA/CPN, por meio de simulações e da técnica de verificação de modelos, descrevendo cenários de avaliação de qualidade. O estudo de caso também é relevante para mostrar como os fabricantes podem reutilizar a MBA/CPN durante um processo de certificação. Portanto, esta pesquisa enfrenta desafios como (1) a integração de casos de garantia e a especificação de requisitos de *hardware* e *software* para fornecer reutilização e (2) a reutilização de um modelo de referência *CPN* de maneira eficiente em termos de tempo e custo.

A abordagem proposta pode beneficiar o processo de certificação pela reutilização de modelos de referência, juntamente com uma especificação de requisitos baseada em casos de garantia especificados com a notação estrutura por metas (*Goal-Structuring Notation - GSN*), nas fases iniciais do processo de desenvolvimento. O modelo de referência foi cuidadosamente validado para reduzir os possíveis impactos negativos de uma especificação

[4]https://bit.ly/2QzJMOT

manual. Também foi realizada uma avaliação empírica com 12 modeladores entrevistados para avaliar a MBA/CPN. Embora um número reduzido de modeladores tenha experimentado a abordagem, tal avaliação forneceu *insights* para melhorar a MBA/CPN. Este estudo está relacionado com os resultados publicados por Costa *et. al.* (2019) [6] e Sobrinho *et. al.* (2017) [35]. Assim, neste trabalho, a seguinte questão principal de pesquisa foi definida: a MBA/CPN é capaz de fornecer uma avaliação de qualidade com custo reduzido e eficiente de SBII? O termo avaliação de qualidade significa avaliar atributos de qualidade, como segurança e eficácia, conforme exigido por agências reguladoras.

Os trabalhos relacionados existentes (descritos no Capítulo 3) não fornecem um modelo de SBII genérico, paramétrico e temporizado para auxiliar os fabricantes na realização de análises detalhadas (por exemplo, controle de infusão e *recalls* comuns) durante o desenvolvimento e certificação. A MBA/CPN aborda essa limitação, considerando um modelo de SBII executável, genérico, paramétrico e temporizado, que inclui controle de infusão e considera as diretrizes do *FDA* e os *recalls* relatados. A contribuição com a abordagem proposta também está relacionada ao uso de casos de garantia *GSN* durante uma engenharia de requisitos baseada em metas e o treinamento baseado em simulação (não exigindo o *CPN/Tools*[5]) de modeladores para melhorar a reutilização.

Portanto, embora seja um consenso que a proposta e o uso de métodos formais para validar o comportamento de sistemas críticos seguros sejam relevantes, esta proposta também funciona como um guia, juntamente com artefatos de projeto reutilizáveis, para desenvolvedores de SBII. O objetivo é ajudá-los a melhorar a qualidade dos sistemas em desenvolvimento usando uma especificação baseada em casos de garantia e modelos de referência *CPN*. Para responder a questão de pesquisa, são apresentados resultados do estudo de caso e a avaliação empírica dos modelos, como descrito na próxima seção.

1.1 Objetivos

O principal objetivo com esta dissertação de mestrado é auxiliar aos modeladores de SBII a aumentar a confiança no comportamento do sistema e avaliar a qualidade de tais sistemas de maneira econômica e eficiente em termos de tempo. Para contemplar o objetivo principal,

[5]https://cpntools.org/

são necessários os seguintes objetivos específicos:

- definir uma abordagem baseada em modelos de SBII a partir do aprimoramento do modelo de referência *CPN* proposto anteriormente por Costa (2019) [5] e Costa *et. al.* (2019) [6] e na adoção e extensão do padrão para troca de casos de garantia (*Assurance Case Exchange Standard - ACES*) proposto por Sobrinho (2016) [34];
- realizar um estudo empírico baseado em um sistema comercial, ou seja, o ACCU-CHECK Spirit versão 2.XX, para avaliar a abordagem proposta.

O modelo de referência *CPN* proposto por Costa (2019) [5] e Costa *et. al.* (2019) [6] é executável, genérico, paramétrico e temporizado (considera a especificação de restrições de tempo e controle preciso da fusão de insulina). No entanto, isso pode ser melhorado considerando as diretrizes[6] da *FDA* e os *recalls* relatados por Gao *et. al.* (2019) [12] para SBII. Conforme mencionado acima, tanto as diretrizes da *FDA* quanto os recentes *recalls* de bombas de infusão de insulina indicam que uma parte considerável dos problemas em tais sistemas está relacionada ao projeto do *software*. Além disso, o modelo aprimorado é avaliado por sua capacidade de melhorar a produtividade dos projetistas de *software* e de ser reutilizado. Novos elementos também são incorporados à definição de *ACES*.

1.2 Principais Contribuições

Este estudo estende resultados de um projeto de pesquisa anterior realizada por Costa (2019) [5] e Costa *et. al.* (2019) [6], composto por quatro novas contribuições principais:

- MBA/CPN para projetar e avaliar a qualidade de SBII;
- um estudo empírico para avaliar a MBA/CPN proposta;
- uma interface de programação de aplicação (*Application Programming Interface - API*) Java para a integração de modelos *CPN* com aplicativos da Web;
- uma discussão sobre as vantagens e desvantagens do aprimoramento do modelo de referência *CPN* apresentado em Costa (2019) [5] e Costa *et. al.* (2019) [6].

[6]https://www.fda.gov/medical-devices/infusion-pumps/infusion-pump-improvement-initiative

A MBA/CPN consiste nas seguintes características:

- definição de um modelo de referência de sistemas de bombas de infusão de insulina reutilizável, paramétrico, temporizado e executável, que inclua controle de infusão e considere as diretrizes da *FDA* e *recalls* relatados por Gao *et. al.* (2019) [12];
- diretrizes para a avaliação dos atributos de qualidade de sistemas de bombas de infusão de insulina com base na reutilização de modelos;
- diretrizes para definição e disponibilização de módulos paramétricos, temporizados e executáveis de um modelo de referência de sistemas de bombas de infusão de insulina, para reutilização durante o processo de desenvolvimento (e integração com casos de garantia *GSN*);
- extensão do *ACES* para permitir a documentação da composição dos módulos do modelo *CPN*.

O estudo empírico é útil para avaliar esses elementos como parte da MBA/CPN. Por outro lado, o objetivo com a API é fornecer serviços para manipulação de modelos *CPN* sem a necessidade de uso de *CPN/Tools*. Tais serviços incluem, mas não se limitam a, obter uma marcação de lugar, verificar se uma determinada transição está habilitada e executar etapas de simulação. A API é baseada na biblioteca *Access/CPN* (WESTERGAARD, 2011 [38]).

1.3 Metodologia

Para contemplar os objetivos desta pesquisa, inicialmente foi realizada a definição de novos requisitos com base nos *recalls* e diretrizes da *FDA* para SBII. Posteriormente, foi realizada a especificação dos novos requisitos, gerando uma versão aprimorada do modelo proposto por Costa (2019) [5] e Costa *et. al.* (2019) [6]. Em seguida, a MBA/CPN foi definido e avaliado seguindo um estudo de caso sobre um sistema comercial e um estudo empírico envolvendo 12 participantes.

O estudo de caso para avaliação do modelo de referência foi realizado utilizando o sistema comercial denominado ACCU-CHEK Spirit. Foram realizadas as especificações do produto e do processo utilizando o *ACES* e também a configuração do modelo de referência

para representar a bomba ACCU-CHEK Spirit. Em seguida, a validação e a verificação foram realizadas por meio de simulação e verificação automática de modelos, respectivamente. O estudo de caso é relevante para mostrar como os fabricantes podem reutilizar o modelo de referência seguindo a MBA/CPN durante um processo de certificação.

Para avaliar a abordagem quanto à capacidade de promover a redução de custos e tempo de desenvolvimento, foi realizada uma avaliação empírica com 12 projetistas de *software*, que foram divididos em dois grupos: o grupo controle e o grupo tratamento. Ambos os grupos foram submetidos aos mesmos experimentos.

Os participantes do grupo de controle foram aqueles que não possuam formação em métodos formais, mas possuam formação na área de Computação. Em contrapartida, os participantes do subgrupo de tratamento foram aqueles que possuam conhecimento em métodos formais, tendo completado pelo menos 6 (seis) meses de curso nesta área.

A avaliação empírica aprovada pelo Comitê de Ética em Pesquisa consiste nas seguintes atividades:

- realização de treinamento em métodos formais utilizando o *CPN* e o *CPN/Tools*;
- treinamento em modelos genéricos de *CPN* com foco na apresentação de um modelo genérico de um sistema de aquisição de sinais biomédicos e de um sistema de aquisição de eletrocardiograma;
- condução do experimento de instanciação do modelo de referência aprimorado baseado no ACCU-CHEK Spirit 2.XX;
- realização do experimento para implementar dois novos requisitos no modelo de referência aprimorado (ou seja, adicionar uma nova representação de bateria e adicionar um recurso para recarregar a bateria);
- aplicação de dois questionários (questionário sobre o perfil do desenvolvedor e questionário para avaliação da reutilização do modelo genérico de bomba de insulina).

1.4 Organização do Documento

Esta dissertação está estruturada em 6 capítulos. No primeiro capítulo, foi apresentada uma introdução sobre o trabalho, composta pela contextualização da pesquisa, problematização, objetivos gerais e específicos, justificativas, contribuições e metodologia aplicada. Os demais capítulos estão organizados da seguinte maneira:

- No Capítulo 2 são apresentados conceitos sobre *GSN* e *CPN*;
- No Capítulo 3 são apresentados trabalhos relacionados ao tema de pesquisa;
- No Capítulo 4 é apresentada a abordagem baseada em modelos;
- No Capítulo 5 são apresentados cenários de avaliação de qualidade;
- No Capítulo 6 é apresenta a avaliação empírica;
- No Capítulo 7 é apresentada a aplicação web;
- No Capítulo 8 são apresentadas discussões;
- Por fim, no Capítulo 9 são apresentadas conclusões e trabalhos futuros.

Capítulo 2

Embasamento Teórico

Neste capítulo são apresentados conceitos sobre casos de garantia, a notação estrutura por metas (*Goal-Structuring Notation - GSN*) e redes de Petri coloridas (*Coloured Petri Nets - CPN*).

2.1 Casos de Garantia

Caso de garantia é um método usado para argumentar que uma reivindicação sobre os requisitos de um sistema é satisfeita [15]. Uma reivindicação é uma proposição que deve ser demonstrada como verdadeira ou falsa a partir de um conjunto de dados ou informações tangíveis denominado de evidências. As declarações que justificam a dedução de uma reivindicação com base em determinadas evidências formam um argumento em um caso de garantia. Portanto, um argumento é um conjunto de afirmações embasada em evidências que justifica o alcance de uma reivindicação sobre as características de um sistema.

Um exemplo simples e didático de reivindicação em um caso de garantia é a afirmativa de que um sistema de um aquecedor é seguro (CALINESCU *et. al.*, 2018 [3]). A demonstração de que essa afirmação é válida é feita com base em três sub-reivindicações: a função de controle do aquecedor é segura; a função de monitoramento do aquecedor é segura; e todas as funções do sistema são independentes. Para argumentar que essas sub-reivindicações são válidas e consequentemente a reivindicação principal é satisfeita, são usadas como evidências para cada uma das três sub-reivindicações os resultados de simulação, teste e prova formal, respectivamente. Assim, uma reivindicação pode ser composta por reivindicações

intermediárias que permitem fundamentar os argumentos para a dedução da reivindicação principal.

A sistemática e rigorosa forma para a construção de argumentos sobre a confiança em sistemas justificam o porquê casos de garantia tem sido uma abordagem com notável adoção no processo de desenvolvimento de aplicações críticas de segurança e de missão. São diversos os domínios onde eles vem sendo utilizados, incluindo desde dispositivos médicos [28, 14] até sistemas aviônicos [2, 31].

Casos de garantia podem ser usados em todas as etapas do processo de desenvolvimento de um sistema (LITTLEWOOD; BLOOMFIELD; BAINBRIDGE, 1998 [20]). Nas etapas iniciais do desenvolvimento de um sistema, evidências extraídas da análise de requisitos e de projeto podem compor os casos de garantia para demonstrar, por exemplo, manutenibilidade do sistema. Evidências obtidas nas fases de implementação, validação e verificação podem ser usadas para argumentar sobre confiabilidade e segurança dos sistemas em desenvolvimento. Além disso, durante o uso do sistema novas evidências coletadas em tempo de execução podem ser incorporadas nos casos de garantia para mantê-los sempre atualizados.

2.2 Notação Estruturada por Metas

GSN é um padrão usado para representar casos de garantia graficamente [1] e seus principais componentes são ilustrados na Figura 2.1. O uso de elementos gráficos, em vez de apenas uma relatório escrito em linguagem natural, facilita a estruturação e o entendimento dos casos de garantia.

Conforme a Figura 2.1 fazem parte da estrutura básica de um argumento especificado com *GSN* os componentes:

- meta (*Goal*): retângulo;
- solução (*Solution*): círculo;
- estratégia (*Strategy*): paralelogramo;
- contexto (*Context*): retângulo com bordas arredondadas;

[1] https://scsc.uk/gsn?page=gsn%202standard

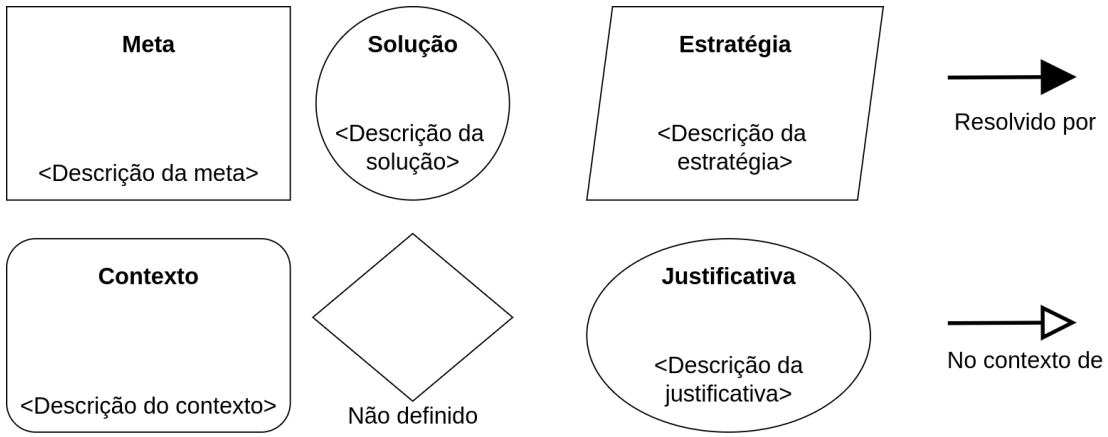


Figura 2.1: Os principais componentes da *GSN*

- justificativa (*Justification*): elipse;
- não definido (*Undefined*): diamante;
- suportado por (*SupportedBy*): seta direcionada sólida; e
- no contexto de (*InContextOf*): seta direcionada oca.

Os componentes `suportado por` e `no contexto de` são usados para conectar objetivos a soluções e realizar associações de contexto, respectivamente. As metas são úteis para fornecer reivindicações sobre a qualidade dos sistemas, enquanto as soluções estão vinculadas à evidência de tal afirmação. Estratégia, contexto e justificativa são componentes complementares para melhorar a clareza e integridade da especificação do caso de garantia. O componente `não definido` fornece uma maneira de ilustrar partes do caso de garantia que ainda estão em especificação. Na Figura 2.2 é apresentado o caso de garantia *GSN* do sistema de aquecimento exemplificado na Seção 2.1 em linguagem natural.

A meta principal (Meta 1) afirma que o sistema de aquecimento é seguro. Esta meta (reivindicação) é dividida em três sub-reivindicações usando a estratégia (Estratégia 1) que aborda a segurança das funções. A primeira sub-reivindicação (Meta 2) consiste em garantir que o sistema de controle é seguro com base em simulações (Solução 1). Na segunda sub-reivindicação (Meta 2́) é declarado que o sistema de monitoramento é seguro com base em resultados alcançados por meio de teste (Solução 2). Por último, na terceira sub-reivindicação (Meta 3) é afirmado que as funções do sistema são independentes com base em resultados obtidos de prova formal (Solução 3).

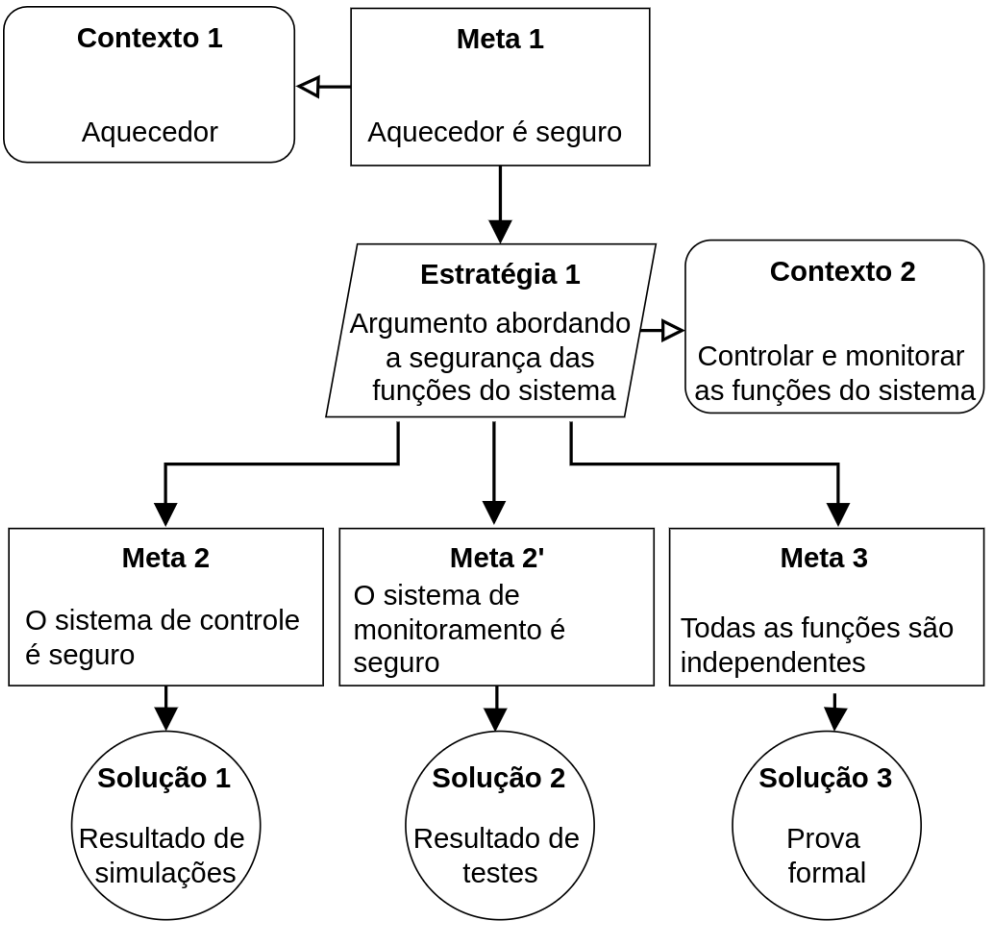


Figura 2.2: Exemplo de um argumento de garantia GSN adaptado de [3]

.

Além dos elementos básicos, a *GSN* inclui os seguintes componentes modulares:

- meta distante (*Away Goal*);
- módulo (*Module*) ;
- contrato (*Contract*);
- solução distante (*Away Solution*);
- contexto distante (*Away Context*); e
- indicator público (*Public Indicator*).

Os componentes modulares auxiliam na modularização do caso de garantia, visando fornecer representações mais compactas, simplificar a manutenção e melhorar a legibilidade.

O uso de módulos permite o agrupamento de argumentações para apoiar claramente as afirmações sobre a qualidade dos sistemas. A *GSN* modular é um dos conceitos fundamentais usados para definir a MBA/CPN.

2.3 Redes de Petri Coloridas

Redes de Petri coloridas (*Coloured Petri Nets - CPN*) é um formalismo matemático com representação gráfica voltada para a modelagem de sistemas de eventos discretos (DING; CHEN; WANG, 2017 [9]), isto é, sistemas cujo estado é alterado a partir da ocorrência de eventos específicos (SAIVES; FARAUT; LESAGE, 2018 [29]). Trata-se de uma extensão da rede de Petri tradicional contendo recursos adicionais para a modelagem que permitem a representação de características mais próximas do funcionamento real dos sistemas. Assim, além de conter os elementos herdados de rede de Petri tradicional (lugar, ficha, arco direcionado, peso de arco, transição e temporização), CPN possui modularização e linguagem de programação *CPN ML* (SAKIB; TARI; BERTOK, 2013 [30]).

Em uma *CPN*, o lugar possui várias interpretações, podendo representar, por exemplo, uma atividade, um predicado ou um estado parcial do sistema. A ficha é o elemento indicador de que um predicado associado a um lugar é verificado. O arco direcionado é o elemento que permite conectar lugar a transição e vice-versa. A transição simboliza um evento que ocorre no sistema. O peso de arco refere-se a um número natural positivo excluindo o zero, que é atribuído a cada arco. A temporização é o recurso que permite a modelagem de questões relacionadas com o tempo, como, por exemplo, a disponibilização de uma ficha em um lugar apenas quando o sistema atinge um tempo específico. A modularização permite que o modelo *CPN* seja dividido em partes gerenciáveis com interfaces bem definidas. Por último, a linguagem de programação *CPN ML*, baseada na linguagem de programação funcional *Standard ML*, fornece os primitivos para a definição de tipos de dados, para descrever a manipulação de dados e para criar modelos compactos e parametrizáveis.

Na Figura 2.3 pode-se observar os principais elementos de um modelo *CPN*. O exemplo trata-se do módulo de *software* para verificação de bateria do segundo refinamento do modelo *CPN* para o Sistema de Bomba de Infusão de Insulina (SBII) melhorado com base na abordagem apresentada neste trabalho. Maiores detalhes sobre o modelo completo são

apresentados no Capítulo 4.

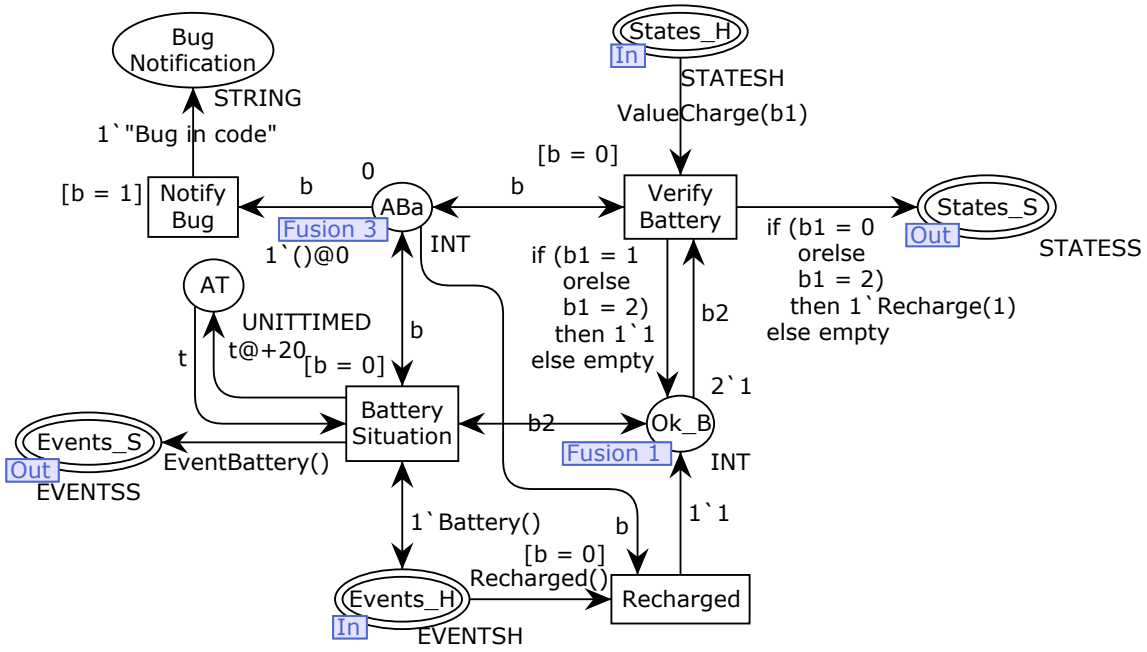


Figura 2.3: Módulo de verificação de bateria do SBII.

Conforme pode ser visto na Figura 2.3, para o módulo de verificação de bateria foram definidos os lugares `Events_S`, `States_S`, `Events_H`, `States_H`, `ABa`, `AT`, `Ok_B` e `Bug Notification`. Os quatro primeiros lugares armazenam eventos e estados de *software* e *hardware*, permitindo a comunicação e a troca de mensagens entre os módulos de *software* e *hardware* que podem afetar ou alterar o estado da bateria do SBII. Por outro lado, os lugares `ABa` e `AT` são usados para, respectivamente, indicar a ocorrência de um *bug* no sistema e armazenar o intervalo de tempo para o sistema realizar nova verificação da situação da bateria (a notação $@ + 20$ especifica uma expressão de atraso de tempo para que uma ficha esteja disponível naquele lugar). Por último, o local `Ok_B` é usado para armazenar a quantidade de instâncias de bateria com carga disponível e o local `Bug Notification` é usado para evidenciar que um *bug* ocorrido por alguma falha no sistema.

Os possíveis eventos que podem ser disparados no módulo de verificação da bateria da Figura 2.3 são representados pelas transições `Battery Situation`, `Verify Battery`, `Recharged` e `Notify Bug`. O disparo da transição `Battery Situation` inicia uma etapa de verificação da carga das baterias disponíveis no sistema, sendo o estado de cada bateria efetivamente verificado pelo disparo da transição `Verify Battery`, que pode le-

var a classificar a bateria como carregada ou descarregada, a depender do seu estado. Como pode ser observado, tal classificação é feita pela execução de um código escrito em *CPN ML*. Por fim, as transições `Recharged` e `Notify Bug` representam as ações de recarga da bateria e notificação da ocorrência de um bug no sistema, respectivamente.

Formalmente uma *CPN* pode ser definida por três partes. A primeira para definir *CPN* não hierárquica, isto é, *CPN* sem módulos. A segunda para definir um módulo de *CPN*. E por último, a terceira parte é a definição de *CPN* hierárquica. A definição da primeira parte, conforme Jensen e Kristensen (2009), e das outras duas partes, de acordo com Jensen e Kristensen (2015), são, respectivamente, as seguintes:

- *CPN* não hierárquica é uma tupla de nove elementos $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ na qual:

    1. P é um conjunto finito de lugares. Em outras palavras, a quantidade de lugares de uma *CPN* é finito.
    2. T é um conjunto finito de transições tal que $P \cap T =$. Ou seja, não existe algum elemento em T igual a algum elemento em P.
    3. $A \subseteq P \times T \cup T \times P$ é um conjunto de arcos direcionados. Isso significa que um arco pode ligar apenas um lugar a uma transição ou vice-versa. Não é possível usar um arco para ligar um lugar a outro lugar ou uma transição a outra transição.
    4. Σ é um conjunto não vazio de cores. Isto é, os lugares de *CPN* possui algum tipo de dado (ou cor), o qual pertence a Σ.
    5. V é um conjunto finito de variáveis tipadas tal que $Tipo[v] \in \Sigma$ para todas as variáveis $v \in V$. Ou melhor, existe uma cor associada a cada variável tal que essa cor é um elemento de Σ.
    6. $C : P \rightarrow \Sigma$ é uma função de conjunto de cor que associa uma cor a cada lugar. Dito de outro modo, cada lugar da *CPN* possui uma cor.
    7. $G : T \rightarrow EXPR_v$ é uma função de guarda que associa uma guarda a cada transição t tal que $Tipo[G(t)] = Bool$. Isto é, a cor de guarda de cada transição é um *booleano*.

8. $E : A \rightarrow EXPR_v$ é uma função de expressão de arco que associa uma expressão de arco a cada arco a tal que $Tipo[E(a)] = C(p)_{\text{MS}}$[2] , no qual p é o lugar conectado ao arco a. Em outros termos, a cor associada a a deve ser a mesma cor associada a p, que é o lugar com o qual a está conectado.
9. $I : P \rightarrow EXPR_\theta$ é uma função de inicialização que associa uma expressão de inicialização a cada lugar p tal que $Tipo[I(p)] = C(p)_{\text{MS}}$. Ou seja, a cor dos valores de inicialização de um lugar deve ser igual a cor daquele lugar.

- Módulo de *CPN* é uma tupla de quatro elementos $CPN_M = (CPN, T_{\text{sub}}, P_{\text{porta}}, PT)$, na qual:

  1. $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ é uma CPN não hierárquica.
  2. $T_{\text{sub}} \subseteq T$ é um conjunto de transições de substituição. Isso significa que uma transição de substituição é também uma transição.
  3. $P_{\text{porta}} \subseteq P$ é um conjunto de lugares porta. Dito de outro modo, um porta é também um lugar.
  4. $PT : P_{\text{porta}} \rightarrow IN, OUT, I/O$ é um conjunto de tipo porta que associa tipos de porta a lugares. Em outras palavras, uma porta deve possuir um dos seguintes tipos: $IN, OUT, I/O$.

- CPN hierárquica é definida como uma tupla de quatro elementos $CPN_H = (S, SM, PS, FS)$, na qual:

  1. S é um conjunto finito de módulos. Cada módulo é um módulo de Rede de Petri Colorida $s = ((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I^s), T^s_{\text{sub}}, P^s_{\text{porta}}, PT_s)$. É necessário que $(P^{s_i} \cup T^{s_i}) \cap (P^{s_j} \cup T^{s_j}) = \theta$ para todo $s_i, s_j \in S$ tal que $i \neq j$.
  2. $SM : T_{\text{sub}} \rightarrow S$ é uma função de sub-módulo que associa um sub-módulo a cada transição de substituição. É necessário que a hierarquia de módulo seja acíclica, ou seja, um sub-módulo não pode possuir transição de substituição associada a um sub-módulo superior na hierarquia.

[2] MS está relacionado a "multiconjunto".

3. PS é uma função de relação $porta\text{-}socket$ que associa uma relação $porta\text{-}socket$ $PS(t) \subseteq P_{\text{sock}}(t) \times P_{\text{porta}}{}^{\text{SM(t)}}$ a cada transição de substituição t. É necessário que $PT(p) = PT(p')$, $C(p) = C(p')$ e $I(p)\langle\rangle$ para todo $(p, p') \in PS(t)$ e $t \in T_{\text{sub}}$. Em síntese, a cada transição de substituição existe uma associação de uma relação $porta\text{-}socket$, de modo que o tipo do $socket$ deve ser igual ao tipo da $porta$ e a cor do $socket$ deve ser igual a cor da $porta$.

4. $FS \subseteq 2^P$ é uma família de conjuntos de fusão (*fusion sets*) não vazios tal que $C(p) = C(p')$ e $I(p)\langle\rangle = I(p')\langle\rangle$ para todo $p, p' \in fs$ e todo $fs \in FS$. Em outras palavras, é possível fundir dois lugares de modo que a marcação de ambos sejam sempre iguais. São semelhantes às variáveis globais conhecidas em muitas linguagens de programação.

Para a modelagem, simulação e execução de códigos *CPN/ML* em um modelo *CPN* pode ser utilizado a ferramenta *CPN/Tools*. No entanto, é possível lidar com modelos *CPN* sem exigir o uso de uma interface gráfica de usuário. A biblioteca *Access/CPN* permite que os usuários manipulem componentes de modelo, realizem simulações e execução de código *CPN/ML* usando a linguagem de programação Java (WESTERGAARD, 2011 [38]). Por exemplo, tal estrutura fornece métodos para disparar as transições habilitadas e obter fichas relacionadas com lugares específicos. Esse tipo de funcionalidade é relevante, por exemplo, para embutir os modelos *CPN* em *software* para treinamento de modeladores sobre como reutilizar a MBA/CPN.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados alguns trabalhos relacionados ao tema da pesquisa desta dissertação. Na Seção 3.1 são apresentados trabalhos sobre abordagens baseadas em modelos. Na Seção 3.2 alguns estudos sobre modularização, composição e reutilização de modelos formais são discutidos. Por último, na Seção 3.3 são apresentados modelos de bombas de infusão de insulina.

3.1 Abordagens Baseadas em Modelos

A aplicação de abordagens baseadas em modelos (*Model-Based Approaches - MBA*) é uma tendência no desenvolvimento de sistemas críticos seguros complexos. Por exemplo, Mian *et. al.* (2019) [23] apresentaram um arcabouço para tradução automática de modelos construídos com a linguagem de projeto e análise de arquitetura (*Architecture Analysis and Design Language - AADL*) em um formato executável por ferramentas de otimização e análise de confiabilidade baseadas em árvore de falhas. O núcleo do arcabouço proposto é a transformação de um modelo de erro baseado em máquina de estado para um modelo de árvore de falhas. O arcabouço foi implementado como um *plug-in* (denominado de *AADL2HiP-HOPS*) para a ferramenta de desenvolvimento de modelo *AADL OSATE* e como forma de ilustrar o seu uso um exemplo de transformação de modelo *AADL* de um sistema de monitoramento de temperatura foi apresentado. Os resultados provaram ser uma maneira eficaz de fornecer árvores de falhas para análise de confiabilidade.

Entezari-Maleki *et. al.* (2020) [10] descreveram uma rede de Petri colorida temporizada

(*Timed Coloured Petri Nets - TCPN*) para avaliar a composição do serviço web em ambientes multinuvem. *TCPN* foi utilizada para modelar o processo de submissão de requisições, análise de serviços compostos, seleção de serviços e provisionamento de serviços em um ambiente multinuvem. Para verificar a acurácia da *MBA*, o modelo proposto foi instanciado em dois diferentes cenários, sendo o primeiro um ambiente composto por três nuvens e cinco tipos de serviços diferentes e o segundo cenário um ambiente mais complexo formado por dez nuvens e quinze tipos de serviços distintos. Comparando os resultados alcançados com modelo *TCPN* em relação ao arcabouço *CloudSim*[1], verificou-se uma maior eficiência do modelo *TCPN* para a avaliação de desempenho da composição dos serviços.

Em outro estudo relacionado, Majma e Babamir (2020) [21] apresentaram um método de tempo de execução automático para verificação contínua do comportamento de um marcapasso controlado por um agente de *software* autônomo e inteligente. Este agente de *software* utiliza uma base de conhecimento para a tomada de decisões em tempo de execução, evitando falhas críticas que possam causar riscos irreversíveis ao paciente. A base de conhecimento consiste em um conjunto de regras que representaram o comportamento do marcapasso e foi modelada com redes de Petri colorida *fuzzy* hierárquica (*Hierarchical Fuzzy CPN - HFCPN*). Assim, o modelo *HFCPN* do marcapasso foi usado como mecanismo de inferência de regras para guiar o agente de *software* na tomada correta de decisões. Para validar a eficácia de encontrar uma regra adequada para tomada de decisão, o modelo foi instanciado para representar três diferentes cenários de uso. Os resultados mostraram uma notória melhoria na inferência de regras para o controle do marcapasso comparado a um mecanismo de inferência simples, com uma melhoria de até 92% no tempo gasto para identificar uma regra.

Por último, García-Valls, Perez-Palacin e Mirandola (2018) [13] propuseram uma abordagem composta por modelos paramétricos para o projeto de sistemas ciber-físicos adaptativos. A parametrização do modelo permitiu a sua configuração para representar condições ambientais distintas e situações que exigem a adaptação do sistema ao longo de sua execução. Redes de petri tradicionais foram usadas para a modelagem e verificação dos modelos. Para demonstrar a aplicabilidade em situações que exigem adaptações on-line limitadas no tempo, a *MBA* foi exemplificada em um caso de uso envolvendo um sistema de comunicação

[1]Arcabouço para modelagem e simulação de infraestruturas e serviços de Computação em Nuvem: https://github.com/Cloudslab/cloudsim

de veículos autônomos, que mostrou realizar adaptação pontual e satisfazer o tempo para as suas atividades nos diferentes modos de operação. Nesta pesquisa de mestrado, foi usada uma ideia semelhante, reutilizando modelos de sistemas de bombas de infusão de insulina paramétricos para avaliar os atributos de qualidade.

3.2 Modularização, Composição e Reutilização de Modelos

Montecchi, Lollini e Bondavalli (2020) [24] definiram formalmente um conceito de templates de modelo para auxiliar a definição de bibliotecas de submodelos genéricos reutilizáveis e a definição de modelos complexos baseados em estados pela composição e instanciação automática de submodelos. Cada elemento da metodologia proposta foi formalmente definido, tendo sido ainda fornecida uma linguagem de especificação de domínio, denominada *Template Models Description Language (TDML)*, para definição precisa das bibliotecas dos submodelos que compõem os modelos de análises concretos. Tais modelos concretos foram montados automaticamente a partir de um conjunto de regras de composição definidas em alto nível. Os autores aplicaram a *MBA* proposta usando um estudo de caso em um sistema distribuído de grande escala.

Por outro lado, Kanoun e Ortalo-Borrel (2000) [18] propuseram uma abordagem modular para modelar a confiabilidade de sistemas tolerantes a falhas usando rede de Petri estocástica generalizada (*Generalized Stochastic Petri Net - GSPN*) para submodelos e composição de modelos. Cada componente do sistema foi especificado de forma isolada com interfaces de comunicação entre os demais componentes claramente bem definidas. Assim, o modelo do sistema foi definido pela composição dos componentes *GSPN*. A abordagem modular em conjunto com o formalismo agregaram flexibilização e reutilização ao modelo, tendo sido aplicado com êxito para a seleção de novas arquiteturas para um sistema de controle de tráfego aéreo francês.

Ainda sobre estudos envolvendo modularização de modelos, Rabah e Kanoun (2003) [26] forneceram uma *MBA* para avaliar medidas de desempenho de sistemas multipropósitos e multiprocessadores usando modelos de arquitetura, modelos de nível de serviço e modelos de política de manutenção. Tais modelos separados possibilitaram análises dos sistemas sob diferentes perspectivas. Por sua vez, Nencioni, Helvik e Heegard (2017) [25] apresentaram

uma abordagem modular e sistemática para avaliação quantitativa das propriedades de redes definidas por *software* (*Software-Defined Networking - SDN*) considerando a correlação de falhas. Como parte dos resultados, os autores disponibilizaram modelos de confiabilidade para redes *SDN* que evitam complexidade, explosão de espaço de estados e altas demandas computacionais.

De forma similar aos estudos apresentados, neste trabalho, foi utilizado o conceito de modelos de referência como templates, aprimorando o estado da arte, integrando os modelos com a especificação de requisitos baseada em casos de garantia e avaliando o nível de compreensibilidade e adaptabilidade dos modelos reutilizáveis.

3.3 Modelos de Bombas de Infusão de Insulina

Com relação aos Sistemas de Bombas de Infusão de Insulina (SBII), pode-se destacar os trabalhos propostos por Silva *et. al.* (2015) [33] e Zhang, Jones e Jetley (2010) [40].

No primeiro trabalho, foi proposto por Silva *et. al.* (2015) uma abordagem baseada em modelos para validação de sistemas médicos físico-cibernéticos, focando em promover o reúso e a produtividade. Nessa abordagem três cenários clínicos diferentes foram modelados e avaliados, sendo um deles um cenário envolvendo uma bomba de infusão de insulina contemplando os principais requisitos e propriedades de segurança para esse tipo de sistema. Os diagramas de blocos do *Simulink*[2] foram utilizados para representar os comportamentos do sistema. No entanto, a modelagem não considerou a especificação de restrições de tempo para modelar o controle de administração da infusão de insulina.

Além disso, a *Food and Drug Administration - FDA* também se preocupa com os atributos de qualidade das bombas de infusão. O projeto de bomba de infusão genérica proposto por Zhang, Jones e Jetley (2010) [40] é um exemplo de iniciativa da *FDA* para aumentar a confiança nos SBII. Por exemplo, o projeto aborda a análise de risco por meio de uma especificação arquitetural genérica, porém trata-se de uma representação de alto nível que não permite a execução de uma instância de modelo considerando restrições de tempo e verificação formal. Um estudo relacionado, que não envolve bomba de infusão de insulina mas provê artefatos de projeto que podem também ser úteis nesse contexto é proposto por Hatcliff

[2]https://www.mathworks.com/products/simulink.html

et. al. (2019) [17]. Nele foi conduzido o projeto aberto de bomba de analgesia controlada pelo paciente para fornecer artefatos como casos de uso, infraestrutura de teste e simulação, artefatos de gerenciamento de risco e casos de garantia.

No entanto, não existe um modelo de SBII executável, genérico, paramétrico e temporizado para auxiliar os fabricantes na realização de análises detalhadas (por exemplo, controle de infusão e *recalls* comuns) durante o desenvolvimento e a certificação. A MBA/CPN aborda essa limitação, considerando um modelo de SBII executável, genérico, paramétrico e temporizado, que inclui controle de infusão e considera as diretrizes da *FDA*[3] e recalls relatados por Gao *et. al.* (2019) [12]. Como outra melhoria, na abordagem proposta é abordada a engenharia de requisitos orientada a metas com base em casos de garantia especificado com a notação estrutura por metas (*Goal-Structuring Notation - GSN*). Essas características são relevantes, por exemplo, para auxiliar na avaliação da qualidade de SBII em desenvolvimento ou identificar problemas de sistemas certificados. Assim, o estado da arte é aprimorado definindo uma *MBA* que fornece as seguintes características:

- definição de um padrão baseado em linguagem de marcação extensível (*EXtensive Markup Language - XML*) para aplicar casos de garantia *GSN* na engenharia de requisitos (nas primeiras etapas no processo de desenvolvimento);
- diretrizes para definição e disponibilização de módulos paramétricos, temporizados e executáveis de um modelo de referência de SBII para reutilização durante o processo de desenvolvimento (e integração com casos de garantia *GSN*);
- diretrizes para avaliação de atributos de qualidade de sistemas de bombas de infusão de insulina com base na reutilização de modelos;
- definição de um modelo de referência de SBII reutilizável, paramétrico, temporizado e executável.

[3]https://www.fda.gov/regulatory-information/search-fda-guidance-documents/infusion-pumps-total-product-life-cycle

Capítulo 4

MBA/CPN

Na Figura 4.1 é ilustrada uma visão geral da MBA/CPN, consistindo nas seguintes etapas de modelagem (M) e análise (A):

- modularização de *hardware* e *software* (M1);
- definição de modelo de referência e instanciação de modelo de referência (M2);
- análises de segurança (A1);
- análises de eficácia e geração de testes abstratos (A2).

Na primeira etapa de modelagem (M1), os fabricantes obtêm módulos de *hardware* e *software CPN* de fontes de requisitos. Na segunda etapa (M2), um modelo de referência é definido por dois refinamentos do sistema. O primeiro resulta em um modelo mais abstrato que não considera conjuntos de cores reais, enquanto o segundo refinamento gera um modelo mais detalhado considerando conjuntos de cores reais e restrições de tempo. A especificação de refinamento do sistema é relevante para a realização de análises sob duas perspectivas: segurança e eficácia. Assim, os fabricantes instanciam os refinamentos do modelo para analisar os comportamentos do sistema avaliando a qualidade. Na primeira etapa de análise (A1), o refinamento do primeiro sistema é utilizado para verificar as propriedades de segurança, evitando o problema da explosão do espaço de estados. Posteriormente, na segunda etapa de análise (A2), o modelo de referência verificado pode ser reduzido (removendo componentes de *hardware* e ajustando as marcações iniciais) para aplicar abordagens para geração

de testes abstratos. As mesmas propriedades devem ser verificadas novamente para o modelo reduzido para garantir a consistência. Finalmente, o refinamento do segundo sistema é usado para analisar as propriedades de segurança e eficácia na segunda etapa de análise. Os fabricantes documentam os requisitos dos Sistemas de Bomba de Infusão de Insulina (SBII) usando casos de garantia para todas as etapas.

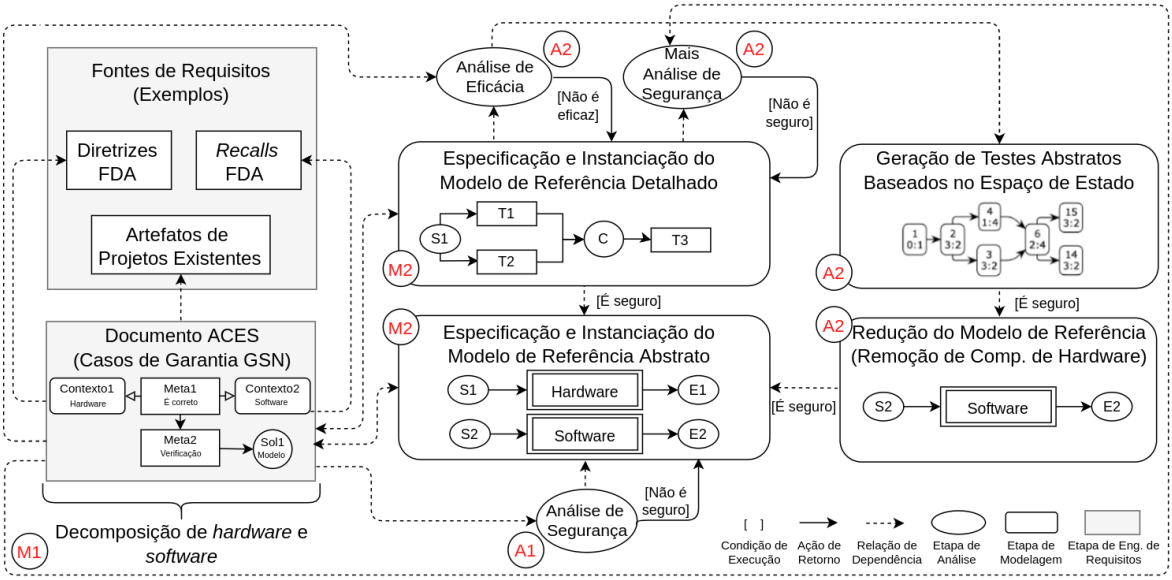


Figura 4.1: Visão geral da MBA/CPN para avaliação da qualidade de SBII

Neste trabalho é recomendado o uso de duas versões diferentes de modelos para reduzir o problema de explosão do espaço de estados, evitando a aplicação de técnicas mais complexas para redução do espaço de estados. A definição de um primeiro refinamento menos expressivo não impacta negativamente a abordagem, pois mantém as propriedades de segurança desejadas, verificadas pela técnica de verificação automática de modelos. É garantido que o modelo *CPN* representa a funcionalidade de uma especificação de acordo com as diretrizes da *Food and Drug Administration - FDA* através da realização de simulações de modelos. O primeiro refinamento do modelo de referência pode ser gerado automaticamente ou manualmente a partir da especificação baseada em casos de garantia. Neste trabalho são descritos modelos *CPN* especificados manualmente devido ao foco em fornecer uma base verificada e validada para outras extensões, conforme detalhado abaixo.

4.1 Decomposição de *hardware* e *software*

A MBA/CPN exige que os fabricantes desenvolvam casos de garantia com base nos requisitos derivados de fontes como sistemas semelhantes, revisões de literatura, *recalls* e diretrizes. Reivindicações, evidências e outros casos de garantia têm representações específicas na MBA/CPN usando a linguagem de marcação extensível (EXtensive Markup Language - XML) e a notação estrutura por metas (*Goal-Structuring Notation - GSN*). Neste trabalho, foi proposto e definido a extensão do padrão para troca de casos de garantia (*Assurance Case Exchange Standard - ACES*) para auxiliar fabricantes e agências reguladoras na especificação e troca de casos de garantia durante os processos de desenvolvimento e certificação. Por exemplo, o *ACES* inclui recursos que permitem aos fabricantes realizar a rastreabilidade dos requisitos do sistema. A utilização do *ACES* é o ponto de partida para a aplicação da MBA/CPN, e continua a ser utilizado durante todo o processo de desenvolvimento devido à sua ligação aos modelos de referência *CPN*. Assim, em vez de documentar e verificar metas aplicando uma abordagem clássica, como manter todos os objetivos satisfeitos (ou seja, abordagem KAOS), para conduzir a engenharia de requisitos orientada a metas, deve ser utilizado o *ACES*.

A especificação *XML* é a base para associar todas as etapas restantes da MBA/CPN. O *ACES* considera os principais conceitos do processo de engenharia de requisitos baseado em *GSN* modular. Um documento *ACES* contém, pelo menos, as notações gráficas definidas na especificação *GSN*. Cada notação gráfica dos elementos *GSN* tem uma representação no *ACES*, relacionando os elementos a *tags* e atributos específicos de um documento *ACES*. Por exemplo, o elemento de evidência contém um *link* de atributo para permitir que as agências reguladoras acessem um artefato de projeto fornecido pelos fabricantes do sistema para oferecer suporte a um argumento. O uso de casos de garantia baseados em uma plataforma padrão independente e bem definida para representar e compartilhar os resultados obtidos pelos fabricantes com os órgãos reguladores pode melhorar o projeto e a avaliação dos sistemas.

O início e o fim de um documento *ACES* é a *tag* <*assuranceCase*>. Para habilitar o controle de versão, cada documento possui uma identificação geral denominada *generalId*, juntamente com a identificação da versão (*versionId*) e a identificação local (*localId*). O

atributo *generalId* da *tag* <*assuranceCase*> é um identificador único para todas as versões do documento *ACES*, enquanto o *versionId* e *localId* possuem identificadores diferentes para cada nova versão para representar as modificações no documento. O início do documento também contém dados específicos sobre o produto em desenvolvimento por meio da *tag* <*device*>.

Considerando que os casos de garantia contêm um conjunto de argumentos relacionados sobre os atributos de qualidade de um sistema (por exemplo, segurança e eficácia), o *ACES* os representa usando as *tags* <*parentArgument*> e <*childArgument*>. A *tag* <*parentArgument*> compõe o corpo da *tag* <*assuranceCase*> e representa a estrutura principal do caso de garantia na *GSN* modular. Em contraste, a *tag* <*childArgument*> representa estruturas que são partes do corpo da *tag* <*parentArgument*>. Nesse caso, um documento *ACES* contém apenas um <*parentArgument*> que pode consistir em vários módulos de caso de garantia: os argumentos filho estão relacionados a módulos *GSN* específicos. Cada argumento *ACES* contém a *tag* <*legalAuthenticator*>, visando registrar o autor das modificações no documento *ACES*.

A especificação *ACES* inclui os principais elementos *GSN*: *Goal*, *Solution*, *Strategy*, *Context*, *Assumption*, *Justification*, *SupportedBy* e *InContextOf*. Além disso, inclui os elementos modulares *GSN*: *Away Goal*, *Module*, *Contract*, *Away Solution*, *Away Context* e *Public Indicator*. O *ACES* estrutura esses elementos em seu corpo usando a *tag* <*group*>. Este elemento tem o atributo denominado *type*, que o restringe a agrupar elementos *GSN* do mesmo tipo. Por exemplo, para representar um *Goal*, é necessário definir uma *tag ACES* para representar um objetivo específico no corpo da *tag* <*group*>. Cada *tag* <*group*> relacionada a um tipo de *tag ACES* é definida apenas uma vez no corpo de cada *tag* <*parentArgument*> e <*childArgument*>.

A *tag* <*goal*> representa um elemento *GSN Goal*. Todos os elementos *GSN* (exceto relacionamentos) contêm, pelo menos, um atributo chamado *id* e uma *tag* filha chamada <*description*>. A *tag* <*goal*> também pode conter os atributos opcionais denominados *public*, *undevelopment* e *toBeSupportedByContract*. Portanto, as metas representam reivindicações de caso de garantia, apoiadas por um conjunto de submetas. No *ACES*, as metas também podem representar requisitos, quando o atributo denominado *requirement* é definido como *true*. Ele permite que os fabricantes documentem os requisitos de qualidade usando o *ACES*. Os

fabricantes podem documentar artefatos de produtos relacionados a esses requisitos usando soluções *GSN*. Para cada objetivo de um módulo *GSN*, um módulo *CPN* (especificação *XML*) ou uma fórmula lógica temporal pode ser incorporada no documento *ACES* (*tag* <*formalDefinition*>) para manter a descrição formal dos requisitos. A *tag* <*formalDefinition*> pode conter as *tags required* e *provided* (interfaces) para permitir a especificação da composição do módulo.

A *tag* <*solution*> define uma solução *ACES* para representar evidências que dão suporte a declarações. O atributo denominado *artifact*, quando definido como *true*, associa a solução a um artefato do produto. Uma *tag* chamada *externalArtifactUrl* conecta uma solução a uma evidência específica. A definição de soluções como artefatos do produto é relevante para possibilitar a rastreabilidade dos requisitos. Para raciocinar sobre conexões entre reivindicações (possivelmente requisitos), os fabricantes usam *Estratégias*. A *tag* <*strategy*> representa uma estratégia que contém um atributo opcional adicional chamado *undevelopment*. Outra característica importante da engenharia de requisitos considerada usando o *ACES* é a fonte de requisitos. Para casos de garantia, o elemento *GSN Context* fornece informações sobre declarações específicas, representadas no *ACES* usando a *tag* <*context*> (com o atributo adicional denominado *public*). No *ACES*, os elementos *GSN Context* definem a origem dos requisitos, definindo o atributo denominado *source* como *true*. Quando este atributo é *true*, uma nova *tag* chamada <*externalSourceUrl*> associa a fonte ao local da fonte declarada. Definir contextos como fonte de requisitos também é relevante para realizar a rastreabilidade de requisitos utilizando o *ACES*.

Também pode ser necessário melhorar a confiança na validade de reivindicações e estratégias usando o elemento *GSN Assumption*, definido pela *tag ACES* <*assumption*>. Além disso, os fabricantes podem fornecer justificativas sobre a definição de reivindicações e estratégias. Portanto, a *tag ACES* <*justification*> representa o elemento *GSN Justification*. Na engenharia de requisitos orientada por metas baseada no *ACES*, a *tag* <*justification*> permite que os fabricantes justifiquem as mudanças nos requisitos. As justificativas adicionam informações na versão obsoleta do requisito definido no documento *ACES* (controle de versão), ou seja, uma justificativa é anexada à *tag* <*goal*> utilizada para representar o requisito obsoleto.

O *ACES* representa conexões entre elementos com a *tag* <*relationships*>, contendo pelo

menos uma *tag* filha. A *tag* <*relationSupportedBy*> é uma notação de ligação usada para indicar relacionamentos entre requisitos e artefatos do projeto (evidência), exigindo os atributos *id*, *type* e *relID*. O atributo *relID* é o identificador que relaciona os elementos *GSN*, respeitando as regras definidas no padrão *GSN*. Há também uma notação de ligação usada para indicar relacionamentos contextuais usando o elemento <*relationInContextOf*> e também contém atributos denominados *id*, *type* e *relID*. Os elementos <*relationSupportedBy*> e <*relationInContextOf*> fazem parte do corpo da *tag* <*relationships*>.

Para avaliação da qualidade, os órgãos reguladores podem incluir os resultados da avaliação no documento *ACES* em análise pelas *tags* <*accepted*> e <*rejected*>. A avaliação dos documentos do *ACES* diz respeito a argumentos individuais. O corpo da *tag* <*rejected*> contém uma descrição da rejeição. Fabricantes e órgãos reguladores podem trocar documentos até que seja dada uma decisão final sobre a certificação do sistema que está em avaliação. Por exemplo, a agência reguladora pode solicitar provas específicas antes da aprovação do sistema. No repositório GitHub da MBA/CPN[1] é fornecida uma descrição mais detalhada da especificação *ACES* como material suplementar a esta dissertação.

Conforme destacado, os casos de garantia baseados no *ACES* permitem que os fabricantes realizem a rastreabilidade dos requisitos e a verificação dos requisitos regulatórios. No entanto, existem elementos de casos de garantia que não desempenham um papel fundamental durante essas atividades. É possível representar formalmente os documentos *ACES* com base nos casos de garantia mais relevantes para rastreabilidade de requisitos. São consideradas apenas metas (<*goal*>) e soluções (<*solution*>), representando requisitos e artefatos do projeto, respectivamente. Um caso de garantia baseado em *ACES* é um grafo orientado $T = (V, A)$, onde V é um conjunto de nós relacionados a objetivos e soluções e A é um conjunto de arestas que conectam nós. Formalmente, é definido como uma tupla de 5 elementos $ACES = (V_g, V_s, v_r, A, R)$, onde

- V_g é um conjunto de nós definidos como metas;
- V_s é um conjunto de nós definidos como soluções, tais que para todos os $v_s \in V_s$ seu grau é 1;
- $V_g \cup V_s$ é um conjunto de nós V de um grafo conexo acíclico T, tal que $V_g \cap V_s = \theta$;

[1] https://github.com/alvarosobrinho/mbacpn

- $A \subseteq V_g \times V_g \cup V_g \times V_s$ é um conjunto de arestas de um grafo conexo acíclico T;
- $v_r \in V_g$ é um nó específico chamado raiz; e
- R é uma função $R : V_g \cup V_s \rightarrow 2^D$ (D são descrições de nós). Cada nó v está relacionado a um conjunto $R(v)$ de descrições (por exemplo, fonte de requisitos).

Na Figura 4.2 é ilustrada a relação entre fabricante e agência reguladora usando a MBA/CPN. O fabricante e o regulador acessam dois repositórios: Repositório de Modelos *CPN* e Repositório de Evidências. Eles mantêm módulos *CPN* de modelos de referência e evidências em tempo real, respectivamente. Os fabricantes buscam módulos de sistema existentes em desenvolvimento no repositório para acessar e compor módulos, gerando versões específicas de sistemas. Quando não há um modelo de referência (ou módulo) disponível que se encaixe no sistema, é necessário gerar um novo modelo que poderá ser publicado no repositório. A agência reguladora avalia os casos de garantia baseados no *ACES* analisando argumentos com evidências vinculadas (publicadas no repositório). O regulador pode verificar o modelo do sistema disponível em tempo real.

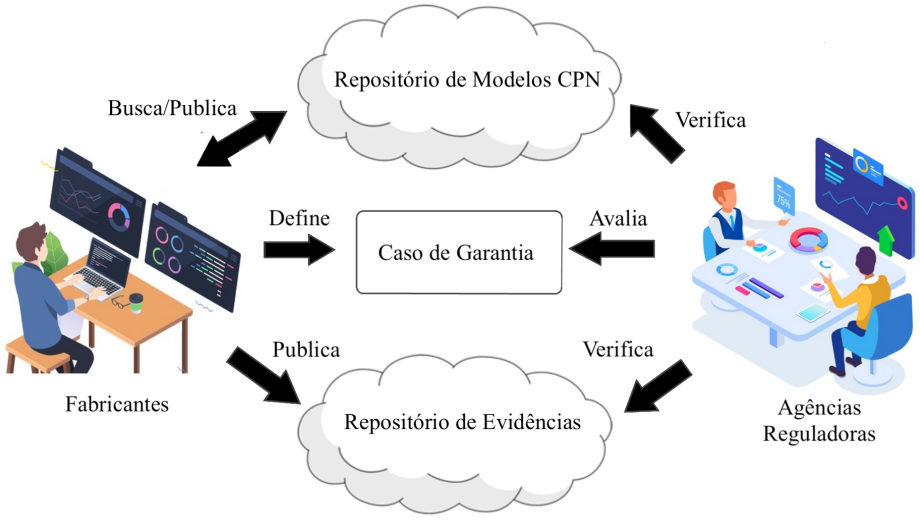


Figura 4.2: Relacionamento entre fabricante e agência reguladora.

Por exemplo, na Figura 4.3 é ilustrado o nível mais alto do caso de garantia para os SBII. O elemento *GSN* denominado *SYSTEM-G1* representa uma afirmação sobre a segurança e eficácia do sistema em desenvolvimento. A meta está relacionada ao contexto de *software*

(*SYSTEM-C1*) e *hardware* (*SYSTEM-C2*). As estratégias suportam argumentos sobre os requisitos do processo (*SYSTEM-S1*) e do produto (*SYSTEM-S2*). Os módulos denominados *SYSTEM-M1* e *SYSTEM-M2* contêm argumentos seguindo essas estratégias. Na Figura 4.4 é apresentada uma amostra do documento *ACES* para o caso de garantia apresentado na Figura 4.3, composto por alguns dos elementos básicos da *GSN*.

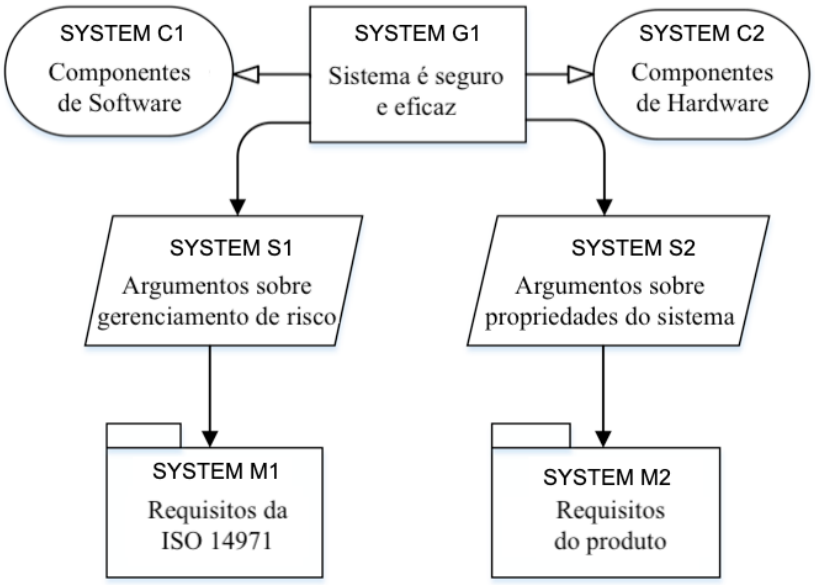


Figura 4.3: Caso de garantia *GSN* de nível mais alto de SBII.

4.2 Primeiro Refinamento do Sistema

Cada elemento meta de um documento *ACES* que contém a *tag* <*formalDefinition*> está relacionado às especificações *CPN* considerando os refinamentos do sistema. O primeiro refinamento de sistema do modelo de referência de SBII possui dois módulos principais, representando todo o sistema com base nos requisitos de *hardware* e *software*. Assim, a modelagem segue as estruturas arquiteturais de decomposição e uso do módulo.

4.2.1 Módulo de *hardware*

O modelo de referência completo é disponibilizado no repositório da MBA/CPN[2] devido à limitação de tamanho das figuras e para disponibilizá-lo para reutilização e simulação. Na

[2]https://github.com/alvarosobrinho/mbacpn

```
...
<parentArgument>
    <group type="goal">
        <goal id="SYSTEM-G1">
            <description>Sistema é seguro e eficaz</description>
            <relationships>
                <relationIncontextOf id="r1" type="context" relId="SYSTEM-C1" />
                <relationIncontextOf id="r2" type="context" relId="SYSTEM-C2" />
                <relationSupportedBy id="r3" type="strategy" relId="SYSTEM-S1" />
                <relationSupportedBy id="r4" type="strategy" relId="SYSTEM-S2" />
            </relationships>
        </goal>
    </group>
    <group type="context">
        <context id="SYSTEM-C1">
            <description>Componentes de Software</description>
        </context>
        <context id="SYSTEM-C2">
            <description>Componentes de Hardware</description>
        </context>
    </group>
    <group type="strategy">
        <strategy id="SYSTEM-S1">
            <description>Argumentos sobre gerenciamento de risco</description>
            <relationships>
                <relationSupportedBy id="r5" type="module" relId="SYSTEM-M1" />
            </relationships>
        </strategy>
        <strategy id="SYSTEM-S2">
            <description>Argumentos sobre propriedades do sistema</description>
            <relationships>
                <relationSupportedBy id="r6" type="module" relId="SYSTEM-M2" />
            </relationships>
        </strategy>
    </group>
    <group type="module">⋯
    </group>
</parentArgument>
...
```

Figura 4.4: Amostra da especificação *ACES* para o caso de garantia apresentado na Figura 4.3.

Figura 4.5 é ilustrado o módulo de *hardware*, ou seja, o ponto de partida da decomposição de *hardware* modular intermediário. O módulo de *hardware* envia mensagens para um módulo de *software* por duas interfaces de saída chamadas *Events_H* e *States_H*, representando trocas de mensagens (isto é, passagem de valores entre os módulos) não valoradas e valoradas. A configuração básica do *hardware* é composta por um botão de inicialização, uma bateria e um cartucho. No entanto, os fabricantes podem configurar o número de componentes de *hardware* usando o recurso de modularização do *CPN*, por exemplo, adicionando mais baterias ao sistema especificado. O modelo de referência inclui essa tática de redundância arquitetural para auxiliar os fabricantes a lidar com *recalls* comuns da *FDA* causados por mau funcionamento da bateria. *Recalls* relacionados com baterias podem resultar em problemas como perda de dados, perda de comunicação, interrupção abrupta da terapia e superinfusão

e subinfusão[3] (GAO *et. al.*, 2019 [12]).

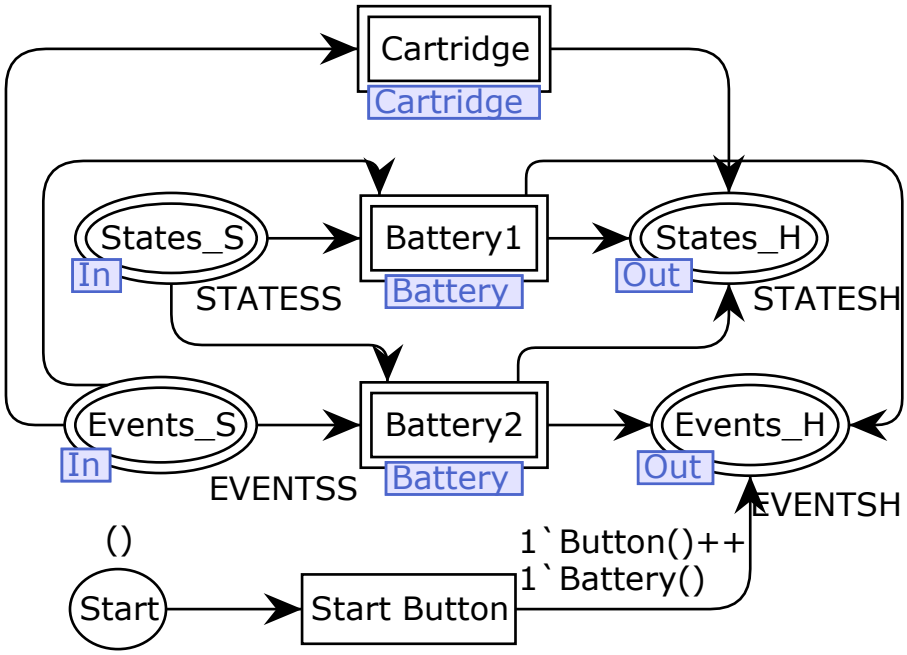


Figura 4.5: Módulo de *hardware* de SBII com duas baterias.

Os recursos de modularização do *CPN* permitem a reutilização do mesmo módulo de bateria para representar vários componentes de bateria. O *status* da bateria é definido como carregado (1) ou descarregado (0) usando um lugar *Value*, e é enviado ao *software* como mensagens de valor (lugar *States_H*). Quando a bateria é recarregada, uma mensagem é enviada ao *software* usando o lugar *Events_H*. Um lugar de fusão é usado para parar a bomba quando ocorre um mau funcionamento. No modelo, também é apresentado o módulo cartucho da bomba, responsável por registrar a capacidade máxima de insulina.

4.2.2 Módulo de *software*

O módulo do *software* foi dividido em três partes relacionadas às etapas de decomposição modular intermediária: verificação da bateria, configuração da bomba e infusão de insulina. A verificação da bateria começa a partir da transição *Battery Situation* (Figura 4.6), enviando uma mensagem ao *hardware* para obter o estado atual da bateria. O *software* recebe

[3]https://www.fda.gov/regulatory-information/search-fda-guidance-documents/infusion-pumps-total-product-life-cycle

uma mensagem de valor do *hardware* (*ValueCharge(b1)*) com o *status* atual e verifica se a bateria está carregada o suficiente para continuar funcionando. Caso contrário, além de liberar o evento de verificação da bateria, o *software* libera a recarga da bateria enviando uma mensagem *Recharge(1)* para o módulo *Battery*. Caso haja mais de uma bateria, o *software* envia a notificação e continua funcionando normalmente. A marcação inicial do lugar de fusão chamado *Ok_B* especifica que o sistema contém apenas uma bateria. Se houver mais de uma bateria, o *software* pode enviar uma mensagem de recarga para uma bateria descarregada e continuar usando as carregadas. Quando todas as baterias estão descarregadas, a bomba para de funcionar, reativada após carregar pelo menos uma bateria. A marcação inicial do lugar de fusão *Ok_B* especifica que o sistema contém apenas uma bateria disponível.

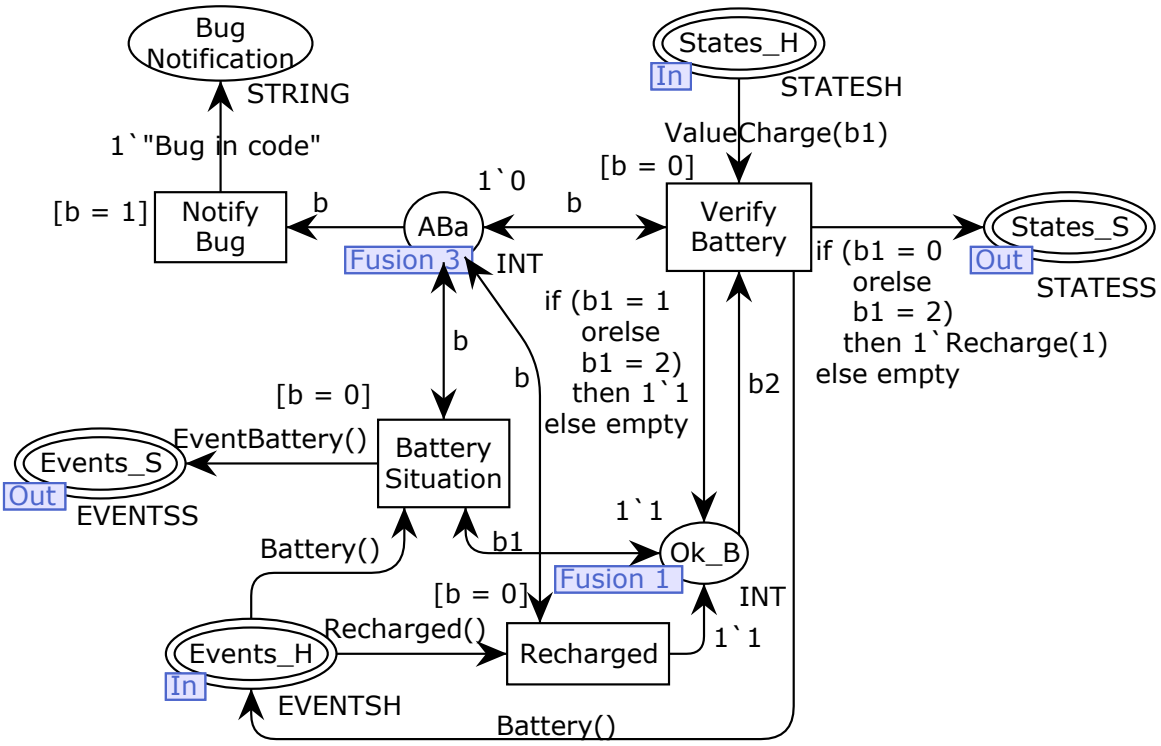


Figura 4.6: Módulo *Verify Battery* do primeiro refinamento do sistema.

Para começar a utilizar o sistema, o usuário deve configurar a bomba definindo o perfil que orienta as operações: padrão ou personalizado. A configuração dos valores para dosagens de insulina muda de acordo com o perfil considerado. O *software* envia uma mensagem ao *hardware* solicitando a capacidade do cartucho atual para iniciar a configuração da bomba. A capacidade e as dosagens de insulina para corretivo basal, bolus e bolus corretivo

são enviadas para um lugar denominado *DC*. As constantes *BASAL*, *BOLUS* e *CBOLUS* são usadas para configurar valores de dosagem predefinidos. Assim, o sistema só pode administrar dosagens de insulina quando essas regras forem cumpridas.

Para o modo personalizado, é necessário definir diariamente valores para diferentes dosagens. Um arquivo de texto, identificado por meio de uma constante *FILE*, carrega as dosagens de configuração para definir os valores do modo de infusão como estruturas de dados de lista, dependendo da prescrição médica para um caso clínico específico. O restante da configuração segue a mesma abordagem do modo padrão. Ao final da configuração, duas listas representam as dosagens basais e em bolus (denominadas *listBa* e *listDC*), sendo as entradas para a etapa de infusão de insulina.

Além de descrever a etapa de configuração do SBII, o modelo de referência contém especificações dos modos de infusão padrão e personalizado. Por fim, o modelo inclui um submódulo para a infusão padrão. A primeira etapa consiste em registrar os dados de configuração, notificando também que a bomba está em execução e carregar o cartucho. Posteriormente, é possível selecionar um modo específico disparando uma das transições chamadas *Adm Basal*, *Adm CBolus* e *Adm Bolus*. O usuário da bomba pode escolher um ou mais tipos de insulina durante a execução do sistema. Caso a soma de todas as dosagens não ultrapasse um limite de segurança, o sistema aplica uma dosagem única composta por todas as insulinas; caso contrário, o sistema aplica a quantidade máxima permitida de dosagem de insulina.

Uma vez que o sistema esteja de acordo com todas as pré-condições, ele aplica a insulina e atualiza a capacidade do cartucho. Quando não há mais insulina, o sistema transita para um estado que indica a necessidade de preenchimento do cartucho. O sistema também atinge este estado quando a capacidade do cartucho está abaixo da dosagem atual de insulina que está sendo aplicada. Para ambas as situações (ou seja, quando o cartucho está zerado ou seu nível está abaixo da dosagem a ser aplicada), a bomba transita do estado parcial de execução para o estado parcial que indica a necessidade de recarga do cartucho. Quando o usuário recarrega a bomba, o sistema volta ao estado parcial de execução.

Quando ocorrem falhas, o sistema deve parar de funcionar imediatamente, informando o usuário sobre o mau funcionamento. Três tipos de falhas de *software* são críticas para o funcionamento da bomba devido aos riscos à segurança dos usuários:

- falhas que ocorrem quando as dosagens de insulina estão sendo selecionadas (nenhuma dosagem foi selecionada);
- falhas que ocorrem quando as dosagens de insulina estão sendo selecionadas (pelo menos uma foi selecionada);
- falhas que ocorrem quando as dosagens de insulina estão sendo aplicadas.

O modo de infusão de insulina personalizado se comporta de forma semelhante ao modo padrão; porém, ao invés dos valores constantes das dosagens, o modo personalizado permite que o usuário defina as dosagens desejadas a partir de um arquivo de texto. Este modo aplica estruturas de dados de lista para representar as dosagens de insulina. Ele os converte no formato de tupla original, removendo a primeira dosagem basal da lista chamada *listBa* para adicionar a dosagem no topo da lista chamada *listDC*. Um lugar é responsável por registrar as demais dosagens basais a serem aplicadas.

A especificação permanece quase igual ao modo padrão, onde a mudança mais significativa diz respeito ao uso das restantes dosagens basais, repetindo a infusão até que não existam fichas de dosagens basais representando a insulina restante. A especificação do processo de recarga do cartucho também é quase igual ao modo padrão, acrescentando novos lugares e transições para controlar três situações:

- o sistema aplicou a dosagem basal atual e é necessário aplicar o restante;
- é necessário aplicar a dosagem basal vigente e há dosagens remanescentes registradas;
- não há dosagem basal atual.

O *BASAL*, *BOLUS*, *CBOLUS*, *CAPCART*, *UPPERDOSELIMIT*, *LOWERDOSELIMIT*, *INFUSIONLIMIT* e *QTDBASAL* são parâmetros de entrada inteiros para configurar a bomba, enquanto o parâmetro *FILE* recebe uma *string* que representa o nome do arquivo contendo os valores de dosagem de insulina basal. Este refinamento do modelo consiste em tipos de dados inteiros para reduzir o problema de explosão do espaço de estados, simplificando a execução da técnica de verificação automática de modelos. Em um artigo científico publicado por Costa *et. al.* (2019) [6] foram relatados resultados preliminares relacionados ao primeiro refinamento. Entretanto, a partir dos resultados preliminares publicados, foram realizados aprimoramentos considerando as diretrizes da *FDA* e *recalls* comuns.

4.3 Segundo Refinamento do Sistema

O segundo refinamento do modelo de referência de SBII fornece uma versão estendida do primeiro refinamento do sistema, incluindo restrições de tempo e valores de tipo de dados reais. Este refinamento é composto pelos mesmos módulos de nível superior do primeiro refinamento; no entanto, a especificação melhora o modelo de referência com uma representação mais detalhada dos SBII para cumprir os requisitos regulatórios rigorosos (por exemplo, a taxa correta de infusão) com tipos de dados reais e temporizados.

A principal modificação compreende a verificação da bateria, controle de administração e infusão de insulina, usando os tipos de dados temporizados chamados *UNITTIMED*, *REALTEMP* e *DATATEMP*. Esses tipos de dados permitem que os fabricantes manipulem tipos de dados de unidades temporizadas, tipos de dados reais temporizados e um produto de valores inteiros e reais. Nesse refinamento, a verificação da bateria é guiada pelo controle da frequência de verificação, realizada a cada 2 unidades de tempo, em vez de liberar a próxima verificação assim que a verificação atual for realizada. O controle de administração é semelhante aos modos de infusão padrão e personalizado. Os refinamentos do modelo de referência estão disponíveis no GitHub[4] para os leitores que desejam analisar e reutilizar as especificações, juntamente com um exemplo de modelo de caso de garantia baseado no *ACES*. Na Figura 4.7 é apresentado o segundo refinamento do modelo para a infusão padrão, descrevendo o controle de administração, a partir da transição *Prepare Partial Application* até a transição *Finish Total App*.

Uma taxa de administração definida pelo usuário orienta o controle de administração e, portanto, considerando uma dosagem de insulina, não significa que a bomba aplique a dosagem de uma só vez. O sistema divide a dosagem para agendar a infusão com base na taxa de administração, e a infusão depende do número de unidades de insulina permitidas. Por exemplo, o modelo representa o controle de administração de insulina basal utilizando o lugar *TABA*, que mantém o tempo de administração de insulina basal. Uma vez que o sistema conduz a infusão de insulina basal, a transição *Release Basal1* é acionada, obtém o próximo tempo de infusão de insulina do lugar *TABA* e configura a próxima dosagem basal. Além disso, o sistema envia o próximo horário para realizar a infusão de insulina para o lugar

[4]https://github.com/alvarosobrinho/mbacpn

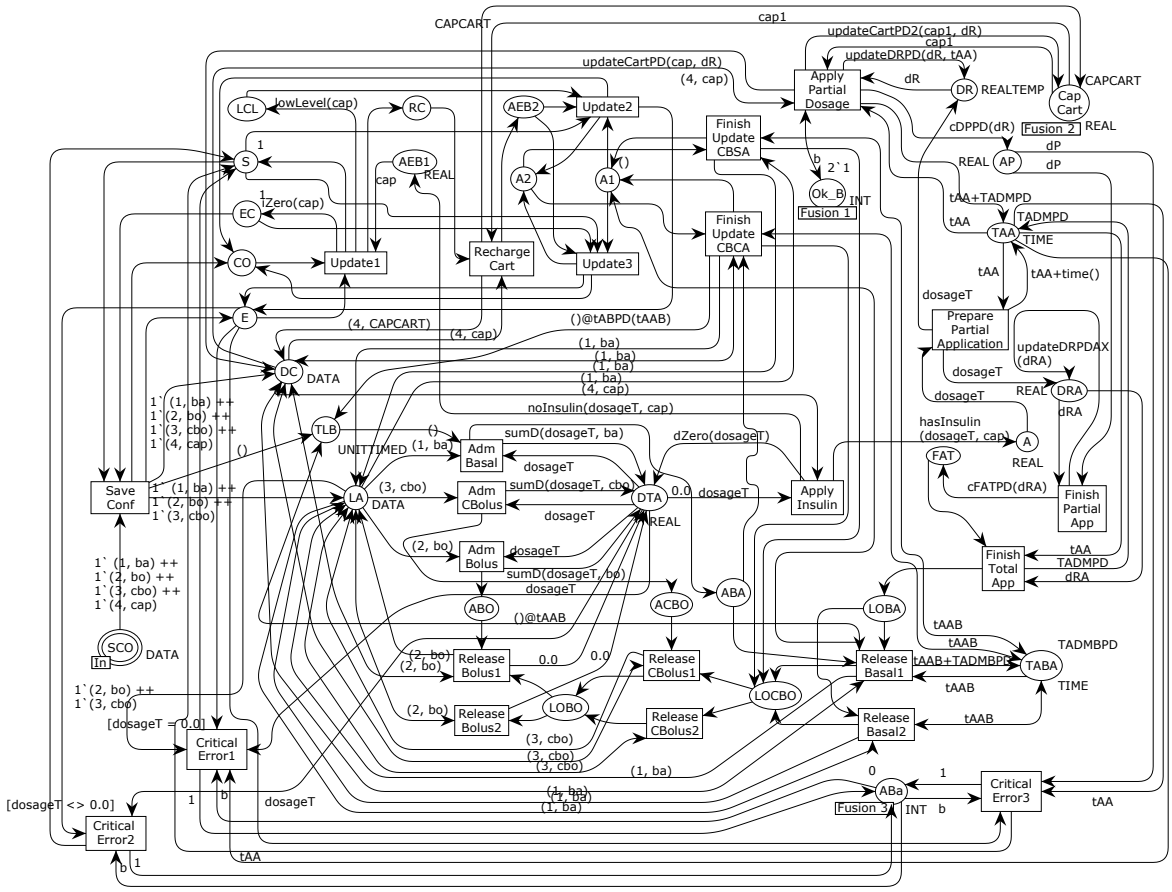


Figura 4.7: Submódulo *Standard Infusion* do segundo refinamento do sistema.

TABA, possibilitando a configuração correta para a infusão basal de insulina.

O segundo refinamento mantém os mesmos parâmetros usados para o primeiro refinamento. No entanto, exceto para o parâmetro *QTDBASAL*, os parâmetros são atribuídos com os tipos de dados reais. O refinamento também inclui seis novos parâmetros de entrada (ou seja, *DADMPD*, *TADMPD*, *TADMPB*, *DADMPS*, *TADMPS* e *TADMPS*) relacionados a taxa de administração e o tempo.

Capítulo 5

Cenários de Avaliação de Qualidade

O modelo de referência é instanciado para representar os requisitos técnicos do ACCU-CHEK Spirit, apresentando cenários de avaliação de qualidade de sistemas de bombas de infusão de insulina usando verificação automática de modelos e simulações. Cada elemento *Solution* de um documento especificado com o padrão para troca de casos de garantia (*Assurance Case Exchange Standard - ACES*), definido como um artefato, vincula-se aos resultados de verificação e validação gerados usando o modelo de Sistemas de Bomba de Infusão de Insulina (SBII).

5.1 Verificação do Primeiro Refinamento do Sistema

A verificação diz respeito à avaliação da qualidade usando propriedades de segurança. Um arquivo chamado *BasalV2AC.txt* contém 24 valores hipotéticos de dosagens basais para o modo de infusão personalizado. Esta atividade consiste em verificar duas propriedades de segurança dos SBII que fazem parte da especificação do ACCU-CHEK Spirit. A técnica de verificação automática de modelos *CPN* é baseada na *Computation Tree Logic* (CTL). As funções disponíveis na biblioteca ASK/CTL do *CPN/Tools* representam as especificidades das propriedades.

A primeira propriedade define que quando o nível do cartucho for 0, a bomba deve ser parada (AF$\neg(cartZeroed) \vee$ AF$(pumpStop \wedge cartEmpty)$). O verificador de modelo ASK-CTL verificou que o modelo atendeu a essa propriedade. O modelo de referência está de acordo com a primeira propriedade dos SBII para os modos padrão e personalizado. A se-

gunda propriedade define que a bomba não pode funcionar quando a dosagem de insulina for maior que o nível do cartucho (AF¬($cartLevel$) ∨ AF($pumpStop$)). O verificador do modelo ASK-CTL também verificou que o modelo está de acordo com a segunda propriedade dos SBII. Assim, estes são dois exemplos de artefatos de projeto associados a elementos de solução do documento *ACES*.

Seguindo a MBA/CPN, o modelo de referência foi reduzido removendo componentes de *hardware* e ajustando as marcações iniciais. O modelo foi verificado novamente para garantir conformidade. Testes abstratos foram gerados para guiar a validação. Para ilustrar a geração do teste abstrato, foi aplicada a ferramenta *MBT/CPN* (WANG *et. al.*, (2019) [36]) usando o espaço de estado do modelo de SBII, resultando em oito testes abstratos discutidos na próxima seção.

5.2 Validação do Segundo refinamento do Sistema

Os recursos de simulação do *software CPN/Tools* possibilitaram a validação do segundo refinamento, analisando funcionalidades relacionadas às restrições de tempo do modelo. As simulações foram usadas para realizar avaliações de qualidade das propriedades de segurança e eficácia. A convenção de formato adotada para representar as restrições de tempo é uma unidade de tempo para 1 segundo, 60 unidades de tempo para 1 minuto e 3600 unidades de tempo para 1 hora.

O parâmetro chamado *FILE* possui o tipo de dado *string* e contém o nome do arquivo utilizado para carregar as dosagens de insulina para o modo personalizado do modelo de referência. Na Figura 5.1 é ilustrada uma amostra desta etapa de simulação. Para a infusão basal de insulina, o modelo obtém a próxima dosagem basal do lugar *DBR* e a envia para o lugar *Next Basal* disparando a transição *Release Basal1*. Então, com o disparo da transição *Release Basal2*, o modelo gera a nova dupla de dosagem basal e a envia para os lugares denominados *LA* e *DC*, compreendendo o próximo tempo para a dosagem basal. Dado que os valores de dosagem de insulina em bolus e bolus corretivo não possuem taxa de administração e são constantes, a etapa de liberação consiste apenas em disparar as transições denominadas *Release CBolus1* e *Release Bolus1*, habilitadas devido à infusão de bolus e dosagens de bolus corretivas. Caso contrário, o modelo habilitaria as transições chamadas

Release CBolus2 e *Release Bolus2*. O lugar *LA* carrega um tipo de dado de tempo real para registrar as 2-tuplas com a hora do relógio atual, mantendo os valores de configuração dos dois tipos de insulina (lugar *DC*).

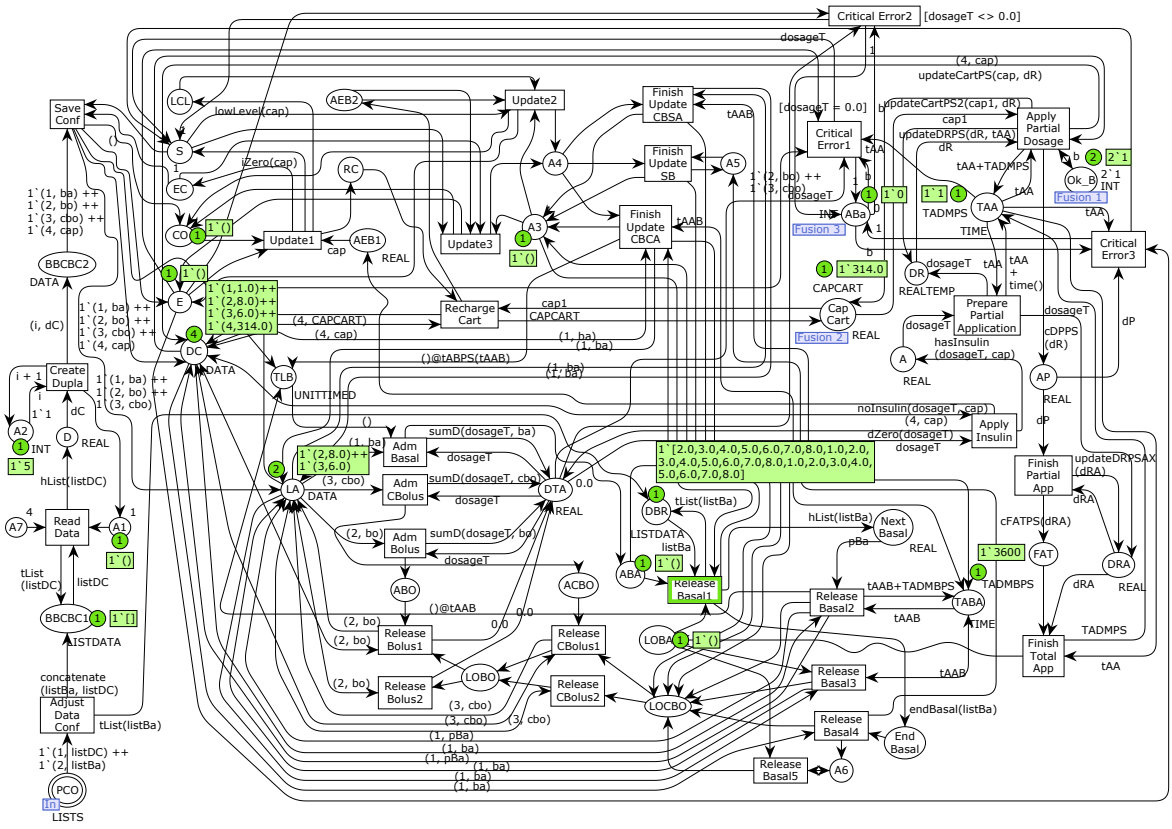


Figura 5.1: Amostra de simulação do submódulo *Personalized Infusion*, representando o ACCU-CHECK Spirit.

O disparo da transição *Basal Read* chama uma função que lê os valores de dosagem basal de um arquivo e os adiciona ao lugar *LBD* como uma lista. O modelo verifica cada valor com base em limites superiores e inferiores predefinidos, utiliza a variável *listBa* para transportar as dosagens basais e adiciona-as ao lugar *DBV*, aguardando a verificação dos valores restantes. Na Tabela 5.1 são descritas as simulações (mais de 3.000 etapas) realizadas para avaliar a completude e conformidade do modelo com base nos oito testes abstratos gerados com a ferramenta *MBT/CPN*. Além disso, foi realizado um grande número de simulações aleatórias para aumentar a confiança, considerando outras características como cargas de bateria desejadas e indesejadas. Os resultados da simulação são outros exemplos de artefatos de projeto associados às soluções no *ACES*.

Tabela 5.1: Descrição das simulações realizadas com base em testes abstratos.

Id	**Descrição**
1	A bomba funciona corretamente (modo de infusão padrão).
2	A bomba funciona corretamente (modo de infusão personalizado).
3	A bomba terminou a execução devido a uma falha crítica de *software* quando as doses de insulina foram selecionadas no modo de infusão padrão (nenhuma dosagem selecionada).
4	A bomba terminou a execução devido a uma falha crítica de *software* quando as doses de insulina foram selecionadas no modo de infusão padrão (pelo menos uma selecionada).
5	A bomba terminou a execução devido a uma falha crítica de *software* quando as doses de insulina estão sendo aplicadas no modo de infusão padrão.
6	A bomba terminou de funcionar devido a uma falha crítica de *software* quando as doses de insulina foram selecionadas no modo de infusão personalizado (nenhuma dosagem selecionada).
7	A bomba terminou a execução devido a uma falha crítica de *software* quando as doses de insulina estão sendo selecionadas no modo de infusão personalizado (pelo menos uma selecionada).
8	A bomba terminou de funcionar devido a uma falha crítica de *software* quando as doses de insulina estavam sendo aplicadas no modo de infusão personalizada.

Capítulo 6

Avaliação Empírica

Visando responder a pergunta principal de pesquisa apresentada na introdução, a avaliação empírica é utilizada para verificar se a MBA/CPN pode promover reusabilidade e produtividade, considerando o ponto de vista dos modeladores.

6.1 Escopo, Modeladores e Variáveis

A metodologia *Goal-Question-Metric (GQM)* (BASILI; ROMBACH, 1988) [1]) guiou a definição da avaliação empírica por meio da análise do uso do modelo de referência em dois aspectos:

- falhas que ocorrem quando as dosagens de insulina estão sendo selecionadas (nenhuma dosagem foi selecionada);
- avaliação para reutilização do ponto de vista dos modeladores, instanciando e estendendo o modelo.

Assim, foram definidas as seguintes questões de pesquisa secundárias: o modelo de referência aumenta a produtividade dos modeladores? (RQ1), e o modelo de referência é reutilizável? (RQ2). As RQs orientaram a especificação das seguintes hipóteses: a produtividade não é aumentada (H0-1), a produtividade é aumentada (HA-1), o modelo de referência não é reutilizável (H0-2), e o modelo de referência é reutilizável (HA-2). H0-1 e HA-1 estão relacionados a RQ1 e H0-2 e HA-2 a RQ2.

Devido as dificuldades trazidas pela pandemia da COVID-19, foram selecionados apenas 12 modeladores usando a técnica de amostragem por conveniência, não tendo sido utilizada nenhuma técnica estatística para dimensionar o tamanho da amostra. Os modeladores foram avaliados em uma universidade federal localizada no Brasil. Na Tabela 6.1 são apresentadas informações sobre os modeladores que participaram do estudo, incluindo questões sobre idade, conhecimento sobre métodos formais, opinião sobre a fase de treinamento, resolução da lista de exercícios, conhecimento sobre *CPN* e conhecimento sobre o trabalho atual.

Tabela 6.1: Questionário identificando os perfis dos modeladores.

Participantes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Idade	19	21	23	21	24	23	23	22	22	30	27	26
Conhecimento sobre métodos formais[1]	N	S	S	S	S	S	S	S	S	S	S	S
Opinião sobre a fase de treinamento[2]	Excelente	Excelente	Excelente	Ótimo	Ótimo	Bom	Excelente	Excelente	Excelente	Excelente	Excelente	Ótimo
Respondeu a lista de exercícios[2]	S	S	S	S	S	S	S	S	S	S	S	S
Conhecimento sobre *CPN*[2]	Bom	Muito Bom	Bom	Bom	Regular	Bom	Bom	Muito Bom	Bom	Bom	Bom	Regular
Conhecimento sobre o trabalho[2]	Muito Bom	Muito Bom	Bom	Bom	Bom	Muito Bom	Muito Bom	Muito Bom	Bom	Bom	Bom	Regular

1: Após a fase de treinamento; 2: Antes da fase de treinamento; S: Sim; N: Não.

Existem duas variáveis dependentes definidas com base nos objetivos com a pesquisa: produtividade dos modeladores e reusabilidade do modelo de referência. Existem quatro variáveis independentes: o modelo de referência, a experiência dos modeladores, o suporte da ferramenta e o ambiente. Essas variáveis são controladas em um nível fixo, o que significa que cada grupo de modeladores (ou seja, controle e tratamento) possui o mesmo modelo de referência, experiência, ferramentas e ambiente.

6.1.1 Procedimento e Medidas

Utilizou-se o *software CPN/Tools* para realizar a avaliação com base em quatro fases: primeira etapa de treinamento, primeira etapa de avaliação, segunda etapa de treinamento e segunda etapa de avaliação. Os modeladores foram preparados na fase de treinamento para especificar modelos *CPN* usando o *CPN/Tools*. Foram utilizados slides e exemplos de *CPN* para permitir apresentações audiovisuais enquanto usava o método específico de resolução de problemas para apoiar atividades de aprendizado prático.

Na primeira etapa de treinamento foram realizadas as seguintes atividades:

- solicitação aos modeladores para resposta do questionário sobre informações pessoais, experiência com métodos formais e experiência com reutilização;
- ministrado um curso de curta duração sobre conceitos e exemplos relacionados com *CPN*, *CPN/Tools* e o modelo de referência. Foi também apresentada uma visão geral do ACCU-CHEK Spirit;
- os modeladores aplicaram as técnicas aprendidas para construir exemplos simples auxiliados durante o curso.

Durante a primeira etapa de avaliação, os modeladores instanciaram o modelo de referência para representar o SBII comercial da ACCUCHEK Spirit. Eles responderam a um questionário sobre os atributos de reutilização dos modelos. Os modeladores foram divididos em grupos de controle (tamanho oito) e tratamento (tamanho quatro). O grupo de tratamento é composto por modeladores que possuem poucas experiências no uso de *CPN* (ou seja, menos de dez horas de curso de *CPN*), enquanto o grupo controle é composto por modeladores que concluíram uma disciplina de seis meses sobre o *CPN* e, consequentemente, pelo menos, seis meses de experiência no uso de *CPN*. Cada modelador foi solicitado a instanciar o modelo de referência em duas horas. Foi preparado um material de avaliação, e, ao aplicar a avaliação, modeladores poderiam sanar dúvidas para resolver equívocos sobre as diretrizes do experimento e a redação do questionário.

A próxima etapa foi a segunda etapa de treinamento, na qual o modelo de referência foi explicado com mais detalhes. Esta etapa foi seguida pela segunda etapa de avaliação, que

compreende a utilização do modelo de referência para estender a especificação. As seguintes atividades foram realizadas na segunda etapa de avaliação:

- cada modelador implementou dois novos requisitos em duas horas (adicionar uma nova representação de bateria e adicionar um recurso para recarregar a bateria);
- os modeladores responderam a um questionário sobre esforço e reutilização.

Foram definidas métricas para avaliar as hipóteses e avaliar os RQs. Assim, foi abordado o tempo para medir a produtividade, computando o tempo necessário para cada grupo terminar os problemas solicitados aos modeladores para resolução durante a fase de avaliação (KITCHENHAM; PICKARD; PFLEEGER, 1995 [19]). Além disso, a reutilização foi medida usando dois fatores: compreensibilidade e adaptabilidade (WASHIZAKI; YAMAMOTO; FUKAZAWA, 2003 [37]). A compreensibilidade está relacionada à facilidade com que o modelador reconhece o significado de um componente do modelo de referência e sua aplicabilidade, enquanto a adaptabilidade significa quão o modelador pode estender o modelo de referência para atender a um novo requisito do sistema.

As hipóteses foram formalizadas para realizar análises estatísticas para o RQ1: a hipótese nula H0-1-1, ou seja, $vU > 1h$, em que vU é o intervalo de tempo (em minutos) necessário para os modeladores concluírem a instanciação do modelo de referência; e a hipótese alternativa HA-1-1, representada por $vU \leq 1h$. Uma hora corresponde a seis vezes o tempo gasto pelos pesquisadores para instanciar, pela primeira vez, o modelo de referência baseado no ACCU-CHEK Spirit. Também foram formalizadas as hipóteses para realizar análises estatísticas relacionadas ao fator esforço: a hipótese nula H0-1-2, ou seja, $vU > 3$, em que vU representa a classificação média das respostas para as questões de esforço; e a hipótese alternativa HA-1-2, representada por $vU \leq 3$.

Também foram formalizadas as hipóteses para realizar análises estatísticas relacionadas ao RQ2. Considerando a compreensibilidade, existe uma hipótese nula H0-2-1, ou seja, $vU \leq 3$, em que vU representa a classificação média das respostas para as questões de compreensibilidade, enquanto a hipótese alternativa HA-2-1 é representado por $vU > 3$. Considerando a adaptabilidade, existe uma hipótese nula H0-2-2, ou seja, $vU \leq 3$, em que vU representa a classificação média das respostas para as questões de adaptabilidade, enquanto a hipótese alternativa HA-2-2 é representado por $vU > 3$.

Portanto, foi aplicado um questionário ao final dos experimentos para coletar métricas para os fatores de esforço e reutilização, fornecendo uma escala Likert de 5 pontos (1) a (5). A interpretação da escala referente às métricas é baseada no esforço de instanciar o modelo de referência durante o experimento 01: (1) representa o melhor resultado e (5) representa o pior resultado. Para as medidas de compreensibilidade e adaptabilidade realizadas durante o experimento 02, a interpretação da escala é (1) para o pior e (5) para o melhor. Os modeladores responderam ao questionário após estender o modelo de referência.

6.2 Análise

Na Figura 6.1 são apresentadas as respostas para esforço e cada fator de reutilização avaliado nos experimentos 01 e 02, respectivamente. Não foram aplicados testes de hipóteses mais complexos como o *t-test* e *Wilcoxon test* porque as hipóteses que foram definidas para avaliar RQ1 e RQ2 apenas utilizam as respostas médias referentes ao fatores considerados neste experimento. A estatística descritiva básica permitiu a avaliação das hipóteses. Para RQ1, foi avaliada a reutilização do modelo de referência com base na análise da melhoria da produtividade dos modeladores. Os 12 modeladores que participaram do experimento concluíram a instanciação do modelo, em média, em seis minutos.

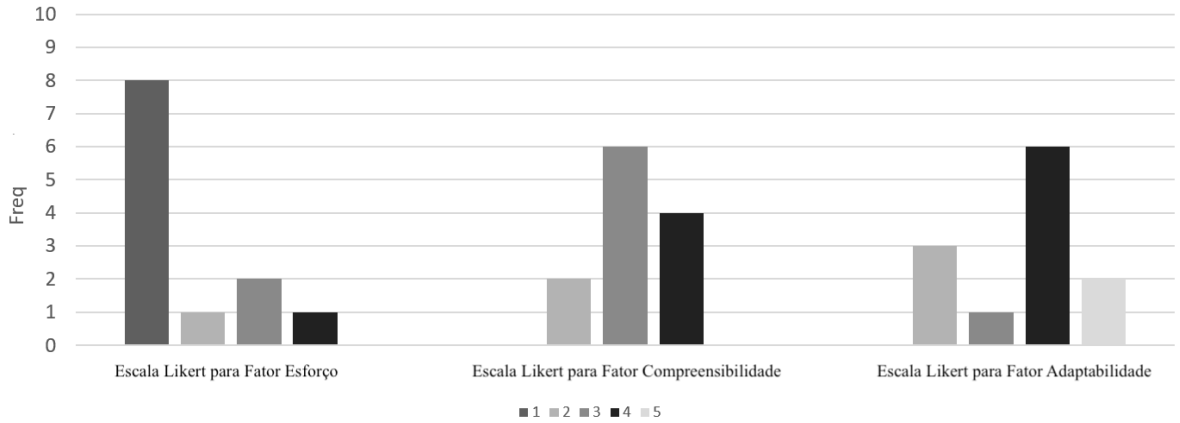


Figura 6.1: Distribuição de respostas de acordo com os fatores de esforço e reutilização.

Quando solicitados a responder ao questionário, 66,7% dos modeladores (8 dos 12 modeladores) afirmaram que a tarefa de instanciar o modelo de referência não apresentou esforço. Em contrapartida, 8,3% afirmaram esforço baixo, 16,7% esforço médio e 8,3% esforço con-

siderável. Assim, as hipóteses H0-1-1 e H0-1-2 foram refutadas, mostrando que o modelo apresentou uma resposta positiva a RQ1 (os modeladores utilizaram apenas 10% do tempo estimado).

Foi avaliada a reutilização do modelo de referência para analisar a RQ2 com base nos fatores de compreensibilidade e adaptabilidade. Na Figura 6.2 é apresentado o número de modeladores afirmando estender o modelo considerando os requisitos 01 e/ou 02, incluindo também o número de modeladores que experimentaram o 02 corretamente. Todos os modeladores implementaram o requisito 01 (ou seja, adicionar uma nova bateria).

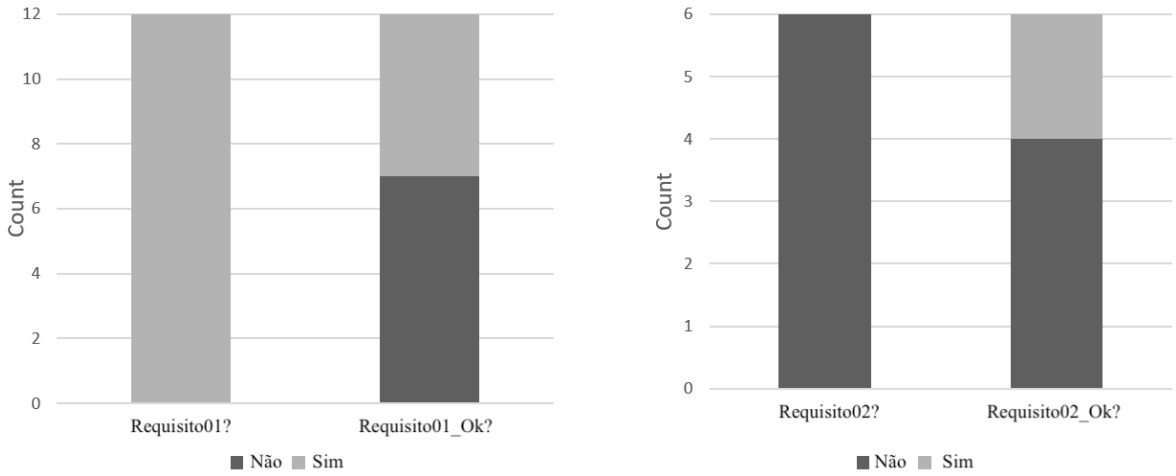


Figura 6.2: Modelo instanciado vs. instanciado corretamente.

No entanto, apenas cinco deles o implementaram com sucesso. A análise do trabalho realizado pelos sete modeladores que não implementaram o requisito 01 com sucesso mostrou que todos falharam na mesma tarefa: a marcação do lugar *Ok_B* (submódulo *Verify Battery*) não foi alterado de 2'1 para 3'1. Seis modeladores tentaram implementar o requisito 02. No entanto, apenas dois deles obtiveram a resposta correta. Os resultados dos quatro modeladores que não alcançaram a implementação correta mostraram erros diferentes, todos relacionados ao uso do *CPN*/ML.

Para analisar o tempo gasto para realizar o segundo experimento, considerou-se o grupo relacionado a cada modelador. Na Figura 6.3 é apresentado o tempo coletado para os grupos de controle e tratamento. Apenas um dos modeladores implementou com sucesso o primeiro e o segundo requisitos do modelo de referência em 25 e 38 minutos, respectivamente. Os modeladores dimensionaram o primeiro requisito como uma tarefa de baixa complexidade,

discordando quando questionados sobre a complexidade do segundo requisito: o modelador do grupo de controle dimensionou como uma tarefa de média complexidade. Em contraste, o modelador do grupo de tratamento a escalou como uma tarefa de baixa complexidade.

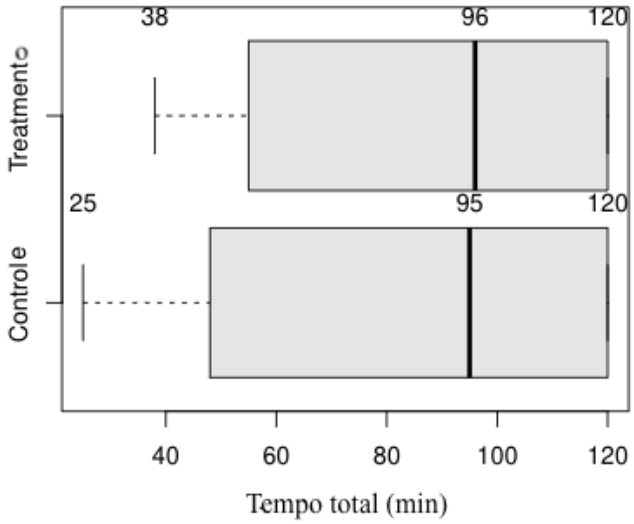


Figura 6.3: Tempo coletado para o grupo controle e tratamento.

A análise da compreensibilidade e adaptabilidade dos resultados apresentados na Figura 6.2 e Figura 6.3 não permitiu concluir que o modelo de referência é totalmente reutilizável, sendo necessária a análise dos resultados apresentados na Figura 6.1. A maioria dos modeladores não entendeu completamente cada componente e funcionamento do modelo de referência. O alto número de módulos exigidos pela complexidade dos sistemas de bombas de infusão de insulina impactaram negativamente na adaptação do modelo para agregar novos requisitos.

A Figura 6.4 é usada para reforçar esta afirmação, apresentando as respostas dos modeladores para a complexidade de cada requisito adicionado ao modelo de referência no segundo experimento. A maioria dos modeladores (9 de 12 participantes, correspondendo a 75%) afirmou que o requisito 01 é de baixa (B) complexidade, enquanto o requisito 02 foi declarado como de média (M) ou alta complexidade (A) por 75% dos participantes do experimento.

A avaliação da RQ2 com base no questionamento apresentado aos modeladores mostrou que o H0-2-1 foi refutado devido à classificação média do fator de compreensibilidade (aproximadamente 3,17). Para H0-2-2, a classificação média do fator de adaptabilidade apresentou aproximadamente 3,58, refutando a hipótese nula. Assim, o modelo de referência apresentou

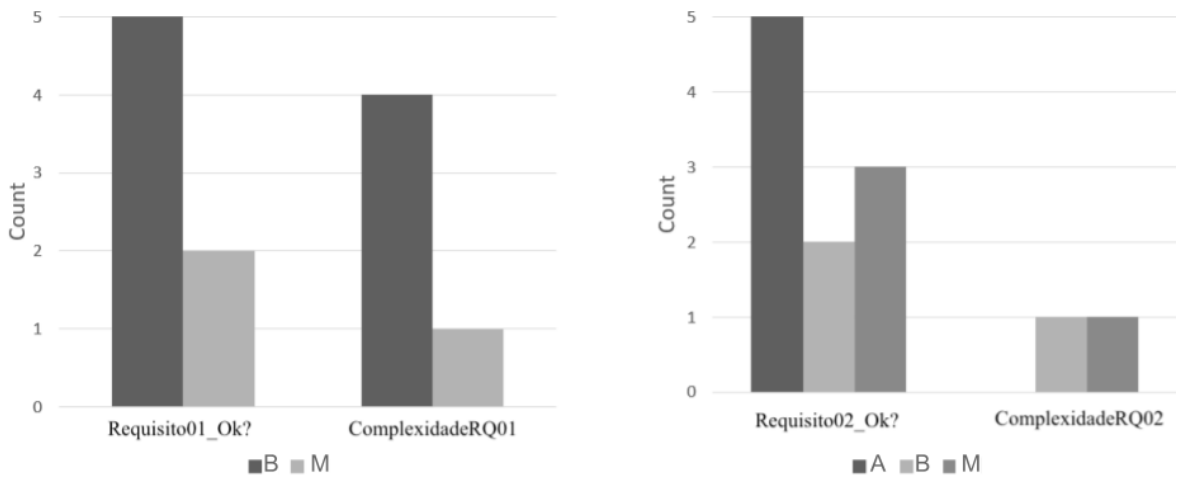


Figura 6.4: Distribuição de respostas considerando a complexidade das funcionalidades.

níveis aceitáveis de compreensibilidade e adaptabilidade.

Capítulo 7

Aplicação Web

Com base nos resultados da avaliação empírica, o *framework Access/CPN* e serviços web foram usados para implementar uma aplicação web, com o objetivo de auxiliar os modeladores a reutilizar a abordagem por meio de treinamento baseado em simulação. O uso do *Access/CPN* permitiu incorporar os modelos de referência em serviços web para realizar simulações sem a interface gráfica do usuário disponível no *CPN/Tools*, melhorando a MBA/CPN. Assim, foram disponibilizados serviços web como consumidores de componentes *Access/CPN* (Figura 7.1).

Como uma API RESTful, a disponibilidade de serviços permite que os modeladores reutilizem facilmente as funcionalidades de *Access/CPN* sem preocupações relacionadas às plataformas de desenvolvimento. Além da aplicação web apresentada neste capítulo, outros desenvolvedores podem reutilizar essa API web para incorporar modelos *CPN* em diferentes cenários. A aplicação fornece *feedback* sobre o significado dos componentes do modelo durante as simulações em segundo plano (ou seja, sem o *CPN/Tools*) para melhorar a produtividade e a compreensão, guiado por uma interface gráfica de usuário fácil de usar.

Além das interfaces públicas disponíveis no *Access/CPN*, a API RESTful disponibilizada fornece novas interfaces públicas para simplificar ainda mais a simulação em segundo plano dos modelos *CPN*. Por exemplo, são disponibilizados serviços web para ajudar os modeladores a interromper a simulação do modelo quando certas condições de parada são satisfeitas, como alcançar transições específicas desejadas. Tais serviços web são relevantes para auxiliar os modeladores na análise de requisitos específicos do sistema durante as simulações.

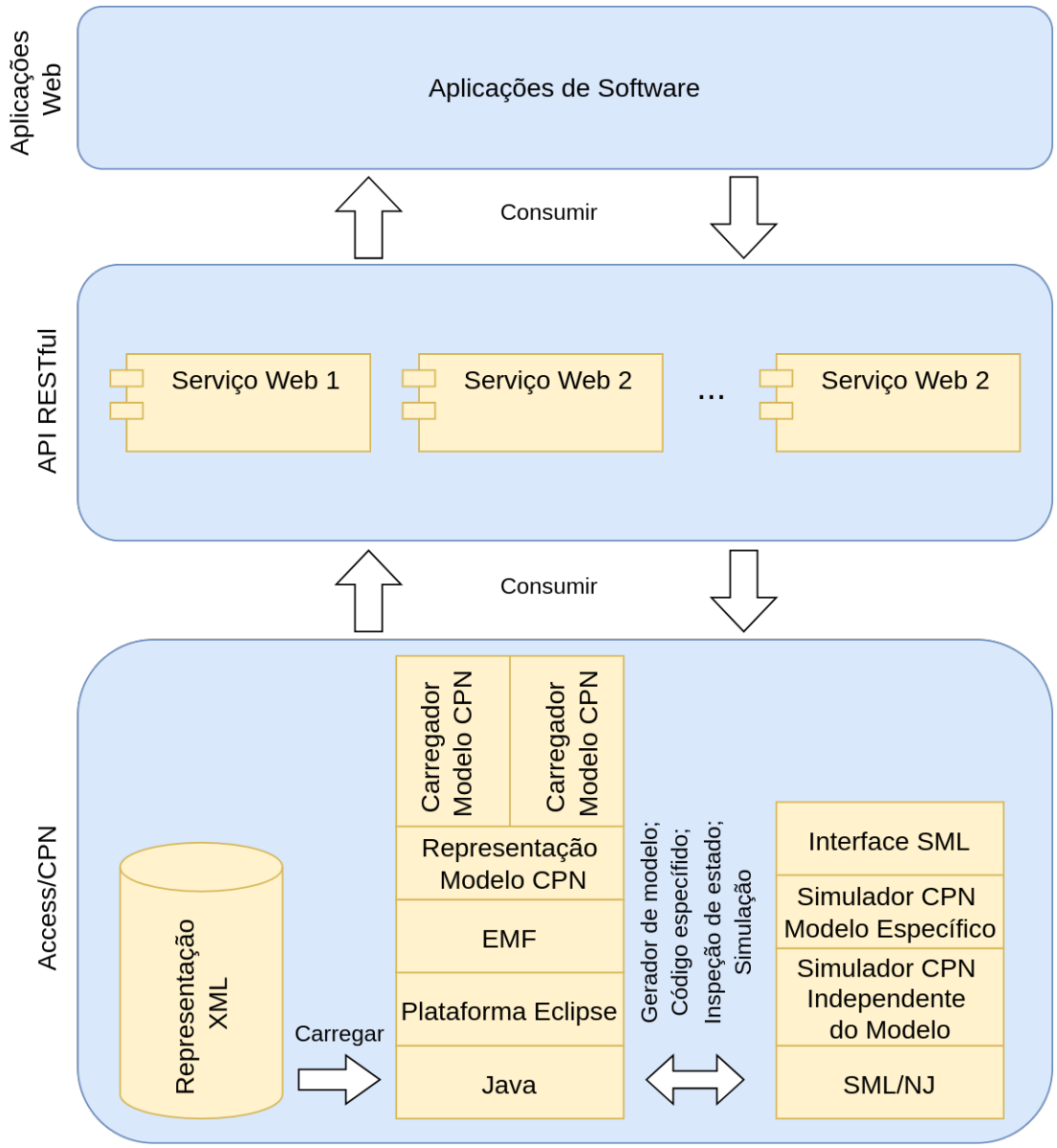


Figura 7.1: Aplicação Web e API implementadas como consumidores de componentes *Access/CPN*.

Todas as funcionalidades disponibilizadas no aplicação web são apresentadas utilizando o diagrama de casos de uso apresentado na Figura 7.2. O sistema foi implementado seguindo o padrão de projeto arquitetural conhecido como *Model-View-Controller* (MVC) e a arquitetura denominada Transferência de Estado Representacional (*Representational State Transfer - REST)*).

O modelador pode usar a aplicação web para manipular o modelo *CPN* de Sistemas de Bomba de Infusão de Insulina (SBII) usando quatro funcionalidades principais:

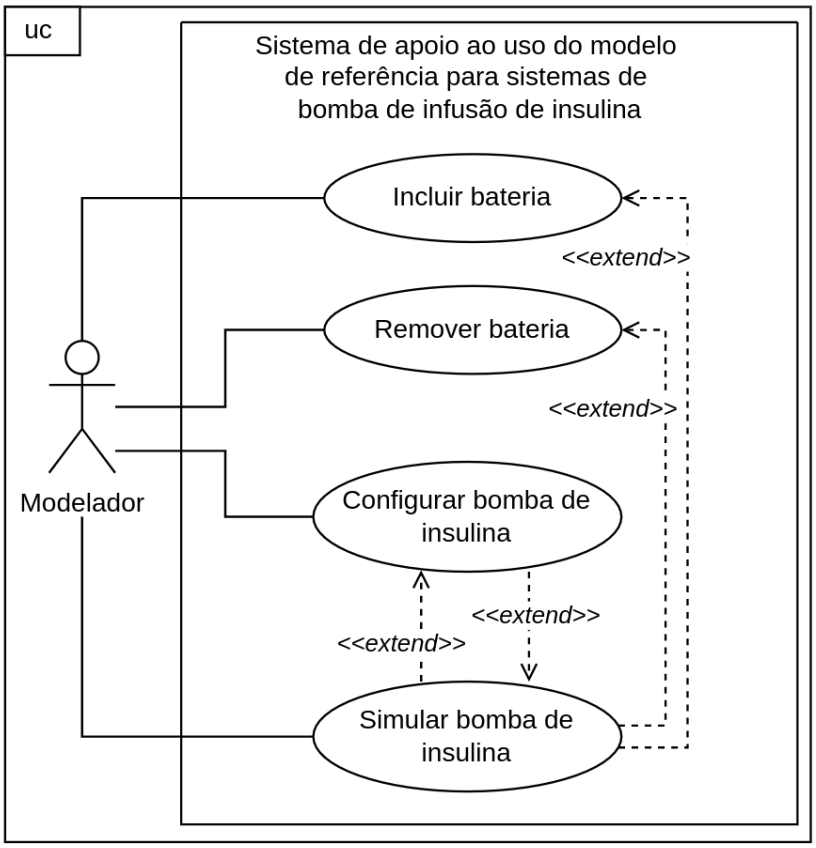


Figura 7.2: Diagrama de caso de uso.

- adicionar uma bateria ao sistema;
- remover uma bateria do sistema;
- configurar a bomba de infusão de insulina;
- simular a bomba de infusão de insulina.

As duas primeiras funcionalidades permitem que o usuário altere a estrutura do modelo *CPN* adicionando/removendo uma instância de um módulo que representa a bateria da bomba de infusão de insulina. Tais mudanças são permitidas devido aos recursos de modularização de *CPN* hierárquico. A terceira funcionalidade permite que os modeladores alterem os valores dos parâmetros do modelo de referência. Por fim, a quarta funcionalidade permite que os modeladores simulem o modelo *CPN* usando a interface gráfica do usuário da aplicação web, sem usar *CPN/Tools*, para melhorar o entendimento do comportamento da bomba sem que o usuário tenha acesso à estrutura interna do modelo.

Nas Figuras 7.3 e 7.4 são apresentados os diagramas de pacotes e classes da API da aplicação, que foram construídos usando UML.

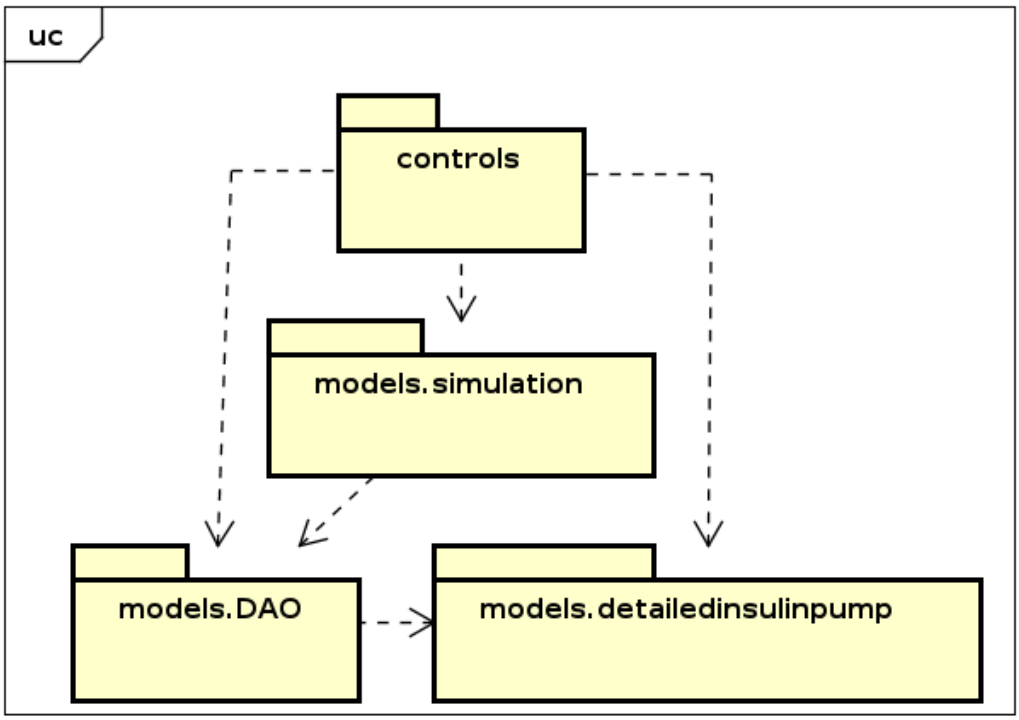


Figura 7.3: Diagrama de pacotes.

O diagrama de pacotes apresentado na Figura 7.3 é usado para apresentar apenas os elementos MVC que fazem parte do *backend*, que são os *controllers* e os *models*. O pacote de controle depende de todos os pacotes de modelo, que foram organizados em três pacotes diferentes, considerando sua finalidade. Assim, o pacote `models.simulation` agrupa as classes de modelo relacionadas ao requisito de simulação da bomba de insulina, o pacote `models.DAO` agrupa as classes que realizam o acesso aos dados e os grupos de pacotes `models.detailedinsulinpump` a classe que representa o modelo *CPN* da bomba de insulina. Além disso, o pacote `models.simulation` depende do pacote `models.DAO`, que depende do pacote `models.detailedinsulinpump`.

O diagrama de classes apresentado na Figura 7.4 é usado para apresentar todas as classes que compõem a API contida nos pacotes de controle e modelo. As classes `BatteryControl`, `InsulinPumpDetailedConfigurationControl` e `InsulinPumpSimulatorControl` pertencem ao pacote de contro-

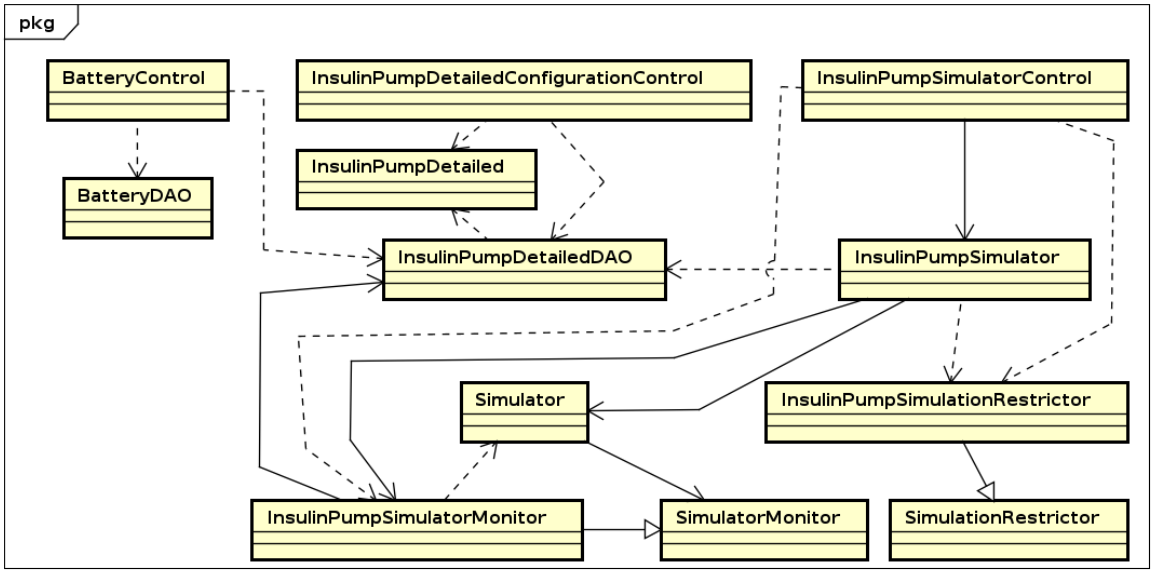


Figura 7.4: Diagrama de classes.

les. Em contraste, as classes `InsulinPumpDetailed`, `BatteryDAO`, `InsulinPumpDetailedDAO`, `InsulinPumpSimulator`, `Simulator`, `InsulinPumpSimulationRestrictor`, `SimulationRestrictor`, `InsulinPumpSimulatorMonitor` e `SimulatorMonitor` são classes de modelo.

As classes de controle integram todos os serviços oferecidos pela API, que são utilizados no *frontend* da aplicação para fornecer ao usuário as funcionalidades oferecidas pelo sistema. Os serviços fornecidos pela classe `BatteryControl` estão adicionando/removendo uma instância do módulo de bateria ao modelo de bomba de insulina e listando o número de instâncias de bateria no modelo. Por sua vez, a classe `InsulinPumpDetailedConfigurationControl` fornece os serviços de listagem e configuração dos parâmetros da bomba de insulina. Por fim, os serviços oferecidos pela classe `InsulinPumpSimulatorControl` envolvem operações relacionadas à simulação do modelo *CPN* da bomba de insulina, como preparar o ambiente de simulação, executar uma etapa de simulação, gerar a simulação e outras que podem ser consultadas no repositório da API.

As classes `BatteryDAO` e `InsulinPumpDetailedDAO` são responsáveis por realizar operações de acesso aos dados. Essas operações consistem na manipulação direta

do arquivo em formato que representa o modelo da bomba de infusão de insulina. Portanto, quando, por exemplo, o usuário utiliza a funcionalidade de adicionar uma instância de bateria ao modelo, a classe `BatteryControl` delega essa operação a ser realizada pela classe `BatteryDAO`, que é responsável por alterar diretamente a estrutura do arquivo que representa a bomba de insulina, adicionando os elementos necessários para representar a nova instância da bateria. Assim, existe uma relação de dependência entre as classes `BatteryControl` e `BatteryDAO`.

A classe `InsulinPumpDetailed` é a única pertencente ao pacote `models.detailedinsulinpump`. É uma classe simples, sem lógica de negócio ou aplicação, contendo apenas atributos e métodos *getters* e *setters*, além de um construtor responsável por inicializar todos os atributos. Ele agrupa todos os parâmetros da bomba de infusão de insulina, que podem ser alterados através do serviço de configuração do modelo fornecido pela classe `InsulinPumpDetailedConfigurationControl`. Esta classe depende da classe `InsulinPumpDetailed` para realizar seus serviços.

Por fim, as classes `InsulinPumpSimulator`, `Simulator`, `InsulinPumpSimulationRestrictor`, `SimulationRestrictor`, `InsulinPumpSimulatorMonitor` e `SimulatorMonitor` são modelos relacionados à funcionalidade de simulação do modelo *CPN* da bomba de insulina. Todo serviço de simulação fornecido pela API depende da classe `InsulinPumpSimulator`, que é responsável por criar o simulador do modelo *CPN* da bomba de insulina e realizar as simulações. Além disso, alguns serviços de simulação dependem das classes `InsulinPumpSimulationRestrictor` e `InsulinPumpSimulatorMonitor`, que permitem, respectivamente, definir uma restrição de simulação para a bomba de insulina e armazenar o estado de simulação atual do modelo. Como pode ser visto no diagrama de classes, essas duas classes estendem as classes `SimulationRestrictor` e `SimulatorMonitor`, respectivamente.

A classe `SimulationRestrictor` foi definida para representar restrições que podem ser usadas para realizar uma etapa de simulação em qualquer modelo *CPN*. Dessa forma, é possível especificar transições que podem e não podem ser acionadas em uma determinada etapa de simulação realizada pelo simulador. Da mesma forma, a classe `SimulatorMonitor` foi definida para permitir o armazenamento de dados de simulação

úteis durante a simulação de qualquer modelo *CPN*. Esses dados são as transições habilitadas e a última transição acionada. Em particular, esta classe `SimulatorMonitor` é usada pela classe `Simulator`.

A classe `Simulator` reúne os métodos básicos para simular qualquer modelo *CPN*, por isso é utilizada pela classe `InsulinPumpSimulator`. Os métodos básicos da classe `Simulator` incluem, por exemplo, executar uma transição, executar um elemento de ligação e verificar a marcação atual do modelo.

Na Figura 7.5 e Figura 7.6 são apresentados exemplos de interfaces gráficas de usuário da aplicação web. Por exemplo, o aplicativo permite que os modeladores simulem a configuração da bomba, mostrando cada etapa da simulação (por exemplo, junto com informações relacionadas a locais e transições). Os serviços web usados para implementar esse *software* podem ser facilmente adaptados para lidar com uma versão diferente do modelo de referência. O aplicativo foi implementado usando a linguagem de programação Java e a biblioteca *JavaScript React* [1]. O aplicativo também foi cuidadosamente testado usando o ambiente de desenvolvimento de API *Postman* [2]. Assim, foram realizadas simulações de modelo *CPN* em segundo plano usando o aplicação web para cada requisito do SBII.

[1] https://www.javascript.com/

[2] https://www.postman.com/

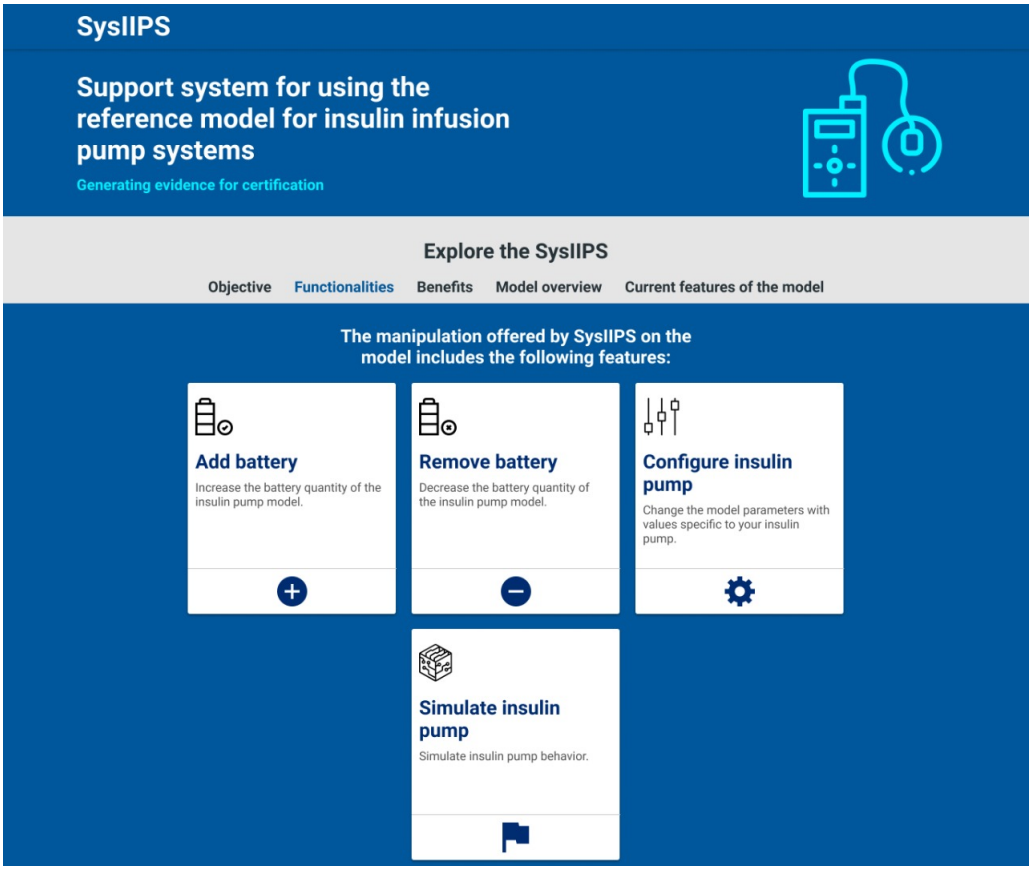


Figura 7.5: Exemplo de interface gráfica do usuário da aplicação web, apresentando as principais funcionalidades.

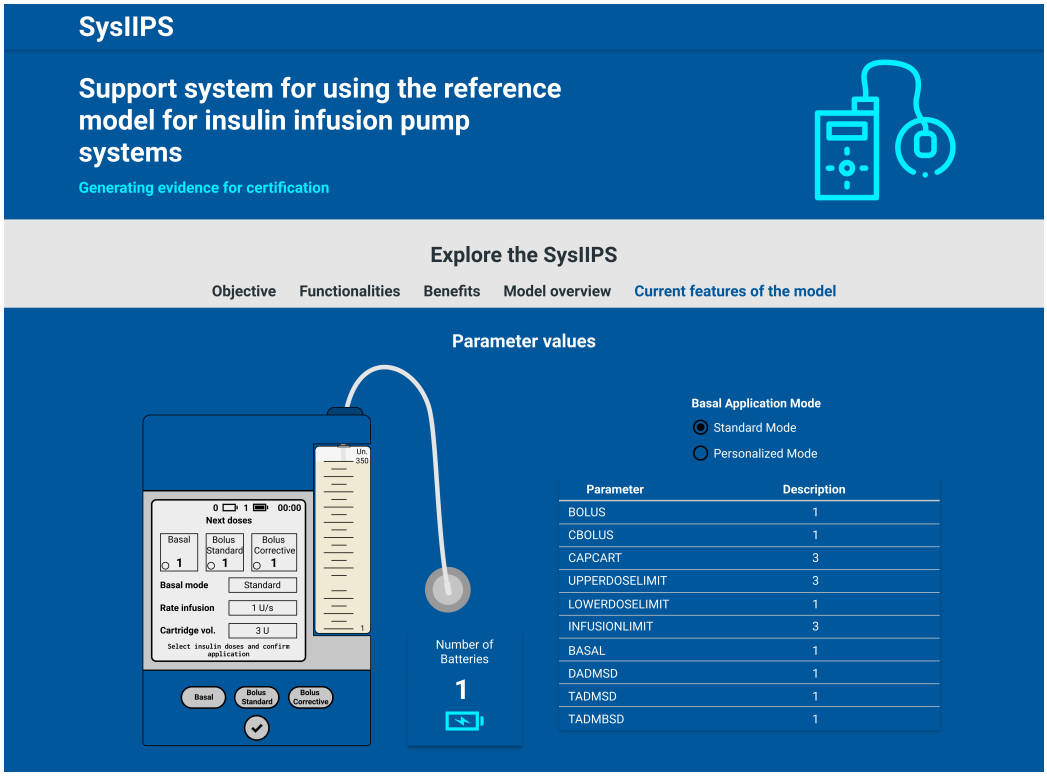


Figura 7.6: Amostra da interface gráfica do usuário da aplicação web para simulação do sistema de bomba de infusão de insulina.

Capítulo 8

Discussão

A MBA/CPN possui características que não foram totalmente consideradas por estudos anteriores, como documentação de casos de garantia para engenharia de requisitos baseada em metas (documentos especificado com o padrão para troca de casos de garantia (*Assurance Case Exchange Standard - ACES*)), configuração (modelo instanciado para diferentes sistemas), temporização (restrições de sistemas de infusão), hierarquia (gerenciamento da complexidade da especificação) e execução (simulação de comportamentos). A documentação de requisitos usando casos de garantia baseados em *ACES* ajuda os fabricantes a fornecer argumentos e evidências para certificação, reutilizando o mesmo documento para gerenciar requisitos. Isso também é relevante para diminuir o tempo de desenvolvimento, considerando que as agências reguladoras governamentais (por exemplo, a *Food and Drug Administration - FDA*) recomendam a apresentação de casos de garantia de Sistemas de Bomba de Infusão de Insulina (SBII) durante o processo de certificação. Fabricantes e agências reguladoras podem trocar o documento *ACES* para melhorar a qualidade dos SBII.

Além disso, para aplicar a MBA/CPN em um cenário real usando um sistema específico, os fabricantes precisam apenas configurar os parâmetros de entrada do modelo de referência para representar as funcionalidades básicas requeridas dos SBII. Se for necessária uma nova funcionalidade para o sistema, a estrutura de módulos do modelo de referência fornece um mecanismo fácil para adaptações. A reutilização do modelo de referência é a base para desenvolver um novo sistema ou realizar avaliações de qualidade dos já existentes, comparando os SBII em avaliação e o modelo. Embora tenham sido manualmente especificados, devido ao foco em fornecer uma base verificada e validada para outras extensões, também

é possível gerar modelos automaticamente com um nível mais alto de abstração (primeiro refinamento do sistema - modelo de referência abstrato da Figura 4.1) inferindo-os da especificação *ACES*. A geração automática de modelos *CPN* pode ser relevante se os fabricantes desejarem incluir novas metas para os SBII em desenvolvimento ou desejarem reutilizar nossa abordagem para outros sistemas críticos.

O modelo formal é um artefato de projeto usado para realizar atividades de verificação e validação para aumentar a confiança nos sistemas de bombas de infusão de insulina. A verificação é relevante para a realização de avaliações de atributos de qualidade durante um processo de certificação. A avaliação da qualidade pode ser realizada comparando cada módulo do modelo de referência verificado com cada módulo dos SBII em avaliação. A validação é relevante para a realização de avaliações de qualidade de segurança e eficácia dos SBII. Por exemplo, quando um *recall* é relatado, o modelo de referência pode ajudar os fabricantes a avaliar o sistema comparando manualmente as saídas com os resultados da simulação do modelo. Isso pode diminuir o custo e o tempo de desenvolvimento corrigindo prontamente os defeitos/falhas.

Considerando a diretriz da *FDA* relacionada ao ciclo de vida das bombas de infusão[1], o modelo de referência atende aos *recalls* mais comuns relatados de bombas de infusão, incluindo erro de *software* e sobreinfusão e infusão. Além disso, com base na mesma diretriz da *FDA*, o modelo atende aos requisitos básicos dos SBII: mecanismo de administração de infusão, reservatório de medicamento, mecanismo de bolus, relógio em tempo real, registro de bomba e seleção de terapia.

Portanto, os fabricantes podem reutilizar a MBA/CPN para desenvolver SBII que atendam aos principais requisitos da *FDA*, corrigir problemas conhecidos e identificar e corrigir problemas desconhecidos relatados por agências reguladoras como *recalls*. No entanto, é essencial destacar outras características relevantes das bombas de infusão que não estão no escopo da MBA/CPN, como falhas de *hardware*, degradação de *hardware*, interfaces de comunicação, fonte de alimentação e requisitos elétricos.

Durante a avaliação empírica, a tarefa de representar um sistema específico, o ACCU-CHEK Spirit, durou apenas alguns minutos, indicando que estender a modelagem não é

[1] https://www.fda.gov/regulatory-information/search-fda-guidance-documents/infusion-pumps-total-product-life-cycle

demorado. Essa descoberta ajuda os fabricantes de sistemas de bombas de infusão de insulina a lidar com o aumento do tempo de desenvolvimento e dos custos do projeto ao aplicar métodos formais. As respostas do questionário para os dois experimentos refletem a opinião favorável dos modeladores entrevistados quanto à reutilização do modelo de referência. Todos os modeladores relataram um desempenho excelente ou ótimo da fase de treinamento, demonstrando conhecimento adequado sobre o estudo (aproximadamente 91,7%) e o *CPN* (aproximadamente 83,3%). Os resultados indicaram uma resposta positiva da pergunta principal de pesquisa. No entanto, o desempenho dos modeladores foi baixo ao realizar modificações no modelo de referência, apresentando um problema de usabilidade. Uma hipótese de solução para este problema é o uso da interface gráfica do usuário para auxiliar os modeladores na compreensão e modificação dos componentes do modelo como um mecanismo de aprendizado. Tal hipótese motivou a implementação da aplicação web para auxiliar o treinamento de modeladores baseado em simulação.

A implementação da aplicação utilizando o *framework Access/CPN* para realizar a execução em segundo plano dos modelos *CPN* possibilitou a melhoria da MBA/CPN, pois não requer conhecimento sobre todos os componentes do modelo para realizar as simulações. O aplicativo fornece *feedback* sobre o significado de tais componentes do modelo para melhorar a compreensão dos modeladores. Além disso, durante a implementação, também foi possível fornecer uma API RESTful (Figura 7.1) que pode ser reutilizada por outros desenvolvedores de sistemas críticos seguros (por exemplo, para realizar outros treinamentos baseados em simulação). O treinamento baseado em simulação usando *software* também pode ser relevante para melhorar o projeto de interfaces gráficas de usuários dos sistemas. Conforme destacado na introdução, um dos problemas enfrentados pelos desenvolvedores de SBII é o fornecimento de interfaces gráficas de usuário inadequadas[2].

No entanto, existem algumas limitações que podem ser destacadas. Dois refinamentos do modelo de referência permitiram a verificação e validação do modelo por meio de simulações. Com a geração do grafo de componentes fortemente conectados, esta abordagem evitou o problema da explosão do espaço de estados ao invés de usar uma técnica mais robusta para reduzir o espaço de estados da versão mais detalhada, por exemplo, método de equivalência. Isso pode ter dificultado o manuseio do modelo, considerando que os fabri-

[2]https://www.fda.gov/medical-devices/infusion-pumps/infusion-pump-improvement-initiative

cantes precisam dividir as atividades de verificação e validação em diferentes versões de especificação. A avaliação empírica mostrou que os modeladores relataram o modelo como um artefato de projeto de fácil compreensão. Além disso, foram utilizadas duas propriedades dos SBII para realizar a verificação do modelo. Isso pode ter limitado a etapa de verificação porque agências reguladoras podem considerar outras propriedades como relevantes no processo de certificação. As simulações do modelo por dosagens hipotéticas de insulina também podem limitar a validação em relação ao uso de dados reais de uma prescrição médica. No entanto, as dosagens foram utilizadas na verificação e validação para ilustrar cenários de avaliação de qualidade ao invés de validar totalmente os comportamentos de um sistema instanciado.

Capítulo 9

Conclusões e Trabalhos Futuros

Declarar existências de outras tecnicas para reduzao de espaço de estados que serão exploradas em trabalhos futuros.

Nesta dissertação foi apresentada uma abordagem baseada em modelos com foco em modelos de referência *CPN* de Sistemas de Bombas de Infusão de Insulina (SBII). O objetivo foi auxiliar os modeladores de SBII na avaliação da qualidade. Também foi descrito um estudo de caso sobre um SBII comercial (ou seja, ACCU-CHEK Spirit). Os modelos de referência são relevantes porque as bombas de infusão de insulina são sistemas críticos seguros usados para monitorar e tratar pacientes com diabetes.

O modelo *CPN* executável, paramétrico, modular e temporizado incluiu os recursos essenciais dos SBII para garantir o funcionamento correto. O estudo de caso foi relevante para estender o modelo para verificar e validar dois refinamentos do modelo como cenários de avaliação de qualidade. Utilizando a técnica de verificação automática de modelos, foram consideradas as verificações formais de duas propriedades de segurança para o primeiro refinamento. O segundo refinamento foi validado usando simulações para analisar o modelo quanto às especificações técnicas do sistema comercial.

Por fim, a avaliação empírica e a implementação da aplicação web para treinamento baseado em simulação demonstrou que a MBA/CPN é reutilizável para o processo de desenvolvimento e certificação de sistemas de bombas de infusão de insulina. Assim, o uso de MBA/CPN pode permitir que os fabricantes reduzam custos e tempo de desenvolvimento ao usar modelos formais de SBII. Os fabricantes de outros dispositivos médicos também podem reutilizá-los definindo e avaliando um modelo de referência específico com base nas etapas

de modelagem propostas.

Como existem carências de estudos que forneçam modelos de sistemas de bombas de infusão de insulina genéricos, paramétricos e temporizados, os métodos aplicados neste estudo possibilitaram o aprimoramento das especificações disponíveis desses sistemas. Assim, tal lacuna de pesquisa também resultou na falta de estudos que forneçam avaliações de produtividade e reutilização de modelos *CPN* de SBII. Neste estudo, tais limitações foram abordadas, contribuindo com o estado da arte. Durante a avaliação empírica, o tempo mostrou-se uma métrica relevante para medir a produtividade (computando o tempo necessário para cada grupo terminar os problemas solicitados durante a fase de avaliação). Dois fatores (compreensibilidade e adaptabilidade) também se mostraram relevantes para medir a reutilização. Em relação às técnicas formais (por exemplo, *CPN* e verificação automática de modelos), embora seja um consenso que a proposta e o uso de métodos formais para validar o comportamento de sistemas críticos de segurança sejam relevantes, a proposta também funciona como um guia, juntamente com artefatos de projetos reutilizáveis, para desenvolvedores de SBII.

É relevante também destacar que os resultados descritos nesta dissertação foram publicados no periódico *Applied Sciences* e na *ITNG 2022 19th International Conference on Information Technology-New Generations*. Os dois artigos podem ser acessados em [8] e [7].

Como trabalho futuro, pretende-se realizar outro estudo empírico para avaliar a aplicação web, integrado ao modelo de referência, para analisar se a compreensibilidade e a adaptabilidade são melhoradas. Pretende-se ainda incluir como nova funcionalidade da aplicação permitir a criação de propriedades de segurança a partir de componentes gráficos intuitivos. Ainda sobre a aplicação web é relevante também a realização de testes de usabilidade.

Além disso, pretende-se investigar a geração automática de modelos *CPN* com um nível mais alto de abstração com base na especificação dos casos de garantias realizada com o padrão para troca de casos de garantia (*Assurance Case Exchange Standard - ACES*). Por fim, investigar a comparação automática das saídas de simulação dos módulos dos SBII em avaliação com os módulos do modelo de referência proposto é outro trabalho futuro planejado. Esses modelos gerados automaticamente e essa comparação automática podem diminuir os esforços dos modeladores durante as etapas de especificação da abordagem proposta.

Bibliografia

[1] BASILI, V.R.; ROMBACH, H.D.. The TAME project: towards improvement-oriented software environments. **IEEE Transactions On Software Engineering**, [S.L.], v. 14, n. 6, p. 758-773, jun. 1988. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/32.6156.

[2] BHATT, Devesh; HALL, Brendan; MURUGESAN, Anitha; OGLESBY, David; BUSH, Eric; ENGSTROM, Eric; MUELLER, Joseph; PELICAN, Michael. Opportunities and challenges for formal methods tools in the certification of avionics software. **2017 IEEE Aerospace Conference**, [S.L.], p. 1-20, mar. 2017. IEEE. http://dx.doi.org/10.1109/aero.2017.7943664.

[3] CALINESCU, Radu; WEYNS, Danny; GERASIMOU, Simos; IFTIKHAR, Muhammad Usman; HABLI, Ibrahim; KELLY, Tim. Engineering Trustworthy Self-Adaptive Software with Dynamic Assurance Cases. **IEEE Transactions On Software Engineering**, [S.L.], v. 44, n. 11, p. 1039-1069, 1 nov. 2018. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tse.2017.2738640.

[4] CHEN, Yihai; LAWFORD, Mark; WANG, Hao; WASSYNG, Alan. Insulin Pump Software Certification. **Foundations Of Health Information Engineering And Systems**, [S.L.], p. 87-106, 2014. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-53956-5_7.

[5] COSTA, Tássio Fernandes. **Um modelo de referência para sistemas de bomba de infusão de insulina**. 2019. 68 f. TCC (Graduação) - Curso de Bacharelado em Interdisciplinar em Tecnologia da Informação, Departamento de Engenharias e Tecnologia, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, 2019.

[6] COSTA, Tassio Fernandes; SOBRINHO, Álvaro; SILVA, Lenardo Chaves e; SILVA, Leandro Dias da; PERKUSICH, Angelo. A Coloured Petri Nets Reference Model of Insulin Infusion Pump Control Systems: assisting the certification process. **IECON 2019 - 45Th Annual Conference Of The Ieee Industrial Electronics Society**, [S.L.], p. 2713-2718, out. 2019. IEEE. http://dx.doi.org/10.1109/iecon.2019.8927111.

[7] COSTA, Tássio Fernandes; SOBRINHO, Álvaro; SILVA, Lenardo Chaves e; SILVA, Leandro Dias da; PERKUSICH, Angelo. A Model-Based Approach for Quality Assessment of Insulin Infusion Pump Systems. **Advances In Intelligent Systems And Computing**, [S.L.], p. 57-64, 2022. Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-97652-1_8.

[8] COSTA, Tássio Fernandes; SOBRINHO, Álvaro; SILVA, Lenardo Chaves e; SILVA, Leandro Dias da; PERKUSICH, Angelo. Coloured Petri Nets-Based Modeling and Validation of Insulin Infusion Pump Systems. **Applied Sciences**, [S.L.], v. 12, n. 3, p. 1475, 29 jan. 2022. MDPI AG. http://dx.doi.org/10.3390/app12031475.

[9] DING, Jie; CHEN, Xiao; WANG, Rui. A Compositional Analysis Method for Petri-Net Models. **IEEE Access**, [s.l.], v. 5, p.27599-27610, 2017. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/access.2017.2772829.

[10] ENTEZARI-MALEKI, Reza; ETESAMI, Sayed Ehsan; GHORBANI, Negar; NIAKI, Arian Akhavan; SOUSA, Leonel; MOVAGHAR, Ali. Modeling and Evaluation of Service Composition in Commercial Multiclouds Using Timed Colored Petri Nets. **IEEE Transactions On Systems, Man, And Cybernetics**: Systems, [S.L.], v. 50, n. 3, p. 947-961, mar. 2020. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tsmc.2017.2768586.

[11] FRECKMANN, Guido; KAMECKE, Ulrike; WALDENMAIER, Delia; HAUG, Cornelia; ZIEGLER, Ralph. Accuracy of Bolus and Basal Rate Delivery of Different Insulin Pump Systems. **Diabetes Technology & Therapeutics**, [S.L.], v. 21, n. 4, p. 201-208, abr. 2019. Mary Ann Liebert Inc. http://dx.doi.org/10.1089/dia.2018.0376.

[12] GAO, Xuemei; WEN, Qiang; DUAN, Xiaolian; JIN, Wei; TANG, Xiaohong; ZHONG, Ling; XIA, Shitao; FENG, Hailing; ZHONG, Daidi. A Hazard Analysis

of Class I Recalls of Infusion Pumps. **Jmir Human Factors**, [S.L.], v. 6, n. 2, p. 1-15, 3 maio 2019. JMIR Publications Inc.. http://dx.doi.org/10.2196/10366.

[13] GARCÍA-VALLS, Marisol; PEREZ-PALACIN, Diego; MIRANDOLA, Raffaela. Pragmatic cyber physical systems design based on parametric models. **Journal Of Systems And Software**, [S.L.], v. 144, p. 559-572, out. 2018. Elsevier BV. http://dx.doi.org/10.1016/j.jss.2018.06.044.

[14] FINNEGAN, Anita; MCCAFFERY, Fergal. A Security Argument Pattern for Medical Device Assurance Cases. **2014 IEEE International Symposium On Software Reliability Engineering Workshops**, [S.L.], p. 220-225, nov. 2014. IEEE. http://dx.doi.org/10.1109/issrew.2014.89.

[15] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE STD 15026-2-2011: IEEE Standard-Adoption of ISO/IEC 15026-2:2011 Systems and Software Engineering-Systems and Software Assurance-Part 2: Assurance Case. New York: IEEE, 2011. 10 p. Disponível em: https://ieeexplore.ieee.org/document/6045293. Acesso em: 07 ago. 2022.

[16] JENSEN, Kurt; KRISTENSEN, Lars M.. Colored Petri nets. **Communications Of The ACM**, [S.L.], v. 58, n. 6, p. 61-70, 21 maio 2015. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/2663340.

[17] HATCLIFF, John; LARSON, Brian; CARPENTER, Todd; JONES, Paul; ZHANG, Yi; JORGENS, Joseph. The open PCA pump project: an exemplar open source medical device as a community resource. **ACM Sigbed Review**, [S.L.], v. 16, n. 2, p. 8-13, 16 ago. 2019. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/3357495.3357496.

[18] KANOUN, K.; ORTALO-BORREL, M.. Fault-tolerant system dependability-explicit modeling of hardware and software component-interactions. **IEEE Transactions On Reliability**, [S.L.], v. 49, n. 4, p. 363-376, 2000. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/24.922489.

[19] KITCHENHAM, B.; PICKARD, L.; PFLEEGER, S.L.. Case studies for method and tool evaluation. **IEEE Software**, [S.L.], v. 12, n. 4, p. 52-62, jul. 1995. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/52.391832.

[20] LITTLEWOOD, P. B.; BLOOMFIELD, R.; BAINBRIDGE, I.. **The use of computers in safety-critical applications**. Londres: Health And Safety Commission, 1998.

[21] MAJMA, Negar; BABAMIR, Seyed Morteza. Model-Based Monitoring and Adaptation of Pacemaker Behavior Using Hierarchical Fuzzy Colored Petri-Nets. **IEEE Transactions On Systems, Man, And Cybernetics:** Systems, [S.L.], v. 50, n. 9, p. 3344-3357, set. 2020. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tsmc.2018.2861718.

[22] MERTZ, Leslie. Automated Insulin Delivery: taking the guesswork out of diabetes management. **IEEE Pulse**, [S.L.], v. 9, n. 1, p. 8-9, jan. 2018. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/mpul.2017.2772685.

[23] MIAN, Zhibao; BOTTACI, Leonardo; PAPADOPOULOS, Yiannis; MAHMUD, Nidhal. Model transformation for analyzing dependability of AADL model by using HiP-HOPS. **Journal Of Systems And Software**, [S.L.], v. 151, p. 258-282, maio 2019. Elsevier BV. http://dx.doi.org/10.1016/j.jss.2019.02.019.

[24] MONTECCHI, Leonardo; LOLLINI, Paolo; BONDAVALLI, Andrea. A Template-Based Methodology for the Specification and Automated Composition of Performability Models. **IEEE Transactions On Reliability**, [S.L.], v. 69, n. 1, p. 293-309, mar. 2020. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tr.2019.2898351.

[25] NENCIONI, Gianfranco; HELVIK, Bjarne E.; HEEGAARD, Poul E.. Including Failure Correlation in Availability Modeling of a Software-Defined Backbone Network. **IEEE Transactions On Network And Service Management**, [S.L.], v. 14, n. 4, p. 1032-1045, dez. 2017. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tnsm.2017.2755462.

[26] RABAH, M.; KANOUN, K.. Performability evaluation of multipurpose multiprocessor systems: “separation of concerns” approach. **IEEE Transactions On Computers**, [S.L.], v. 52, n. 2, p. 223-236, fev. 2003. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tc.2003.1176988.

[27] RATHORE, Heena; WENZEL, Lothar; AL-ALI, Abdulla Khalid; MOHAMED, Amr; DU, Xiaojiang; GUIZANI, Mohsen. Multi-Layer Perceptron Model on Chip for Secure Diabetic Treatment. **IEEE Access**, [S.L.], v. 6, p. 44718-44730, 2018. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/access.2018.2854822.

[28] RAY, Arnab; CLEAVELAND, Rance. Security Assurance Cases for Medical Cyber–Physical Systems. **IEEE Design & Test**, [S.L.], v. 32, n. 5, p. 56-65, out. 2015. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/mdat.2015.2468222.

[29] SAIVES, Jeremie; FARAUT, Gregory; LESAGE, Jean-jacques. Automated Partitioning of Concurrent Discrete-Event Systems for Distributed Behavioral Identification. **IEEE Transactions On Automation Science And Engineering**, [s.l.], v. 15, n. 2, p.832-841, abr. 2018. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tase.2017.2718244.

[30] SAKIB, Kazi; TARI, Zahir; BERTOK, Peter. PETRI NETS. **Verification Of Communication Protocols In Web Services**, [s.l.], p.27-56, 18 out. 2013. John Wiley & Sons, Inc. http://dx.doi.org/10.1002/9781118720103.ch3.

[31] SCHWIERZ, Andreas; FORSBERG, Hakan. Assurance Case to Structure COTS Hardware Component Assurance for Safety-Critical Avionics. **2018 IEEE/Aiaa 37Th Digital Avionics Systems Conference (Dasc)**, [S.L.], p. 1-10, set. 2018. IEEE. http://dx.doi.org/10.1109/dasc.2018.8569774.

[32] SIVAKUMAR, M. S.; CASEY, Valentine; MCCAFFERY, Fergal; COLEMAN, Gerry. Improving Verification & Validation in the Medical Device Domain. **Systems, Software And Service Process Improvement**, [S.L.], p. 61-71, 2011. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-22206-1_6.

[33] SILVA, Lenardo; ALMEIDA, Hyggo; PERKUSICH, Angelo; PERKUSICH, Mirko. A Model-Based Approach to Support Validation of Medical Cyber-Physical Systems. **Sensors**, [S.L.], v. 15, n. 11, p. 27625-27670, 30 out. 2015. MDPI AG. http://dx.doi.org/10.3390/s151127625.

[34] SOBRINHO, Álvaro Alvares de Carvalho César. **Um Método para o Desenvolvimento e Certificação de Software de Sistemas Embarcados Baseado em Redes de Petri Coloridas e Casos de Garantia**. 2016. 133 f. Tese (Doutorado) - Curso de Pós-Graduação em Ciência da Computação, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2016. Disponível em: http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/660. Acesso em: 30 jul. 2022.

[35] SOBRINHO, Álvaro; SILVA, Leandro Dias da; PERKUSICH, Angelo; CUNHA, Paulo; CORDEIRO, Thiago; LIMA, Antonio Marcus Nogueira. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. **Software & Systems Modeling**, [S.L.], v. 18, n. 2, p. 1467-1485, 14 ago. 2017. Springer Science and Business Media LLC. http://dx.doi.org/10.1007/s10270-017-0616-7.

[36] WANG, Rui; KRISTENSEN, Lars Michael; MELING, Hein; STOLZ, Volker. Automated test case generation for the Paxos single-decree protocol using a Coloured Petri Net model. **Journal Of Logical And Algebraic Methods In Programming**, [S.L.], v. 104, p. 254-273, abr. 2019. Elsevier BV. http://dx.doi.org/10.1016/j.jlamp.2019.02.004.

[37] WASHIZAKI, H.; YAMAMOTO, H.; FUKAZAWA, Y.. A metrics suite for measuring reusability of software components. Proceedings. **5Th International Workshop On Enterprise Networking And Computing In Healthcare Industry (IEEE Cat. No.03Ex717)**, [S.L.], p. 211-223, 2003. IEEE Comput. Soc. http://dx.doi.org/10.1109/metric.2003.1232469.

[38] WESTERGAARD, Michael. Access/CPN 2.0: a high-level interface to coloured petri net models. **Applications And Theory Of Petri Nets**, [S.L.], p. 328-337, 2011. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-21834-7_19.

[39] WOODCOCK, Jim; LARSEN, Peter Gorm; BICARREGUI, Juan; FITZGERALD, John. Formal methods. **ACM Computing Surveys**, [S.L.], v. 41, n. 4, p. 1-36, out. 2009. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/1592434.1592436.

[40] ZHANG, Y.; JONES, P. L.; JETLEY, R.. A Hazard Analysis for a Generic Insulin Infusion Pump. **Journal Of Diabetes Science And Technology**, [S.L.], v. 4, n. 2, p. 263-283, 1 mar. 2010. SAGE Publications. http://dx.doi.org/10.1177/193229681000400207.