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

Um Método para o Desenvolvimento e Certificação de Software de Sistemas Embarcados Baseado em Redes de Petri Coloridas e Casos de Garantia

Álvaro Alvares de Carvalho César Sobrinho

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

Área de Concentração: Ciência da Computação

Prof. D.Sc. Angelo Perkusich
Prof. D.Sc. Leandro Dias da Silva
(Orientadores)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

C421m César Sobrinho, Álvaro Alvares de Carvalho.

Um método para o desenvolvimento e certificação de software de sistemas embarcados baseado em redes de petri coloridas e casos de garantia / Álvaro Álvares de Carvalho César Sobrinho. ï Campina Grande, 2016.

133 f. : il. color.

Tese (Doutorado em Ciência da Computação) ï Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2016.

"Orientação: Prof. Dr. Angelo Perkusich, Prof. Dr. Leandro Dias da Silvaò.

Referências.

1. Redes de Petri Coloridas. 2. Sistemas Embarcados. 3. Software Embarcado. 4. Casos de Garantia. I. Perkusich, Angelo. II. Silva, Leandro Dias da. III. Título.

CDU 004.7(043)

Resumo

Sistemas embarcados estão presentes em atividades diárias da população em geral, de ambientes domésticos até industriais e governamentais. O uso de sistemas embarcados tem aumentado como resultado, por exemplo, da disseminação da comunicação sem fio, de dispositivos eletrônicos com custos e tamanhos reduzidos, e de *software* embarcado em equipamentos eletrônicos. *Software* embarcado pode ser projetado como parte, desde sistemas embarcados simples para o controle de equipamentos domésticos, até sistemas críticos de segurança. Quanto mais complexo um sistema embarcado, maior a probabilidade de ocorrer situações adversas que ofereçam riscos financeiros, físicos, entre outros. Em sistemas embarcados críticos de segurança (e.g., médicos, aviônicos e aeroespaciais), falhas podem resultar em desastres naturais e danos à integridade física da população. Diante deste cenário, sistemas devem ser desenvolvidos de modo que sejam seguros e eficazes, e que estejam em conformidade com requisitos regulatórios. Portanto, um desafio importante que emerge dessa situação é o desenvolvimento de sistemas de acordo com sua especificação de requisitos, e ao mesmo tempo confiáveis e certificáveis. É no contexto de sistemas embarcados críticos de segurança que se insere esse trabalho. Propõe-se um método para o desenvolvimento e certificação de *software* desses sistemas. O método é baseado em redes de Petri coloridas (*Coloured Petri Nets - CPN*) e casos de garantia (*assurance cases*) representados com a notação estruturada por metas (*Goal Structuring Notation - GSN*). Conceitos associados com os processos de certificação prescritivo (padrões de processo) e baseado em metas (características de produto) são integrados durante o processo de desenvolvimento. Além disso, a definição e rastreabilidade de requisitos regulatórios e específicos do produto, juntamente com a verificação de conformidade com requisitos regulatórios, é realizada por meio de casos de garantia. Por fim, neste trabalho também é apresentado um estudo de caso sobre um sistema de Eletrocardiografia (ECG) configurado como um monitor cardíaco. Esse estudo de caso serve como cenário de implementação e avaliação experimental do método.

Abstract

Embedded systems are part of the general population's everyday life, from domestic, to industrial and governmental environments. The use of embedded systems has grown as a result, for example, of the dissemination of wireless communication, low power and portable electronic devices, and software embedded into electronic equipments. Embedded software can be designed to compose from simple embedded systems used to control domestic equipments, to safety-critical systems. The most complex an embedded system is, the more adverse situations are likely to occur, leading to financial risks, safety risks, among other. In safety-critical embedded systems (e.g., medical, avionics, and aerospace), failures may result in natural disasters and injuries to the population. Given this scenario, systems must be developed in order to be safe and effective, and to conform to regulatory requirements. Therefore, an important challenge that raises from this situation is to develop systems according to their requirements specification, and at the same time, being reliable and certifiable. This work is applied in the context of safety-critical embedded systems. A method to develop and certify software embedded in these systems is proposed. The method is based on Coloured Petri Nets (CPN) and assurance cases represented with the Goal Structuring Notation (GSN). Concepts related to prescriptive (process standards) and goal based (product features) certification processes are integrated during the development process. Moreover, the requirements specification and regulatory and product-specific requirements traceability, along with the verification of conformance to regulatory requirements, is carried out through assurance cases. Finally, a case study on an Electrocardiography (ECG) system configured as a cardiac monitor is presented. The case study is useful as an implementation scenario and experimental evaluation of the method.

Conteúdo

Glosário

ACES - *Assurance Cases Exchange Standard*

AD - *Analógico-Digital*

AGR - *Arquivo de Gerenciamento de Risco*

ACSR - *Algebra of Communicating Shared Resources*

ANSI - *American National Standards Institute*

ANVISA - *Agência Nacional de Vigilância Sanitária*

AOD - *Actor-Oriented Design*

CBD - *Certification-Based Development*

CPN - *Coloured Petri Nets*

CPS - *Cyber-Physical Systems*

CTL - *Computation Tree Logics*

DFT - *Discrete Fourier Transform*

DoD - *Department of Defense*

ECG - *Eletrocardiografia*

EEG - *Eletroencefalografia*

EGG - *Eletrogastrografia*

EMG - *Eletromiografia*

ESM - *Extended State Machines*

FAA - *Federal Aviation Administration*

FDA - *Food and Drug Administration*

FMECA - *Failure, Mode, Effects and Criticality Analysis*

FFT - *Fast Fourier Transform*

FTA - *Fault Tree Analysis*

FIR - *Finite Impulse Response*

GSN - *Goal Structuring Notation*

IDFT - *Inverse Discrete Fourier Transform*

IEC - *International Electrotechnical Commission*

IEEE - *Institute of Electrical and Electronics Engineers*

IFFT - *Inverse Fast Fourier Transform*

IIR - *Infinite Impulse Response*

ISO - *International Organization for Standardization*

MAE - *Mean Absolute Error*

MCPS - *Medical Cyber-Physical Systems*

MDPnP - *Medical Devices Plug-and-Play*

RTOS - *Real-Time Operating System*

RMSE - *Root Mean Squared Error*

RPN - *Risk Priority Number*

SME - *Sistemas Médicos Embarcados*

SOA - *Service-Oriented Architecture*

TRoS - *Timed and Resource-oriented Statecharts*

UML - *Unified Modeling Language*

XML - *Extensive Markup Language*

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Sistemas embarcados são equipamentos pouco visíveis que estão presentes no dia a dia da população. Estes sistemas incluem processadores, além de sensores e/ou atuadores, e são cada vez mais presentes em equipamentos médicos, automóveis, aeronaves, eletrodomésticos, eletroeletrônicos, entre outros. Por exemplo, tais sistemas são utilizados para monitorar o funcionamento cardíaco, controlar *airbags* em carros atuais, operações de voo em aeronaves, microondas em ambientes domésticos, e robôs em setores industriais [44]. A pouca visibilidade durante o consumo humano é uma das principais características que os diferenciam de computadores convencionais utilizados para o processamento de informações, tal como processadores de texto. Os sistemas embarcados são compostos por componentes de *hardware* e *software*. O *software* escrito para controlar funcionalidades desse tipo de sistema é denominado *software* embarcado.

Portanto, sistemas embarcados incluem desde equipamentos mais simples, até sistemas críticos de segurança. Sistemas críticos de segurança são sistemas nos quais falhas podem gerar situações indesejadas, e, consequentemente, resultar em riscos à integridade física de seres humanos. Neste caso, sistemas devem ser desenvolvidos de uma maneira que propriedades de segurança e eficácia sejam satisfeitas. Segurança significa a ausência de riscos inaceitáveis, enquanto que, eficácia significa atingir plenamente os resultados esperados. Segurança e eficácia são atributos muito importantes que devem ser considerados durante o desenvolvimento de sistemas embarcados críticos de segurança [47]. Por exemplo, na área médica, falhas em sistemas podem induzir cuidadores a erros durante o diagnóstico, monitoramento, e tratamento de pacientes.

Em determinados sistemas embarcados críticos de segurança (incluindo sensoriamento e atuação), denominados sistemas físico-cibernéticos (*Cyber-Physical Systems - CPS*), é necessário também considerar a integração de capacidades computacionais e físicas [8]. A integração de sistemas resulta na composição de sistemas de sistemas com o objetivo de lidar com problemas específicos. O comportamento deste tipo de sistema pode ser influenciado por variáveis relacionadas com o ambiente e o ser humano, e gerar propriedades emergentes como resultado da integração de sistemas. O marcapasso cardíaco é um exemplo de um tipo específico de sistema físico-cibernético (*Medical Cyber-Physical Systems - MCPS*) [39].

O desenvolvimento de sistemas embarcados críticos de segurança é geralmente composto por atividades principais de modelagem, concepção e análise (veja Figura 1.1, adaptada de [44]). Modelagem é uma atividade associada com o entendimento de projetistas sobre o comportamento de sistemas, na qual modelos são criados para representar sistemas e refletir suas propriedades. A atividade de concepção é associada com a definição de uma estruturada de artefatos de *hardware* e *software* para possibilitar a construção de sistemas (e.g., técnicas utilizadas). Análise é uma atividade utilizada para entender o por que um sistema possui um comportamento específico, seja desejado ou indesejado [44]. A condução dessas atividades é em determinados momentos sobreposta e o resultado de cada uma delas é utilizado para melhorar o projeto do produto.

Linguagens de especificação formal são ferramentas úteis para representar e simular a dinâmica discreta de sistemas embarcados durante as atividades de modelagem e concepção. Redes de Petri coloridas (*Coloured Petri Nets - CPN*) é um exemplo de linguagem para a modelagem formal e validação de sistemas complexos [34]. A partir da criação de modelos formais, é possível entender melhor o problema abordado e identificar, em etapas iniciais do projeto, problemas que normalmente seriam identificados após a construção de protótipos e versões finais de sistemas reais. O modelo gerado é uma maneira de representar um sistema na qual detalhes desnecessários são omitidos. Existem casos também (e.g., sistemas físico-cibernéticos) em que deve ser considerada a dinâmica contínua de sistemas utilizando técnicas de modelagem, tais como equações diferenciais. Além disso, modelos híbridos de sistemas podem ser construídos, caso exista a necessidade de integrar as dinâmicas discreta e contínua em uma especificação [43].

Entretanto, uma especificação formal de um sistema embarcado é geralmente construída

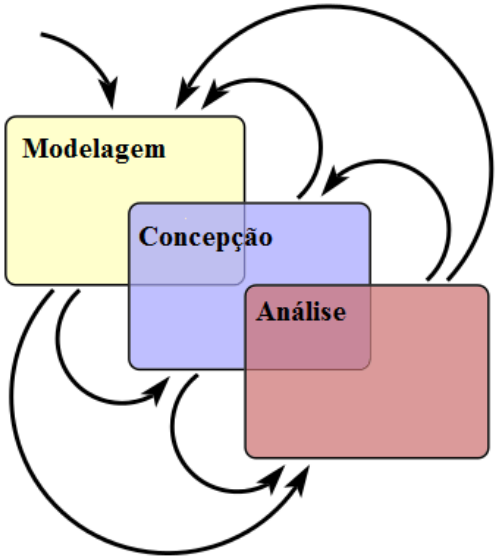


Figura 1.1: Esquema para as atividades principais no desenvolvimento de sistemas embarcados críticos de segurança.

a partir de uma descrição informal composta por requisitos funcionais e não funcionais. Portanto, problemas podem ser introduzidos durante a tradução entre representação informal e formal, e comportamentos indesejados podem não ser facilmente identificados no modelo formal construído. Neste caso, métodos de análise e verificação são úteis para representar comportamentos desejados e indesejados, e verificar se um modelo está em conformidade com sua especificação. A lógica temporal de árvore de computação (*Computation Tree Logics - CTL*) e a técnica de verificação de modelos (*model checking*) são exemplos de métodos associados com a atividade de análise. Propriedades podem ser especificadas com a lógica temporal CTL e sua validade verificada por meio do algoritmo de verificação de modelos [13].

Dado que o funcionamento incorreto de sistemas embarcados críticos de segurança pode gerar situações indesejadas e, consequentemente, riscos à integridade física de seres humanos, existe a preocupação de governos federais com a sua qualidade (e.g., atributos de segurança e eficácia). Governos controlam a comercialização de sistemas por meio de processos de certificação definidos e fiscalizados por agências reguladoras como uma maneira de aumentar a confiança no funcionamento desses sistemas. Ou seja, a certificação provida por entidades de supervisão, e não a autocertificação realizada por meio dos próprios

fabricantes. Pode-se observar historicamente que a autocertificação de sistemas críticos não é um processo eficaz de avaliação [51]. Portanto, agências reguladoras são responsáveis por avaliar se sistemas embarcados estão aptos para comercialização.

Algumas áreas passíveis de regulamentação governamental incluem a área médica, área de aviação, área de transporte terrestre, entre outras. Na área médica, por exemplo, a certificação de Sistemas Médicos Embarcados (SME) é útil para aumentar a confiança em seu funcionamento e evitar riscos à segurança de pacientes e operadores. Sistemas são liberados para comercialização, se, e somente se, agências responsáveis pela regulamentação concluírem que os mesmos são seguros e eficazes para a sua entrada no mercado. Agências que regulam SME, tais como a Agência Nacional de Vigilância Sanitária (ANVISA) no Brasil e a Administração de Alimentos e Drogas (*Food and Drug Administration - FDA*) nos Estados Unidos, exigem o uso de padrões prescritivos e a apresentação de relatórios técnicos (resultados de testes, verificação e validação, entre outros) por meio de processos prescritivos de certificação [17]. Por exemplo, a conformidade com a norma ISO 14971 [25] para a aplicação de gerenciamento de risco em sistemas médicos é solicitada pela ANVISA e FDA.

O processo prescritivo é a abordagem de certificação utilizada pela maioria das agências reguladoras de sistemas críticos de segurança. Nesta abordagem, agências reguladoras utilizam padrões prescritivos para avaliar se um determinado sistema atende aos requisitos necessários para comercialização. Padrões prescritivos são propostos e mantidos por agências internacionais de padronização, como, por exemplo, a Organização Internacional para Padronização (*International Organization for Standardization - ISO*) e a Comissão Eletrotécnica Internacional (*International Electrotechnical Commission - IEC*). Padrões são diretrizes compostas por metodologias e métodos relacionados com atividades específicas realizadas durante o ciclo de vida do desenvolvimento de sistemas. Métodos podem ser aplicados, por exemplo, para a avaliação de riscos durante a atividade de gerenciamento de riscos.

Por outro lado, problemas com o processo prescritivo de certificação em alguns setores da indústria encorajam agências reguladoras a aderirem ao uso de padrões baseados em metas durante o processo de certificação (*goal-based certification*) [51; 93; 94]. Padrões baseados em metas são compostos por metas de alto nível relacionadas com características específicas

de sistemas. A conformidade com as metas especificadas neste tipo de padrão é geralmente demonstrada por meio de casos de garantia (*assurance cases*) [50].

Casos de garantia são métodos úteis para argumentar que sistemas embarcados críticos de segurança satisfazem propriedades de segurança, eficácia, entre outras propriedades importantes. As argumentações geradas como casos de garantia podem ser estruturadas graficamente, tal como a notação estruturada por metas (*Goal Structuring Notation - GSN*) [60]. Algumas agências reguladoras já reconhecem atualmente a importância do uso de casos de garantia durante a certificação de sistemas críticos. Por exemplo, a FDA sugere o uso de casos de garantia na apresentação de evidências relacionadas com a segurança de SME, como é o caso da iniciativa de melhoramento de bombas de infusão [23]. A administração de aviação federal (*Federal Aviation Administration - FAA*) nos Estados Unidos, solicita o uso de casos de garantia de segurança (*safety cases*) relacionados com características específicas de sistemas de aeronaves não tripuladas durante seu processo de certificação [21].

O reconhecimento da importância de casos de garantia é embasado em grande parte por pesquisas acadêmicas conduzidas na área de sistemas embarcados críticos de segurança. Na área de aviação por exemplo, em um trabalho descrito por Denney e Pai [19], é apresentada uma metodologia para a montagem automática de casos de segurança juntamente com um estudo de caso sobre sistemas de aeronaves não tripuladas. Existem discussões também sobre o uso de modelos de casos de garantia como um tipo de padrão para a certificação de sistemas na área médica [94]. A área automotiva é outro exemplo no qual casos de garantia são apresentados como uma técnica promissora para certificação [100].

Neste contexto, a tese de doutorado apresentada neste documento está relacionada ao desenvolvimento de *software* de sistemas embarcados críticos de segurança que são passíveis de certificação, como por uma agência governamental. Estes sistemas incluem, mas não se limitam a: sistemas médicos, sistemas aviônicos e sistemas automotivos. Portanto, sistemas embarcados, como, por exemplo, controles remotos de aparelhos eletrônicos e aparelhos domésticos como microondas, estão fora do escopo deste trabalho.

1.1 Problemática

Software faz parte da maiorias das tecnologias de defesa e exploração mundiais (espaciais e aéreas) e de cuidado à saúde pública. Portanto, falhas em sistemas críticos de segurança (sistemas intensivos de *software*) podem causar desastres naturais catastróficos e expor a população a riscos que comprometam sua integridade física. A maioria das falhas em sistemas críticos, bem conhecidas e divulgadas publicamente, ocorreram por motivos de erros relacionados com a interação de *software* e *hardware* que deveriam ter sido identificados em fases iniciais de desenvolvimento.

Exemplos clássicos de falhas causadas por erros em *software* incluem a perda de *Mars Polar Lander* em 1999, o desastre *USS Yorktown* em 1998, e a explosão do *Ariane 5* em 1996. No caso da sonda espacial norte-americana *Mars Polar Lander*, foi determinado que um colapso no projeto de *software* resultou no corte dos motores de descida do veículo enquanto estava ainda a 40 metros acima da superfície [90]. No segundo exemplo, um navio de guerra norte-americano ficou imobilizado por horas em alto mar. A falha foi gerada pela introdução de um dado de entrada incorreto (zero) por um membro da tripulação, resultando em *overflow*[1] e corrupção de zona de memória. No terceiro exemplo, o foguete *Ariane 5* da agência espacial européia explodiu aproximadamente 40 segundos após um desvio de trajeto. Em análises foi identificado que o erro ocorreu pela produção de uma exceção ao tentar converter um número em ponto flutuante de 64 bits para um inteiro de 16 bits [58].

Atualmente, na área médica por exemplo, defeitos relacionados com componentes de *software* e *hardware* são ainda comuns em SME no mercado. A FDA registrou 5.294 notificações de defeitos em equipamentos médicos entre 2006 e 2011 [3]. Dentre as 5.294, 1.210 notificações foram associadas com equipamentos que possuem *software*. SME de classe II, como, por exemplo, eletrocardiógrafos foram associados a 90.5% das notificações. Trinta e três porcento das notificações foram relacionadas com *software*, enquanto que 66.7% foram distribuídas entre dispositivos de entrada e saída (e.g., sensores e botões), *hardware* (e.g., chips de memória e curto-circuito), bateria (e.g., suprimento, carga e descarga de energia), entre outros (e.g., reinicialização e documentação).

Neste contexto, alguns desafios relacionados com o desenvolvimento e certificação de

[1] Quando a quantidade finita se torna muito grande tal que sua representação se torna impossível [1].

sistemas embarcados críticos de segurança incluem, mas não se limitam a: conformidade com requisitos regulatórios, abordagem ágil no ambiente regulatório, engenharia de usabilidade, regulamentação de dispositivos em rede, e regulamentação de aplicações para dispositivos móveis [30].

O processo de certificação prescritivo possui problemas na garantia de confiança no funcionamento de sistemas embarcados críticos de segurança. Padrões prescritivos são documentos subjetivos escritos em linguagem natural no qual sua utilização correta depende da interpretação de seus requisitos por fabricantes de sistemas. A garantia de segurança obtida de maneira somente implícita durante o ciclo de vida de desenvolvimento, como é o caso neste processo, não é suficiente para garantir a segurança e eficácia do sistema [29]. Isto ocorre porque atributos relacionados com características do produto não são avaliados. Fabricantes somente analisam atributos relacionados com as atividades realizadas durante o ciclo de vida de desenvolvimento. Por exemplo, fabricantes de SME somente identificam, analisam, e aplicam medidas de controle durante o gerenciamento de risco utilizando a norma ISO 14971. A análise de características específicas relacionadas com os produtos não são requisitos definidos por agências reguladoras durante o desenvolvimento de suas especificações.

Por outro lado, no processo de certificação baseado em metas, a utilização de padrões baseados em metas também pode resultar em problemas, mesmo com o benefício da geração de garantias explícitas relacionadas com propriedades de sistemas. Ao utilizar casos de garantia, existem dificuldades, tais como a definição de grau de confiança em argumentos e evidências [26], definição de métricas para a avaliação de propriedades do sistema por agências reguladoras, e a integração de casos de garantia com outras atividades do ciclo de vida de desenvolvimento.

Métodos formais são técnicas úteis durante o desenvolvimento e certificação de sistemas críticos de segurança em um processo de certificação baseado em metas. Entretanto, o uso de métodos formais requer ferramentas de suporte e conhecimento especialista. Isso pode resultar em aumento de tempo e custos [66], o que, algumas vezes, desmotiva o uso deste tipo de método na indústria. Neste trabalho é argumentado que fabricantes podem reutilizar modelos de referência de sistemas específicos ou classes de sistemas para reduzir custos e tempo ao utilizar métodos formais durante o processo de desenvolvimento.

Além disso, agências reguladoras utilizam resultados de testes realizados em protótipos do sistema, como, por exemplo, comparações entre o protótipo e sistemas já certificados para avaliar se sistemas embarcados críticos de segurança em desenvolvimento são seguros e eficazes. Porém, quando defeitos são identificados somente em testes realizados com protótipos do sistema, ambos o projeto e o protótipo devem ser corrigidos aumentando custos de desenvolvimento e o tempo para entrada no mercado. A responsabilidade por avaliar protótipos de sistemas é geralmente terceirizada, como é o caso da verificação de qualidade e segurança de SME no Brasil pelo Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro)[2]. Em um cenário em que testes normalmente realizados em protótipos do sistema começam a ser conduzidos durante as fases iniciais do projeto, custos de desenvolvimento e o tempo para entrada no mercado poderiam ser reduzidos consideravelmente.

Um desafio importante, considerado nesta tese de doutorado, e relacionado ao desenvolvimento e certificação de sistemas embarcados críticos de segurança, é a identificação de mecanismos para integrar o processo de desenvolvimento de sistemas (incluindo o processo prescritivo de certificação) com a definição de casos de garantia. É interessante destacar que isso deve ser realizado de modo que possibilite a rastreabilidade entre requisitos (sejam estes relacionados ao processo ou produto) e artefatos de projeto (evidências), e inspeções regulatórias de maneira automatizada. A integração dos processos de certificação prescritivo e baseado em metas já é, por si só, um desafio importante a ser considerado por fabricantes e agências reguladoras de sistemas embarcados críticos de segurança.

1.2 Objetivo

O objetivo principal com este trabalho é estabelecer um método que resulte no aumento da confiança no funcionamento de *software* durante o desenvolvimento de sistemas embarcados críticos de segurança, como, por exemplo, sistemas médicos, aviônicos e automotivos. Para isso, é apresentado um método para o desenvolvimento e certificação de *software* de sistemas embarcados baseado em redes de Petri coloridas e casos de garantia, e na integração dos processos de certificação prescritivo e baseado em metas. Mais

[2]http://www.inmetro.gov.br/

especificamente, os objetivos perseguidos neste trabalho são:

1. definir o método para o desenvolvimento e certificação de *software* de sistemas embarcados;

2. realizar um estudo de caso sobre sistemas de aquisição de sinais biomédicos para a implementação e avaliação experimental do método.

A característica principal do método apresentado neste trabalho é definida pela integração dos processos de certificação prescritivo e baseado em metas durante o desenvolvimento e certificação de sistemas embarcados críticos de segurança. O método é composto por um conjunto de atividades definidas por meio de conceitos e técnicas associadas com padrões prescritivos (normas ISO, IEC, entre outras), padrões baseados em metas (casos de garantia), e métodos formais (CPN, verificação de modelos, entre outras). Cada uma das atividades se relacionam e produzem artefatos de projeto que podem ser reaproveitados durante o processo de certificação.

O estudo de caso definido como objetivo específico é útil tanto para avaliar experimentalmente o método apresentado, quanto para descrever um cenário de uso que pode ser replicado por fabricantes de sistemas embarcados. Um sistema de Eletrocardiografia (ECG) foi o sistema de aquisição de sinal biomédico escolhido para o estudo de caso. Apesar de ser um sistema embarcado para sensoreamento, sistemas de ECG podem ser considerados como críticos porque resultados de medições inconsistentes podem induzir cuidadores a erros durante o tratamento e diagnóstico de pacientes (e.g., em um cenário de unidade de terapia intensiva). Além disso, sistemas de ECG fazem parte de sistemas mais complexos, como, por exemplo, sistemas de marcapassos cardíacos.

A linguagem de especificação formal CPN e a técnica de verificação de modelos foram utilizadas para modelar, verificar, e validar o processo de gerenciamento de risco baseado na norma ISO 14971. Além disso, requisitos funcionais e não funcionais do sistema de ECG foram modelados, verificados, e validados utilizando CPN, a transformada de Fourier, a técnica de verificação de modelos, e as métricas de raiz quadrada do erro quadrático médio (*Root Mean Squared Error - RMSE*) [91] e erro absoluto médio (*Mean Absolute Error - MAE*) [4]. Argumentos e evidências relacionadas ao funcionamento correto do sistema de

ECG foram representados como casos de garantia estruturados com a notação estruturada por metas (*Goal Structuring Notation - GSN*) [60].

Apesar do foco em sistemas de aquisição de sinais biomédicos durante o estudo de caso, o método proposto neste trabalho pode ser aplicado no desenvolvimento e certificação de outros sistemas embarcados críticos de segurança. Neste contexto, fabricantes de sistemas devem analisar características específicas de cada um deles para definir as técnicas utilizadas na aplicação do método. Por exemplo, a transformada de Fourier, utilizada durante a modelagem e validação no estudo de caso, foi definida especificamente para sistemas de aquisição de sinais biomédicos. Consequentemente, é necessário analisar se esta técnica é adequada para outros produtos considerando características específicas de cada tipo de sistema embarcado crítico de segurança desenvolvido.

1.3 Justificativa e Relevância

Sistemas embarcados críticos de segurança estão presentes no dia a dia da população no transporte aéreo, transporte terrestre, cuidados médicos, entre outras atividades. Como descrito anteriormente, riscos de ocorrência de desastres naturais catastróficos e exposição da população a situações perigosas que comprometam a sua integridade física são eminentes. Riscos podem ser resultados do funcionamento incorreto de sistemas, e são as principais justificativas para a proposta de soluções que auxiliem na sua redução. Por exemplo, as situações catastróficas ocorridas por erros no *software* em *Mars Polar Lander*, *USS Yorktown*, e *Ariane 5*, são indicadores históricos sobre a relevância da garantia de segurança e eficácia de sistemas. No caso específico de SME, notificações de defeitos em sistemas disponíveis no mercado indicam a necessidade da disponibilização de abordagens que contribuam para aumentar a confiança no funcionamento deste tipo de sistema crítico [3].

Existe a falta de consenso entre pesquisadores, agências reguladoras e fabricantes sobre qual o processo de certificação de sistemas embarcados críticos de segurança mais eficaz, qual deve ser utilizado, e mesmo se estes devem ser integrados (processos prescritivos e baseados em metas). A necessidade de adesão a um processo de certificação utilizando padrões baseados em metas é destacada em alguns trabalhos acadêmicos [74; 51; 18]. Em outros trabalhos são apresentados argumentos sobre a integração dos processos de

certificação prescritivo e baseado em metas [87; 93]. Isso justifica a criação de novas pesquisas relacionadas ao desenvolvimento e certificação de sistemas embarcados críticos de segurança.

Abordagens têm sido propostas para auxiliar fabricantes e agências reguladoras na utilização de padrões baseados em metas [85] e métodos formais [61; 54; 38]. Entretanto, existe a carência de pesquisas associadas com a proposta de abordagens para auxiliar a integração dos processos de certificação prescritivo e baseado em metas. A viabilidade da integração destes processos de certificação tem sido destacada em pesquisas recentes, porém, métodos que possibilitem esta integração ainda não foram suficientemente investigados. A integração pode beneficiar fabricantes e agências reguladoras com as melhores características de cada um dos processos de certificação. Por exemplo, um método deste tipo pode auxiliar fabricantes de sistemas embarcados na diminuição de custos de desenvolvimento como resultado da identificação de defeitos durante as fases iniciais no projeto do sistema embarcado.

A relevância para o desenvolvimento deste trabalho está associada com contribuições obtidas pela disponibilização de um método para aumentar a confiança no funcionamento de sistemas embarcados críticos de segurança, e com os resultados obtidos no estudo de caso realizado sobre sistemas de aquisição de sinais biomédicos. Uma vez que o método está disponível, fabricantes de sistemas embarcados críticos de segurança podem aplicá-lo para criar sistemas mais seguros e eficazes, e reutilizar os artefatos gerados durante a aplicação do método como evidências para certificação. Por outro lado, agências reguladoras podem se beneficiar com o método apresentado ao receber os resultados obtidos por fabricantes de maneira estruturada por meio de casos de garantia, e realizar avaliações de requisitos regulatórios (como casos de garantia em GSN). Portanto, a disponibilização do método para o desenvolvimento e certificação de sistemas embarcados críticos de segurança é a principal contribuição obtida com a realização desta tese de doutorado.

É interessante também destacar a importância da formalização e padronização dos conceitos de casos de garantia em GSN, e a integração desses conceitos com o processo de desenvolvimento de sistemas embarcados críticos de segurança. No primeiro caso, um padrão para a representação e compartilhamento de casos de garantia baseado na linguagem de marcação extensível (*Extensive Markup Language - XML*) e na notação gráfica GSN [60]

é definido como parte do método apresentado. Por outro lado, para a utilização de casos de garantia durante o processo de desenvolvimento, atividades realizadas durante o processo de engenharia de requisitos são relacionadas com a definição de casos de garantia em GSN durante a aplicação do método. A definição e rastreabilidade de requisitos de produto e processo por meio de casos de garantia são atividades relevantes, e são consideradas na definição do método apresentado. Isso possibilita que fabricantes analisem relações entre requisitos e artefatos de projeto, e que agências reguladoras avaliem argumentos e evidências apresentadas de maneira automatizada.

O estudo de caso apresentado não foi somente útil para avaliar experimentalmente o método proposto, mas também para demonstrar como fabricantes de sistemas embarcados críticos de segurança podem aplicá-lo. Além disso, os artefatos gerados durante o estudo de caso sobre sistemas de aquisição de sinais biomédicos são considerados como outras contribuições obtidas neste trabalho. Por exemplo, tanto o modelo CPN construído para representar os requisitos da norma ISO 14971, quanto o modelo CPN de referência de sistemas de aquisição de sinais biomédicos podem ser reutilizados por fabricantes durante o processo de desenvolvimento de SME em projetos comerciais.

1.4 Organização do Documento

Este documento de tese de doutorado está estruturado em 6 capítulos. No primeiro capítulo foi apresentada uma introdução sobre o trabalho, composta por contextualização do tema pesquisado, problemática, objetivos principal e específicos, e justificativa e relevância para a execução dessa pesquisa. O restante dos capítulos está organizado da seguinte maneira:

- no Capítulo 2 são introduzidos conceitos sobre redes de Petri coloridas, casos de garantia, a notação estruturada por metas, e a transformada de Fourier. Estes conceitos foram utilizados durante o desenvolvimento deste trabalho;
- no Capítulo 3 são apresentados trabalhos relacionados com o tema de pesquisa;
- no Capítulo 4 é apresentado o método proposto para o desenvolvimento e certificação de sistemas embarcados críticos de segurança. O método proposto é o objetivo

principal com a realização deste trabalho;

- no Capítulo 5 é descrito o estudo de caso sobre sistemas de aquisição de sinais biomédicos conduzido para avaliar experimentalmente o método proposto e apresentar um cenário de uso. Um sistema de ECG configurado para o monitoramento de pacientes foi estudado;

- no Capítulo 6 são apresentadas conclusões obtidas com a realização deste trabalho, bem como a descrição de algumas direções futuras de pesquisa.

Capítulo 2

Fundamentação Teórica

Neste capítulo são definidos conceitos sobre as principais técnicas utilizadas durante a definição do método apresentado no Capítulo 4 e na realização do estudo de caso apresentado no Capítulo 5. Mais especificamente, são descritos conceitos fundamentais sobre as redes de Petri coloridas (*Coloured Petri Nets - CPN*), casos de garantia (*assurance cases*), a notação estruturada por metas (*Goal Structuring Notation - GSN*), e a transformada de Fourier.

2.1 Redes de Petri Coloridas

Carl A. Petri apresentou no inicio da década de 1960 a linguagem matemática e gráfica redes de Petri como uma alternativa para modelar e analisar sistemas de eventos discretos. Redes de Petri é útil para representar sistemas concorrentes, assíncronos e distribuídos [56]. Existem classes de formalismos baseados em redes de Petri, nos quais a quantidade de informação representada por marcações de lugares os diferencia (redes de Petri de alto e baixo nível). A linguagem de modelagem CPN é um tipo de rede de Petri de alto nível. O trabalho apresentado por Desel e Reisig [20] é recomendável para leitores que não possuem conhecimentos básicos sobre redes de Petri.

No início da década de 1980, Kurt Jensen apresentou em usa tese de doutorado a linguagem formal CPN para possibilitar a modelagem e validação de sistemas complexos [33]. A linguagem de modelagem CPN é uma extensão de redes de Petri que inclui recursos de linguagem de programação por meio da linguagem CPN ML. Dentre os recursos disponíveis, pode-se destacar a manipulação de variáveis, operações matemáticas, estruturas

condicionais e restrições de tempo.

A linguagem CPN ML é uma extensão da linguagem de programação funcional ML, na qual inclui sintaxe para a declaração de conjuntos de cor (*color sets*) e tipos de variáveis. O uso do padrão ML prove expressividade para modelar e manipular dados em escala industrial, além de ser utilizado na implementação de simulações, análise de espaços de estado, e análise de desempenho. Tipos básicos de dados herdados do padrão ML são utilizados como conjuntos de cor simples (por exemplo, *string* e *int*), enquanto que produtos de conjuntos de cor podem ser definidos a partir desses tipos básicos. Por outro lado, com o sistema de tipos CPN ML, pode-se verificar automaticamente a consistência de expressões definidas em modelos CPN.

Modelos CPN são compostos por lugares (elipses), transições (retângulos) e arcos (setas) [97]. Cada lugar possui um tipo associado denominado conjuntos de cor e pode conter uma marcação inicial. Conjuntos de cor são utilizadados para representar tipos de dados permitidos em cada lugar, denominados cores de fichas (*token colors*). Arcos de entrada conectam lugares a transições e arcos de saída conectam transições a lugares. Uma marcação representa o estado de um modelo CPN que consiste da distribuição de fichas por todos os lugares.

A simulação de modelos CPN é baseada no disparo de transições. Uma transição é somente habilitada se regras para o disparo de transições são satisfeitas, tais como regras associadas com expressões de arco em arcos de entrada. Por exemplo, a quantidade de fichas em todos os lugares de entrada deve ser maior ou igual ao peso dos arcos de entrada conectados a transições. Projetistas de modelos CPN utilizam inscrições de rede para definir conjuntos de cor, condições de guarda, marcas de tempo (*time stamps*), segmentos de código, expressões de arco, entre outras.

Uma Rede de Petri Colorida não hierárquica (*non-hierarchical Coloured Petri Net*) [34] é definida como 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.
2. T é um conjunto finito de transições tal que $P \cap T = \theta$.
3. $A \subseteq P \times T \cup T \times P$ é um conjunto de arcos direcionados.
4. Σ é um conjunto não vazio finito de conjuntos de cores.

5. V é um conjunto finito de variáveis tipadas tal que $Tipo[v] \in \Sigma$ para todas as variáveis $v \in V$.

6. $C : P \rightarrow \Sigma$ é uma função de conjunto de cor que associa um conjunto de cor a cada lugar.

7. $G : T \rightarrow EXPR_V$ é uma função de guarda que associa uma guarda a cada transição t tal que $Tipo[G(t)] = Bool$.

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 $Tipo[E(a)] = C(p)_{MS}$[1], no qual p é o lugar conectado ao arco a.

9. $I : P \rightarrow EXPR_\theta$ é uma função de inicialização que associa uma expressão de inicialização a cada lugar p tal que $Tipo[I(p)] = C(p)_{MS}$.

Por outro lado, Huber et al. introduziu o conceito de CPN hierárquica em 1990 [35]. Com este conceito, projetistas podem criar composições de modelos CPN por meio de módulos CPN reutilizáveis. Modularidade é útil para simplificar modelos complexos e extensos ao separa-los em vários módulos com tamanhos reduzidos. Um módulo de Rede de Petri Colorida (*Coloured Petri Net Module*) é definido como uma tupla de quatro elementos $CPN_M = (CPN, T_{sub}, P_{porta}, PT)$, na qual:

1. $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ é uma Rede de Petri Colorida não hierárquica.

2. $T_{sub} \subseteq T$ é um conjunto de transições de substituição.

3. $P_{porta} \subseteq P$ é um conjunto de lugares porta.

4. $PT : P_{porta} \rightarrow IN, OUT, I/O$ é uma função de tipo porta que associa tipos de porta a lugares.

Portanto, uma Rede de Petri Colorida hierárquica (*Hierarchical Coloured Petri Net*) é também definida como uma tupla de quatro elementos $CPN_H = (S, SM, PS, FS)$, na qual:

[1] MS está relacionado a "multiconjunto".

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, Σ^s, V^s, C^s, G^s, E^s, I^s), T^s_{sub}, P^s_{porta} , PT^s). É necessário que ($P^{s_1} \cup T^{s_1}$)$\cap$($P^{s_2} \cup T^{s_2}$) = θ para todo s_1, $s_2 \in S$ tal que $s_1 \neq s_2$.

2. $SM : T_{sub} \rightarrow S$ 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.

3. PS é uma função de relação porta-socket que associa uma `relação porta-socket` $PS(t) \subseteq P_{sock}(t) \times P^{SM(t)}_{porta}$ a cada transição de substituição t. É necessário que $ST(p)$ = $PT(p'), C(p)$ = $C(p')$ e $I(p)\langle\rangle$ para todo (p, p') $\in PS(t)$ e todo $t \in T_{sub}$.

4. $FS \subseteq 2^P$ é um conjunto 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 $fs \in FS$.

Neste contexto, é possível editar, simular e analisar (e.g., análise de espaços de estado e verificação de modelos) modelos CPN hierárquicos ou não hierárquicos, temporizados ou não, utilizando a ferramenta CPN/Tools. A linguagem CPN ML é utilizada no CPN/Tools para realizar declarações e definir inscrições de rede. A ferramenta contém características, tais como simulações eficientes de redes temporizadas e não temporizadas, geração e análise de espaços de estado parciais e completos, e relatórios padrões de espaços de estado relacionados com propriedades de vivacidade (*liveness*), limitação (*boundedness*), justiça (*fairness*), e casa (*home*). O CPN/Tools é o resultado de um projeto de pesquisa, denominado projeto CPN2000, conduzido na Universidade de Aarhus [36].

Exemplo Filósofos. Na Figura 2.1 é apresentado um exemplo clássico de um modelo CPN não hierárquico para o sistema de filósofos[2]. Considere cinco filósofos sentados em uma mesa redonda circular, na qual existe um prato de arroz posicionado no meio da mesa. Cada filósofo alterna entre pensar e se alimentar. Considere também que um filósofo necessita de dois *chopsticks* para se alimentar e que ele somente pode utilizar os que estão exatamente ao seu lado (previne que dois vizinhos se alimentem ao mesmo tempo). Além disso, em algum momento um filósofo é envenenado após se alimentar. Note que o modelo é composto por quatro transições e quatro lugares. As inscrições `Chopsticks(p)`

[2]http://cpntools.org/documentation/examples/dining_philosophers_sp

e `DeixaChopstick(p)` são funções associadas com ações de selecionar e liberar *chopsticks*.

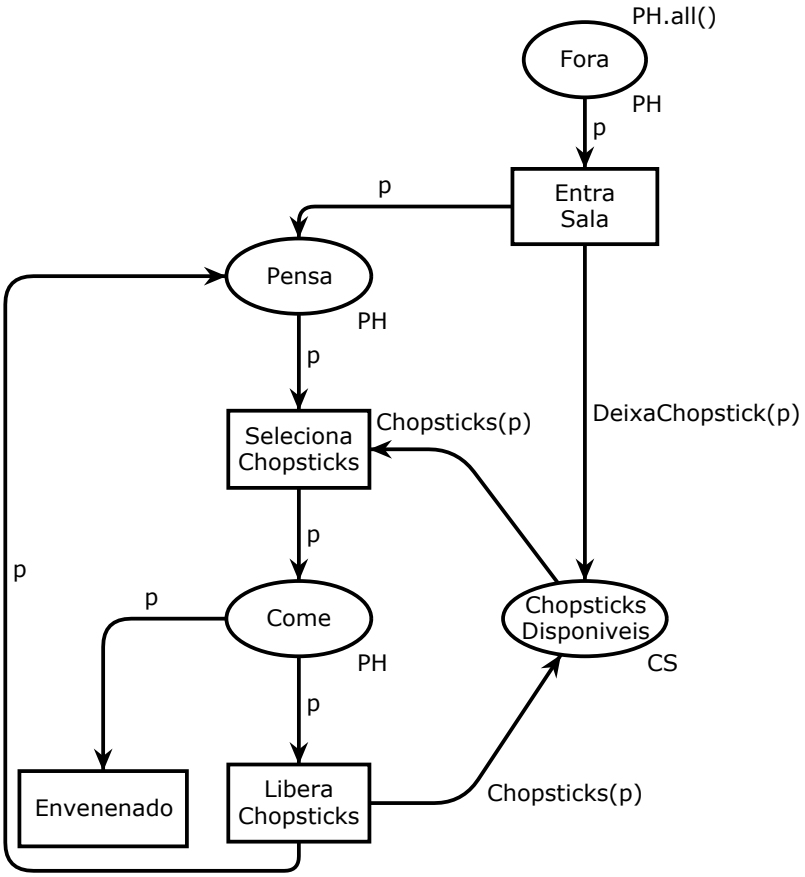


Figura 2.1: Exemplo clássico de um modelo CPN para o sistema de filósofos.

2.1.1 Espaços de Estado

Um espaço de estado é um grafo direcionado $G = (N, E)$ no qual existe um nó $n \in N$ para cada marcação alcançável da rede e um arco $e \in E$ para cada ocorrência de elemento de ligação. Existe um arco rotulado no espaço de estado com um elemento de ligação (t, b) a partir de um nó representando a marcação M_1 para um nó representando a marcação M_2, se e somente se, o elemento de ligação (t, b) estiver habilitado em M_1 e a ocorrência de (t, b) resulta na marcação M_2. Em um elemento de ligação (t, b), t consiste de uma transição, e b consiste de uma ligação, na qual b de uma transição t é uma função mapeada para cada variável v da transição t para um valor $b(v)$ pertencente ao tipo da variável v (i.e., $b(v) \in Tipo[v]$).

Na maioria dos casos, a geração do espaço de estado é seguida pela geração do grafo de componentes fortemente conectados (*Strongly Connected Components - SCC*). Um grafo direcionado $G = (N, E)$ é definido como conectado se existe um caminho a partir de um nó $n_1 \in N$ para $n_2 \in N$, e também do nó n_2 para o nó n_1. Um componente fortemente conectado de um grafo direcionado $G = (N, E)$ (subgrafo de G) é um conjunto máximo de vértices (nós) $C \subseteq N$ tal que para todo par de vértice n_1 e n_2, existe um caminho direto de n_1 para n_2 e de n_2 para n_1. O grafo fortemente conectado é utilizado pela ferramenta de espaço de estado do CPN/Tools para determinar propriedades comportamentais padrões de modelos.

Considerando o exemplo do sistema de filósofos, na Figura 2.2 é apresentado o espaço de estado do modelo ilustrado na Figura 2.1 gerado por meio da ferramenta de espaços de estado do CPN/Tools. Note que o espaço de estado é composto por vinte estados, nos quais somente foram expandidas informações sobre fichas contidas em lugares para os estados 4, 13, 16 e 17. Além disso, informações sobre elementos de ligação somente foram expandidas para a transição entre os estados 4 e 5, e entre os estados 8 e 12. O restante das informações foram suprimidas por motivo de espaço e legibilidade.

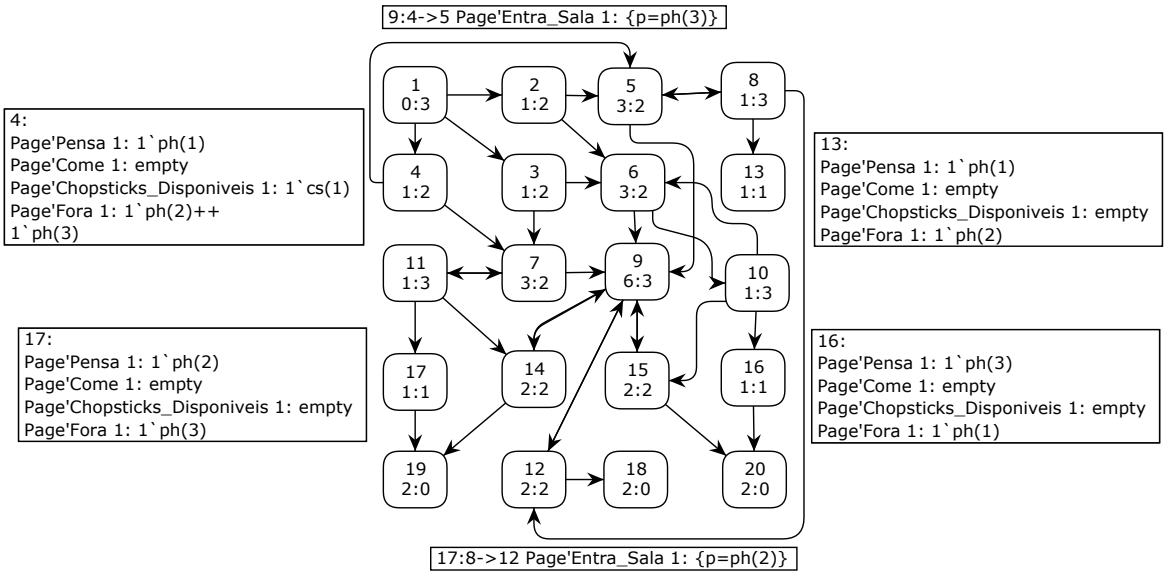


Figura 2.2: Espaço de estado do modelo CPN para o sistema de filósofos.

2.1.2 Análise de Espaços de Estado

O espaço de estado de um modelo pode ser utilizado para verificar propriedades comportamentais padrões (*liveness*, *boundedness*, *fairness*, *home*) por meio de um relatório de espaço de estado disponibilizado pela ferramenta CPN/Tools. O relatório de espaço de estado é produzido automaticamente pela ferramenta CPN/Tools e é composto por informações estatísticas (relacionadas ao tamanho do espaço de estado) e propriedades comportamentais padrões. Por exemplo, propriedades comportamentais padrões podem ser investigadas da seguinte maneira:

- vivacidade (*liveness*): verificação da existência de transições vivas ou mortas em um modelo CPN;
- limitação (*boundedness*): verificação de limites inferiores e superiores de fichas contidas em lugares;
- justiça (*fairness)*: verificação da existência de disparos infinitos de transições em um modelo CPN;
- casa (*home)*: verificação de existência de lugares que podem ser alcançados a partir de qualquer estado do espaço de estado.

Entretanto, além de propriedades padrões, propriedades mais específicas de modelos podem ser verificadas utilizando funções de consulta predefinidas (escritas com a linguagem CPN ML) disponíveis na ferramenta de espaço de estado. As funções de consulta predefinidas são utilizadas também na ferramenta de espaço de estado do CPN/Tools para gerar o relatório de espaço de estado. Um exemplo de função de consulta predefinida é a função `Reachable`, útil para determinar se existe uma sequência de ocorrências a partir da marcação do primeiro nó até a marcação de um segundo nó. Aplicando a função

```
Reachable'(1,9)
```

no espaço de estado apresentado na Figura 2.2, é possível identificar, por exemplo, que existe um caminho a partir do nó 1 até o nó 9 definido como [1, 3, 7, 9]. Outros exemplos incluem funções para verificar a existência de marcações mortas (`DeadMarking` e `AllDeadMarkings`) e instâncias de transições mortas (`TIsDead` e `ListDeadTIs`).

A documentação descrita em [37] é recomendável para mais informações sobre as especificações da ferramenta de espaço de estado do CPN/Tools.

2.1.3 Verificação de Modelos - ASK-CTL

A verificação de modelos é um método automático utilizado para verificar propriedades representadas em modelos de sistemas. O algoritmo de verificação de modelos é utilizado para investigar se uma determinada propriedade especificada em lógica temporal é satisfeita por um modelo formal. A única atividade realizada pelo usuário durante a verificação de modelos é a especificação de propriedades e a execução do algoritmo [13].

O algoritmo para a verificação de modelos CPN é executado utilizando a biblioteca ASK-CTL e o CPN/Tools. ASK-CTL é uma extensão da lógica temporal de árvore de computação (*Computation Tree Logics - CTL*) que permite a representação de fórmulas de estado e transição sobre espaços de estado CPN. Os principais operadores lógicos ASK-CTL são descritos a seguir:

- $EV(\varphi)$ - significa que é eventualmente possível alcançar um estado onde a fórmula φ é satisfeita;
- $POS(\varphi)$ - significa que é possível alcançar um estado onde a fórmula φ é satisfeita;
- $EXIST\_NEXT(\varphi)$ - significa que existe um estado sucessor imediato onde a fórmula φ é satisfeita;
- $FORALL\_NEXT(\varphi)$ - significa que a fórmula φ é satisfeita para todos os estados sucessores imediatos;
- $MODAL(\varphi)$ - significa a alteração entre fórmulas de estado e transição;
- $NF(<$ expressão de nó $>)$ - Significa uma função de nó para subfórmulas de estado;
- $AF(<$ expressão de arco $>)$ - significa uma função de arco para subfórmulas de transição;
- $NOT(\varphi)$, $AND(\varphi, \varphi')$, e $OR(\varphi, \varphi')$ - possui a mesma interpretação padrão do $\neg$, $\wedge$, e $\vee$, respectivamente.

Esses operadores são utilizados para percorrer os caminhos obtidos na geração do espaço de estado do modelo CPN. Os operadores ASK-CTL `EV`, `POS`, `EXIST_NEXT`, e `FORALL_NEXT` possuem o mesmo significado dos operadores básicos CTL `AF`, `EF`, `EX`, e `AX`, respectivamente. Exemplos de árvores geradas com os operadores `EV`, `POS`, `EXIST_NEXT`, e `FORALL_NEXT` disponibilizados em ASK-CTL são apresentados na Figura 2.3. Os círculos destacados são estados nos quais uma propriedade específica é satisfeita. Com a aplicação do operador `EV` $\equiv$ `AF`, a fórmula φ é satisfeita para todos os caminhos, em algum momento no futuro. Para o operador `POS` $\equiv$ `EF`, a fórmula φ é satisfeita para pelo menos um caminho em algum momento no futuro. Para a junção dos operadores `EV` $\equiv$ `AF` e `EXIST_NEXT` $\equiv$ `EX`, a fórmula φ é satisfeita para todos os caminhos em algum momento no futuro considerando um sucessor imediato. Por fim, para a junção dos operadores `POS` $\equiv$ `EF` e `FORALL_NEXT` $\equiv$ `AX`, a fórmula φ é satisfeita para pelo menos um caminho em algum momento no futuro considerando todos os seus sucessores imediatos.

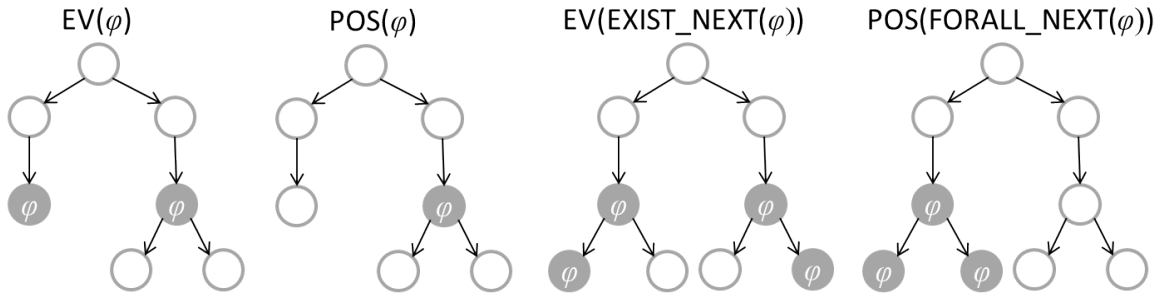


Figura 2.3: Exemplos de árvores geradas para representar o comportamento ao aplicar operadores ASK-CTL.

Para exemplificar a aplicação da verificação de modelos com ASK-CTL, considere o modelo CPN (Figura 2.1) e o espaço de estado (Figura 2.2) para o exemplo do sistema de filósofos apresentado anteriormente. Uma característica deste sistema é que, em algum momento, um dos filósofos acaba envenenado após se alimentar. Portanto, esta propriedade deve ser satisfeita no modelo CPN especificado. A fórmula ASK-CTL

```
MODAL(POS(AF("ph(3) envenenado", Envenenado 3)));
```

pode ser utilizada para representar esta propriedade. Dado que a propriedade está especificada, a função `eval_node` disponível na biblioteca ASK-CTL é aplicada para

executar a verificação. O resultado da verificação de modelos para a propriedade descrita anteriormente é apresentado na Figura 2.4. Note que o resultado da verificação foi verdadeiro, significando que a propriedade é satisfeita no modelo CPN. A documentação descrita em [12] é recomendável para mais informações sobre as especificações da biblioteca ASK-CTL.

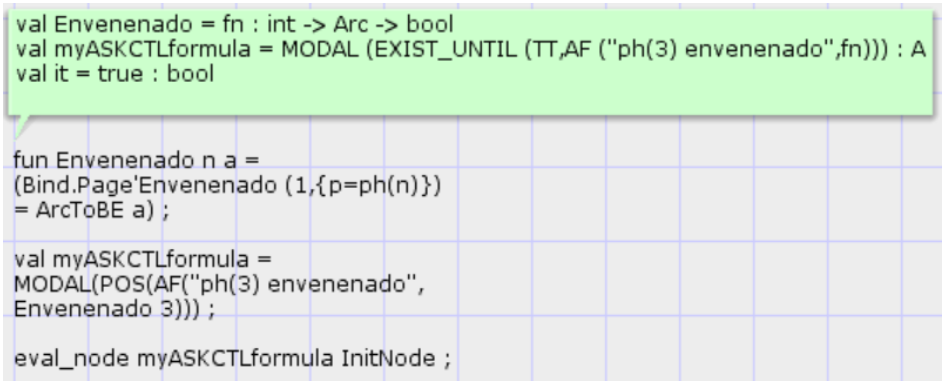


Figura 2.4: Exemplo de aplicação da verificação de modelos com ASK-CTL e a ferramenta CPN/Tools.

2.2 Caso de Garantia e a Notação Estruturada por Metas

Caso de garantia (*assurance case*) é um método composto por argumentações utilizadas para demonstrar a conformidade de um sistema em relação as suas especificações. Um argumento é um conjunto de declarações interconectadas sobre a conformidade de desenvolvedores com propriedades de um sistema. Argumentos são construídos utilizando afirmações e evidências:

- afirmações são declarações (proposições) sobre propriedades relacionadas aos sistemas;
- evidências são informações relevantes utilizadas para inferir a presença de determinadas propriedades em sistemas.

O resultado na criação de um caso de garantia é geralmente um documento descrito em linguagem natural que contém argumentos convincentes suportados por evidências [22]. A conformidade com propriedades de segurança, confiança, disponibilidade, eficácia e

desempenho de sistemas, pode ser utilizada como argumento em casos de garantia [26]. Um exemplo de propriedade que pode ser afirmada como satisfeita inclui que o sistema é livre de impasses (*deadlocks*) [32]. Neste caso, exemplos de evidências que podem ser utilizadas para suportar esta afirmação incluem resultados da análise de espaço de estado do modelo do sistema.

Um caso de garantia é uma forma rigorosa de criar argumentos sobre a confiança em sistemas. Esse método está relacionado com atividades realizadas durante o desenvolvimento de produtos (e.g., especificação de requisitos, projeto, implementação, verificação, e implantação). A utilização deste método pode beneficiar desenvolvedores com a redução de custos com certificação e melhoria na qualidade de sistemas. Casos de garantia podem propiciar a avaliação mais fácil e rápida da confiança em afirmações sobre propriedades de sistemas em um processo de revisão realizado por agências reguladoras. Por outro lado, desenvolvedores podem ser beneficiados com menor ciclo de desenvolvimento como consequência de menos defeitos identificados por agências reguladoras, e redução de tempo para aprovação.

Exemplos de casos de garantia especializados incluem os que representam questões sobre segurança de usuários (*safety case*), segurança de sistemas (*security case*), usabilidade (*usasility case*), e confiança (*dependability case*). Como um tipo de caso de garantia, os casos de segurança, por exemplo, são compostos por afirmações, argumentos, e evidências. Afirmações são requisitos de segurança ou objetivos. Argumentos justificam a aceitabilidade do sistema com relação a segurança baseada em evidências. Evidências servem para apoiar argumentos sobre o funcionamento de sistemas visando aumentar o nível de confiança em sua segurança. Ou seja, casos de segurança devem incluir evidências de segurança que proporcionem suporte aos argumentos sobre o comportamento de sistemas em um determinado contexto para ganhar confiança na segurança do sistema (requisitos de segurança ou objetivos).

Um argumento sem evidência é infundado, e uma evidência sem argumento é inexplicável. Na Figura 2.5 (adaptada de [27]) é apresentado o relacionamento entre os três componentes de casos de segurança (requisitos, argumentos e evidências). A ligação entre evidências e afirmações é realizada em argumentos de segurança. Uma afirmação que um sistema é seguro (e.g., o *software* é livre de falhas) deve ser associada com evidências (e.g.,

análise de árvore de falhas (*Fault Tree Analysis - FTA*) e modelo formal) por intermédio de argumentos de segurança (e.g., satisfaz todos os requisitos de segurança).

Entretanto, um erro recorrente na construção de casos de garantia é a falta de explicação (argumentação) para relacionar afirmações com evidências apresentadas. A falta de argumentação dificulta o entendimento do leitor (e.g., uma agência reguladora) sobre o cumprimento de requisitos funcionais e não funcionais do sistema. Além disso, a quantidade de informações descrita em um caso de garantia pode dificultar o entendimento e avaliação dos argumentos construídos. Neste contexto, alguns padrões têm sido propostos para auxiliar a organização e estruturação de casos de garantia, tal como a notação estruturada por metas (*Goal Structuring Notation - GSN*) [60].

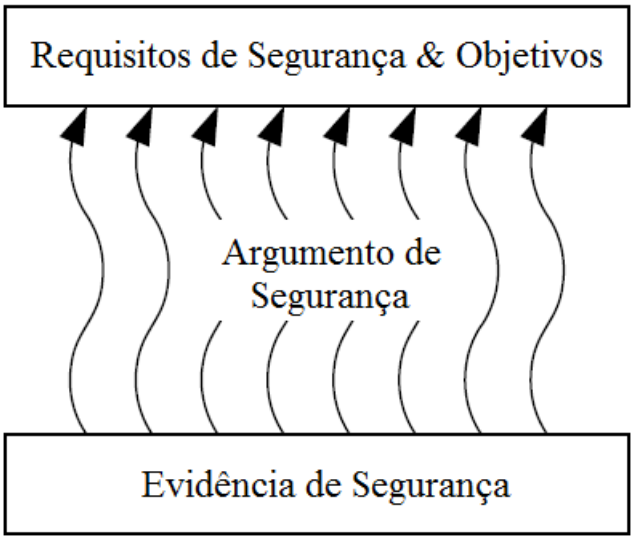


Figura 2.5: Argumento sobre segurança.

A notação estruturada por metas é um padrão utilizado para representar casos de garantia graficamente. Os principais elementos disponíveis nesse padrão são ilustrados na Figura 2.6. O padrão GSN é composto por elementos para representar metas (retângulo), soluções (círculo), estratégias (paralelogramo), elementos não definidos (losango), justificativas (elipse), e contextos (retângulo com bordas arredondadas). Os elementos *Resolvido por* e *No contexto de* são utilizados para conectar metas a soluções e a associação com contextos, respectivamente.

Além disso, argumentos podem ser representados como módulos GSN por meio de uma extensão da GSN denominada GSN modular. Um módulo de argumento GSN pode

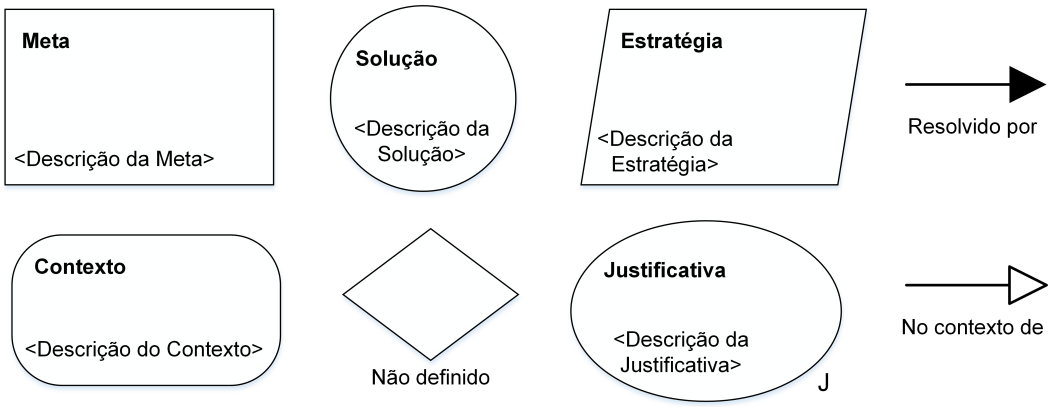


Figura 2.6: Principais elementos da GSN.

ser referenciado a partir de outros argumentos propiciando benefícios em reutilização de argumentos e facilitando a manutenção em caso de alterações de um módulo em um determinado contexto (modular e composicional). Na Figura 2.7 são apresentados exemplos de elementos utilizados em um GSN modular. O primeiro elemento é uma meta definida em um módulo que pode ser reutilizada em vários módulos diferentes. O segundo elemento é uma referência para um módulo específico. Uma referência é utilizada para conectar o módulo em que a mesma foi declarada, com um argumento localizado em outro módulo do caso de garantia [27]. O documento no qual o padrão GSN é descrito [60], é recomendável para leitores que não possuem conhecimentos básicos sobre os elementos da GSN e exemplos de utilização.

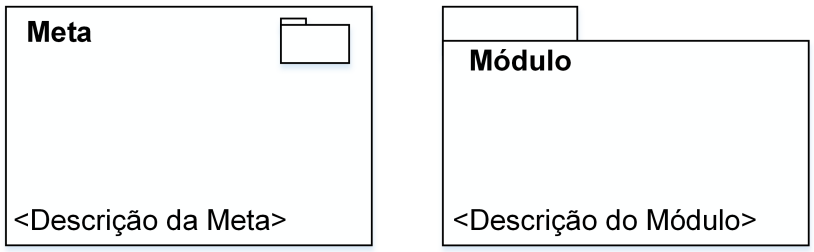


Figura 2.7: Exemplos de elementos da GSN Modular.

2.3 Transformada de Fourier

Jean-Baptiste Joseph Fourier foi o primeiro pesquisador a conduzir um estudo sistemático sobre séries para representar funções periódicas em termos de senos e cossenos, denominada série de Fourier. Este tipo de série é útil para representar sinais periódicos. Quando é necessário representar sinais periódicos, a transformada de Fourier é útil para estudá-los e analisá-los no domínio da frequência. O par da transformada de Fourier com a função $X(\omega)$ são associadas com as Equações 2.1 e 2.2 [59]. A Equação 2.1 é definida como a transformada de Fourier e a Equação 2.2 é a transformada inversa.

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{2.1}$$

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega \tag{2.2}$$

A transformada de Fourier mais utilizada no processamento de sinal digital é a transformada de Fourier discreta (*Discrete Fourier Transform - DFT*). Algoritmos projetados para calcular a transformada rápida de Fourier (*Fast Fourier Transform - FFT*) são métodos eficientes para implementar a DFT. A DFT é definida como:

$$S(k) = \sum_{n=0}^{N-1} s(n)W_N^{kn} \tag{2.3}$$

onde $k = 0, 1, ..., N-1$, N coeficientes da DFT; $s(n), n = 0, 1, ..., N-1$ é uma sequência amostrada de maneira uniforme; T é o intervalo de amostragem; $W_N = e^{-j2\pi/N}$ é a N-ésima raiz da unidade; W_N^{kn} é definido como $e^{[(\frac{-j2\pi}{N})kn]}$; e $S(k)$, $k = 0, 1, ..., N-1$ é o k-ésimo coeficiente da DFT $j = \sqrt{-1}$ [67].

A transformada de Fourier discreta inversa (*Inverse Discrete Fourier Transform - IDFT*) é definida como:

$$s(n) = \frac{1}{N}\sum_{n=0}^{N-1} S(k)W_N^{-kn} \tag{2.4}$$

onde $n = 0, 1, ..., N-1$, N amostras de dados; $(W_N^{kn})^*$ é definido como $W_N^{-kn} = e^{[(\frac{j2\pi}{N})kn]}$; e o sobrescrito * indica operação de conjugado complexo.

A DFT pode ser computada em um tempo de execução no pior caso em $O(N \lg_2 N)$ operações por meio de um algoritmo FFT, enquanto que o cálculo da DFT de forma direta requer $O(n^2)$ operações de pontos flutuantes (redução significante de complexidade) [64]. Algoritmos FFT se tornaram amplamente conhecidos na metade da década de 1960 por meio do trabalho de Cooley e Tukey [14]. Posteriormente, variações do algoritmo FFT Cooley-Tukey (decimação em tempo) foram propostas, tais como os algoritmos FFT Sand-Tukey (decimação em frequência).

Implementações de algoritmos FFT e a transformada rápida de Fourier inversa (*Inverse Fast Fourier Transform - IFFT*) estão disponíveis em ferramentas comerciais amplamente utilizadas, como, por exemplo, o *software* Matlab[3]. As funções denominadas `fft` e `ifft`, utilizadas neste trabalho, são disponibilizadas no Matlab para calcular FFT e IFFT para vetores unidimensionais, respectivamente.

2.4 Sumário do Capítulo

Neste capítulo foram apresentados conceitos relacionados com a linguagem de modelagem redes de Petri coloridas (incluindo definição formal, espaços de estado, e verificação de modelos ASK-CTL), caso de garantia, a notação estruturada por metas, e a transformada de Fourier. CPN foi a linguagem de especificação formal utilizada durante a definição do método e realização do estudo de caso sobre sistema de aquisição de sinais biomédicos desenvolvido como cenário de uso e avaliação experimental do método proposto. A técnica de casos de garantia foi utilizada para representar argumentos e evidências no método juntamente com GSN. A transformada de Fourier foi a técnica utilizada durante a especificação e validação dos modelos de sistemas de aquisição de sinais biomédicos.

Apesar de ser um conceito amplamente conhecido na área de engenharia, a transformada de Fourier é apresentada neste documento para contextualizar seu uso durante o estudo de caso realizado[4]. Os algoritmos FFT e IFFT foram utilizado na definição do modelo de um sistema de Eletrocardiografia (ECG), enquanto que somente o algoritmo FTT foi utilizado na análise dos resultados de simulações do modelo no domínio da frequência (validação).

[3] https://www.mathworks.com/help/matlab/math/fourier-transforms.html

[4] principalmente para leitores de outras áreas de atuação

Portanto, é importante que os leitores deste documentos estejam familiarizados com os conceitos fundamentais associados com esta técnica.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados trabalhos relacionados ao tema da tese de doutorado apresentada neste documento. Eventos adversos e a quantidade elevada de notificações de defeitos registrados por agências reguladoras são os principais motivos para a proposta de pesquisas relacionadas ao desenvolvimento e certificação de sistemas embarcados críticos de segurança. O foco em muitas dessas pesquisas é o aumento da confiança no funcionamento de sistemas.

Por exemplo, no trabalho proposto por Stensrud et al. [87] é descrita uma abordagem para transformar padrões prescritivos de *software* em casos de segurança (*safety cases*) baseados em metas. É proposto que os elementos de um caso de segurança específico sejam disponibilizados em um formato padrão para simplificar a reutilização de documentos desenvolvidos. Como estudo de caso da abordagem proposta, requisitos da norma IEC 61508 foram convertidos em um conjunto de padrões de casos de garantia usando a notação estruturada por metas (*Goal Structuring Notation - GSN*).

No trabalho descrito por Dechev e Stroustrup [18], é apresentado um arcabouço para a certificação de sistemas críticos orientado a produtos e baseado em modelos. O arcabouço é definido considerando o conceito de melhoria de código fonte e análise por meio de ferramentas e técnicas avançadas de programação. Os autores apresentam um estudo de caso sobre o processo de desenvolvimento e verificação baseado em modelos de um componente crítico de um sistema de dados de missão de laboratório de propulsão a jato.

Existem trabalhos nos quais autores propõem a modelagem e verificação do comportamento de sistemas de *software* usando métodos formais. Kim et al. [41] apresentam

uma abordagem formal para a especificação de modelos de *software* de aplicação e sistema operacional de tempo real (*Real-Time Operating System - RTOS*). A álgebra de recursos de comunicação compartilhados (*Algebra of Communicating Shared Resources - ACSR*) e diagramas de estados orientados a recursos e temporizados (*Timed and Resource-oriented Statecharts - TRoS*) são usados para modelar o comportamento do *software* de plataformas e de aplicação, respectivamente.

Por outro lado, Wu e Schnieder [97] apresentam uma abordagem para gerar modelos de redes de Petri coloridas (*Coloured Petri Nets - CPN*) hierárquicos baseados em diagramas de sequência, e aplicam analises de espaço de estado e a verificação de modelos (*model checking*) em propriedades padrões e específicas de sistema definidas com lógica temporal modal ASK-CTL. Um estudo de caso foi realizado para projetar um subsistema de bordo de um sistema de controle de trem para avaliar a abordagem proposta.

Arney et al. [7] desenvolveram um modelo de referência para uma bomba de infusão genérica de analgésico controlada por pacientes. No modelo de referência, provido pela Administração de Alimentos e Drogas (*Food and Drug Administration - FDA*), cuidadores, sistema de bombeamento, conjunto de infusão, e o paciente são representados. Máquinas de estado finitas estendidas (*Extended State Machines - ESM*) foram usadas para especificar o modelo de referência. As máquinas de estado foram traduzidas em modelos UPPAAL[1] para possibilitar a verificação de propriedades associadas com análises de risco do sistema em desenvolvimento.

A partir do modelo de referência proposto por Arney et al. [7], Kim et al. [40] conduziram uma engenharia dirigida a modelos para implementar um *software* de bomba de infusão genérica de analgésico controlada por pacientes. O modelo de referência foi traduzido em uma rede de autômatos e código independente de plataforma foi gerado usando as ferramentas UPPAAL e TIMES, respectivamente.

Barbosa et al. [9] utilizaram a linguagem de modelagem formal Petri nets como um arcabouço genérico para auxiliar em decisões arquiteturais durante o desenvolvimento e certificação de sistemas médicos embarcados. Um estudo de caso sobre uma bomba de infusão genérica foi realizado para ilustrar como é possível satisfazer requisitos de rastreabilidade (e.g., da arquitetura até requisitos de segurança).

[1] Mais informações disponíveis em http://www.uppaal.org/

Em outro trabalho com foco na área médica, Silva et al. [75] descrevem uma abordagem baseada em modelos para validar sistemas médicos embarcados usando o paradigma de projeto orientado a atores (*Actor-Oriented Design - AOD*). A abordagem é avaliada por meio de um estudo de caso de uma unidade de cuidado intensivo e um cenário clínico de controle de glicose sanguínea e bomba de infusão.

Jiang et al. [38] apresentam um estudo de caso sobre um marcapasso cardíaco para avaliar uma metodologia composta por especificação formal, verificação, e validação de modelos. Os autores representam o marcapasso como modelos de autômatos temporizados com a ferramenta UPPAAL. Fabricantes podem usar a especificação formal, verificação e validação para projetar e certificar sistemas médicos ao aplicar a metodologia.

Em contrapartida, Han et al. [28] descrevem um modelo de espaço de estado de Eletromiografia (EMG). O modelo é usado para realizar estimativas de movimentos contínuos de partes do corpo humano. Estimativas são conduzidas baseadas em sinais EMG (sistema de aquisição de sinal biomédico). Os autores realizaram um estudo de caso sobre a articulação do cotovelo humano para avaliar a abordagem proposta.

Li et al. [47] apresentam um estudo de caso sobre traqueostomia com laser para avaliar uma abordagem de verificação de modelos de sistemas híbridos para dispositivos médicos *Plug-and-Play* (*Medical Devices Plug-and-Play - MDPnP*). Modelos formais foram especificados como autômatos híbridos. Dispositivos, como, por exemplo, sensor O2, bisturi a laser, sensor SpO2, e ventilador, compõem a aplicação MDPnP modelada.

Nas próximas seções, alguns trabalhos relacionados diretamente com esta tese são discutidos em mais detalhes juntamente com uma síntese comparativa com esta pesquisa. No final deste capítulo, considerações gerais sobre pontos positivos e negativos nos trabalhos relacionados apresentados são descritas para contextualizar sua relação com esta pesquisa.

3.1 Especificação Formal Confiável para Certificação de Software

No trabalho proposto por Méry e Singh [54; 53], é apresentada uma metodologia para a certificação de *software* em sistemas complexos utilizando métodos formais. O objetivo é disponibilizar um modelo de certificação que possui foco em eficácia (atributo

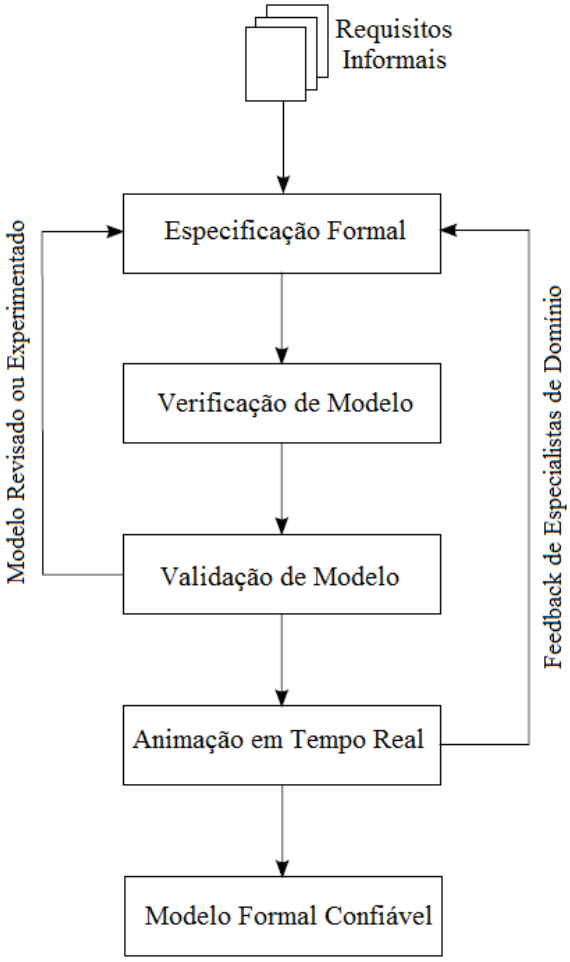


Figura 3.1: Metodologia de desenvolvimento de modelos formais confiáveis.

de qualidade do produto). A metodologia é baseada em refinamentos de abstrações do sistema em desenvolvimento. As etapas definidas na metodologia proposta são apresentadas na Figura 3.1 (adaptada de [53]). A metodologia é composta por atividades de especificação informal de requisitos utilizando documentos escritos em linguagem natural, especificação formal de requisitos baseada na especificação informal, verificação dos modelos formais baseada em refinamentos utilizando provadores de teoremas e outros métodos de prova, verificação de grau de acurácia na especificação formal em relação ao sistema real utilizando verificação de modelos, e o desenvolvimento de animações do sistema baseado em dados reais utilizando um arcabouço de animação.

Note que existe uma retroalimentação entre as atividades de validação do modelo e especificação forma a medida que o modelo é revisado ou experimentado. Isso significa que o modelo formal pode ser alterado posteriormente para sua adequação com a especificação

informal. Além disso, existe outra retroalimentação entre a atividade de animação em tempo real e a especificação formal. Neste caso, é verificado se o modelo formal está de acordo com requisitos de participantes do processo (*stakeholders*). Esse tipo de verificação é importante porque problemas em uma especificação geralmente são encontrados somente durante a execução de um sistema.

A metodologia foi proposta para resolver problemas relacionados a requisitos e métricas para garantir a segurança de sistemas. Além disso, um estudo de caso sobre um sistema de marcapasso cardíaco foi desenvolvido para avaliar a metodologia. O sistema foi especificado utilizando a linguagem de modelagem Event-B. Os modelos são integrados com animações criadas em Flash para realizar simulações. O sistema modal mais abstrato para o sistema de marcapasso de dois eletrodos é apresentado pelos autores durante a apresentações dos resultados obtidos no estudo de caso (veja Figura 3.2). Na Figura 3.2a, o passo é representado por transições *Pace ON* e *Pace OFF* para os eletrodos. Posteriormente (Figura 3.2b), o passo é refinado pelo sensoriamento (atividade do coração), o valor de tensão limiar de estimulação de um coração, e um gerador de pulso. No terceiro refinamento, estratégias operacionais diferentes sobre intervalo de histerese são adicionadas (Figura 3.2c). Por fim, no quarto refinamento, a técnica de adaptação de passo no modo operacional de bradicardia do marcapasso é introduzida (Figura 3.2d) [54].

Entretanto, os autores não consideram de maneira explícita na definição da metodologia o uso de casos de garantia para representar argumentos sobre a segurança e eficácia do sistema em desenvolvimento. Além disso, questões importantes relacionadas com esse tópico não são abordadas, como, por exemplo, a definição e rastreabilidade de requisitos durante a utilização de métodos formais e casos de garantia. A relação de um tipo de caso de garantia (caso de segurança) e os resultados obtidos com a metodologia é somente mencionado durante a sua definição. Os autores têm evoluído em sua pesquisa com estudos sobre a modelagem formal de sistemas considerando a integração de conhecimentos de domínio e modelos formais [2].

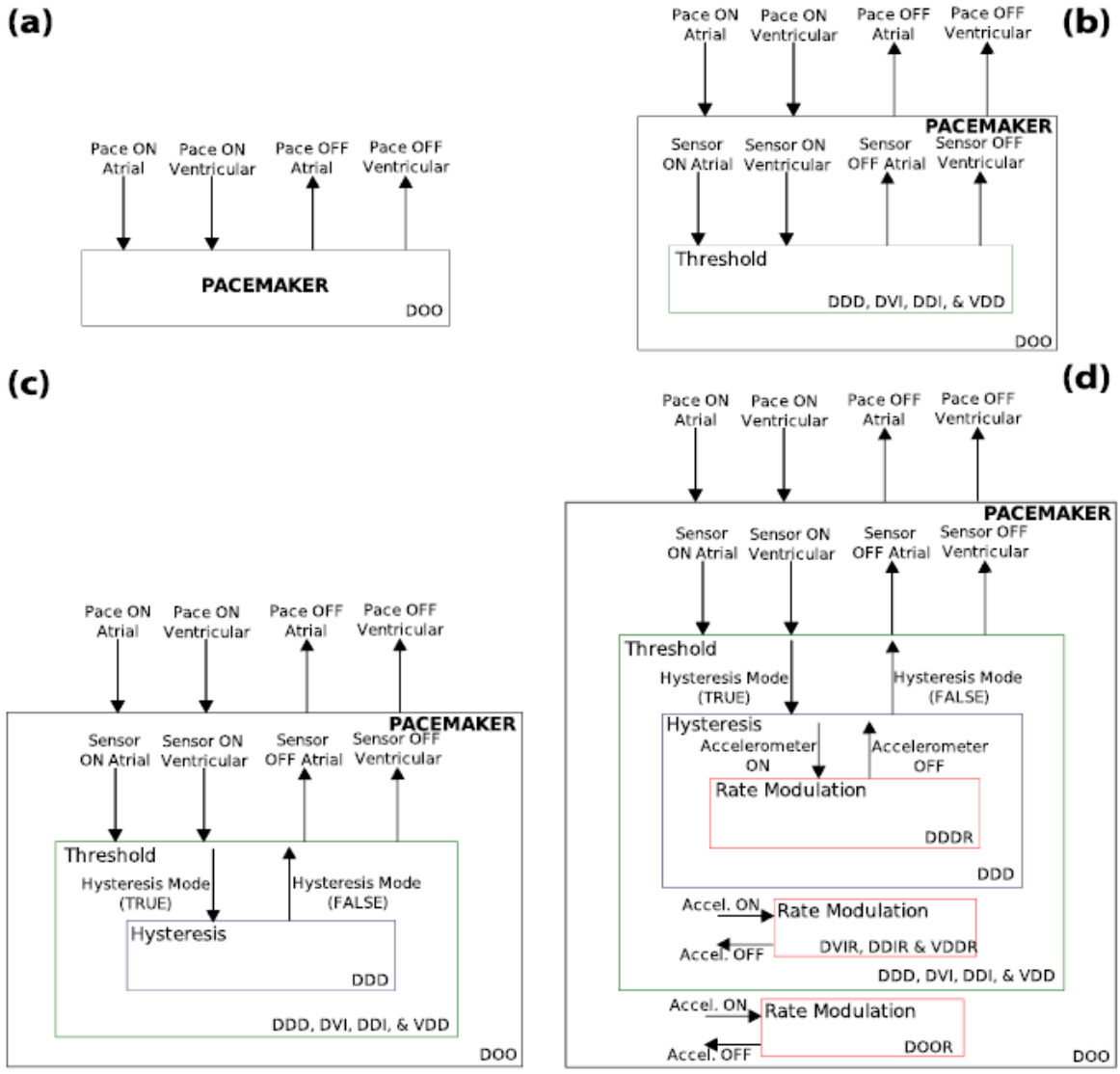


Figura 3.2: Refinamento de marcapasso de dois eletrodos usando um gráfico de refinamento.

3.2 Análise de Segurança Dirigida a Modelos de Sistemas Médicos

Por outro lado, no trabalho apresentado por Pajic et al. [61] é descrito um estudo de caso sobre um sistema de bomba de infusão (malha fechada) utilizando simulação e verificação de modelos de autômatos temporizados detalhados e abstratos. Neste caso, a ênfase está na interação do ambiente (físico) e o sistema médico em malha fechada. A arquitetura de sistema, composta por cuidador, supervisor, controlador de rede, dispositivos, e paciente, é apresentada na Figura 3.3 (adaptada de [61]). O Paciente e o cuidador são componentes

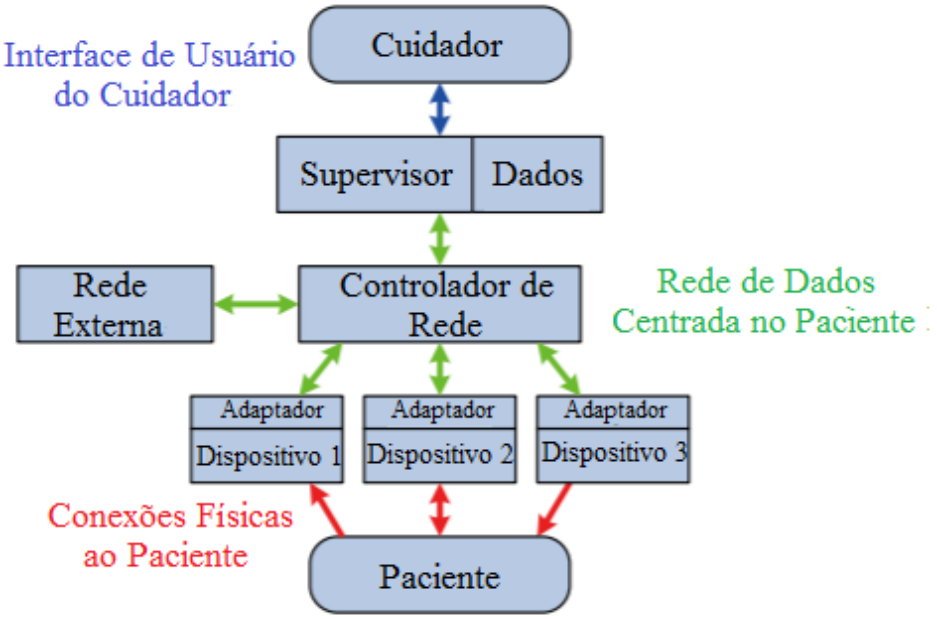


Figura 3.3: Arquitetura de dispositivo médico com controle automático.

humanos do sistema, enquanto que o supervisor é o sistema de computador utilizado para executar o algoritmo de controle. Os dispositivos são conectados ao controlador de rede. Esse controlador é responsável por rastrear dispositivos conectados e suas capacidades.

A análise baseada em simulação de modelos detalhados compostos por dinâmicas do paciente (contínua) é combinada com a verificação de modelos abstratos na abordagem utilizada durante o estudo de caso. A abordagem baseada em modelos foi utilizada para provar propriedades de segurança de dispositivos em nível de modelagem. Modelos detalhados do sistema foram desenvolvidos com a ferramenta Simulink[2]. Por outro lado, modelos abstratos foram desenvolvidos como autômatos temporizados utilizando a ferramenta UPPAAL.

Dentre os modelos de autômatos temporizados apresentados pelos autores está o modelo da bomba de infusão (veja Figura 3.4). Quando a bomba é iniciada (estado `on`), é verificado se foi realizada a sua programação (estado `programmed`). Os estados `running` e `bolusing` são utilizados para representar o estado de bomba operacional. Além disso, a bomba pode ser definida para um estado de parada (estados `Rstopped` ou `Bstopped`). Portanto, a partir de um estado operacional, a bomba pode ser alterada para um estado

[2]Mais informações disponíveis em http://www.mathworks.com/products/simulink/

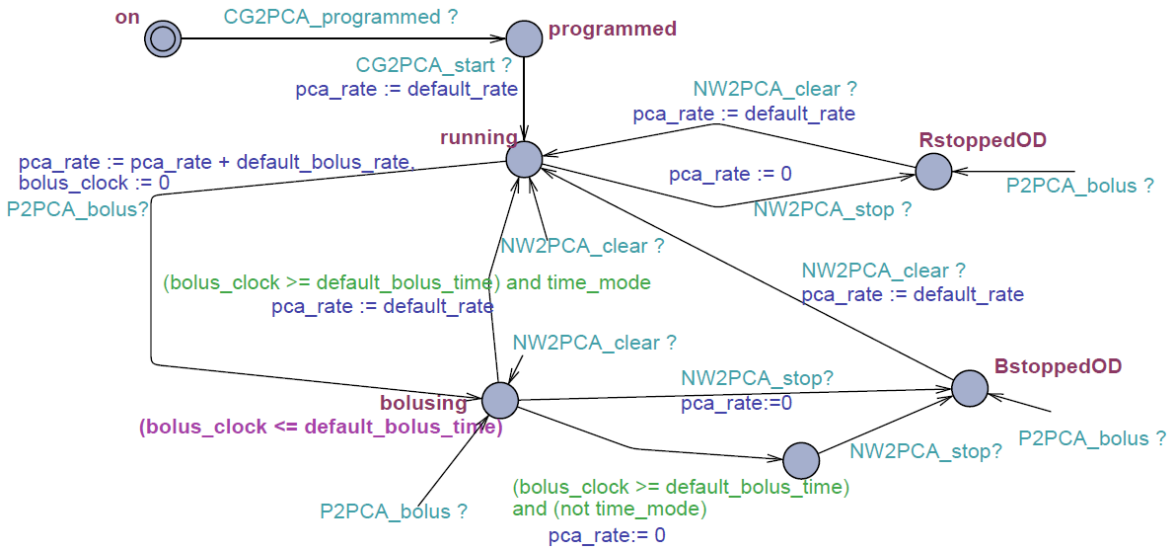


Figura 3.4: Autômato temporizado para a bomba de infusão.

de parada considerando eventos específicos [61]. Modelos de autômatos temporizados relacionados com um oxímetro de pulso, rede, e supervisor também são apresentados. O modelo de rede é utilizado para representar a comunicação entre dois autômatos (e.g., comunicação entre supervisor e bomba de infusão).

O estudo de caso desenvolvido é considerado como a etapa inicial na proposta de uma metodologia para a análise de propriedades de segurança em sistemas de dispositivos médicos em malha fechada. Os resultados obtidos com o estudo de caso podem ser utilizados para o desenvolvimento de casos de garantia de segurança para a aprovação de sistemas por agências reguladoras durante o processo de certificação.

Entretanto, problemas com o processo de certificação utilizado atualmente por agências reguladoras não são considerados por Pajic et al. [61]. Somente a modelagem, verificação e validação de sistemas, e casos de garantia de segurança são estudados. Agências reguladoras deveriam aderir a padrões baseados em metas ao utilizar esta metodologia, alterando o processo de certificação atual. Portanto, os problemas com alterações no processo de certificação podem ser gerados com esta abordagem, tal como falta de métricas. Além disso, a definição e rastreabilidade de requisitos durante o uso de métodos formais e casos de garantia também não são amplamente considerados. Atualmente, trabalhos têm sido realizados com foco em sistemas seguros e resistentes aos ataques maliciosos [31].

3.3 Desenvolvimento Baseado em Certificação

Em contrapartida, no trabalho apresentado por Steele [85] é proposto um arcabouço para o desenvolvimento baseado em certificação (*Certification-Based Development - CBD*) de sistemas críticos de segurança. Propõe-se a resolução de problemas encontrados por agências reguladoras durante o processo de certificação de sistemas críticos de segurança utilizando padrões baseados em metas. Os seguintes problemas são destacados:

1. falta de métricas para realizar avaliações de qualidade em casos de garantia;
2. variabilidade no desenvolvimento de casos de garantia.

O foco com CBD é suprir as necessidades de uma agência reguladora e um sistema específico independente de domínio de aplicação. O desenvolvimento do sistema é alinhado de acordo com as responsabilidades da agência reguladora. O processo definido com o CBD está relacionado a geração de casos de segurança para facilitar a colaboração entre desenvolvedores e agências reguladoras. Propõe-se também a avaliação do arcabouço utilizando um estudo de caso sobre um sistema de informação de diabetes. O objetivo com o CBD é possibilitar a certificação mais rápida e precisa de sistemas críticos.

O arcabouço é composto por propriedades independentes de domínio, ativos de certificação, e uma instância específica de dispositivo (veja Figura 3.5, adaptada de [85]). A proposta com um conjunto de propriedades independentes de domínio é assegurar a documentação de propriedades importantes frequentemente necessárias por domínios. Requisitos e propriedades de segurança são definidos como ativos de certificação. Por fim, uma instância específica de dispositivo de um processo de desenvolvimento deve ser associada com cada dispositivo de interesse.

Além disso, uma ferramenta para criação de casos de garantia foi desenvolvida como um dos resultados obtidos em CBD [86]. Pode-se criar, inspecionar, validar, manter casos de garantia representados com GSN. Dentre as funcionalidades do conjunto de ferramentas disponibilizado, é possível definir, alterar, e consultar nós na estrutura gráfica de um documento GSN (veja Figura 3.6). Note que a ferramenta é uma aplicação para a plataforma Microsoft Windows criada com o arcabouço .NET e que as edições e apresentações gráficas são realizadas com o Microsoft Visio. Uma ferramenta para a edição de casos de garantia

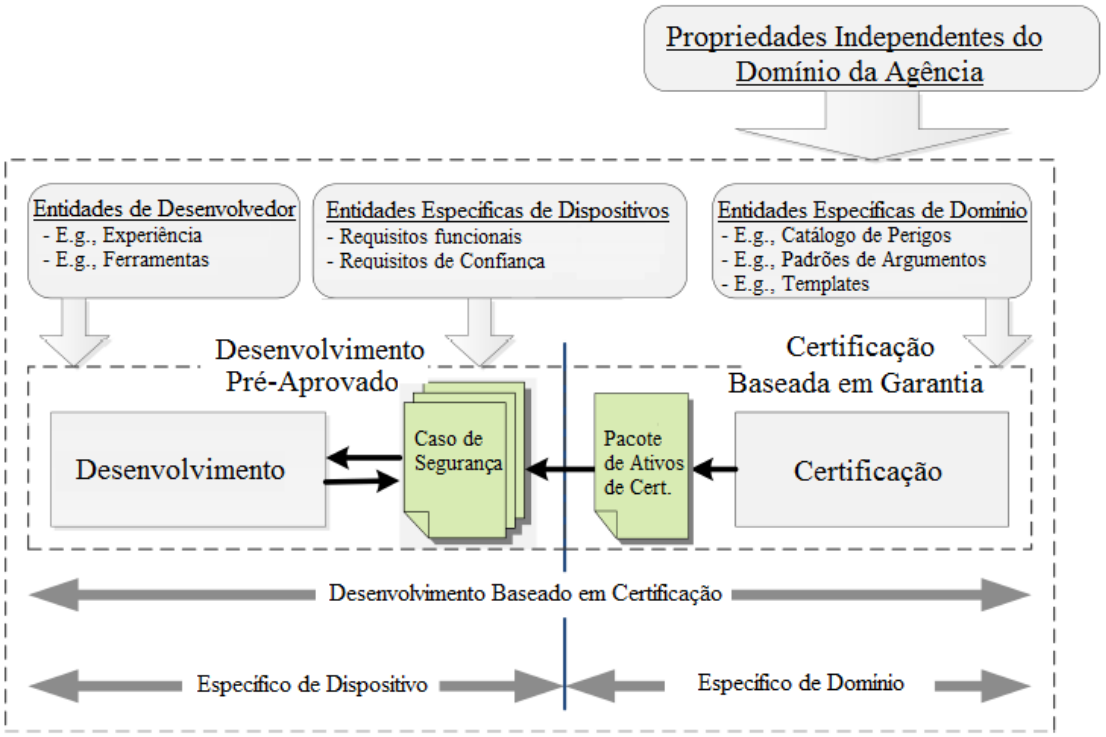


Figura 3.5: Desenvolvimento baseado em certificação.

em GSN é útil, porém, a utilização de plataformas proprietárias e a falta da definição e disponibilização de uma estrutura padrão para a representação de documento dificultam o compartilhamento dos resultados obtidos. Outra ponto importante é que isso dificulta também o processamento de maneira automatizada por sistemas de informação de agências reguladoras.

Abordagens frequentemente utilizadas para aumentar a confiança no funcionamento deste tipo de sistema tais como métodos formais, e técnicas de verificação e validação não são mencionadas. Neste contexto, a garantia de maior precisão na certificação de sistemas críticos obtida com o CBD pode ser questionada caso técnicas apropriadas não sejam utilizadas. Por outro lado, problemas relacionados com padrões definidos em processos prescritivos de certificação de sistemas críticos não são considerados durante a apresentação do arcabouço. Por exemplo, a subjetividade em padrões utilizados pode resultar em interpretações incorretas de prescrições durante o desenvolvimento de sistemas críticos. Métodos formais podem ser alternativas para reduzir a subjetividade encontrada em requisitos especificados em padrões prescritivos.

O arcabouço apresentado por Steele [85] está relacionado com um trabalho de tese de

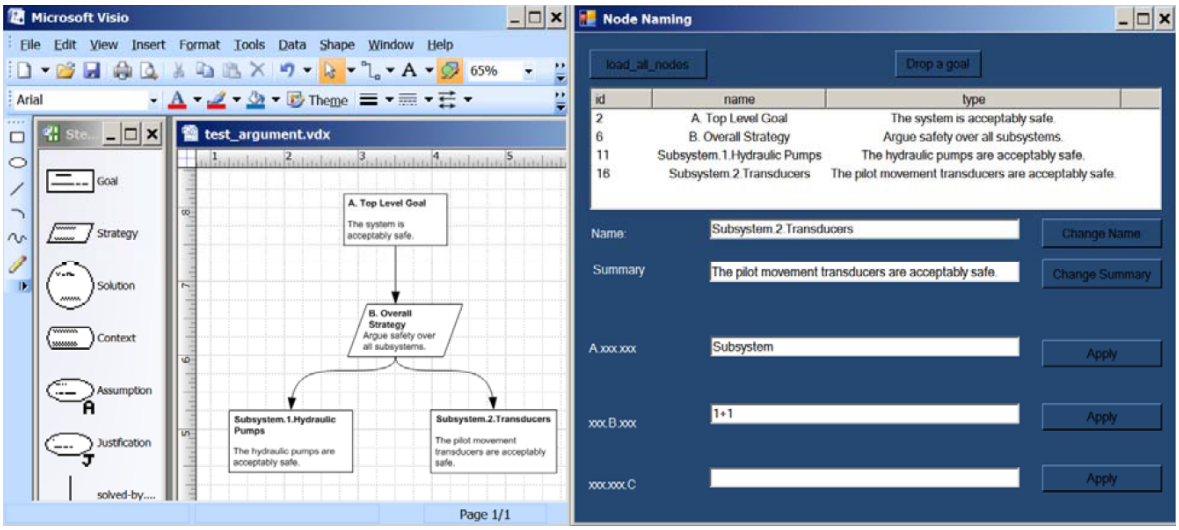


Figura 3.6: Ferramenta para a criação de casos de garantia.

doutorado realizado na universidade da Virginia. Como descrito anteriormente, o autor apresenta propostas para a avaliação do método utilizando um sistema crítico de segurança para o monitoramento contínuo de glicose e bomba de infusão. Porém, até o momento da escrita deste documento, não apresentou resultados de estudos de caso para a implementação e avaliação do método.

3.4 Sumário do Capítulo

Apesar das contribuições descritas nos trabalhos apresentados anteriormente, com o uso, por exemplo, de métodos formais e casos de garantia, limitações importantes podem ser destacadas. O principal objetivo nestes trabalhos está em alterar o foco no processo de desenvolvimento e certificação atual com o uso de métodos formais e padrões baseados em metas. Entretanto, não existe um consenso na literatura em relação a mudança no processo de certificação atual para o uso exclusivo destas abordagens. Diferente destes trabalhos, as atividades e etapas descritas no método proposto neste documento são definidas considerando os dois processos de certificação como complementares. Tal definição possibilita que fabricantes e agências reguladoras usufruam de benefícios apresentados em padrões prescritivos e baseados em metas.

Neste contexto, padrões prescritivos são documentos bem definidos, e sua utilização torna mais fácil o processo de avaliação de requisitos regulatórios por agências reguladoras.

Por exemplo, atividades descritas no gerenciamento de risco de dispositivos médicos baseado no padrão ISO 14971 são eficientes durante a análise, avaliação, e controle de riscos [25]. Isso ajuda a prevenir defeitos conhecidos. Por outro lado, padrões baseados em metas estão relacionados com sistemas embarcados críticos de segurança em desenvolvimento para tornar mais explícitas as avaliações de requisitos de segurança e eficácia utilizando argumentos e evidências bem definidas. Portanto, os pontos positivos de cada uma das abordagens são importantes durante o processo de desenvolvimento e certificação desses sistemas complexos.

Por fim, mas não menos importante, durante a utilização de padrões baseados em metas, tais como casos de garantia, é possível considerar características relacionadas com processos de desenvolvimento de *software*. Casos de garantia podem ser reaproveitados durante todo o ciclo de vida de desenvolvimento de sistemas críticos. Por exemplo, componentes de casos de garantia representados com a notação estruturada por metas podem ser associados com requisitos funcionais e não funcionais de *software*, e algoritmos podem ser aplicados para realizar a rastreabilidade entre requisitos e artefatos de projeto, e para avaliar a conformidade com requisitos regulatórios definidos por agências de certificação. Durante a especificação do método proposto nesta tese, além de considerar as contribuições apresentadas em trabalhos relacionados, são abordadas a definição, rastreabilidade e avaliação de requisitos representados por meio de casos de garantia (algumas das principais contribuições apresentadas).

Capítulo 4

Método

Neste capítulo é apresentado um método para o desenvolvimento e certificação de *software* de sistemas embarcados críticos de segurança. Dentre exemplos destes sistemas pode-se citar sistemas médicos, aviônicos, automotivos e aeroespaciais. O objetivo com a definição e aplicação do método é aumentar a confiança no funcionamento de sistemas embarcados. Falhas em sistemas podem resultar em erros, e, por exemplo, gerar desastres naturais catastróficos e danos à integridade física de seres humanos. O método é especificado com base na linguagem de modelagem formal redes de Petri coloridas (*Coloured Petri Nets - CPN*) e casos de garantia (*assurance cases*) representados com a notação estruturada por metas (*Goal Structuring Notation - GSN*).

4.1 Visão Geral

A utilização dos conceitos de casos de garantia e da notação estruturada por metas possibilita a integração de artefatos gerados a partir da modelagem formal, verificação, validação, e simulação de requisitos de padrões prescritivos e de sistemas embarcados críticos de segurança. Além disso, casos de garantia representados com a GSN modular são utilizados durante a avaliação de sistemas embarcados por agências reguladoras. A visão geral do método é ilustrada na Figura 4.1. Este método é composto por quatro atividades principais: `Especificação Informal e Semiformal`, `Requisitos de Padrões`, `Requisitos do Produto`, e `Casos de Garantia`. Existem também três atividades de suporte denominadas `Argumentos`

de Processo, Argumentos de Produto e Casos de Teste. O foco com a proposta deste método está em atividades desenvolvidas por fabricantes de sistemas embarcados críticos de segurança e no compartilhamento de resultados obtidos com agências reguladoras.

A modelagem, verificação, validação, e simulação de padrões prescritivos e de requisitos do sistema são realizadas utilizando uma especificação informal e semiformal do sistema. As especificações informais e semiformais podem ser desenvolvidas com linguagem natural e representações gráficas, respectivamente. Fabricantes devem especificar os requisitos contidos em padrões prescritivos utilizando a linguagem de modelagem formal CPN, e implementar esses requisitos com base nos modelos formais criados durante o processo de desenvolvimento do produto. Técnicas, como, por exemplo, a análise de criticidade e modo de efeito de falhas (*Failure, Mode, Effects and Criticality Analysis - FMECA*) e a verificação de modelos (*Model Checking*), são utilizadas para gerar evidências que apóiam argumentações sobre conformidade com requisitos de segurança e eficácia do sistema. Requisitos contidos em padrões prescritivos são representados utilizando casos de garantia e a GSN modular para gerar argumentações sobre a conformidade de fabricantes com as especificações definidas nos padrões por meio da atividade de suporte Argumentos de Processo.

Além disso, requisitos funcionais e não funcionais devem ser modelados, verificados, e validados na atividade Requisitos do Produto. A linguagem de modelagem formal CPN é utilizada para a especificação de requisitos do sistema. Requisitos funcionais e não funcionais também são representados utilizando casos de garantia e a GSN modular para gerar argumentações relacionadas com propriedades de segurança e eficácia do sistema por meio da atividade de suporte Argumentos de Produto. As atividades Requisitos de Padrões e Requisitos do Produto podem ser implementadas em paralelo por equipes de profissionais especializados. Os resultados obtidos por fabricantes de sistemas embarcados críticos de segurança são estruturados como argumentos e evidências em casos de garantia representados com GSN durante a atividade Casos de Garantia.

Os casos de garantia gerados por fabricantes devem ser representados utilizando a linguagem de marcação independente de plataforma denominada linguagem de marcação extensível (*Extensible Markup Language - XML*) durante a atividade Casos de

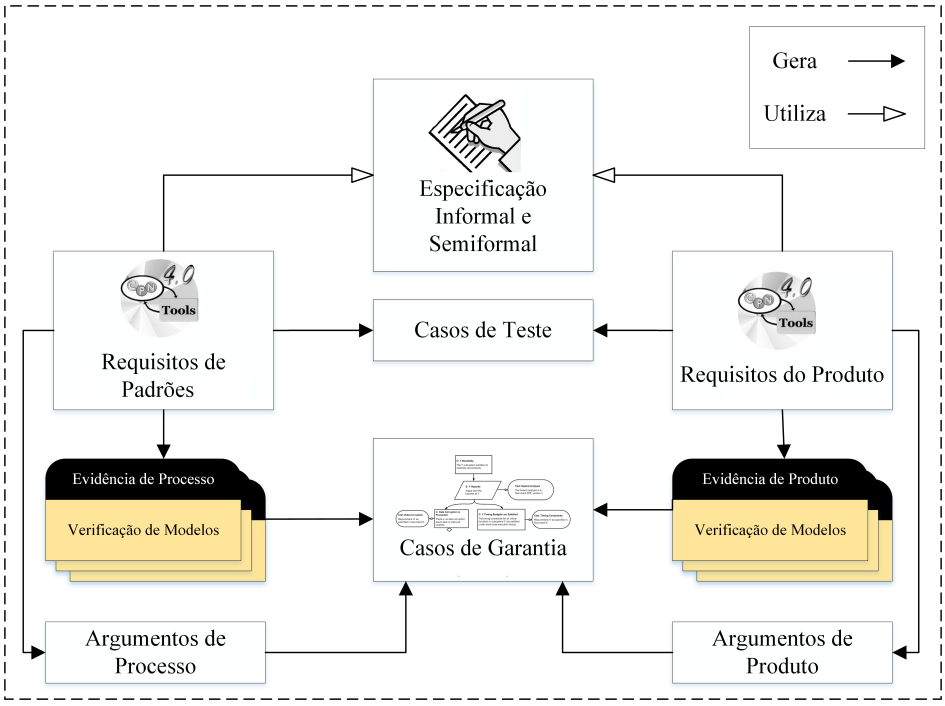


Figura 4.1: Visão geral do método de desenvolvimento e certificação de sistemas embarcados críticos de segurança.

`Garantia`. Com isso, o caso de garantia pode ser compartilhado entre sistemas de informação diferentes. Portanto, sistemas mantidos por fabricantes podem ser utilizados para enviar e receber casos de garantia para/de sistemas mantidos por agências reguladoras de sistemas embarcados críticos de segurança.

Sistemas de informação mantidos por agências reguladoras devem ser capazes de interpretar casos de garantia representados com XML durante a atividade `Casos de Garantia` definida no método. Com isso, é possível que agências reguladoras analisem argumentos e evidências apresentadas por fabricantes no desenvolvimento de sistemas embarcados críticos de segurança, e realizem solicitações de novas evidências quando necessário. Isso pode facilitar o processo de avaliação realizado por agências reguladoras. Por exemplo, os casos de garantia podem ser compartilhados até a aceitação dos argumentos e evidências fornecidas pelo fabricante de sistemas. Os resultados de avaliações podem ser inseridos no mesmo arquivo no formato XML utilizado para representar o caso de garantia.

4.2 Atividades

Como descrito anteriormente, o método foi especificado como um conjunto de atividades que devem ser realizadas durante o desenvolvimento e certificação de *software* de sistemas embarcados críticos de segurança. Mais especificamente, fabricantes devem obter conformidade com as atividades de `Especificação Informal e Semiformal`, `Requisitos de Padrões`, `Requisitos do Produto`, e `Casos de Garantia`.

É importante destacar que existe um relacionamento entre as atividades definidas no método. Artefatos de projeto gerados a partir da implementação da atividade `Especificação Informal e Semiformal` são utilizados pelas atividades `Requisitos de Padrões` e `Requisitos do Produto`. Por outro lado, nas atividades `Requisitos de Padrões` e `Requisitos do Produto`, artefatos de projeto são gerados como entrada para a atividade principal `Casos de Garantia` (evidências) e atividades de suporte `Argumentos de Processo`, `Argumentos de Produto`, e `Casos de Teste`.

4.2.1 Especificação Informal e Semiformal

A especificação de requisitos de *software* utilizando linguagem natural e representações gráficas está incluída nesta atividade. Fabricantes de sistemas embarcados críticos de segurança devem manter um documento descrevendo características do sistema tais como escopo, definição, e requisitos funcionais e não funcionais. Requisitos podem ser especificados por meio de linguagem natural, linguagem natural estruturada, e notações gráficas. Linguagem natural é a maneira mais comum e amplamente utilizada para especificar requisitos. Um exemplo de especificação informal de um sistema de marcapasso em linguagem natural é disponibilizado por Boston Scientific [71]. A linguagem natural estruturada, por sua vez, é composta por modelos para padronizar a estrutura de documentos de especificação. Por fim, um exemplo de notação gráfica amplamente aderida é a linguagem de modelagem unificada (*Unified Modeling Language - UML*), que é composta por diagramas tais como, casos de uso, classe, sequência, componentes e atividades.

O uso de representações gráficas semiformais é útil para especificar requisitos

de *software* e requisitos relacionados com *hardware* (quando necessário) de sistemas embarcados críticos de segurança. Por exemplo, modelos formais de *hardware* de sistemas podem ser gerados baseados em equações utilizadas para representar saídas definidas em diagramas de circuitos do sistema. Por outro lado, modelos formais de *software* podem ser gerados a partir de representações gráficas especificadas utilizando a linguagem de modelagem unificada. Ambas as especificações informais e semiformais de componentes de *hardware* e *software* são utilizadas dependendo do contexto relacionado com o sistema embarcado crítico de segurança em desenvolvimento. Em alguns casos, pode ser somente necessário utilizar especificações de requisitos de *software* em diferentes perspectivas com diagramas UML, enquanto que, em outros casos, características específicas definidas na construção do circuito de um sistema embarcado devem ser analisadas.

4.2.2 Requisitos de Padrões

Padrões de *software* possuem um papel muito importante no gerenciamento de qualidade de *software*. Uma parte importante da garantia de qualidade é a definição ou seleção de padrões que devem ser aplicados ao processo de desenvolvimento de *software* ou produto de *software*. Como parte desse processo, ferramentas e métodos para suportar o uso desses padrões podem ser selecionados. Padrões de processo são utilizados para definir os processos que devem ser seguidos durante o desenvolvimento de *software*. Organizações internacionais, como, por exemplo, o departamento de defesa dos Estados Unidos (*Department of Defense - DoD*), instituto nacional americano de padrões (*American National Standards Institute - ANSI*), organização internacional para padronização (*International Organization for Standardization - ISO*), comissão eletrotécnica internacional (*International Electrotechnical Commission - IEC*) e o instituto de engenheiros eletricistas e eletrônicos (*Institute of Electrical and Electronics Engineers - IEEE*), suportam a produção de padrões prescritivos.

Neste contexto, a natureza crítica de segurança de determinados sistemas embarcados torna necessário que fabricantes desses sistemas utilizem padrões prescritivos associados com a garantia de segurança e eficácia (atributos de qualidade). Padrões prescritivos são definidos por agência reguladoras como requisitos durante um processo prescritivo de certificação. Por exemplo, o padrão ISO 14971 é definido por agências reguladoras (e.g.,

a Agência Nacional de Vigilância Sanitária - ANVISA) como a diretriz utilizada durante o processo de gerenciamento de risco de dispositivos médicos. Outros padrões prescritivos, tal como o padrão IEC 62304 [24] para o desenvolvimento de *software* de dispositivos médicos também podem ser considerados nesta atividade do método. Entretanto, como contextualizado no Capítulo 1, padrões prescritivos podem ser considerados como documentos subjetivos porque são especificados por meio de uma linguagem natural. Isso implica que a implementação correta de requisitos depende do modo como profissionais responsáveis pelo uso de padrões os interpretam.

Portanto, um conjunto de tarefas é definido na atividade `Requisitos de Padrões` para reduzir a subjetividade encontrada em padrões prescritivos. Esta atividade é representada no diagrama de blocos ilustrado na Figura 4.2. Os requisitos definidos em um padrão prescritivo específico são representados na primeira tarefa descrita no diagrama (`Requisitos de Padrão Prescritivo`). Neste trabalho é proposto que estes requisitos sejam especificados durante a segunda tarefa com a linguagem de especificação formal CPN (`Especificação Formal`). Na terceira tarefa (`Verificação e Validação de Modelos`), os modelos formais desenvolvidos durante a especificação formal devem ser verificados e validados utilizando a técnica de verificação de modelos e simulações. Fórmulas especificadas com a lógica temporal ASK-CTL relacionadas aos requisitos do padrão devem ser utilizadas durante a verificação de modelos com o *software* CPN/Tools.

Por outro lado, a simulação de modelos deve ser realizada com o auxílio de especialistas no desenvolvimento de sistemas para identificar problemas no processo modelado. Caso problemas sejam identificados em modelos do processo, a especificação formal do padrão prescritivo deve ser corrigida. O padrão prescritivo somente pode ser utilizado após a realização das atividades de verificação e validação do modelo formal. Representações gráficas externas podem também ser integradas com simulações de modelos para apresentar os requisitos de uma maneira mais realística. Note que as três primeiras tarefas foram especificadas considerando a metodologia para a certificação de *software* em sistemas complexos descrita por Méry e Singh [54; 53].

É proposto neste método que fabricantes implementem os requisitos definidos no padrão prescritivo durante o desenvolvimento de sistemas embarcados críticos de segurança

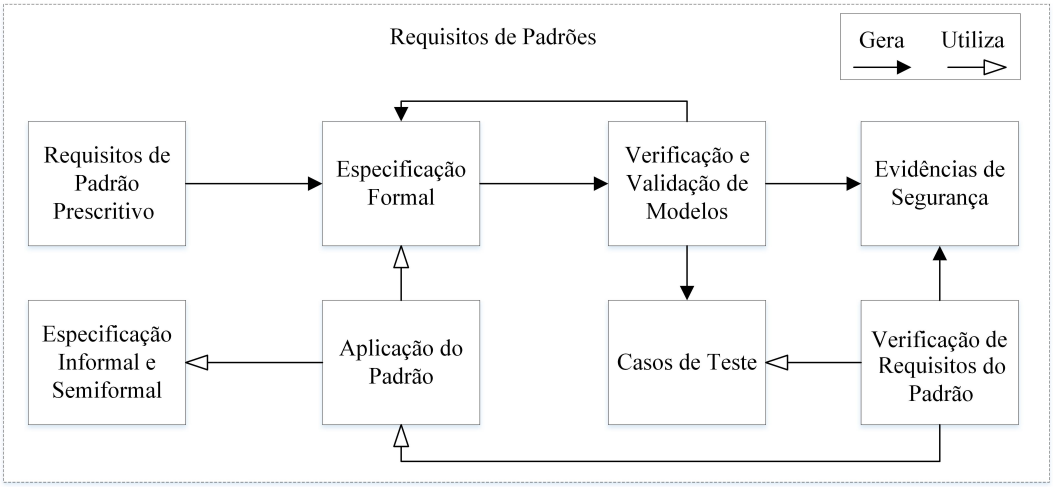


Figura 4.2: Diagrama de blocos para a atividade de requisitos de padrões.

baseados nos modelos formais e na especificação informal e semiformal do sistema (tarefa `Aplicação do Padrão`). Dado que modelos formais são disponibilizados, sequências de teste são geradas a partir do espaço de estado do modelo e definidas como casos de teste (tarefa `Casos de Teste`). Por fim, verificações de requisitos contidos no padrão são realizadas (tarefa `Verificação de Requisitos do Padrão`) com base nos casos de teste e evidências geradas para prover suporte ao processo de certificação do produto em desenvolvimento. Realizar verificações com base em casos de teste é importante porque possíveis entradas e saídas desejadas são definidas e testadas. Além disso, cenários são testados a partir de sequências de teste.

O algoritmo definido por Wu e Schnieder [96] (veja Algoritmo 1) pode ser considerado para a geração de casos de teste a partir de sequências de teste extraídas do espaço de estado do modelo formal definido com CPN (caminhos no grafo). Cada um dos caminhos está relacionado com possíveis cenários ou combinação de cenários. Neste algoritmo, todos os caminhos no grafo de espaço de estado são obtidos a partir de um estado inicial específico de maneira recursiva. O espaço de estado do modelo CPN pode ser extraído utilizando funções predefinidas no *software* CPN/Tools[1], e o grafo obtido é definido como entrada para o algoritmo. A saída do algoritmo é composta por um conjunto de caminhos iniciados a partir do estado inicial até um estado alvo (estado final). A utilização de um algoritmo desse tipo é importante porque resultados obtidos com funções predefinidas no *software* CPN/Tools

[1]Funções disponíveis na ferramenta de espaço de estado podem ser encontradas em [37]

Algoritmo 1: GERAÇÃO DE CAMINHO DE TESTE

Entrada: Espaço de estado

Saída: Caminhos iniciando do estado inicial

```
1  Caminhos (Nó Inicial, Nó Final, Caminho)
2  visitado[Nó Inicial] = Verdadeiro;
3  para cada nó no grafo faça
4      se não existe caminho entre nó inicial e nó atual
5      OU nó atual já foi visitado então
6          continue;
7          se nó atual = nó final então
8              adicione ao caminho;
9      Caminhos(nó atual, Nó final, Caminho + nó atual);
10     visitado[nó atual] = falso;
11 fim
```

estão associados somente com um dos possíveis caminhos no grafo percorrido (e.g., a função `Reachable`). Obter todos os caminhos possíveis significa gerar todos os cenários de teste possíveis em um espaço de estado do modelo especificado.

Técnicas, tal como a FMECA podem também ser utilizadas durante a implementação dos requisitos definidos em padrões prescritivos. Resultados obtidos durante a verificação e validação de modelos formais, e a implementação dos requisitos definidos no padrão modelado devem ser documentados por fabricantes. Estes documentos são utilizados como evidências relacionadas com a interpretação e uso correto do padrão prescritivo (tarefa `Evidências de Segurança`).

4.2.3 Requisitos do Produto

Além de requisitos de padrões prescritivos, deve-se considerar a especificação e análise formal de requisitos específicos de sistemas. A utilização isolada de padrões prescritivos não é suficiente para garantir a qualidade do sistema embarcado em desenvolvimento. Neste caso, um conjunto de tarefas é definido para aumentar a confiança no funcionamento correto de sistemas na atividade `Requisitos do Produto`. Essa atividade é representada com

o diagrama de blocos ilustrado na Figura 4.3. A especificação informal e semiformal de sistemas embarcados críticos de segurança desenvolvidas em linguagem natural e representações gráficas, e utilizadas por fabricantes durante a modelagem do sistema são representadas na primeira tarefa (`Especificação Informal e Semiformal`). A especificação formal de sistemas deve ser gerada utilizando a linguagem de modelagem formal CPN (tarefa `Especificação Formal`).

É necessário garantir que os modelos formais foram gerados corretamente. Neste contexto, a verificação e validação dos modelos formais devem ser realizadas para verificar a conformidade com a especificação informal e semiformal de sistemas (tarefa `Verificação e Validação de Modelos`). Além disso, erros podem ser identificados na especificação informal e semiformal. Existe uma realimentação entre a verificação e validação de modelos, e as especificações informal, semiformal, e formal, como é ilustrado na segunda e terceira tarefa. A simulação do sistema também pode ser integrada com uma representação gráfica externa para representar os resultados de maneira mais realística, e, consequentemente, simplificar a validação dos modelos especificados. Note que as três tarefas também foram definidas considerando a metodologia para a certificação de *software* em sistemas complexos descrita por Méry e Singh [54; 53]

Como descrito anteriormente, a especificação informal e semiformal de componentes de *hardware* e *software* são utilizadas para gerar a especificação formal com a linguagem de modelagem formal CPN nesta atividade do método. Além disso, caso a especificação formal não represente de maneira suficiente o comportamento do sistema, os modelos formais podem ser integrados com modelos gerados por outras ferramentas. Por exemplo, em sistemas de aquisição de sinais biomédicos (e.g., Eletrocardiografia - ECG), a dinâmica discreta de sistemas pode ser especificada em modelos formais com CPN, enquanto que filtros digitais podem ser desenvolvidos utilizando o *software* Matlab [92]. Neste caso, a especificação formal e os filtros digitais podem ser integrados para representar o comportamento real de um sistema embarcado específico. Esta integração pode ser realizada por uma comunicação TCP/IP entre a ferramenta de modelagem CPN/Tools e o Matlab. A atividade `Requisitos do Produto` possui como principal artefato de projeto um modelo composto por representações de componentes de *hardware* e *software*.

Fabricantes só podem utilizar o modelo durante outras atividades do método se o mesmo

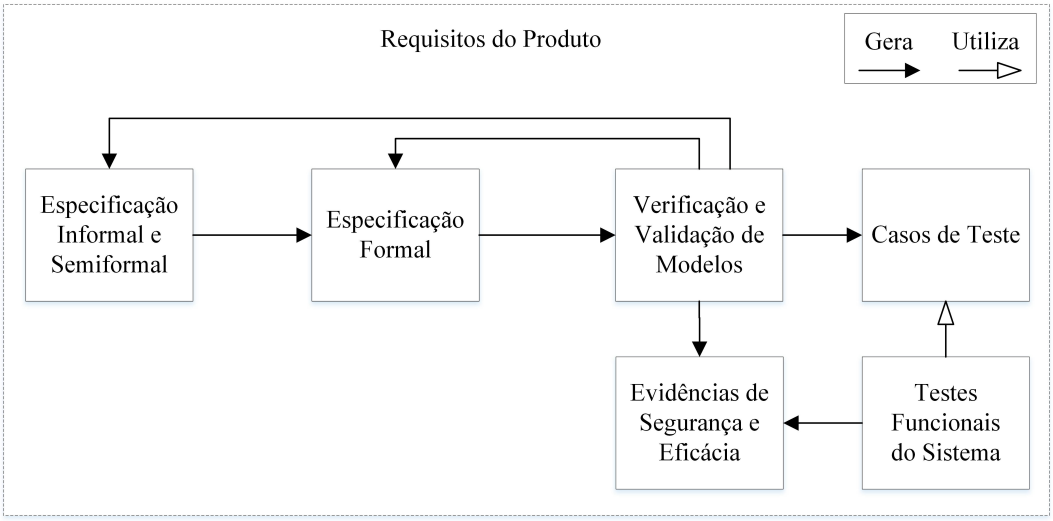


Figura 4.3: Diagrama de blocos para a atividade Requisitos do Produto.

estiver verificado e validado. Caso contrário, problemas identificados na especificação informal, semiformal, e formal devem ser resolvidos. As verificações no modelo formal devem ser realizadas utilizando a técnica de verificação de modelos (*model checking*) baseadas em propriedades de sistemas especificadas com a lógica temporal ASK-CTL. Por outro lado, especialistas no desenvolvimento de sistemas devem participar da atividade de validação do modelo formal utilizando simulações.

Por exemplo, em um sistema de ECG, atividades para a validação de modelos formais podem ser realizadas da seguinte maneira:

1. utilização dos mesmos dados de entrada em um *software* para simular o circuito de sistemas e em modelos formais para comparar os resultados;

2. utilização dos mesmos dados de entrada em sistemas reais e em modelos formais para comparar os resultados relacionados com a ocorrência de ruídos em sinais característicos de seres humanos.

Na primeira atividade de validação, sinais de entrada são utilizados para simular os dois eletrodos de sinal conectados no corpo de pacientes. Um componente que representa um gerador de função contido em um sistema de simulação de circuitos (e.g., *software* de simulação Proteus [48]) e dados gerados com funções do Matlab podem ser as entradas para a representação do circuito e para modelos de sistemas, respectivamente. Com isso, é possível comparar as saídas relacionadas com a amplificação e filtragem de sinais separadamente.

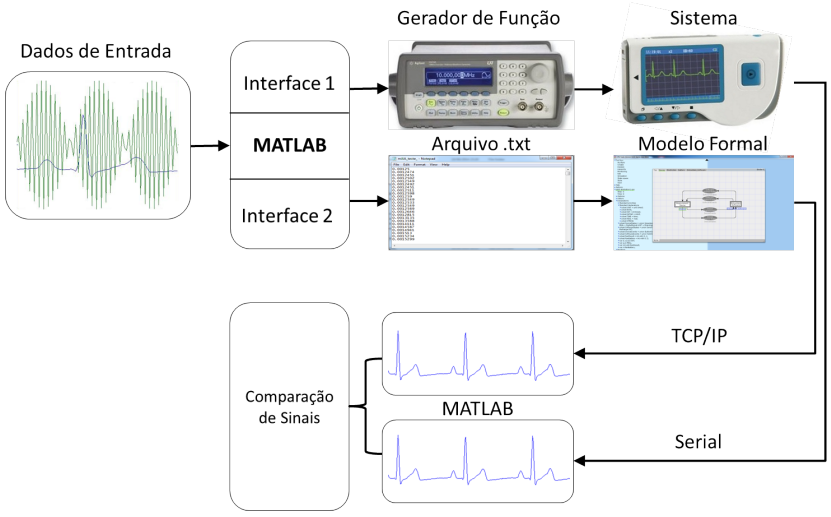


Figura 4.4: Diagrama de blocos para validação de modelos.

Sinais de entrada também são utilizados para simular dois eletrodos de sinal durante a segunda atividade de validação. Fabricantes podem utilizar uma imagem para representar um formato de onda característica. Além disso, ruídos devem ser gerados utilizando o Matlab ou isolados de registros de seres humanos disponíveis em bases de dados tais como a PHYSIONET[2].

Após gerar as entradas, fabricantes podem utilizar um simulador de ECG conectado ao sistema de ECG. Com isso, pode-se definir os mesmos sinais de entrada no modelo formal e no dispositivo real para comparar os resultados obtidos. Isso permite a validação do modelo por meio de sinais característicos com a ocorrência de ruído. Essa atividade de validação é representada no diagrama de blocos ilustrado na Figura 4.4. Note que dados de saída de ambos sistemas são comparados para avaliar a qualidade dos resultados.

Métricas para a avaliação de qualidade devem ser utilizadas durante a comparação dos sinais de saída gerados pelo modelo formal em relação as saídas desejadas (obtidas por um sistema real). Os resultados obtidos com os cálculos associados com estas métricas de desempenho devem ser o mais próximo possível de zero. Exemplos de métricas incluem a raiz quadrada do erro quadrático médio (*Root Mean Squared Error - RMSE*) [91] e o erro absoluto médio (*Mean Absolute Error - MAE*) [4].

A simulação do sistema somente pode ser realizada após as atividades de verificação e validação utilizando, por exemplo, as abordagens descritas acima. A simulação dos modelos

[2]Disponível em http://www.physionet.org/

formais deve ser conduzida com o *software* CPN/Tools. Além disso, como já descrito, é proposta a integração da simulação de modelos com uma representação gráfica externa para apresentar os resultados de uma forma mais realista. Por exemplo, uma aplicação desenvolvida com o Matlab pode ser integrada com o *software* CPN/Tools para apresentar a forma de onda obtida pelo modelo formal em tempo de simulação.

Caso já existam modelos formais definidos, verificados e validados previamente (modelos de referência), fabricantes podem reutilizá-los para reduzir problemas identificados com o uso de métodos formais, como, por exemplo, custos, tempo de desenvolvimento, e desmotivação (contextualizados no capítulo 1, Seção 1.1). Modelos de referência são definidos para representar as principais características de um sistema ou uma classe de sistemas. Fabricantes podem usar modelos de referência para reduzir o número de defeitos em sistemas e gerar evidências de segurança e eficácia, ao estendê-los em representações de um sistema específico em desenvolvimento. Neste caso, fabricantes podem seguir diretamente para as tarefas subsequentes contidas na atividade `Requisitos do Produto`. Portanto, após a criação de um modelo de referência de uma classe de sistema embarcado, não é necessário repetir as tarefas já realizadas para desenvolver um novo sistema específico. Porém, note que em determinados casos, atividades de verificação e validação adicionais podem ser necessárias para garantir que o modelo possui o comportamento esperado em um sistema embarcado específico.

Como na atividade `Requisitos de Padrões`, dado que verificações e validações foram realizadas, evidências de segurança e eficácia são geradas a partir dos modelos formais (tarefa `Evidências de Segurança e Eficácia`). Além disso, testes funcionais do sistema são conduzidos utilizando casos de teste disponíveis (tarefa `Casos de Teste`). Como descrito anteriormente, sequências de teste são geradas considerando o Algoritmo 1 por meio do espaço de estado do modelo CPN. Portanto, pode-se testar vários cenários de uso do sistema embarcado para avaliar se requisitos funcionais foram contemplados durante a sua especificação.

4.2.4 Casos de Garantia

Fabricantes de sistemas embarcados críticos de segurança devem desenvolver casos de garantia contendo argumentações sobre a segurança e eficácia do sistema durante esta

atividade. Casos de garantia devem ser representados utilizando módulos GSN relacionados aos requisitos especificados em padrões prescritivos (atividade `Requisitos de Padrões`) e requisitos funcionais e não funcionais do sistema (atividade `Requisitos do Produto`). Argumentos e evidências de processo (padrões prescritivos) e produto (sistema embarcado) são estruturados como módulos GSN. No contexto do método apresentado neste documento, argumentos e evidências de processo e produto são definidos como:

- **argumentos de processo** é um conjunto de afirmações relacionadas aos requisitos definidos em padrões prescritivos;
- **argumentos de produto** é um conjunto de afirmações relacionadas aos requisitos funcionais e não funcionais do sistema;
- **evidências de processo** é um conjunto de resultados obtidos durante a atividade `Requisitos de Padrões` utilizado para aumentar a confiança em argumentos de processo;
- **evidências de produto** é um conjunto de resultados obtidos durante a atividade `Requisitos do Produto` utilizado para aumentar a confiança em argumentos de produto.

Retomando o exemplo sobre sistemas de aquisição de sinais biomédicos descrito na seção anterior, amostras de argumentos e evidências de produto e processo no desenvolvimento de um sistema de ECG genérico são apresentados na Tabela 4.1. Cada linha na tabela está relacionada com argumentos e evidências que podem ser gerados a partir das atividades `Requisitos de Padrões` e `Requisitos do Produto` descritas nas seções anteriores.

Argumentos, evidências, e os demais componentes dos módulos GSN devem ser representados com a linguagem de marcação independente de plataforma XML por meio de um padrão para a representação e compartilhamento de documentos de casos de garantia em GSN. Isso é realizado para possibilitar o compartilhamento dos resultados obtidos por fabricantes de sistemas com agências reguladoras, a realização da atividade de

Tabela 4.1: Exemplos de argumentos e evidências.

Argumento	**Evidência**
As etapas de filtragem, amplificação e conversão são realizadas de maneira correta durante a execução do sistema de ECG (argumento de produto 1).	Comparação entre resultados obtidos com a simulação do modelo de sistemas e com um sistema real utilizando os mesmos sinais de entrada (evidência de produto 1).
Aquisição de sinal é somente realizada com valores de impedância eletrodo-pele aceitáveis (argumento de produto 2).	Aplicação da técnica de verificação de modelos (*model checking*) (evidência de produto 2).
Todos os riscos são identificados (argumento de processo 1).	Aplicação da técnica FMECA (evidência de processo 1).
Todas as medidas de controle são identificadas (argumento de processo 2).	Relatório de medidas de controle fornecido por especialistas no desenvolvimento de sistemas (evidência de processo 2).

rastreabilidade de requisitos conduzida durante o processo de engenharia de requisitos de sistemas, e a verificação automatizada de requisitos para a certificação.

A visão geral da atividade `Casos de Garantia` é descrita no diagrama de blocos ilustrado na Figura 4.5. Note que todas as tarefas associadas com esta atividade estão centralizadas em uma especificação XML (tarefa `Especificação XML`). Esta especificação deve ser realizada com base no padrão para a representação e compartilhamento de documentos de casos de garantia (*Assurance Cases Exchange Standard - ACES*) definido no método. Conceitos relacionados com a rastreabilidade de requisitos e com a notação gráfica GSN [60] são utilizados para gerar a especificação XML baseada no padrão ACES. Isso possibilita que agências reguladoras e fabricantes conduzam de forma automatizada verificações de requisitos regulatórios (tarefa `Verificação de Requisitos Regulatórios`) e a rastreabilidade de requisitos (tarefa `Rastreabilidade de Requisitos`), respectivamente.

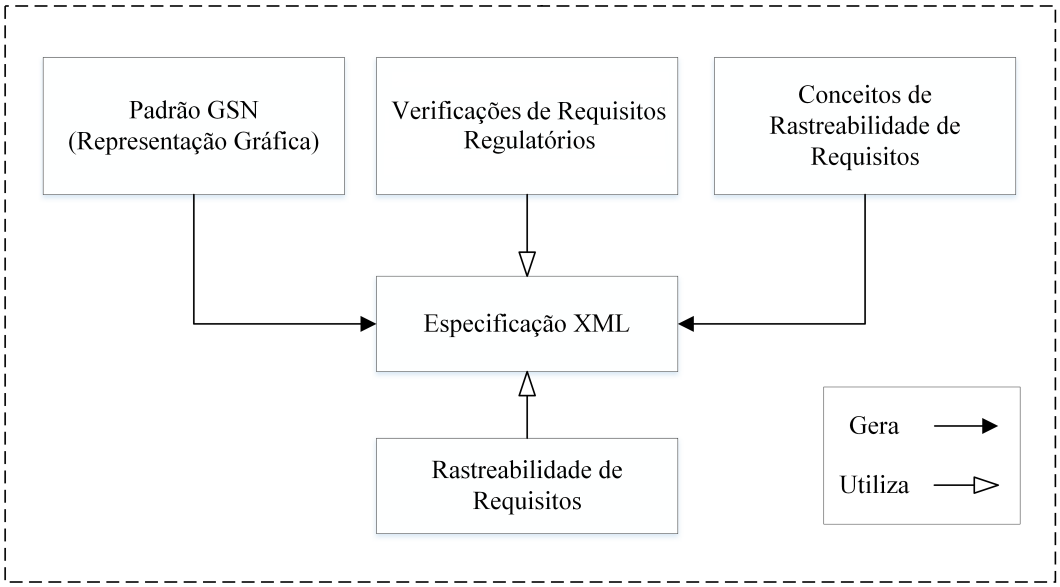


Figura 4.5: Diagrama de blocos para a atividade casos de garantia.

Padrão de Casos de Garantia ACES

Como descrito anteriormente, é proposto neste método que os resultados obtidos por fabricantes durante o desenvolvimento de sistemas embarcados críticos de segurança sejam representados em um formato de caso de garantia estruturado utilizando a GSN modular. Fabricantes e agências reguladoras devem utilizar o padrão de casos de garantia ACES para representar casos de garantia com GSN. As notações gráficas dos elementos de GSN estão documentadas em uma especificação disponível em [60], que foi desenvolvida em um processo envolvendo usuários GSN da academia e indústria entre 2007 e 2011. O documento resultante da implementação do padrão ACES deve ser composto por notações definidas nesta especificação.

Notações gráficas dos elementos de GSN devem ser representadas utilizando XML. Cada elemento GSN deve ser relacionado com etiquetas e atributos definidos no documento XML. Evidências incluídas no documento devem conter referências para possibilitar o acesso de agências reguladoras aos artefatos de projeto mantidos por fabricantes de sistemas. A definição de casos de garantia com base em um padrão bem definido e independente de plataforma é útil para o compartilhamento de resultados obtidos por fabricantes e de avaliações realizadas por agências reguladoras. Isso possibilita também a execução de atividades de rastreabilidade de requisitos e verificação de requisitos regulatórios por meio

do próprio documento de caso de garantia.

Um caso de garantia ACES deve seguir as seguintes características principais:

- **administração**: um caso de garantia ACES é mantido por um fabricante de um produto em desenvolvimento;
- **autenticação**: um caso de garantia ACES deve ser autenticado;
- **totalidade**: a autenticação de um caso de garantia ACES se aplica a todo o caso de garantia, e não a partes isoladas do documento;
- **codificação**: um caso de garantia ACES deve ser codificado utilizando a linguagem de marcação extensível;
- **controle de versão**: devem ser mantidas versões de requisitos definidos em casos de garantia ACES.

O início e fim de um caso de garantia ACES deve ser definido utilizando o elemento `<assuranceCase>`. No início do corpo do caso de garantia, informações específicas sobre o produto em desenvolvimento devem ser definidas utilizando o elemento `<device>`. A razão social do fabricante (`<manufacturerLegalName>`), nome fantasia do fabricante (`<manufacturerFantasyName>`), endereço do fabricante (`<manufactuerAddress>`), telefone do fabricante (`<manufacturerPhone>`), e-mail do fabricante (`<manufacturerEmail>`), CNPJ do fabricante (`<manufacturerUniqueIdentifier>`), nome do dispositivo (`<deviceName>`), e descrição do dispositivo (`<deviceDescription>`) são exemplos de elementos que podem ser definidos no corpo do elemento `<device>`.

Casos de garantia são compostos por argumentações sobre propriedades específicas do produto em desenvolvimento (e.g., segurança e eficácia). Portanto, argumentos são representados no padrão ACES utilizando os elementos `<parentArgument>` e `<childArgument>`. O elemento `<parentArgument>` deve ser definido no corpo do elemento `<assuranceCase>` após o elemento `<device>`. O elemento `<parentArgument>` é utilizado para representar a estrutura principal do caso de garantia estruturado com a GSN modular. O elemento `<childArgument>` é utilizado para

representar estruturas contidas no corpo do elemento <parentArgument>. Neste caso, um caso de garantia ACES somente pode conter um <parentArgument> que pode ser composto por vários elementos <childArgument>. Argumentos filho (<childArgument>) devem estar relacionados com módulos GSN específicos.

Os elementos da GSN utilizados durante a definição do padrão ACES incluem os principais elementos da GSN (i.e., *Goal*, *Solution*, *Strategy*, *Context*, *Assumption*, *Justification*, *SupportedBy* e *InContextOf*) e da GSN modular (i.e., *Away Goal*, *Module*, *Contract*, *Away Solution*, *Away Context*, e *Public Indication*). No padrão ACES, estes elementos são estruturados no corpo do caso de garantia ACES utilizando o elemento <group>. Este elemento possui o atributo *type* que é definido por valores restritos ao mesmo nome dos elementos GSN. Sabendo que um caso de garantia pode possuir diversos elementos GSN, é possível defini-los por agrupamentos. Por exemplo, para a representação de uma meta (*Goal*), deve-se definir um elemento XML para representar uma meta específica no corpo do elemento <group>. Cada elemento <group> relacionado com um tipo de elemento GSN somente pode ser definido uma vez dentro do corpo de cada um dos elementos <parentArgument> e <childArgument>.

No padrão GSN são permitidas ligações entre elementos específicos. No padrão ACES, esses relacionamentos devem ser representados utilizando o elemento <relationships>. Sua declaração é opcional, todavia quando ocorrer, deve conter pelo menos um elemento filho utilizado para representar ligações entre elementos GSN. Os elementos principais de um caso de garantia GSN e suas respectivas definições no padrão ACES são descritos na Figura 4.6.

Requisitos funcionais e não funcionais podem ser definidos como sub-afirmações/alegações. Para isso, o atributo denominado *requirement* deve ser definido como *true* (o valor padrão para o atributo *requirement* é *false*). Por outro lado, artefatos de projeto relacionados ao produto em desenvolvimento (e.g., *software*) devem ser associados com soluções. Neste caso, o atributo denominado *artifact* deve ser definido como *true* (o valor padrão para o atributo *artifact* é *false*). Além disso, um atributo denominado externalArtifactUrl deve ser definido para conectar uma solução com um artefato de projeto específico apresentado pelo fabricante como evidência durante um processo de certificação.

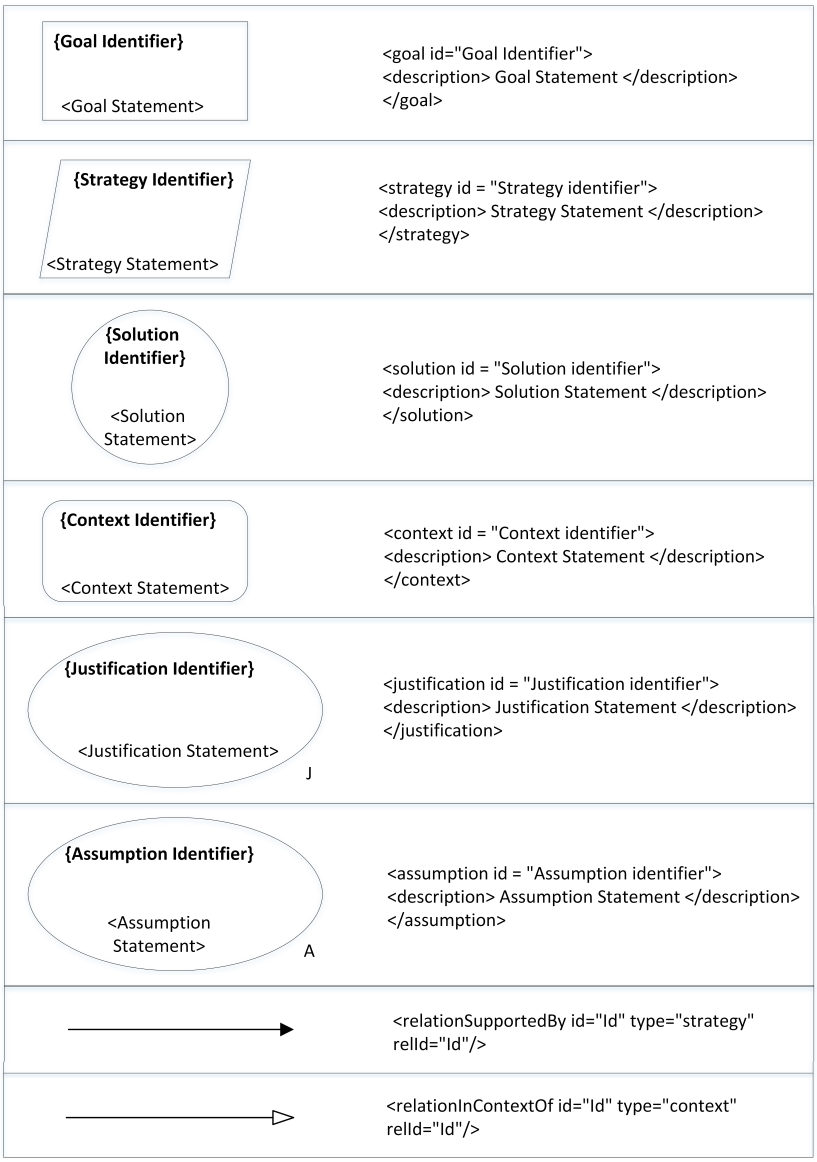


Figura 4.6: Elementos principais de GSN e suas definições em ACES.

Em contrapartida, contextos podem ser utilizados para definir a origem de requisitos funcionais e não funcionais. Neste caso, o atributo denominado *source* deve ser definido como *true* (o valor padrão para o atributo *source* é *false*). Quando uma origem de requisito é declarada, um novo elemento denominado `<externalSourceUrl>` deve ser adicionado e associado com a localização da referência utilizada. O elemento `<justification>`, por sua vez, pode ser utilizado para justificar alterações em requisitos. Neste caso, a justificativa deve ser inserida na versão obsoleta do requisito definido no caso de garantia ACES. Ou seja, uma justificativa deve ser ligada ao elemento `<goal>` utilizado para

representar o requisito obsoleto. As definições de sub-afirmações/alegações como requisitos, soluções como artefatos de projeto, contextos como origem, e justificativas como alterações em requisitos são úteis para a realização da rastreabilidade de requisitos utilizando o padrão ACES.

Além dos elementos de GSN mencionados anteriormente, o documento gerado em XML deve conter informações gerais sobre o produto e resultados de avaliações realizadas por agências reguladoras de sistemas. Informações gerais podem incluir nome, descrição do produto, escopo, e classe do sistema. Por outro lado, elementos utilizados para realizar avaliações podem estar relacionados com argumentos e evidências isolados ou em conjunto. Com esse padrão, é possível integrar os sistemas mantidos por fabricantes e agências reguladoras durante o processo de certificação.

Representação de Casos de Garantia ACES

Como tem sido destacado durante este capítulo, o documento de caso de garantia ACES é composto por um conjunto de informações sobre o sistema embarcado crítico de segurança (e.g., escopo e dados de fabricante), elementos GSN, e resultados de avaliações de agências reguladoras. Casos de garantia ACES devem ser representados de modo que seja possível realizar atividades de rastreabilidade de requisitos e verificações de requisitos regulatórios de maneira automatizada. Entretanto, existem componentes de casos de garantia que não possuem papel fundamental durante essas atividades. Neste caso, a representação de uma grande parte das informações contidas no caso de garantia não possui influência nos resultados, e, consequentemente, o processamento dessas informações se torna desnecessário (argumentações podem ser bastante extensas). Portanto, para as tarefas `Rastreabilidade de Requisitos` e `Verificação de Requisitos Regulatórios`, é necessário representar o caso de garantia em GSN formalmente com base nas informações mais relevantes contidas no caso de garantia ACES.

Neste contexto, deve-se focar em metas (`<goal>`) e soluções (`<solution>`) porque esses elementos possuem um papel fundamental, tanto para a rastreabilidade, quanto para a verificação de requisitos. Mais especificamente, um caso de garantia ACES é representado como um grafo orientado do tipo árvore $T = (V, A)$, onde V é um conjunto de vértices relacionados com metas e soluções, e A é um conjunto de arestas que conectam os vértices.

Formalmente, um caso de garantia ACES é definido como uma tupla de cinco elementos $ACES = (V_g, V_s, v_r, A, R)$, na qual:

- V_g é um conjunto de vértices definidos como metas;
- V_s é um conjunto de vértices definidos como soluções tal que para todo $v_s \in V_s$ o seu grau é igual a 1;
- $V_g \cup V_s$ é um conjunto de vértices V de um grafo acíclico conectado T tal que $V_g \cap V_s = \theta$;
- $A \subseteq V_g \times V_g \cup V_g \times V_s$ é um conjunto de arestas de um grafo acíclico conectado T;
- $v_r \in V_g$ é um vértice específico denominado de raiz da árvore.
- R é um função $R : V_g \cup V_s \rightarrow 2^D$, onde D são descrições de nós.

Um caminho no grafo orientado do tipo árvore T é definido como uma sequência de descrições $c = R(v_1), R(v_2), R(v_3), ...$, tal que $v \in V$. Na função de rótulo R, é definido para cada nó v o conjunto $R(v)$ das descrições associadas com v. Portanto, cada informação relevante para um nó (e.g., fonte de requisito e identificação do requisito) pode ser associada com uma descrição específica.

Durante a aplicação do método, um documento XML especificado com o padrão de casos de garantia ACES deve ser representado utilizando uma estrutura de dados adequada com base na definição formal apresentada anteriormente. Mais especificamente, é proposto que uma estrutura de dados de árvore enraizada com ramificações ilimitadas seja definida para possibilitar a implementação da atividade de rastreabilidade de requisitos de produto e processo, e a verificação de requisitos regulatórios durante o desenvolvimento e certificação de um sistema embarcado crítico de segurança.

Uma representação de filho da esquerda e irmão da direita pode ser utilizada para estruturar uma árvore com ramificações ilimitadas (veja Figura 4.7 adaptada de [15]). O campo `Filho da esquerda[x]` aponta para o filho da extremidade esquerda do nó `x`, enquanto que o campo `Irmão da direita[x]` aponta para o irmão de `x` situado imediatamente a sua direita. Se `x` não possui nenhum filho, `Filho da esquerda[x] = nulo`. Por outro lado, `Irmão da direita[x] = nulo` significa que o nó `x` é o

filho da extremidade direita. Por fim, o ponteiro para o nó pai de `x` é representado utilizando o campo `pai[x]`. Quando um nó `x` é a raiz da árvore, o campo `pai` é definido com o valor nulo como referência. Neste contexto, o nó raiz e os nós intermediários são definidos como metas GSN, e os nós folha são sempre definidos como soluções GSN no grafo.

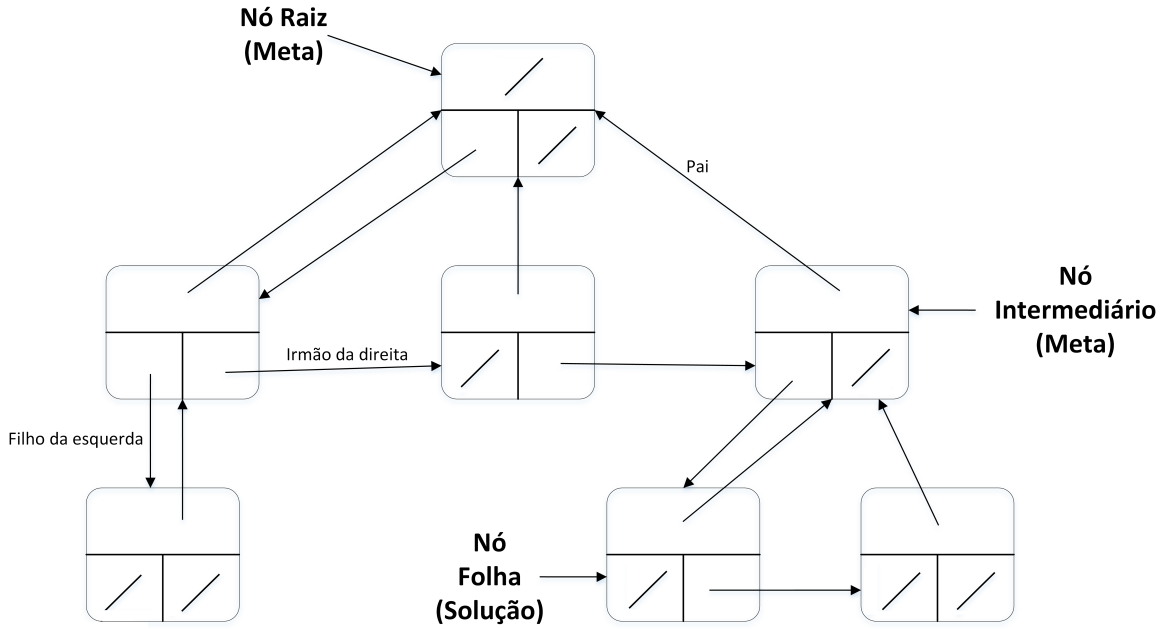


Figura 4.7: Estrutura de árvore enraizada com ramificações ilimitadas.

Note que o nó raiz da árvore e os nós intermediários são representações do elemento `<goal>` em ACES, enquanto que os nós folha são representações do elemento `<solution>` em ACES. É importante destacar que os outros elementos de casos de garantia GSN foram ignorados durante a representação. Como descrito anteriormente, isso ocorre pelo foco no método nas tarefas `Rastreabilidade de Requisitos` e `Verificação de Requisitos Regulatórios` (veja Figura 4.5).

Para gerar a árvore enraizada com ramificações ilimitadas seguindo a definição formal do caso de garantia ACES, considere o pseudocódigo apresentado no Algoritmo 2. Note que o algoritmo possui como entrada um documento XML especificado com base no padrão ACES. Todo o documento é percorrido em busca de elementos associados com metas e soluções para o caso de garantia. A medida que são encontrados, esses elementos são adicionados como nós na árvore de acordo com seu tipo específico, enquanto os demais elementos são ignorados (e.g., estratégias e suposições). O algoritmo possui como saída uma

Algoritmo 2: GERAÇÃO DA ÁRVORE ENRAIZADA

Entrada: Documento XML

Saída: Árvore enraizada com ramificações ilimitadas

```
1  enquanto não chegar ao fim do documento faça
2      Continue;
3      se meta ou solução então
4          se primeira meta então
5              adicione como um nó raiz no grafo;
6          senão
7              adicione como um nó intermediário no grafo;
8          fim
9          se existe relacionamento de suporte com meta ou solução então
10             adicione cada meta ou solução como vértice ligado ao nó no grafo;
11     senão
12         ignore elemento;
13     fim
14 fim
```

árvore enraizada com ramificações ilimitadas. Observe que, em um cenário de uso real, só é necessário gerar a árvore um única vez no início da execução de atividades de rastreabilidade de requisitos e verificação de requisitos regulatórios. Uma vez que a representação está disponível, outros algoritmos podem ser aplicados para extrair informações desejadas.

O pseudocódigo descrito no Algoritmo 3 pode ser utilizado durante a atividade de rastreabilidade de requisitos. Note que uma árvore enraizada com ramificações ilimitadas gerada seguindo a representação formal de casos de garantia ACES é definida como entrada para o algoritmo. O primeiro passo nessa atividade é a extração de uma subárvore para o nó associado com o requisito em análise. Na função denominada `BuscaNó`, é utilizado o conceito de algoritmo de busca em largura para grafos [15]. A busca é realizada a partir do nó inicial (raiz) da árvore enraizada, e os nós filhos de cada nó são expandidos em busca do nó desejado. O algoritmo é executado recursivamente até que o nó desejado é identificado. Quando o nó desejado é encontrado, a função `GeraSubárvore` é utilizada para obter todos

os nós associados (filhos) com o nó desejado (subárvore).

Algoritmo 3: RECUPERAR SUBÁRVORE PARA NÓ DESEJADO

Entrada: Árvore enraizada com ramificações ilimitadas

Saída: Subárvore para o nó desejado

```
1  BuscaNó (Nó Inicial, Nó Desejado, Árvore de Busca)
2  visitado[Nó Inicial] = Verdadeiro;
3  para cada nó filho do nó inicial faça
4      expanda o nó;
5      se o nó expandido é o nó desejado então
6          desejado[Nó Atual] = Verdadeiro;
7          adicione nó desejado como nó raiz da subárvore;
8          GeraSubárvore(Nó Atual, Subárvore);
9      senão
10         BuscaNó(Nó Atual, Nó Desejado, Árvore de Busca + Nó Atual);
11         visitado[Nó Atual] = Falso;
12     fim
13 fim
14 GeraSubárvore (Nó Desejado, Subárvore)
15 visitado[Nó Desejado] = Verdadeiro;
16 para cada nó filho do nó desejado faça
17     expanda o nó;
18     GeraSubárvore(Nó Atual, Subárvore + Nó Atual);
19 fim
```

É importante destacar que a saída obtida com a execução do Algoritmo 3 é um subgrafo do grafo acíclico conectado (gerado por meio do Algoritmo 2). O subgrafo gerado é composto por todos os artefatos de projeto associados com o requisito em análise. Portanto, é possível recuperar informações sobre o requisito, como, por exemplo, a origem do requisito e os artefatos gerados durante a sua implementação. Isso é útil durante o processo de engenharia de requisitos, bem como durante o processo de avaliação de agências reguladoras.

O tempo de execução no pior caso do algoritmo de busca em largura utilizado é $O(|V_g|)$.

Ou seja, no pior caso, o tempo de execução é igual a quantidade de nós do tipo meta. Note que, se os outros elementos GSN em ACES fossem considerados na definição formal do padrão, o tempo de execução poderia aumentar consideravelmente, de acordo com a quantidade de elementos incluídos e com a distância entre o nó desejado e o nó raiz. Essa é uma das justificativas para a definição apresentada.

Para exemplificar a aplicação da definição formal e dos algoritmos descritos acima para realizar a tarefa `Rastreabilidade de Requisitos`, considere o exemplo do sistema de ECG utilizado durante este capítulo. Considere também os argumentos e evidências apresentadas na Tabela 4.1. Dado que o caso de garantia em GSN relacionado com esses argumentos e evidências são representados com ACES, o resultado obtido com a aplicação do Algoritmo 1 seria semelhante ao grafo ilustrado na Figura 4.8. É importante destacar que um grafo completo para o sistema de ECG seria consideravelmente maior em um cenário real. Note que para realizar a rastreabilidade entre o requisito representado pelo nó $v3$ e seus artefatos de projeto, o Algoritmo 2 deve ser aplicado, gerando a subárvore destacada ne Figura 4.8 (linha pontilhada).

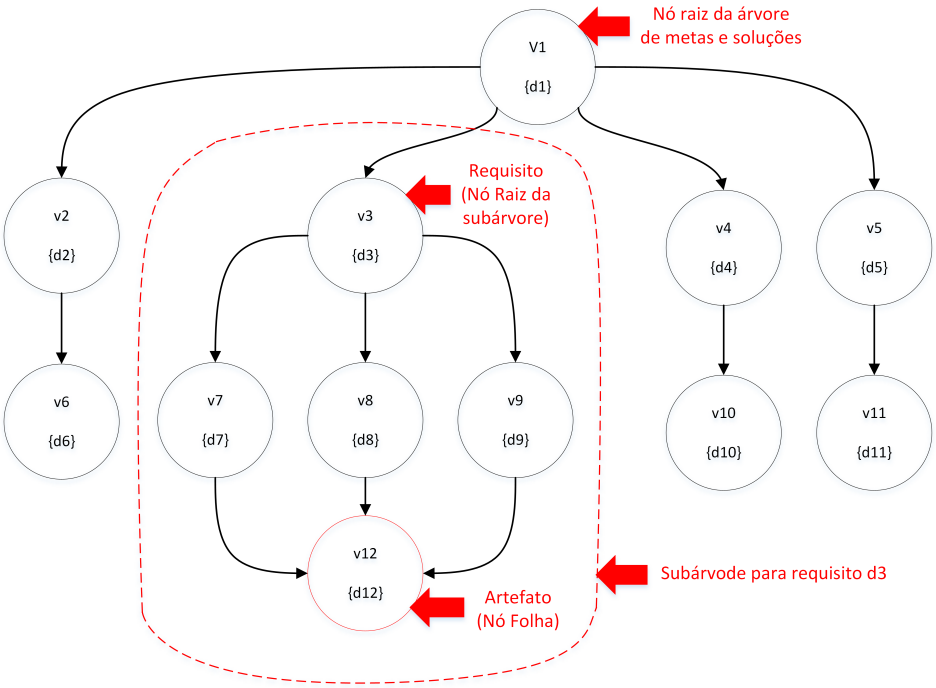


Figura 4.8: Exemplo de árvore enraizada com ramificações ilimitadas para o exemplo de ECG.

Formalmente, o caso de garantia ACES para o grafo apresentado na Figura 4.8 é definido

da seguinte maneira:

- V_g = $\{v1, v2, v3, v4, v5, v7, v8, v9\}$;
- V_s = $\{v6, v10, v11, v12\}$;
- v_r = $v1$;
- A = $\{(v1, v2), (v1, v3), (v1, v4), (v1, v5), (v2, v6), (v3, v7), (v3, v8), (v3, v9), (v4, v10), (v5, v11), (v7, v12), (v8, v12), (v9, v12)\}$;
- R = $\{(v1, \{d1\}), (v2, \{d2\}), (v3, \{d3\}), (v4, \{d4\}), (v5, \{d5\}), (v6, \{d6\}), (v7, \{d7\}), (v8, \{d8\}), (v9, \{d9\}), (v10, \{d10\}), (v11, \{d11\}), (v12, \{d12\})\}$;

Para a rastreabilidade de artefatos de projeto associados com o requisito $v3$ (argumento, linha 1, Tabela 4.1), os seguintes caminhos e execuções sobre descrições são definidos como:

- c_1 = v3, v7, v12 e d_1 = d3, d7, d12;
- c_2 = v3, v8, v12 e d_2 = d3, d8, d12;
- c_3 = v3, v9, v12 e d_3 = d3, d9, d12.

Por exemplo, a descrição $d3$ = p_1, onde p_1 é uma proposição definida de acordo com o argumento descrito na linha 1 da Tabela 4.1. Neste caso, é declarado que "as etapas de filtragem, amplificação e conversão são realizadas de maneira correta durante a execução do sistema de ECG". Esse requisito é desmembrado em três requisitos: filtragem de sinal correta ($d7$), amplificação de sinal correta ($d8$), e conversão de sinal correta ($d9$). O nó folha ($v12$) está relacionado com uma evidência declarada como: $d12$ = "comparação entre resultados"(linha 1, Tabela 4.1).

Avaliação de Requisitos Regulatórios

Por fim, a agência reguladora deve manter uma lista de controle de atributos relacionados com a segurança e eficácia de sistemas avaliados. Agências devem possuir sistemas capazes de interpretar documentos baseados no padrão de casos de garantia ACES (gerados por

fabricantes de sistemas). A segurança e eficácia de sistemas devem ser avaliadas com base em argumentações e evidências estruturadas nestes documentos.

Resultados de avaliações devem ser inseridos no próprio documento XML por meio definição de elementos `<accepted>` e `<rejected>` em argumentos do caso de garantia ACES avaliado, juntamente como uma descrição do motivo da rejeição. Argumentos completos ou específicos podem ser aceitos ou rejeitados. O compartilhamento do documento XML é realizado até a decisão final da agência reguladora com relação a aprovação (ou não) de sistemas. Por exemplo, a agência pode ou não aprovar uma argumentação específica do fabricante e solicitar mais evidências para aumentar sua confiança no funcionamento do sistema desenvolvido. Note que a mesma estrutura de dados e algoritmos podem ser utilizados durante a verificação de requisitos regulatórios.

Durante a avaliação da implementação de padrões prescritivos, a mesma lista de controle de atributos mantida por agências reguladoras para verificar a conformidade com requisitos de padrões utilizados pode ser reaproveitada durante esta atividade. Entretanto, documentos de casos de garantia ACES são o foco na análise. Retomando o exemplo utilizado durante todo o capítulo, a lista de controle utilizada por agências reguladoras para avaliar o documento XML (caso de garantia), considerando características específicas de um sistema de ECG, deve conter ao menos os seguintes requisitos:

1. verificação de valores de impedância eletrodo-pele (*software*);
2. verificação de níveis de bateria (*software*);
3. funcionamento correto de amplificadores (*hardware*);
4. funcionamento correto de filtros (*software/hardware*);
5. funcionamento correto de conversores (*hardware*).

Requisitos são avaliados utilizando métricas associadas com a avaliação de segurança e eficácia do sistema embarcado crítico de segurança. Portanto, resultados obtidos com a técnica de verificação de modelos são recomendados como métricas para os requisitos 1 e 2 (nível de modelos do sistema). As métricas para avaliação de desempenho RMSE e MAE (mencionadas anteriormente) são recomendadas para os requisitos 3, 4, e 5 (nível de modelo e protótipo do sistema).

Dado que é possível gerar a representação seguindo a definição de caso de garantia ACES com o Algoritmo 2 e realizar a rastreabilidade de requisitos considerando o Algoritmo 3, esses passos também podem ser utilizados por agências reguladoras para verificar argumentos e evidências apresentadas por fabricantes. Portanto, argumentos (requisitos) e evidências (artefatos de projeto) podem ser rastreados e avaliados.

Como exemplo de cenário de uso, uma arquitetura orientada a serviços (*Service-Oriented Architecture - SOA*) [84] pode ser utilizada para possibilitar que a técnica de verificação de modelos seja utilizada para confirmar os resultados apresentados por fabricantes de sistemas embarcados. Mais especificamente, serviços podem ser definidos por fabricantes para cada um dos artefatos de projeto gerados, e disponibilizados para consumo de agências reguladoras. Cada um dos artefatos podem ser rastreados para seus requisitos, e fórmulas ASK-CTL utilizadas por agências reguladoras para executar novamente as verificações. A implementação de serviços web é suportada pelo *software* CPN/Tools.

É importante destacar que o foco nestas avaliações está em requisitos regulatórios e na completude de argumentações (não na sua validade). Para avaliar a validade de argumentações, é necessário considerar todos os componentes definidos em um caso de garantia GSN. No contexto deste trabalho, completude é definida como a conformidade com requisitos mínimos necessários para um sistema embarcado crítico de segurança específico, enquanto que validade é definida como o grau de convencimento de que os argumentos e evidências apresentadas são convincentes. A verificação de validade está fora do escopo deste trabalho.

Um diagrama de atividades é apresentado na Figura 4.9 para ilustrar as ações realizadas, desde o início da aplicação do método, para possibilitar a realização da tarefa `Verificação de Requisitos Regulatórios`. O fluxo é iniciado pela especificação informal e semiformal de requisitos do sistema embarcado, seguido pela especificação formal de requisitos de padrões e de produto. Uma vez que essas atividades são finalizadas, o caso de garantia ACES é definido por meio de um documento XML. Caso o documento esteja de acordo com as especificações do padrão ACES, a arvore enraizada com ramificações ilimitadas é gerada, e a rastreabilidade e validação de cada um dos requisitos regulatórios é conduzida. O processo é repetido até que todos os requisitos contidos em argumentos sejam validados. A transição entre caso de garantia como um documento XML,

representação formal com grafos, e a aplicação de algoritmos para percorrer caminhos nos grafos gerados, deve ficar clara durante a implementação da atividade `Casos de Garantia` no método proposto.

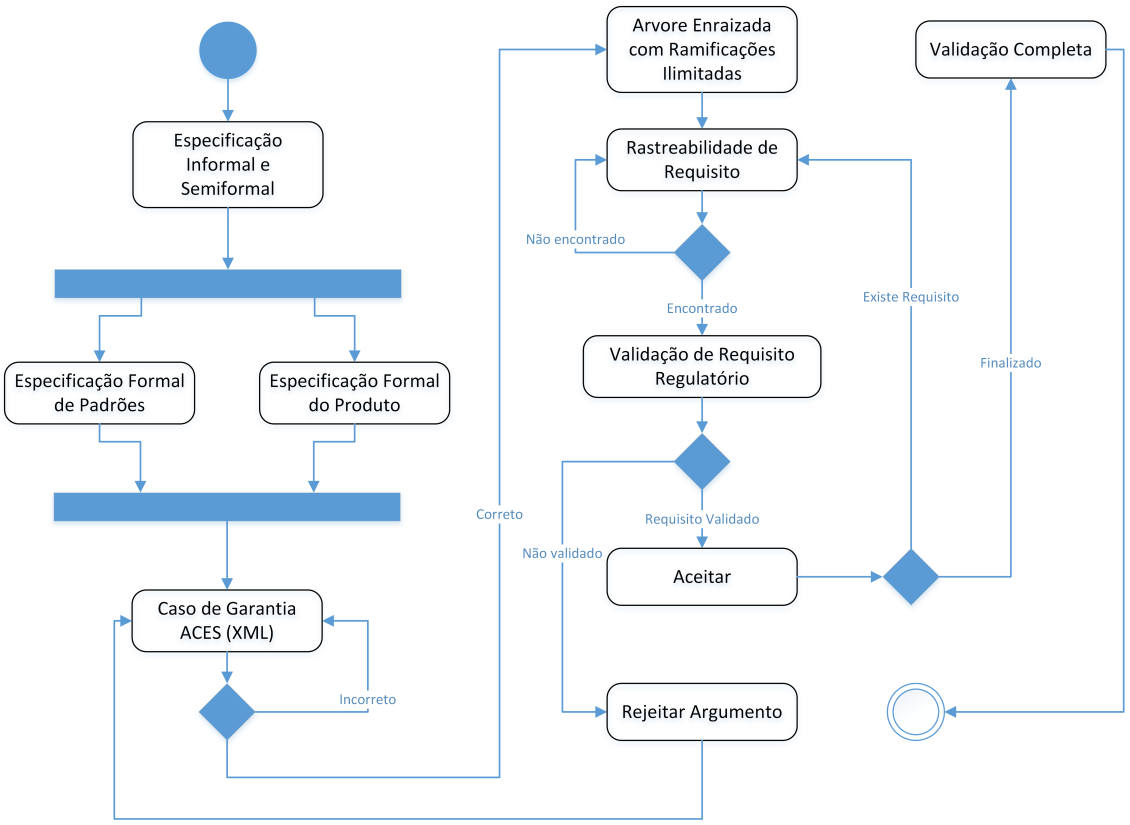


Figura 4.9: Diagrama de atividades da tarefa de verificação de requisitos regulatórios.

4.3 Sumário do Capítulo

Neste capítulo foi apresentado um método para aumentar a confiança no funcionamento de sistemas embarcados críticos de segurança. Os conceitos de redes de Petri coloridas, de casos de garantia e da notação estruturada por metas descritos no Capítulo 2 foram utilizados durante a definição do método. Cada uma das atividades principais `Especificação Informal e Semiformal`, `Requisitos de Padrões`, `Requisitos do Produto`, e `Casos de Garantia` definidas no método foram descritas em detalhes.

Além disso, foi abordada a realização da rastreabilidade de requisitos, compartilhamento

de resultados obtidos com agências reguladoras, e verificação de requisitos regulatórios utilizando o padrão de casos de garantia ACES. Conceitos sobre a metodologia definida para a especificação de requisitos do produto são também apresentados no artigo [78]. Uma visão geral do método apresentado neste capítulo foi descrita nos artigos [76] e [77].

Capítulo 5

Estudo de Caso: Sistemas de Aquisição de Sinais Biomédicos

Neste capítulo é apresentado um estudo de caso sobre um tipo específico de sistema médico embarcado crítico de segurança. Um sistema de aquisição de sinais biomédicos (sistema de Eletrocardiografia - ECG) configurado como monitor cardíaco foi escolhido como sistema embarcado durante a realização do estudo de caso.

Apesar de ser um sistema embarcado para sensoreamento, sistemas de ECG podem ser considerados como críticos porque resultados de medições inconsistentes podem induzir cuidadores a erros durante o tratamento e diagnóstico de pacientes (e.g., em um cenário de unidade de terapia intensiva). Além disso, sistemas de ECG fazem parte de sistemas mais complexos, como, por exemplo, sistemas de marcapassos cardíacos. O estudo de caso é utilizado para avaliar experimentalmente o método proposto e demonstrar como fabricantes podem aplicá-lo durante o desenvolvimento de um tipo de *software* de sistema embarcado específico.

5.1 Descrição do Sistema Utilizado

Esta seção está relacionada com a aplicação da atividade `Especificação Informal e Semiformal de Requisitos` definida no método apresentado no Capítulo 4. A área médica foi escolhida porque sistemas médicos embarcados são passíveis de certificação por uma entidade governamental. Agências reguladoras, como, por exemplo,

a Administração de Alimentos e Drogas (*Food and Drug Administration - FDA*) e a Agência Nacional de Vigilância Sanitária (ANVISA), são responsáveis por avaliar se Sistemas Médicos Embarcados (SME) estão aptos para comercialização. Fabricantes de SME devem comercializar sistemas que não ofereçam riscos à saúde de seus usuários, ou que ao menos minimizem os riscos a um nível aceitável, em que benefícios clínicos superem os riscos. Existem casos em que é necessário que características, tal como a segurança de um sistema, sejam avaliadas. Os SME são geralmente utilizados para o diagnóstico e tratamento de pessoas com, desde problemas relativamente simples de saúde, até com quadros clínicos complexos. Exemplos de sistemas simples e complexos incluem o controle de medicamentos e de ambientes cirúrgicos, respectivamente.

O tipo de sistema médico escolhido para a realização do estudo de caso foi o de sistemas de aquisição de sinais biomédicos. A aquisição de sinais biomédicos é um processo utilizado para isolar uma determinada grandeza física possibilitando a realização de análises em sistemas digitais atuais. Transdutores (ou sensores) são utilizados para transformar sinais biológicos em sinais elétricos. A saída elétrica do sensor (sinal analógico) passa por etapas de condicionamento e conversão de sinais [95]. Na etapa de condicionamento, sinais analógicos podem ser amplificados e filtrados. A amplificação do sinal é necessária porque sinais elétricos possuem baixa amplitude, enquanto que, a filtragem é realizada para eliminar interferências indesejadas no sinal.

Sistemas de aquisição de sinais biomédicos são compostos por componentes de *hardware* e *software* [46]. O *hardware* é geralmente composto por transdutores, amplificadores, filtros, e conversores. O *software* é utilizado para realizar verificações, como, por exemplo, valores de impedância eletrodo-pele[1] e níveis de bateria disponíveis, e também o processamento digital de sinais. Eletrocardiógrafia (ECG) [68], Eletrogastrografia (EGG) [98], Eletroencefalografia (EEG) [49], e Eletromiografia (EMG) [99] são exemplos de sistemas de aquisição de sinais biomédicos.

Por exemplo, o processo de aquisição de sinais biomédicos de um sistema de ECG é representado com o diagrama de blocos ilustrado na Fig. 5.1. No primeiro passo do processo de aquisição de sinais biomédicos, eletrodos são posicionados no corpo do paciente para

[1]Valores de impedância eletrodo-pele são utilizados para verificar se eletrodos estão posicionados corretamente no corpo de pacientes.

realizar a aquisição dos sinais. No segundo passo, o condicionamento de sinal é realizado por meio do processamento analógico de sinais (i.e, amplificação e filtragem). O terceiro passo consiste da conversão do sinal analógico para digital. Note que o sinal biomédico pode ser processado ainda mais durante o terceiro passo do processo de aquisição de sinais (e.g., com a aplicação de filtros digitais). No quarto passo, um sinal de ECG no tempo discreto é obtido.

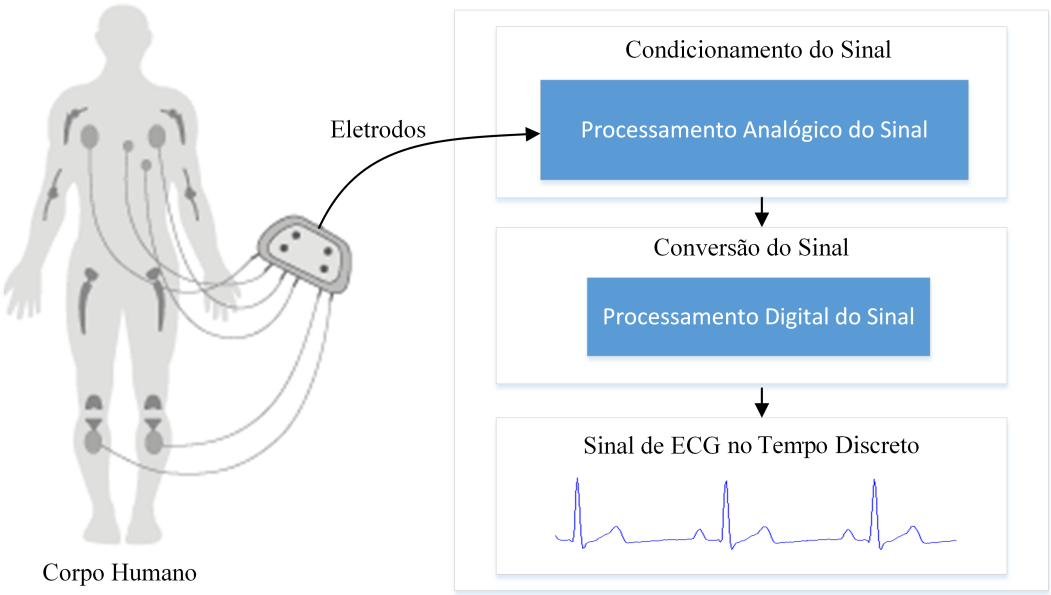


Figura 5.1: Diagrama de blocos para o processo de aquisição de sinais de ECG.

5.1.1 Amplificação

Sinais analógicos são providos por transdutores em escalas de microvolts (uV) ou milivolts (mV) e com processamento de pouca energia. Esses sinais possuem uma amplitude muito baixa para o processamento, portanto, devem ser amplificados por amplificadores operacionais para possibilitar sua manipulação. Além disso, dependendo da especificação de um amplificador, pode ser necessário projetá-lo em `cascata` por meio de dois ou mais estágios de amplificação. Neste caso, o sinal de saída do primeiro estágio de amplificação é acoplado na entrada do próximo estágio, e assim sucessivamente.

Na construção de equipamentos para realização da aquisição de sinais biomédicos,

amplificadores de instrumentação podem ser utilizados para amplificar o sinal. Amplificadores de instrumentação recebem dois sinais como entradas e disponibilizam uma saída diferencial. A diferença entre as tensões de dois sinais é amplificada, e sinais comuns entre as duas entradas são rejeitados. A saída de um amplificador de instrumentação pode ser representada utilizando a seguinte equação:

$$v_o = (A_d v_{Id} + A_{cm} v_{Icm}) \tag{5.1}$$

onde $v_{Id} = v_{I2} - v_{I1}$ é o sinal diferencial de entrada, $v_{Icm} = \frac{1}{2}(v_{I1} + v_{I2})$ é o sinal de entrada de modo comum, A_d é o ganho diferencial, A_{cm} é o ganho de modo comum (idealmente zero), e v_o é o sinal de saída [72].

5.1.2 Filtragem

Filtros passivos são utilizados para eliminar ruídos durante a aquisição do sinal desejado. Ruídos de alta e baixa frequência podem ser atenuados com filtros de passa-baixa (i.e., rejeita frequências acima da frequência de corte), passa-alta (i.e., rejeita frequências abaixo da frequência de corte), passa-faixa (i.e., rejeita frequências fora da faixa de frequências de corte), e rejeita-faixa (i.e., rejeita frequência específica). Além dos filtros passivos, existem também filtros ativos. Nos filtros ativos, além da atenuação de ruídos, é possível aplicar ganhos na amplitude do sinal.

A resposta de um filtro de passa-baixa com frequência de corte normalizada de 0.3 rad/amostra é apresentada na Figura 5.2. O comportamento ideal de um filtro passa-baixa é representado com uma linha pontilhada. É possível observar que existem desvios no comportamento do filtro real em relação ao comportamento ideal. Nas próximas seções são apresentados conceitos sobre a filtragem de sinais utilizando filtros analógicos e digitais, juntamente com conceitos sobre a filtragem de sinais no domínio da frequência. Esses conceitos são apresentados para contextualizar a técnica de filtragem escolhida durante o restante do estudo de caso sobre o sistema de ECG.

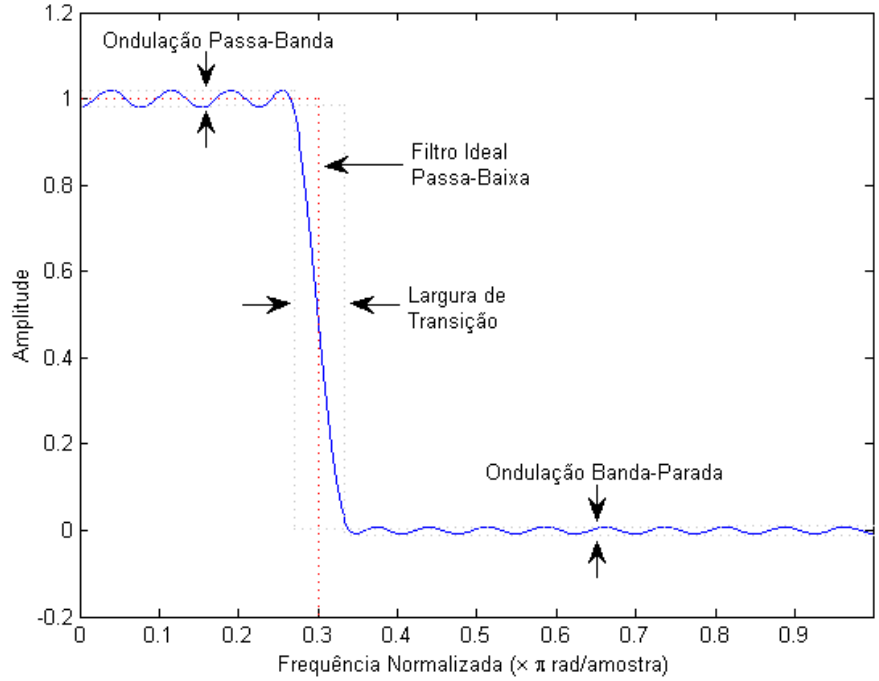


Figura 5.2: Resposta de um filtro de passa-baixa com frequência normalizada de 0.3 rad/amostra.

Filtragem Analógica e Digital

Filtros analógicos são utilizados para filtrar dados indesejados como, por exemplo, os adicionados por ruídos de alta frequência em linhas de transmissão de sistemas de potência. Butterworth, Chebyshev, and Bessel são tipos de filtros analógicos comumente utilizados [55]. Algumas características de filtros podem ser observadas para a escolha de um filtro em um projeto de sistema. Por exemplo, um filtro Butterworth possui a resposta mais plana durante a banda passante, mas possui inclinação muito baixa na fase de decaimento além de não possuir boas características com relação ao espectro de fase. Um filtro Chebyshev possui transição mais íngreme, porém, também com características pobres de fase. Um filtro Bessel é um filtro de fase linear com resposta de amplitude menos acentuada que filtros Butterworth e melhores propriedades no domínio do tempo que filtros Butterworth e Chebyshev.

Entretanto, mesmo com a apresentação de benefícios em sua utilização, filtros analógicos estão sujeitos a umidade, temperatura, entre outros fatores climáticos e ambientais. Portanto, podem introduzir deslocamento não linear de fase, distorção do sinal de entrada, entre outros erros [10]. Por exemplo, se um filtro não possui o mesmo atraso para os diversos

componentes do formato de onda, existirá uma distorção no formato da onda de saída.

Por outro lado, filtros digitais são programáveis e não estão sujeitos aos mesmos fatores que os filtros digitais. Isso significa que filtros digitais eliminam erros associados com condições do ambiente e climáticas. É possível obter especificações de desempenho que seriam difíceis de se obter com filtros analógicos. Projetistas de filtros digitais devem calcular valores de coeficientes ao invés de valores de resistores, capacitores e indutores de filtros analógicos. Existem dois tipos de filtros digitais: Resposta de Impulso Finito (*Finite Impulse Response - FIR*) e Resposta de Impulso Infinito (*Infinite Impulse Response - IIR*). Projetistas de filtros podem obter níveis de desempenho que não seriam possíveis com filtros analógicos ao utilizarem filtros FIR, enquanto que filtros IIR possibilitam a simulação do desempenho de filtros analógicos tradicionais.

Filtragem no Domínio da Frequência

Como nos filtros analógicos e digitais descritos anteriormente, é possível remover dados indesejados de um sinal desejado por meio de uma filtragem de sinais no domínio da frequência. Com a filtragem no domínio da frequência, a integridade de sinais é preservada e a referência temporal é retida. Esse tipo de filtragem pode ser realizada utilizando os conceitos da transformada de Fourier apresentados no Capítulo 2.

O diagrama de blocos ilustrado na Figura 5.3 é utilizado para representar os passos de uma abordagem de filtragem no domínio da frequência. O primeiro passo utilizado para aplicar a filtragem no domínio da frequência é computar a transformada de Fourier do sinal de entrada ($x(n)$) utilizando o algoritmo da transformada rápida de Fourier (*Fast Fourier Transform - FFT*) (i.e., $X(j\omega) = \text{FFT}[x(n)]$). Neste caso, FFT$[x(n)]$ recebe um vetor de dados ($x(n), n = 1, ..., N$) e gera um novo vetor $X(j\omega)$ do mesmo tamanho. Uma vez que a FFT do sinal de entrada é obtida ($X(j\omega)$), é necessário atenuar os coeficientes indesejados para se obter o sinal filtrado ou desejado ($X_f(j\omega)$) por meio da seguinte equação:

$$X_f(j\omega) = X(j\omega) \odot H(j\omega) \tag{5.2}$$

no qual $H(j\omega)$ representa a FFT de uma ou a combinação de funções de transferência de filtros passa alta, passa baixa e rejeita faixa, e $X_f(j\omega)$ é o sinal filtrado obtido pela multiplicação de cada elemento de $X(j\omega)$ e $H(j\omega)$ utilizando o operador $\odot$. Finalmente,

$X_f(j\omega)$ é transformado inversamente utilizando o algoritmo da transformada rápida de Fourier inversa (*Inverse Fast Fourier Transform - IFFT*) para se obter o sinal de saída filtrado no domínio do tempo $x_f(n)$. A filtragem no domínio da frequência descrita nesta seção foi a técnica de filtragem de sinais utilizada durante a especificação formal do sistema de ECG. Como em filtros digitais, os problemas associados com características climáticas e ambientais encontrados em filtros analógicos não são encontrados durante a aplicação desta técnica. Além disso, problemas associados com características no espectro de fase em filtros digitais não são encontrados.

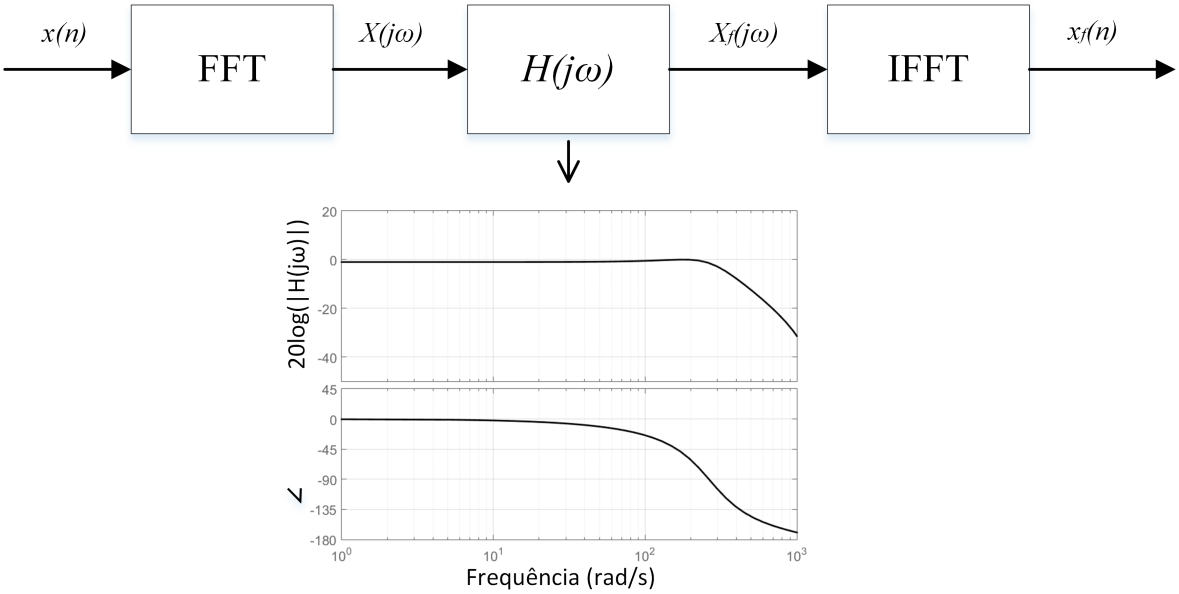


Figura 5.3: Abordagem de filtragem no domínio da frequência.

5.1.3 Conversão

Na etapa de conversão, o sinal analógico é convertido para o formato digital. A conversão possibilita a manipulação do sinal em sistemas digitais atuais. A conversão do sinal é necessária para possibilitar a aplicação dos conceitos da filtragem no domínio da frequência descritos na seção anterior. Funções simples de processamento de sinais podem ser realizadas por meio do processamento analógico, entretanto, em situações mais complexas pode ser necessário realizar o processamento digital de sinais. Um conversor Analógico-Digital (AD) é utilizado para converter sinais de entrada em códigos em uma escala

específica considerando o número de bits do conversor. O comportamento simplificado de um conversor AD pode ser representado pela seguinte equação:

$$v_d = \left\lfloor \frac{v_o}{(V_{REF}/2^N)} \right\rfloor \tag{5.3}$$

onde $\lfloor q \rfloor$ representa o operador de arredondamento (determina o maior inteiro que não excede q), v_d denota o código digital representando o sinal analógico v_o obtido pela Equação 5.1, N denota o número de bits do conversor e V_{REF} sua tensão de referência [63].

A partir da classe de sistemas de aquisição de sinais biomédicos, um sistema específico de ECG foi selecionado para a realização do estudo de caso. O sistema de ECG selecionado é configurado como um monitor cardíaco baseado em um ECG *front-end* (AD8232) [5] e no microcontrolador analógico de baixo consumo, ARM *cortex* M2 com conversores sigma-delta (ADUCM360) [6]. Portanto, um monitor cardíaco compacto de baixo consumo é considerado pela combinação desses componentes e a aplicação da configuração de monitor cardíaco definido no *data sheet* do componentes AD8232 (veja Figura 5.4, adaptada de [5]). Note que dois eletrodos de sinal e um eletrodo de referência podem ser conectados ao componente AD8232. Com isso, pode-se amplificar os sinais de entrada, e, posteriormente, convertê-los com o componente ADUCM360 para que possam ser processados no formato digital e registrados na memória (ou exibidos).

Considerando as principais características técnicas dos componentes escolhidos (AD8232 e ADUCM360), o sistema de ECG para o monitoramento cardíaco definido para este estudo de caso consiste de:

- dois eletrodos de sinal (braços esquerdo e direito) e um eletrodo de referência (perna direita);
- um conversor AD de 24 bits;
- um amplificador de instrumentação;
- um filtro passa alta de 0.5 Hz de dois polos, seguido por um filtro passa baixa de 40 Hz de dois polos;
- modo de rejeição comum em uma faixa de frequência de 50 Hz até 60 Hz;

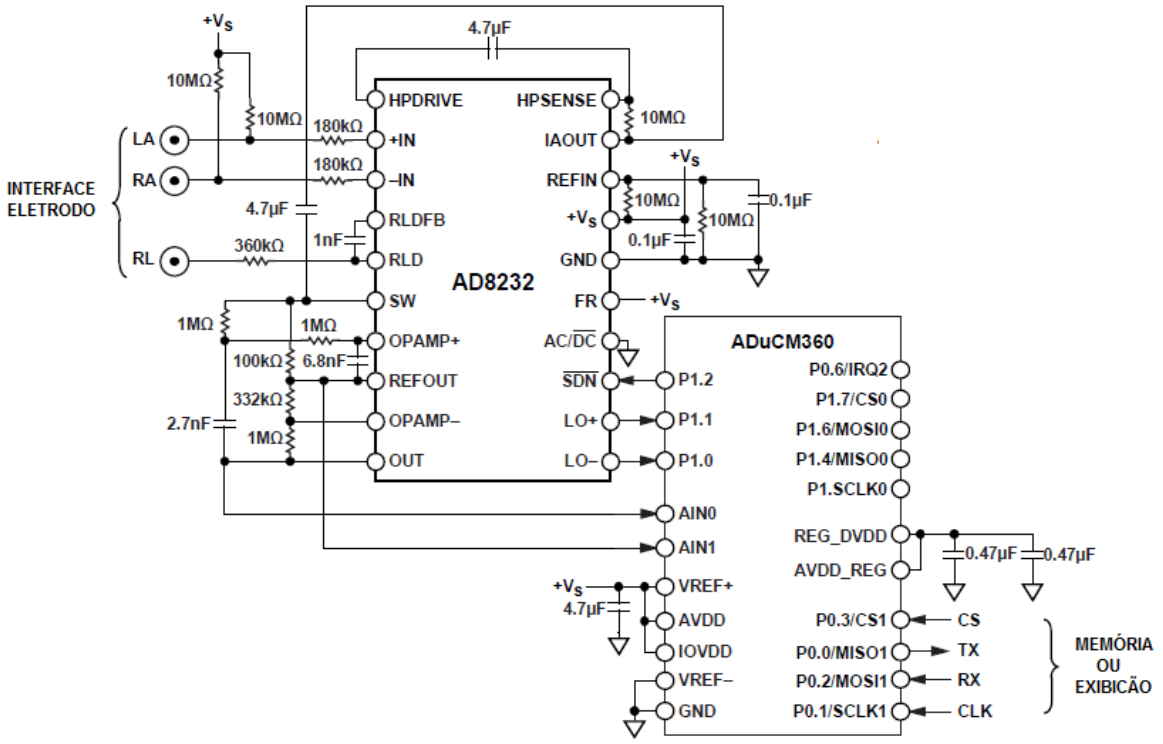


Figura 5.4: Sistema para o monitoramento cardíaco composto por componentes AD8232 e ADUCM360.

- bateria de 1.8 V até 3.6 V (máximo).

Um sistema de ECG deste tipo, configurado como um monitor cardíaco [89; 88], é útil para a verificação do formato dos sinais de ECG por cuidadores, por exemplo, em uma unidade de tratamento intensivo. No sistema considerado, é assumido que pacientes permanecem relativamente quietos durante medições.

5.2 Especificação Formal de Requisitos do Produto

Um modelo de referência de sistemas de aquisição de sinais biomédicos foi especificado de acordo com a definição formal de redes de Petri coloridas hierárquica apresentada no Capítulo 2 e na abordagem de filtragem no domínio da frequência descrita anteriormente. Um especialista no projeto de sistemas médicos foi entrevistado nesta etapa para aumentar a confiança na especificação do sistema. O especialista foi consultado para validar a especificação por meio da análise de resultados de simulação do modelo formal construído

para cada passo do processo de aquisição de sinais.

O módulo principal do modelo de referência de sistemas de aquisição de sinais biomédicos é ilustrado na Figura 5.5. Componentes relacionados ao *hardware* e *software* deste tipo de sistema foram especificados durante a modelagem e divididos em sub-módulos utilizando o mecanismo de estrutura hierárquica de CPN. O modelo é composto por dois sub-módulos denominados `Hardware` e `Software` associados com a representação dos componentes de *hardware* e *software* de sistemas de aquisição de sinais biomédicos. Os conjuntos de cor `ESTADOSS`, `EVENTOSS`, `EVENTOSH` e `ESTADOSH` são definidos para permitir a manipulação de um conjunto de tipos de dados em seus respectivos lugares. Por exemplo, o lugar `Estados_S`, associado com o conjunto de cor `ESTADOSS` pode conter fichas iguais ao conjunto de tipos de dados reais e inteiros como sua cor de ficha. Portanto, é possível que os sub-módulos `Hardware` e `Software` compartilhem estados no modelo CPN.

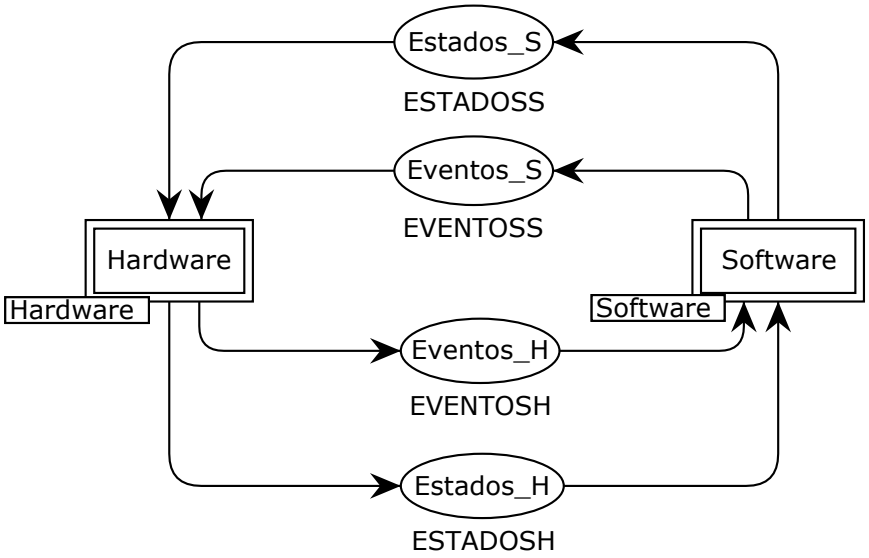


Figura 5.5: Módulo principal do modelo de referência de sistemas de aquisição de sinais biomédicos.

5.2.1 Sub-módulo Hardware

Um evento para iniciar o sistema é representado no sub-módulo de `Hardware` (veja Figura 5.6) como uma transição denominada `Botao Iniciar`. Uma ficha que não

possui tipo de dado definido (conjunto de cor `UNIT`) é consumida e enviada para o lugar `Eventos_H` como uma cor de ficha associada com a expressão de arco `Botao()`. Isso significa que um evento é enviado a partir de *Hardware* para o *Software* notificando que o sistema foi iniciado. Além disso, eletrodos, bateria e o processo de aquisição de sinais do sistema foram representados utilizando as transições de substituição `Eletrodos`, `Bateria`, e `Processa Sinal`.

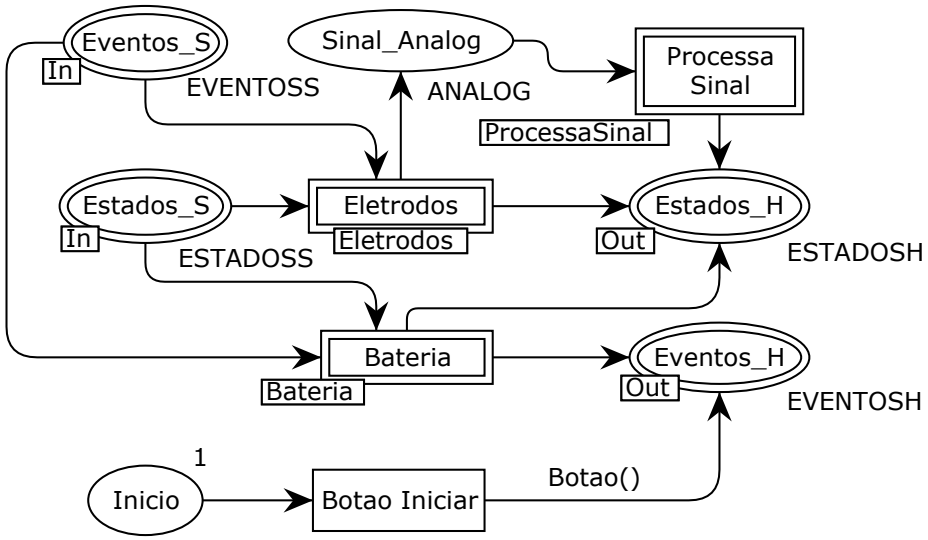


Figura 5.6: Módulo do hardware do sistema.

Na transição de substituição `Bateria`, fichas relacionadas com eventos e estados de *software* são consumidas para disponibilizar a situação atual do nível de bateria e para recarregá-la, quando necessário. Na transição de substituição `Eletrodos`, sinais analógicos adquiridos a partir dos eletrodos são enviados para a realização dos passos de amplificação, filtragem e conversão do processo de aquisição de sinais. O lugar `Sinal_Analogico` é conectado com a transição de substituição `Processa Sinal` por meio de um arco de saída na transição de substituição `Eletrodos` para possibilitar o envio dos sinais.

Sub-módulo Eletrodos

O modelo de referência foi especificado com eletrodos de sinal e referência por meio do sub-módulo apresentado na Figura 5.7. A marcação inicial dos lugares `Eletrodos_Sinal` e `Eletrodos_Referencia` é associada com o número de eletrodos para cada sistema

de aquisição de sinais biomédicos representado. Valores de impedância eletrodo-pele são gerados quando a transição `Impedancia Eletrodos` é disparada baseada em fichas sem tipos de dados (arcos a partir dos lugares `Eletrodos_Sinal` e `Eletrodos_Referencia`) e do tipo de dados inteiro (arco a partir do lugar `Estados_S`) utilizadas para representar os eletrodos e um valor de tensão (expressão de arco `EnviaT(t)`). Além disso, valores de impedância são enviados para o sub-módulo `Software` (lugar `Estados_H`) com a expressão de arco `Impedancia(imp)`.

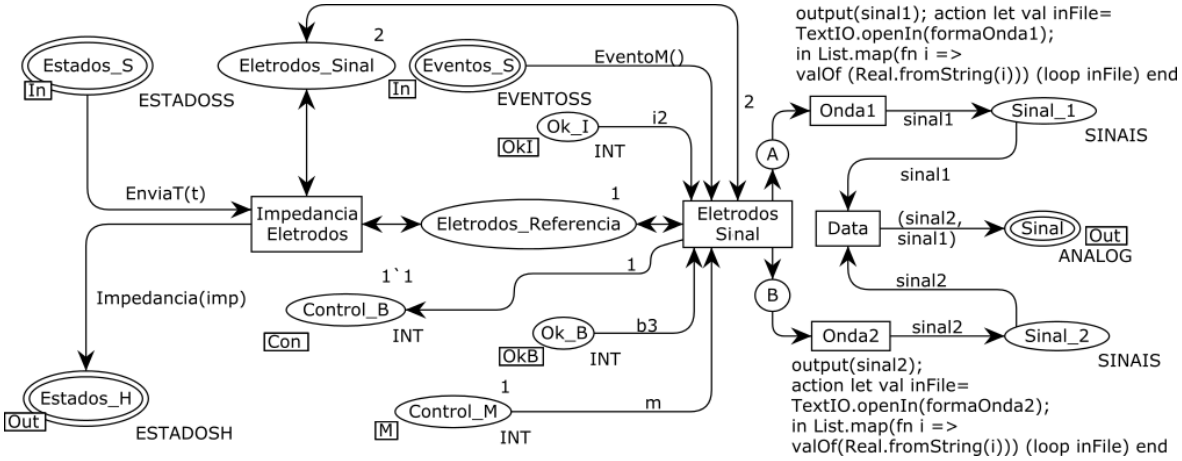


Figura 5.7: Sub-módulo de eletrodos do sistema.

A aquisição de sinais a partir de eletrodos de sinal também é representada com o sub-módulo `Eletrodos` usando a transição `Eletrodos Sinal`. Esta transição é somente habilitada se os lugares de fusão `Ok_I` e `Ok_B` contém fichas. Os lugares de fusão `Ok_I` e `Ok_B` estão relacionados com valores aceitáveis de impedância eletrodo-pele e situação de bateria, e são parte de um conjunto de fusão utilizado para conectar lugares no sub-módulo `Eletrodos` com o sub-módulo `Software`. Isso significa que esses lugares estão contidos nos dois sub-módulos, e, quando alguma mudança ocorre em um lugar em um conjunto, a mudança é refletida em outros lugares contidos no conjunto de fusão. Uma vez que a transição `Eletrodos Sinal` é disparada, fichas são geradas para os lugares `A` e `B` para habilitar as transições `Onda1` e `Onda2`.

Portanto, é possível carregar arquivos de texto associados com a representação digitalizada de sinais analógicos adquiridos a partir dos eletrodos de sinal. Note que é necessário definir os dois primeiros parâmetros de entrada que fabricantes de sistemas de aquisição de sinais biomédicos devem configurar para realizar simulações com o modelo de

referência: as variáveis do tipo *string* `formaOnda1` e `formaOnda2`.

Sub-módulo Bateria

A situação da bateria do sistema (transição de substituição `Bateria` no sub-módulo `Hardware`) foi especificada utilizando o sub-módulo ilustrato na Figura 5.8. Neste sub-módulo, a situação atual da bateria e a ação realizada para recarregar a bateria são especificadas. Uma ficha sem tipo de dados definido é recebida na transição `Valor Atual` a partir do sub-módulo `Software` quando a situação atual da bateria é solicitada (expressão de arco `EventoBateria()`), e a situação atual é retornada utilizando a expressão de arco `Valor(b)`. Por outro lado, uma ficha do tipo inteiro é recebida na transição `Recarrega` a partir do lugar de entrada `Estados_S` de acordo com a expressão de arco `Recarrega(b)` quando o nível atual da bateria está baixo. Fichas do tipo inteiro e sem tipo de dado definido também são enviadas para representar o novo valor do nível da bateria (lugar `Valor`) e um evento de carga total com a expressão de arco `Recarregado()`, respectivamente. A expressão de arco `Recarregado()` está associada com um arco de saída para conectar a transição `Recarrega` com o lugar `Eventos_H`.

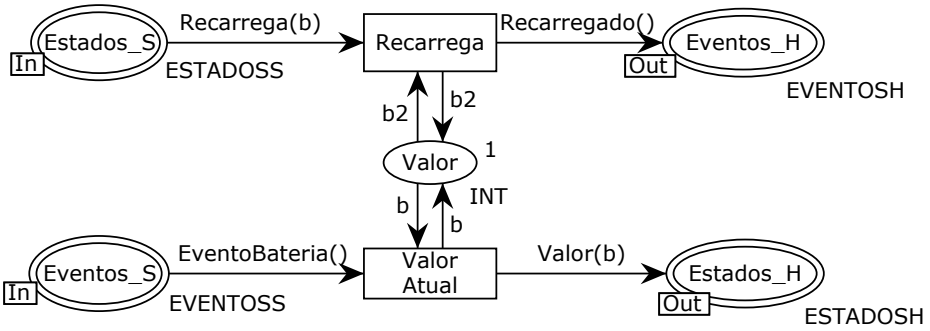


Figura 5.8: Sub-módulo da bateria do sistema.

Sub-módulo Processo de Aquisição de Sinais

Como descrito no início deste capítulo, o processo de aquisição de sinais é utilizado para isolar uma determinada grandeza física para possibilitar a realização de análises em sistemas digitais. Sinais elétricos passam por passos de condicionamento (processamento analógico) e conversão (processamento digital) de sinais. Portanto, três passos foram modelados para

representar o sistema: amplificação, filtragem, e conversão de sinais.

Os passos do processo de aquisição de sinais foram representados com o sub-módulo ilustrado na Figura 5.9. O sub-módulo é composto por uma transição de substituição associada com um amplificador de instrumentação, e duas transições relacionadas aos passos de filtragem e conversão (conversor AD). A representação digital dos sinais adquiridos pelos eletrodos de sinal, representados no sub-módulo `Eletrodos` (veja Figura 5.7), são compartilhados com a transição de substituição `Amplificador Instrumentacao` por meio da sua associação com o lugar `Sinal_Analog`. O lugar `Sinal_Analog` é associado com o conjunto de cor `ANALOG` que contém fichas iguais ao produto de duas listas do tipo real como sua cor de ficha. Fichas são consumidas na transição de substituição `Filtragem Sinal` a partir do lugar de entrada `Preamplificado` (sinais preamplificados obtidos no passo de amplificação). Por fim, as fichas são consumidas na transição `ADC` a partir do passo de filtragem (lugar `Filtrado`), e uma função CPN ML é utilizada para converter os sinais para um formato digital. Os sinais convertidos são enviados como estados de *hardware* para o sub-módulo `Software` por meio do lugar de saída `Estados_H`.

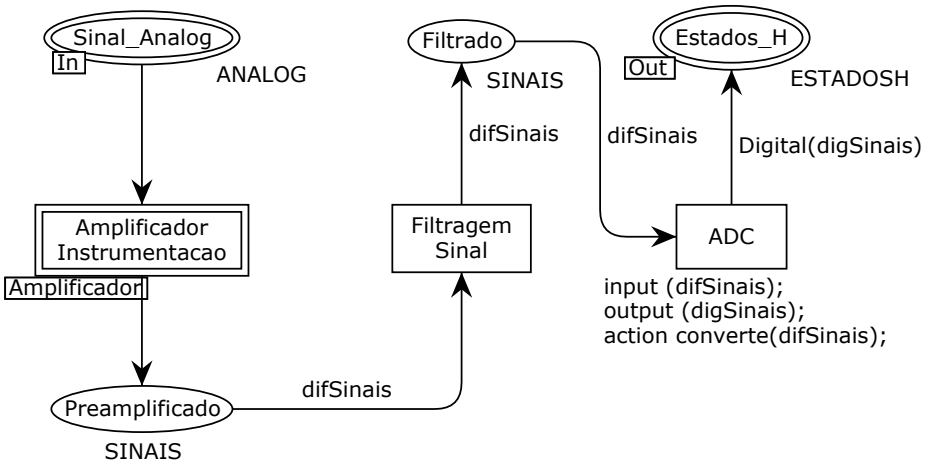


Figura 5.9: Sub-módulo do processo de aquisição de sinais do sistema.

Um amplificador de instrumentação clássico com três amplificadores operacionais é representado no sub-módulo `Amplificador Instrumentacao` (veja Figura 5.10) [42]. Dois sinais de entrada são recebidos a partir do sub-módulo `Eletrodos`, e um único sinal de saída diferencial é provido levando em consideração o modo comum dos sinais de entrada. Note que o sinal diferencial é amplificado utilizando um ganho diferencial

específico. O amplificador de instrumentação foi modelado de acordo com a Equação 5.3, descrita neste capítulo. Portanto, mais dois parâmetros são necessários para que fabricantes possam realizar simulações utilizando o modelo de referência: as variáveis do tipo real `ganhoDiferencial` e `ganhoModoComum`. A função CPN ML `saidaDifAmp(list REAL:s1,list REAL:s2)` foi definida e associada com a transição `Amplificador Diferencial` para computar a Equação 5.1. O sinal diferencial amplificado v_o é definido como entrada para a transição `Filtragem Sinal` (veja Figura 5.9).

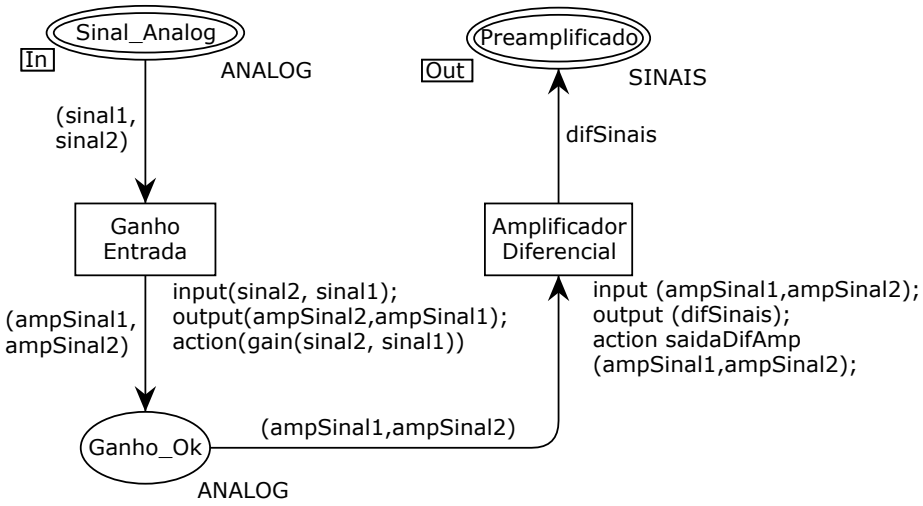


Figura 5.10: Sub-módulo do amplificador de instrumentação do sistema.

A porta de saída `Preamplificado` no sub-módulo `Amplificador Instrumentacao` é associada com a porta de entrada na transição `Filtragem Sinal` para iniciar o passo de filtragem analógica do processo de aquisição de sinais. Este passo no modelo de referência é somente modelado como o evento de transição entre os passos de amplificação e conversão. A filtragem de sinais em uma faixa de frequência para um sistema de aquisição de sinal biomédico específico é especificada durante a representação do *software* no processamento digital de sinais. Portanto, mais dois parâmetros são necessários para que fabricantes possam realizar simulações utilizando o modelo de referência: as variáveis do tipo real `ganhoPassaAlta` e `ganhoPassaBaixa`. Esses parâmetros são utilizados para definir filtros ativos durante o passo de filtragem do processo de aquisição de sinais. Com isso, fabricantes podem configurar um sistema específico para ganho do amplificador em cascata.

No passo de conversão, um conversor AD é especificado para converter a representação

digitalizada de sinais analógicos em códigos com uma escala definida de acordo com o número de bits do conversor. O comportamento simplificado de um conversor AD foi representado com base na Equação 5.3. A função CPN ML `converte(list REAL:dif)` foi associada com a transição `ADC` na Figura 5.9 para computar a equação. O parâmetro `list REAL:dif` desta função é um sinal de saída no domínio do tempo. Portanto, mais três parâmetros devem ser configurados por fabricantes durante a simulação do modelo de referência: as variáveis do tipo real `tensaoReferencia`, `bitsConversor`, e `ajusteLinhaBase`. O parâmetro `ajusteLinhaBase` é útil para ajustar a linha de base do código digital v_d.

5.2.2 Sub-módulo Software

Valores de impedância eletrodo-pele, situação de bateria, códigos digitais, entre outros estados e eventos são recebidos no sub-módulo `Software` a partir do sub-módulo `Hardware`. Primeiramente, um evento é recebido para indicar que o sistema foi iniciado (transição `Iniciar`), uma tensão de 5V é enviada para o eletrodo de referência (transição `Tensao Eletrodos`), e a situação atual da bateria é consultada. Uma vez que os valores de impedância e bateria são obtidos, é realizada a sua verificação para determinar se são aceitáveis ou não, considerando a especificação do sistema de aquisição de sinais biomédicos específico em desenvolvimento.

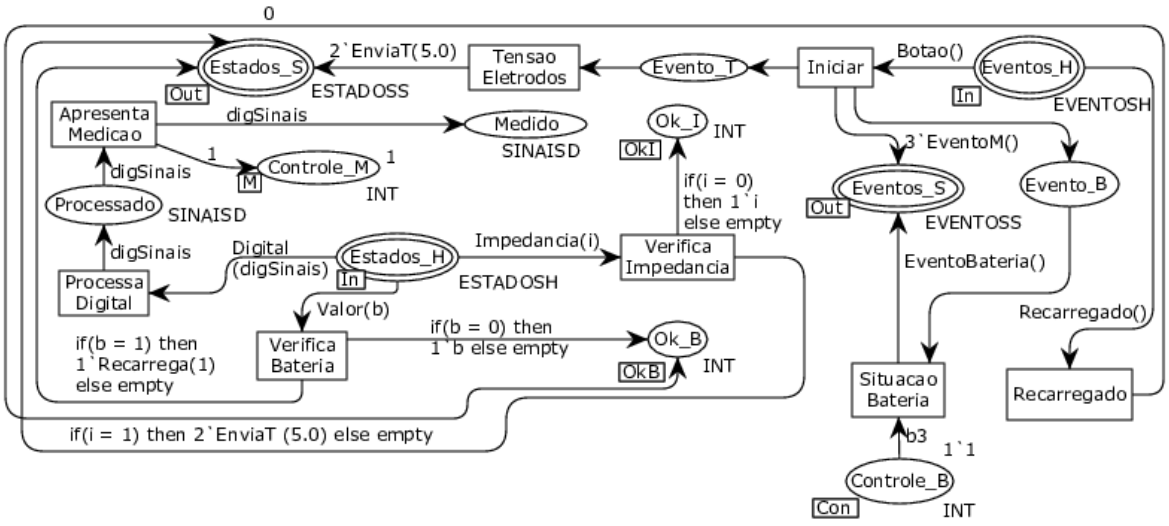


Figura 5.11: Módulo do software do sistema.

As variáveis do tipo inteiro `i` e `b` foram associadas com a transição `Verifica Impedancia` e `Verifica Bateria` no sub-módulo `Software` para analisar se os valores são aceitáveis (`i` = 0 e `b` = 0) ou inaceitáveis (`i` = 1 and `b` = 1). Em caso de valores inaceitáveis de impedância, os valores são verificados novamente até que apresentem valores aceitáveis. Em caso de valores inaceitáveis de bateria, um evento é enviado solicitando que seja realizada a recarga da bateria utilizando a estrutura condicional associada com a expressão do arco de saída `if(b = 1) then 1`Recharge(1) else empty`. A aquisição de sinais só é permitida se os valores de impedância eletrodo-pele e bateria são aceitáveis. Em caso de valores aceitáveis, a aquisição de sinal é permitida enviando o valor contido nas variáveis `i` e `b` para os lugares `Ok_I` e `Ok_B`.

Uma vez que a aquisição de sinais é permitida, e o sinal analógico passou pelos passos de condicionamento e conversão de sinal, um evento é enviado para a transição de substituição `Processa Digital` indicando que o processamento digital de sinais pode ser realizado. A filtragem digital de sinais foi modelada baseada na abordagem do domínio da frequência apresentada na Figura 5.3 para representar o comportamento dos filtros algebricamente.

O sub-módulo `Processa Digital` associado com a filtragem no domínio da frequência é ilustrado na Figura 5.12. As funções CPN ML `fft(list REAL:dif)`, `atenua(list REAL:f)` e `ifft(list REAL:f)` foram definidas para aplicar a filtragem no domínio da frequência utilizando o algoritmo da transformada rápida de Fourier, a atenuação dos coeficientes da função de transferência, e o algoritmo da FFT inversa, respectivamente. Neste contexto, a FFT do código digital v_d ($X(j\omega) = \text{FFT}[v_d]$), o sinal filtrado ou desejado $X_f(j\omega)$, e o sinal de saída filtrado no domínio do tempo $x_f(n)$ ($x_f(n) = \text{IFFT}[X_f(j\omega)]$) são obtidos. Portanto, mais um parâmetro é necessário para que fabricantes possam realizar simulações utilizando o modelo de referência: a variável do tipo string `tipoSistema`.

Fabricantes devem definir o valor da variável `tipoSistema` utilizando uma das seguintes constantes: `monitoraECG`, `diganosticoECG`, `EGG`, `EEG`, e `EMG`. As constantes são definidas para que seja possível configurar o modelo de referência com os coeficientes adequados para a função de transferência relacionada com cada um dos sistemas de aquisição em desenvolvimento. O modelo de referência não está limitado aos sistemas descritos anteriormente. Uma vez que um novo tipo de sistema é identificado, é possível

adiciona-lo facilmente associando uma nova constante com os coeficientes da função de transferência do filtro desejado. Quando a filtragem digital do sinal é realizada, os resultados são disponibilizados para apresentação por meio da transição `Apresenta Medicao` (veja Figura 5.11).

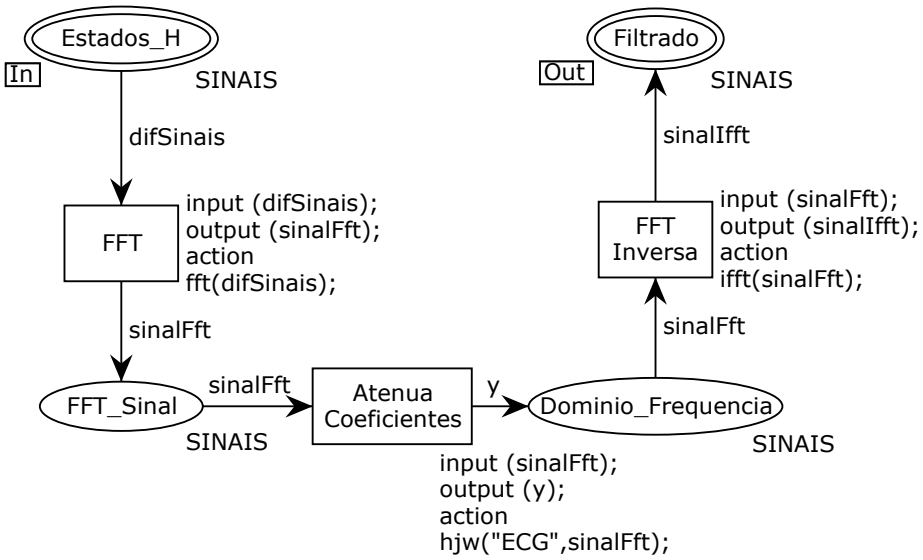


Figura 5.12: Sub-módulo de filtragem de sinal do sistema.

O disparo da transição `Apresenta Medicao` é utilizado para indicar que os sinais digitais estão prontos para ser apresentados para a realização de análises visuais. Por exemplo, projetistas podem integrar o modelo de referência com uma representação gráfica externa para simular um sistema composto por uma interface de usuário para apresentar os sinais adquiridos.

5.2.3 Modelo de um Sistema de ECG

Uma vez que o modelo de referência de sistemas de aquisição de sinais biomédicos está disponível, é somente necessário estender o sub-módulo `Software`, alterar tipos de dados de variáveis, e configurar parâmetros de entrada para representar o sistema de ECG definido no início deste capítulo. Neste contexto, as transições `Verifica Impedancia` e `Verifica Bateria` foram modificadas para transições de substituição para estender o sub-módulo `Software`. O sub-módulo é composto, portanto, por outros sub-módulos para verificação da impedância eletrodo-pele e situação de bateria do sistema de ECG.

O sub-módulo `Verifica Impedancia` é ilustrado na Figura 5.13. A verificação da impedância eletrodo-pele é realizada para dois eletrodos de sinais na transição `Verificacao Impedancia`. Se ao menos um dos valores de impedância estiver acima de um limite de 5000.0 Ohms, um estado contendo tensões de referência é enviado para o módulo do *software* do sistema para requisitar novas verificações. Caso contrário, uma ficha de tipo de dado inteiro é enviada contendo o valor 0 para o lugar `Ok_I`. Note que o tipo de dado da variável `i` foi modificado de inteiro para real e uma variável `cont` foi adicionada para possibilitar a verificação.

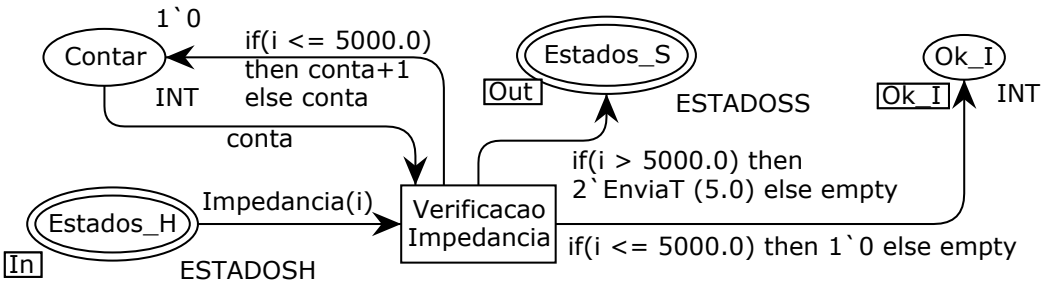


Figura 5.13: Sub-módulo de verificação de impedância do sistema.

No sub-módulo `Verifica Bateria` (veja Figura 5.14), foi definido que é necessário ao menos a metade da carga total da bateria para executar o sistema de ECG corretamente (1.8 V). As transições `Inicio de Vida` e `Fim de Vida` foram adicionadas para realizar a verificação. Quando a carga total da bateria está maior ou igual que 1.8 V, uma ficha do tipo inteiro é enviada por meio da transição `Inicio de Vida` contendo o valor 0 para o lugar `Ok_B`. Caso contrário, a recarga total da bateria é solicitada no *software* utilizando a expressão do arco `Recarrega(3.6)` associada com o arco de saída da transição `Fim de Vida`. Note que o tipo de dado da variável `b` foi alterada de inteiro para real para possibilitar a verificação.

Além disso, uma função CPN ML foi adicionada na transição `Apresenta Medicao` (módulo *Software*) para que a extensão do modelo de referência possa ser conectada com uma representação gráfica externa, e que resultados da simulação do modelo do sistema ECG possam ser apresentados de maneira mais realística. Isso é útil durante a validação do modelo por meio de especialistas do domínio.

A função predefinida disponibilizada na biblioteca de comunicação com processos

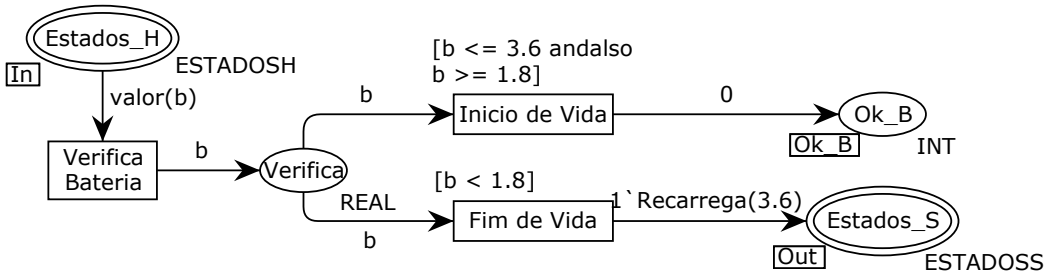


Figura 5.14: Sub-módulo de verificação de bateria do sistema.

externos Comms/CPN ML

openConnection:string*string*int $\rightarrow$ unit

foi utilizada a partir da ferramenta CPN/Tools para conectar o modelo como um cliente *socket* de uma representação gráfica (servidor *socket*). O primeiro parâmetro é um identificador único do tipo *string* associado com uma nova conexão. O segundo e terceiro parâmetro são o nome do servidor e o número da porta associada. Posteriormente, a função

send:string * 'a *('a $\rightarrow$ Word8Vector.vector) $\rightarrow$ unit

foi utilizada para enviar as medições (código digital v_d) por meio da conexão *socket*. O primeiro parâmetro é um identificador do tipo *string* para a conexão *socket*, o segundo é o dado a ser enviado, e o terceiro é uma função para codificar o dado enviado. A função de codificação serve para codificar dados enviados em uma sequência de bytes.

Um exemplo de simulação de modelo utilizando uma comunicação Java *socket* entre o modelo estendido de ECG e uma representação externa para apresentar registros de ECG em tempo de simulação é ilustrada na Figura 5.15. Note que este tipo de simulação é útil para realizar avaliações do projeto de um sistema de ECG com especialistas que não possuem conhecimento técnico sobre métodos formais tais como CPN e a técnica de verificação de modelos (*model checking*). Por exemplo, um cardiologista pode validar o funcionamento do sistema por meio da verificação dos formatos de onda característicos de ECG (ondas P, Q, R, S, e T) obtidos com a execução do sistema.

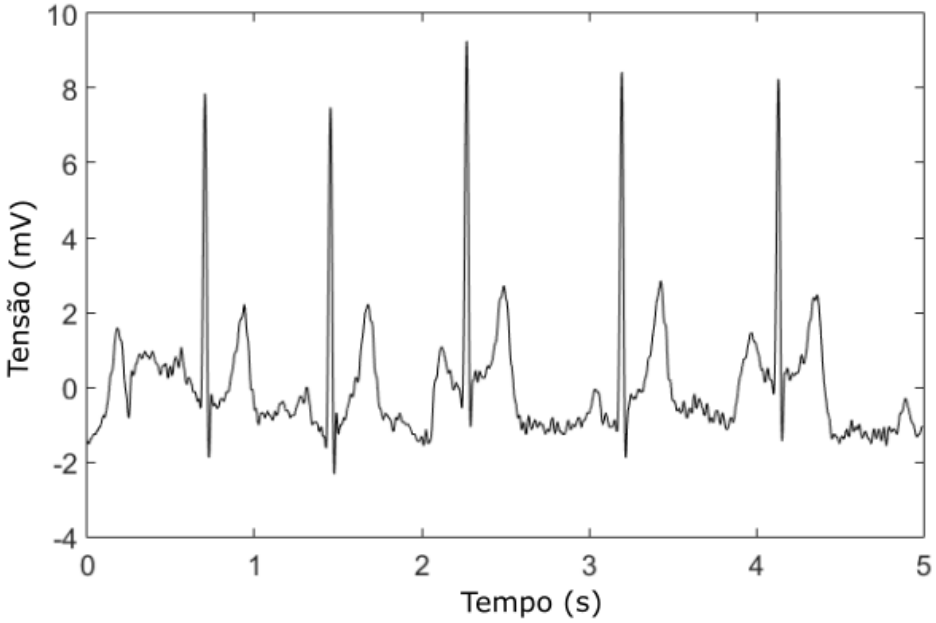


Figura 5.15: Exemplo de resultados de simulação de modelo usando uma representação gráfica externa.

Verificação do Modelo de ECG

Uma vez que o modelo de referência é estendido para representar o sistema de ECG, é possível utiliza-lo para aumentar a confiança em seu funcionamento e gerar evidencias para a certificação no nível de modelagem. Resultados de verificação de modelos são utilizados para demonstrar que o projeto do sistema de ECG em desenvolvimento está de acordo com requisitos não funcionais definidos na atividade `Especificação Informal e Semiformal de Requisitos`. Portanto, o espaço de estado do modelo CPN foi gerado, e a ferramenta de espaço de estado do CPN/Tools foi utilizada para analisar as propriedades padrões casa (*home property*), vivacidade (*liveness property*), e justiça (*fairness property*), registradas no relatório de espaço de estado do modelo.

Uma amostra do relatório do espaço de estado gerado para o modelo de ECG é apresentada na Tabela 5.1. O espaço de estado completo possui 98 nós e 197 arcos. A primeira propriedade analisada foi a propriedade casa (*home*). Neste caso, foi verificado que a única marcação casa é uma marcação final do modelo. Isso significa que é possível alcançar um estado final a partir de qualquer estado inicial. A segunda propriedade analisada foi a propriedade de vivacidade (*liveness*). Foi verificado que o modelo de ECG não possui

Tabela 5.1: Amostra do relatório de espaço de estado gerado no CPN/Tools.

Propriedade	Casa	Vivacidade	Justiça
Resultados	Home Marking[96]	Dead Markings[96]	No infinite occurrence sequences
Estatísticas	Nodes[98]	Arcs[197]	Status[full]

nenhuma transição viva, o que significa que não existe o disparo infinito de transições. Além disso, a única marcação morta registrada no relatório foi identificada como um estado final, significando que o modelo possui o comportamento esperado (i.e., finalizado com medições realizadas contidas no lugar `Medido`). Note que ainda existem fichas no final da execução do modelo CPN (marcação morta), entretanto, não existem transições não disparadas (marcação casa). Finalmente, não existem sequências de ocorrências infinitas de disparos de transições (propriedade de justiça). Portanto, a primeira evidência foi gerada utilizando o modelo de referência especificado por meio dos relatórios padrões de espaço de estado.

Além da análise de espaço de estado, o modelo de ECG foi analisado utilizado a ferramenta CPN/Tools e a técnica de verificação de modelos (*model checking* com a biblioteca ASK-CTL) [73] considerando propriedades específicas do sistema de ECG especificado. Quatro propriedades relacionadas com requisitos não funcionais foram verificadas (veja Tabela 5.2.3). Entretanto, o número de propriedades não está limitado às propriedades descritas neste documento. Um código CPN ML foi definido para especificar cada uma das fórmula ASK-CTL e a verificação de modelos foi executada. O resultado da execução da técnica de verificação de modelos para as quatro propriedades foi "verdadeiro". Isso significa que os requisitos não funcionais especificados foram satisfeitos durante a especificação do modelo formal do sistema de ECG.

Tabela 5.2: Propriedades do Sistema de ECG.

ID	Propriedade	Fórmula ASK-CTL	Fórmula CTL
1	Um exame não pode ser iniciado com eletrodos posicionados incorretamente e bateria descarregada	NOT(POS(AND(NF(AS),AND(NF(OkB),NF(OkI)))))	EF¬(AS ∧ (OkB ∧ OkI))
2	Uma aquisição de sinal não pode ser iniciada enquanto a bateria é carregada	NOT(POS(AND(NF(S),AND(NF(OkB), NF(OkI)))))	EF¬(S ∧ (OkB ∧ OkI))
3	Um exame não pode ser finalizado com eletrodos posiciomados incorretamente e bateria descarregada	POS(AND(NF(M),AND(NF(OkB),NF(OkI))))	EF(M ∧ (OkB ∧ OkI))
4	Uma aquisição de sinal não pode ser interrompida a menos com a finalização do exame ou bateria descarregada	POS(AND(NF(B), POS(OR(NF(M), NF(S)))))	EF(B ∧ EF(M ∨ S))

VERIFICANDO PROPRIEDADE 1

Na primeira propriedade, é especificado que um exame não pode ser iniciado se os eletrodos estão posicionados incorretamente no corpo de pacientes, e se a bateria do sistema não possui carga suficiente para toda a duração do exame. O posicionamento incorreto dos eletrodos afeta negativamente o formato da onda de ECG. Resultados de impedância eletrodo-pele são utilizados para verificar o posicionamento dos eletrodos. Além disso, o sistema deve possuir bateria suficiente para a realização da aquisição de sinais de maneira ininterrupta. Neste caso, a bateria deve estar acima da metade da carga total.

As declarações de fórmula denominadas `AS`, `OkB`, e `OkI` são associadas com os lugares de instância `Hardware'Sinal`, `Eletrodos'Ok_B`, e `Eletrodos'Ok_I` no sub-módulo `Eletrodos`. Na declaração `AS`, é verificado se existem fichas no lugar associado com sinais analógicos na representação do *hardware* do sistema. Na declaração `OkB` (função ML `isntOkB`), é verificado se existe uma ficha com o valor 1. Na declaração `OkI` (função ML `isntOkI`), é verificado se existe um tupla de dois elementos com o valor (1,1).

A verificação de modelos com a fórmula ASK-CTL para a propriedade 1 utilizando a ferramenta CPN/Tools é ilustrada na Figura 5.16(a). Note que o código CPN ML e o resultado são ilustrados demonstrando que o modelo satisfaz a fórmula ASK-CTL. Portanto, a segunda evidência foi gerada utilizando o modelo de referência especificado.

VERIFICANDO PROPRIEDADE 2

Na segunda propriedade, é especificado que uma nova aquisição de sinal biomédico não pode ser iniciada enquanto a bateria do sistema está sendo recarregada. Isso significa que se o sistema de ECG não possui bateria suficiente (i.e., acima da metade da carga total do sistema), uma nova aquisição de sinal de ECG não deve ser iniciada. Além disso, uma aquisição em andamento não deve ser mantida até que a carga da bateria do sistema esteja completa.

As declarações de fórmula denominadas `S`, `OkB` e `OkI` são predicados associados com os lugares de instância `Bateria'Recarrega`, `Eletrodos'Ok_B`, e `Eletrodos'Ok_I`. Na declaração `S`, é verificado se existem uma ficha associada com evento para recarregar

bateria. Na declaração `OkB` (função ML `isntOkB`), é verificado se existe uma ficha com o valor 0. Na declaração `OkI` (função ML `isntOkI`), é verificado se existe um tupla de dois elementos com o valor (0,0).

A verificação de modelos com a fórmula ASK-CTL para a propriedade 2 utilizando a ferramenta CPN/Tools é ilustrada na Figura 5.16(b). Portanto, a terceira evidência foi gerada utilizando o modelo de referência especificado.

VERIFICANDO PROPRIEDADE 3

É necessário também verificar se uma aquisição de sinal biomédico é sempre finaliza com valores de impedância eletrodo-pele baixos (i.e., eletrodos bem posicionados) e valores de bateria altos (i.e., carga de bateria suficiente) durante a execução do sistema. Como descrito anteriormente, um sistema utilizado para adquirir sinais com eletrodos posicionados incorretamente e com carga insuficiente de bateria pode prover registros de ECG imprecisos.

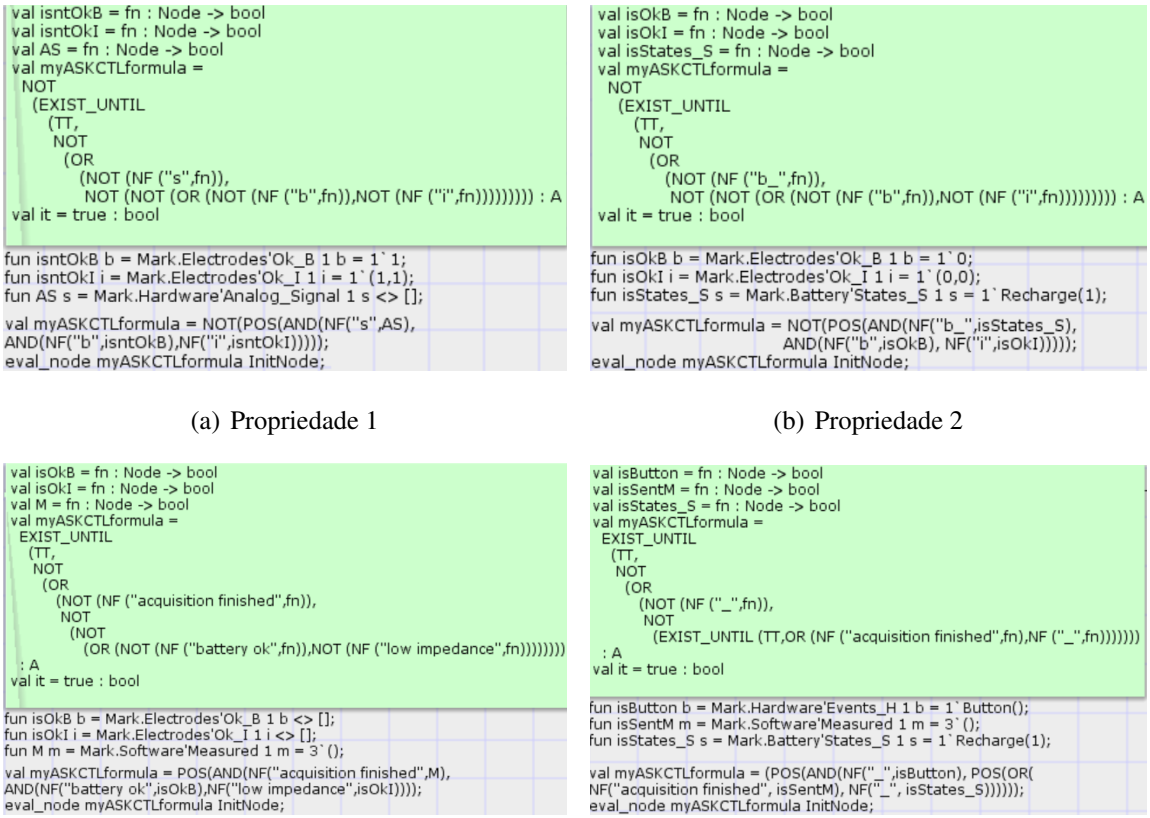
As declarações de fórmula denominadas `M`, `OkB`, e `OkI` são predicados associados com o número de fichas nos lugares de instância `Software'Medido`, `Eletrodos'Ok_B`, e `Eletrodos'Ok_I`. Na declaração `M`, é verificado se existem três fichas no lugar associado com as medições realizadas. Na declaração `OkB` (função ML `isOkB`), é verificado se existem fichas no lugar `Ok_B`. Na declaração `OkI` (função ML `isOkI`), é verificado se existem fichas no lugar `Ok_I`.

A verificação de modelos com a fórmula ASK-CTL para a propriedade 3 utilizando a ferramenta CPN/Tools é ilustrada na Figura 5.16(c). Portanto, a terceira evidência foi gerada por meio do modelo de referência especificado.

VERIFICANDO PROPRIEDADE 4

Finalmente, é necessário verificar aquisições de sinais biomédicos são realizadas no sistema de maneira ininterrupta. Ou seja, uma aquisição deve ser realizada até que um exame seja finalizado ou a carga de bateria esteja baixa. Registros completos de ECG devem ser providos por meio do sistema de ECG para possibilitar análises mais precisas.

As declarações de fórmula denominadas `B`, `M` e `S` são predicados relacionados com os lugares de instância `Hardware'Eventos_H`, `Software'Medido`, `Bateria'Estados_S`. Na declaração `B` (função ML `isButton`), é verificado se



(a) Propriedade 1

(b) Propriedade 2

(c) Propriedade 3

(d) Propriedade 4

Figura 5.16: Resultados da verificação de modelos com fórmulas ASK-CTL.

existe um evento relacionado com o botão iniciar. Na declaração `M`, é verificado se existem três fichas no lugar associado com as medições realizadas. Na declaração `S`, é verificado se existem uma ficha associada com evento para recarregar bateria.

A verificação de modelos com a fórmula ASK-CTL para a propriedade 4 utilizando a ferramenta CPN/Tools é ilustrada na Figura 5.16(d). Portanto, a quarta evidência foi gerada por meio do modelo de referência especificado.

Validação do Modelo de ECG

Como quinto e sexto exemplos, é apresentado como fabricantes de sistemas podem utilizar atividades de validação do modelo de ECG para gerar artefatos de projeto e evidências para certificação no nível de modelagem. Valores foram definidos para os seguintes parâmetros de entrada do modelo de referência: `ganhoDiferencial`, `ganhoModoComum`, `tipoSistema`, `tensaoReferencia`, `bitsConversor`, e `ajusteLinhaBase`. Isso foi útil para realizar simulações de acordo com as especificações

do sistema de ECG em desenvolvimento. Os parâmetros de entrada do modelo de referência, juntamente com descrição e valores definidos, são descritos na Tabela 5.3. Valores foram definidos com base nas principais características de dois componentes comerciais: o ECG *front-end* (AD8232) e o microcontrolador analógico de baixo consumo, ARM *cortex* M2 com conversores sigma-delta (ADUCM360).

Tabela 5.3: Parâmetros de Entrada do Modelo de Referência

Parâmetro	Descrição	Valor
formaOnda1	Sinal do eletrodo A	vazio
formaOnda2	Sinal do eletrodo B	vazio
ganhoDiferencial	Ganho diferencial do amplificador de instrumentação	1.0
ganhoPassaBaixa	Ganho do filtro passa baixa	1.0
ganhoPassaAlta	Ganho do filtro passa alta	1.0
ganhoModoComum	Ganho de modo comum do amplificador de instrumentação	0.0
tipoSistema	Tipo de sistema representado	monitoraECG
tensaoReferencia	Tensão de referência	3.0
bitsConversor	Número de bits do conversor	24
ajusteLinhaBase	Ajuste da linha de base	7.0

Note que os parâmetros `formaOnda1` e `formaOnda2` não foram definidos porque o foco principal com este exemplo de validação é a avaliação do desempenho dos filtros configurados para o ECG. Entretanto, é possível que fabricantes de sistemas utilizem o modelo de ECG para testar cada uma das outras etapas do processo de aquisição de sinais biomédicos separadamente. Portanto, ao invés de definir os parâmetros `formaOnda1` e `formaOnda2` para representar os sinais adquiridos a partir dos eletrodos do sistema, o sinal ilustrado na Figura 5.17 foi definido como a marcação inicial do lugar `Estados_H` (i.e., o sinal de saída v_d do conversor AD) no modelo.

É considerado que as etapas de condicionamento e conversão do sistema foram aplicadas corretamente na sinal biomédico. Os parâmetros de entrada relacionados com a etapa de amplificação de sinais foram definidos considerando o comportamento ideal de um

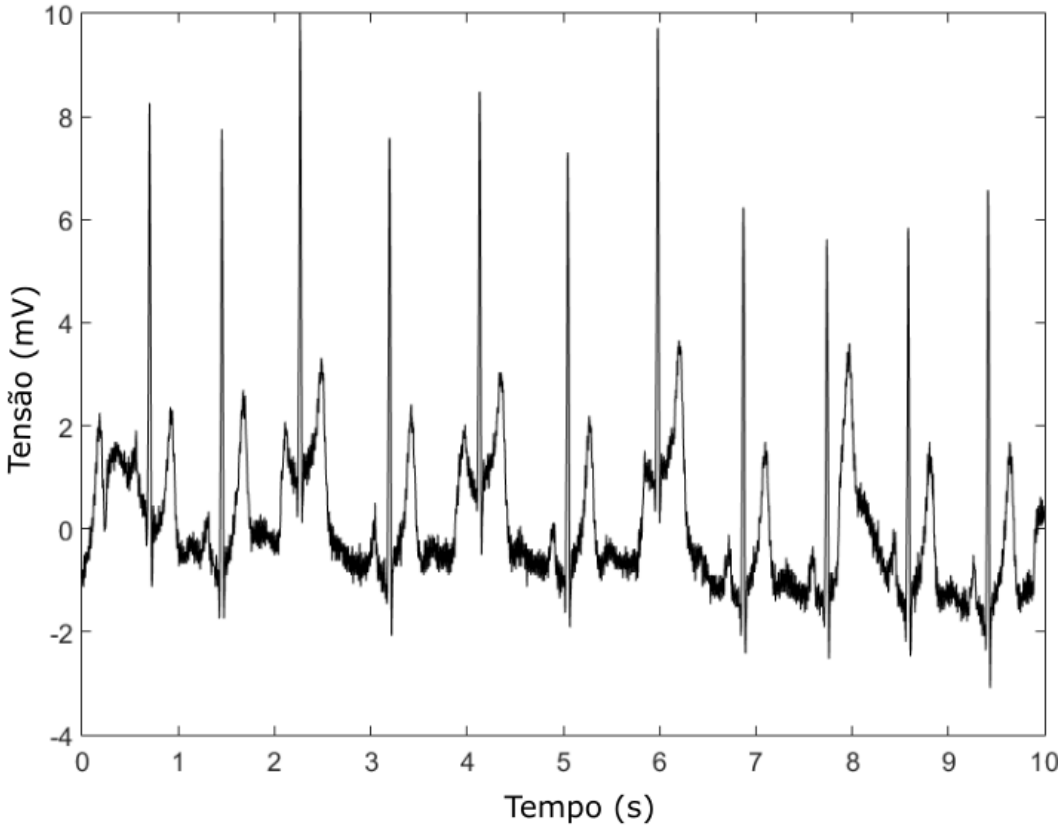


Figura 5.17: Registro de ECG Person_01rec_1 disponibilizado na base de dados PHYSIONET ECG-ID com duranção de 10 segundos.

amplificador de instrumentação (`ganhoPassaBaixa` (1.0), `ganhoPassaAlta` (1.0), `ganhoModoComum` (0.0) e `ganhoDiferencial` (1.0)).

Registros de ECG disponibilizados na base de dados PHYSIONET ECG-ID foram utilizados para realizar simulações com o modelo de ECG estendido. Os registros contidos na base de dados (i.e., sinais filtrados e não filtrados) foram adquiridos por meio de um ECG *single-lead* e eletrodos do tipo *limb clamp*. Estudos experimentais envolveram 90 voluntários para a execução de exames de ECG. Foram gerados registros de ECG com pacientes sentados sem limite de taxa cardíaca, e estados físico e emocional. Os dados coletados consistem de 310 registros de ECG *I-lead* de 90 indivíduos com duração de 20 segundos cada, amostrados em 500Hz com precisão de 12 bits (intervalos de 1000 amostras). A configuração *Lead I* foi escolhida porque é de fácil medição e não é sensível a pequenas variações na localização de eletrodos usando um ECG *single-lead*. *Lead I* é a diferença potencial entre os eletrodos posicionados na mão direita e mão esquerda.

Neste estudo de caso, registros de ECG com duração de 10 segundos (i.e., intervalos de 5000 amostra) a partir do registro denominado *Person_01rec_1* disponível na base de dados PHYSIONET ECG-ID. O registro de ECG selecionado é ilustrado na Figura 5.17. Note que o sinal é composto por formatos de onda características do ECG e ruídos de alta e baixa frequência, e possui uma faixa nominal de $\pm$10mV.

Primeiramente, o parâmetro de entrada `tipoSistema` foi definido com a constante `monitoraECG`. Quando esse parâmetro é definido, o modelo de referência é configurado como monitor cardíaco. Ou seja, são utilizado os coeficientes da função de transferência de filtros compostos pela combinação de filtros passa baixa, passa alta, e rejeita faixa com frequência de corte de 0.5 Hz, 40 Hz, e 50 Hz, respectivamente. A resposta de frequência ($H(j\omega)$) obtida com a banda passante 0.5 Hz e 40 Hz é ilustrada na Figura 5.18 utilizando um diagrama de bode.

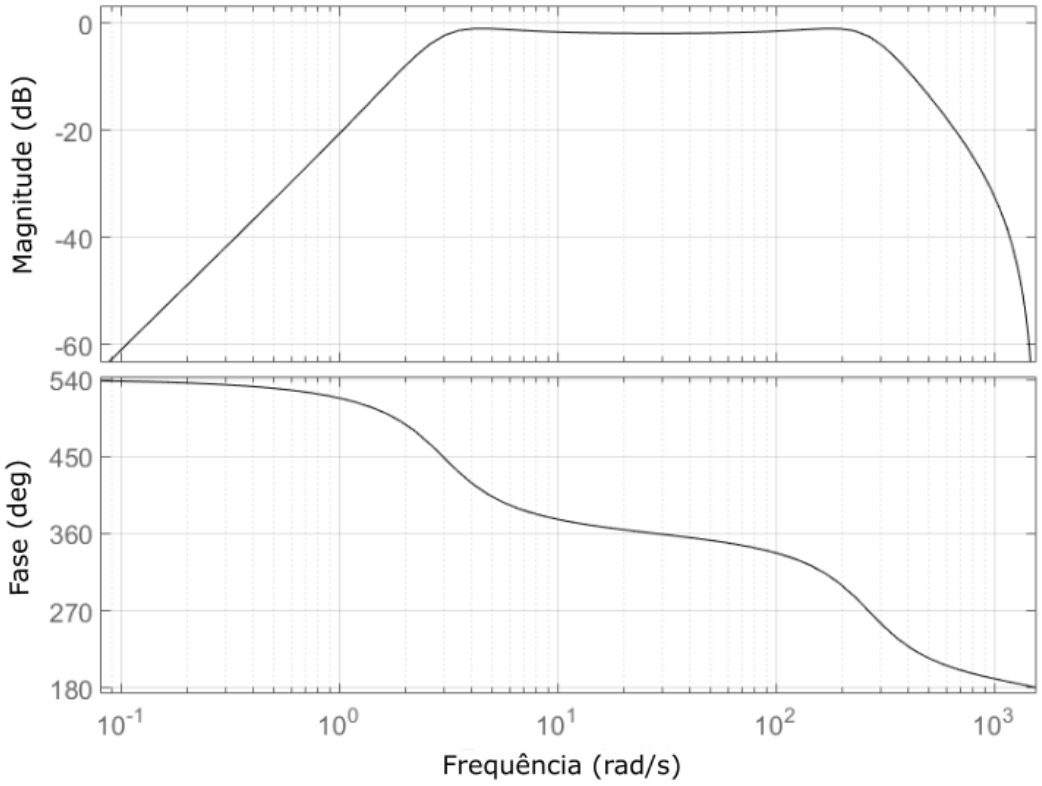


Figura 5.18: Resposta de frequência do filtro do modelo ($H(j\omega)$) configurado como monitor cardíaco.

A resposta de frequência obtida com os filtros especificados no estudo de caso está de acordo com a resposta de frequência da configuração de monitor cardíaco apresentada na

especificação (*data sheet*) do *single-lead*, monitor cardíaco *front end* AD8232 apresentada na Figura 5.19 (adaptada de [5]). Note que uma vez que os filtros definidos para o modelo de ECG possuem a mesma resposta de frequência do *front end*, é possível afirmar que o modelo de ECG é uma representação do comportamento deste tipo de equipamento usando uma configuração de monitor cardíaco.

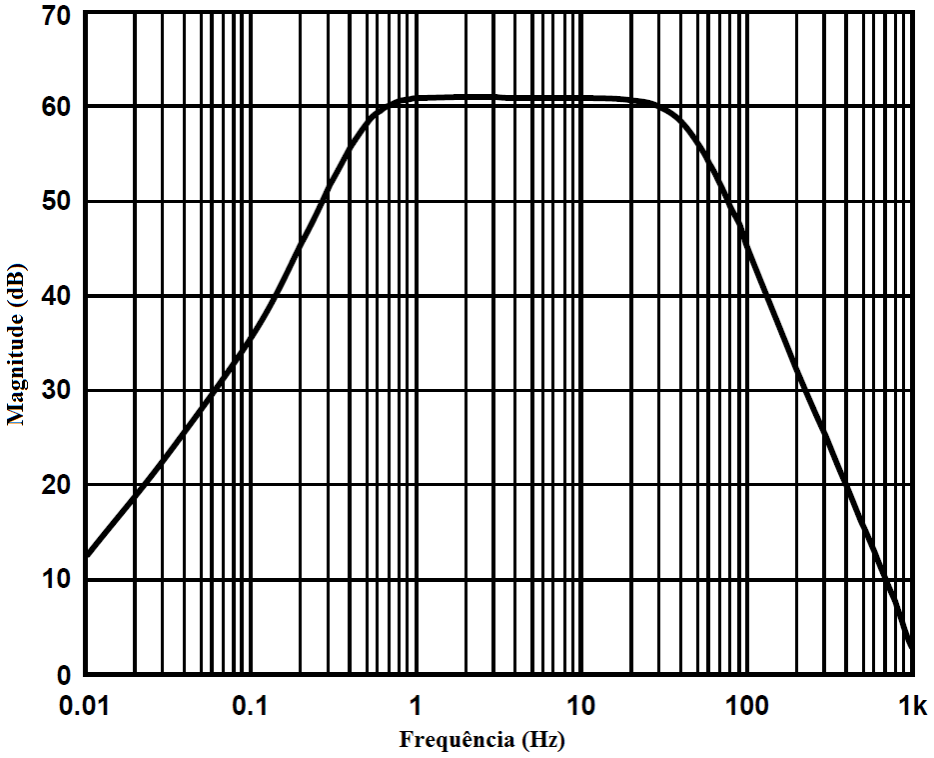


Figura 5.19: Resposta de frequência para a configuração de monitor cardíaco front end AD8232.

Simulações do modelo de ECG foram executadas e sinais de saída filtrados no domínio do tempo $x_f^{MO}(n)$ ($x_f(n)$) foram obtidos. Portanto, foi possível gerar o sexto exemplo de artefato de projeto e evidência para argumentar que o modelo de ECG é eficaz. Uma comparação no domínio da frequência entre o sinal de saída do modelo de ECG (i.e., sinal filtrado e apresentado no domínio do tempo $x_f^{MO}(n)$) e registros filtrados disponíveis na base de dados PHYSIONET ECG-ID foi realizada por meio da aplicação do algoritmo FFT (veja Figura 5.20). $|X(j\omega)|$ e $|X_f^{PH}(j\omega)|$ são a magnitude dos sinais não filtrados e dos sinais filtrados de ECG contidos na base de dados PHYSIONET ECG-ID. Por outro lado, $|X_f^{MO}(j\omega)|$ é a magnitude do registro de ECG filtrado obtido a partir do modelo de ECG.

Observe que os sinais ($|X_f^{MO}(j\omega)|$) obtidos com o modelo de ECG foram filtrados com qualidade tão boa quanto os sinais obtidos pelos filtros utilizados para eliminar os componentes de frequência indesejados (e.g., abaixo de 0.5 Hz) a partir do registro de ECG como destacado na Figura 5.20. Registros de ECG são frequentemente compostos por interferências de linha de energia residuais (entre 50 Hz e 60 Hz, dependendo de padrões regionais) que podem influenciar negativamente no formato da onda de ECG característica [69].

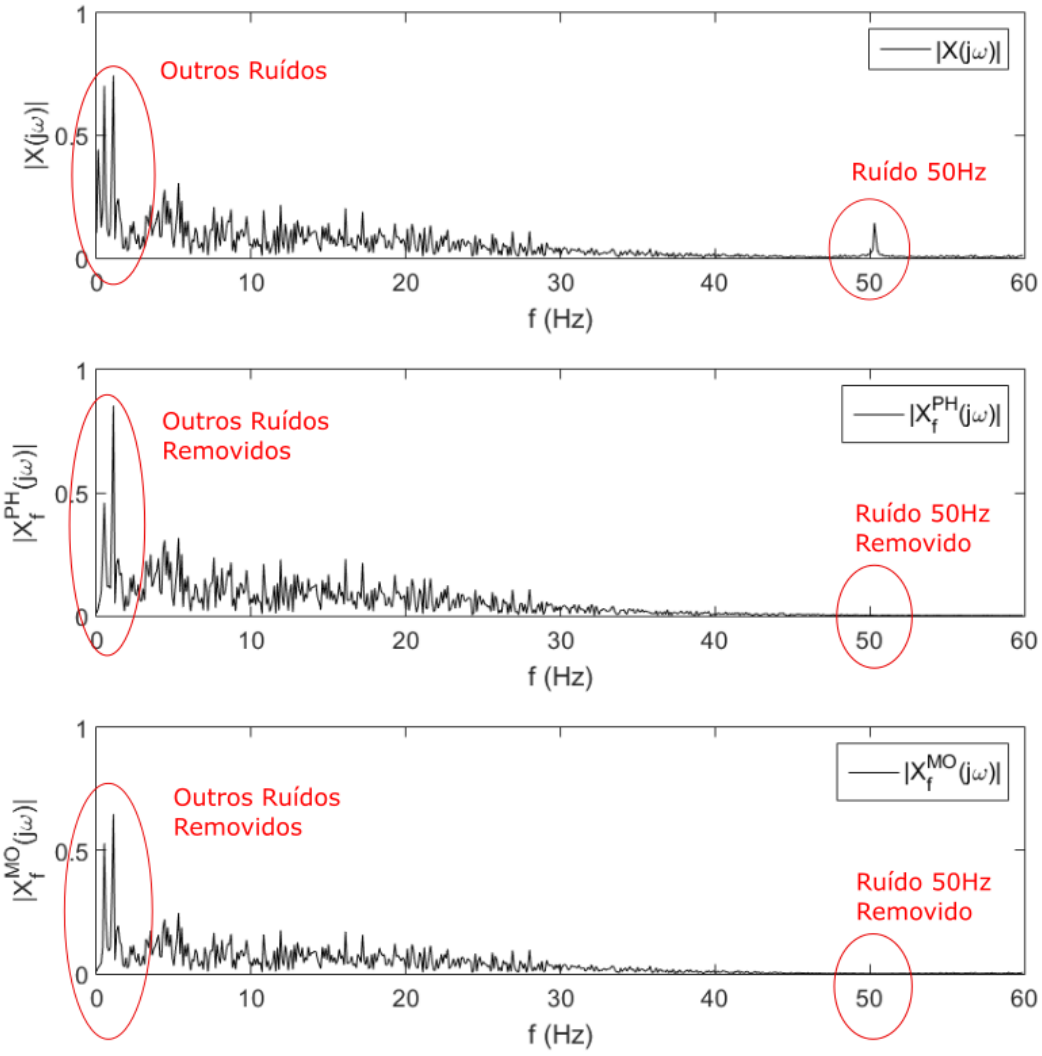


Figura 5.20: Comparação no domínio da frequência entre o modelo de ECG e registros disponíveis na base de dados PHYSIONET ECG-ID.

Os registros filtrados com o modelo de ECG e os registros disponíveis na base de dados PHYSIONET ECG-ID também foram analisados no domínio do tempo como o sétimo

exemplo de artefato de projeto e evidência para certificação. Registros do modelo de ECG e da base de dados são ilustrados na Figura 5.21. O registro de ECG completo de 10 segundos da base de dados PHYSIONET ECG-ID *Person_01rec_1* é composto por 11 pulsos cardíacos. Note que existem diferenças quando os registros são sobrepostos (linha pontilhada utilizada para os registros do modelo de ECG). Entretanto, isso não significa que o modelo não possui um bom desempenho. Portanto, a amplitude dos sinais foram normalizadas entre 0 e 1 e métricas de qualidade de desempenho foram utilizadas para complementar a análise no domínio da frequência apresentada na Figura 5.20..

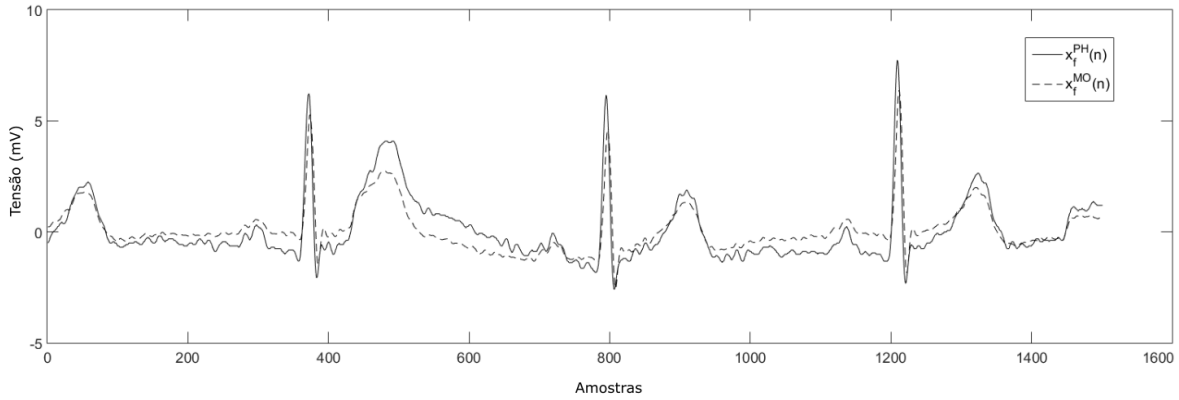


Figura 5.21: Amostra de comparação entre sinais filtrados usando o modelo de ECG e disponíveis na base de dados PHYSIONET ECG-ID.

Métricas de qualidade (desempenho) foram úteis para identificar que, a pesar das diferenças visuais entre os pulsos cardíacos parecerem relevantes, o erro quantitativo obtido entre os sinais foi baixo. Essas métricas são utilizadas para comparar erros entre sinais obtidos e sinais desejados [70]. Valores de erro quadrático médio (*Mean Squared Error - MSE*) e erro absoluto médio (*Root Mean Absolute Error - MAE*) foram calculados utilizando os 11 vetores de pulsos cardíacos do modelo de ECG (linha pontilhada) e registros filtrados da base de dados (linha contínua). A métrica MAE entre os registro do modelo e da base de dados é definida por:

$$\text{MAE}_{MO} = \frac{1}{N}\sum_{n=1}^{N}\left|x_f^{PH}(n) - x_f^{MO}\right| \tag{5.4}$$

onde N é o número de amostras do sinal, x_f^{MO} é o sinal filtrado obtido com o modelo de ECG, e x_f^{PH} é o sinal filtrado disponível na base de dados PHYSIONET ECG-ID. Por outro

lado, a métrica RMSE entre os registros do modelo e da base de dados é definida por:

$$\text{RMSE}_{MO} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[x_f^{PH}(n) - x_f^{MO}\right]^2} \tag{5.5}$$

onde os mesmos parâmetros utilizados para a métrica MAE_{MO} são aplicados para a métrica RMSE_{MO}. Note que RMSE_{MO} é o desvio padrão do erro quadrático médio (*Mean Squared Error - MSE*).

Os resultados calculados por meio de MAE_{MO} e RMSE_{MO} para os 11 vetores de pulsos cardíacos (i.e., $x_f^{MO}(n)$) e $x_f^{PH}(n)$) são apresentados na Tabela 5.4. Os resultados obtidos ao aplicar MAE_{MO} (média = 0.0543 mV) e RMSE_{MO} (média = 0.0765 mV) são considerados como esperados porque foram analisados baseados na escala do eixo y dos sinais filtrados (i.e., entre 0 e 1).

Tabela 5.4: Resultados obtidos com as métricas MAE_{MO} e RMSE_{MO}*

Pulso Cardíaco	MAE_{MO}	RMSE_{MO}
1	0.0350	0.0616
2	0.0364	0.0640
3	0.0890	0.1039
4	0.0491	0.0742
5	0.0751	0.0900
6	0.0386	0.0640
7	0.0995	0.1153
8	0.0367	0.0608
9	0.0593	0.0781
10	0.0357	0.0600
11	0.0429	0.0700
Mean	0.0543	0.0765

* Escala em milivolts

Por exemplo, um RMSE_{MO} igual a 0.0600 mV (vetor de pulso cardíaco 10) é um bom resultado porque significa que não foram obtidos erros significativos entre as amostras analisadas. Note que os sinais filtrados ilustrados na Figura 5.21 correspondem aos vetores

de pulso cardíaco 6, 7 e 8 apresentados na Tabela 5.4. Portanto, é possível argumentar que foi obtido um bom desempenho com o modelo de ECG, quando comparado com os registros de ECG filtrados disponíveis na base de dados PHYSIONET ECG-ID, porque foi observado que a média de erros tendeu a zero e os erros entre as amostras não foram significativos.

5.3 Requisitos do padrão ISO 14971

O padrão ISO 14971, relacionado ao processo de gerenciamento de equipamentos médicos, foi escolhido como cenário de aplicação da etapa *Requisitos de Padrões* definida no método. O processo de gerenciamento de risco foi especificado formalmente baseado em entrevistas com um especialista no desenvolvimento de sistemas médicos e no documento no qual o padrão é descrito [25]. Além disso, um exemplo de utilização da especificação para conduzir o gerenciamento é apresentado.

5.3.1 Especificação Formal

Etapas de análise de risco, avaliação de risco, controle de risco, e avaliação de risco residual geral foram especificadas como uma rede de Petri colorida hierárquica do processo de gerenciamento de risco. A representação do modelo CPN é ilustrada na Figura 5.22. A representação completa não é apresentada porque o foco neste documento está na utilização de resultados de simulação e verificação de modelos para aumentar a confiança na realização correta do processo de gerenciamento de risco baseado no padrão ISO 14971. O restante dos modelos é descrito detalhadamente por Neto [57] e Sobrinho et al. [81], juntamente com atividades de verificação (*model checking*) e validação (simulações) de modelos. Esta atividade foi desenvolvida de acordo com os requisitos descritos no método proposto neste trabalho. Na próxima seção são apresentados alguns resultados de validação para exemplificar como fabricantes de sistemas embarcados críticos de segurança podem gerar evidências de processo durante o processo de certificação.

Validação do Modelo do Padrão ISO 14971

Simulações foram utilizadas para validar o modelo CPN do processo de gerenciamento de risco. Várias simulações foram definidas considerando diferentes marcações iniciais

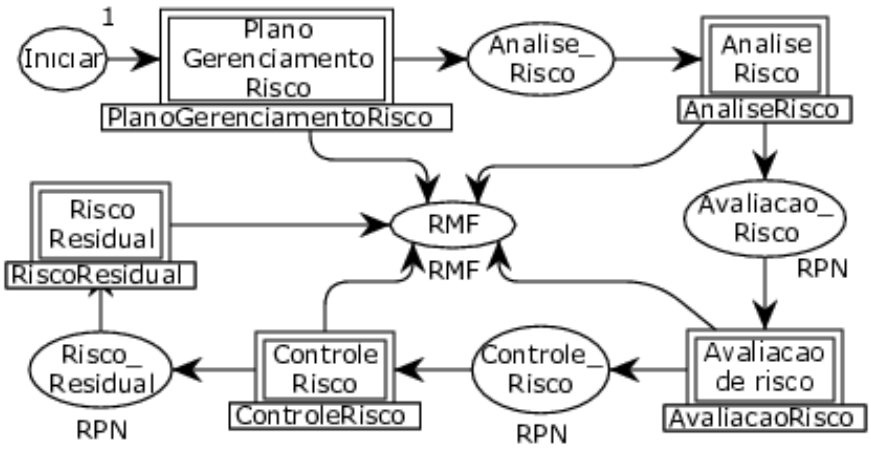


Figura 5.22: Representação do modelo do processo de gerenciamento de risco.

com o *software* CPN/Tools. Um especialista no desenvolvimento de sistemas médicos acompanhou as simulações para verificar se as especificações estão em conformidade com os requisitos da ISO 14971. O intuito foi verificar se os requisitos definidos para as atividades de gerenciamento de risco foram satisfeitos. Por exemplo, é necessário assegurar que valores de severidade, frequência, e detecção foram definidos, e se o número de prioridade de risco (*Risk Priority Number - RPN*) foi calculado durante a atividade de estimativa de risco.

Uma amostra de uma das simulações do módulo de estimativa de risco é ilustrada na Figura 5.23. Nesta simulação, as marcações iniciais foram associadas aos lugares `Estima` (1‘1++1‘2++1‘3++1‘4++), `Avalia Severidade` (1‘5), `Avalia Frequencia` (1‘3), e `Avalia Detecta` (1‘5). Note que a marcação do modelo CPN foi alterada após os passos de simulação. Os valores foram definidos (de 1 até 5) usando uma função aleatória de números inteiros de acordo com a especificação do plano de gerenciamento de risco. Portanto, novos valores de severidade, frequência e detecção foram gerados com o modelo para o risco com número de identificação 2 (ficha contida no lugar `Detecta`), que foram enviados para o lugar `RMF`. Isso significa que valores para calcular o RPN são sempre gerados antes da atividade de avaliação no modelo do processo de gerenciamento de risco. Um RPN é calculado por meio da multiplicação de valores de severidade, frequência e detecção.

Resultados de simulação da atividade estimativa de risco são usados para avaliar os riscos. Uma amostra da simulação do módulo de avaliação é apresentada na Figura 5.24. Marcações iniciais desse módulo estão associados com os lugares `Avalia` (valores

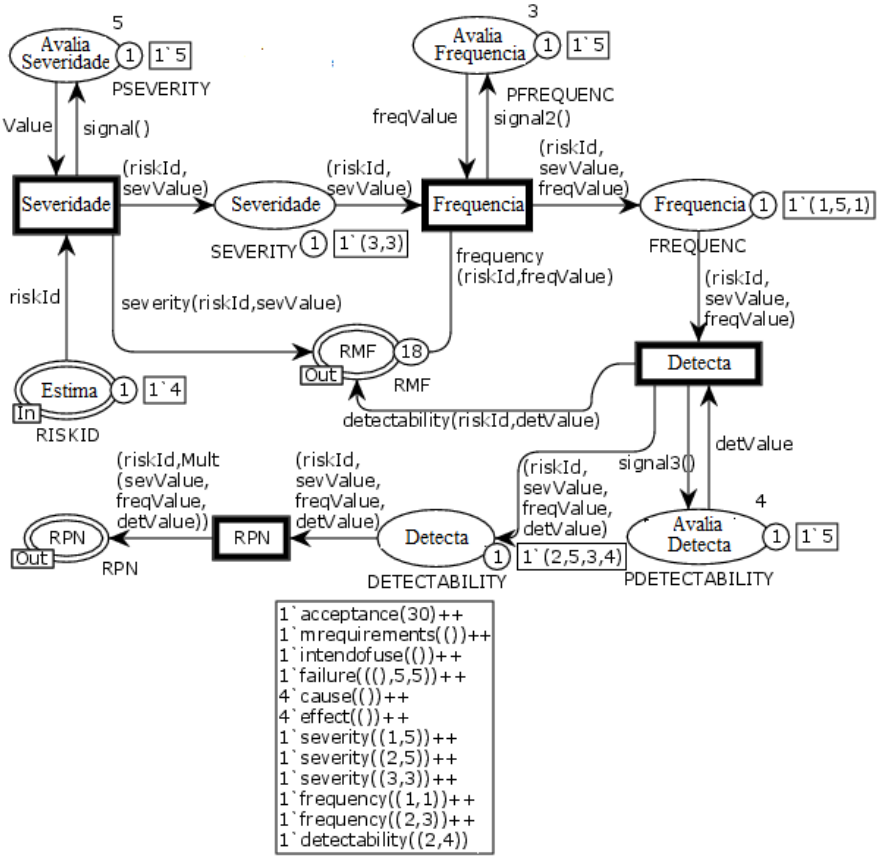


Figura 5.23: Amostra de simulação do módulo de estimativa de risco.

de identificador de risco e RPN relacionados com o módulo de estimativa de risco - 1‘(1,9)++1‘(3,15)++1‘(2,32)) e `Limite Risco` (1‘30). Esse módulo é usado para avaliar a aceitabilidade de riscos baseado no valor de RPN e no limite superior de aceitação definido no plano de gerenciamento de risco.

Avaliações são realizadas pela comparação entre valores de RPN e limites superiores de aceitação. Note que após a avaliação de risco (transição `Avaliar`), o risco 1‘(2,32) foi classificado como inaceitável (lugar `Acima Limite`), e os riscos 1‘(1,9) e 1‘(3,15) foram classificados como aceitáveis (lugar `Risco Aceito`). Os riscos aceitos são enviados para o lugar `RMF`. Isso significa que os riscos foram classificados corretamente e somente os riscos aceitos foram registrados no arquivo de gerenciamento de riscos.

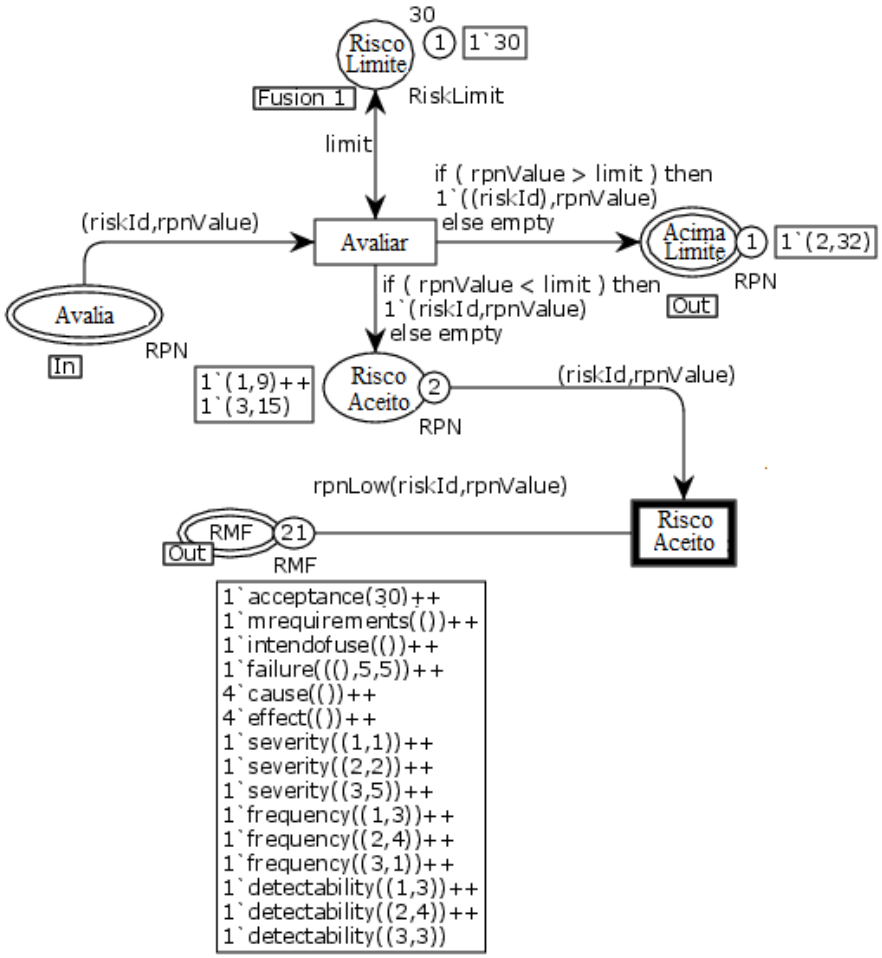


Figura 5.24: Amostra de simulação do módulo de avaliação de risco.

5.3.2 Processo de Gerenciamento de Risco de um Sistema de ECG

O processo de gerenciamento de risco de um sistema de ECG foi realizado com base no modelo CPN especificado. Um plano de gerenciamento de risco foi definido incluindo as etapas de análise de risco, avaliação de risco, controle de risco, e avaliação de risco residual geral. Além disso, critérios de aceitação de risco foram estabelecidos baseados no número de prioridade de risco utilizando a técnica análise de criticidade e modo de efeito de falhas (*Failure, Mode, Effects and Criticality Analysis - FMECA*) em um plano de gerenciamento de risco. Um RPN foi obtido a partir do produto dos valores máximos de severidade, ocorrência, e detecção. Estes valores foram especificados em uma escala de 1 a 5, resultando em um RPN de 125. Um limite superior para aceitação de risco definido com valor de 20 por cento do RPN. Portanto, os critérios de aceitação de risco foram:

- riscos são somente aceitáveis com RPN < 25;
- riscos com RPN de 25 até 31 são toleráveis, caso benefícios superem os riscos;
- riscos com RPN > 31 são intoleráveis.

Riscos foram identificados utilizando entrevistas com um especialista, revisões da literatura, e verificações de sistemas similares. A técnica FMECA e entrevistas baseadas nos questionários descritos no anexo C da ISO 14971 foram utilizados para auxiliar na identificação dos riscos. Dezenove falhas foram identificadas e registradas em uma matriz FMECA. Uma amostra da matriz FMECA é apresentada na Tabela 5.5.

Tabela 5.5: Amostra da matriz FMECA para identificação de riscos.

Componente	**Função**	**Modo Potencial de Falha**	**Efeito Potencial de Falha**	**Causa Potencial**
Microcontrolador	Conversão A/D	Conversão ruim	Perda na qualidade do sinal	Hardware com defeito
Bluetooth	Envio de dados	Desvaneciamento e terminal oculto	Perda de dados	Barreiras físicas e distância
Bateria	Suprimento de energia	Nível de bateria baixo	Sinal com erro	Erro em software
Eletrodos	Aquisição de sinais	Posicionamento incorreto	Sinal com erro	Operador destreinado

Uma classificação baseada no nível de prioridade utilizando o RPN foi definida após a identificação das falhas. Riscos relacionados com a qualidade do sinal foram definidos com valor de severidade 5 porque interferências no sinal afetam negativamente toda a operação do sistema. O valor mais alto de RPN obtido na classificação foi relacionado com a *Blindagem do Cabo* com as falhas *Isolamento elétrico ruim* e *Defeito na blindagem do cabo*. Os níveis de severidade (5-desastroso), detecção (5-muito baixa), e ocorrência (4-provável) foram os mesmos para as duas falhas (RPN = 100). Falhas com RPN mais altos receberam prioridade durante o restante do processo de gerenciamento de risco.

O limite para aceitação de risco de 20% do RPN definido no RMP é utilizado durante a etapa de avaliação de risco. Caso um risco esteja acima do limite, é necessário iniciar a etapa de controle de risco. Caso contrário, é somente necessário registrar o risco no AGR. Os

riscos *Conversão A/D ruim* (RPN = 5) e *Cabos com qualidade ruim* (RPN = 10) obtiveram os valores de RPN abaixo do limite de aceitação. A etapa de controle de risco foi aplicada no restante dos riscos.

O controle de risco foi realizado baseado na classificação de RPN. Medidas de controle de risco foram analisadas e adotadas para reduzir os riscos. Entretanto, novos riscos podem ser introduzidos ao aplicar a redução de riscos. Portanto, a severidade, ocorrência, e detecção devem ser definidas novamente para assegurar que os riscos foram reduzidos e que nenhum novo risco foi introduzido. Uma amostra da matriz FMECA relacionada ao controle de riscos é apresentada na Tabela 5.6. É possível verificar que o risco *Posicionamento incorreto* possui RPN de 45. Isto significa que o posicionamento incorreto de eletrodos é uma falha inaceitável que deve ser reduzida. Assim, o risco foi reduzido diminuindo a severidade (S = 5), ocorrência (O = 3), e detecção (D=3). Medidas de controle de risco foram definidas analisando causas potenciais e fontes de falhas. Por exemplo, a medida de controle para o posicionamento incorreto de eletrodos foi o treinamento de usuários. Porém, a redução do risco (RPN de 45 para 30) não foi suficiente (limite de aceitação de 25).

Tabela 5.6: Amostra da matriz FMECA para controle de riscos.

Modo Potencial de Falha	**S**	**O**	**D**	**N**	**M**	**S**	**O**	**D**	**N**
Posicionamento Incorreto	5	3	3	45	Treinar	5	2	3	30
Configuração do Amp. Op. Incorreta	5	5	2	50	Corrigir Projeto	5	2	3	30
Funcionamento do Amp. In. Incorreto	5	3	2	30	Testar	5	1	1	5

Quatro riscos residuais foram identificados durante o processo de gerenciamento de risco. Três riscos não foram mitigados durante a etapa de controle de risco. O risco *Posicionamento incorreto* é um exemplo de risco residual. De acordo com o padrão ISO 14971, é necessário analisar se os benefícios médicos superam o risco de posicionamento incorreto de eletrodos. Neste contexto, os benefícios médicos superaram o risco porque sistemas de aquisição de sinais biomédicos são importantes ferramentas no diagnóstico e tratamento de pacientes com, por exemplo, doenças cardiovasculares e desordens gástricas. O restante dos riscos foi também definido como aceitável. Além disso, o posicionamento incorreto de eletrodos

pode ser verificado por sistemas de *software* (modelos apresentados anteriormente).

5.4 Casos de Garantia

Esta seção está relacionada com a implementação da atividade `Casos de Garantia` definida no método apresentado no Capítulo 4. A modelagem de casos de garantia utilizando GSN foi útil para representar argumentações sobre propriedades de segurança e eficácia do sistema de ECG, e o relacionamento entre argumentos e evidências (de produto e processo) durante o estudo de caso.

Argumentações definidas no caso de garantia são suportadas por artefatos de projeto gerados durante as atividades `Requisitos de Padrões` e `Requisitos do Produto` por meio de conexões entre evidências e afirmações (metas). Note que o caso de garantia deve ser criado de acordo com o padrão para a representação e compartilhamento de casos de garantia (*Assurance Cases Exchange Standard - ACES*) definido como parte do método proposto. Afirmações, argumentos, e evidências foram estruturadas em módulos GSN associados com as atividades `Requisitos de Padrões` e `Requisitos do Produto`.

A representação mais abstrata do modelo de caso de garantia em GSN para o sistema de ECG é apresentado na Figura 5.25. O elemento `SISTEMA-G1` (retângulo) é utilizado para representar a meta que contém uma afirmação sobre a eficácia e segurança do sistema em desenvolvimento. Esta meta está relacionada aos contextos de *software* (`SISTEMA-C1`) e *hardware* (`SISTEMA-C2`). Isso significa que as argumentações construídas para suportar a meta principal é composta por argumentos sobre a conformidade com requisitos de *software* e *hardware*. Os argumentos que suportam a meta são estruturados baseados em estratégias de argumentações sobre o processo de gerenciamento de riscos do sistema e propriedades específicas do produto (estratégias `SISTEMA-S1` e `SISTEMA-S2`, respectivamente). Argumentos são estruturados em módulos relacionados com cada uma das estratégias (`SISTEMA-M1` e `SISTEMA-M2`).

Por exemplo, a meta principal (`SISTEMA-G1`) foi definida de acordo com a especificação do padrão ACES ilustrada na Figura 5.26. Note que a especificação é composta por um conjunto de elementos da linguagem de marcação XML. O modelo GSN do sistema

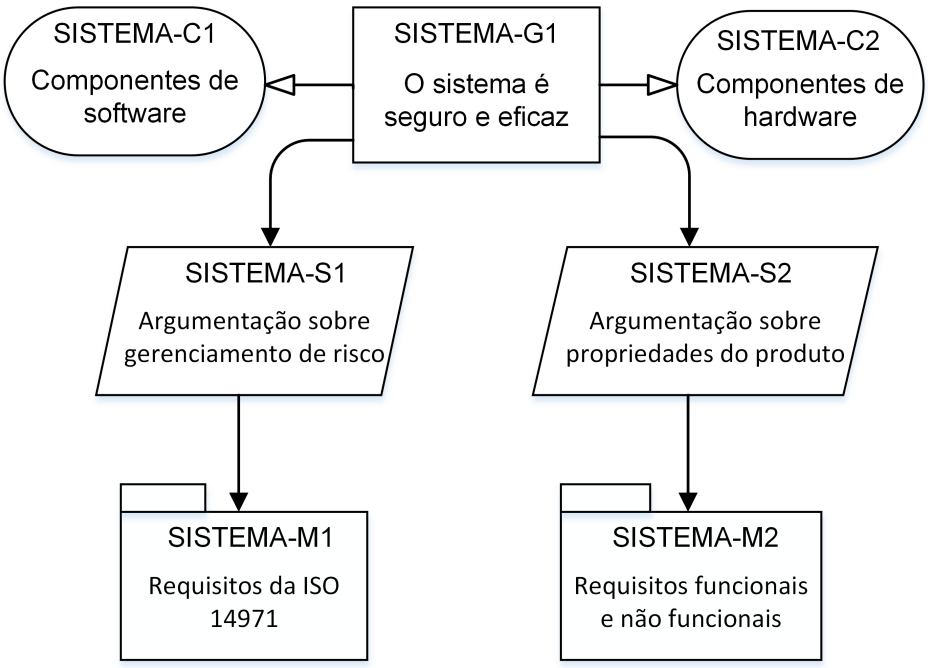


Figura 5.25: Modelo GSN do sistema de aquisição.

de aquisição é representado pela definição do elemento meta principal (elemento `<goal>`) e o seu relacionamento (elemento `<relationships>`) com os outros elementos contidos no modelo GSN (Figura 5.25). Dois componentes de contexto (elemento `<context>`) e estratégia (elemento `<strategy>`) são utilizados para contextualizar e planejar a estrutura do caso de garantia.

É importante destacar que a definição de cada um dos elementos de um módulo é realizada por agrupamentos com o elemento `<group>`. O módulo principal é representado com o elemento `<parentArgument>`, enquanto que os módulos `SISTEMA-M1` e `SISTEMA-M2` são representados com o elemento `<childArgument>`. Cada um desses módulos são compostos por seus elementos ACES `<group>` para cada componente GSN. Por questão de espaço e legibilidade, a definição completa de todos os elementos GSN associados com os módulos e a definição principal do documento ACES com os elementos `<assuranceCase>` e `<Device>` foram omitidas.

O restante dos modelos de casos de garantia estruturados com GSN (i.e., argumentos filhos), apresentados nos próximos parágrafos, seguem a especificação ACES (XML) descrita no Capítulo 4. Portanto, somente a representação gráfica em GSN são apresentadas para exemplificar e relacionar argumentos com os artefatos de projeto e evidências geradas

```
<parentArgument>
   <childArgument>
       <group type="goal">
          <goal id="ISO14971-G1">
              <description> Gerenciamento de risco em conformidade com a ISO 14971 </description>
          </goal>
       </group>
   </childArgument>
   <childArgument>
       <group type="goal">
          <goal id="PRODUTO-G4">
              <description> Modelo formal em conformidade com especificação </description>
          </goal>
       </group>
   </childArgument>
   <group type="context">
      <context id="SISTEMA-C1">
         <description> Componentes de software </description>
      </context>
      <context id="SISTEMA-C2">
         <description> Componentes de hardware </description>
      </context>
   </group>
   <group type="strategy">
      <strategy id="SISTEMA-S1">
         <description> Argumentação sobre gerenciamento de risco </description>
         <relationships>
         <relationSupportedBy id="r5" type="goal" relId="ISO14971-G1"/>
         </relationships>
        </strategy>
        <strategy id="SISTEMA-S2">
           <description> Argumentação sobre propriedades do produto </description>
           <relationships>
           <relationSupportedBy id="r6" type="goal" relId="PRODUTO-G4"/>
           </relationships>
      </strategy>
     </group>
     <group type="goal">
        <goal id="SISTEMA-G1" type="first">
           <description> O sistema é seguro e eficaz </description>
           <relationships>
               <relationInContextOf id="r1" type="context" relId="SISTEMA-C1"/>
               <relationInContextOf id="r2" type="context" relId="SISTEMA-C2"/>
               <relationSupportedBy id="r3" type="strategy" relId="SISTEMA-S1"/>
               <relationSupportedBy id="r4" type="strategy" relId="SISTEMA-S2"/>
           </relationships>
        </goal>

     </group>
</parentArgument>
</assuranceCase>
```

Figura 5.26: Exemplo de especificação ACES para o modelo GSN do sistema de aquisição.

nas seções anteriores para o sistema de ECG (e.g., resultados de verificação e validação de modelos CPN).

Os argumentos relacionados ao módulo `ISO14971-M1` são apresentados na Figura 5.27. Este módulo contém argumentos referentes ao processo de gerenciamento de risco baseado no padrão ISO 14971. A meta principal é uma afirmação sobre a conformidade do fabricante com requisitos especificados com o padrão no contexto de requisitos de segurança

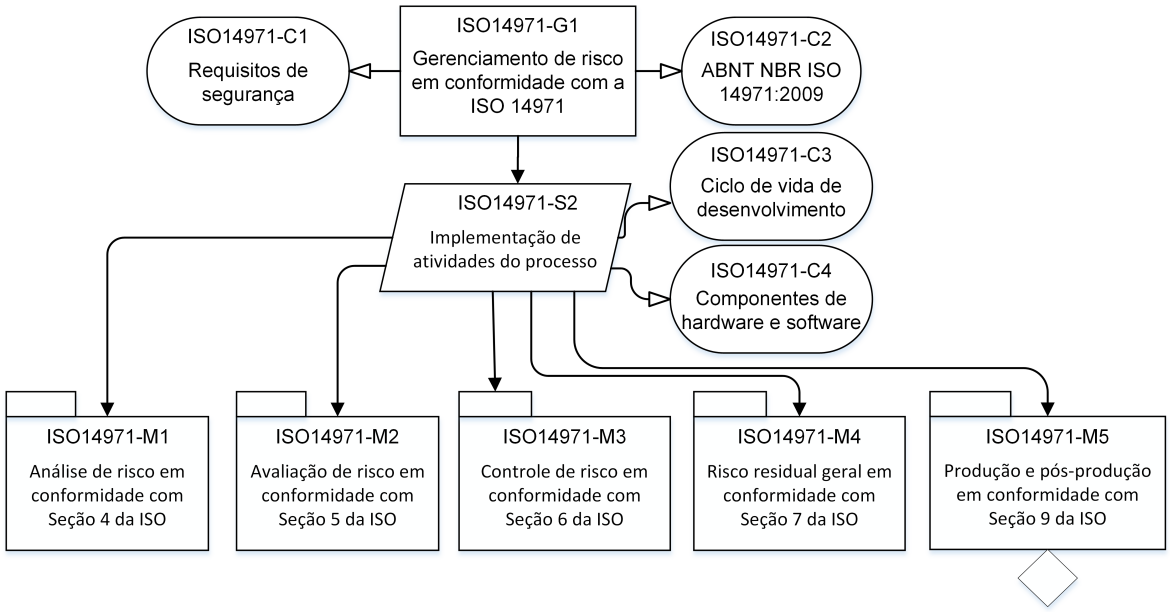


Figura 5.27: Modelo GSN da ISO 14971.

e do padrão ISO 14971. Estes argumentos também contém outros módulos com argumentos específicos sobre cada atividade do processo de gerenciamento de risco. Atividades incluem a análise de risco, avaliação de risco, controle de risco, risco residual geral, e produção e pós-produção.

O módulo relacionado com a atividade de análise de risco é iniciado com uma afirmação sobre a conformidade em sua realização (veja Figura 5.28). A argumentação é composta por conformidade na identificação de má utilização e utilização destinada; identificação de perigos em condições normais e de falha; e estimativa de risco para situações perigosas. Evidências utilizadas incluem o anexo C do padrão ISO 14971, experiências clínicas, entrevistas com especialistas, revisões bibliográficas, matriz FMECA coluna C (veja Tabela 5.5), e matriz FMECA coluna E (veja Tabela 5.6).

Argumentos sobre a realização correta da avaliação de risco são apresentados Figura 5.29. A meta principal nesta argumentação está relacionada com a conformidade do fabricante do sistema de ECG com os requisitos para a avaliação de risco. As avaliações de risco estão no contexto de situações perigosas e são realizadas de acordo com um número de prioridade de risco. Suporte para a argumentação de conformidade com a avaliação de risco é provido por meio de resultados obtidos com o cálculo de prioridade de risco (evidência `ISO14971-E7`).

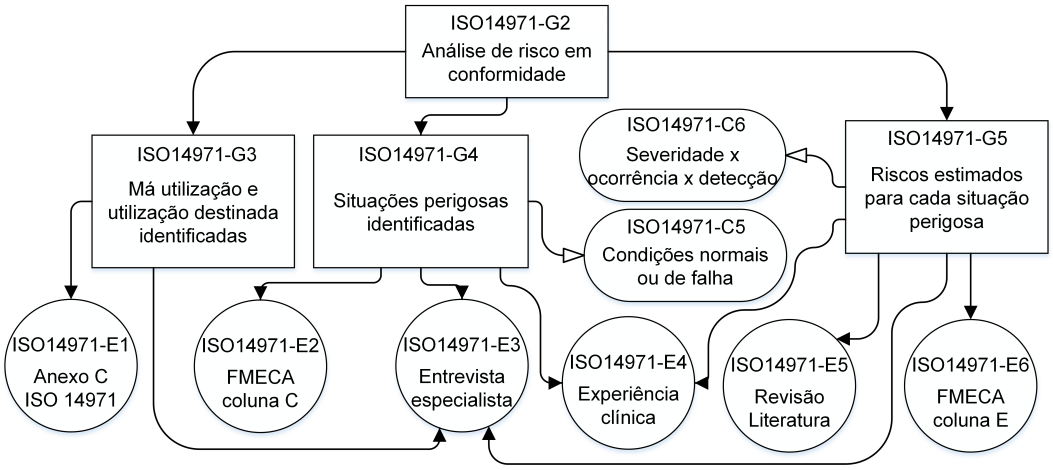


Figura 5.28: Modelo GSN da etapa de análise de risco.

O controle de risco (terceira etapa, Figura 5.30) deve ser realizado caso a redução do risco seja necessária (contexto `ISO14971-C9`). Os argumentos que suportam a afirmação de conformidade com o controle de risco são a identificação de medidas de controle de risco, verificação e implementação de medidas de controle, avaliação de riscos residuais, realização de análise de risco e benefício, e análise de efeitos de medidas de controle. Estas argumentações são suportadas por medidas descritas na ISO 14971, AGR, e análises da literatura (evidências).

A quarta etapa do processo de gerenciamento de risco (módulo `ISO14971-M4`) é relacionada com a avaliação de aceitabilidade de risco residual geral (veja Figura 5.31). Esta etapa é realizada com base no RMP. A afirmação de que os benefícios médicos superam os riscos residuais provê suporte para a argumentação. A análise de dados e literatura é a evidência que suporta estas afirmações. A quinta etapa do processo de gerenciamento de

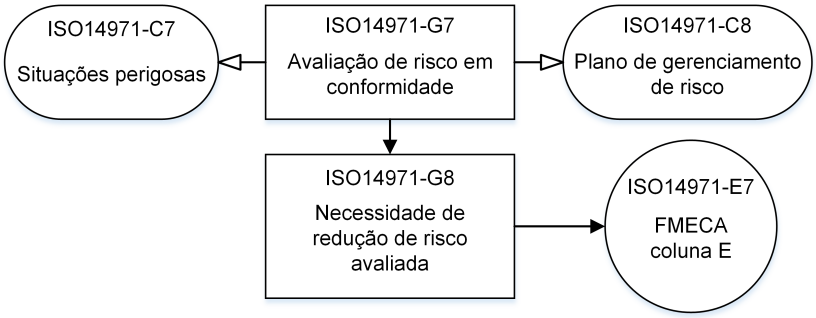


Figura 5.29: Modelo GSN da etapa de avaliação de risco.

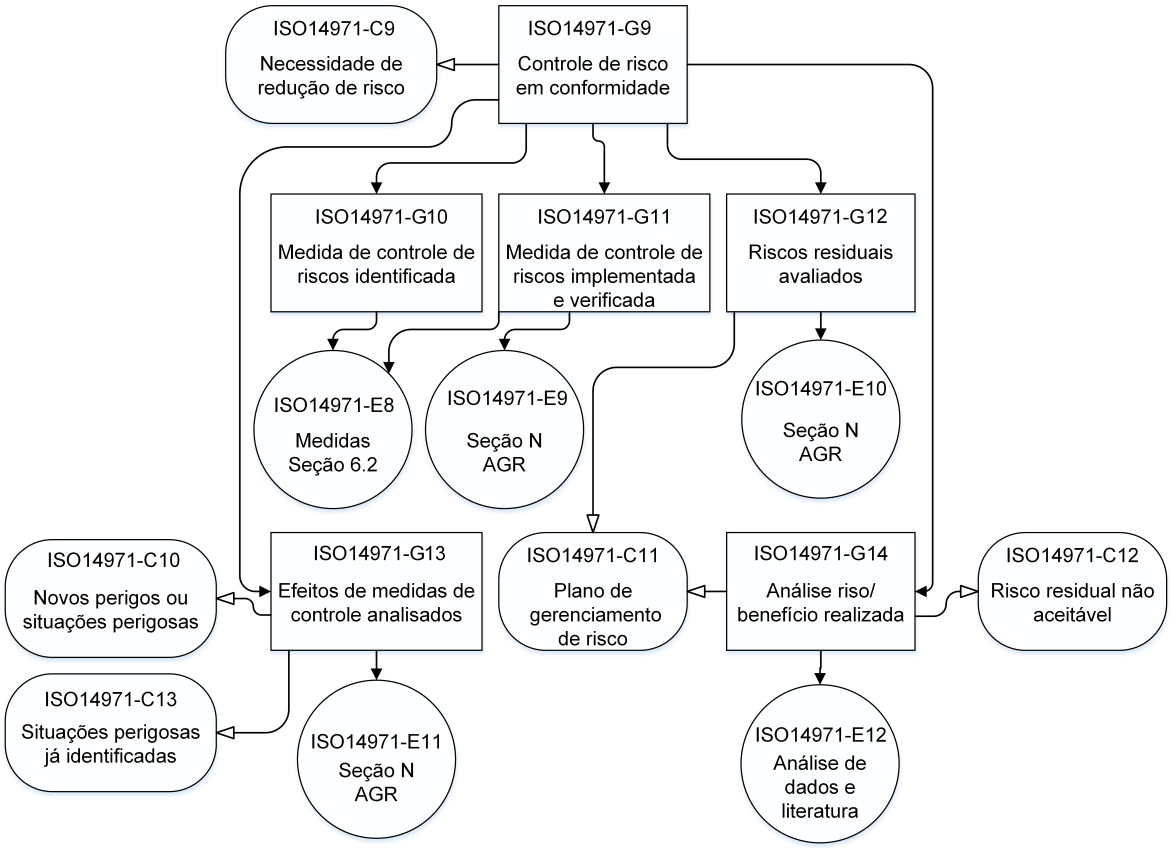


Figura 5.30: Modelo GSN da etapa de controle de risco.

risco (veja Figura 5.27, módulo `ISO14971-M5`) está definida no caso de garantia como ainda não desenvolvida (losango) por estar fora do escopo deste trabalho.

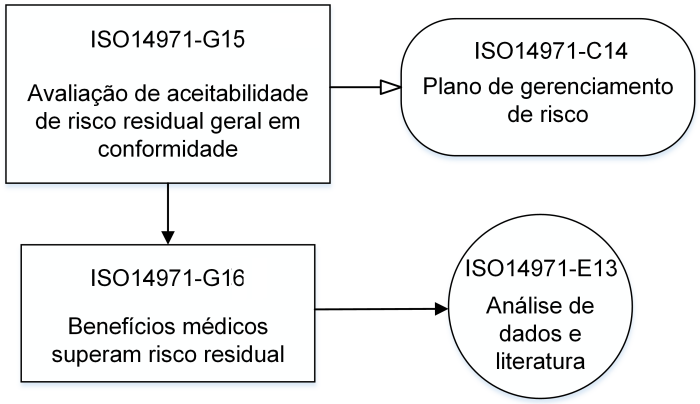


Figura 5.31: Modelo GSN da etapa de avaliação de risco residual geral.

Além de argumentos de processo, características relacionadas ao produto também são representadas com GSN. Argumentos e evidências são apresentados no módulo

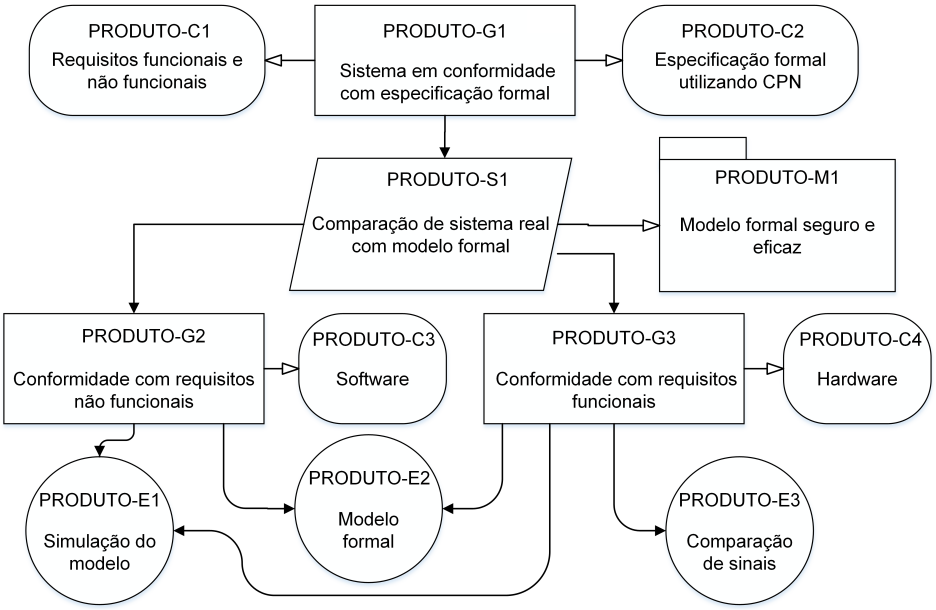


Figura 5.32: Modelo GSN de requisitos do sistema de aquisição.

`SISTEMA-M2` (veja Figura 5.25) ilustrado na Figura 5.32. O caso de garantia foi desenvolvido baseado na especificação em linguagem natural e no modelo formal do sistema de ECG descritos nas seções anteriores. Este módulo é iniciado com a afirmação de conformidade do sistema com a especificação formal. A estratégia utilizada para as argumentações é a comparação do sistema real com o modelo formal. Nesta comparação é argumentado a conformidade com requisitos funcionais e não funcionais no contexto de *hardware* e *software*. O modelo formal (veja Figura 5.5), resultados de simulações, e resultados de comparações entre modelo e protótipo do sistema em desenvolvimento podem ser utilizados como evidências que suportam as argumentações.

Entretanto, é necessário realizar argumentações sobre a segurança e eficácia do modelo formal para garantir a confiança na comparação entre protótipo do sistema real e modelo formal. São necessárias argumentações sobre a conformidade do modelo formal com a especificação em linguagem natural e modelos semiformais. Este módulo do caso de garantia (veja Figura 5.32, módulo `M1`) é ilustrado na Figura 5.33. As estratégias utilizadas para essa argumentação foram as verificações de propriedades de segurança (*software*), eficácia (*hardware*), e de características gerais do modelo formal (e.g., *deadlocks*). As argumentações utilizadas foram a conformidade com valores de impedância eletrodo-pele, situação de bateria, eficácia na aquisição de sinais, e conformidade com características

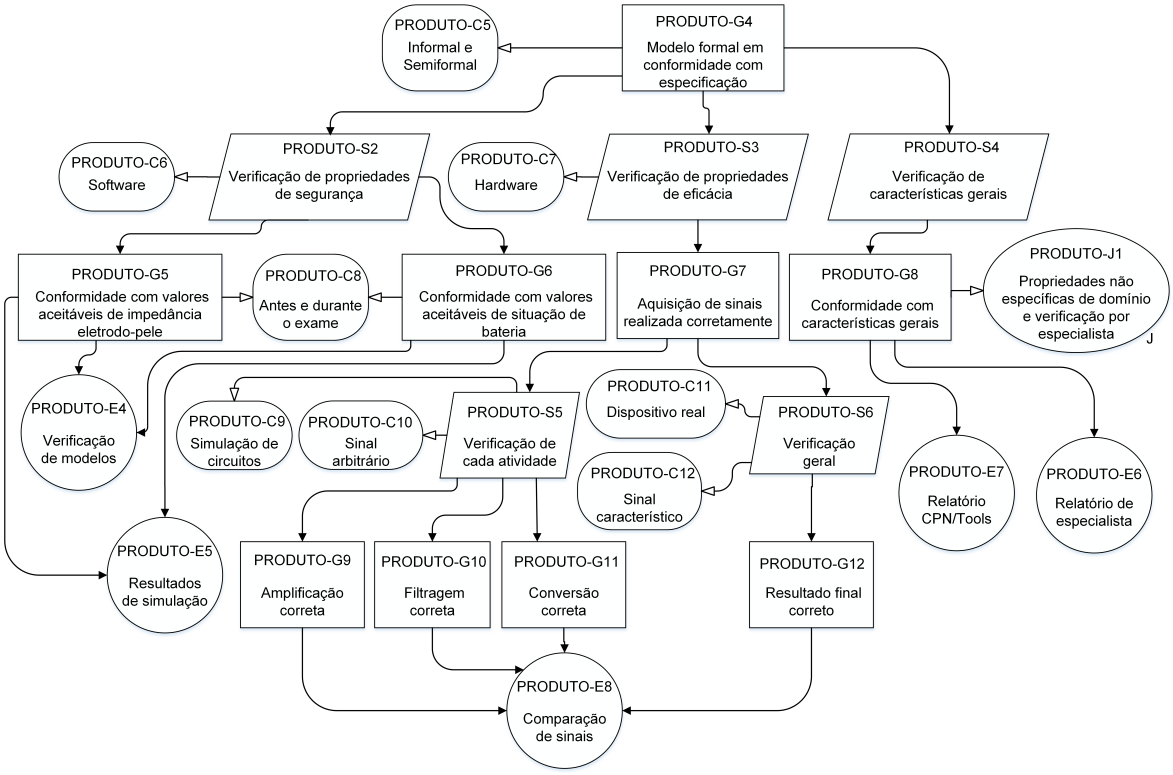


Figura 5.33: Modelo GSN da especificação formal do sistema de aquisição.

gerais. Resultados de verificação de modelos (veja Figura 5.2.3), simulação, comparação entre sistema e modelo (veja Figura 5.20 e Tabela 5.4), relatórios de especialistas, e relatório do CPN/Tools (veja Tabela 5.1) podem ser utilizados como evidências.

5.5 Sumário do Capítulo

Neste capítulo foram apresentados resultados de um estudo de caso sobre sistemas de aquisição de sinais biomédicos. O objetivo com o estudo de caso foi avaliar experimentalmente o método descrito no Capítulo 4 e apresentar um cenário de uso que fabricantes de sistemas embarcados críticos de segurança podem se basear para aplicar o método. A técnica de verificação de modelos, CPN, o padrão ISO 14971, casos de garantia, e GSN foram utilizados durante o estudo de caso para implementar as atividades definidas no método.

Bons resultados de verificação e validação foram obtidos com os modelos ECG criados em conformidade com a atividade `Requisitos do Produto`. Por outro lado, a criação

o modelo formal do processo de gerenciamento de risco em conformidade com a atividade `Requisitos de Padrões` contribuiu para a realização das etapas definidas no processo de maneira mais confiável. A implementação da atividade `Casos de Garantia` foi útil para interligar os resultados obtidos durante as outras atividades e gerar argumentos relacionados com a segurança e eficácia do sistema de ECG.

Resultados relacionados com o estudo de caso foram publicados em alguns eventos acadêmicos. O modelo de referência de sistemas de aquisição de sinais biomédicos e casos de garantia GSN construídos são apresentados por Sobrinho et al. [80]. Resultados iniciais associados com a especificação de modelos de ECG foram descritos por Sobrinho et al. [82]. Um simulador de ECG utilizado durante atividades iniciais de validação do modelo de ECG é apresentado por Sobrinho et al. [79]. Por fim, a modelagem e verificação do processo de gerenciamento de risco são também apresentadas por Sobrinho et al. [81].

Além disso do estudo de ECG, a especificação formal de um sistema de Eletrogastrografia (EEG) é apresentada por Sobrinho et al. [83] como resultados complementares. Um simulador de ECG também é apresentado por Cunha et al. [16]. É recomendável a leitura dos artigos descritos em [83] e [16] para mais informações sobre os resultados complementares.

Capítulo 6

Considerações Finais

Sistemas embarcados são geralmente sistemas críticos de segurança. Problemas neste tipo de sistema podem resultar em riscos catastróficos para seres humanos. Agências reguladoras definem requisitos para aprovar sistemas antes de sua comercialização. A maioria das agências reguladoras utiliza padrões prescritivos, como, por exemplo, a ISO 14971 [25], durante o processo de certificação atual (processos prescritivos de certificação). Por outro lado, existe um processo de certificação baseado em metas onde são realizadas avaliações de características relacionadas ao produto em desenvolvimento [85].

Fabricantes devem utilizar técnicas que diminuam problemas no projeto de sistemas embarcados críticos de segurança. A aplicação de técnicas tais como, métodos formais, pode aumentar a confiança em seu funcionamento correto, e demonstrar para agências reguladoras que sistemas são seguros e eficazes. Documentação inequívoca e bem definida é gerada utilizando métodos formais. Técnicas para verificação de propriedades de segurança, como, por exemplo, a verificação de modelos (*model checking*) [47], podem ser aplicadas em modelos formais para aumentar a confiança na especificação de sistemas. Fabricantes podem também utilizar modelos de *hardware* durante comparações de resultados entre a simulação de modelos e saídas de sistemas reais para verificar a eficácia do modelo formal. Resultados obtidos com a modelagem, verificação, e validação são úteis para auxiliar fabricantes a aprovar sistemas durante o processo de certificação. Agilidade no processo de certificação resulta na entrada de sistemas mais cedo no mercado.

Apesar dos recentes avanços em pesquisas relacionadas com a segurança e eficácia de sistemas embarcados críticos de segurança, ainda existem problemas com estes

sistemas no mercado. É necessário realizar verificações e validações em fases iniciais de desenvolvimento para evitar a identificação de defeitos somente em protótipos do sistema. Identificação de defeitos em protótipos pode resultar em modificações no projeto e, consequentemente, no desenvolvimento de uma nova versão do sistema físico. Isso aumenta os custos de desenvolvimento e o tempo para entrada de sistemas no mercado.

Neste documento foi apresentado um método para auxiliar fabricantes de sistemas embarcados críticos de segurança durante o desenvolvimento e certificação integrando os processos de certificação prescritivo e baseado em metas. O número elevado de notificações relacionadas com sistemas médicos, e registradas pela Administração de Alimentos e Drogas (*Food and Drug Administration - FDA*) [3], é um exemplo de indicador que pode ser utilizado para justificar a necessidade de novas abordagens para diminuir a quatidade de defeitos identificados em componentes de *hardware* e *software* em sistemas críticos de segurança específicos.

Um estudo de caso sobre sistemas de aquisição de sinais biomédicos foi conduzido para avaliar experimentalmente o método proposto e demonstrar como fabricantes podem aplicá-lo. O estudo de caso foi composto por atividades relacionadas ao processo de gerenciamento de risco, modelagem formal, e casos de garantia (*assurance cases*) [50]. Os requisitos definidos no padrão ISO 14971 foram modelados e validados formalmente por meio de redes de Petri coloridas (*Colored Petri Nets - CPN*) [34] e de simulações de modelos, respectivamente. As etapas definidas no processo de gerenciamento de risco foram realizadas em conformidade com os requisitos do padrão ISO 14971 utilizando o método de análise de criticidade e modo de efeito de falhas (*Failure, Mode, Effects and Criticality Analysis - FMECA*) [45].

A linguagem de especificação formal CPN também foi utilizada para modelar o comportamento de componentes de *hardware* e *software* durante este trabalho. Modelos CPN foram criados com base em equações de saída de circuito. Um modelo de referência de sistema de aquisição de sinais biomédicos foi especificado utilizando CPN e formalmente verificado com a técnica de verificação de modelos baseado em propriedades definidas com a lógica temporal modal ASK-CTL [11].

O modelo de referência foi estendido e validado para representar um sistema de Eletrocardiografia (ECG) [52]. Além disso, os modelos apresentados neste trabalho podem

ser facilmente estendidos por fabricantes para representar outros sistemas de aquisição de sinais biomédicos, como, por exemplo, sistemas de Eletroencefalografia (EEG) [49] e Eletromiografia (EMG) [99]. Por fim, modelos de casos de garantia de um sistema de ECG foram especificados com a notação estruturada por metas (*Goal Structuring Notation - GSN*) [60] considerando as definições do padrão para representação e compartilhamento de casos de garantia (*Assurance Cases Exchange Standard - ACES*) durante a última atividade do estudo de caso desenvolvido.

É possível destacar, por meio do estudo de caso sobre o sistema de ECG, que o modelo de referência é útil para gerar artefatos de projeto e, consequentemente, evidências de segurança e eficácia de sistemas específicos durante o processo de certificação. O modelo de referencia foi estendido com base no ECG *front-end* (AD8232) e no microcontrolador analógico de baixo consumo, ARM *cortex* M2 com conversores sigma-delta (ADUCM360) usando uma configuração simples de monitor cardíaco. O modelo estendido foi utilizado para gerar evidências de segurança e eficácia por meio de atividades de verificação e validação, da transformada de Fourier, e de métricas de qualidade. A abordagem de validação foi baseada em dados de seres humanos disponibilizados na base de dados PHYSIONET ECG-ID e análises no domínio do tempo e frequência. Com os resultados de verificação de modelos com ASK-CTL, pode-se observar que o modelo satisfaz propriedades de segurança específicas. Por outro lado, com análises no domínio do tempo e frequência, pode-se evidenciar o bom desempenho dos filtros projetados.

O sistema de ECG foi representado com base nas principais características dos componentes AD8232 e ADUCM360 porque são equipamentos comerciais comercializados por empresas bem estabelecidas no mercado. Além disso, uma configuração simples de monitor cardíaco foi definida para apresentar cenários de uso de geração de evidências de uma maneira mais clara possível. Entretanto, é possível especificar modelos mais detalhados e funções mais complexas de outros sistemas embarcados críticos de segurança usando o método apresentado.

Portanto, pode-se concluir que o método pode ser utilizado em um processo de certificação considerando padrões prescritivos e baseados em metas. Casos de garantia podem ser usados por todo o ciclo de desenvolvimento de sistemas embarcados críticos de segurança, desde a definição de requisitos, até atividades de verificação e validação.

A atividade de verificação e validação engloba várias atividades realizadas durante o desenvolvimento de *software*, como, por exemplo, a aplicação de estratégias de teste de *software* [84; 65]. É possível também realizar a rastreabilidade de requisitos regulatórios e requisitos específicos do produto. Rastreabilidade de requisitos possui um papel fundamental durante o desenvolvimento de *software* e para possibilitar a verificação automatizada de requisitos regulatórios por meio de casos de garantia.

6.1 Perspectivas e Trabalhos Futuros

O método foi apresentado neste documento como um arcabouço conceitual. Portanto, é importante também como trabalho futuro, automatizar o arcabouço para auxiliar fabricantes durante a aplicação do método para o desenvolvimento e certificação de sistemas embarcados críticos de segurança. A aplicação de todas as atividades especificadas no método devem ser automatizadas, e formalizações e algoritmos associados com atividades devem ser implementados seguindo as definições apresentadas.

Por outro lado, o padrão de casos de garantia ACES deve ser especificado em mais detalhes. Por exemplo, um esquema XML deve ser definido para possibilitar a verificação de conformidade com as especificações do padrão definido para GSN [60] e com especificações do próprio padrão ACES. Características relacionadas com autenticação e controle de versão podem ser também estudadas e definidas em mais detalhes.

Como sugestão de perspectivas e trabalhos futuros associados ao estudo de caso desenvolvido, pode-se destacar a aplicação do método descrito neste documento durante a especificação de uma aplicação para o diagnóstico de doenças cardiovasculares. Além disso, é importante especificar sistemas embarcados críticos de segurança mais complexos. Por exemplo, pode-se aplicar o método durante um estudo de caso sobre sobre sistemas de marcapasso cardíaco e gástrico.

Direções futuras sobre a definição de uma arquitetura de *software* como um cenário de implantação do método apresentado no Capítulo 4 são também necessárias. Uma arquitetura orientada a serviços (*Service-Oriented Architecture - SOA*) [62] deve ser definida para possibilitar a interoperabilidade entre sistemas de informações mantidos por fabricantes de sistemas embarcados críticos de segurança e por agências reguladoras. A arquitetura de

software deve ser definida com base nas especificações do método proposto neste documento de tese de doutorado.

Bibliografia

[1] IEEE standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70, Aug 2008.

[2] Y. Ait-Ameur and D. Méry. Making explicit domain knowledge in formal system development. *Science of Computer Programming*, 121:100–127, June 2016.

[3] H. Alemzadeh, R.K. Iyer, Z. Kalbarczyk, and J. Raman. Analysis of safety-critical computer failures in medical devices. *IEEE Security Privacy*, 11(4):14–26, July 2013.

[4] S. Ameer and O. Basir. Objective image quality measure based on weber-weighted mean absolute error. In *9th International Conference on Signal Processing*, pages 728–732, Oct 2008.

[5] Analog Devices. Single-Lead, Heart Rate Monitor Front End Data Sheet AD8232, 2013.

[6] Analog Devices. Low Power, Precision Analog Microcontroller with Dual Sigma-Delta ADCs, ARM Cortex-M3, Data Sheet ADuCM360/ADuCM361, 2014.

[7] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky. Formal methods based development of a pca infusion pump reference model: Generic infusion pump (gip) project. In *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability*, pages 23–33, June 2007.

[8] R. Baheti and H. Gill. Cyber-physical systems. *The Impact of Control Technology*, 2011.

[9] P. E. S. Barbosa, M. Morais, K. Galdino, M. Andrade, L. Gomes, F. Moutinho, and J. C. A. de Figueiredo. Towards medical device behavioural validation using petri nets.

In *2013 IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 4–10, June 2013.

[10] M. S. Chavan, R. A. Agarwala, and M. D. Uplane. Interference reduction in ecg using digital fir filters based on rectangular window. *WSEAS Transactions on Signal Processing*, 4(5):340–349, May 2008.

[11] A. Cheng, S. Christensen, and K. H. Mortensen. Model checking coloured petri nets exploiting strongly connected components. In *Proceedings of the International Workshop on Discrete Event Systems, WODES96. Institution of Electrical Engineers, Computing and Control Division*, pages 169–177, 1997.

[12] Cheng, A. and Christensen, S. and Mortensen, K. H. Model Checking Coloured Petri Nets Exploiting Strongly Connected Components. 1997.

[13] E. M. Clarke Jr., O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, Cambridge, MA, USA, 1999.

[14] J. W. Cooley and J. W. Tukey. Mathematics of computation. *An Algorithm for the Machine Calculation of Complex Fourier Series*, 1965.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivesr, and C. Stein. *Algoritmos: Teoria e Prática*. Elsevier, 2st edition, 2002.

[16] P. C. N. Cunha, A. Sobrinho, L. D. Silva, A. Perkusich, and J. R. A. Miranda. *Design of a Simulation Device to Test Electrogastrography (EGG) Systems*, pages 1–14. IGI-Global, 1 edition, 2016.

[17] Agência Nacional de Vigilância Sanitária. Manual para regularização de equipamentos médicos na anvisa, 2010.

[18] D. Dechev and B. Stroustrup. Model-based product-oriented certification. In *16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, pages 295–304, April 2009.

[19] E. Denney and G. Pai. Automating the assembly of aviation safety cases. *IEEE Transactions on Reliability*, 63(4):830–849, Dec 2014.

[20] J. Desel and W. Reisig. The concepts of petri nets. *Software & Systems Modeling*, 14(2):669–683, 2014.

[21] Federal Aviation Administration. Unmanned Aircraft Systems (UAS) Operational Approval, 2013.

[22] D. Feng and C. Eyster. Risk-based requirements management framework with applications to assurance cases. In *IEEE Aerospace Conference*, pages 1–11, March 2013.

[23] Centre for Devices, U.S. Food Radiological Heath, and Drug Administration. Infusion pump improvement initiative white paper, 2010.

[24] International Organization for Standardization. Medical device software - software life cycle processes, 2006.

[25] International Organization for Standardization. Medical devices - application of risk management to medical devices, 2007.

[26] J. B. Goodenough, C. B. Weinstock, and A. Z. Klein. Eliminative induction: A basis for arguing system confidence. In *35th International Conference on Software Engineering (ICSE)*, pages 1161–1164, May 2013.

[27] I. Habli and T. Kelly. Process and product certification arguments: Getting the balance right. *ACM SIGBED Review*, 3(4):1–8, October 2006.

[28] J. Han, Q. Ding, A. Xiong, and X Zhao. A state-space emg model for the estimation of continuous joint movements. *IEEE Transactions on Industrial Electronics*, 62(7):4267–4275, 2015.

[29] R. Hawkins, I. Habli, T. Kelly, and J. McDermid. Assurance cases and prescriptive software safety certification: A comparative study. *Safety Science*, 59(0):55–71, 2013.

[30] N. Hrgarek. Certification and regulatory challenges in medical device software development. In *2012 4th International Workshop on Software Engineering in Health Care (SEHC)*, pages 40–43, June 2012.

[31] R. Ivanov, M. Pajic, and I. Lee. Attack-resilient sensor fusion for safety-critical cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 15(1):1–24, February 2016.

[32] E. Jee, I. Lee, and O. Sokolsky. Assurance cases in model-driven development of the pacemaker software. In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification, and Validation*, volume 6416 of *Lecture Notes in Computer Science*, pages 343–356. Springer Berlin Heidelberg, 2010.

[33] K. Jensen. Coloured petri nets and the invariant-method. *Theoretical Computer Science*, 14(3):317–336, 1981.

[34] K. Jensen and L. M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[35] K. Jensen and L. M. Kristensen. Colored petri nets: a graphical language for formal modeling and validation of concurrent systems. *Communications of the ACM*, 58(6):61–70, June 2015.

[36] K. Jensen, L. M. Kristensen, and L. Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, 9(3):213–254, May 2007.

[37] Jensen, K. and Christensen, S. and Kristensen, L. M. CPN Tools State Space Manual. 2006.

[38] Z. Jiang, M. Pajic, R. Alur, and R. Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2):191–213, 2014.

[39] Z. Jiang, M. Pajic, and R. Mangharam. Cyber-physical modeling of implantable cardiac medical devices. *Proceedings of the IEEE*, 100(1):122–137, Jan 2012.

[40] B. Kim, A. Ayoub, O. Sokolsky, I. Lee, P. Jones, Y. Zhang, and R. Jetley. Safety-assured development of the gpca infusion pump software. In *Proceedings of the Ninth ACM International Conference on Embedded Software*, pages 155–164, 2011.

[41] J. Kim, I. Kang, J. Choi, I. Lee, and S. Kang. Formal synthesis of application and platform behaviors of embedded software systems. *Software & Systems Modeling*, 2(14):839–859, May 2015.

[42] C. Kitchin and L. Counts. *A designer's guide to instrumentation amplifiers*. Analog Devices, 3th edition, 2006.

[43] E. A. Lee. Fundamental limits of cyber-physical systems modeling. *ACM Transactions on Cyber-Physical Systems*, 1(1):1–26, November 2016.

[44] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems - A Cyter-Physical Systems Approach*. LeeSeshia.org, 2th edition, 2015.

[45] Y. Lee, D. Kim, J. Kim, and H. Kim. New fmeca methodology using structural importance and fuzzy theory. *IEEE Transactions on Power Systems*, 26(4):2364–2370, Nov 2011.

[46] S. Li, L. D. Xu, and X. Wang. A continuous biomedical signal acquisition system based on compressed sensing in body sensor networks. *IEEE Transactions on Industrial Informatics*, 9(3):1764–1771, Aug 2013.

[47] T. Li, F. Tan, Q. Wang, L. Bu, J. Cao, and X. Liu. From offline toward real time: A hybrid systems model checking and cps codesign approach for medical device plug-and-play collaborations. *IEEE Transactions on Parallel and Distributed Systems*, 25(3):642–652, March 2014.

[48] L. Li-jun and L. Zong-qiang. Design and simulation of led clock circuit based on proteus. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, volume 6, pages 315–316, Aug 2010.

[49] L. Liao, S. Wu, C. Liou, S. Lu, S. Chen, S. Chen, L. Ko, and C. Lin. A novel 16-channel wireless system for electroencephalography measurements with dry spring-loaded sensors. *IEEE Transactions on Instrumentation and Measurement*, 63(6):1545–1555, June 2014.

[50] C. Lin and W. Shen. Generation of assurance cases for medical devices. In R. Lee, editor, *Computer and Information Science*, volume 566 of *Studies in Computational Intelligence*, pages 127–140. Springer International Publishing, 2015.

[51] T. Maibaum and A. Wassyng. A product-focused approach to software certification. *Computer*, 41(2):91–93, Feb 2008.

[52] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith. A dynamical model for generating synthetic electrocardiogram signals. *IEEE Transactions on Biomedical Engineering*, 50(3):289–294, March 2003.

[53] D. Méry and N. K. Singh. Trustable formal specification for software certification. In *Proceedings of the 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation - Volume Part II*, ISoLA'10, pages 312–326. Springer-Verlag, 2010.

[54] D. Méry and N. K. Singh. Formal specification of medical systems by proof-based refinement. *ACM Transaction on Embedded Compuing Systems*, 12(1):1–25, January 2013.

[55] P. Mitros. Filters with decreased passband error. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 63(2):131–135, Feb 2016.

[56] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 1989.

[57] C. R. Neto. Especificação Formal do Gerenciamento de Risco de Equipamentos Médicos Baseado na ISO 14971:2009 Utilizando Redes de Petri Coloridas. Master's thesis, Universidade Federal de Alagoas, 2015.

[58] N. M. C. Oliveira. *Prática de Computação*. Coimbra, 1th edition, 2006.

[59] A. V. Oppenheim, A. S. Willsky, and S. Hamid. *Fast Fourier Transform - Algorithms and Applications*. Pearson, 2th edition, 1996.

[60] on behalf of the Contributors Origin Consulting (York) Limited. Gsn community standard version 1, 2011.

[61] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee. Model-driven safety analysis of closed-loop medical systems. *IEEE Transactions on Industrial Informatics*, 10(1):3–16, Feb 2014.

[62] M. P. Papazoglou and W. Heuvel. Service oriented architectures: Approaches, technologies and research issues. *The VLDB Journal*, 16(3):389–415, July 2007.

[63] M. J. M. Pelgrom. *Analog-to-Digital Conversion*. Springer Netherlands, 1th edition, 2010.

[64] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3th edition, 2007.

[65] R. S. Pressman and B. R. Maxim. *Engenharia de Software: Uma Abordagem Profissional*. McGraw-Hill, 8th edition, 2016.

[66] J. Qadir and O. Hasan. Applying formal methods to networking: Theory, techniques, and applications. *IEEE Communications Surveys Tutorials*, 17(1):256–291, Firstquarter 2015.

[67] K. R. Rao, Kim, and J. J. D. N.and Hwang. *Fast Fourier Transform - Algorithms and Applications*. Springer Netherlands, 1th edition, 2010.

[68] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, and Y. Lian. A level-crossing based qrs-detection algorithm for wearable ecg sensors. *IEEE Journal of Biomedical and Health Informatics*, 18(1):183–192, Jan 2014.

[69] N. Razzaq, S. A. A. Sheikh, M. Salman, and T. Zaidi. An intelligent adaptive filter for elimination of power line interference from high resolution electrocardiogram. *IEEE Access*, 4:1676–1688, 2016.

[70] M. Schlechtingen, I. F. Santos, and S. Achiche. Using data-mining approaches for wind turbine power curve monitoring: A comparative study. *IEEE Transactions on Sustainable Energy*, 4(3):671–679, July 2013.

[71] Boston Scientific. Pacemaker system specification, 2007.

[72] A. S. Sedra and K. C. Smith. *Microelectronic Circuits*. Oxford, 6th edition, 2009.

[73] Y. Seifi, S. Suriadi, E. Foo, and C. Boyd. Analysis of two authorization protocols using colored petri nets. *International Journal of Information Security*, pages 1615–5262, 2014.

[74] Z. Shao. Certified software. *Communications of the ACM*, 53(12):56–66, December 2010.

[75] L. C. Silva, M. Perkusich, F. M. Bublitz, H. O. Almeida, and A. Perkusich. A model-based architecture for testing medical cyber-physical systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 25–30. ACM, 2014.

[76] A. Sobrinho. Biomedical signal acquisition systems: Towards a hybrid methodology for certification-based development. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pages 799–802, July 2014.

[77] A. Sobrinho. Certification-based development methodology of biomedical signal acquisition systems. In *ICSE 2017 PhD and Young Researchers Warm Up Symposium/CBSoft*, 2014.

[78] A. Sobrinho, P. Cunha, L. D. Silva, A. Perkusich, T. Cordeiro, and J. Rêgo. A methodology for modeling and simulation of biomedical signal acquisition devices. In *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, pages 227–231, 2015.

[79] A. Sobrinho, P. Cunha, L. D. Silva, A. Perkusich, T. Cordeiro, and J. Rêgo. A simulation approach to certify electrocardiography devices. In *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, 2015.

[80] A. Sobrinho, P. Cunha, L. D. Silva, A. Perkusich, T. Cordeiro, and J. Segundo. Arguing effectiveness of biomedical signal acquisition devices using colored petri nets models and assurance cases in gsn. In *The 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2016.

[81] A. Sobrinho, J. Neto, P. Cunha, L. D. Silva, and A. Perkusich. A colored petri nets model of the risk management process based on the iso 14971 standard. In *41st Annual Conference of the IEEE Industrial Electronics Society*, pages 475–480, 2015.

[82] A. Sobrinho, A. Perkusich, L. Dias da Silva, T. Cordeiro, J. Rego, and P. Cunha. Towards medical device certification: A colored petri nets model of a surface electrocardiography device. In *40th Annual Conference of the IEEE Industrial Electronics Society*, pages 2645–2651, Oct 2014.

[83] A. Sobrinho, A. Perkusich, L. Dias da Silva, and P. Cunha. Using colored petri nets for the requirements engineering of a surface electrogastrography system. In *2014 12th IEEE International Conference on Industrial Informatics*, pages 221–226, July 2014.

[84] I. Sommervile. *Software Engineering*. Pearson, 9st edition, 2011.

[85] P. Steele. Certification-based development of critical systems. In *2012 34th International Conference on Software Engineering (ICSE)*, pages 1575–1578, June 2012.

[86] P. Steele, K. Collins, and J. C. Knight. Certification-based development of critical systems. In *Proceedings of the 29th International Systems Safety Conference*, August 2011.

[87] E. Stensrud, T. Skramstad, J. Li, and J. Xie. Towards goal-based software safety certification based on prescriptive standards. In *2011 First International Workshop on Software Certification (WoSoCER)*, pages 13–18, Nov 2011.

[88] X. Sun and Y. Zhang. Design and implementation of portable ecg and body temperature monitor. In *International Symposium on Computer, Consumer and Control*, pages 188–192, 2014.

[89] T. V. Tran and W. Chung. Ieee-802.15.4-based low-power body sensor node with rf energy harvester. *Bio-Medical Materials Engineering*, 24:3503–3510, 2014.

[90] I. Tumer and C. Smidts. Integrated design-stage failure analysis of software-driven hardware systems. *IEEE Transactions on Computers*, 60(8):1072–1084, Aug 2011.

[91] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, Jan 2009.

[92] L. Wanhammar. *Analog Filters Using MATLAB*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[93] A. Wassyng, T. Maibaum, M. Lawford, and H. Bherer. Software certification: Is there a case against safety cases? In R. Calinescu and E. Jackson, editors, *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, volume 6662 of *Lecture Notes in Computer Science*, pages 206–227. Springer Berlin Heidelberg, 2011.

[94] A. Wassyng, N. K. Singh, M. Geven, N. Proscia, H. Wang, M. Lawford, and T. Maibaum. Can product-specific assurance case templates be used as medical device standards? *IEEE Design Test*, 32(5):45–55, Oct 2015.

[95] J. G. Webster. *Medical Instrumentation Application and Design*. Wiley, 4th edition, 2009.

[96] D. Wu and E. Schnieder. Scenario-based modeling of the on-board of a satellite-based train control system with colored petri nets. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3045–3061, Nov 2016.

[97] D. Wu and E. Schnieder. Scenario-based system design with colored petri nets: an application to train control systems. *Software & Systems Modeling*, pages 1–23, 2016.

[98] F. Xu and G. Yan. Toward a wireless electronic capsule with microsensors for detecting dysfunction of human gastric motility. *IEEE Sensors Journal*, 15(4):2194–2202, April 2015.

[99] W. Youn and J. Kim. Development of a compact-size and wireless surface emg measurement system. In *ICCAS-SICE*, pages 1625–1628, Aug 2009.

[100] H. Yu, C. Lin, and B. Kim. Automotive software certification: Current status and challenges. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 9(1):74–80, 2016.