

UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**ANÁLISE DE CONFIABILIDADE DE MONITORES
MULTIPARÂMETRO UTILIZADOS EM UNIDADES DE
TERAPIA INTENSIVA**

MESTRANDO

MATHEUS SOARES DE ARAUJO

ORIENTADOR

LEANDRO DIAS DA SILVA

COORIENTADOR

ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO

MACEIÓ, AL

NOVEMBRO - 2020

MATHEUS SOARES DE ARAUJO

ORIENTADOR

LEANDRO DIAS DA SILVA

COORIENTADOR

ÁLVARO ALVARES DE CARVALHO CÉSAR SOBRINHO

MACEIÓ, AL

NOVEMBRO - 2020

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

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

A663a Araujo, Matheus Soares de.

Análise de confiabilidade de monitores multiparâmetro utilizados em Unidades de Terapia Intensiva / Matheus Soares de Araujo. – 2021.

103 f. : il.

Orientador: Leandro Dias da Silva.

Co-orientador: Álvaro Alvares de Carvalho César Sobrinho.

Dissertação (mestrado em Informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2020.

Bibliografia: f. 95-102.

Apêndice: f. 103.

1. Confiabilidade de *software*. 2. Modelagem computacional. 3. Monitor multiparamétrico. 4. Unidades de Terapia Intensiva. 5. Chess, Metodologia. I. Título.

CDU: 004.052



Folha de Aprovação

MATHEUS SOARES DE ARAÚJO

ANÁLISE DE CONFIABILIDADE DE MONITORES MULTIPARAMÉTRICOS UTILIZADOS EM UNIDADES DE TERAPIA INTENSIVA

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 30 de novembro de 2020.

Banca Examinadora:

Prof. Dr. LEANDRO DIAS DA SILVA
UFAL – Instituto de Computação
Orientador

Prof. Dr. ALVARO ALVARES DE CARVALHO CESAR SOBRINHO
UFAPE – Universidade Federal do Agreste de Pernambuco
Coorientador

Prof. Dr. THIAGO DAMASCENO CORDEIRO
UFAL – Instituto de Computação
Examinador Interno

Prof. Dr. GILBERTO FRANCISCO MARTHA DE SOUZA
USP – Universidade de São Paulo
Examinador Externo

Prof. Dr. LEONARDO MONTECCHI
UNICAMP – Universidade Estadual de Campinas
Examinador Externo

Agradecimentos

Venho por meio deste agradecer para a realização deste trabalho,

Aos meus orientadores Leandro Dias e Álvaro Sobrinho, que me ajudaram de todas as formas possíveis para que a realização deste trabalho fosse possível. A experiência que obtive com vocês com certeza estou levando ensinamentos valiosos para a minha vida e serei sempre grato.

Ao professor Leonardo Montecchi por sempre estar disponível e ajudar com informações de extrema importância sobre a ferramenta CHESS.

Ao professor Paulo Cunha por ser entrevistado em vários momentos deste trabalho e prover informações estratégicas sobre a utilização de sistemas médicos em unidades hospitalares.

À Érika, minha companheira e esposa, por durante todo este tempo estar ao meu lado me apoiando e incentivando em momentos difíceis.

À minha família por possibilitar condições concretas e emotivas para concluir esta etapa, me dando apoio e dedicação, especialmente, à minha mãe, meu pai e meu irmão.

Aos meus amigos por deixarem a vida mais fácil.

Aos meus colegas de trabalho pelo apoio e paciência em um momento decisivo de minha vida e carreira acadêmica.

À CAPES e a FAPAL por prover bolsa por um período muito importante em minha vida.

Agradeço a todos que, de algum modo, contribuíram na longa trajetória acadêmica dessa jornada.

Resumo

Um sistema de monitoramento multiparâmetro geralmente é aplicado para monitorar a condição clínica de pacientes em Unidades de Terapia Intensiva (UTIs). As UTIs são ambientes de monitoramento contínuo utilizados para hospedar pacientes em estados graves de saúde. Esses sistemas de sistemas (*Systems of Systems* (SoS)), usados em UTIs, são compostos por conjuntos de sistemas constituintes (*Constituent Systems* (CS)) para medir valores de parâmetros, como, por exemplo, frequência cardíaca, frequência respiratória e temperatura. Devido à natureza crítica e relevância de UTIs, este tipo de SoS deve ser o mais confiável possível. Principalmente, em situações de emergência, como é o caso da pandemia de COVID-19, que resultou na sobrecarga de sistemas de saúde em todo o mundo. Sistemas de monitoramento multiparâmetro em UTIs devem possuir altos níveis de confiabilidade, dado que falhas podem acarretar riscos à integridade física de pacientes. Neste trabalho, foram realizadas análises de confiabilidade e fornecidas recomendações para auxiliar no gerenciamento dos sistemas de monitoramento multiparâmetro utilizados em UTIs, considerando também estratégias de manutenção preventiva. Foram elicitados requisitos com base em entrevistas com um profissional que possui mais de quinze anos de experiência em manutenção em hospitais. Além disso, foram analisados sistemas existentes e desenvolvidas revisões da literatura. Foi definido um modelo de sistemas de monitoramento multiparâmetro para UTIs usando a linguagem de modelagem CHES-ML. Posteriormente, foram realizadas análises de confiabilidade aplicando a análise baseada em estados do *plugin* CHES-SBA, simulando diferentes cenários. Com base nas análises, foi identificado que a fonte de alimentação principal e a bateria são os CS que apresentaram maiores impactos negativos na confiabilidade de todo o sistema em situações de falha. Em situações de emergência, intervalos de tempo reduzidos de manutenção preventiva, quando aplicados durante um curto período, mostraram-se estratégias promissoras para aumentar a confiabilidade de sistema.

Palavras-chave: Confiabilidade, Modelagem, Monitor Multiparâmetro, Unidades de Terapia Intensiva, Metodologia CHES.

Abstract

A multi-parameter monitoring system is usually applied to keep track of the clinical condition of patients in Intensive Care Units (ICUs). ICUs are continuous monitoring environments used to host patients in serious health conditions. These Systems of Systems (SoS), used in ICUs comprise of a set of Constituent Systems (CS) to measure parameters such as heart rate, respiratory frequency, and temperature. Due to the critical nature and relevance of ICUs, such SoS shall be as reliable as possible. This is specially true in emergency situations, as is the case of the COVID-19 outbreak, which resulted in the burden of health care systems. Multi-parameter monitoring systems in ICUs shall have high levels of reliability, given that failures can provide risks to the safety of patients. We performed reliability analysis and provided insights to assist the management of multi-parameter monitoring systems used in ICUs, also considering preventive maintenance. We elicited requirements by interviewing a professional with more than fifteen years of hospital experience in maintenance. In addition, we analyzed existing systems and literature reviews. Therefore, we modeled multi-parameter monitoring systems for ICUs using the CHESS-ML modeling language. Afterward, we conducted reliability analysis by applying the CHESS-SBA plugin state-based analysis, simulating different scenarios. Based on the reliability analysis, we identified that main power supply and the battery are the CS that present major negative impacts in the total reliability the entire system in failure situations. In emergency situations, reduced time ranges of preventive maintenance, when applied during a short period, showed to be promising strategies to increase quality of multi-parameter monitoring systems for ICUs.

Keywords: Reliability, Modeling, Multi-parameter Monitor, Intensive Care Units, CHESS Methodology.

Lista de siglas e abreviaturas

ACM - *Association for Computing Machinery*
ANVISA - Agência Nacional de Vigilância Sanitária
CDF - *Cumulative Distribution Function*
CPN - *Coloured Petri Nets*
CPU - *Central Process Unit*
CS - *Constituent Systems*
ECG - Eletrocardiograma
EUA - Estados Unidos da América
FDA - *Food and Drug Administration*
FMEA - *Failure Mode and Effects Analysis*
FMECA - *Failure Mode, Effects & Criticality Analysis*
FPTC - *Failure Propagation Transformation Calculus*
FTA - *Fault Tree Analysis*
IEEE - *Institute of Electrical and Electronics Engineers*
ISTI-CNR - Instituto de Ciência e Tecnologia da Informação
LCD - *Liquid-crystal Display*
ML - *Modeling Language*
MTBF - *Mean Time Between Failures*
MTTF - *Mean Time To Failure*
MTTR - *Mean Time to Repair*
PDF - *Probability Density Function*
RSL - Revisão Sistemática da Literatura
SpO2 - Saturação de Oxigênio
SBA - *System Based Analysis*
SoS - Sistema de Sistemas
SYSML - *Systems Modeling Language*
VDM-SL - *Vienna Development Method - Specification Language*
VSL - *Value Specification Language*
UML - Linguagem Unificada de Modelagem

UNFI - Universidade de Florença

UTI - Unidade de Terapia Intensiva

Conteúdo

1	Introdução	1
1.1	Problemática	5
1.2	Motivação	8
1.3	Objetivos	9
1.4	Contribuições	10
1.5	Metodologia	10
1.6	Organização do Documento	12
2	Embasamento Teórico	13
2.1	Confiabilidade de Sistemas	13
2.1.1	Análise de Confiabilidade	14
2.1.2	Distribuições de Probabilidade Exponencial	16
2.2	CHES - <i>Modeling Language</i>	17
2.2.1	Arquitetura do Sistema	18
2.2.2	Arquitetura do Software	19
2.2.3	Arquitetura do Hardware	21
2.2.4	Geração de Instâncias	22
2.2.5	Alocações	24
2.3	CHES - <i>State Based Analysis</i>	24
2.3.1	<i>SimpleStochasticBehavior</i>	28
2.3.2	<i>FLABehavior</i>	29
2.3.3	<i>ErrorModelBehavior</i>	30
2.3.4	Manutenção de Componentes	32
2.3.5	Execução de Análises e Métricas	34

2.4	Considerações Finais	36
3	Trabalhos Relacionados	38
3.1	Revisão Sistemática da Literatura	38
3.1.1	Protocolo da Revisão	38
3.1.2	Observações Sobre Trabalhos Aceitos	41
3.1.3	Análise Sobre a Revisão Sistemática	46
3.2	Outros Trabalhos Relacionados	47
3.2.1	Aplicação da Metodologia CHESSE	47
3.2.2	Sistemas Médicos	48
3.2.3	Análise de Confiabilidade	50
3.3	Considerações Finais	51
4	Especificação do Sistema	52
4.1	Descoberta e Elicitação de Requisitos	52
4.2	Arquitetura e Organização dos Modelos	56
4.3	Modelagem de um Sistema de Monitoramento Multiparâmetro	58
4.3.1	Monitor Multiparâmetro	58
4.3.2	Sistema Constituinte de Gerenciamento de Energia	60
4.3.3	Sistema Constituinte de ECG e Verificador de Respiração	64
4.3.4	Sistema Constituinte de Termômetro	66
4.3.5	Sistema Constituinte de Oximetria de Pulso	67
4.3.6	Sistema Constituinte de Verificação de Pressão Arterial	69
4.4	Considerações Finais	71
5	Análises de Confiabilidade	73
5.1	Configuração de Métricas Utilizadas	74
5.2	Análise com SimpleStochasticBehavior e ErrorModelBehavior	77
5.3	Estratégias de Manutenção	80
5.4	Considerações Finais	83
6	Discussões	84
6.1	Resultados de Análise	84

6.2 Dificuldades Encontradas e Sugestões de Melhorias	89
6.3 Considerações Finais	91
7 Conclusões	93

Lista de Figuras

1.1	Passos metodológicos utilizados durante a pesquisa.	11
2.1	Vida útil de componentes similares, adaptado de [68].	15
2.2	Média dos testes de vida útil de componentes similares, adaptado de [68].	15
2.3	Definição de blocos e portas de comunicação a nível de sistema, adaptado de [1].	19
2.4	Definição de diagrama de blocos internos, adaptado de [1].	19
2.5	Definição de componentes, interfaces e implementações de componentes, adaptado de [52].	21
2.6	Definição do diagrama de estrutura composta, adaptado de [52].	22
2.7	Pacote de instâncias gerados através do recurso Build Instances, adaptado de [52].	23
2.8	Estados de um componente no CHESS, adaptado de [52].	25
2.9	Exemplo de modelagem da propagação de uma falha em cadeia, adaptado de [52].	27
2.10	Componente com informações de confiabilidade adicionadas utilizando o estereótipo SimpleStochasticBehavior, adaptado de [1].	29
2.11	Componente com informações sobre o comportamento de falha utilizando o estereótipo FLABehavior, adaptado de [1].	30
2.12	Diagrama de Máquina de Estados com o estereótipo ErrorModel	33
2.13	Componente do tipo Activity com o estereótipo Repair para simular estratégias de manutenção com o CHESS-SBA, adaptado de [52].	34
3.1	Gráfico de classificação dos artigos selecionados.	42
3.2	Fluxograma utilizado para representar o processo de execução da RSL.	43

4.1 Fluxo similar de captura de sinal entre todos os sistemas de um sistema de monitoramento multiparâmetro.	55
4.2 Hierarquia de blocos definidas para o sistema de monitoramento multiparâmetro.	57
4.3 Diagrama de definição de blocos para o sistema de monitoramento multiparâmetro.	59
4.4 Diagrama de bloco interno para o monitor multiparâmetro.	61
4.5 Comunicação entre os componentes que gerenciam energia do sistema.	62
4.6 Máquina de estado representando o modo de falha para bateria.	63
4.7 Máquina de estado representando o modo de falha para fonte.	63
4.8 Máquina de estado representando o modo de falha para controlador da fonte.	64
4.9 Diagrama de definição de blocos para o módulo de ECG e verificador de respiração.	65
4.10 Diagrama de bloco interno para o módulo de ECG e verificador de respiração.	66
4.11 Diagrama de definição de blocos para o sistema constituinte de Termômetro.	67
4.12 Diagrama de bloco interno para o sistema constituinte de Termômetro.	68
4.13 Diagrama de definição de blocos para o sistema constituinte de Oxímetro de Pulso.	68
4.14 Diagrama de bloco interno para o sistema constituinte de Oxímetro de Pulso.	69
4.15 Diagrama de definição de blocos para o sistema constituinte de Verificador de Pressão Arterial.	70
4.16 Diagrama de bloco interno para o sistema constituinte de Verificador de Pressão Arterial.	71
5.1 Componente Filter configurado com o estereótipo SimpleStochasticBehavior.	78
5.2 Componente MultiparameterMonitor configurado com o estereótipo StateBasedAnalysis para executar análise de confiabilidade.	78
5.3 Gráfico com a confiabilidade do Monitor Multiparâmetro sem estratégias de manutenção e com estratégias de manutenção para 1 ano e 6 meses.	81

5.4	Gráfico com a confiabilidade do Monitor Multiparâmetro com estratégias de manutenção para 1 mês e 3 meses.	82
6.1	Longevidade do Monitor Multiparâmetro com os estereótipos SimpleStochasticBehavior e ErrorModelBehavior.	86
6.2	Longevidade do Monitor Multiparâmetro com estratégias de manutenção utilizadas em hospitais atualmente e propostas por este trabalho.	88

Lista de Tabelas

2.1 Métricas que podem ser analisadas com o estereótipo StateBasedAnalysis, Tabela adaptada de [52].	36
3.1 Quantidade de trabalhos coletados nas base de dados.	40
3.2 Número de trabalhos classificados realizando análises de títulos e resumos.	40
3.3 Número de trabalhos aceitos para contribuir com a RSL.	41
5.1 Taxa de ocorrência de falha dos componentes utilizando a distribuição expo- nencial.	76
5.2 Conversão de tempo (dias/meses) para horas.	77
5.3 Resultados de confiabilidade utilizando os estereótipos SimpleStochasticBehavior e ErrorModelBehavior.	80
5.4 Confiabilidade com estratégias de manutenção preventiva ocorridas em 6 meses e 1 ano.	81
5.5 Confiabilidade de um sistema de monitoramento multiparâmetro com pro- postas de estratégias de manutenção preventiva para 1 mês e 3 meses.	82

Capítulo 1

Introdução

Os organismos vivos possuem muitos sistemas, como, por exemplo, o corpo humano, que compreende o sistema nervoso, sistema cardiovascular e sistema músculoesquelético. Cada sistema é composto por vários subsistemas responsáveis por realizar processos fisiológicos. Processos fisiológicos são fenômenos complexos, que incluem a estimulação e controle nervoso hormonal, e ações que podem ser mecânicas, elétricas ou bioquímicas. Esses processos fisiológicos podem se manifestar como sinais que refletem sua natureza e atividades. Sinais podem ser bioquímicos, na forma de hormônios ou neurotransmissores; elétricos, na forma de potencial ou corrente; e físicos, na forma de pressão ou temperatura [14].

Ao longo dos anos e com o avanço da ciência e tecnologia, a utilização de sistemas de aquisição de sinais biomédicos tem se tornado mais comum, tanto para a realização de exames de acompanhamento, quanto para o monitoramento e tomada de decisão auxiliando profissionais da saúde em procedimentos médicos. Um exemplo de uso é a coleta de informações para a tomada de decisão do profissional responsável pelo atendimento. O corpo humano emite sinais biológicos que são desenvolvidos usando atividades celulares e processos fisiológicos. A partir de processos relacionados com o corpo humano, é possível medir sinais elétricos aplicando eletrodos na superfície da pele, saturação de oxigênio incidindo luz na pele em extremidades do corpo, temperatura utilizando sensores de temperatura e respiração utilizando sensores de capnografia. Com os sinais emitidos pelo corpo, é possível identificar o funcionamento de alguns processos vitais, como, por exemplo, o funcionamento do coração usando o Eletrocardiograma (ECG) [60], verificação de temperaturas usando termômetros e checagem de nível de oxigênio no sangue usando oxímetro de pulso. Por exemplo,

estes sinais vitais são utilizados para o monitoramento de pacientes em ambientes de Unidades de Terapia Intensiva (UTIs).

O surgimento das UTIs se deu a partir da necessidade do aperfeiçoamento e concentração de recursos materiais e humanos para o atendimento de pacientes em estados graves e críticos, ainda considerados como recuperáveis. Para os pacientes em estados recuperáveis, foi percebida a necessidade de observação constante, com assistência médica e de enfermagem contínua, centralizadas em um único local [19]. No Brasil, por exemplo, a capacidade de UTIs é, normalmente, de 6% a 10% do total de leitos existentes em hospitais, possuindo iluminação adequada, gerador de energia específico para UTIs, ambiente climatizado, paredes laváveis, leitos, monitores multiparâmetro, estativas e ventiladores pulmonares [39].

As UTIs não possuem espaço para falhas, tanto humana, quanto de algum sistema ou aparelho utilizado no ambiente para monitorar o paciente ou para ajudar a manter o paciente com vida [73]. A criticidade em ambientes hospitalares, como UTIs, tem chamado a atenção de pesquisadores e motivado o desenvolvimento de novas técnicas para aumentar a confiança no funcionamento, além da aplicação de técnicas já utilizadas em outras áreas, como, por exemplo, o relatório de eventos adversos, que é um componente essencial na aviação para expandir a segurança de sistemas [75].

Um sistema de monitoramento multiparâmetro é um Sistema de Sistemas (SoS) que possibilita a análise de diversos dados fisiológicos, fornecendo informações para auxiliar na tomada de decisão dos profissionais de saúde quanto à situação clínica de pacientes adultos, pediátricos e neonatais. Tal sistema é utilizado em setores de emergência, UTIs, centros cirúrgicos e ambulatoriais. Um SoS coordena sistemas constituintes (*Constituent Systems* (CS)) autônomos que fornecem serviços para atingir um objetivo específico [20]. Com um monitor multiparâmetro, é possível acompanhar diversas informações fisiológicas, fornecendo dados para os profissionais de saúde analisarem o estado clínico de pacientes adultos, pediátricos e neonatos. O monitor multiparâmetro é um equipamento fundamental nos ambientes citados anteriormente, por exibir resultados sem a necessidade de que ocorra interferência humana para coletar os resultados manualmente do paciente. Para realizar o monitoramento de sinais do paciente, um monitor multiparâmetro possui CS como, por exemplo, o ECG, verificador de respiração, termômetro, verificador de pressão arterial invasiva, verificador de pressão não invasiva e saturação de oxigênio [7].

Com estes CS, é possível coletar sinais do paciente e configurar alarmes em um monitor multiparâmetro de acordo com o tipo de paciente que está sendo monitorado, seja uma criança, um adulto ou um idoso, para assegurar que o paciente esteja sendo monitorado de forma precisa de acordo com sua situação de saúde. Ao longo dos anos, os monitores multiparâmetro foram evoluídos tecnologicamente para suportar simplicidade de operação, alta disponibilidade, confiabilidade e alto nível de precisão das informações coletadas do paciente [47]. Portanto, monitores multiparâmetro são sistemas médicos críticos no monitoramento de pacientes em estado grave em UTIs. Neste contexto, quando uma falha ocorre, pode resultar em indisponibilidade do equipamento médico em um determinado período de tempo, e, conseqüentemente, gerar danos à vida do paciente e também induzir a tomada de decisões equivocadas [63].

Em um cenário de pandemia, como, por exemplo, a da COVID-19, vivenciada no ano de 2020, ambientes de UTIs são usados para hospedar um alto número de pacientes que estão gravemente doentes [31]. O acometimento respiratório é uma das principais complicações ocasionadas pela COVID-19, como também o agravamento de alguma comorbidade, como, por exemplo, diabetes, cardiopatia ou doença respiratória, como asma ou pneumonia [34]. Complicações clínicas relacionadas com a COVID-19 têm levado inúmeros pacientes à óbito em todo o mundo [33].

Monitores multiparâmetro em UTIs têm auxiliado profissionais de saúde no combate da pandemia da COVID-19, dado que são relevantes para o monitoramento contínuo de vários parâmetros fisiológicos de um paciente, possuindo foco na medição de saturação de oxigênio (SpO₂) e também para a medição de índice de respiração do paciente. Com isso, monitores multiparâmetro que possuem altos índices de confiabilidade, são indispensáveis em UTIs que estão com paciente com COVID-19, por ser possível, através deste equipamento monitorar sinais vitais do paciente e ajudar profissionais de saúde para a tomada de decisões em momentos críticos [71].

Para a construção de um monitor multiparâmetro devem ser desenvolvidos componentes de *software* e *hardware*. Componentes de *software* têm a responsabilidade de verificação de índices de bateria, verificação de impedância eletrodo-pele, processamento das informações coletadas do paciente e a visualização dos dados coletados do paciente em um monitor. Os componentes de *hardware* possuem elementos, como por exemplo eletrodos, filtros, trans-

dutores, amplificadores, conversores e microprocessadores [45].

Para o desenvolvimento de um projeto de um monitor multiparâmetro é necessário elicitar requisitos funcionais e não funcionais. Requisitos funcionais são necessidades, características ou funcionalidades esperadas, que devem ser atendidas na construção de sistemas. Por outro lado, requisitos não funcionais são requisitos que não estão associados diretamente às funcionalidades fornecidas no sistema. Portanto, estão relacionados com propriedades de sistemas críticos, como confiabilidade, tempo de resposta e desempenho [57]. Os requisitos não funcionais devem ser mensuráveis, sendo possível verificar se os requisitos do sistema obedecem aos padrões definidos na elicitação de requisitos. Segundo Pressman e Maxim [58], confiabilidade é definida em termos estatísticos como a probabilidade de operação livre de falhas de um sistema, em um ambiente específico, durante um determinado período de tempo. Neste contexto, confiabilidade é categorizada como um requisito não funcional.

Pressman e Maxim [58] complementam que a confiabilidade pode ser medida e calculada utilizando dados históricos de desenvolvimento. Por exemplo, para um *software* que possui uma confiabilidade de 0.96 em um tempo de 8 horas de processamento, se este for executado 100 vezes e exigir 8 horas de tempo de processamento, a probabilidade desse *software* funcionar corretamente (sem falhas) é de 96 das 100 vezes [58].

Durante a história da humanidade, alguns acontecimentos incentivaram a atenção redobrada para requisitos não funcionais de confiabilidade. Por exemplo, durante a segunda guerra mundial, a Alemanha teve sérios problemas de confiabilidade com os mísseis V-1 e V-2 [46]. Portanto, um matemático da equipe de construção dos mísseis, Pieruschka, afirmou que a confiabilidade de um sistema em série é o produto das confiabilidades individuais de cada componente [10]. Com isso, houve a necessidade de aplicar melhorias em certos componentes para atingir resultados com valores aceitáveis no contexto ao qual estava sendo aplicado, resultando em um aumento significativo na confiabilidade dos mísseis. Outros exemplos de fatores históricos incluem a não disponibilidade de equipamentos eletrônicos durante a guerra da Coreia e as quedas de energia no ano de 1965 nos estados da região norte dos Estados Unidos da América (EUA) [16].

Uma falha em um sistema implica na interrupção do bom funcionamento do sistema em cumprir sua função de acordo com o qual foi projetado. Portanto, a confiabilidade de um sistema pode ser quantificada pela probabilidade de se manter disponível por um determinado

período de tempo. Disponibilidade descreve a probabilidade de que o sistema esteja pronto para o funcionamento e por quanto tempo o sistema está disponível para ser utilizado [58].

Atualmente, existem algumas ferramentas para modelar componentes de sistemas, *software* e *hardware*, e com isso realizar análises de propriedades de confiabilidade. Um exemplo é a metodologia CHES¹ [13]. Com a metodologia CHES, é possível modelar sistemas utilizando a linguagem de modelagem CHES-Modeling Language (CHES-Linguagem de Modelagem (CHES-ML))², composta por subconjuntos da *Unified Modeling Language (UML)* e *Systems Modeling Language (SysML)*. A linguagem de modelagem SysML oferece suporte para a especificação, análise e projeto de sistemas [37]. Já a UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de *software* [12]. Além da modelagem realizada utilizando o CHES-ML, é possível utilizar o *plugin* para análise baseada em estados (*CHES-State Based Analysis (CHES-SBA)*) para realizar análises de confiabilidade usando o modelo desenvolvido. Além disso, o *plugin CHES-Failure Logic Analysis (CHES-FLA)* pode ser utilizado para realizar análise de lógica de falha baseada em modelos [25]. O CHES-SBA executa análise de confiabilidade utilizando o formalismo de redes de Petri estocástica, calculando medidas de confiabilidade probabilística por taxas de falha e comportamento de propagação de componentes que foram descritos com o CHES-ML. As redes de Petri estocásticas são geradas automaticamente pelo *plugin CHES-SBA* sem interferência do usuário na realização de análises.

1.1 Problemática

Falhas em sistemas críticos são passíveis de causar danos à vidas humanas. Os sistemas críticos são divididos em sistemas críticos seguros, de missão e de negócio. Sistemas críticos seguros são sistemas nos quais falhas podem resultar em perda de vida, prejuízo ou dano ambiental. Sistemas críticos de missão são sistemas nos quais falhas podem ocasionar a deficiência de alguma atividade que está dirigida a metas. Sistemas críticos de negócios são sistemas nos quais falhas podem resultar em perdas econômicas elevadas [38].

¹<https://www.eclipse.org/chess/>

²<https://github.com/montex/CHES-SBA/wiki/Modeling-the-Architecture>

Alguns sistemas necessitam de índices de confiabilidade dentro dos padrões do contexto que serão utilizados [58]. Por exemplo, sistemas críticos seguros, por lidarem com o risco de expor a integridade física de seres humanos, devem possuir confiabilidade elevada. No ano de 2020, a estação espacial internacional apresentou falhas em três sistemas críticos, sendo uma dessas falhas em um sistema de geração de oxigênio. Os tanques que armazenam a água que é separada em oxigênio e hidrogênio por eletrólise estavam vazios. Nesta situação, os tripulantes da estação não foram afetados, pois a estação possui sistemas redundantes que garantiram a segurança dos tripulantes.

No contexto deste trabalho, ambientes de UTIs, por lidarem com pacientes de estado grave, não são passíveis de erros. Uma falha de um sistema médico ou humano em UTIs pode colocar em risco a vida do paciente em monitoramento, ou até mesmo resultar em óbito. Em um estudo realizado em um período de dois anos, foi relatado que 76 pacientes foram a óbito devido a incidentes relacionados a sistemas médicos, além de 197 vítimas de incidentes graves. Dentre esses incidentes, foram categorizados o erro humano e falhas ocasionadas pelo uso de sistemas médicos [11].

Um monitor multiparâmetro, utilizado em UTIs, é um sistema crítico no monitoramento do paciente, pois profissionais podem acompanhar os sinais vitais do paciente de forma automatizada e escalonar atendimento para outros pacientes. Com um monitor multiparâmetro indisponível para um paciente, o profissional de saúde precisa realizar procedimentos de checagem de sinais vitais de forma manual no paciente periodicamente, o que é inviável pelo número reduzido de profissionais disponíveis em UTIs. Uma das causas que podem levar a indisponibilidade de um monitor multiparamétrico em UTIs é a falta de energia. Para lidar com este problema, um dos requisitos de confiabilidade de um monitor multiparâmetro é que o mesmo deve funcionar por energia elétrica ou por bateria. Em casos nos quais a energia elétrica esteja ausente por algum motivo não previsto, é necessário manter o dispositivo funcionando por alimentação via bateria [7].

A pandemia da COVID-19 evidenciou a necessidade das UTIs serem usadas para tratar um elevado número de pacientes com problemas respiratórios em um curto período de tempo. Porém, vários países ao redor do mundo ficaram sem leitos de UTIs, além da falta de sistemas médicos, devido a superlotação das UTIs com pacientes diagnosticados com COVID-19 [31]. Portanto, sistemas médicos, incluindo monitores multiparâmetro utilizados em UTIs, devem

ser confiáveis, dada a criticidade que o ambiente possui. Os sistemas de monitoramento multiparâmetro são recursos essenciais para o apoio no enfrentamento de pandemias como a COVID-19 [71].

Cada leito em UTIs em combate a COVID-19, além do monitor multiparâmetro, deve possuir ventilador, bomba de seringa e cama ondulada. Uma das complicações que o paciente com COVID-19 pode desenvolver são problemas respiratórios. O monitor multiparâmetro tem a funcionalidade de monitorar índices de respiração do paciente. Portanto, a ocorrência de uma falha em um monitor multiparâmetro, ou indisponibilidade do mesmo, acarreta na diminuição do número de pacientes monitorados em UTIs e compromete a segurança do paciente que está sem monitoramento.

Com o conhecimento do comportamento das falhas de um monitor multiparâmetro, gestores de UTIs podem definir estratégias de manutenção que mais se adequam a cada tipo de falha, seja ela preventiva, corretiva ou preditiva. A realização de manutenção preventiva possui um menor custo do que a realização da troca de um componente no momento ou depois que a falha ocorreu [64]. Além do conhecimento sobre o comportamento de falhas, é necessário o conhecimento sobre a importância do equipamento nos procedimentos clínicos ou nas atividades de apoio, seu nível de obsolescência, características de construção e possibilidade de substituição durante a manutenção. Esse tipo de conhecimento auxilia os gestores de UTIs e os técnicos na análise de detecção de falhas, bem como no conhecimento da urgência da realização do serviço para obter o nível de confiabilidade exigido [15].

Técnicas de manutenção e redundância são relevantes para aumentar a confiabilidade no funcionamento de sistemas e o tempo de vida útil, ocasionando uma maior longevidade do sistema [38]. Um desafio neste trabalho é a realização de análises de confiabilidade de um SoS (monitor multiparamétrico) utilizado em UTI para diferentes intervalos de tempo, no qual este SoS é composto por CS, visto que cada um possui uma probabilidade de falha específica, e que o funcionamento em conjunto leva à propagação de falhas de um CS a outro e a todo o sistema.

1.2 Motivação

Na última década, a *Food and Drug Administration* (FDA) registrou 5.294 notificações de defeitos em sistemas médicos entre 2006 e 2011. Dentre esses 5.294, 1.210 notificações foram associadas com equipamentos que possuem *software*. Entre estes resultados, 33% possuíam falhas relacionadas ao *software*, e 66.7% são falhas que podem ser distribuídas entre dispositivos de entrada e saída (sensores e botões), componentes de *hardware* (curtos circuitos e chips de memórias) e em baterias [6].

Profissionais, tanto no mercado, quanto na área acadêmica, têm buscado construir ferramentas, tendo como objetivo contemplar os requisitos desejados com altos níveis de confiabilidade em sistemas. Atualmente, por exemplo, existem modelagens de sistemas de aquisição de sinais biomédicos desenvolvidos utilizando modelos formais com redes de Petri coloridas, Notação Z e Event-B na área da saúde [49].

Em estudos em sistemas da área médica, é possível observar, por exemplo, aplicações em bombas de infusão, máquinas de raio-x e marca-passos [28]. O desenvolvimento baseado em modelos formais pode produzir uma aplicação mais confiável e disponível do que os métodos mais tradicionais. Com isso, alguns temas para o desenvolvimento de sistemas utilizando modelos formais têm recebido uma atenção e maior dedicação, que são: geração de casos de testes, verificação probabilística, verificação de confiabilidade em sistemas de saúde, verificação de privacidade em sistemas de saúde, verificação de integração de dados e *big data* em serviços de saúde [28]. Neste contexto, o *plugin* CHESS-SBA utiliza o formalismo de redes de Petri estocásticas para calcular a confiabilidade de modelos desenvolvidos com CHESS-ML.

Sistemas médicos compostos por *hardware* sofrem desgastes de acordo com a sua utilização e com o passar do tempo. Portanto, a manutenção preventiva em intervalos de tempo é importante para aumentar longevidade e disponibilidade (relevante, por exemplo, em UTIs), aumentando a confiabilidade e diminuindo o risco de falha [69]. Com a pandemia da COVID-19, as UTIs têm recebido um alto número de pacientes, maior do que o normal, o que intensifica a importância da manutenção dos sistemas. UTIs que não possuem estratégias de manutenção definidas podem correr riscos que comprometam o funcionamento correto, paralisando o atendimento, aumentando o risco de vida dos pacientes, limitando a quantidade

de pacientes que podem hospedar, e também falta de suporte para a equipe médica. A manutenção preventiva também tem um fator impactante no aspecto financeiro da manutenção de UTIs, principalmente no quesito de planejamento financeiro, no qual a gerência de UTIs deve estar ciente de quanto em quanto tempo será feita a manutenção, de quanto serão os custos e da menor probabilidade de falha que os sistemas médicos terão [69].

A metodologia CHES contém ferramentas que auxiliam os projetistas na modelagem de sistemas e na realização de análises de confiabilidade. A aplicabilidade da metodologia CHES já pôde ser validada em outros domínios, como, por exemplo, no meio automotivo, realizando análises de confiabilidade de um sistema de freio híbrido automotivo [13]. Portanto, uma motivação deste trabalho é avaliar a possibilidade de aplicar a metodologia CHES, junto com a linguagem CHES-ML e o plugin CHES-SBA para a análise de confiabilidade de sistemas de monitoramento multiparâmetros. Assim, a avaliação da confiabilidade de um modelo desenvolvido de sistemas de monitoramento multiparâmetros com o CHES em diferentes intervalos de tempo com e sem estratégias de manutenção preventiva, poderá ser validada para que gestores de hospitais possam definir estratégias de manutenção mais adequadas tendo como foco custos financeiros e o ganho de confiabilidade na longevidade do equipamento.

1.3 Objetivos

O objetivo principal deste trabalho é apresentar a aplicabilidade da metodologia CHES a sistemas médicos, apresentando resultados de análises de confiabilidade de sistemas de monitoramento multiparâmetro para utilização em UTIs. Com isso, apresentar recomendações para a gerência de ambientes hospitalares na manutenção preventiva de sistemas médicos, reduzindo a indisponibilidade que esses sistemas podem causar no atendimento de pacientes em UTIs. Como objetivos específicos, os seguintes são destacados:

- modelar sistemas de monitoramento multiparâmetro com a metodologia CHES;
- realizar análises de confiabilidade de sistemas de monitoramento multiparâmetro com a metodologia CHES em diferentes intervalos de tempo (cenários); e

- realizar discussões com as análises de confiabilidade coletadas e sobre a aplicabilidade do arcabouço CHESS a sistemas médicos, considerando uso e manutenção preventiva.

1.4 Contribuições

Com base nos requisitos de sistemas de monitoramento multiparâmetro em ambientes de UTIs, foi possível apresentar e analisar confiabilidade. Portanto, exemplos de contribuições obtidas com trabalho são:

- disponibilização de um modelo de sistemas de monitoramento multiparâmetro utilizando a metodologia CHESS;
- apresentação de discussões sobre a confiabilidade de sistemas em diferentes intervalos de tempo;
- recomendações sobre estratégias de manutenção preventiva de sistemas de acordo com as análises de confiabilidade apresentadas; e
- discussão sobre o uso do arcabouço CHESS no domínio de sistemas médicos, e sugestões de melhorias para melhor aplicabilidade a este domínio.

1.5 Metodologia

A metodologia usada neste trabalho é composta por quatro etapas, apresentadas na Figura 1.1. A primeira etapa é composta por uma Revisão Sistemática da Literatura (RSL) com foco em trabalhos relacionados a especificação de modelos formais aplicados a sistemas de aquisição de sinais biomédicos. Uma RSL é um processo metodológico composto por protocolos com o objetivo de revisar e sintetizar trabalhos de uma área específica. O objetivo com a RSL realizada foi apresentar pesquisas relacionadas com modelos formais e sistemas médicos. Essa etapa foi de suma importância, pois foi formado um embasamento teórico inicial sobre quais ferramentas já foram utilizadas para a especificação de sistemas de aquisição de sinais biomédicos.

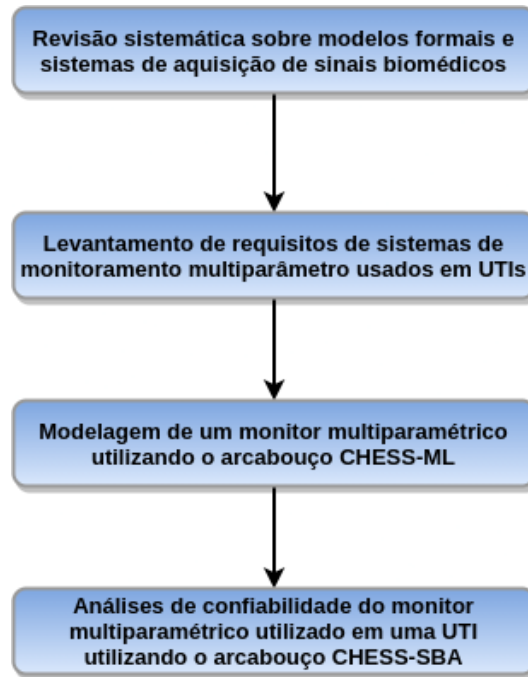


Figura 1.1: Passos metodológicos utilizados durante a pesquisa.

A segunda etapa metodológica foi caracterizada pelo levantamento de requisitos de sistemas de monitoramento multiparâmetro para utilização em ambientes de UTIs. O levantamento de requisitos foi realizado de duas formas: (1) pesquisa de trabalhos na literatura e (2) entrevista com um profissional da área de eletrônica com conhecimento em sistemas de aquisição de sinais biomédicos e com mais de quinze anos de experiência na manutenção de sistemas médicos em hospitais. Com o levantamento de requisitos, foi possível aumentar o nível de conhecimento sobre um monitor multiparâmetro, sabendo de quais componentes ele é composto, que tipo de informação é coletada e as probabilidades de falha dos componentes, utilizadas nas análises de confiabilidade.

Após o levantamento de requisitos, o terceiro passo metodológico foi a modelagem, com a metodologia CHES, de sistemas de monitoramento multiparâmetro utilizados em UTIs. Neste ponto, foi muito importante representar todos os componentes essenciais de um monitor multiparâmetro com o CHES-ML, considerando o monitoramento de um paciente.

Na quarta etapa metodológica foram executadas análises de confiabilidade com o CHES-SBA usando o modelo desenvolvido no terceiro passo. Foi analisada a confiabilidade de sistemas de monitoramento multiparâmetro em 3, 6, 9, 12, 18 e 24 meses. Posteriormente, foi possível realizar uma análise referente à manutenção preventiva destes sistemas

médicos em UTIs.

1.6 Organização do Documento

Esta dissertação de mestrado está organizada em 7 capítulos. Neste capítulo foi apresentada uma introdução sobre o trabalho com problemática, motivação, objetivos, contribuições e passos metodológicos seguidos nesta pesquisa. O restante da dissertação está organizada da seguinte maneira:

- no Capítulo 2 são apresentados conceitos e definições referentes a confiabilidade de sistemas, ao CHESS-ML e CHESS-SBA. Esses conceitos são fundamentais para a modelagem de um sistema utilizando a metodologia CHESS e para a análise de confiabilidade do modelo desenvolvido;
- no Capítulo 3 é apresentado uma RSL com foco em trabalhos relacionados na área médica utilizando modelos formais. Além disso, são apresentados outros trabalhos relacionados com esta pesquisa, mas não identificados na RSL;
- no Capítulo 4 é descrita a especificação de sistemas de monitoramento multiparâmetro com todos os seus componentes, portas de comunicação e tipagem de dados desenvolvido utilizando o CHESS-ML;
- no Capítulo 5 são apresentadas as análises de confiabilidade executadas com o CHESS-SBA no modelo apresentado no Capítulo 4;
- no Capítulo 6 são apresentadas discussões dos resultados, relacionando as análises de confiabilidade de sistemas de monitoramento multiparâmetro (para ambientes de UTIs) utilizando a metodologia CHESS; e
- no Capítulo 7 são apresentadas conclusões e direcionamentos para trabalhos futuros.

Capítulo 2

Embasamento Teórico

Neste capítulo, é apresentado o embasamento teórico dos conceitos relacionados com este trabalho. São abordados os conceitos e definições de confiabilidade de sistemas, a arquitetura da linguagem CHES-ML, detalhes do *plugin* CHES-SBA, do estereótipo FLABehavior que utiliza o *plugin* CHES-FLA, como definir estratégias de manutenção e como executar análises de confiabilidade.

2.1 Confiabilidade de Sistemas

Após o término da primeira guerra mundial, o termo confiabilidade passou a adquirir um significado tecnológico, no qual os primeiros experimentos envolvendo confiabilidade foram relacionados em comparativos entre aviões de um até quatro motores. Nestes experimentos envolvendo aviões, a confiabilidade dos aviões era medida pelo número de acidentes por hora de voo do avião [22]. Outro marco histórico que colaborou para os estudos envolvendo confiabilidade foi a corrida espacial até a lua, principalmente pelo fato de que ir até a lua envolvia a vida de humanos. Marcos como esses colaboraram para a evolução de estudos envolvendo confiabilidade, assim o surgimento da primeira associação de engenheiros de confiabilidade nos Estados Unidos em 1963 e a publicação do primeiro periódico (IEEE - *Transaction on Reliability*) influenciou para a utilização de confiabilidade em outras áreas, como, por exemplo, na análise dos riscos associados a construção e operação de usinas nucleares.

Lafraia [42] define que a confiabilidade é a probabilidade de um componente, sistema, máquina ou equipamento operar sem falhas, dentro das especificações para as quais foi pro-

jetado e construído, por um determinado período de tempo.

Todo o sistema em utilização ao qual deseja-se medir a confiabilidade, deve estar associado a um tempo de vida, que é considerado como a duração do bom funcionamento do sistema até a falha, ou duração de vida. Portanto, o tempo de vida de um sistema é representado através de uma variável com uma distribuição de probabilidade que pode ser estimada através da função de confiabilidade do componente. Como todos os valores de probabilidade que representam a confiabilidade, os valores são apresentados em intervalos entre 0 e 1 [43].

Fatores que interrompem o bom funcionamento do sistema são considerados como falhas. Uma falha em um sistema ou componente, representa a interrupção de um item desempenhar sua função esperada, neste caso um item pode representar um subsistema, um componente ou até mesmo um sistema por completo. Para confiabilidade, a falha é definida como a impossibilidade de um sistema ou componente cumprir sua função especificada [4].

Sistemas utilizados em condições ideais, ou em condições de ambiente iguais (boas condições ou má condições) podem apresentar tempos de falha distintos. Isso significa que alguns sistemas pode apresentar falhas mais cedo, enquanto outros podem demorar mais para falhar em sua duração de vida. A partir disso, estudos de tempo de falhas devem ser descritos em termos probabilísticos [24].

Para a determinação de confiabilidade de um sistema, a função operacional do sistema precisa ser bem clara durante sua definição, sendo possível determinar quais serão as formas de utilização do sistema e os possíveis modos de falha que o sistema terá. Atavés disso, o cenário que o sistema está sendo executado deve ser bem caracterizado, contendo detalhes do meio físico e ações de manutenção e operação [44].

2.1.1 Análise de Confiabilidade

Para realizar a análise do tempo de vida de um sistema, é necessário utilizar uma função de confiabilidade, que pode ser definida como a probabilidade de um sistema funcionar sem falha até um determinado período de tempo (t). Portanto, uma das principais funções probabilísticas utilizadas para representar o tempo de bom funcionamento até a falha, é representada por: $R(t) = P(T > t)$, onde $R(t)$ representa a confiabilidade no instante representado e T é uma variável aleatória que representa o tempo entre a falha do sistema [72].

Na Figura 2.1 são apresentados três gráficos nos quais são representadas as vidas úteis

de componentes similares. O eixo y dos gráficos possui a representação da confiabilidade do componente, enquanto no eixo x a representação do tempo em horas. O primeiro gráfico, da esquerda para a direita, é composto pelo tempo de vida útil de um componente que possui uma alta confiabilidade com um valor muito próximo de 1 até um tempo de 350 horas, quando este tempo é atingido a confiabilidade foi para 0, o que representa que uma falha ocorreu quando atingiu o tempo de 350 horas. O segundo e terceiro gráficos podem ser interpretados da mesma forma que o primeiro gráfico, porém, são gráficos relacionados que equiparam a vida útil de vários componentes. Com este exemplo, é possível observar que o uso de vários componentes similares em ambientes ideais ou não ideais pode apresentar valores distintos de vida útil.

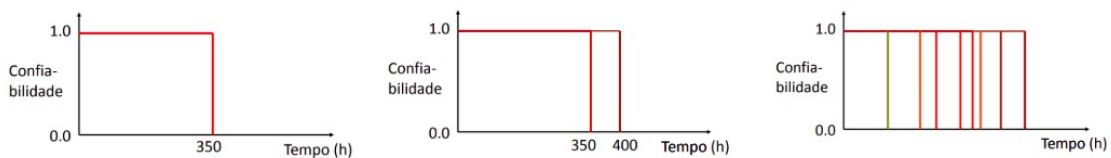


Figura 2.1: Vida útil de componentes similares, adaptado de [68].

A função de confiabilidade dos componentes referenciados na Figura 2.1 é apresentada na Figura 2.2, na qual é representada a média de vida útil de componentes similares, neste caso $R(t_0)$ é equivalente a confiabilidade (probabilidade de não falhar) no instante t_0 , ou a probabilidade de que a vida útil do componente exceda o instante t_0 .

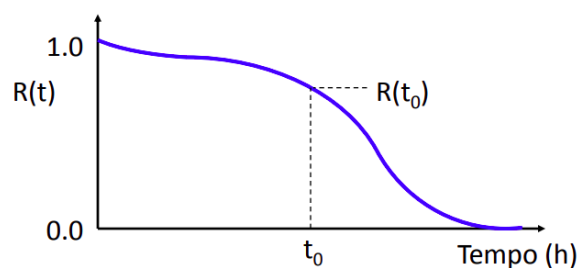


Figura 2.2: Média dos testes de vida útil de componentes similares, adaptado de [68].

O indicador de tempo médio entre falhas (*Mean Time Between Failures* - MTBF) representa a quantidade média de tempo que um componente funciona normalmente até falhar. Para o MTBF é assumido que se alguma falha ocorreu, o componente foi reparado imediatamente e voltou a funcionar normalmente [40]. Para obter o valor do MTBF é necessário

utilizar três variáveis, que são:

- **TD**: representa o tempo ideal que o componente irá funcionar, caso não seja preciso realizar uma pausa para manutenção;
- **TM**: representa o tempo de manutenção. Em outras palavras, a quantidade de tempo que o componente ficou parado devido a realização de reparos e falhas; e
- **P**: representa o número de falhas que aconteceu com o componente.

Portanto, o MTBF pode ser calculado usando a seguinte equação:

$$MTBF = \frac{(TD - TM)}{P}$$

Por exemplo, considere um componente que possui um tempo ideal de funcionamento de 24 horas, devendo funcionar durante esse tempo sem pausa para manutenção. Durante as 24 horas, o componente obteve duas falhas que levaram 30 minutos cada uma para realizar o reparo e voltar ao seu estado saudável. Pode-se calcular o MTBF deste componente da seguinte forma: $MTBF = \frac{(24 - (0,5 + 0,5))}{2}$, no qual 24 é referente a TD que representa o tempo ideal de funcionamento do componente; 0,5 + 0,5 o valor para TM que contém a soma das duas paradas de 30 minutos cada para o componente convertida em horas; e 2 o valor de P, que representa o número de paradas do componente. Portanto, o MTBF para esse componente terá um valor de 11,5 horas [48].

2.1.2 Distribuições de Probabilidade Exponencial

Existem técnicas estatísticas que necessitam de especificação de distribuição de probabilidade e que podem tornar os indicadores de confiabilidade mais precisos. Essas técnicas estatísticas podem ser chamadas de técnicas estatísticas paramétricas para o tempo de vida. Dentre essas técnicas pode-se citar a distribuição de probabilidade exponencial [36].

A distribuição de probabilidade exponencial é caracterizada pela função da taxa de falha constante, que é uma função absolutamente contínua. Portanto, duas funções podem ser utilizadas para descrever uma distribuição de probabilidade, que são: (1) função de densidade de probabilidade (*Probability Density Function* - PDF), a qual descreve a probabilidade para cada valor da função, sendo $P(X = k)$; e (2) função de distribuição acumulada (*Cumulative Distribution Function* - CDF), na qual é descrita a probabilidade para um valor $\leq k$, sendo

($X \leq k$). A CDF indica a probabilidade que o sistema tenha falhado até o tempo k . Para a análise de confiabilidade, a CDF é a distribuição mais utilizada comumente.

A representação da distribuição CDF em distribuição de probabilidade exponencial, é representada através da seguinte expressão de distribuição acumulada:

$$F(t) = 1 - e^{-\lambda t}$$

A distribuição CDF pode ser utilizada para calcular a probabilidade de falha de um sistema, no qual tem-se um determinado tempo t e uma taxa de falha do sistema ou componente informado. Por exemplo, utilizando a distribuição CDF pode-se ter a distribuição de probabilidade exponencial de um sistema com informação de taxa de ocorrência de falha de 20% de probabilidade de falhar em um intervalo de tempo de 1 ano.

O uso do MTBF na distribuição exponencial é equivalente a média da distribuição exponencial, sendo representado na expressão por:

$$MTBF = \frac{1}{\lambda}$$

Este trabalho utiliza valores para representar a taxa de ocorrência de falha de componentes com distribuição de probabilidade exponencial através de MTBF e CDF. Existem outras distribuições de probabilidade que são utilizadas no contexto de confiabilidade como Weibull e Log-normal. Entretanto, neste trabalho só será utilizada a distribuição de probabilidade exponencial.

2.2 CHESSE - Modeling Language

A metodologia CHESSE oferece suporte para a modelagem de diferente níveis: arquitetura de sistemas, *software* e *hardware*. Para isso, é utilizado o CHESSE-ML, uma linguagem gráfica composta por alguns componentes de SysML e UML. Para SysML, são utilizados: diagramas de definição de blocos e diagramas de blocos internos, ambos aplicados para a modelagem de visão e arquitetura de sistema. Para UML, são utilizados os componentes de diagramas de classes, diagramas de estrutura e diagramas de estrutura composta, aplicados para a modelagem de arquitetura de *software* e de *hardware*.

2.2.1 Arquitetura do Sistema

Na arquitetura de sistema, são necessárias algumas definições iniciais. É preciso criar um diagrama de definição de blocos para definir os blocos que representam a arquitetura do sistema. Neste nível de arquitetura, componentes são modelados utilizando os elementos `blocks` de SysML.

Adicionando todos os blocos do sistema a serem modelados na visão de sistema, não é necessário criar várias cópias dos blocos modelados previamente. Os blocos modelados podem ser instanciados várias vezes. Todos os blocos podem conter portas, que podem ser representadas utilizando um componente chamado de `flowport`. Ao aplicar `flowports`, é possível realizar a interação de dados entre os blocos, como, por exemplo, a entrada e/ou saída de informações entre os blocos. As `flowports` possuem uma propriedade chamada de `direction`, que pode receber três valores: `in`, `out` ou `inout`. Com isso, o modelador pode definir o tipo de informação trafegada entre os blocos modelados e com as portas de comunicação definidas. No modelo apresentado na Figura 2.3, é ilustrada a especificação de um sistema detector do nível da água de um rio. É possível observar um componente que representa um sensor de detecção da água de um rio (`RiverSensorDetector`), um componente que representa o processamento dos dados do sensor (`RiverDetector`), um tipo de dado `Alarm` e um componente de mais alto nível desse modelo representado na cor roxa (`RiverDetectorSystem`), que irá conter um diagrama de blocos internos para representar as instâncias e comunicação dos blocos através das portas. Todas as imagens contidas nesta seção foram criadas desenvolvendo um modelo com o arcabouço CHESS baseado nas documentações contidas em [52] e [1].

Após definir todos os blocos necessários a nível de sistema e a comunicação entre os blocos com as portas, é necessário definir um diagrama de blocos interno para representar a comunicação definida no diagrama de blocos, e, portanto, definir o comportamento de blocos. Com o diagrama de bloco interno criado, pode-se referenciar os blocos definidos no diagrama de blocos, adicionando referências como `parts` para os blocos definidos no diagrama de blocos. Na Figura 2.4, é ilustrada a declaração de um diagrama de bloco interno, continuando o exemplo apresentado na Figura 2.3.

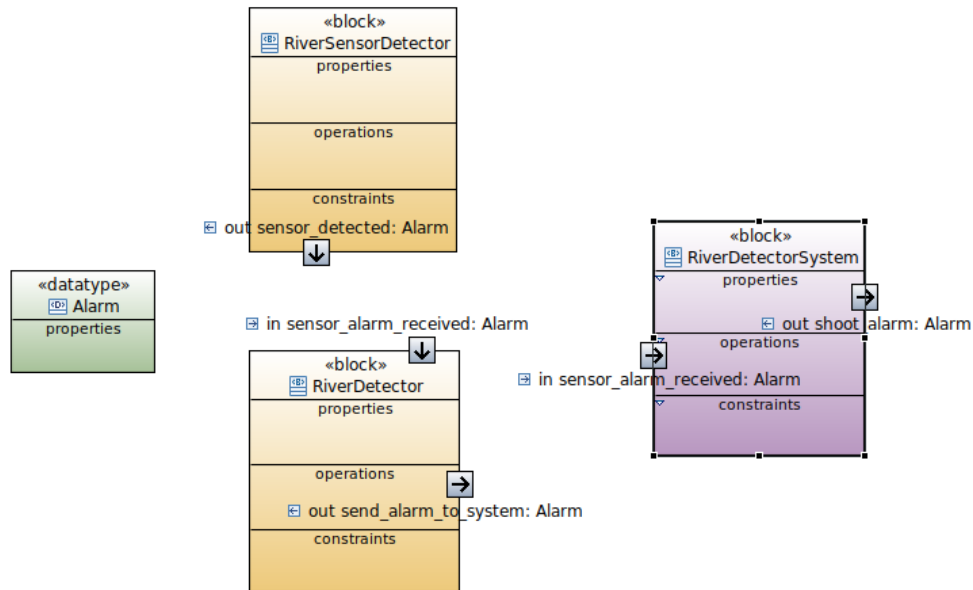


Figura 2.3: Definição de blocos e portas de comunicação a nível de sistema, adaptado de [1].

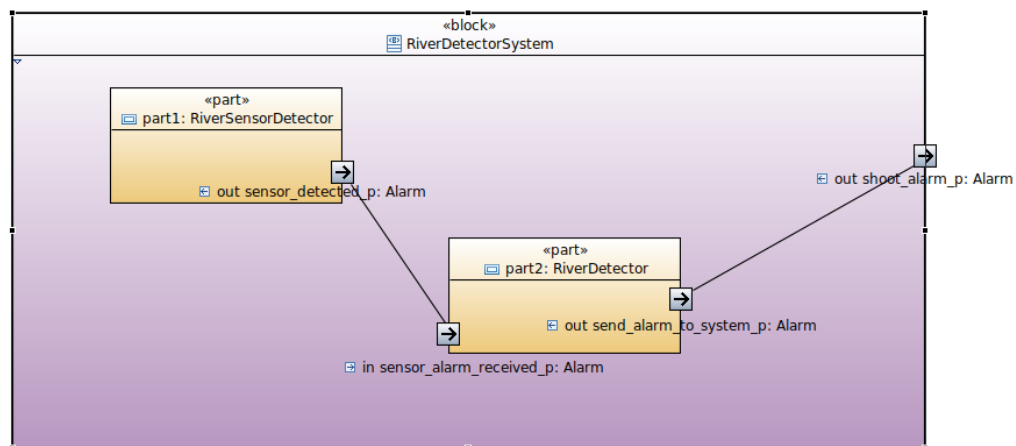


Figura 2.4: Definição de diagrama de blocos internos, adaptado de [1].

2.2.2 Arquitetura do Software

A modelagem da arquitetura de *software* com o CHESS-ML possui uma abordagem similar ao diagrama de classes UML para realizar a modelagem de um conjunto de componentes disponíveis. Também é possível definir diagramas de estrutura composta UML para cada componente definido. Para definir modelos com o CHESS nesse nível, é necessário considerar interfaces e tipos de componentes, implementações de componentes e composições.

Para definir interfaces, tipos de componentes e implementações de componentes, é ne-

cessário utilizar o diagrama de classes, que possibilita a comunicação dos componentes através das interfaces definidas. Em um diagrama de classes UML utilizando o CHESS-ML, uma ordem de definição de componentes deve ser seguida: (1) definir as interfaces, (2) definir os tipos de componentes, (3) definir as implementações dos componentes e (4) definir as relações entre os componentes e interfaces. Para definir um tipo de componente, deve-se adicionar um componente com o estereótipo `ComponentType`. Para definir uma implementação de componente, deve-se adicionar um componente com o estereótipo `ComponentImplementation`. Para toda implementação de componente, é necessário referenciar qual o seu tipo, que deve possuir uma relação direta com o estereótipo `ComponentType`.

Na Figura 2.5 são ilustradas as definições de algumas interfaces, componentes e implementações de componentes. Todas as implementações de componentes possuem um relacionamento de realização com o componente ao qual possui a implementação. Um relacionamento de realização é necessário, neste caso, dado que o componente que terá a implementação possui um conjunto de informações para executar os comportamentos especificados. Na Figura 2.5, o componente chamado de `Manager` possui um relacionamento de dependência com três interfaces, as quais são necessárias para o funcionamento do modelo. Estas interfaces possuem relacionamentos de realização com algum componente que possui informações para a sua implementação.

Para realizar a comunicação de forma explícita da arquitetura de *software* que foi definida utilizando o diagrama de classes, é necessário criar um diagrama de composição para cada tipo de componente (`ComponentType`) e definir portas de comunicação. O tipo da porta é a interface definida com a qual o tipo de componente possui uma relação. São definidos quais interfaces o componente deve requerer ou prover. A comunicação definida para cada componente é herdada na implementação do componente com a qual possui relação.

Com cada tipo de componente, possuindo uma definição de composição através do diagrama de composição, é possível definir um diagrama de composição para todo o modelo (interfaces, tipos de componentes e implementação de componentes) definido no diagrama de classes. Para isso, um componente que representa a comunicação deve ser definido no diagrama de classe. Posteriormente, deve ser criado um diagrama de composição.

No diagrama de composição devem ser adicionadas as referências das implementações

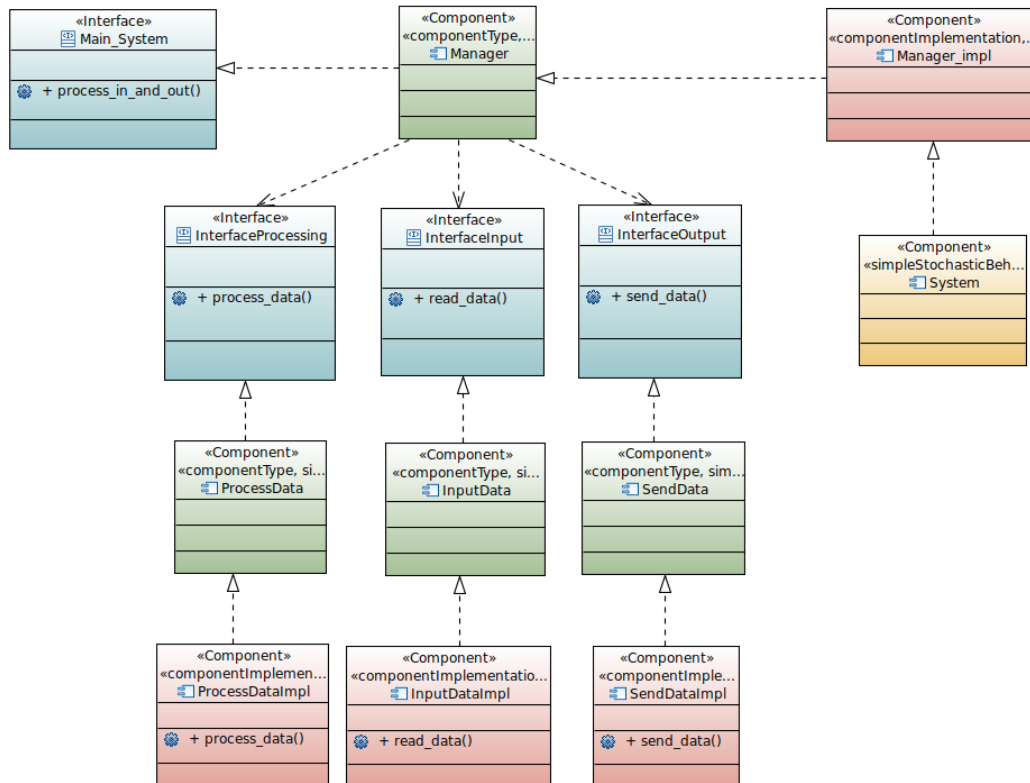


Figura 2.5: Definição de componentes, interfaces e implementações de componentes, adaptado de [52].

dos componentes definidas no diagrama de classe. Essas referências são do tipo `parts`, que formam a estrutura interna do diagrama de composição. Portanto, o diagrama de estrutura composta define como as instâncias do componente são conectadas. Na Figura 2.6 é apresentada a definição do diagrama de composição para o modelo apresentado na Figura 2.5, utilizando as implementações dos componentes como `parts` do diagrama de composição.

2.2.3 Arquitetura do Hardware

A modelagem da arquitetura de *hardware* é similar à modelagem de arquitetura de *software*. Utilizando a visão de desenvolvimento, o primeiro passo consiste em criar um diagrama de classes, no qual é definido cada componente de *hardware*. Os componentes de *hardware* são modelados como elementos UML do tipo `Component`, onde o mesmo componente pode ser instanciado várias vezes. Como na modelagem de *software*, os componentes na modelagem de *hardware* também possuem portas de comunicação, que podem

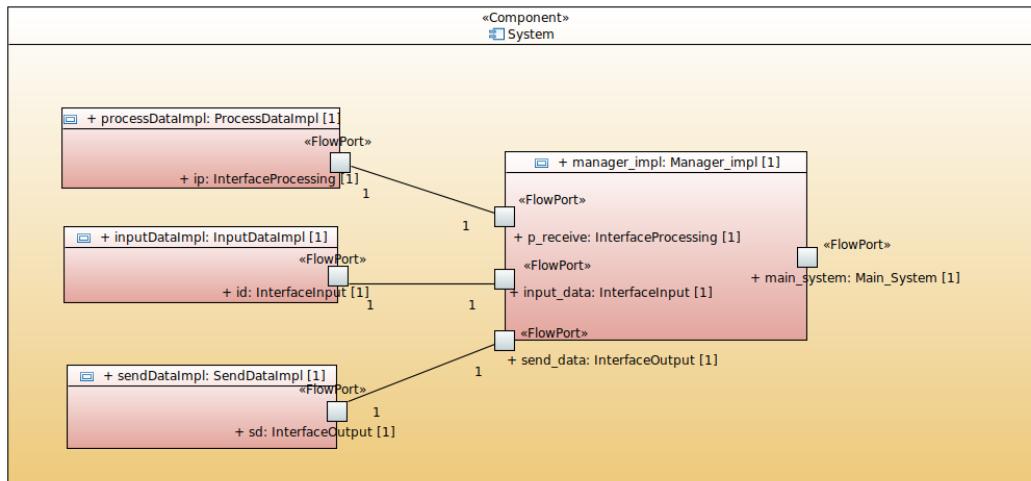


Figura 2.6: Definição do diagrama de estrutura composta, adaptado de [52].

ser modeladas utilizando os `flowports`, contendo a propriedade de `direction` para determinar a direção das informações trocadas entre os componentes. As portas no nível de *hardware* não precisam de uma interface associada explicitamente. Portanto, a modelagem se torna mais simples em relação à arquitetura de *software*.

Após definir os blocos que formam a arquitetura em nível de *hardware*, é necessário detalhar a estrutura interna, explicitando como os componentes são organizados e conectados. Neste contexto, deve-se criar um diagrama de estrutura composta para cada componente que deve ser detalhado e adicionar as instâncias dos outros componentes definidos anteriormente como um `part` do elemento composto de maior nível hierárquico (e que representa todo o comportamento do que foi definido no diagrama de classes).

2.2.4 Geração de Instâncias

Os diagramas de estrutura composta suportam apenas a especificação de um nível de composição, não sendo possível modelar no mesmo diagrama, por exemplo, a seguinte situação: existem os componentes A e C, na qual o componente A é uma composição de B e C, e o componente C é uma composição de D e E. Caso seja necessário realizar esse tipo de modelagem, é necessário criar dois diagramas de estrutura composta separados, no qual, no primeiro diagrama é modelado a estrutura interna do componente A, e no segundo diagrama é modelado a estrutura interna do componente C.

A metodologia CHESS possui uma funcionalidade para gerar de forma automática a árvore completa de instâncias a partir de um conjunto de diagramas de bloco interno. Esta funcionalidade é chamada de `Build Instances` e está disponível para os elementos com estereótipo `Component` e `Block`. Para gerar as instâncias a partir de um conjunto de diagramas de bloco, o componente que representa o diagrama de bloco interno deve possuir o estereótipo `CHGaResourcePlatform` do CHESS-ML atribuído. Normalmente, o estereótipo `CHGaResourcePlatform` é atribuído aos elementos de níveis superiores do sistema, *software* e *hardware*.

Utilizando o recurso de `Build Instances` da metodologia CHESS, é retornado como resultado, no mesmo modelo, um novo pacote UML contendo a árvore de instâncias com o elemento com o estereótipo `CHGaResourcePlatform` como raiz. Este novo pacote contém elementos do tipo `InstanceSpecification` gerados a partir da composição de diagramas de estrutura composta ou diagramas de blocos internos. As instâncias são geradas para `blocks` ou `components`, bem como para conectores que os interligam.

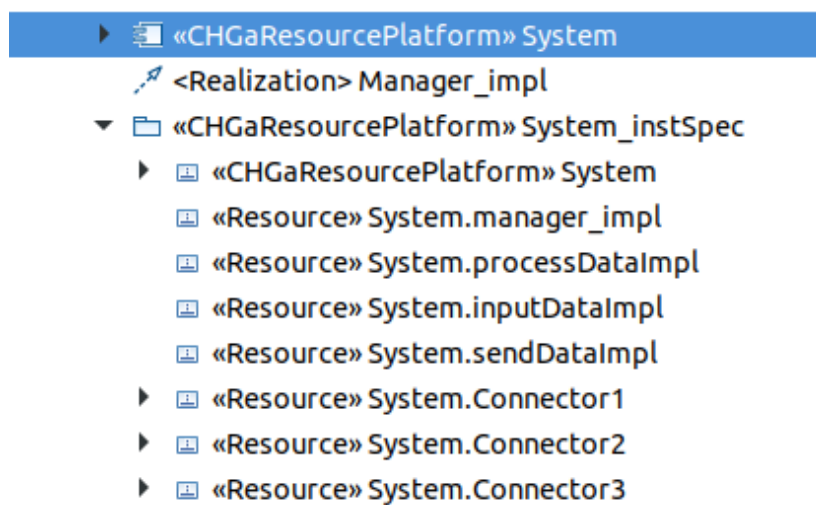


Figura 2.7: Pacote de instâncias gerados através do recurso `Build Instances`, adaptado de [52].

Na Figura 2.7 é apresentado um exemplo de pacote de instâncias geradas usando a funcionalidade `Build Instances` do CHESS com o componente `System`, que é o componente de maior nível hierárquico do modelo. Gerando instâncias, é possível verificar se a estrutura pretendida da arquitetura do sistema, *software* ou *hardware* está correta. A execução do recurso `Build Instances` é um pré-requisito para a execução do CHESS-SBA,

além de outras técnicas de análise.

2.2.5 Alocações

A metodologia CHESS permite que o usuário que está utilizando o arcabouço possa especificar a alocação de *software* ao *hardware*. Isso pode ser utilizado para vários propósitos em diferentes técnicas de análise. O CHESS-SBA é uma das formas utilizadas para verificar caminhos de propagação.

Em cada instância do componente de *software*, a alocação especifica a instância do componente de *hardware* na qual está alocada, criando conexões entre o *software* e as arquiteturas de *hardware*. Para realizar esta alocação utilizando o CHESS-ML, é necessário utilizar o UML Comment com estereótipo Assign. O atributo `from` aceita um conjunto de instâncias de *software*, enquanto o atributo `to` aceita uma única instância de *hardware*. Os comentários de alocação são anexados ao componente de nível superior da arquitetura de *hardware*.

A alocação só funciona em instâncias de componentes. Os campos de `from` e `to` devem conter elementos de `InstanceSpecification` gerados usando o recurso `Build Instances`. Portanto, o comando `Build Instances` deve ser executado, anteriormente, para o componente de *software* de nível superior e para o componente de *hardware* superior.

2.3 CHESS - State Based Analysis

O CHESS-SBA é um *plugin* da metodologia CHESS que oferece suporte para a realização de análises de confiabilidade do sistema que foi modelado utilizando CHESS-ML. O termo "*state based*", cuja tradução literal para português é "baseado em estados", refere-se ao fato de que as representações do sistema para confiabilidade são realizadas com base nos estados relevantes para as análises de confiabilidade e as possíveis transições entre os estados [50].

Ao modelar um componente com o CHESS-ML e executar o CHESS-SBA, são extraídas as informações necessárias para as análises de confiabilidade, sendo gerada uma rede de Petri estocástica [9] representando o seu comportamento. A rede de Petri estocástica é gerada

automaticamente ao utilizar a ferramenta CHESS-SBA, abstraindo totalmente do usuário a especificação de uma rede de Petri manualmente. O fato do CHESS-SBA gerar a rede de Petri estocástica automaticamente denota uma vantagem, dado que o usuário não precisa dedicar tempo para aprender rede de Petri. Portanto, o desenvolvimento de modelos com rede de Petri estocástica esta fora do escopo deste trabalho. Após a análise por simulação da rede de Petri estocástica gerada, métricas são anotadas como resultados no modelo original do CHESS em SysML, permanecendo disponível para o usuário visualizar e realizar interpretações.

Como é possível observar na Figura 2.8, um componente modelado usando a metodologia CHESS pode possuir dois estados: (1) o estado saudável, que representa o estado que o componente está funcionando de forma correta; e (2) o estado de falha, que representa um componente que forneceu um serviço incorreto. Com a geração automática do modelo formal, este tipo de detalhe é abstraído do modelador. O sistema deve iniciar no estado saudável, e, após um determinado período de tempo, caso algo não ocorra como planejado, o sistema transita para o estado de falha. Posteriormente, o sistema pode retornar para o estado inicial. Portanto, a confiabilidade do modelo pode ser medida usando a probabilidade de um componente não falhar em um determinado tempo t [52].

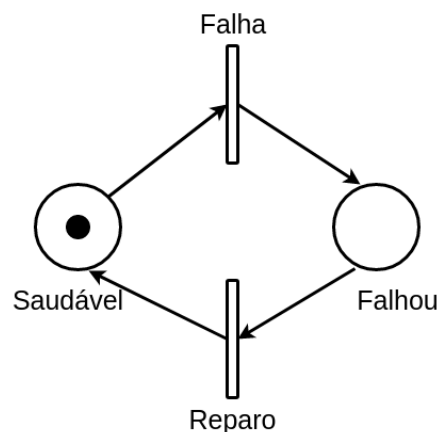


Figura 2.8: Estados de um componente no CHESS, adaptado de [52].

Utilizando a abordagem de análise baseada em estados, diferentes métricas podem ser computadas. O CHESS-SBA implementa três métricas para a execução de análises: confiabilidade, que é a probabilidade de continuar no estado saudável, partindo de 0 até um

determinado tempo t ; disponibilidade, que é a proporção de tempo em que o sistema fica em estado saudável em um determinado intervalo de tempo [entre 0 e t]; e probabilidade de falha na demanda, que é a probabilidade do sistema falhar em uma demanda para fornecer o serviço solicitado. Para o escopo desse trabalho, será apenas utilizada a métrica de análise de confiabilidade [54].

É possível modelar, de forma mais específica, outros estados adicionais para um componente. Por exemplo, além dos estados que representam o estado saudável e o estado de falha, o estado de falha pode conter mais detalhes nas transformações realizadas, especificando como a falha se originou, onde se originou e se esta falha se propagou para outros componentes.

Com o CHESS-SBA, as transições entre os estados especificados (estado de falha ou estado saudável) ocorrem com base nos valores das probabilidades e do tempo de atraso das redes estocásticas geradas. Uma falha em um sistema pode representar uma manifestação de comportamento incorreto de um componente ao qual foi especificado.

Ao definir um modelo e executar análises de confiabilidade com o CHESS-SBA, algum componente do modelo pode ter um comportamento fora do padrão, podendo ocasionar uma falha. Em uma implementação real de um *software* fora de um ambiente de modelagem, como, por exemplo, o CHESS, o comportamento fora do padrão pode se dar, por exemplo, por uma implementação feita de maneira incorreta, trecho de código defeituoso ou instrução incorreta. A simulação de análise de confiabilidade com o arcabouço CHESS através da rede de Petri estocástica gerada ao notar uma inconsistência no componente desenvolvido, detecta um defeito, o que pode fazer com que um erro seja gerado em tempo de execução. A partir de um estado de erro, uma falha pode ser gerada e propagada para os componentes externos. No exemplo apresentado na Figura 2.9, pode-se observar, no Componente A (cor azul), a origem de um defeito e a evidência do erro para gerar uma falha. Com a falha gerada no Componente A, esta falha é propagada para o Componente B (cor amarela) através da interface de comunicação entre esses componentes. A falha é propagada do Componente A para o Componente B, e dentro do Componente B, a falha propaga-se dentro do componente como uma falha externa, que irá afetar o funcionamento correto do componente B.

Com a modelagem já desenvolvida utilizando o CHESS-ML, é possível iniciar o processo de verificação de análises baseadas em estados. As análises de confiabilidade, por

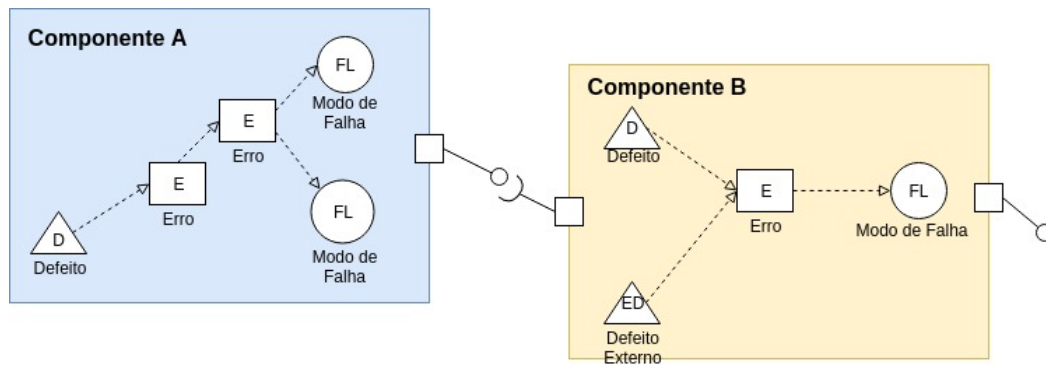


Figura 2.9: Exemplo de modelagem da propagação de uma falha em cadeia, adaptado de [52].

exemplo, não necessitam de um modelo completo do sistema, bastando apenas uma modelagem parcial de algum componente. Para iniciar a análise a partir do modelo desenvolvido, é necessário seguir alguns passos, que são: (1) enriquecer os elementos com informações de propriedades de confiabilidade; (2) definir informações sobre manutenções periódicas de atividades; (3) definir análise de contexto (qual parte do sistema será analisado pela ferramenta); (4) definir métricas de interesse; atribuir valores para os parâmetros do modelo; e (5) executar as análises.

O CHESS oferece suporte para notações científicas. Por exemplo, o número 0.000001 pode ser representado por 1.0E-6. As distribuições de probabilidade podem ser representadas por *MARTE Value Specification Language (VSL) Syntax*, que é uma extensão de UML que fornece suporte para especificação, projeto e verificação. O MARTE consiste em fundamentos para a descrição baseada em modelos de sistemas embarcados ou em tempo real. O VSL, que é uma linguagem de especificação de valores, possui sintaxe específica para receber valores e aplicar análises [18]. O CHESS suporta distribuições de probabilidade, como, por exemplo: exponencial, determinística, uniforme, normal, *gamma* e *weibull*.

Para realizar o enriquecimento das informações de confiabilidade com o CHESS, a metodologia CHESS oferece três estereótipos de componentes que podem ser aplicados aos elementos do modelo utilizando os blocos do SysML ou UML. Esses três estereótipos são: `SimpleStochasticBehavior`, `FLABehavior` e o `ErrorModelBehavior`. Para a modelagem, os dois primeiros estereótipos podem coexistir no mesmo bloco/componente. Outras combinações, como, por exem-

pl, `SimpleStochasticBehavior` e `ErrorModelBehavior`, ou `FLABehavior` e `ErrorModelBehavior`, não são permitidas para o mesmo componente.

Nas próximas seções serão apresentados os estereótipos disponíveis no CHESS-SBA para que o modelador adicione informações de confiabilidade nos componentes desejados do sistema. Além disso, é também apresentado como o CHESS-SBA pode funcionar com manutenção e métricas utilizadas para executar e coletar análises de confiabilidade.

2.3.1 *SimpleStochasticBehavior*

O `SimpleStochasticBehavior` é uma maneira simples de adicionar informações de confiabilidade em um componente definido em uma modelagem de *software* ou sistema utilizando o CHESS-ML. Desta forma, o componente só pode ser afetado por apenas um tipo de defeito interno, fazendo com que o componente falhe de imediato. O estereótipo `SimpleStochasticBehavior` possui três atributos, que são: `failureOccurrence`, `failureModesDistribution` e `repairDelay`.

O atributo `failureOccurrence` é o único que obrigatoriamente necessita de um valor para a utilização deste estereótipo. Com esse atributo, é possível especificar o tempo para a ocorrência de uma falha do componente. Para atribuir um valor a este atributo variável, deve-se utilizar uma distribuição de probabilidade.

Com o atributo `failureModesDistribution`, é possível definir modos de falha do componente, quando este componente entra em estado de falha, usando a probabilidade adicionada na variável `failureOccurrence`. Ao definir os modos de falha, deve-se adicionar a informação de probabilidade do modo de falha acontecer, caso o componente falhe. Por exemplo, caso a probabilidade de falha definida na variável `failureOccurrence` seja de $1.0E-5$ e a variável `failureModesDistribution` possua o valor `omission:0.9, valueSubtle:0.1`, se o componente falhar, o componente possui 0.9 de probabilidade de emitir um modo de falha do tipo `omission` e 0.1 de emitir um modo de falha do tipo `valueSubtle`.

Com o atributo `repairDelay`, é possível criar estratégias relacionadas com a manutenção de componentes ou sistemas no geral, sendo possível definir o atraso de reparo do momento que o componente falhe até o momento em que o componente pode ser restaurado para o seu estado original saudável. Na Figura [2.10](#), é apresentado

um componente com informações de confiabilidade configuradas utilizando o estereótipo `SimpleStochasticBehavior` e a configuração das variáveis do estereótipo. O componente `RiverDetector` possui uma probabilidade de falha de $1.0E-5$, um atraso de reparo de 0.1 e dois modos de falha (o modo de falha `omission` com probabilidade de 0.9 e o modo de falha `valueSubtle`, com probabilidade de falha 0.1).

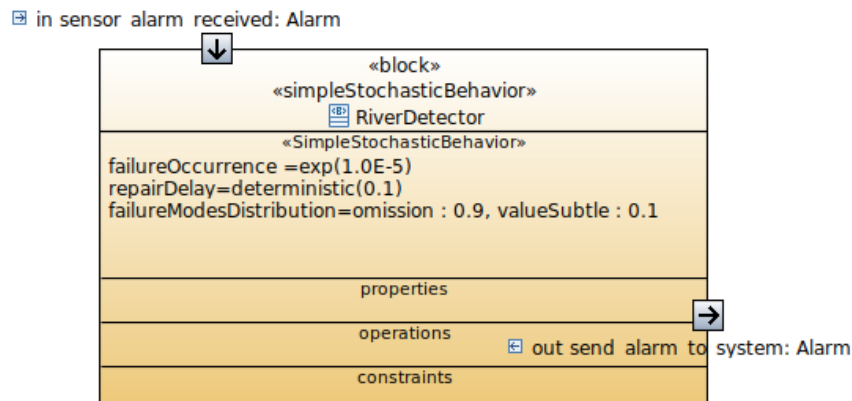


Figura 2.10: Componente com informações de confiabilidade adicionadas utilizando o estereótipo `SimpleStochasticBehavior`, adaptado de [11].

2.3.2 FLA Behavior

Com o estereótipo `FLA Behavior`, é possível definir o comportamento de falha dos componentes em termos de lógica de falha. Portanto, é possível especificar como os componentes podem propagar ou transformar os modos de falha, sendo a maneira mais conveniente de especificar o comportamento da propagação entre componentes e redundância. Este estereótipo é utilizado principalmente para a realização de análises utilizando o CHESS-FLA, mas também pode ser processado pelo CHESS-SBA.

O estereótipo `FLA Behavior` possui apenas um atributo para definir o comportamento da falha dos componentes, chamado de *Failure Propagation Transformation Calculus* (FPTC). Uma expressão que representa o cálculo de transformação de propagação de falha necessita de informações como as portas, os modos de falha que a porta pode receber e qual a porta de saída e o modo de falha para essa porta de saída. Na Figura 2.11, pode-se observar um componente que possui três portas, duas portas de entrada nomeadas de `in1` e `in2` e

uma porta de saída nomeada de `out`. Com o estereótipo `FLABehavior` configurado, o valor adicionado na variável, para a expressão `FPTC`, significa que, se a porta `in1` receber um modo de falha `omission` e a porta `in2` receber um modo de falha `valueSubtle`, a porta `out` apresentará um modo de falha `omission`, o que será propagado para o componente conectado a porta `out` do componente `Detector`.

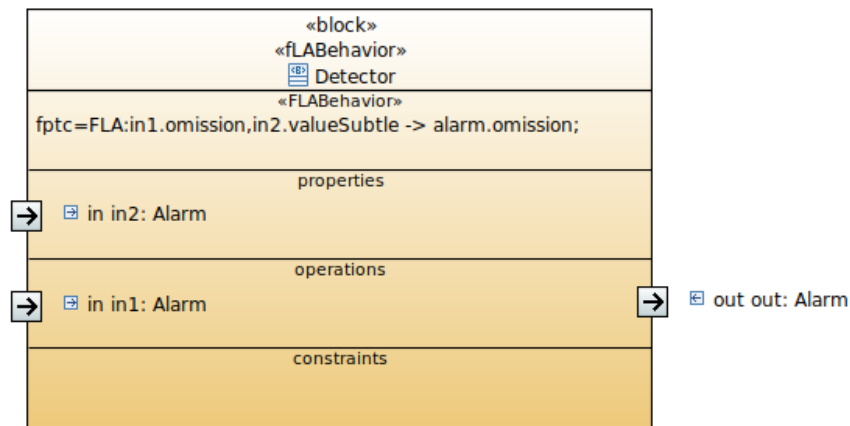


Figura 2.11: Componente com informações sobre o comportamento de falha utilizando o estereótipo `FLABehavior`, adaptado de [1].

Usando o `FLABehavior` e o `CHESS-FLA` [25] [26], o modelador pode manipular a especificação qualitativa do comportamento dos componentes de forma individual, em termos de falhas de componentes e suas causas, dando suporte a automatizar, parcialmente, a síntese da *Fault Tree Analysis* (FTA) e da *Failure Mode and Effects Analysis* (FMEA) [13].

2.3.3 *ErrorModelBehavior*

O `ErrorModelBehavior` permite o fornecimento de mais detalhes sobre as falhas e erros que podem acontecer, a nível de *software*, *hardware* e sistema. Os detalhes das falhas podem ser especificados utilizando um tipo específico de máquina de estados chamado de `ErrorModel`. Portanto, o `ErrorModel` contém informações sobre propagação de falhas/erros em um determinado componente ou bloco.

Para modelar um comportamento de falha com o `ErrorModelBehavior`, é necessário criar um diagrama de máquina de estados e adicionar o estereótipo de `ErrorModel` no bloco que representa a modelagem de falha. Com o estereótipo `ErrorModel`, é possível

identificar uma máquina de estados como modelo de erro do CHESS, que não possui atributos. Como em máquina de estados convencionais, um modelo de erro deve possuir um estado inicial, que irá representar o estado inicial íntegro do componente.

Após definir o estado inicial da máquina de estados, é necessário criar transições para representar o caminho e propagações entre os estados definidos na máquina de estados. O arcabouço CHESS fornece estereótipos que podem ser adicionados as transições para representar comportamentos desejados de acordo com o contexto na especificação de máquina de estados. A seguir, são apresentados os três estereótipos que podem ser adicionados as transições em uma máquina de estado com o CHESS-SBA.

O estereótipo `InternalFault` pode ser adicionado em uma transição para definir a probabilidade de ativação de defeito. Com o atributo `occurrence`, é possível especificar como uma distribuição de probabilidade o tempo para a ocorrência da falha no componente. O estereótipo `InternalPropagation` pode representar a propagação de um erro interno que ocorre no componente. A propagação pode ocorrer após um período de tempo ou por falhas externas. Os atributos que representam esses comportamentos para o `InternalPropagation` são: `delay` e `externalFaults`. Caso o atributo `delay` não seja definido, o CHESS-SBA considera a propagação imediata com um valor de `delay` igual a zero. Um outro atributo do `InternalPropagation` é o atributo `weight`, que pode ser utilizado em momentos que duas ou mais transições sejam habilitadas ao mesmo tempo. Quando isto ocorre, os pesos definidos para a bifurcação irão definir o destino que a transição alcançará.

O estereótipo `Failure` representa a ocorrência de falha de um componente com propagação de erros para os componentes externos usando portas. Ao ocorrer a transição, pode-se fazer com que o componente se mova para outro estado ou para o mesmo. A representação da propagação de erros para componentes externos pode ser representada com o atributo `mode`. Em uma transição com o estereótipo `Failure`, o atributo `mode` define quais portas são afetadas quando uma falha ocorre. Essa especificação deve conter a porta de saída e o modo de falha.

Uma máquina de estados contém estados para identificar um estado atual do sistema. Com o CHESS, pode-se identificar um estado de erro adicionando o estereótipo `ErrorState`. Ao adicionar um estado no diagrama e o estereótipo de `ErrorState`,

é representado que, durante a execução de análises de confiabilidade com o SBA, se o estado for alcançado, será encontrado um estado de erro.

Na Figura 2.12 é ilustrado um diagrama de máquina de estados chamado de `AlarmDetectorErrorModel`, que utiliza o estereótipo `ErrorModel`. Neste diagrama pode-se verificar a definição de estados de erro, início da máquina de estado em fase íntegra e também as transições com os estereótipos `internalFault`, `internalPropagation` e `Failure`. O diagrama é iniciado com um estado inicial de integridade chamado de `initial`. A partir do estado inicial, existe uma transição com o estereótipo `internalFault`, que possui uma probabilidade de ocorrência de $\exp(1.0E-4)$. Essa transição tem como destino um estado de uma escolha. A partir deste ponto, o destino pode ser dois estados de erro: `NoDetected` e `AlarmFault`. O estado de erro `AlarmFault` possui um peso maior, com o valor de 0.6 atribuído ao atributo `weight` do estereótipo `InternalPropagation` da transição que conecta a escolha com o estado de erro `AlarmFault`. O estado de erro `NoDetected` tem um peso menor, com o valor de 0.4 atribuído ao atributo `weight` do estereótipo `InternalPropagation` da transição que conecta a escolha ao estado de erro `NoDetected`. Os estados de erro `AlarmFault` e `NoDetected` possuem cada um, uma transição de falha para ele mesmo, que representa a propagação de erros para as interfaces externas. A transição com o estereótipo `Failure` para o estado de erro `AlarmFault` emite uma falha pela porta do componente que o `ErrorModel` está atribuído, nomeada de `out` e do modo de falha `omission`. O estado de erro `NoDetected` emite uma falha para os componentes externos pela porta `out` e do modo de falha `valueSubtle`.

2.3.4 Manutenção de Componentes

Com o CHESS-SBA, é possível simular políticas de manutenção para aumentar a confiabilidade do sistema modelado. Políticas de manutenção podem ser configuradas quando um componente possui o estereótipo `SimpleStochasticBehavior`. Utilizando o atributo `repairDelay`, é possível especificar o atraso probabilístico. Posteriormente, o componente é restaurado para o seu estado de integridade. Esta configuração pode ser bastante útil, porém, possui limitações relacionadas com configurações de manutenção mais avançadas.

O CHESS-SBA fornece uma estratégia para realizar a configuração de manutenções mais

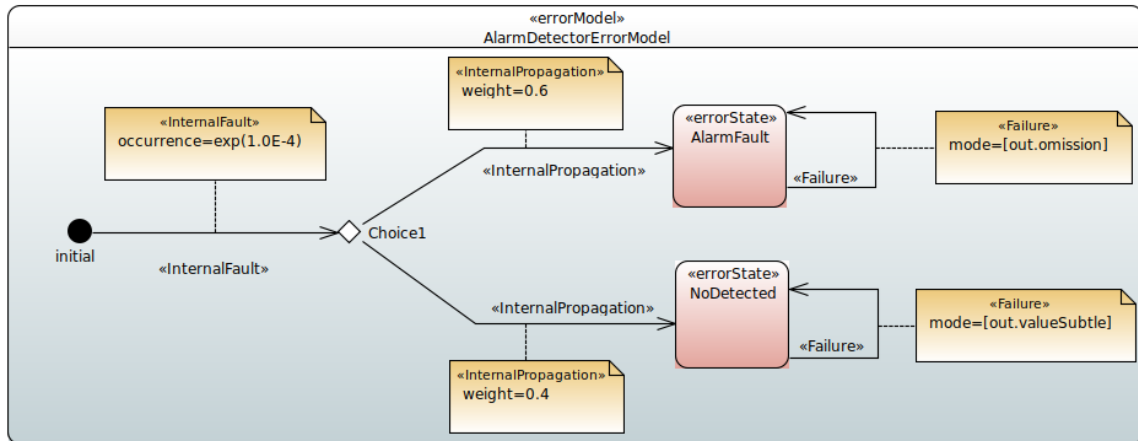


Figura 2.12: Diagrama de Máquina de Estados com o estereótipo ErrorModel

avançadas. Essa estratégia é baseada no conceito de atividade, na qual uma estratégia de manutenção é uma coleção de atividades que são executadas quando condições são satisfeitas. A modelagem de manutenção mais detalhada é utilizada para componentes que possuem os estereótipos `ErrorModelBehavior` ou `FLABehavior`, que são estereótipos que não possuem atributos para a realização de reparos.

Utilizando a visão de `DependabilityView`, é possível criar um diagrama de estrutura composta UML e definir um bloco de atividade. Para um bloco de atividade, deve-se adicionar o estereótipo `Repair` para definir estratégias de manutenção mais avançadas. O estereótipo `Repair` é o único estereótipo desenvolvido para suportar transformações referentes à manutenção. O estereótipo `Repair` representa uma atividade de manutenção que faz com que uma instância de componente retorne ao seu estado de integridade original. Com a estratégia de manutenção utilizando o estereótipo `Repair`, é possível simular a substituição de um *hardware* ou a reinicialização de um *software*. Além disso, é possível representar manutenções preventivas por regras que podem ser definidas com atributos do estereótipo.

O estereótipo `Repair` possui 4 atributos, que são: `targets`, `when`, `duration` e `probSuccess`. O atributo `targets` está relacionado com instâncias dos componentes/blocos que serão reparadas pela atividade de manutenção configurada. O atributo `when` está relacionado com políticas para a execução da atividade de manutenção. As políticas de manutenção para o atributo `when` podem ser definidas em uma periodicidade em intervalos de tempo, em detecção de falhas com modos de falhas, em detecção de estados de erro, en-

tre outros. O atributo `duration` representa o tempo necessário para realizar a atividade de manutenção. O atributo `probSuccess` representa a probabilidade de que a atividade seja bem sucedida levando as instâncias do estado de falha para o seu estado de integridade original. Os atributos `probSuccess` e `duration` são opcionais.

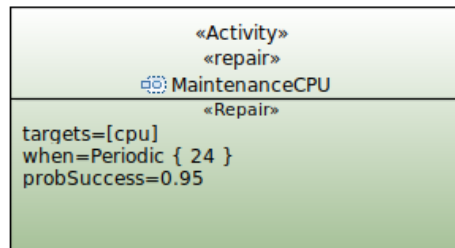


Figura 2.13: Componente do tipo `Activity` com o estereótipo `Repair` para simular estratégias de manutenção com o CHESS-SBA, adaptado de [52].

Na Figura 2.13 é apresentado um componente do tipo `Activity` definido em um diagrama UML de estrutura composta. No diagrama UML de estrutura composta, foi definido o componente de atividade `MaintenanceCPU`. No componente `MaintenanceCPU`, foi adicionado o estereótipo `Repair` e algumas definições foram atribuídas aos atributos do estereótipo. O atributo `targets` recebeu a instância `CPU` para representar o componente que irá receber a manutenção. O atributo `when` recebeu a regra de periodicidade que será executada a cada 24 horas. O atributo `probSuccess` recebeu o valor de 95%, que representa que a atividade de manutenção na CPU com periodicidade de execução a cada 24 horas possui 95% de chance de ser bem sucedida.

2.3.5 Execução de Análises e Métricas

O estereótipo `StateBasedAnalysis`, que é utilizado na visualização de `DependabilityView`, tem a responsabilidade de executar as análises do sistema modelado com os os componentes que possuem informação de probabilidade de falha através dos estereótipos apresentados nas seções anteriores. Para configurar a execução de análise de um sistema modelado com o estereótipo `StateBasedAnalysis`, é necessário atribuir valores a alguns atributos do estereótipo.

O estereótipo `StateBasedAnalysis` compreende os atributos `platform`,

`measure`, `measureEvaluationResult`, `targetDepComponent`, `targetPort` e `targetFailureMode`. O atributo `platform` representa o sistema em que serão executadas as análises, que devem ser um pacote de instâncias geradas a partir do comando `Build Instances`. O atributo `measure` é a definição da métrica que será avaliada no sistema. Neste trabalho, a métrica usada é a confiabilidade (*Reliability*). O atributo `measureEvaluationResult` é um atributo de somente leitura, que contém o resultado da análise adicionada automaticamente após a execução do CHESS-SBA. Os atributos `targetDepComponent` e `targetPort` são os alvos para a execução de análises, para a instância de um componente e para a porta de um componente. Por fim, o atributo `targetFailureMode` tem como alvo um modo de falha que foi atribuído através de um estereótipo no componente definido no atributo `targetDepComponent`.

Para utilizar o estereótipo `StateBasedAnalysis`, é necessário definir a análise de contexto de quais métricas serão aplicadas e quais são os objetivos, apenas para as análises de confiabilidade. O `StateBasedAnalysis` é uma extensão de MARTE e utiliza sintaxe semelhante ao VSL, sendo possível atribuir valores para analisar propriedades de confiabilidade. Por exemplo, na Tabela 2.1 pode-se observar como é possível configurar as métricas para realizar simulações com o estereótipo `StateBasedAnalysis` do sistema modelado. Na primeira coluna são apresentadas as métricas que podem ser utilizadas no atributo `measure` do estereótipo `StateBasedAnalysis`, onde podem ser realizadas análises de confiabilidade (*Confiabilidade*, disponibilidade (*Availability*) e probabilidade de falha sob demanda (PFD), enquanto que, na segunda coluna, é apresentada a descrição sobre o funcionamento de cada métrica.

Análises podem ser executadas na ferramenta `DEEM Simulator`. O `DEEM Simulator` é uma ferramenta desenvolvida na Universidade de Florença (UNFI) e no Instituto de Ciência e Tecnologia da Informação (ISTI-CNR). Essa ferramenta é executada no sistema operacional Linux. Para os resultados, pode-se ter como exemplo, a confiabilidade esperada de um componente A no tempo ($t = 10.000 = 10^4 \text{hora}$). Porém, este mesmo componente A utilizado no exemplo possui uma taxa de ocorrência de falha de (10^{-12}hora) para uma confiabilidade esperada próxima a 1. Este exemplo foi consultado em [52]. O componente A obteve o resultado de 0.9126. O resultado da confiabilidade do componente A foi abaixo do resultado esperado, pois era esperado um resultado mais próximo de 1. A causa

Tabela 2.1: Métricas que podem ser analisadas com o estereótipo `StateBasedAnalysis`, Tabela adaptada de [52].

Medidas	Descrição
Reliability (instante de tempo= t)	Probabilidade de que o componente não falhe até o tempo t .
Availbility (instante de tempo= t)	Probabilidade de que, em um tempo t , o componente não se encontre em um estado de falha (existindo a possibilidade de ter falhado antes, e também considerando os reparos).
Availbility (fim de intervalo= t)	Fração de tempo que o componente não falha no intervalo entre 0 e t [$0, t$].
PFD (t)	Probabilidade de falha sob demanda, calculada em $(1 - \text{Confiabilidade}(\text{instantOfTime} = t))$.

do resultado exemplificado pode ter ocorrido originalmente nos componentes que compõem o componente A. Por exemplo, dentro do componente A existem dois componentes: o componente B e o componente C. O componente B possui uma taxa de ocorrência de falha maior e propagou esta falha para o componente C, que foi afetado pelo componente B, que interferiu diretamente no resultado geral da análise de todo o componente, o componente A.

2.4 Considerações Finais

Neste capítulo foram apresentadas fundamentações teóricas. Na primeira seção deste capítulo, foram apontadas as definições de confiabilidade, análise de confiabilidade e a distribuição de probabilidade exponencial. Na segunda seção, foram apresentadas a linguagem CHES-ML e as possibilidades de que o usuário possui ao utilizar a metodologia CHES, podendo modelar uma arquitetura de sistema, arquitetura de *hardware* ou arquitetura de *software*. Para cada arquitetura, foram apresentadas definições que devem ser seguidas no uso da metodologia CHES. Com CHES-SBA, foi possível observar conceitos de geração e propagação de falhas, os estereótipos (`SimpleStochasticBehavior`, `ErrorModelBehavior` e `FLABehavior`). Também foi apresentado como configurar

estratégias de manutenção de componentes, como são executadas as análises de confiabilidade e as métricas que podem ser utilizadas. A fundamentação teórica apresentada neste capítulo é importante para que o leitor possa entender noções de análises de confiabilidade, como um sistema é modelado utilizando o CHESS-ML e como as análises de confiabilidade são executadas e interpretadas com o CHESS-SBA.

Capítulo 3

Trabalhos Relacionados

Neste capítulo serão abordados, em detalhes, o planejamento e a execução de uma Revisão Sistemática da Literatura (RSL), com o intuito de coletar trabalhos e adquirir conhecimento sobre modelos formais, sistemas médicos e sistemas de aquisição de sinais biomédicos. Além disso, são apresentados outros trabalhos, que não foram considerados na revisão sistemática, para complementar a revisão.

3.1 Revisão Sistemática da Literatura

3.1.1 Protocolo da Revisão

O objetivo com a RSL executada neste trabalho foi identificar trabalhos relacionados no âmbito da saúde, usados como base para o desenvolvimento desta dissertação do mestrado. Com isso, foram definidas algumas questões, que, ao final da RSL, realizando a leitura e análise dos trabalhos, são respondidas com o intuito de enriquecer a contextualização sobre sistemas médicos. Estas perguntas são:

1. Existem modelagens especificadas (ou *frameworks*) que possuem como objetivo principal do trabalho a análise de propriedades como, por exemplo, confiabilidade?
2. Existem trabalhos utilizando modelos formais, como, por exemplo, redes de Petri coloridas, mesmo que em contextos diferentes, mas que podem ser utilizados como base para essa pesquisa?

3. É possível identificar na literatura modelagens de sistemas de aquisições de sinais biomédicos, ou modelagens de sistemas no âmbito da saúde?

Para responder às perguntas definidas para esta RSL, foram utilizadas quatro bases de dados: *Institute of Electrical and Electronics Engineers (IEEE) Xplore*, *Springer*, *Web of Science* e *Association for Computing Machinery (ACM) Library*. O idioma utilizado nos artigos buscados nesta RSL foi o Inglês. Foram também aplicados os métodos de pesquisa de fonte *Snowballing* e busca manual. Para definir as *strings* de buscas, foi necessário definir algumas palavras chaves. As *strings* de buscas são essenciais para realizar a pesquisa de trabalhos acadêmicos. As palavras chaves definidas foram:

- *Biomedical Signals*;
- *Biomedical Signals Acquisition*;
- *Formal Methods*;
- *Health Systems*;
- *Signals Acquisition*; e
- *Systems of Systems*.

A partir das palavras chaves definidas, pôde-se definir as *strings* de buscas para uso nas bases de dados. As *strings* de buscas foram definidas com o intuito de buscar trabalhos que sobre modelos formais (*Formal Methods*), sistemas médicos (*Biomedical Signal Acquisition* e *Health Systems*) e sistemas de sistemas (*Systems of Systems*). Foram criadas três *strings* de buscas com o operador *AND* e palavras chaves. As *strings* de buscas utilizadas foram:

- *Formal Methods AND Biomedical Signal Acquisition*;
- *Formal Methods AND Health Systems*; e
- *Health Systems and Systems of Systems*.

A partir das *strings* de buscas definidas, realizou-se as buscas nas bases dados obtendo 328 trabalhos ao todo. Dos 328 trabalhos, 16 foram duplicados. Na Tabela 3.1 são apresentadas a quantidade de trabalhos obtidos em cada base e a quantidade de trabalhos duplicados para cada base.

Tabela 3.1: Quantidade de trabalhos coletados nas base de dados.

	ACM	Springer	IEEE	Web of Science	TOTAL
Número de trabalhos coletados	152	70	48	58	328
Duplicados	11	0	0	5	16

Para iniciar a classificação dos trabalhos coletados nesta primeira busca, foi necessário definir os critérios de inclusão (iniciados por (I) na lista a seguir) e exclusão (iniciados por (E) na lista a seguir). Os critérios de inclusão e exclusão na classificação dos artigos foram:

- (I) artigos publicados entre 2014 até o ano de 2019;
- (I) trabalhos sobre modelagem de sistemas na área da saúde utilizando modelos formais;
- (I) trabalho sobre modelagem de sistemas de aquisições de sinais biomédicos.
- (E) não está em Inglês;
- (E) não está no período de tempo desejado;
- (E) inacessível de forma gratuita; e
- (E) não apresenta tema com foco em modelagem de sistemas na área da saúde que use modelos formais.

Os estudos iniciais para seleção dos 328 trabalhos foram baseados na leitura dos resumos de cada trabalho, análises das palavras chaves e leitura dos títulos. Após realizar a primeira análise, 15 trabalhos foram aceitos e 297 foram rejeitados, como pode ser observado na Tabela [3.2](#).

Tabela 3.2: Número de trabalhos classificados realizando análises de títulos e resumos.

	Aceitos	Rejeitados	Duplicados
Número de trabalhos	15	297	16

Com os 15 trabalhos coletados a partir das análises dos títulos e resumos, foram realizadas investigações mais detalhadas. Nesta etapa, as análises foram realizadas com a leitura completa dos 15 trabalhos classificados na etapa anterior, aplicando os critérios de inclusão e exclusão. Ao ler os 15 trabalhos e aplicar os critérios de exclusão e inclusão, chegou-se a conclusão de que 8 trabalhos específicos estavam aptos para responder às perguntas pré-definidas no início desta revisão. Na tabela 3.3 é apresentado o número de trabalhos aceitos, rejeitados e duplicados ao final da revisão. Na Figura 3.1 são apresentados gráficos de setores com a quantidade de trabalhos aceitos e rejeitados e a porcentagem da quantidade de trabalhos coletados em cada base de trabalhos acadêmicos utilizada.

Tabela 3.3: Número de trabalhos aceitos para contribuir com a RSL.

	Aceitos	Rejeitados	Duplicados
Número de trabalhos	8	304	16

Os passos realizadas na RSL pode ser observado na Figura 3.2. Na mesma figura, é apresentado um fluxograma com os primeiros passos referentes às definições da revisão sistemática, realização das buscas e trabalhos adquiridos, aplicação de critérios de inclusão e exclusão e a quantidade de trabalhos aceitos e rejeitados ao final da revisão. No Apêndice A, é possível observar uma tabela com detalhes de todas as publicações que foram aceitas para esta RSL. Na tabela, estão contidas informações de título, autores e base na qual foi encontrado trabalho específico.

3.1.2 Observações Sobre Trabalhos Aceitos

Tahir *et al.* (2015) [70] abordaram a importância do uso de modelos formais na especificação de sistemas. Por possibilitar a realização de provas matemáticas, modelos formais podem auxiliar na especificação de sistemas com mais confiabilidade, desde as fases iniciais do processo de desenvolvimento de sistemas, podendo identificar erros previamente. O foco no trabalho é a aplicação do método de desenvolvimento (*Vienna Development Method - Specification Language (VDM-SL)*). Os autores utilizaram modelos formais para especificar um sistema *e-Health* para a área da saúde (o mesmo sistema foi especificado em outro trabalho com a notação Z [8]). O objetivo com o sistema especificado é auxiliar pacientes que

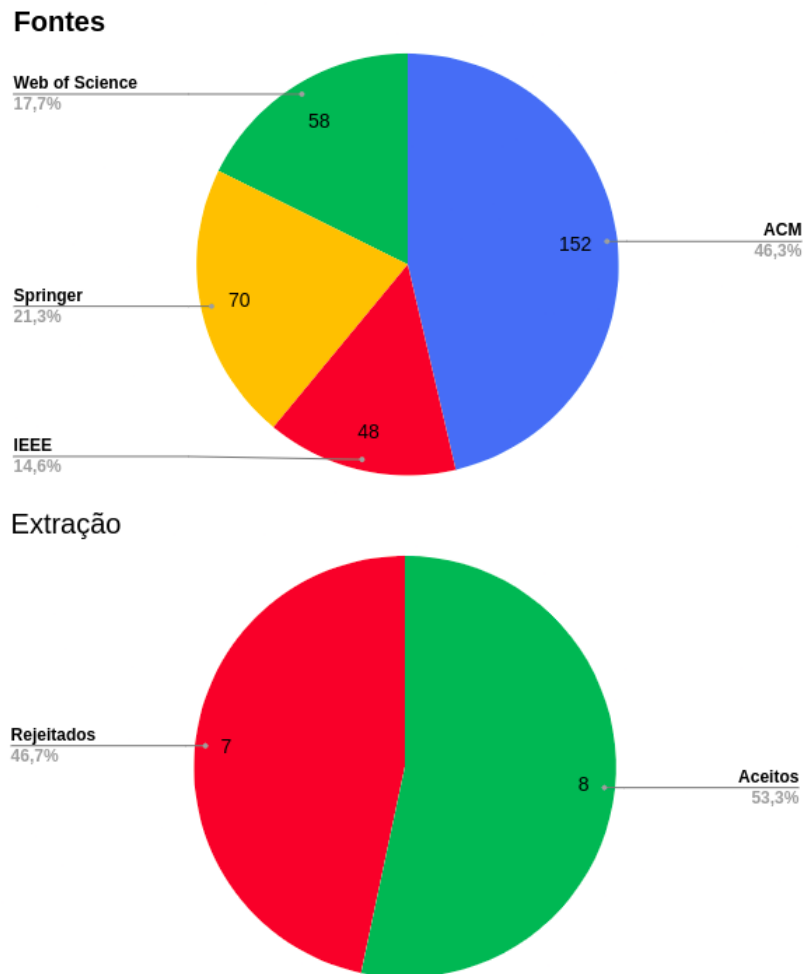


Figura 3.1: Gráfico de classificação dos artigos selecionados.

fornecem informações para médicos usando a Internet. O sistema utilizado prover funcionalidades, como, por exemplo, a realização de *checkup online* e manipulação de prescrições médicas. Com os requisitos elicitados para o sistema, foram definidas especificações formais utilizando VDM-SL para auxiliar desenvolvedores na implementação do sistema. A especificação formal utilizada neste trabalho é relevante para a geração de código-fonte de maneira rápida, segura e clara. Além disso, existem vários aspectos em um sistema de saúde que precisam ser modelados previamente para evitar erros, como, por exemplo, segurança. Os autores destacaram que existem vários aspectos no sistema que não foram abordados no trabalho, porém, não detalham quais outros aspectos são necessários.

Mashkooor e Egyed (2018) [49] apresentaram um estudo de caso para modelar um sistema de hemodiálise. Os autores abordaram as dificuldades e criticidades no desenvolvimento de

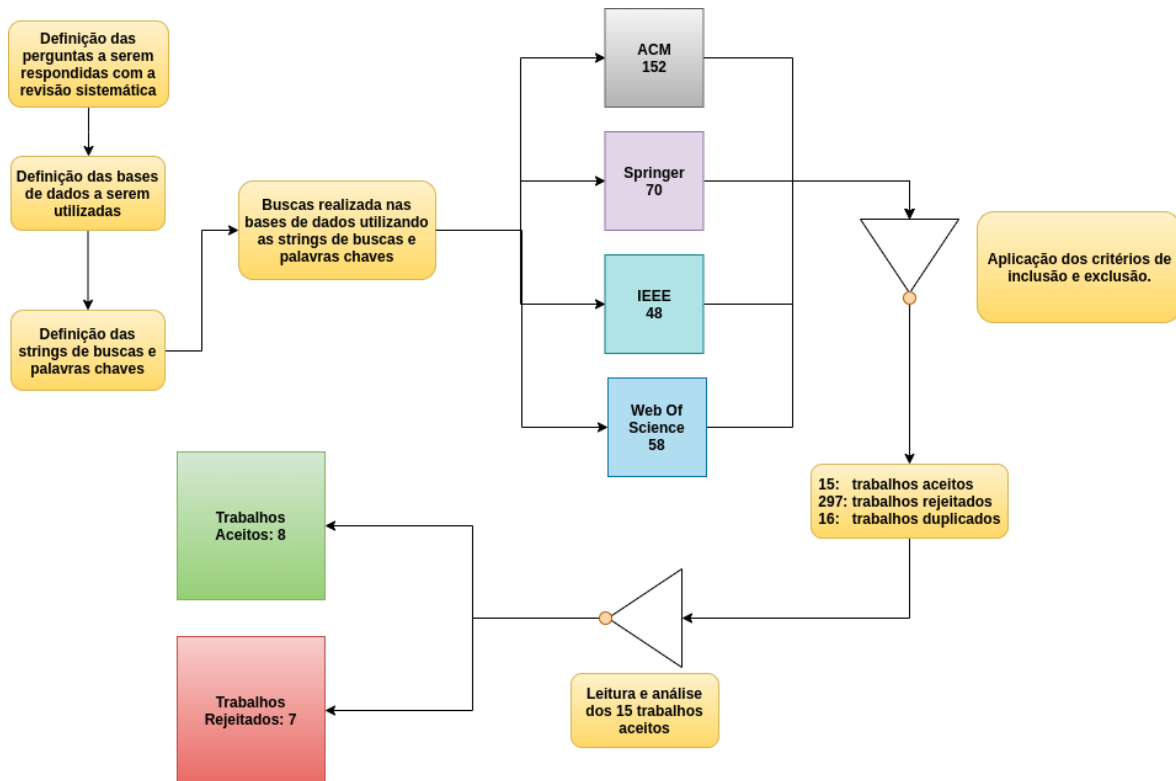


Figura 3.2: Fluxograma utilizado para representar o processo de execução da RSL.

um sistema na área de saúde. Neste contexto, foram definidos passos necessários e recomendações para o processo de desenvolvimento, como, por exemplo, identificar possíveis situações de risco, requisitos de segurança e verificar se os requisitos elicitados são realmente satisfeitos para o ambiente de uso. Os autores utilizaram modelos formais tais como *Event-B* e a notação *Z*. Foram abordados alguns problemas relacionados com o contexto de sistemas de hemodiálise, como, por exemplo, o tratamento de dados que não suportaram abstrações simples. Para a especificação do sistema, foram considerados detalhes de *hardware* e *software*. Foi concluído que este tipo de sistema é bastante complexo e crítico, necessitando, conseqüentemente, de um grande refinamento de projeto. Foram consideradas também algumas limitações de ferramentas para a modelagem, como, por exemplo, a geração de código automático, prova mecânica de propriedades, demonstração automática de teoremas e explosão de espaços de estados.

Azeem *et al.* (2014) [8] aplicam a linguagem de especificação *Z* a um sistema *e-Health*. A especificação do *e-Health* foi desenvolvida com o *Z Schema*, o qual provê uma completa especificação do que será usado no desenvolvimento, verificação e manutenção dos proces-

dos do sistema. Os autores abordam que a especificação formal Z, neste projeto, teve uma grande utilidade para detectar muitos erros. Com essa descoberta prévia de erros pode-se concluir que o custo de correção durante a etapa de projeto foi menor do que seria se fosse corrigido futuramente. Outro ponto que a utilização dessa especificação formal trouxe de vantagem para este trabalho, foi que influenciou significativamente na quantidade de dias para o desenvolvimento do sistema, assim os autores realizam comparações de metodologias comuns de desenvolvimento, com metodologias de desenvolvimento que utilizam a linguagem de especificação Z.

Qaddoumi *et al.* (2018) [59] focaram na análise de propriedades de governança de qualidade em sistema de sistemas (*Systems-of-Systems* (SoS)) aplicados para o cuidado de pacientes com câncer. Foi abordado o desafio de aplicar governança de qualidade de maneira adequada para SoS, alcançando a cooperação necessária entre múltiplos sistemas heterogêneos. Neste contexto, os autores apresentaram um arcabouço chamado *OntoSoS.QM.Gov.* Durante a definição do arcabouço, os autores enfrentaram desafios, como, por exemplo, a representação da semântica e a generalização do arcabouço para SoS com governança de qualidade. Entretanto, algumas limitações podem ser identificadas, como, por exemplo, a avaliação de qualidade de componentes do sistema que têm como função determinar se os pacientes são bebês, crianças ou adolescentes. Além disso, são destacadas limitações referentes a robustez e segurança de SoS construídos.

Gehlot *et al.* (2016) [29] motivaram o desenvolvimento de uma ferramenta para compartilhar dados médicos. Foi argumentado que a utilização de modelos formais é relevante para reduzir o número de defeitos e custos, além de melhorar a eficácia e a entrega das informações. A modelagem da ferramenta foi realizada utilizando redes de Petri coloridas (*Coloured Petri Nets* (CPN)). O principal modelo especificado está relacionado com o processo realizado por profissionais de saúde, como, por exemplo, enfermeiros para a coleta de sangue em uma unidade de saúde. Para a atividade de validação, foi analisado o impacto da alocação de recursos em relação à espera do paciente ao realizar a coleta de sangue. Os autores implementaram monitores com o CPN para observar, inspecionar, controlar ou modificar algumas simulações do modelo especificado (em caso de ocorrência de comportamentos inesperados). São observadas métricas, como, por exemplo, o tempo de permanência de pacientes em Unidades de Terapia Intensiva (UTIs). Com tal tipo de informação, os responsáveis pela

organização do ambiente podem tomar decisões sobre quando os recursos (humano ou material) devem ser movidos de uma localidade para outra, reduzindo custos sem sacrificar a qualidade do atendimento médico. Gehlot *et al.* (2016) [29] demonstraram o uso de CPN na modelagem e simulação de processos de saúde, porém, somente com foco no processo de coleta de sangue de pacientes em UTIs. Isto pode ser considerado como uma limitação, dado que outros processos que ocorrem paralelamente podem impactar no funcionamento ideal de todo o ambiente.

Fradinho *et al.* (2014) [23] consideraram a modelagem de um sistema de gerenciamento de hospitais desenvolvido para um hospital de Boston e um hospital de Londres. No entanto, cada um dos hospitais, possui particularidades que devem ser consideradas durante o levantamento de requisitos, como, por exemplo, desenvolvimento da lógica de negócio utilizada no hospital e documentação. A modelagem da solução não se restringe somente ao compartilhamento de informações médicas dos pacientes entre os profissionais, onde também são consideradas informações referentes ao gerenciamento do hospital (por exemplo, custos). Foi identificado um ganho de desempenho para a administração dos hospitais utilizando uma arquitetura mais robusta e a utilização de um SOS. Porém, é necessário ainda validar e mensurar o ganho obtido com a mudança na arquitetura do *software* administrativo.

Nos estudos apresentados por Sobrinho *et al.* (2019) [66] e Sobrinho *et al.* (2014) [67] são abordadas problemáticas relacionadas com a regulamentações necessárias de sistemas médicos críticos. Os autores modelaram sistemas de aquisições de sinais biomédicos, especificamente ECG utilizando redes de Petri coloridas com a ferramenta *CPN-Tools* para contemplar os requisitos de segurança e eficácia com foco em certificações realizadas por agências regulamentadoras, como, por exemplo, a FDA e a Agência Nacional de Vigilância Sanitária (ANVISA). Sobrinho *et al.* (2014) [67] realiza a especificação dos requisitos gerais do ECG, desde a verificação do nível da bateria até aplicação de filtros e conversão de sinais e através da modelagem desenvolvida com *CPN-Tools* pôde ser validado que modelagens como a deste trabalho podem ser úteis em processos de certificação de sistemas médicos críticos. Utilizando a modelagem desenvolvida para um ECG com *CPN-Tools*, ao qual foi validado a utilização em processos de certificações de sistemas médicos críticos, Sobrinho *et al.* (2019) [66] realizaram um estudo de caso composto por mais detalhes das verificações de propriedades de segurança e eficácia. O trabalho também utilizou o *ASK-CTL - model chec-*

king para verificar os modelos e os estados gerados e assim poder validar se as propriedades eram satisfeitas de forma correta.

3.1.3 Análise Sobre a Revisão Sistemática

O objetivo com esta RSL foi a realização de buscas de trabalhos acadêmicos na literatura, filtrar trabalhos específicos de acordo com o contexto definido nesta RSL e então responder as perguntas definidas previamente no protocolo da RSL. Após a leitura dos trabalhos identificados, foi possível responder às perguntas de pesquisa definidas no protocolo.

Para a primeira pergunta de pesquisa, foram identificados trabalhos relacionados com a especificação de modelos e o foco em verificações de propriedades de segurança, confiabilidade, eficácia e integridade dos dados. Por exemplo, nos trabalhos apresentados por Tahir *et al.* (2015) [70], Mashkoor e Egyed (2018) [49] e Qaddoumi *et al.* (2018) [59] são apresentados modelos com foco em verificação de propriedades desejadas em sistemas em geral.

Para a segunda pergunta de pesquisa, existem trabalhos acadêmicos relacionados com modelos formais, como, por exemplo, trabalhos que abordam modelagens de SoS, e/ou utilizam notação Z, Event-B, B, ou CPN [70] [49] [8] [29] [67] [66]. Neste contexto, alguns dos trabalhos identificados foram úteis para auxiliar na especificação de sistemas de monitoramento multiparâmetro, apresentada nesta dissertação. Foram identificados dois trabalhos relacionados com a modelagem de sistemas de aquisições de sinais biomédicos (um sistema de ECG) [67] e [66]. Com estes trabalhos, foi possível agregar informações para a elicitação de requisitos de um sistema de monitoramento multiparâmetro.

Para a terceira pergunta de pesquisa, de acordo com a RSL, a partir dos critérios de inclusão e observações realizadas, foi possível observar que todos os artigos classificados como aceitos, foram relacionados com sistemas no âmbito da saúde (poucos com foco em sistemas de aquisição de sinais biomédicos). Além disso, nota-se que não existem trabalhos que abordam a modelagem e análise de confiabilidade de sistemas para a área da saúde que utilizam a metodologia CHESS e o *plugin* CHESS-SBA ao qual este último utiliza o formalismo de redes de Petri estocástica. Isto evidencia uma lacuna de pesquisa, que é abordada nesta dissertação.

3.2 Outros Trabalhos Relacionados

O objetivo, com este capítulo, é abordar outros trabalhos relacionados disponíveis na literatura, complementando a discussão apresentada na RSL. Os trabalhos descritos neste capítulo foram identificados por buscas manuais utilizando o Google Acadêmico. Mais especificamente, são abordadas obras que utilizam a metodologia CHESSE, trabalhos relacionados com sistemas médicos de forma em geral e trabalhos que tratam sobre a análise de confiabilidade de forma em geral.

3.2.1 Aplicação da Metodologia CHESSE

Bressan *et al.* (2018) [13] apresentaram um processo sistemático para projetistas e analistas utilizarem a metodologia CHESSE para realizar análises de confiabilidade. Foram aplicados os processos de modelagens e análises da metodologia CHESSE a um sistema híbrido automotivo do mundo real. Os autores abordaram análises de confiabilidade aplicadas em conformidade com normas de segurança para o âmbito do sistema modelado. As análises de confiabilidade foram baseadas em modelos de sistemas críticos em conformidade com a ISO 26262 [56]. É detalhada a modelagem de um sistema híbrido de freio automotivo, especificando a arquitetura do sistema, subsistemas, componentes, portas de entradas e saídas e os conectores necessários entre os componentes. A modelagem desenvolvida é composta por 10 componentes, distribuídos em unidades de freio, pedal mecânico, barramentos de comunicação e baterias. Os autores detalharam como foi realizado o processo de enriquecimento de informações sobre os atributos de confiabilidade e disponibilidade utilizando os estereótipos `SimpleStochasticBehavior`, `ErrorModelBehavior` e `FLABehavior`. Foi definida uma medida como confiabilidade equivalente a 1 ano (instante de tempo = 8760 horas), significando que a probabilidade de que nenhuma das unidades de freio falhe em um ano.

Montecchi e Gallina [53] abordaram sistemas sociotécnicos, que são caracterizados pela utilização de forma heterogênea de diversas entidades, como, por exemplo, a aplicação de sistemas no domínio petrolífero. O objetivo é garantir o funcionamento correto e seguro de sistemas. É proposto um metamodelo composto por ferramentas para modelar entidades

sócio-técnicas e propriedades relacionadas à segurança. Neste contexto, o CHES-SBA é utilizado para a modelagem de comportamentos de falhas e reparos dos componentes. Os componentes do metamodelo proposto são representados através do conceito de *State* (estado) que permite definir qualquer tipo de estado, seja ele um estado saudável ou um estado de falha. As direções futuras descritas no trabalho estão relacionadas com a coleta de evidências de segurança usando análises executadas.

Assim, os trabalhos de modelagens e análise de confiabilidade de sistemas utilizando o CHES-SBA proveem exemplos de como modelar sistemas, planejar e executar análises de confiabilidade no sistema desenvolvido.

3.2.2 Sistemas Médicos

No trabalho apresentado por Al-Hamadi [5], o funcionamento de um ECG é apresentado para detectar os sinais elétricos emitidos por um corpo humano. Ao identificar padrões em sinais, o trabalho discutiu a complexidade no diagnóstico com ECG. Neste contexto, foi especificado um modelo abstrato de um ECG ao qual Event-B é utilizado para formalizar e validar as especificações de sinais de ECG, visando diminuir as inconsistências e os problemas com as interpretações dos sinais coletados. Para diminuir as inconsistências dos sinais, foram definidas propriedades relacionadas com a qualidade do sinal gerado pelo ECG e seus comportamentos. Também sobre sinais de ECG, é descrita por Gonçalves [30] a problemática do armazenamento e transmissão de dados coletados a partir de um ECG, o que motivou a padronização desses dados. O foco que esses padrões possuem se resume a como as informações e os dados devem ser visualizados por usuários através de monitores. Padrões de dados de um ECG utilizados na Europa e nos EUA adotado pela FDA e também padrões para realização de pesquisa de fisiologia cardíaca são citados pelo trabalho. Os autores testaram uma hipótese de que uma ontologia de referência sobre um ECG pode ser empregada para alcançar a integração semântica entre os padrões de dados de ECG. Foram especificados modelos utilizando UML para determinados contextos relacionados com ECG, incluindo a aquisição e gravação dos dados cardíacos coletados do paciente.

Wei *et al.* (2009) [74] apresentam o projeto de um monitor multiparâmetro utilizado para o monitoramento de sinais de pacientes em anestesia profunda. O trabalho aborda em detalhes quais sinais vitais um sistema de monitoramento multiparâmetro coleta de um paciente,

como as informações são apresentadas para o usuário através do monitor, quais filtros são aplicados nos sinais coletados e também quais tarefas o *software* é responsável por executar em um sistema de monitoramento multiparâmetro. Como este sistema de monitoramento multiparâmetro foi projetado para monitorar pacientes em anestesia profunda, requisitos foram definidos e funcionalidade implementadas a nível de *software* para que o sistema considere uma variedade de parâmetros fisiológicos capturados pelo sistema para analisar o nível de anestesia que o paciente está. Com a utilização de vários parâmetros para verificar o nível de anestesia do paciente, isto aumenta a credibilidade da informação fornecida pelo sistema de monitoramento multiparâmetro.

No trabalho exposto por Paech e Wetter [55] foram abordados os desafios de se obter qualidade de *software* em sistemas médicos, como, por exemplo, a complexidade e variabilidade de sistemas médicos. São apresentadas algumas recomendações para melhorar a qualidade de sistemas médicos utilizando o método da engenharia de *software* MOQARE. Neste caso, o MOQARE foi importante para elicitar e priorizar requisitos de qualidade, o que contribui na especificação de confiabilidade do sistema. Como estudo de caso, foi utilizado um sistema de aconselhamento de medicamentos. Foram elicitados os requisitos do sistema, priorizando os atributos de qualidade em diversos cenários que pode acontecer em um hospital, incluindo cenários de epidemia. A elicitação de requisitos foi realizada considerando medidas para controle para defeitos. Entretanto, a realização de análise utilizando a metodologia do MOQARE em sistemas com alta complexidade como o trabalho de Paech e Wetter [55] descreve pode consumir muito tempo, dificultando a tomada de decisão em relação a se o grande tempo destinado para isto vale a pena. Devido à alta complexidade, o trabalho apresenta limitações de tempo para a utilização da metodologia MOQARE com sistemas médicos.

As especificações dos modelos utilizando UML e projetos de sistemas de monitoramento multiparâmetro foram úteis para a contextualização do funcionamento entre os CS para o sistema de monitoramento multiparâmetro deste trabalho. Como também, ações tomadas em alguns trabalhos apresentados nesta seção, contribuíram de forma indireta para o aumento da confiabilidade das informações geradas por alguns sistemas.

3.2.3 Análise de Confiabilidade

Gao e Wang [27] realizaram análises de confiabilidade em um sistema *k-out-of-n* que representa um sistema com n componentes, onde esse sistema funciona se ao menos k dos n componentes do sistema funcionarem. Foram calculados a função de confiabilidade e o tempo médio de falha usando o método de transformações de Laplace, discutindo a otimização da relação custo-efetividade para fornecer recomendações gerenciais. A manutenção preventiva orientou o reparo do sistema.

Kumar e Jain [41] apresentaram a análise de confiabilidade de um sistema de usinagem multicomponente (uma planta de energia solar e sistema de manuseio de materiais). Os autores aplicaram a abordagem do método matricial para analisar o comportamento transiente, além de analisar a sensibilidade da confiabilidade do sistema e o tempo médio de falha. Zhang [76] estudou sistemas *k-out-of-n* ponderados aleatoriamente, realizando análises de confiabilidade. A análise de confiabilidade foi realizada usando ensaios de Bernoulli com diferentes probabilidades de seleção. Singh e Singh [65] analisaram o impacto de falhas de componentes individuais na confiabilidade geral de sistemas usando uma abordagem Bayesiana. Os autores propuseram uma técnica para determinar a criticidade de componentes e realizaram um estudo de caso em um sistema de usina nuclear. Além disso, Tsarouhas [72] realizou análises de confiabilidade e capacidade de manutenção de um sistema de diálise hospitalar. O autor apresentou estatísticas descritivas das falhas dos componentes do sistema, calculou a confiabilidade e a capacidade de manutenção usando intervalos de tempo diferentes e discutiu os resultados para fornecer recomendações com o objetivo de auxiliar na manutenção.

De maneira similar aos trabalhos relacionados apresentados nesta seção, nesta dissertação, a confiabilidade foi calculada para auxiliar na definição de estratégias gerenciais, considerando uso e manutenção preventiva de sistemas de monitoramento multiparâmetro utilizados em UTIs. Também foi analisado o impacto de falhas de componentes individuais na confiabilidade geral de sistemas. O estudo da confiabilidade de sistemas de monitoramento multiparâmetro para UTI é um tópico ainda não abordado no estado da arte atual. Esse tipo de estudo é relevante para subsidiar a gestão das UTIs, evitando a falta de recursos em situações de emergência, como é o caso da pandemia da COVID-19.

3.3 Considerações Finais

Este capítulo apresentou trabalhos relacionados que foram importantes para a evolução desta pesquisa. Primeiro, uma RSL abordando trabalhos que utilizam modelos formais, sistemas médicos e sistemas de aquisição de sinais biomédicos. Esta RSL foi útil para conhecer tipos de sistemas utilizados no âmbito da saúde e também quais os formalismos utilizados para realizar verificações de propriedades em sistemas médicos.

Para complementar os trabalhos coletados na RSL, foram realizadas as buscas de alguns trabalhos relacionados que tratam sobre a modelagem e análises de confiabilidade utilizando o arcabouço CHESS, trabalhos sobre sistemas médicos de forma em geral e por último, trabalhos que tratam de análise de confiabilidade de sistemas de forma em geral. Este capítulo de trabalhos relacionados agregou no conhecimento da ferramenta CHESS, em requisitos e funcionamento de sistemas médicos em geral, CS para sistemas de monitoramento multiparâmetro e análises de confiabilidade de sistemas.

Capítulo 4

Especificação do Sistema

Neste capítulo, é apresentado do que são compostos sistemas de monitoramento multiparâmetro e detalhes de funcionamento dos CS. Assim, dois exemplos de monitores multiparâmetros utilizados pelo mercado atualmente foram considerados como base para a utilização deste trabalho, juntamente com as informações adquiridas através das entrevistas realizadas com o profissional de eletrônica de equipamentos médicos.

Este capítulo também apresenta como foi o processo de coleta de requisitos para o sistema de monitoramento multiparâmetro, e então como foi definida e realizada a modelagem do sistema utilizando o CHESS-ML.

4.1 Descoberta e Elicitação de Requisitos

Os sistemas de aquisição de sinais biomédicos possuem a função de coletar sinais biomédicos de um paciente que está em observação, sendo possível o profissional de saúde realizar avaliações do estado atual do paciente [14]. Esses sistemas estão cada vez mais inseridos em ambientes como clínicas, hospitais e UTIs, sendo utilizados para ajudar no monitoramento de paciente, e para ajudar na tomada de decisão do profissional responsável, como, por exemplo, médicos e enfermeiros. Podem ser citados como exemplos de sistemas de aquisição de sinais biomédicos: Eletrocardiograma (ECG) responsável por avaliar os batimentos cardíacos do paciente em um determinado intervalo de tempo; verificadores de pressão arterial invasiva e não invasiva; termômetros; verificadores de respiração; e oxímetro de pulso [61].

Os sistemas de monitoramento multiparâmetro geralmente medem eletrocardiograma (ECG), termômetro, oxímetro de pulso, frequência respiratória e pressão arterial. Além disso, eles são compostos pela fonte de alimentação principal (ou seja, energia elétrica), bateria e uma tela de monitor LCD (*Liquid-crystal display*) para visualização das informações coletadas do paciente. Para um sistema de monitoramento multiparâmetro poder realizar a coleta de todas essas informações citadas, o sistema deve ser composto por sistemas constituintes (CS), onde cada sistema possui lógica e tratamento de dados distintos.

Portanto, para realizar a elicitação de requisitos de um sistema de monitoramento multiparâmetro utilizado em UTIs para um paciente, foram conduzidas entrevistas com um profissional com mais de quinze anos de experiência em manutenção de sistemas médicos. O profissional contribuiu fornecendo informações sobre funcionamento interno, fluxos de comunicação interno dos CS, fluxos de comunicação entre os CS e valores de probabilidade de falhas dos componentes do sistema. Além disso, foram analisados sistemas existentes de monitoramento multiparâmetro (por exemplo, [2] e [3]) e revisões da literatura.

Um sistema de monitoramento multiparâmetro possui componentes de *hardware* e *software*. Os componentes de *hardware* podem ser exemplificados em: eletrodos, sensores, microprocessadores, amplificadores, filtros e conversores; já os componentes de *software* podem ser citados em: verificação de impedância, processamento de dados coletados, configuração de alarmes e verificação de nível da bateria. Assim, esses componentes fazem parte de CS de um sistema de monitoramento multiparâmetro, como: (1) ECG; (2) oximetria de pulso; (3) termômetro; (4) verificador de pressão invasiva e não invasiva; (5) e verificador de respiração.

O Eletrocardiograma (ECG) como um sistema constituinte de um monitor multiparâmetro é utilizado para verificar as atividades elétricas do coração, auxiliando no diagnóstico de patologias, tais como arritmias cardíacas, constatação de áreas do coração afetadas por infartos e doenças congênitas [17]. A Oximetria de pulso é um método não invasivo que permite estimar a saturação de oxigênio da hemoglobina arterial ou níveis de oxigênio no sangue. Com o oxímetro de pulso é possível saber a qualidade do oxigênio que está sendo enviado para partes mais afastadas do coração, como, por exemplo, braços e pernas. A porcentagem de saturação de oxigênio ligada à hemoglobina no sangue é chamada de SaO₂, e quando é medida por um oxímetro de pulso, esse valor é chamado de SpO₂ [51].

Através de sensores de temperatura alocados no corpo humano, é possível checar a temperatura do corpo humano e poder verificar se o valor da temperatura está de acordo com os padrões para o ser humano. Utilizando medidores de pressão arterial aplicados ao paciente, é possível verificar informações referentes à medição de pressão invasiva e não invasiva do paciente. Com a utilização de eletrodos no corpo do paciente, medições referentes a respiração do paciente podem ser coletadas [62].

Para a utilização e operação do monitor multiparâmetro são necessárias algumas configurações iniciais, onde o responsável pela operação do monitor multiparâmetro deve posicionar os eletrodos, leds e sensores corretamente no corpo do paciente, e então o *software* realizar verificações de impedância eletrodo-pele e impedância torácica. As verificações de impedância eletrodo pele e impedância torácica, são importantes pois são realizadas verificações se os sinais coletados a partir dos eletrodos estão de acordo com o que o sistema espera para continuar a coleta das informações do paciente. A segunda verificação está relacionada à alimentação do monitor multiparâmetro. O monitor multiparâmetro deve estar sendo alimentado via energia do hospital através de uma tomada por exemplo, ou através da energia de uma bateria. Os níveis de bateria aceitáveis podem variar de acordo com o tamanho da bateria e modelo do monitor multiparâmetro utilizado. Por exemplo, no monitor multiparâmetro [2], a fonte de alimentação suporta uma voltagem entre 100V até 250V e uma bateria com capacidade de 23000mAh.

Com as verificações de impedância e alimentação, o sistema de monitoramento multiparâmetro está apto a iniciar a captura de sinais. Basicamente, todos os CS inseridos dentro de um sistema de monitoramento multiparâmetros possuem uma mesma lógica de fluxo após realizar a captura de sinais, que é: (1) enviar o sinal em formato analógico para o amplificador de instrumentação realizar a amplificação do sinal; (2) amplificador após amplificar o sinal, envia o sinal para os filtros removerem ruídos e informações desnecessárias do sinal; (3) após a remoção de ruídos do sinal realizada pelos filtros, o sinal é enviado para o conversor que realiza a conversão de sinal analógico para sinal digital. Os CS de ECG, verificador de pressão arterial, termômetro, oxímetro de pulso e verificador de respiração possuem este fluxo de funcionamento envolvendo o sinal capturado do paciente até poder ser visualizado através do monitor LCD. O fluxo de funcionamento dos CS descrito neste parágrafo, pode ser observado na Figura 4.1.

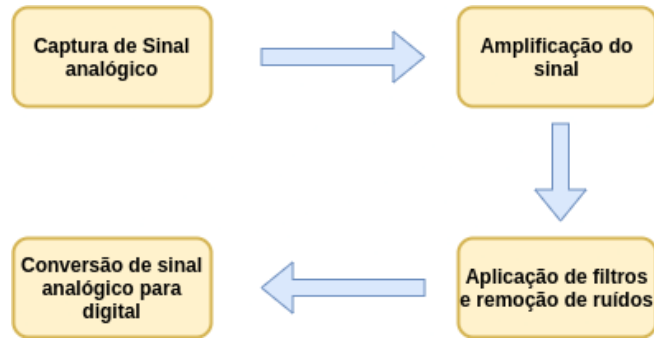


Figura 4.1: Fluxo similar de captura de sinal entre todos os sistemas de um sistema de monitoramento multiparâmetro.

O amplificador de instrumentação tem a função de receber um sinal de entrada, e retornar esse sinal mais forte, representando um ganho de sinal. O processo de ganho de sinal deve ser realizado por uma função pré-definida responsável por definir o valor do ganho a se multiplicar, essa função pode ser diferente em cada CS.

Sistemas que trabalham com sinais, como, por exemplo, o sistema de monitoramento multiparâmetro, sofrem com a interferência de sinais indesejados e ruídos. Assim, são definidos alguns filtros que podem ser aplicados ao sinal capturado. Por exemplo, para o sistema constituinte de ECG são aplicados filtros de passa baixa, passa alta, passa banda e rejeita banda. O filtro de passa baixa é responsável por eliminar todo o sinal acima do valor permitido, o filtro de passa alta é responsável por eliminar todo sinal abaixo do valor permitido. O filtro de passa banda, é responsável por permitir o sinal válido, que opera entre o valor mínimo permitido e o valor máximo permitido, ou seja, é o sinal que opera no intervalo correto e é utilizado pelo ECG.

Após serem aplicados filtros para remoção de ruídos em cada CS, cada CS deve enviar o sinal filtrado para realizar a conversão de sinal analógico para sinal digital. Cada CS após realizar a conversão de sinal analógico para digital, deve enviar o sinal para um microprocessador do monitor multiparâmetro. O microprocessador tem a função de processar todas as informações que os CS proveram, e então poder realizar verificações de alarmes de acordo com os padrões que foram configurados pelo profissional de saúde. Após o microprocessador realizar a verificação dos sinais adquiridos pelos CS, deve enviar as informações dos sinais coletados do paciente em formato de histórico para a exibição através de um monitor LCD.

Os sistemas de monitoramento multiparâmetros [2] [3] são exemplos de sistemas nos quais é possível realizar a verificação de sinais vitais, como, por exemplo, pressão arterial, frequência de respiração, saturação de oxigênio, temperatura e atividades rítmicas do coração. Esses exemplos de sistemas de monitoramento multiparâmetro também possuem fonte de alimentação oriunda através de energia elétrica ou bateria e um monitor LCD onde é possível visualizar todas as informações que o sistema de monitoramento multiparâmetro coletou.

4.2 Arquitetura e Organização dos Modelos

A modelagem foca em representar o funcionamento de um monitor multiparâmetro alocado em UTIs de uso exclusivo para um paciente. Com isso, a modelagem do monitor multiparâmetro contém os componentes necessários para a realização da coleta de informações de sinais vitais do paciente e monitoramento. A modelagem foi desenvolvida com a linguagem CHES-ML e a visão de sistema (*system view*) do CHES. As motivações que levaram para o modelo ter sido desenvolvido com a arquitetura de sistema foram que, o modelo é composto por elementos de *software* e *hardware*, e que através dessa arquitetura pode-se abstrair vários elementos, como, por exemplo, valores de aplicação de filtros e funções de amplificação de sinal.

Para a modelagem do monitor multiparâmetro, foram definidos alguns padrões para facilitar a legibilidade dos diagramas. Os `blocks` na cor azul representam blocos simples de mais baixo nível. Os `blocks` na cor verde são os blocos que possuem um nível hierárquico maior do que os blocos na cor azul, através dos blocos da cor verde é possível definir diagramas de bloco interno, representar cada CS que compõe o sistema de monitoramento multiparâmetro e a comunicação interna do modelo. `Blocks` definidos de cor vermelha, possuem o mais alto nível hierárquico de todo o modelo. Na Figura 4.2 é apresentada a hierarquia dos blocos que são utilizados neste modelo desenvolvido para um monitor multiparâmetro. Na cor amarela, são definidos alguns `datatypes`, que são utilizados para definir o tipo de informação utilizado na comunicação entre os blocos.

Para o modelo de um sistema de monitoramento multiparâmetro com o CHES, foram definidos inicialmente alguns diagramas de definição de blocos, diagramas que são respon-

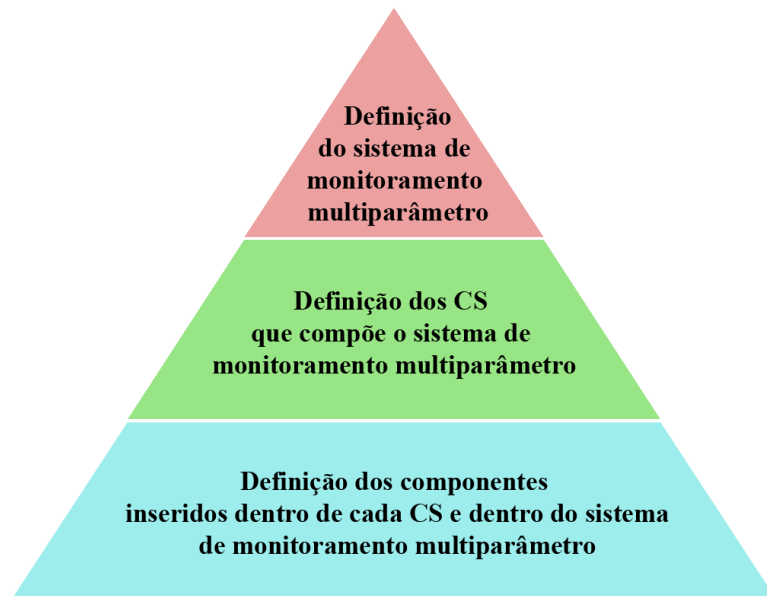


Figura 4.2: Hierarquia de blocos definidas para o sistema de monitoramento multiparâmetro.

sáveis por representar CS do sistema de monitoramento multiparâmetro como, por exemplo, ECG e respiração, termômetro, oxímetro de pulso e verificador de pressão arterial. Nos diagramas que representam os CS do sistema de monitoramento multiparâmetro, é possível observar blocos na cor azul e verde. Também foi necessário definir um diagrama de blocos que representasse todo o sistema de monitoramento multiparâmetro e a comunicação dos CS com outros componentes do sistema. No diagrama de definição de blocos que representa a definição do sistema de monitoramento multiparâmetro de maior nível hierárquico, é possível observar blocos na cor azul e um bloco na cor vermelha. O bloco de cor azul no diagrama de definição de blocos do monitor multiparâmetro, representa blocos definidos e utilizados somente no diagrama de mais alto nível referente ao monitor multiparâmetro.

A próxima seção irá apresentar detalhes do sistema de monitoramento multiparâmetro utilizando o CHESS-ML, sendo categorizados através dos diagramas de definição de blocos que representam os CS de ECG e verificador de respiração, oxímetro de pulso, termômetro, verificador de pressão arterial e o gerenciamento de todos esses CS através do diagrama de

definição de blocos principal para o monitor multiparâmetro.

4.3 Modelagem de um Sistema de Monitoramento Multiparâmetro

A modelagem de sistemas de monitoramento multiparâmetro foi definida com diagramas de definição de blocos e diagramas de bloco interno. Posteriormente, foi realizada a conexão desses modelos representando um modelo só, o modelo para sistemas de monitoramento multiparâmetro. A modelagem para o sistema de monitoramento multiparâmetro está dividida em alguns diagramas de definição de blocos, que são: (1) monitor multiparamêtro; (2) ECG e respiração; (3) termômetro; (4) oxímetro de pulso; e (5) verificador de pressão arterial.

As subseções a seguir apresentam os modelos contendo os detalhes de definição de blocos, tipos de dados e comunicação de blocos através dos diagramas blocos internos.

4.3.1 Monitor Multiparâmetro

O modelo para sistemas de monitoramento multiparâmetro é o principal modelo para o funcionamento de todos os CS e do monitor multiparâmetro. É através do modelo do monitor multiparâmetro que os CS apresentados nas subseções a seguir são conectados. Porém, é necessário definir alguns blocos específicos do monitor multiparâmetro no diagrama de definição de blocos. Foram definidos três blocos para satisfazer os requisitos de alimentação do sistema de monitoramento multiparâmetro, que foram: `PowerSupply`, `Battery` e `PowerSupplyController`. O bloco `PowerSupply` representa a fonte de alimentação principal do sistema, onde recebe a energia diretamente de uma tomada por exemplo. O bloco `Battery` representa a bateria acoplada ao monitor multiparamêtro. E o bloco `PowerSupplyController`, é o bloco que representa o controlador de alimentação do monitor multiparamêtro, este bloco recebe a energia fornecida através da `PowerSupply` e da `Battery`, e distribui para todo o monitor multiparamêtro. Através do `PowerSupplyController` pode-se adicionar técnicas de redundância para a fonte de alimentação principal do sistema. É importante res-

saltar que o `PowerSupplyController` atua gerenciando a energia para que o monitor multiparamêtro possua o maior tempo disponível possível. Com a definição do `PowerSupplyController`, para que o monitor multiparâmetro esteja indisponível no quesito de energia, é preciso que a fonte não esteja sendo alimentada (por falta de energia e que o gerador da UTI também não esteja disponível) e que a bateria esteja totalmente descarregada ou sem condições de uso, este comportamento acontece somente utilizando o estereótipo `ErrorModelBehavior`.

Além dos blocos referentes à energia citados no parágrafo anterior, foram definidos mais três blocos no diagrama de definição de blocos do monitor multiparâmetro, que são: `Central Process Unit - (CPU)`, `MonitorLCD` e `MultiparameterMonitor`. O bloco `CPU`, é responsável por receber e processar todos os dados do monitor paramétrico, como dados referentes à energia da bateria, frequência cardíaca e respiratória, oximetria de pulso, sinal de temperatura do paciente e pressão arterial invasiva e não invasiva. Também foram definidos os `datatypes` `StreamResults` e `VoltageDC` para o monitor multiparâmetro. Na Figura 4.3 é apresentado o diagrama de definição de bloco do sistema de monitoramento multiparâmetro.

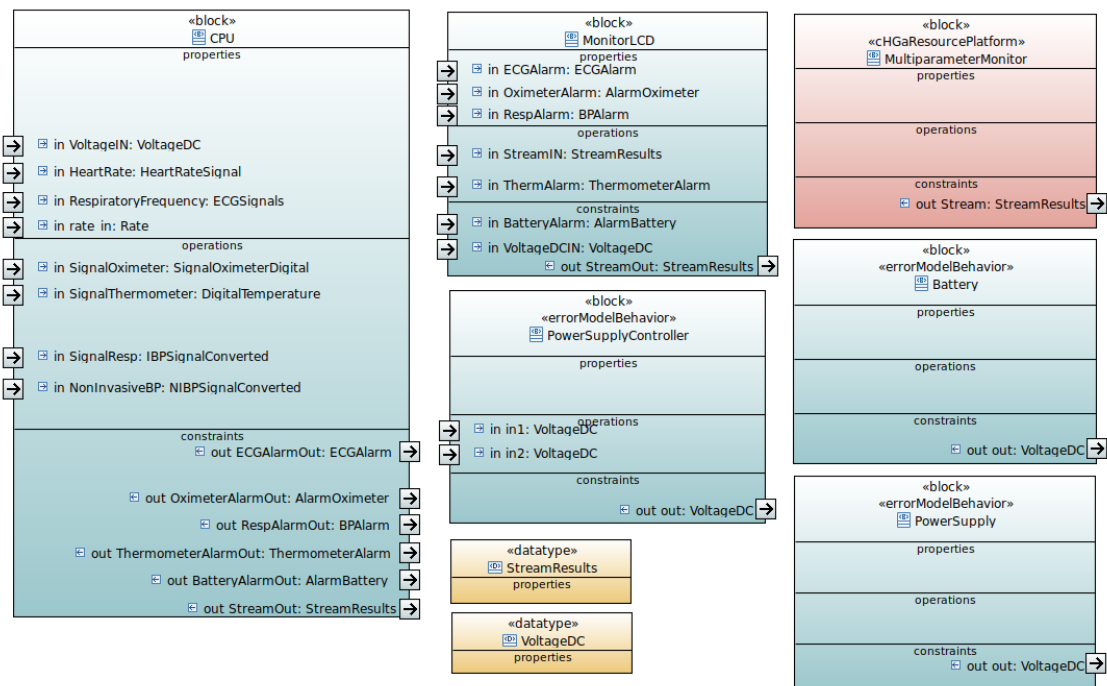


Figura 4.3: Diagrama de definição de blocos para o sistema de monitoramento multiparâmetro.

Ao criar o diagrama de bloco interno do monitor multiparâmetro, os CS de ECG, verificador de pressão arterial, termômetro e oxímetro de pulso precisaram ser adicionados. Portanto, os blocos que representam o mais alto nível desses CS citados (blocos de cor verde) foram adicionados ao diagrama interno de bloco do monitor multiparâmetro. Os modelos de ECG e respiração, verificador de pressão arterial, termômetro e oxímetro de pulso precisam de energia para funcionar. A energia é provida pelo `PowerSupplyController` para cada um desses CS, então existe o comportamento interno de cada CS (apresentado nas subseções a seguir) para coletar os sinais e prover como saída de cada módulo. Os resultados de cada CS são enviados para o CPU.

Ao receber todas essas informações providas de todos os CS do sistema (ECG e respiração, termômetro, oxímetro de pulso, verificador de pressão arterial e `PowerSupplyController`) a CPU encapsula todas essas informações através do datatype `StreamResults` e envia para o bloco `MonitorLCD`. A CPU também envia para o `MonitorLCD` alertas referentes às informações coletadas através dos CS. Esses alarmes são gerados a partir de parâmetros configurados pelo profissional da saúde que está operando o monitor multiparâmetro, tais configurações são realizadas de acordo com o tipo do paciente que está em monitoramento, seja o paciente uma criança, um adulto ou um idoso.

O bloco `MonitorLCD` representa a tela de informações sobre o monitoramento do paciente em formato de histórico, esta tela de informações fica configurada no leito ou estativa para fácil visualização. O `MonitorLCD` recebe todas essas informações providas através da CPU, e também recebe alimentação provida através do bloco `PowerSupplyController`. Com essas informações, o `MonitorLCD` possui apenas uma porta de saída contendo as informações e alarmes referentes ao monitoramento do paciente, a porta que provê todas essas informações é a porta `StreamOut` com o datatype `StreamResults`. Na Figura 4.4 é apresentado o diagrama de bloco interno que representa todo o sistema de monitoramento multiparâmetro de mais alto nível.

4.3.2 Sistema Constituinte de Gerenciamento de Energia

O sistema constituinte especificado para gerenciar a alimentação do sistema de monitoramento multiparâmetro não foi especificado através da definição de um diagrama de definição de blocos, e sim utilizando os blocos definidos no diagrama do monitor multiparâ-

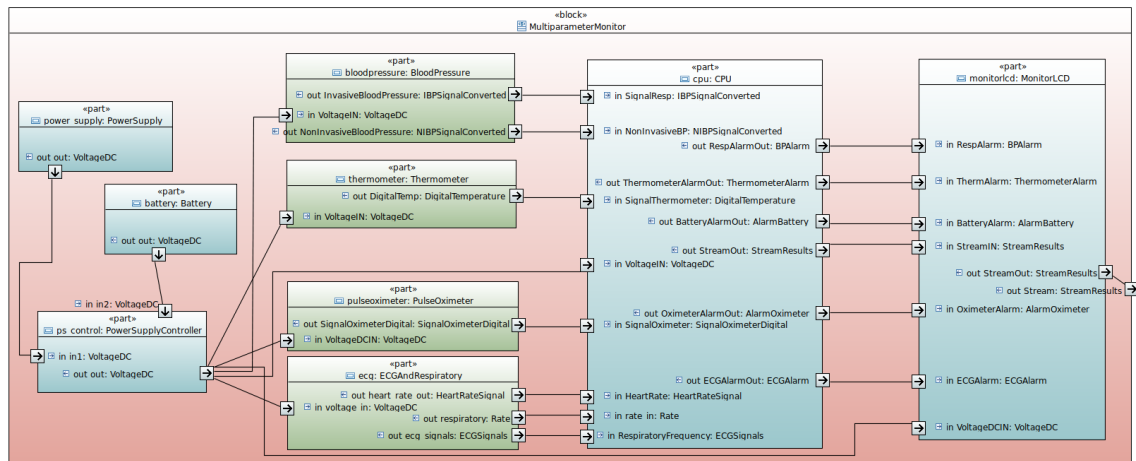


Figura 4.4: Diagrama de bloco interno para o monitor multiparâmetro.

metro. No diagrama de definição de blocos do monitor multiparâmetro, foram definidos três blocos referentes a alimentação do sistema, que são: `PowerSupply`, `Battery` e `PowerSupplyController`. Também para o gerenciamento de energia, foi definido o datatype `VoltageDC`.

Utilizando o datatype `VoltageDC`, o bloco `PowerSupply` através da porta `out` envia energia para o bloco `PowerSupplyController` que recebe essa informação através da porta `in1`. Também utilizando o datatype `VoltageDC`, o bloco `Battery` através da porta `out` envia energia para o bloco `PowerSupplyController` que recebe essa informação através da porta `in2`. A comunicação entre esses blocos pode ser observada através da Figura 4.5.

A partir desta comunicação definida entre os blocos `Battery`, `PowerSupply` e `PowerSupplyController`, foram definidas representações com informações relacionadas a modos de propagação de falhas desses blocos, utilizando o diagrama de máquina de estados do CHES-ML e o estereótipo `ErrorModel`. Essas representações foram definidas com o intuito de adicionar redundância quanto à fonte de alimentação principal do sistema, as quais são utilizadas quando os componentes referente à energia estão configurados com o estereótipo `ErrorModelBehavior`, podendo aumentar a confiabilidade de todo o sistema.

A Figura 4.6 é apresentado o modelo de erro baseado em estado que representa os comportamentos do sistema constituinte `Battery`. O modelo contém dois estados de erros:

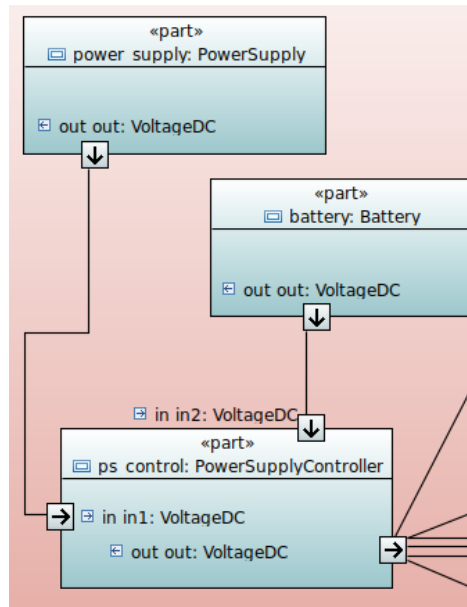


Figura 4.5: Comunicação entre os componentes que gerenciam energia do sistema.

BatteryFault, representando um defeito na bateria; e LackofCharge, representando a descarga da bateria. O estado inicial é o estado healthy, indicando que o sistema está na condição desejada (bom funcionamento). A transição para um estado de erro é disparada por uma falha interna que resulta em uma escolha entre BatteryFault e LackofCharge. Esses estados de erro possuem pesos diferentes, que representam uma maior probabilidade de ocorrência para LackofCharge (60%) do que para BatteryFault (40%). Neste modelo, foram utilizados os modos de falha omission (BatteryFault) e valueSubtle (LackofCharge). A distribuição de probabilidade exponencial ilustrada neste modelo é usada durante a análise de confiabilidade.

Para a fonte de alimentação (PowerSupply) as informações referentes à propagação da falha e apenas um modo de falha são possíveis observar na Figura 4.7. É apresentado o modelo de erro baseado em estado que representa os comportamentos do PowerSupply. Semelhante ao modelo apresentado na Figura 4.6, foi definido um estado desejado (healthy) e um estado de erro (PowerSupplyFault). O estado PowerSupplyFault representa um defeito ou falta de eletricidade na rede elétrica. Neste caso, foi utilizado apenas um modo de falha omission para representar a falha. A distribuição de probabilidade exponencial ilustrada neste modelo também é utilizada durante a análise de confiabilidade.

O bloco PowerSupplyController que recebe e gerencia energia dos blo-

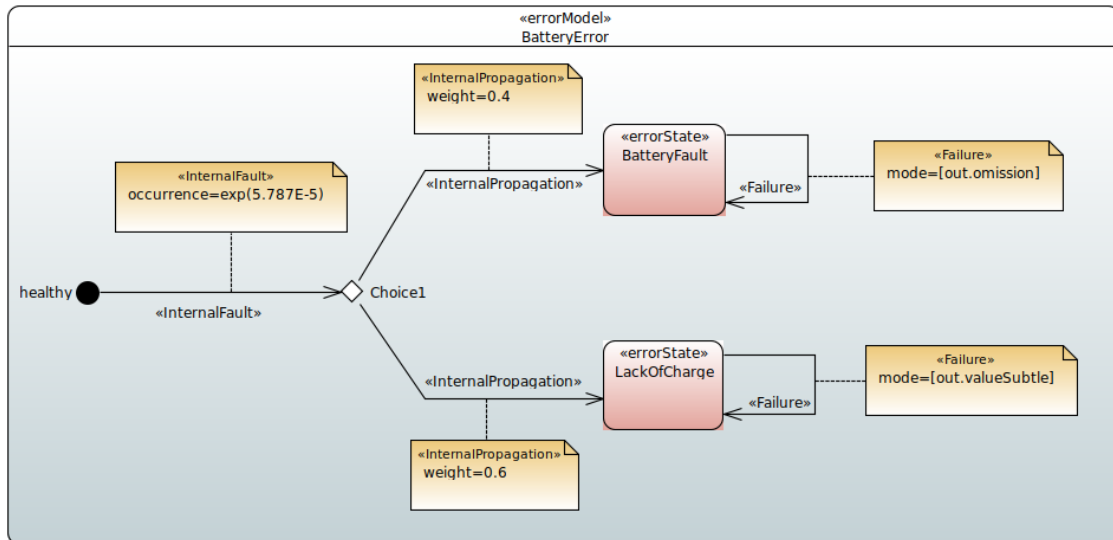


Figura 4.6: Máquina de estado representando o modo de falha para bateria.

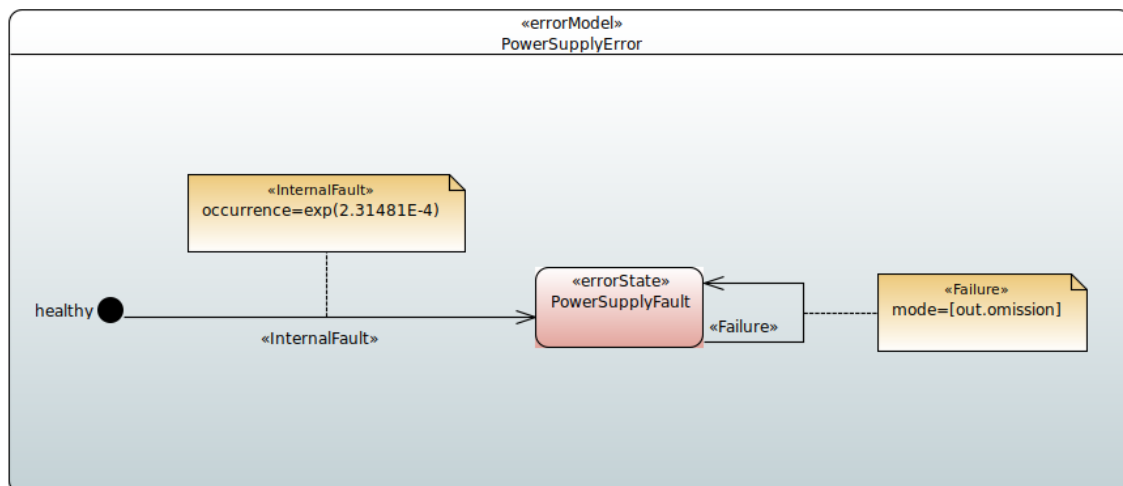


Figura 4.7: Máquina de estado representando o modo de falha para fonte.

cos Battery e PowerSupply, integra os modos de falha dos CS Battery e PowerSupply. Na Figura 4.8 é apresentado o modelo de erro baseado em estado que representa os comportamentos do sistema constituinte PowerSupplyController. O estado inicial deste modelo é um estado healthy, representando uma condição desejada genérica do sistema. Este modelo de erro também contém dois estados de erro (condições indesejadas): BatteryFault, representando um defeito ou falta de carga na bateria (ou seja, o modelo de falha da bateria retornou omission ou valueSubtle); e NoEnergyAvailable, representando que nenhuma fonte de alimentação está disponível,

onde que a bateria retornou `omission` ou `valueSubtle` através da porta `in2`, e a fonte de alimentação retornou `omission` através da porta `in1`. Portanto, a especificação representa que um sistema de monitoramento multiparâmetro só se torna indisponível se a fonte de alimentação principal (`PowerSupply`) e a fonte de alimentação redundante (`Battery`), ficarem indisponíveis.

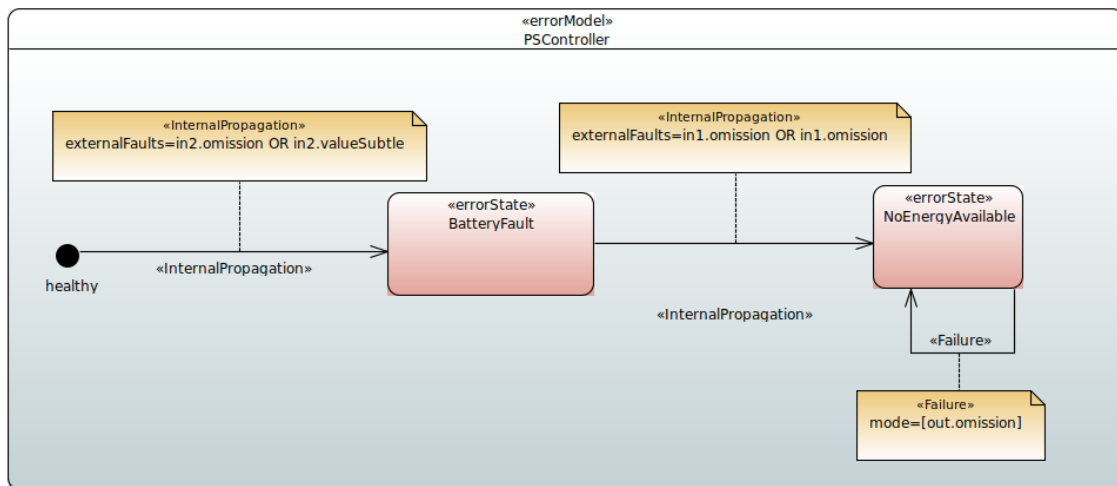


Figura 4.8: Máquina de estado representando o modo de falha para controlador da fonte.

4.3.3 Sistema Constituinte de ECG e Verificador de Respiração

O CS de ECG e verificador de respiração são compostos por blocos que representam eletrodos, amplificador de sinal, filtros e conversores. Para realizar a verificação de respiração do paciente, também são utilizados os mesmos eletrodos para coletar informações referente à frequência cardíaca, e então a informação é transmitida para amplificadores, filtros e conversores específicos que tratam das informações referentes à respiração do paciente.

No diagrama de definição de blocos, foram definidos os blocos: `Electrodes`, `AmplifierECG`, `AmplifierRespiratory`, `FiltersECG`, `FilterRespiratory`, `ConverterECG`, `ConverterRespiratory` e `ECGandRespiratory`. No diagrama de definição de blocos, também foram definidos os *datatypes*, `RateRespiratory`, `SignalBioElectric`, `HeartRateSignal`, `ECGsignals` e `ECGAlarm` que são utilizados como tipo nas portas de comunicação definida em cada `block`. É possível observar o modelo de definição de blocos para o

módulo do ECG e respiração na Figura 4.9.

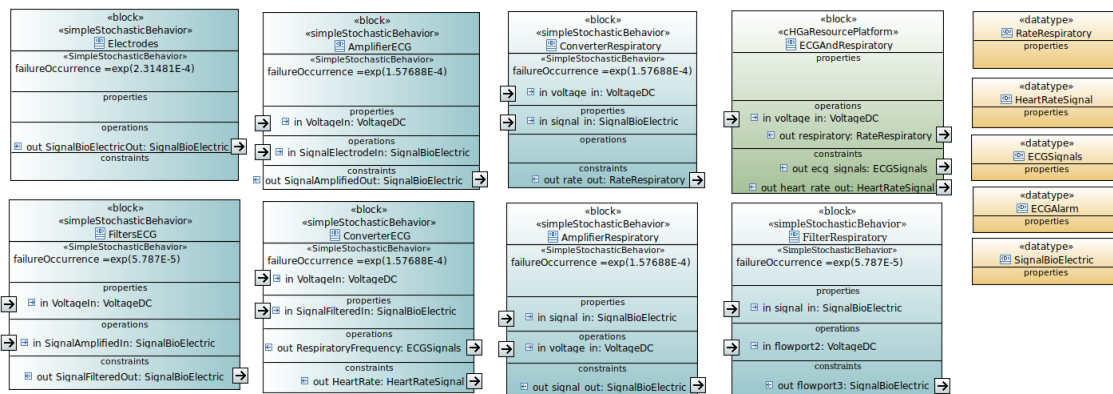


Figura 4.9: Diagrama de definição de blocos para o módulo de ECG e verificador de respiração.

Para o funcionamento de todos os módulos, é necessário que o módulo seja alimentado por energia, que pode ser provida através da bateria ou fonte e distribuído a todo o sistema através do bloco `PowerSupplyController`. Ao ter a energia como entrada no módulo de ECG e respiração, a energia é direcionada para os blocos `AmplifierECG`, `AmplifierRespiratory`, `FiltersECG`, `FilterRespiratory`, `ConverterECG` e `ConverterRespiratory` através das portas de comunicação definidas em cada um desses blocos com o *datatype* `VoltageDC`, que representa uma energia contínua. Com o módulo de ECG e respiração sendo alimentado por energia, dois processos acontecem, um para a saída de sinal referente à frequência cardíaca do paciente e outro para a respiração do paciente.

Para a frequência cardíaca do paciente, o bloco `AmplifierECG` pode receber o sinal analógico coletado do bloco `Electrodes`, no *datatype* `SignalBioElectric`, tal sinal é amplificado e enviado para os filtros. No bloco de `FiltersECG`, são aplicados filtros de passa baixa, passa alta e passa banda para a remoção de ruídos no sinal. Após a aplicação de filtros no sinal, o sinal é enviado para o bloco `ConverterECG`, cuja função é converter o sinal de analógico para digital e de prover duas informações, que são: sinal de frequência cardíaca e sinais do ECG, através dos *datatypes* `HeartRateSignal` e `ECGSignals`.

As informações de respiração do paciente, o processo é iniciado através do bloco `AmplifierRespiratory` que recebe o sinal analógico coletado do bloco `Electrodes`, no *datatype* `SignalBioElectric`, onde esse sinal é amplificado e envi-

ado para os filtros. No bloco de `FilterRespiratory`, são aplicados filtros para remoção de ruídos no sinal. Após a aplicação de filtros no sinal, o sinal é enviado para o bloco `ConverterRespiratory`, que possui a função de converter o sinal de analógico para digital e de prover a informação referente a taxa de respiração do paciente através do *datatype* `RateRespiratory`.

Na Figura 4.10 é apresentada a comunicação dos blocos através das portas definidas em cada bloco com os *datatypes* atribuídos no diagrama de blocos interno do ECG e respiração.

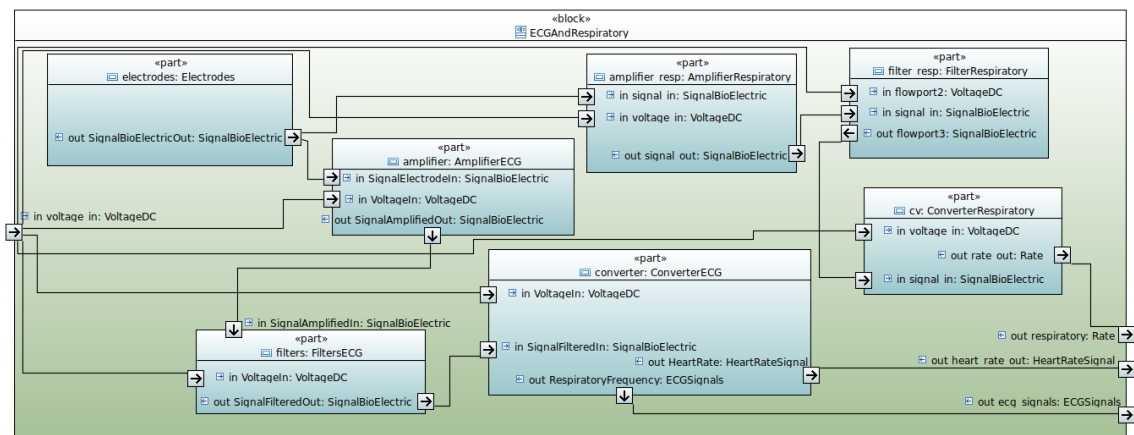


Figura 4.10: Diagrama de bloco interno para o módulo de ECG e verificador de respiração.

4.3.4 Sistema Constituinte de Termômetro

O módulo de um termômetro para um sistema de monitoramento multiparâmetro, em sua organização é bem similar ao sistema constituinte do ECG e respiração. No diagrama de definição de blocos do termômetro, são definidos os blocos `Filter`, `Converter`, `Amplifier`, `SensorPTC` e `Thermometer`. No diagrama de definição de blocos do termômetro, também foram definidos os *datatypes* utilizados neste módulo, que são: `DigitalTemperature`, `ThermometerSignalCaptured` e `ThermometerAlarm`. Na Figura 4.11 é apresentado o diagrama de definição de blocos para o sistema constituinte do termômetro.

Ao ser alimentado por energia provida da fonte ou bateria, o sistema constituinte de termômetro inicia o seu funcionamento para capturar sinais do paciente a fim de medir a sua temperatura. O sistema constituinte de termômetro recebe a energia em forma de corrente

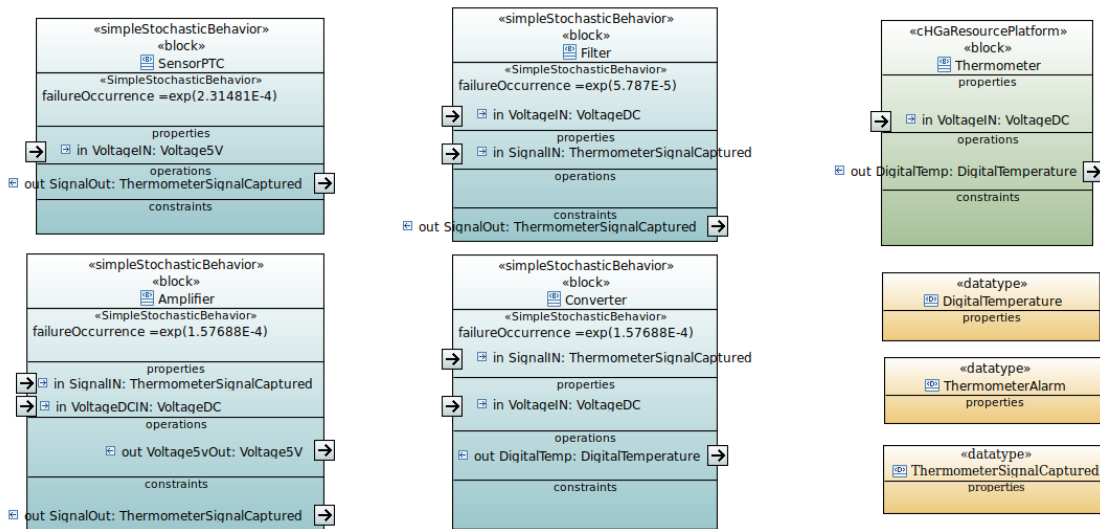


Figura 4.11: Diagrama de definição de blocos para o sistema constituinte de Termômetro.

continua, onde é encaminhada para os blocos Converter, Amplifier e Filter, após isso, o bloco Amplifier envia energia para o bloco SensorPTC (que representa um sensor para realizar a captura de sinal de temperatura), que inicia a captura de sinal do paciente de forma analógica. Após o bloco SensorPTC capturar o sinal referente à temperatura do paciente, o sinal é enviado para o bloco Amplifier que amplifica o sinal e envia para o bloco Filter, onde filtros são aplicados no sinal para a remoção de ruídos. Após a referida etapa, o sinal é enviado para o bloco Converter, que possui a funcionalidade de converter o sinal de analógico para digital. Após o sinal ser convertido de analógico para digital, o bloco Converter envia como saída para o bloco que representa todo o termômetro o sinal convertido de analógico para digital. Na Figura 4.12 é apresentado o diagrama de bloco interno, onde contém a comunicação entre os blocos para o modelo do termômetro.

4.3.5 Sistema Constituinte de Oximetria de Pulso

O sistema constituinte de oxímetro de pulso detém a responsabilidade, dentro de um sistema de monitoramento multiparâmetro, de medir a saturação de oxigênio do paciente que está sendo monitorado. Para que sua responsabilidade seja cumprida, foram definidos os blocos Leds, PhotoReceptor, Amplifier, Filters, Converter e PulseOximeter, este último de maior nível hierárquico, representa toda a comunica-

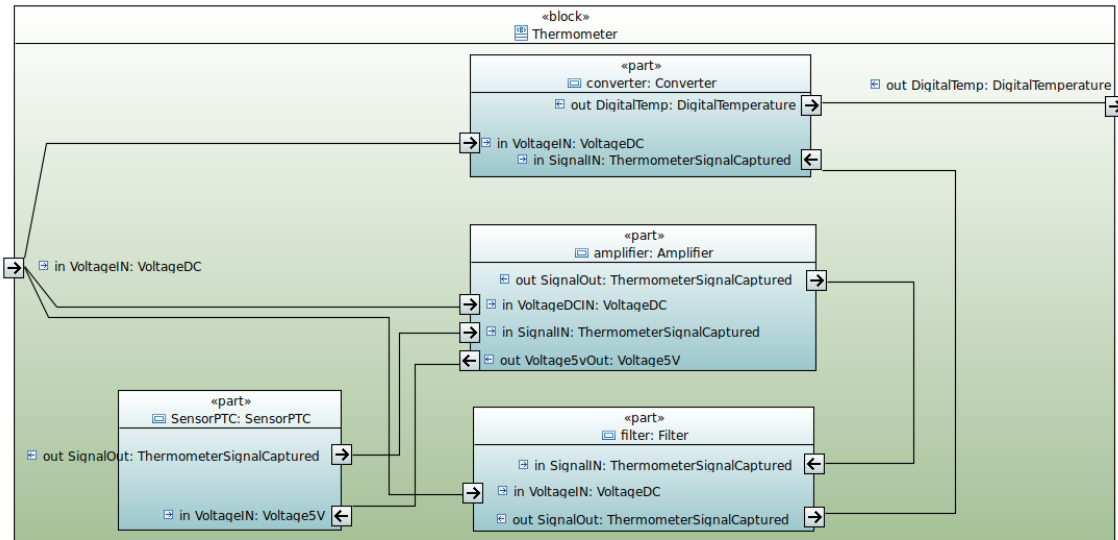


Figura 4.12: Diagrama de bloco interno para o sistema constituinte de Termômetro.

ção do modelo no diagrama de bloco interno. Para o oxímetro de pulso, têm-se dois blocos (Leds e PhotoReceptor) que representam os sensores para a coleta de sinais do paciente. No diagrama de definição de blocos para o sistema constituinte do oxímetro de pulso, também foram definidos os datatypes utilizados neste CS, que são: Voltage5V, LigthCaptured, SignalOximeterDigital, AlarmOximeter. Na Figura 4.13 é apresentado o diagrama de definição de blocos para o sistema constituinte de oxímetro de pulso, onde contém a definição dos blocos, portas de comunicação e datatypes.

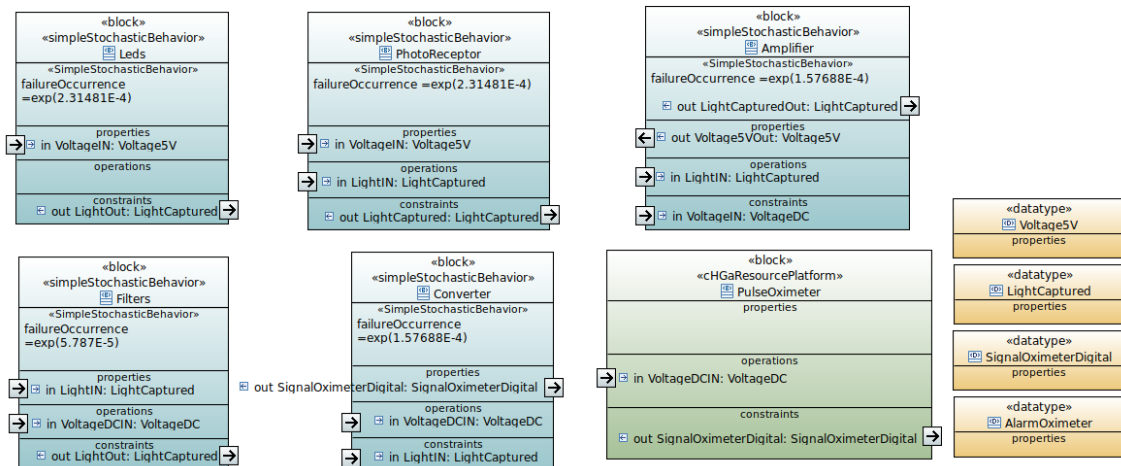


Figura 4.13: Diagrama de definição de blocos para o sistema constituinte de Oxímetro de Pulso.

O funcionamento do sistema constituinte do oxímetro de pulso inicia-se através da recepção de energia provida por fonte ou bateria. Três blocos são responsáveis por receber a energia em formato de corrente contínua, são os blocos: Amplifier, Converter e Filters. O bloco Amplifier, após receber a energia em formato de corrente contínua, envia energia para os blocos de Leds e PhotoReceptor em uma tensão menor de 5V através do *datatype* Voltage5V. O bloco Leds emite uma luz na pele do paciente, e a partir disso o bloco PhotoReceptor realiza a captura de sinal analógico do paciente referente à saturação de oxigênio. Com o sinal capturado, o bloco PhotoReceptor envia o sinal para o bloco Amplifier, onde é realizada a amplificação do sinal e enviado para aplicar filtros a fim de remover ruídos do sinal. Após a remoção de ruídos realizado pelos filtros, o sinal é enviado para o conversor, que converte o sinal de analógico para digital, e retorna o sinal como saída representando a saturação de oxigênio do paciente. Na Figura 4.14 é apresentado o diagrama de bloco interno, que contém a comunicação e organização dos blocos referentes ao modelo do oxímetro de pulso.

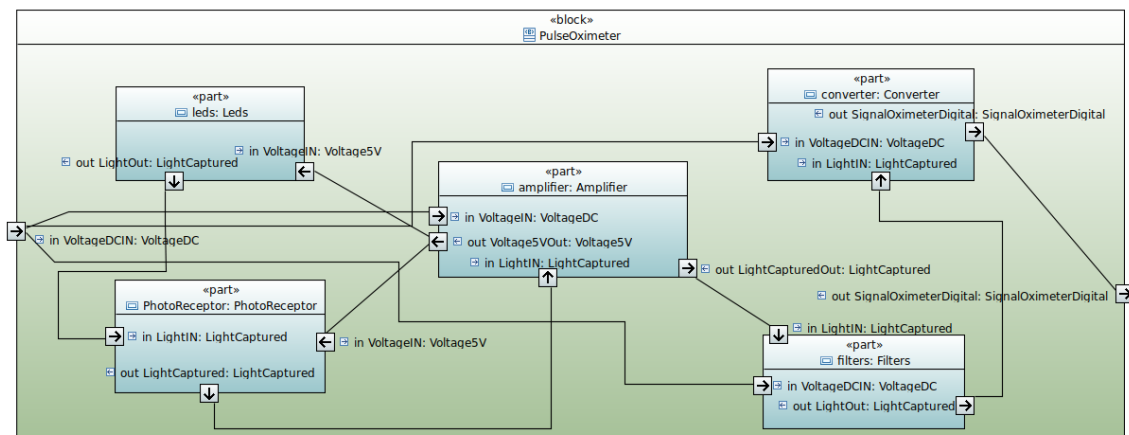


Figura 4.14: Diagrama de bloco interno para o sistema constituinte de Oxímetro de Pulso.

4.3.6 Sistema Constituinte de Verificação de Pressão Arterial

O sistema constituinte responsável por realizar a verificação de pressão arterial de um paciente em um monitor multiparâmetro possui em seu diagrama de definição de blocos, os blocos: Sensor, Amplifier, Filter, Converter e BloodPressure, este último, o bloco de maior nível hierárquico deste CS, ao qual representa toda

a comunicação dos blocos definidos no diagrama de definição de blocos para o verificador de pressão arterial do paciente. No diagrama de definição de blocos para a verificação de pressão arterial do paciente, também foram definidos os *datatypes* `SignalCaptured`, `IBPSignalConverted`, `NIBPSignalConverted` e `BPAlarm`. O *datatype* `IBPSignalConverted` representa o sinal convertido para a informação de pressão arterial invasiva, e `NIBPSignalConverted` representa o sinal convertido de pressão arterial não invasiva. Na Figura 4.15 é apresentado o diagrama de definição de blocos para o sistema constituinte de verificação de pressão arterial de um monitor multiparâmetro.

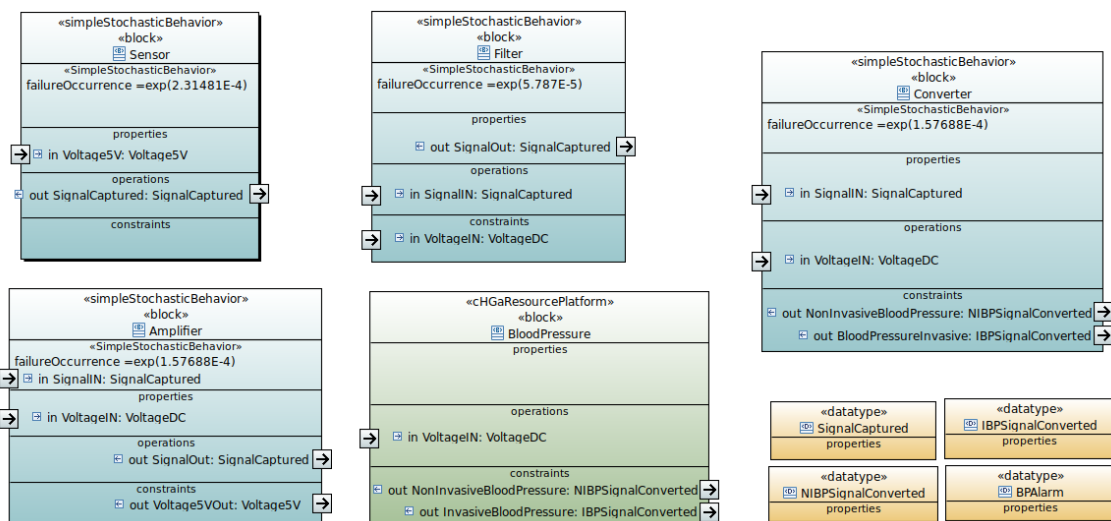


Figura 4.15: Diagrama de definição de blocos para o sistema constituinte de Verificador de Pressão Arterial.

Com os blocos do sistema constituinte de verificação de pressão arterial definidos no diagrama de definição de blocos, foi definido o diagrama de definição de bloco interno no bloco `BloodPressure`. O sistema constituinte de verificação de pressão arterial inicia o funcionamento com a recepção de energia distribuída para os blocos `Converter`, `Amplifier` e `Filter`. O bloco `Amplifier` recebe a corrente contínua provida da bateria ou fonte, converte em uma tensão menor de 5V e envia essa tensão para o bloco `Sensor`, que precisa de uma alimentação para iniciar a captura de sinal do paciente. Ao realizar a captura de sinal analógico do paciente, o bloco `Sensor` envia o sinal capturado para o bloco `Amplifier`, que realiza a amplificação do sinal e envia para o sinal amplificado para o bloco `Filter`.

Com o sinal no bloco *Filter*, são aplicados filtros para a remoção de ruídos desse sinal e então o sinal é enviado para o bloco *Converter*. No bloco *Converter*, o sinal é convertido de analógico para digital, podendo ser retornadas dois tipos de informação, que são: pressão arterial invasiva representada pelo tipo de dado *IBPSignalConverted* ou pressão arterial não invasiva representada pelo tipo de dado *NIBPSignalConverted*. Na Figura 4.16 é apresentado o diagrama de bloco interno onde contém a comunicação entre os blocos para o sistema constituinte de verificação de pressão arterial.

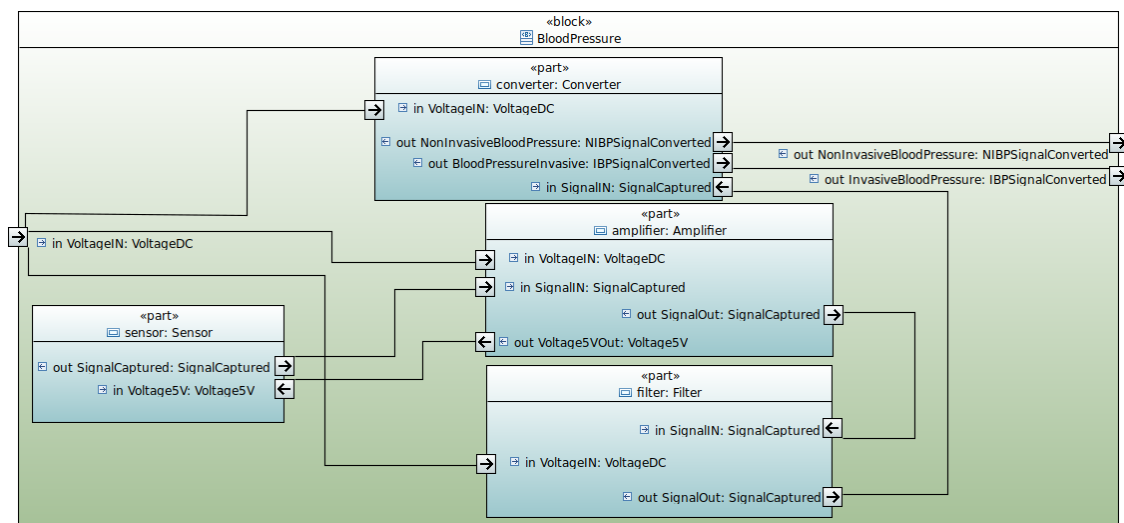


Figura 4.16: Diagrama de bloco interno para o sistema constituinte de Verificador de Pressão Arterial.

4.4 Considerações Finais

Ao longo deste capítulo foi apresentado em detalhes como estão organizados os sistemas de monitoramento multiparâmetro que foram citados como referência de modelos utilizados no mercado atualmente [2] [3]. Além disso, foi detalhado do que são compostos tais sistemas. Através disso, foi realizada a elicitação de requisitos de um sistema de monitoramento multiparâmetro, onde informações de funcionamento foram coletadas a partir de revisões da literatura e a partir de entrevistas realizadas com um profissional de eletrônica com mais de quinze anos de experiência.

As informações apresentadas na seção 4.1 foram de suma importância para que a

modelagem de um sistema de monitoramento multiparâmetro fosse desenvolvida utilizando CHES-ML, e que também modos de falha fossem definidos para o sistema constituinte de energia. A partir do modelo desenvolvido com o CHES-ML apresentado neste capítulo, foi possível realizar análises de confiabilidade com os estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`, e que estratégias de manutenção possam ser configuradas e simuladas com este modelo desenvolvido, podendo realizar comparações sobre a confiabilidade do modelo entre os estereótipos utilizados e com ou sem estratégias de manutenção. Portanto, as análises executadas com este modelo podem apresentar aos gestores de hospitais o ganho de confiabilidade que é possível obter utilizando estratégias de manutenção e adicionando redundância na alimentação do sistema.

Capítulo 5

Análises de Confiabilidade

Neste capítulo são apresentadas as análises de confiabilidade executadas do modelo desenvolvido de sistemas de monitoramento multiparâmetro para uma UTI. Foi definida a execução das análises de confiabilidade do monitor multiparâmetro em UTIs aos quais foram determinados seis intervalos de tempo distintos, que foram: três meses, seis meses, nove meses, doze meses (um ano), dezoito meses (um ano e meio) e vinte e quatro meses (dois anos). Assim, foi possível observar o nível de confiança do monitor multiparâmetro ao decorrer de dois anos em UTIs. Também foi executada a análise de confiabilidade para o tempo médio que um paciente fica internado em UTIs na pandemia da COVID-19, que é de doze dias [32].

As análises de confiabilidade foram executadas utilizando os estereótipos aplicado aos componentes `SimpleStochasticBehavior` e `ErrorModelBehavior` do CHES-SBA. Foram obtidos resultados sem estratégias de manutenção aplicadas ao monitor multiparâmetro, e com estratégias de manutenção. De acordo com o profissional de eletrônica em sistemas médicos entrevistado para este trabalho, os hospitais realizam manutenções preventivas em suas UTIs em períodos de 6 meses e 1 ano. Com isso, este trabalho apresenta análises de confiabilidade de um monitor multiparâmetro com estratégias de manutenção para 6 meses e 1 ano. Além das estratégias de manutenção definidas e utilizadas por hospitais atualmente, este trabalho também propõe estratégias de manutenção em períodos de 1 mês e 3 meses, apresentando o ganho dessas propostas em relação às estratégias de manutenção utilizadas atualmente (6 meses e 1 ano) e também sem manutenções.

As próximas seções apresentam a configuração das informações de probabilidade e estereótipo nos componentes e os resultados coletados com o estereótipos

`SimpleStochasticBehavior` e `ErrorModelBehavior` sem estratégias de manutenções preventivas e com o estereótipo `ErrorModelBehavior` com estratégias de manutenção preventiva.

5.1 Configuração de Métricas Utilizadas

Para executar as análises de confiabilidade com o CHES-SBA, o primeiro passo é adicionar as informações de probabilidade de falha nos blocos modelados com o CHES-ML do monitor multiparâmetro. Como os CS do monitor multiparâmetro possuem componentes similares, como, por exemplo, filtros, conversores e amplificadores, a probabilidade de falha de um componente que é utilizado em vários subsistemas, é a mesma probabilidade de falha. A probabilidade de falha do componente amplificador do ECG, é a mesma probabilidade de falha do amplificador do oxímetro de pulso, termômetro e verificador de pressão arterial. Isso equivale também para os filtros e conversores.

Além dos componentes citados no parágrafo anterior, outros componentes similares são os componentes responsáveis por captar a informação juntamente ao paciente, como, por exemplo, eletrodos, sensores, *photo receptor* e *leds*. São componentes que possuem atribuições similares de coletar o sinal, porém nomes diferentes de acordo com o tipo de componente para coletar o sinal em que cada CS utiliza. Na representação real, esses componentes representam componentes externos em um sistema de monitoramento multiparâmetro. A informação da probabilidade de falha de um componente externo coletada com o profissional de eletrônica entrevistado por esse trabalho foi utilizada nos componentes de eletrodos, sensores, *photo receptor* e *leds*.

No modelo de sistemas de monitoramento multiparâmetro desenvolvido para este trabalho, foram desenvolvidos três componentes que são responsáveis pela alimentação de todo o sistema do monitor multiparâmetro, que são: `Battery` (bateria), `PowerSupply` (fonte de alimentação principal) e `PowerSupplyController` (controlador de alimentação do sistema). A fonte possui uma probabilidade de falha, a qual é utilizada no componente `PowerSupplyController`. O componente bateria possui uma probabilidade de falha diferente da fonte e do controlador de alimentação do sistema.

Com base na Figura 4.4, que apresenta a comunicação de todos os componentes de sis-

temas de monitoramento multiparâmetro, é possível observar os componentes definidos no diagrama de definição de blocos do monitor multiparâmetro, e observar os CS que o compõem. Visualizando desta forma, os componentes que representam os CS não possuem probabilidade de falha, mas seus componentes internos sim. O componente CPU não possui probabilidade de falha, de acordo com o profissional de eletrônica entrevistado para este trabalho, por se tratar de um micro processador, possui um histórico de falha bem próximo de zero, sendo esse valor irrelevante para os tempos de análises neste trabalho.

O componente MonitorLCD, de acordo com o profissional de eletrônica, possui uma probabilidade de falha bem próxima a zero. De acordo com a modelagem do monitor multiparâmetro apresentado na Figura 4.4, ele tem a responsabilidade de apresentar todas as informações coletadas através dos outros componentes e subsistemas do monitor multiparâmetro. O componente MonitorLCD é responsável por enviar as informações para a única porta de saída do sistema de monitoramento multiparâmetro nomeada de Stream. Ao executar as análises de confiabilidade, o CHESS-SBA necessita que o componente responsável por enviar informações para a porta de saída de todo o sistema possua informações de probabilidade de falha. Portanto, foi atribuído o estereótipo SimpleStochasticBehavior com uma probabilidade de falha $\exp(1.0E - 20)$, com finalidade apenas de informar ao CHESS-SBA que esse componente possui uma informação de probabilidade de falha, porém com um valor muito próximo a zero que não afeta as análises de confiabilidade do monitor.

Assim, todos os componentes do monitor multiparâmetro desenvolvidos para este trabalho, exceto o componente CPU, estão contemplados com informações de probabilidade de falha. Este trabalho possui a definição de suas métricas para a realização de análise de confiabilidade baseada em horas. Embora as métricas utilizadas sejam baseadas em horas, as unidades de medida nunca são especificadas nos modelos com CHESS-SBA, é assumido que a mesma unidade de medida seja utilizada em todo o modelo para grandezas relacionadas a horas. Para executar as análises de confiabilidade o valor da variável t do atributo Reliability do estereótipo StateBasedAnalysis e também os valores da taxa de ocorrência de falha dos componentes estão baseado em horas. A Tabela 5.1 apresenta todos os componentes do sistema e as informações de probabilidade de falha utilizadas com a distribuição de probabilidade exponencial.

Todos os componentes exceto conversores e amplificadores, possuem informações de

taxa de ocorrência de falha em MTBF, onde esse valor utilizando a distribuição de probabilidade exponencial é o inverso do MTBF, que é representado através da fórmula $MTBF = \frac{1}{\lambda}$. É importante ressaltar que os valores apresentados na Tabela 5.1 exceto para os componentes conversores e amplificadores, representam o MTBF de cada componente, por exemplo, o componente bateria possui um MTBF de 17280 horas. Os componentes que utilizam MTBF como informação de taxa de ocorrência de falha são representados por horas contínuas de uso sem pausa. Os valores utilizados no amplificador e conversor foram coletados em porcentagem, o que representa que, amplificador e conversor possuem a probabilidade de falhar até 5 anos de 0.1%, tal valor foi adicionado ao chess em distribuição de probabilidade exponencial utilizando a função de distribuição acumulada (CDF).

Tabela 5.1: Taxa de ocorrência de falha dos componentes utilizando a distribuição exponencial.

Componente	Ocorrência de Falha	Distribuição Exponencial
Eletrodos	4320 horas - 6 meses	exp(2.31481E-4)
Leds	4320 horas - 6 meses	exp(2.31481E-4)
Sensores	4320 horas - 6 meses	exp(2.31481E-4)
PhotoReceptor	4320 horas - 6 meses	exp(2.31481E-4)
Fonte	4320 horas - 6 meses	exp(2.31481E-4)
Controlador de Fonte	4320 horas - 6 meses	exp(2.31481E-4)
Filtros	17280 horas - 2 anos	exp(5.787E-5)
Bateria	17280 horas - 2 anos	exp(5.787E-5)
Amplifier	0,1% em 5 anos	exp(2.3E-8)
Conversor	0,1% em 5 anos	exp(2.3E-8)

Com as referidas informações de probabilidade de falha, são executadas análises de confiabilidade para os intervalos de tempo de 12 dias (tempo médio que um paciente fica em uma UTI com COVID-19), 3 meses, 6 meses, 9 meses, 12 meses, 18 meses e 24 meses. É importante observar que as informações de probabilidade apresentadas por este trabalho de um sistema de monitor multiparâmetro são de referência ao uso em 24h por dia. Tomando como base o fato de que um dia possui 24 horas e um mês possui 30 dias, foi executada a multiplicação de $(24 * 30)$ e então definido para este trabalho que um mês possui 720 horas.

A Tabela 5.2 apresenta os valores em horas utilizados para realizar as análises nos intervalos de tempo apresentados no início deste parágrafo.

Tabela 5.2: Conversão de tempo (dias/meses) para horas.

Tempo - Dias/Meses	Horas
12 dias	288 horas
3 meses	2160 horas
6 meses	4320 horas
9 meses	6480 horas
12 meses	8640 horas
18 meses	12960 horas
24 meses	17280 horas

5.2 Análise com SimpleStochasticBehavior e ErrorModelBehavior

Foram executadas análises de confiabilidade com o CHES-SBA utilizando dois estereótipos, `SimpleStochasticBehavior` e `ErrorModelBehavior`. Todos os componentes do sistema modelado foram configurados com o estereótipo `SimpleStochasticBehavior` e tiveram as informações de probabilidade de falha adicionadas ao atributo `failureOccurrence`. Na Figura 5.1 é apresentado o componente `Filter` configurado com o estereótipo `SimpleStochasticBehavior`, onde o atributo `failureOccurrence` possui o valor de $exp(5.787E - 5)$, apresentado na Tabela 5.1.

Após configurar todos os componentes com o estereótipo `SimpleStochasticBehavior` e adicionar as informações de probabilidade de falha, foi criado um diagrama de classe UML e definido um componente com o estereótipo `StateBasedAnalysis` para configurar os parâmetros da execução das análises do monitor multiparâmetro onde todos os componentes estão configurados com o estereótipo `SimpleStochasticBehavior`. Na Figura 5.2 é apresentada a configuração do componente `MultiparameterMonitor` com o estereótipo `StateBasedAnalysis`. Foi configurado para a variável `measure` o valor de `Reliability {instantOfTime`

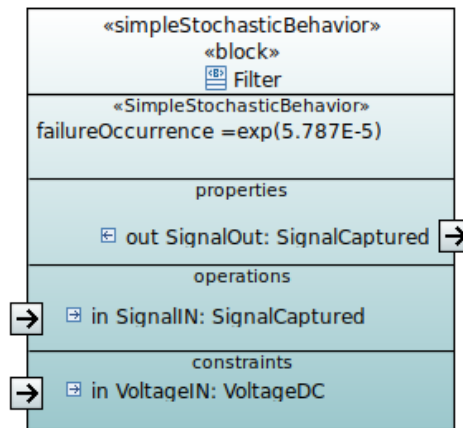


Figura 5.1: Componente Filter configurado com o estereótipo SimpleStochasticBehavior.

= 4320}, que representa que será realizada análise de confiabilidade em um intervalo de tempo de 4320 horas (6 meses), para o componente MonitorLCD do pacote de instâncias que representa todo o sistema de monitoramento multiparâmetro. Como na modelagem do sistema do monitoramento multiparâmetro, o componente que envia as informações para a porta de saída é o MonitorLCD, então, ele é atribuído para o atributo targetDepComponent, pois através dele é viável validar a probabilidade de falha de todo o sistema, se algum componente falhar, a confiabilidade do MonitorLCD será afetada.

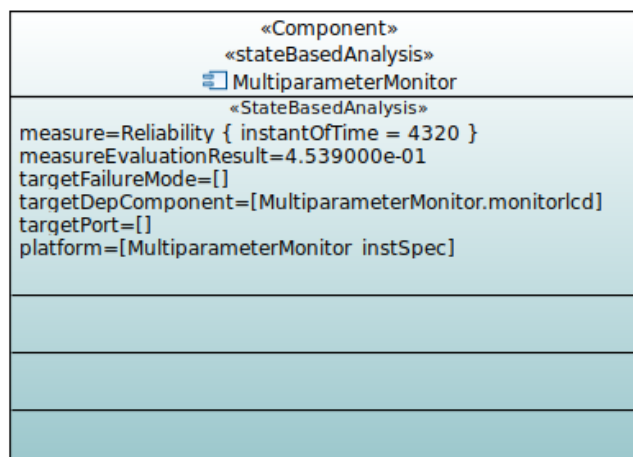


Figura 5.2: Componente MultiparameterMonitor configurado com o estereótipo StateBasedAnalysis para executar análise de confiabilidade.

Com todos os componentes configurados com estereótipo

SimpleStochasticBehavior, o processo de configuração do componente MultiparameterMonitor da Figura 5.2 foi executado em diversas oportunidades alterando o valor da unidade de tempo para obter a confiabilidade do monitor multiparâmetro nos intervalos de tempo da Tabela 5.2.

Para representar a falta de redundância no gerenciamento de energia ao sistema de monitoramento multiparâmetro foram executadas as análises onde todos os componentes estão configurados com o estereótipo SimpleStochasticBehavior. Após executar as análises de confiabilidade para o sistema de monitoramento multiparâmetro sem redundância para o gerenciamento de energia do sistema, foi realizada a configuração dos componentes relacionados à alimentação do sistema com as máquinas de estados desenvolvidas utilizando o estereótipo ErrorModel. Essa configuração representa a adição de tática de redundância para o gerenciamento de energia do sistema de monitoramento multiparâmetro através dos componentes Battery, PowerSupply e PowerSupplyController aos quais foram atribuídos os estereótipos ErrorModelBehavior em suas definições, e os diagramas de máquinas de estados BatteryError, PowerSupplyError e PSController foram adicionados aos componentes respectivamente. Após a referida configuração, todos os outros componentes do sistema permaneceram com a configuração do estereótipo SimpleStochasticBehavior. Posteriormente a configuração, foram executada novamente as análises realizando alterações na variável measure de acordo com o intervalo de tempo no componente MonitorMultiparametric, responsável por executar as análises.

Após executar as análises com todos os componentes configurados com os estereótipos SimpleStochasticBehavior e posteriormente configurar os componentes relacionados com alimentação do sistema com o ErrorModelBehavior, os resultados foram adicionados à Tabela 5.3. As análises apresentadas na Tabela 5.3 apresentam a confiabilidade do sistema de monitoramento multiparâmetro sem tática de redundância para o gerenciamento de energia do sistema (SimpleStochasticBehavior) e com tática de redundância para o gerenciamento de energia do sistema (ErrorModelBehavior), portanto é possível observar a confiabilidade do sistema nos dois cenários que as análises foram executadas. Vale ressaltar que para 24 meses utilizando o SimpleStochasticBehavior, o resultado de 0.011% foi obtido com o CHES diminuído o nível de confiança de 99% para 80%. Todos

os outros resultados de análises de confiabilidade foram obtidos com o nível de confiança em 99%.

Tabela 5.3: Resultados de confiabilidade utilizando os estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`.

Tempos de Execução	SimpleStochasticBehavior	ErrorModelBehavior
12 dias (COVID-19)	8.583000e-01	9.988000e-01
3 meses	3.274000e-01	9.505000e-01
6 meses	1.070000e-01	8.560000e-01
9 meses	3.395440e-02	7.585000e-01
12 meses	1.149509e-02	6.508000e-01
18 meses	1.180366e-03	5.017000e-01
24 meses	1.146667e-04 (*)	3.814000e-01

5.3 Estratégias de Manutenção

Após deter os resultados de confiabilidade de um monitor multiparâmetro através dos estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`, foram configuradas estratégias de manutenção preventiva apenas para os componentes relacionados ao gerenciamento de energia do sistema (`Battery`, `PowerSupply` e `PowerSupplyController`). De acordo com o profissional de eletrônica entrevistado por este trabalho, as manutenções preventivas realizadas pelos hospitais em seus sistemas médicos, são feitas em intervalos de 6 meses e 12 meses. De acordo com entrevistas realizadas com o profissional de eletrônica as manutenções preventivas são executadas em um período de tempo curto e possuem um alto índice de sucesso, assim este trabalho inferiu os valores de probabilidade de sucesso da manutenção de 95% com duração de 5 minutos.

A Tabela 5.4 e a Figura 5.3 apresentam os resultados de confiabilidade sem a definição de estratégias de manutenção preventiva, onde os componentes `Battery`, `PowerSupply` e `PowerSupplyController` estão configurados com o estereótipo `ErrorModelBehavior` e o restante dos componentes de todo o modelo com o estereótipo `SimpleStochasticBehavior`. Também apresenta a confiabilidade do monitor

multiparâmetro com estratégias de manutenção preventiva definidas para 6 meses e 1 ano, nos quais os componentes `Battery`, `PowerSupply` e `PowerSupplyController` estão configurados com o estereótipo `ErrorModelBehavior` e o restante dos componentes configurados com o estereótipo `SimpleStochasticBehavior`.

Tabela 5.4: Confiabilidade com estratégias de manutenção preventiva ocorridas em 6 meses e 1 ano.

Tempos de Execução	Sem Manutenção	Manutenção de 1 ano	Manutenção de 6 meses
12 dias (COVID-19)	9.988000e-01	9.988000e-01	9.988000e-01
3 meses	9.505000e-01	9.505000e-01	9.505000e-01
6 meses	8.560000e-01	8.560000e-01	8.560000e-01
9 meses	7.585000e-01	7.585000e-01	8.163000e-01
12 meses	6.508000e-01	6.508000e-01	7.512000e-01
18 meses	5.017000e-01	5.792000e-01	6.605000e-01
24 meses	3.814000e-01	4.434000e-01	5.888000e-01

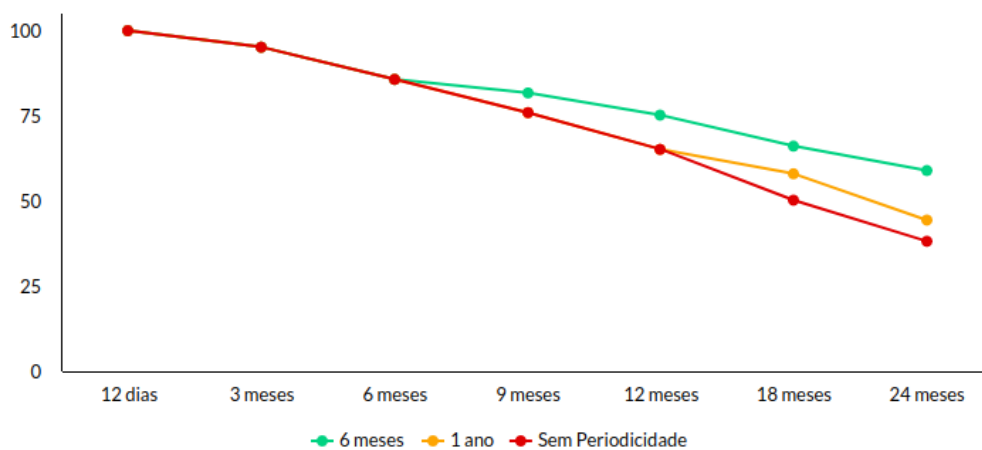


Figura 5.3: Gráfico com a confiabilidade do Monitor Multiparâmetro sem estratégias de manutenção e com estratégias de manutenção para 1 ano e 6 meses.

Após simular as estratégias de manutenção preventiva que são realizadas nos hospitais atualmente com a metodologia CHES, este trabalho também realizou análises de confi-

abilidade com estratégias de manutenção preventiva para 1 mês e 3 meses, com o intuito de apresentar a confiabilidade que essas novas estratégias de manutenção preventiva trazem comparando com as estratégias de manutenção utilizadas atualmente. A confiabilidade do monitor multiparâmetro com estratégias de manutenção preventiva para 1 mês e 3 meses estão apresentadas na Tabela 5.5 e na Figura 5.4. Os valores de MTBF em horas contínuas utilizados para taxa de ocorrência de falha nos componentes justificam por exemplo, o baixo valor de confiabilidade do sistema de monitoramento multiparâmetro em 24 meses com uma estratégia de manutenção periódica configurada para cada 3 meses.

Tabela 5.5: Confiabilidade de um sistema de monitoramento multiparâmetro com propostas de estratégias de manutenção preventiva para 1 mês e 3 meses.

Tempos de Execução	Periodicidade de 1 mês	Periodicidade de 3 meses
12 dias (COVID-19)	9.988000e-01	9.988000e-01
3 meses	9.803000e-01	9.505000e-01
6 meses	9.646000e-01	9.093000e-01
9 meses	9.420000e-01	8.701000e-01
12 meses	9.372000e-01	8.381000e-01
18 meses	9.104000e-01	7.759000e-01
24 meses	8.919000e-01	7.374000e-01

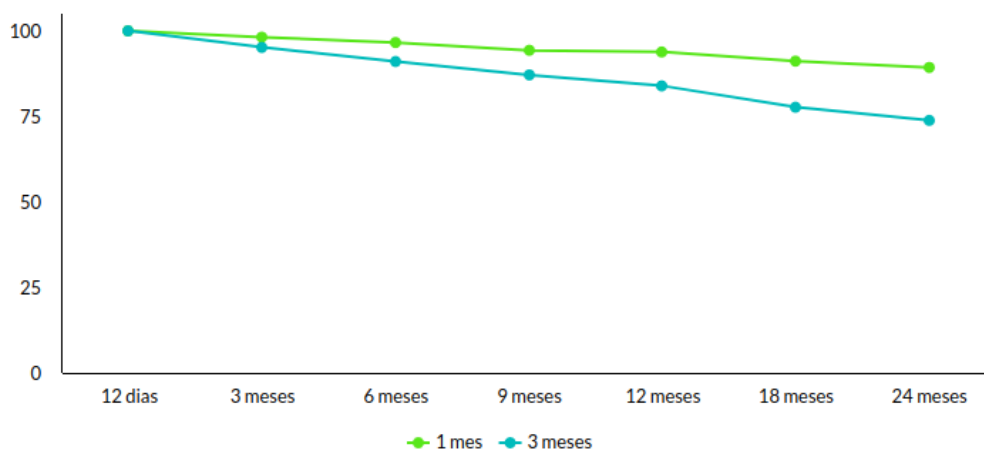


Figura 5.4: Gráfico com a confiabilidade do Monitor Multiparâmetro com estratégias de manutenção para 1 mês e 3 meses.

5.4 Considerações Finais

Este capítulo apresentou como foram realizadas as análises de confiabilidade e os resultados de confiabilidade de um monitor multiparâmetro desenvolvidas com a metodologia CHESS. Assim, este capítulo trilhou com as informações desde as métricas utilizadas com os valores de probabilidade de falha atribuídos aos componentes, tempos de execução de análises e como foram definidos, configuração dos estereótipos e estratégias de manutenção preventiva para execução das análises. A execução e resultado das análises de confiabilidade que este capítulo aborda, é de suma importância para a discussão que compara os resultados de confiabilidade entre os estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`, e também o ganho que a manutenção preventiva leva a confiabilidade do monitor multiparâmetro, podendo ajudar a gerência dos hospitais a adotar novas estratégias de manutenção preventiva principalmente em cenários de pandemia, como a vivida por hospitais de todo o mundo devido à COVID-19.

Capítulo 6

Discussões

Um grande número de características relacionadas ao meio ambiente e aos operadores pode impactar negativamente a confiabilidade dos sistemas de monitoramento multiparâmetro, como a baixa qualidade da rede elétrica e entradas incorretas de valores de parâmetros. No entanto, tais atores externos estão fora do escopo deste estudo. Portanto, apenas os componentes internos dos sistemas são tratados. Assim, as seções deste capítulo realizarão discussões e comparativos sobre os resultados de análise de confiabilidade sem manutenção e com manutenção obtidos com o CHESS, e também apresentarão dificuldades encontradas e sugestões de melhorias para o arcabouço CHESS.

6.1 Resultados de Análise

Foram analisados os impactos dos sistemas constituintes na confiabilidade dos sistemas de monitoramento multiparâmetro. A fonte de alimentação principal tem o maior impacto na probabilidade de falha, seguida pelo controlador da fonte de alimentação e bateria. Quando o controlador da fonte de alimentação falha, todo o sistema é afetado. O impacto dos outros CS (ECG e verificador de respiração, termômetro, oxímetro de pulso e verificador de pressão arterial) em todo o sistema é quase o mesmo, porque eles são compostos de componentes semelhantes (ou seja, amplificadores, filtros e conversores) que possuem probabilidade de falhas iguais entre os CS. A principal diferença entre o sistema constituinte (ECGAndRespiratory, Thermometer, BloodPressure e PulseOximeter) é o número de sensores usados na especificação de cada um.

A não utilização da técnica arquitetônica de redundância para o gerenciamento de energia do sistema de monitoramento multiparâmetro (Tabela 5.3, coluna 2), conforme esperado para um sistema novo, a probabilidade de que, após 288 horas (12 dias), o sistema não tenha falhado é de 85.83%. 12 dias é o tempo médio de internação em UTI de pacientes com COVID-19 [32]. Conforme o tempo de uso aumenta, a probabilidade de falha aumenta consideravelmente. Por exemplo, após 18 meses, a confiabilidade reduz para 0.11%. Na abordagem com o estereótipo `SimpleStochasticBehavior`, a propagação da falha ocorre instantaneamente impactando todos os componentes conectados, onde não existem definições de redundância para os componentes de gerenciamento de energia.

A inclusão da técnica arquitetônica de redundância de fontes de alimentação, utilizando o estereótipo `ErrorModelBehavior`, aumentou a confiabilidade do sistema para todos os cenários de tempo de uso (Tabela 5.3, coluna 3). Vale ressaltar que a confiabilidade apresentada para `ErrorModelBehavior` na Tabela 5.3 não possui estratégias de manutenção. Ao representar melhor o sistema adicionando a técnica de redundância para a alimentação de energia do sistema utilizando o estereótipo `ErrorModelBehavior`, houve um aumento nos resultados de análise de confiabilidade que se dá devido às relações entre as Figuras 4.6, 4.7 e 4.8. Quando a bateria (Figura 4.6) ou fonte de alimentação principal (Figura 4.7) falha, um modo de falha é emitido na porta de saída do sistema constituinte, exigindo o controlador de fonte de alimentação (Figura 4.8) para verificar a disponibilidade de eletricidade em todo o sistema. Se o componente de fonte de alimentação (`PowerSupply`) e o componente de bateria (`Battery`) emitirem falhas (`omission` ou `valueSubtle`), todo o sistema de monitoramento multiparâmetro será afetado e ficará indisponível. Por exemplo, esta redundância, no cenário de tempo de uso de 3 meses, aumentou a confiabilidade de 32.74% (`SimpleStochasticBehavior`) para 95.05% (`ErrorModelBehavior`); enquanto, para 24 meses, aumentou de 0.011% para 38.14%.

Em cenários de situação de emergência, a melhoria da confiabilidade ocasionada pela tática da redundância, é relevante para auxiliar na diminuição dos impactos da escassez de sistemas médicos nas UTIs. No cenário para COVID-19 (12 dias), foi obtido um ganho de confiabilidade de 85.83% (`SimpleStochasticBehavior`) para 99.88% (`ErrorModelBehavior`). Na Figura 6.1 é apresentado o comparativo da longevidade adquirida com as análises de confiabilidade executadas entre os estereótipos

`SimpleStochasticBehavior` e `ErrorModelBehavior`. Com a técnica de redundância aplicada ao gerenciamento de energia, pode-se observar o aumento da confiabilidade para todo o sistema de monitoramento multiparâmetro, onde todos os componentes do sistema estão mais assegurados de que estejam sendo alimentados por energia corretamente.

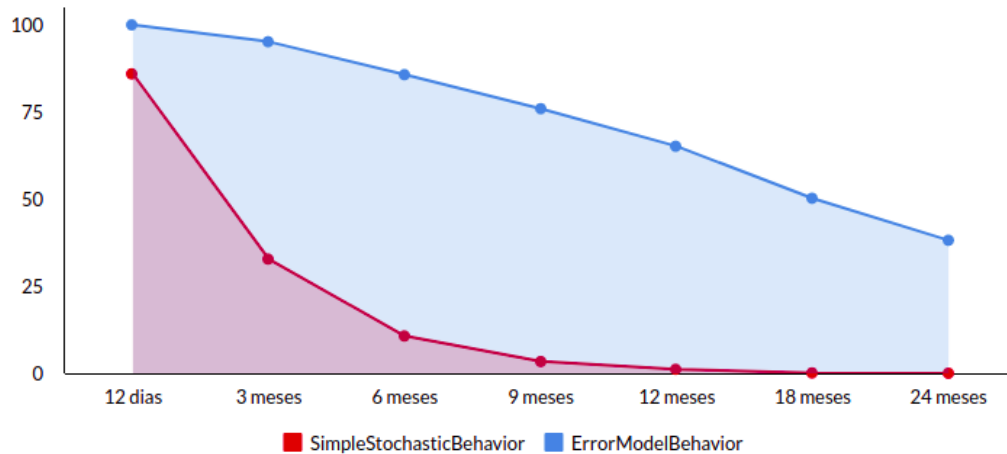


Figura 6.1: Longevidade do Monitor Multiparâmetro com os estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`.

Se a manutenção preventiva dos CS `PowerSupply`, `Battery`, e `PowerSupplyController` forem considerados, por um período padrão de tempo (a cada 6 ou 12 meses), a confiabilidade aumenta em um cenário de tempo de 6 meses (Tabela 5.4). Por exemplo, com o cenário de tempo de uso de 9 meses, a confiabilidade aumentou de 75.85% para 81.63%. Em um cenário do mundo real (por exemplo a pandemia de COVID-19), a manutenção preventiva pode fazer com que o número de sistemas médicos indisponíveis em UTIs seja cada vez menor, quando comparada aos resultados sem manutenção preventiva (Tabela 5.3, coluna 3). Portanto, os benefícios em realizar a manutenção preventiva são elevados, mostrando a relevância da análise de confiabilidade para auxiliar os gestores no planejamento de estratégias de manutenção. Os custos de manutenção preventiva de todo um sistema no ambiente hospitalar são baixos (por exemplo, limpeza e troca de capacitores), e diminuem ainda mais, se mais atenção for dada à realização de manutenções aos quais com ações simples evitam danos maiores a todo o sistema. A escolha entre manutenção periódica de 6 ou 12 meses depende dos recursos financeiros ou humanos de um hospital específico. Por exemplo, em países de baixa e média renda, há um grande

número de hospitais públicos em condições precárias. Em cenários de UTIs não pandêmicas (ou epidêmicas), a manutenção periódica de 12 meses pode ser suficiente para fornecer o mínimo de qualidade de serviço necessário, pois, no cenário de tempo de uso de 24 meses, por exemplo, há uma redução relativamente baixa de 14.45% em confiabilidade quando comparada à manutenção periódica de 6 meses. Caso contrário, a manutenção periódica de 6 meses é o melhor cenário.

Para considerar cenários de UTI em pandemia (ou epidemia), foram analisados mais dois cronogramas de manutenção preventiva (Tabela 5.5) para melhorar a discussão e fornecer intuições para orientar a definição de estratégias de manutenção. Usando uma manutenção periódica de 3 meses, para o cenário de tempo de uso de 12 meses, a confiabilidade aumentou de 65.08% (sem manutenção) para 83.81% (com manutenção); ao usar uma manutenção periódica de 1 mês, a confiabilidade aumentou de 65.08% (sem manutenção) para 93.72% (com manutenção). Portanto, há um aumento de 9.91% ao usar a manutenção periódica de 1 mês em vez da manutenção periódica de 3 meses. Esse ganho de confiabilidade é relevante para cenários de UTIs de pandemia (ou epidemia), devido à sazonalidade dos vírus (conhecidas como ondas). Por exemplo, em março de 2020, a organização mundial de saúde declarou uma situação de pandemia devido aos surtos de COVID-19 (a primeira onda). No final de outubro de 2020, alguns países europeus anunciaram medidas de bloqueio devido aos novos surtos de COVID-19 relatados (a segunda onda), representando uma diferença de apenas 8 meses entre a primeira e a segunda ondas. Além disso, até outubro de 2020, outros países, como Brasil e Estados Unidos da América, ainda enfrentavam a primeira onda, devido às características específicas de tais países. Considerando os curtos períodos entre as ondas (ou seja, 8 meses) ou o ponto final incerto da primeira, intervalos de tempo reduzidos para manutenção preventiva podem ajudar a abordar a escassez de sistemas médicos em UTIs. Por exemplo, se um sistema de monitoramento multiparâmetro for adquirido por um hospital no início da pandemia, com manutenção periódica de 6 meses, a confiabilidade é de 81.63% com um cenário de tempo de uso de 9 meses. Em contraste, para uma manutenção periódica de 3 meses, a confiabilidade aumenta para 87.01%; enquanto para uma manutenção periódica de 1 mês, a confiabilidade aumenta para 94.20%. A manutenção periódica de 1 mês, quando comparada a uma manutenção periódica de 6 meses, representa um aumento de 12.57% no cenário de tempo de uso de 9 meses (ou seja, dentro da janela de tempo da segunda

onda na Europa). Isso indica possíveis benefícios na definição de planos de manutenção de emergência para situações de pandemia/epidemias, mesmo para países de baixa e média renda.

Na Figura 6.2 é apresentado o ganho de confiabilidade que as manutenções preventivas podem agregar em um sistema de monitoramento multiparâmetro. É representado o comparativo das manutenções periódicas de 1 ano, 6 meses, 3 meses, 1 mês e sem manutenção.

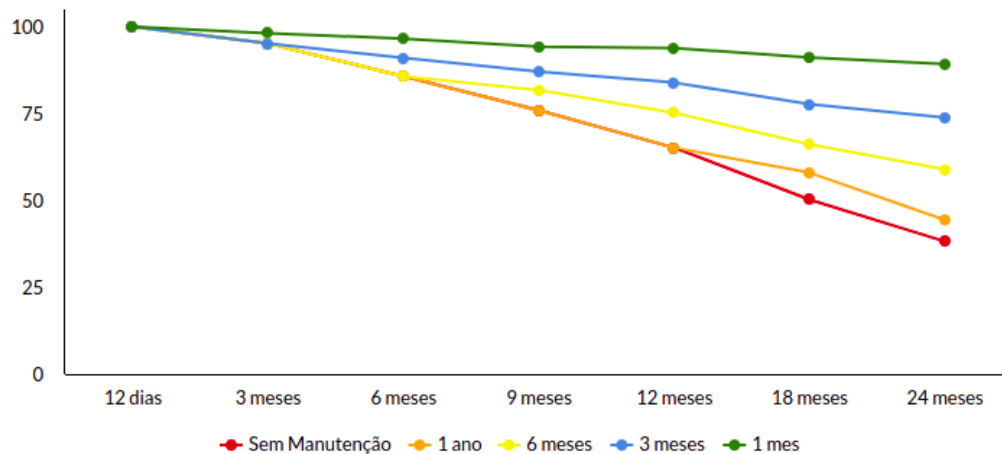


Figura 6.2: Longevidade do Monitor Multiparâmetro com estratégias de manutenção utilizadas em hospitais atualmente e propostas por este trabalho.

Estratégias de manutenção geralmente são vistas como centro de custos pelos gestores de ambientes hospitalares, porém são custos nos quais o investimento deve ser realizado de imediato. De acordo com informações coletadas através do profissional de eletrônica, a maior parte dos custos de manutenção preventiva são para efetuar a compra de peças, pois o custo com mão de obra já está dentro do planejamento do hospital, visto que os hospitais já possuem profissionais para executar manutenção ou contrato com alguma empresa terceirizada. Com estratégias de manutenção preventiva, os técnicos de manutenção realizam a observação dos equipamentos e podem detectar se sujeira no equipamento ou um capacitor danificado pode estar causando algum tipo de aquecimento, logo é mais viável financeiramente realizar a limpeza do equipamento ou a troca de um capacitor que possui um baixo custo, do que todo o equipamento ser danificado e ter que realizar a compra de um novo.

Equipes técnicas de manutenção de sistemas médicos possuem histórico de falhas de equipamentos e *checklists* de itens a checar, portanto, ao realizar uma manutenção preven-

tiva pode-se consultar o histórico de falhas e observar se as condições necessárias para que aquela falha ocorra estão acontecendo, através disso é possível evitar que sistemas médicos fiquem indisponíveis em ambientes de alta criticidade. As estratégias de manutenção preventiva definidas neste trabalho estão configuradas apenas para atacar os componentes `PowerSupply`, `Battery` e `PowerSupplyController` possuindo uma duração de 5 minutos e uma probabilidade de sucesso de 95%. Como a manutenção é voltada apenas para os componentes responsáveis por gerenciar a energia e com uma probabilidade de sucesso de 95%, o sistema de monitoramento multiparâmetro aumenta sua confiabilidade e assim é prolongada em um intervalo de tempo maior, mas por essas condições citadas não faz com que o sistema por completo volte ao estado de novo. É importante ressaltar que todos os outros componentes não são afetados pela manutenção, devido à propriedade *memoryless* da distribuição exponencial. Estratégias de manutenção preventiva como as supracitadas possuem baixo custo financeiro, porém podem prolongar consideravelmente a longevidade de um equipamento médico como as análises de confiabilidade deste trabalho apresentaram.

O modelo apresentado por este trabalho pode ser utilizado para a realização de novas análises de confiabilidade em diferentes intervalos de tempo e também com novos valores para taxa de ocorrência de falha dos componentes, visto que os valores de taxa de ocorrência de falha não são facilmente fornecido por empresas e especialistas. Assim, pesquisadores e por fabricantes podem utilizar o modelo afim de realizar testes e validações de novas configurações para a realização de análises de confiabilidade do equipamento.

Neste estudo, apesar do foco na aplicação da metodologia CHES para calcular a confiabilidade de sistemas de monitoramento multiparâmetro, a metodologia de análise de confiabilidade e as discussões apresentadas sobre manutenção são relevantes (e podem ser reutilizadas) para outros sistemas médicos, como respiradores por exemplo.

6.2 Dificuldades Encontradas e Sugestões de Melhorias

Através deste estudo, pode-se depreender que a metodologia CHES é uma ferramenta muito útil para a modelagem de sistemas, *software* e *hardware* utilizando UML ou SysML. Porém, algumas dificuldades foram encontradas durante a definição dos modelos com o CHES-ML como também durante as análises com o CHES-SBA. Ao realizar algumas

alterações em componentes que utilizam estereótipos do CHES-SBA e então executar as análises, algumas mensagens de erro eram retornadas e muitas vezes por algum erro de sintaxe, porém a mensagem de erro retornada não era tão clara informando qual tipo de erro era, não informando o componente que estava apresentando erro e onde estava ocorrendo o erro de sintaxe. A definição errônea de configurações na ferramenta realizadas pelo usuário poderiam ser informadas de forma clara através da ferramenta ajudando o usuário na utilização de forma correta.

Foi observado que a deleção de alguns componentes ou transições diretamente na região da ferramenta utilizada para definir a modelagem causava inconsistência de informação. Por exemplo, em alguns momentos ao realizar a deleção de um componente ou transição diretamente na região de definição, o objeto deletado ainda era visível na região *Model Explorer*, isso ocorre porque o *Papyrus* um dos elementos do arcabouço CHES-SBA para especificar modelos, tem a possibilidade de esconder um elemento do diagrama, ou apagar o elemento do modelo, por ser duas ações diferentes acaba criando uma confusão no usuário ao realizar manutenções no modelo. Dita inconsistência pode ser observada na definição de máquina de estados e também para a visão de *software*. Este comportamento fazia com que não fosse possível o CHES-SBA executar algumas análises de confiabilidade. Portanto, este trabalho utilizou algumas técnicas de cópia de segurança, nas quais a cada alteração realizada nos modelos desenvolvidos uma nova cópia era criada.

O CHES-SBA utiliza a ferramenta *DEEM Simulator* para executar as análises, um servidor desenvolvido e hospedado na Universidade de Florença (UNFI) e no Instituto de Ciência e Tecnologia da Informação (ISTI-CNR). Durante algumas execuções de análises de confiabilidade deste trabalho, o servidor DEEM se mostrou indisponível, onde era realizada uma tentativa de conexão, porém o servidor retornava uma mensagem de conexão recusada ou não retornava resultados permanecendo em uma mensagem de que estava sendo realizada uma conexão onde a conexão não era realizada e o resultado e análise de confiabilidade do sistema não era retornada.

Utilizando o estereótipo `ErrorModelBehavior` para representar o comportamento de falha do componente `PowerSupplyController`, foram realizadas tentativas de criar uma regra de falhas externas através do atributo `externalFaults` na transição com o estereótipo `InternalPropagation`. Essa regra de falha externa se dá em que o com-

ponente `PowerSupplyController` recebe do componente `Battery` a partir da porta `in2` e os modos de falha `omission` e `valueSubtle`. Do componente `PowerSupply` pode-se receber o modo de falha `omission` na porta `in1`. Logo, uma condição lógica que poderia ser utilizada simplificando a lógica aplicada no modelo apresentada na Figura 4.8, seria: `externalFaults=(in2.omission OR in2.valueSubtle) AND in1.omission`. Porém, o CHES-SBA não permitiu a execução desta lógica definida para tratar os modos de falhas.

No geral, o CHES se mostrou uma ferramenta muito importante para modelagem e análise de confiabilidade de sistemas, pois ele possui uma curva de aprendizado baixa e uma velocidade de aprendizado rápida, visto que existem documentações disponíveis para o CHES-ML e para o CHES-SBA. Uma sugestão para que a velocidade de aprendizado seja maior, seria a disponibilização de modelos didáticos, que possuíssem a definição de arquiteturas de *software*, *hardware*, sistemas, estratégias de manutenção, e componentes configurados com os estereótipos `SimpleStochasticBehavior`, `ErrorModelBehavior` e `FLABehavior` para análise de confiabilidade.

6.3 Considerações Finais

Este capítulo apresentou discussões sobre as análises de confiabilidade realizadas para um sistema de monitoramento multiparâmetro com a metodologia CHES. Foram identificados que os componentes referentes a alimentação do sistema, afetam todo o sistema por completo mais do que os outros sistemas constituintes e componentes. Por exemplo, se fonte de alimentação principal e a bateria se tornam indisponíveis, o controlador de alimentação não possui alimentação para distribuir para todo o sistema, afetando todo o restante do sistema, tornando-o indisponível por completo. Também foi identificado que componentes como `MonitorLCD` e `CPU` não interferem na probabilidade de falha do sistema, e que os CS possuem impactos iguais caso falhas sejam geradas.

Neste capítulo, também foram apresentados os comparativos de confiabilidade entre os estereótipos `SimpleStochasticBehavior` e `ErrorModelBehavior`, e discutidos os ganhos que a redundância neste caso aplicado para a alimentação do sistema trouxeram ao quesito confiabilidade. A partir dos resultados de confiabilidade com o

`ErrorModelBehavior`, foram aplicadas e simuladas estratégias de manutenção, as quais os hospitais utilizam atualmente (6 meses e 1 ano) e estratégias de manutenção propostas (1 mês e 3 meses). Com o ganho de confiabilidade, foram discutidas as vantagens de manutenção preventiva para cenários de pandemia como a do COVID-19 e a relação de custo-benéfico que um hospital pode ter ao realizar manutenção preventiva.

Por fim, este capítulo também apresentou detalhes de dificuldades encontradas ao utilizar a metodologia CHES, sugerindo possíveis melhorias.

Capítulo 7

Conclusões

A metodologia CHESSE possibilitou a análise de confiabilidade de sistemas de monitoramento multiparâmetro para Unidades de Terapia Intensiva (UTIs) considerando diferentes cenários de tempo de uso. A inclusão da redundância da fonte de alimentação impactou positivamente a confiabilidade dos sistemas em todos os cenários. Com a análise de confiabilidade, foi identificado que a fonte de alimentação principal e a bateria são os sistemas constituintes que apresentam maiores impactos negativos na confiabilidade total de todo o sistema em situações de falha. Em situações de emergência não pandêmica (epidemias), a manutenção preventiva com periodicidade de 6 meses e 12 meses apresenta um impacto relevante no aumento da confiabilidade dos referidos sistemas. Em situações de emergência (por exemplo, pandemias), intervalos de tempo reduzidos de manutenção preventiva, quando aplicados em curto período, mostraram-se estratégias (relação custo-benefício) promissoras para aumentar a qualidade dos sistemas de monitoramento multiparâmetro para UTIs. Portanto, a análise apresentada neste estudo é relevante para os gestores de UTIs para o planejamento de estratégias de manutenção para enfrentar situações de emergência, como a pandemia do COVID-19.

Para a continuação deste trabalho como trabalhos futuros, podem-se destacar alguns pontos que podem enriquecer as informações deste modelo e validação com padrões que o mercado utiliza, como, por exemplo a ISO 14971 [21] e a IEC 60812 [35] que abordam sobre gerenciamento de risco de dispositivos médicos e explica como modo de falha e análise de efeitos (*Failure Mode and Effect Analysis* - FMEA) e modos de falha, efeitos e variante de análise de criticidade (*Failure Mode, Effects & Criticality Analysis* - FMECA) são

planejados, documentados e executados respectivamente. Pode ser utilizado o estereótipo `FLABehavior` nos componentes do modelo desenvolvido por este trabalho para a geração de FTA e geração de tabelas FMEA. Outra evolução deste trabalho está voltada ao detalhamento de modo de falha para sensores, eletrodos e leds, onde podem ser desenvolvidas máquinas de estados com o estereótipo `ErrorModel` e adicionar técnicas de redundância para aumentar a confiabilidade desses componentes de sistemas de monitoramento multiparâmetro e estratégias de manutenção mais avançadas. Também como evolução deste trabalho podem ser analisados como verificar a confiabilidade de baterias aos quais sofrem desgastes na vida útil ao ser carregada e descarregada na sua utilização, fazendo com que ela não funcione da mesma forma em um estado de nova após a utilização por algum tempo.

Bibliografia

- [1] Chess modeling language. Disponível em <https://www.eclipse.org/chess/publis/CHESSMLprofile.pdf>. Acesso em: 09 nov. 2020.
- [2] Gima vital signs monitors. Disponível em https://www.gimaitaly.com/en/assets/pdfprodotti/243_eng.pdf. Acesso em: 22 dez. 2020.
- [3] Intellivue mx600 and mx700 patient monitor. Acesso em <http://5.imimg.com/data5/DM/FE/MY-3822441/philips-intellivue-patient-monitors.pdf>. Acesso em: 22 dez. 2020.
- [4] A. Abreu and A. C. Neto. Método computacionalmente intensivo para o cálculo dos intervalos de confiança para valores da função de confiabilidade. *XVI Simpósio de Pesquisa operacional & Logística da Marinha*, 2011.
- [5] H. Al-Hamadi, A. Gawanmeh, and M. Al-Qutayri. Formalizing electrocardiogram (ecg) signal behavior in event-b. In *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 55–60, Oct 2014.
- [6] 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, 2013.
- [7] E. R. Araújo et al. Implantação de monitor multiparamétrico com acesso remoto em leito de uti para cirurgia cardíaca. 2019.
- [8] M. W. Azeem, M. Ahsan, N. M. Minhas, and K. Noreen. Specification of e-health system using z: A motivation to formal methods. In *International Conference for Convergence for Technology-2014*, pages 1–6. IEEE, 2014.

- [9] F. Bause and P. S. Kritzinger. *Stochastic petri nets*, volume 1. Vieweg Wiesbaden, 2002.
- [10] I. Bazovsky. *Reliability theory and practice*. Courier Corporation, 2004.
- [11] L. Beydon, P. Y. Ledenmat, C. Soltner, F. Lebreton, V. Hardin, D. Benhamou, F. Clergue, and G. Laguenie. Adverse events with medical devices in anesthesia and intensive care unit patients recorded in the french safety database in 2005–2006. *Anesthesiology: The Journal of the American Society of Anesthesiologists*, 112(2):364–372, 2010.
- [12] G. Booch, J. Rumbaugh, and I. Jacobson. *UML: guia do usuário*. Elsevier Brasil, 2006.
- [13] L. P. Bressan, A. L. de Oliveira, L. Montecchi, and B. Gallina. A systematic process for applying the chess methodology in the creation of certifiable evidence. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 49–56. IEEE, 2018.
- [14] E. N. Bruce. *Biomedical signal processing and signal modeling*. Wiley New York:, 2001.
- [15] S. J. Calil and M. S. T. Bottesi. Gerenciamento da manutenção de equipamentos hospitalares. In *Gerenciamento da manutenção de equipamentos hospitalares*, pages 108–108. 2002.
- [16] H. F. d. Castro et al. Otimização da confiabilidade e disponibilidade em sistemas redundantes. 2003.
- [17] P. C. Chaves and A. L. Moreira. Electrocardiografia. *Faculdade de Medicina da Universidade do Porto. Porto*, page 26, 2001.
- [18] A. Cuccuru, C. Mraidha, A. Radermacher, S. Gérard, L. Rioux, T. Vergnaud, and O. Hachet. Methodological guidelines on the usage of marte vsl for specification of time constraints. In *2nd Workshop on Model Based Engineering for Embedded Systems Design (M-BED 2011), Grenoble, France*, 2011.
- [19] V. da Silva Carvalho Vila and L. A. Rossi. O significado cultural do cuidado humanizado em unidade de terapia intensiva:—muito falado e pouco vivido. *Rev Latino-am Enfermagem*, 10(2):137–44, 2002.

- [20] D. el Diehn I. Abou-Tair, A. Khalifeh, S. Alouneh, and R. Obermaisser. Incremental, distributed, and concurrent service coordination for reliable and deterministic systems-of-systems. *IEEE Systems Journal*, pages 1–12, 2020.
- [21] D. Flood, F. McCaffery, V. Casey, R. McKeever, and P. Rust. A roadmap to iso 14971 implementation. *Journal of Software: Evolution and Process*, 27(5):319–336, 2015.
- [22] F. Fogliato and J. L. D. Ribeiro. *Confiabilidade e manutenção industrial*. Elsevier Brasil, 2009.
- [23] J. M. S. Fradinho, D. J. Nightingale, and M. T. W. Fradinho. A systems-of-systems perspective on healthcare: Insights from two multi-method exploratory cases of leading u.s. and u.k. hospitals. *IEEE Systems Journal*, 8(3):795–802, Sep. 2014.
- [24] M. A. Freitas and E. A. Colosimo. *Confiabilidade: análise de tempo de falha e testes de vida acelerados*. Escola de Engenharia da UFMG/Fundacao Christiano Ottoni, 1997.
- [25] B. Gallina, M. A. Javed, F. U. Muram, and S. Punnekkat. A model-driven dependability analysis method for component-based architectures. In *2012 38th Euromicro Conference on Software Engineering and Advanced Applications*, pages 233–240. IEEE, 2012.
- [26] B. Gallina, E. Sefer, and A. Refsdal. Towards safety risk assessment of socio-technical systems via failure logic analysis. In *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pages 287–292. IEEE, 2014.
- [27] S. Gao and J. Wang. Reliability and availability analysis of a retrial system with mixed standbys and an unreliable repair facility. *Reliability Engineering & System Safety*, 205:107240, 2021.
- [28] A. Gawanmeh, H. Al-Hamadi, M. Al-Qutayri, S.-K. Chin, and K. Saleem. Reliability analysis of healthcare information systems: State of the art and future directions. In *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, pages 68–74. IEEE, 2015.

- [29] V. Gehlot, L. Matthew, and E. B. Sloane. From data to processes—use of modeling and simulation in healthcare. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 320–323. IEEE, 2016.
- [30] B. Gonçalves, G. Guizzardi, and J. G. Pereira Filho. Using an eeg reference ontology for semantic interoperability of eeg data. *Journal of Biomedical Informatics*, 44(1):126–136, 2011.
- [31] G. Grasselli, M. Greco, A. Zanella, G. Albano, M. Antonelli, G. Bellani, E. Bonanomi, L. Cabrini, E. Carlesso, G. Castelli, et al. Risk factors associated with mortality among patients with covid-19 in intensive care units in lombardy, italy. *JAMA internal medicine*, 2020.
- [32] G. Grasselli, A. Zangrillo, A. Zanella, M. Antonelli, L. Cabrini, A. Castelli, D. Cereda, A. Coluccello, G. Foti, R. Fumagalli, G. Iotti, N. Latronico, L. Lorini, S. Merler, G. Natalini, A. Piatti, M. V. Ranieri, A. M. Scandroglio, E. Storti, M. Cecconi, A. Pesenti, and for the COVID-19 Lombardy ICU Network. Baseline Characteristics and Outcomes of 1591 Patients Infected With SARS-CoV-2 Admitted to ICUs of the Lombardy Region, Italy. *JAMA*, 323(16):1574–1581, 04 2020.
- [33] V. Grech. Unknown unknowns—covid-19 and potential global mortality. *Early Human Development*, page 105026, 2020.
- [34] J. R. Greenland, M. D. Michelow, L. Wang, and M. J. London. Covid-19 infection: implications for perioperative and critical care physicians. *Anesthesiology*, 132(6):1346–1361, 2020.
- [35] I. H. T. Guideline. Quality risk management. *Q9, Current step*, 4:408, 2005.
- [36] R. T. Hermeto. Análise de sobrevivência na modelagem do tempo de vida de redes de sensores sem fio. 2014.
- [37] J. Holt and S. Perry. *SysML for systems engineering*, volume 7. IET, 2008.
- [38] S. Ian. Engenharia de software. *6a. edição*, Addison-Wesley/Pearson, 2003.

- [39] E. Knobel, C. R. Laselva, and D. F. Moura Júnior. Terapia intensiva: enfermagem. In *Terapia intensiva: enfermagem*, pages 636–636. 2006.
- [40] M. Krasich. How to estimate and use mttf/mtbf would the real mtbf please stand up? In *2009 Annual Reliability and Maintainability Symposium*, pages 353–359, 2009.
- [41] P. Kumar and M. Jain. Reliability analysis of a multi-component machining system with service interruption, imperfect coverage, and reboot. *Reliability Engineering & System Safety*, 202:106991, 2020.
- [42] J. R. B. Lafraia. *Manual De Confiabilidade, Manutenibilidade: E DISPONIBILIDADE*. Qualitymark Editora Ltda, 2001.
- [43] L. M. Leemis. *Reliability: probabilistic models and statistical methods*. Prentice Hall New Jersey, 1995.
- [44] D. V. Lemes. *Proposta de método de análise de confiabilidade de sistemas eletrônicos empregando dados de retorno em garantia. 2006. 164f*. PhD thesis, Tese (Doutorado)—Dissertação (Mestrado em Engenharia)—Escola Politécnica da . . . , 2006.
- [45] S. Li, L. Da 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, 2013.
- [46] F. Lourenço. *Estudo de confiabilidade de baterias de chumbo-ácido e o impacto do tempo de pátio na sua confiabilidade*. PhD thesis, Universidade de São Paulo, 2010.
- [47] C. R. Madureira, K. Veiga, and A. F. M. Sant’ana. Gerenciamento de tecnologia em terapia intensiva. *Revista Latino-Americana de Enfermagem*, 8(6):68–75, 2000.
- [48] Marketing. Mtbf e mttr – saiba tudo sobre estes indicadores da indústria! Disponível em <https://sigga.com.br/blog/mtbf-e-mttr-indicadores-da-industria/>, 2019. Acesso em: 25 out. 2020.

- [49] A. Mashkooor and A. Egyed. Analysis of experiences with the engineering of a medical device using state-based formal methods. In *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pages 75–82. IEEE, 2018.
- [50] S. Mazzini, J. M. Favaro, S. Puri, and L. Baracchi. Chess: an open source methodology and toolset for the development of critical systems. In *EduSymp/OSS4MDE@MoDELS*, pages 59–66, 2016.
- [51] H. Mejía Salas and M. Mejía Suárez. Oximetría de pulso. *Revista de la Sociedad Boliviana de Pediatría*, 51(2):149–155, 2012.
- [52] L. Montecchi. Welcome to chess-sba wiki! Disponível em <<https://github.com/montex/CHESS-SBA/wiki>>, 2017. Acesso em: 25 out. 2020.
- [53] L. Montecchi and B. Gallina. Safeconcert: A metamodel for a concerted safety modeling of socio-technical systems. In M. Bozzano and Y. Papadopoulos, editors, *Model-Based Safety and Assessment*, pages 129–144, Cham, 2017. Springer International Publishing.
- [54] L. Montecchi, P. Lollini, and A. Bondavalli. A Reusable Modular Toolchain for Automated Dependability Evaluation. In *7th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2013)*, pages 298–303, Torino, Italy, 2013.
- [55] B. Paech and T. Wetter. Rational quality requirements for medical software. In *2008 ACM/IEEE 30th International Conference on Software Engineering*, pages 633–638. IEEE, 2008.
- [56] R. Palin, D. Ward, I. Habli, and R. Rivett. Iso 26262 safety cases: Compliance and assurance. *6th IET International Conference on System Safety 2011*, 2011.
- [57] W. d. P. Paula Filho. Engenharia de software: Fundamentos. *Métodos e Padrões-2a edição-LTC-2003*, 2003.
- [58] R. Pressman and B. Maxim. *Engenharia de Software-8ª Edição*. McGraw Hill Brasil, 2016.

- [59] E. Qaddoumi, M. Odeh, Z. Khan, M. Kossmann, A. Tbakhi, and A. Mansour. A semantically-enriched quality governance framework for systems of systems applied to cancer care. In *2018 1st International Conference on Cancer Care Informatics (CCI)*, pages 15–24. IEEE, 2018.
- [60] J. P. B. D. V. Raminhos. Aquisição de sinais fisiológicos. *Aplicação ao controlo de uma plataforma móvel a partir do EOG*, page 75, 2009.
- [61] R. M. Rangayyan. *Biomedical signal analysis*, volume 33. John Wiley & Sons, 2015.
- [62] C. Redmond. Trans-thoracic impedance measurements in patient monitoring. *EDN Network*, 2013.
- [63] G. d. S. R. Ribeiro, R. C. d. Silva, M. d. A. Ferreira, and G. R. d. Silva. Deslizes, lapsos e enganos no uso de equipamentos por enfermeiros na unidade de terapia intensiva. *Revista da Escola de Enfermagem da USP*, 50(3):419–426, 2016.
- [64] E. Y. Sakurada et al. As técnicas de análise dos modos de falhas e seus efeitos e análise da árvore de falhas no desenvolvimento e na avaliação de produtos. 2001.
- [65] P. Singh and L. Singh. Impact analysis of change in component reliabilities in safety-critical systems. *Quality and Reliability Engineering International*, 35(6):2051–2065, 2019.
- [66] A. Sobrinho, L. D. da Silva, A. Perkusich, P. Cunha, T. Cordeiro, and A. M. N. Lima. Formal modeling of biomedical signal acquisition systems: source of evidence for certification. *Software & Systems Modeling*, 18(2):1467–1485, Apr 2019.
- [67] A. Sobrinho, A. Perkusich, L. D. da Silva, T. Cordeiro, J. Rêgo, and P. Cunha. Towards medical device certification: a colored petri nets model of a surface electrocardiography device. In *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, pages 2645–2651. IEEE, 2014.
- [68] G. Stolfi. Confiabilidade. Disponível em <<http://www.lcs.poli.usp.br/~gstolfi/PPT/Confiabilidade.pdf>>, 2017. Acesso em: 25 out. 2020.

- [69] A. Subhan. Equipment maintenance, biomedical. *Encyclopedia of Medical Devices and Instrumentation*, pages 289–321, 2006.
- [70] H. M. Tahir, M. Nadeem, and N. A. Zafar. Specifying electronic health system with vienna development method specification language. In *2015 National Software Engineering Conference (NSEC)*, pages 61–66. IEEE, 2015.
- [71] D. K. Tempe, G. C. Khilnani, J. Passey, and B. Sherwal. Challenges in preparing and managing the critical care services for a large urban area during covid-19 outbreak: Perspective from delhi. *Journal of Cardiothoracic and Vascular Anesthesia*, 2020.
- [72] P. H. Tsarouhas. Reliability analysis into hospital dialysis system: A case study. *Quality and Reliability Engineering International*, 29(8):1235–1243, 2013.
- [73] L. S. van Galen, P. W. Struik, B. E. Driesen, H. Merten, J. Ludikhuizen, J. I. van der Spoel, M. H. Kramer, and P. W. Nanayakkara. Delayed recognition of deterioration of patients in general wards is mostly caused by human related monitoring failures: a root cause analysis of unplanned icu admissions. *PloS one*, 11(8):e0161393, 2016.
- [74] W. Wang, F. Guan, N. Hu, X. Jiang, and Z. Xu. Design multiparameter anaesthesia depth monitor. In *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4, 2009.
- [75] A. W. Wu, P. Pronovost, and L. Morlock. Icu incident reporting systems. *Journal of Critical Care*, 17(2):86 – 94, 2002.
- [76] Y. Zhang. Reliability analysis of randomly weighted k-out-of-n systems with heterogeneous components. *Reliability Engineering & System Safety*, 205:107184, 2021.

Apêndice A: Trabalhos Aceitos na Revisão Sistemática da Literatura

Titulo	Autores	Base Coletada
Analysis of Experiences with the Engineering of a Medical Device Using State-Based Formal Methods	A. Mashkoor; A. Egyed	IEEE Xplore
A Semantically-Enriched Quality Governance Framework for Systems of Systems applied to Cancer Care	E. Qaddoumi; M. Odeh; Z. Khan; M. Kossmann; A. Tbakhi; A. Mansour	IEEE Xplore
A Systems-of-Systems Perspective on Healthcare: Insights From Two Multi-Method Exploratory Cases of Leading U.S. and U.K. Hospitals	J. M. S. Fradinho; D. J. Nigh-tingale; M. T. W. Fradinho	IEEE Xplore
Formal modeling of biomedical signal acquisition systems: source of evidence for certification	Sobrinho, Alvaro; da Silva, Leandro Dias; Perkusich, Angelo; Cunha, Paulo; Cordeiro, Thiago; Lima, Antonio Marcus Nogueira	Springer
From Data to Processes-Use of Modeling and Simulation in Healthcare	Gehlot, Vijay and Mathew, Liji and Sloane, Elliot B.	Web of Science
Specification of e-Health system using Z: A motivation to formal methods	M. W. Azeem; M. Ahsan; N. M. Minhas; K. Noreen	IEEE Xplore
Specifying electronic health system with vienna development method specification language	H. M. Tahir; M. Nadeem; N. A. Zafar	IEEE Xplore
Towards medical device certification: A colored Petri Nets model of a surface electrocardiography device	A. Sobrinho; A. Perkusich; L. Dias da Silva; T. Cordeiro; J. RÃago; P. Cunha	IEEE Xplore