



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Verificação Autônoma de Contexto de Pacientes a Partir de Faixas de Risco na SA-BSN

Carlos Eduardo Taborda Lottermann

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientadora
Prof.a Dr.a Genaina Nunes Rodrigues

Brasília
2024



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Verificação Autônoma de Contexto de Pacientes a Partir de Faixas de Risco na SA-BSN

Carlos Eduardo Taborda Lottermann

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof.a Dr.a Genaina Nunes Rodrigues (Orientadora)
CIC/UnB

Prof. Dr. José Edil Guimarães de Medeiros
ENE/FT/UnB

MSc Ricardo Diniz Caldas
Chalmer University of Technology

Prof. Dr. João Luiz Azevedo de Carvalho
Coordenador do Curso de Engenharia da Computação

Brasília, 24 de Setembro de 2024

Dedicatória

Dedico este trabalho aos meus pais, João Carlos e Elizete, que sempre me apoiaram e acreditaram em mim e que também puderam me proporcionar a educação necessária para chegar até aqui.

Dedico também aos meus irmãos, Muriele e João Pedro, que serviram de inspiração para minha formação.

Por fim, dedico este trabalho para todas as pessoas que, de alguma forma, se envolveram com o projeto da SA-BSN.

Agradecimentos

Agradeço a todos os colegas do Laboratório de Engenharia de Software. Em especial a minha orientadora Prof.a Dr.a Genáina Rodrigues que confiou e acreditou em mim. Também aos colegas Caio Alencar, que muitas vezes conseguiu clarear melhor minhas ideias, Vicente De Moraes, que está comigo desde o meu início no laboratório e ao Ricardo Caldas, que foi um grande tutor durante esses anos.

A CJR, especialmente os colegas que foram diretores comigo, Arthur, Cássio, Daniel, Marcelo e Pedro, que viraram grandes amigos meus, transformaram minha vivência universitária e trouxeram muitos momentos felizes.

A minha fé.

E a todos que contribuíram no desenvolvimento da SA-BSN, sem essas pessoas esse trabalho não seria possível.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Com os avanços da tecnologia e a ubiquidade de sistemas modernos é esperado que estes consigam se adaptar de acordo com diferentes contextos em que eles operam. Nesse cenários, equipamentos, como dispositivos vestíveis, podem monitorar os sinais vitais de uma pessoa em diferentes estados de atividade. A *Self-Adaptive Body Sensor Network* (SA-BSN), em português, Rede de Sensores Corporais Auto-Adaptativa, é um protótipo de um Sistema Auto-Adaptativo, que opera de forma autônoma, identificando estados de emergência do paciente monitorado por esses sensores através de faixas de risco de cada um destes. Porém diferentes estados de atividade do paciente monitorado podem ser caracterizados por diferentes faixas de risco. Dessa forma, caso a SA-BSN não identifique esse contextos e altere as faixas de risco de acordo, falsos alertas de risco podem ser gerados.

A proposta desse trabalho é estender a SA-BSN para que ela consiga, de forma autônoma, verificar e adaptar-se para diferentes estados de atividade da pessoa assistida. Para isso, um modelo baseado em heurística foi implementado ao sistema permitindo que, em tempo real, ele identifique estados de atividade dinamicamente, a partir de faixas de risco pré-definidas, e faça a adaptação para esses estados corretamente. Dessa forma permitindo que a SA-BSN se torne mais adaptável e dinâmica.

Palavras-chave: Body Sensor Network, controlador, monitoramento de sinais vitais, contextos dinâmicos, sistemas auto-adaptativos

Abstract

With advances in technology and the ubiquity of modern systems, it is expected that they will be able to adapt according to different contexts in which they operate. In these scenarios, equipment such as wearable devices can monitor a person's vital signs in different states of activity. *Self-Adaptive Body Sensor Network* (SA-BSN), in Portuguese, Self-Adaptive Body Sensor Network, is a prototype of a Self-Adaptive System, which operates autonomously, identifying emergency states of the patient monitored by these sensors through ranges of risk of each of these. However, different activity states of the monitored patient can be characterized by different risk ranges. Therefore, if SA-BSN does not identify these contexts and does not detect a change in the risk state accordingly, false risk alerts may be triggered.

The purpose of this work is to extend SA-BSN so that it can, autonomously, check and adapt to different states of activity of the person being assisted. To achieve this, a model based on heuristics was implemented in the system, allowing it to dynamically identify activity states in real time, based on pre-defined risk ranges, and adapt to these states correctly. This allows SA-BSN to become more adaptable and aware of context changes of the assisted person.

Keywords: Body Sensor Network, controller, vital signs monitoring, dynamic contexts, self-adaptive systems

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Contribuição	3
1.3	Organização do trabalho	3
2	Referencial teórico e tecnologias adotadas	4
2.1	SA-BSN	4
2.1.1	Visão arquitetural	7
2.1.2	Cadeias de Markov	8
2.1.3	Configuração dos sensores	8
2.1.4	SA-BSN como um Sistema Ciberfísico	10
2.2	<i>Robot Operating System</i>	11
2.2.1	Tópicos	12
2.2.2	Serviços	12
2.2.3	Servidor de parâmetros	13
2.3	<i>Contextual Goal Model</i>	13
2.3.1	<i>Goal Model</i>	15
2.3.2	Análise de contexto	16
2.3.3	Exemplos	18
2.4	Sistemas Auto-adaptativos	18
3	Controlador de Contextos da SA-BSN	21
3.1	Definição de contextos	22
3.1.1	Contextos a partir das faixas de risco	22
3.1.2	Cuidados na definição de contextos	24
3.2	Inferência	25
3.2.1	Controlador	26
3.2.2	Visão processual	28
3.2.3	Análise de contexto	29

3.3	Alteração arquitetural	29
3.3.1	Alteração nos nós existentes	31
3.3.2	Remodelagem do <i>Goal Model</i>	31
3.3.3	Adição de novo nó	31
3.3.4	Requisitos funcionais e não funcionais	32
4	Validação	34
4.1	Configuração	34
4.1.1	Análise	36
4.2	Definição das faixas de risco	36
4.3	Resultados esperados	37
4.4	Experimento	38
5	Conclusão	41
5.1	Conclusões gerais	41
5.2	Limitações	41
5.3	Trabalhos futuros	42
5.3.1	Ampliação das informações de ambiente	42
5.3.2	Criação de faixas de risco	42
5.3.3	Implementação de diferentes algoritmos	42
	Referências	43
	Anexo	44
I	Parâmetros de inicialização	45

Lista de Figuras

2.1	<i>Goal Model</i> utilizado para representar a SA-BSN	5
2.2	Recorte do gráfico de nós e tópicos da SA-BSN. Detalhe para o tópico /TargetSystemData.	6
2.3	Perspectiva arquitetural da SA-BSN como um SAS [1]	7
2.4	Exemplo de um gráfico representando uma cadeia de Markov dos estados de uma máquina	9
2.5	Trecho da configuração parâmetros de frequência cardíaca da cadeia de Markov e das faixas de risco.	9
2.6	Comunicação entre <i>Master</i> , <i>Publisher</i> e <i>Subscriber</i> através de tópicos	12
2.7	Comunicação entre <i>Client</i> e <i>Server</i> através de um serviço	13
2.8	Como parâmetros podem ser definidos em um arquivo	13
2.9	Exemplo de um <i>Goal Model</i>	18
2.10	Diagrama conceitual de um sistema que utiliza MAPE-K	19
3.1	Exemplo de parâmetros para descrição de diferentes contextos como parâmetros da SA-BSN.	23
3.2	Exemplo de parâmetros para descrição de diferentes contextos como parâmetros da SA-BSN	24
3.3	Processo do modelo de inferência	28
3.4	Análise do contexto C1, nessa imagem contexto se refere ao estado de atividade	30
3.5	Novo <i>Goal Model</i> proposto para a SA-BSN	32
3.6	Recorte do gráfico de nós e tópicos após adição do nó / <i>context_adaptation</i>	33
4.1	Gráfico da frequência cardíaca ao longo do tempo, com marcação das faixas de risco.	35
4.2	Gráfico da oxigenação ao longo do tempo, com marcação das faixas de risco.	35
4.3	Valores de risco durante a execução da SA-BSN	36
4.4	Faixas de risco de BPM para contexto 1	37
4.5	Faixas de risco de BPM para contexto 2	38

4.6	Gráfico da frequência cardíaca ao longo do tempo com as faixas de baixo risco e filtro de passa baixas e contextos esperados	39
4.7	Gráfico do contexto durante a execução	39
4.8	Valores de risco durante a execução SA-BSN com o controlador ativado . .	40
I.1	Exemplo de parâmetros de inicialização do nó controlador. Parte 1/2 . . .	46
I.2	Exemplo de parâmetros de inicialização do nó controlador. Parte 2/2 . . .	47

Lista de Tabelas

3.1	Valores de baixo risco de frequência cardíaca para diferentes contextos . . .	23
4.1	Faixas de risco para contexto 0	37
4.2	Faixas de risco de BPM para contexto 1	37
4.3	Faixas de risco de BPM para contexto 2	37
4.4	Resultados e comparação das execuções	40

Lista de Abreviaturas e Siglas

APBD Pressão arterial diastólica.

APBS Pressão arterial sistólica.

API Interface de Programação de Aplicação.

BPM Batimentos por minuto.

CGM *Contextual Goal Model*.

DTMC Cadeia de Markov em Tempo Discreto.

MAPE-K Monitorar, Analisar, Planejar, Executar, Conhecimento.

POO Programação Orientada a Objetos.

ROS *Robot Operating System*.

SA-BSN *Self-Adaptive Body Sensor Network*.

SAS Sistemas Auto-Adaptativos.

WP *World Predicate*.

Capítulo 1

Introdução

O monitoramento de sinais vitais é uma prática muito comum na medicina moderna, principalmente dentro de hospitais, locais em que, normalmente, os pacientes são monitorados constantemente por equipamentos. Esses dispositivos permitem que médicos e enfermeiros consigam acompanhar de forma instantânea a progressão do estado do paciente; através de parâmetros como frequência cardíaca, pressão arterial, temperatura e saturação de oxigênio no sangue; permitindo que qualquer alteração significativa nestes possa ser tratada rapidamente.

Porém, com os avanços tecnológicos, o que antes era restrito aos hospitais acaba se tornando acessível para qualquer pessoa e a qualquer momento. Graças ao desenvolvimento de dispositivos vestíveis que possuem sensores biomédicos capazes de monitorar uma variedade de sinais vitais, assim como os equipamentos dos hospitais. Esses dispositivos podem ser relógios inteligentes, pulseiras e até anéis; formatos que permitem que o usuário os utilize constantemente em qualquer ocasião, inclusive fazendo exercícios ou dormindo, não só dentro de casa, mas em qualquer lugar.

A capacidade de monitorar sinais vitais de forma contínua e remota pode salvar vidas. Através do fornecimento de dados em tempo real sobre a saúde do usuário, esses dispositivos podem auxiliar na detecção de doenças, como a fibrilação atrial [2] ou acidentes físicos, como quedas [3]. Porém, caso não sejam bem implementados, podem ser problemáticos, como a geração de alertas falsos [4] ou a chamada de serviços de emergência sem necessidade.

Um dos principais desafios de engenharia na implementação desses dispositivos é a construção de sensores confiáveis e precisos, uma vez que eles devem ser capazes de operar em uma variedade de ambientes e situações diferentes e por múltiplos períodos de tempo. Por exemplo, uma pessoa pode utilizá-los ao correr ou nadar. Logo, algoritmos de processamento de sinais vitais presentes nesses dispositivos devem ser capazes de evitar ruídos e interferências e fornecer dados fidedignos.

Diferentemente dos hospitais, que possuem profissionais da saúde disponíveis constantemente para fazer a interpretação dos dados e averiguar melhor o estado do paciente, dispositivos vestíveis precisam operar de forma autônoma. Isso também representa um desafio significativo de implementação, pois a interpretação dos sinais varia de pessoa para pessoa e da atividade que ela está realizando. O que representa um estado de risco para um indivíduo pode não representar para outro. Por exemplo, uma pessoa idosa normalmente possui uma frequência cardíaca menor, enquanto uma pessoa jovem realizando atividades físicas terá uma frequência cardíaca maior.

Além disso, para o mesmo usuário, os sinais vitais podem variar bastante de acordo com a atividade que está realizando. Por exemplo, uma pessoa fazendo exercícios físicos tende a possuir um padrão vital diferente de quando está em repouso. Isso também se torna um desafio de implementação, pois, caso esses estados não sejam considerados, alertas falsos podem surgir.

"Os Sistemas Auto-Adaptativos (SAS) surgem como uma alternativa de implementação eficiente, visto que têm a capacidade de ajustar seu comportamento ou configurações em tempo real, com base nas observações do ambiente, do próprio sistema e de seus objetivos [5]. Uma das aplicações de SAS é em sistemas críticos, que são aqueles em que a falha pode resultar em perdas de vidas, danos significativos a propriedades e/ou ao meio ambiente [6]. Isso inclui sistemas de monitoramento médicos, já que seus erros ou falhas podem adiar o tratamento de eventuais complicações e, dependendo do estado do paciente, causar danos irreparáveis.

A *Self-Adaptive Body Sensor Network* (SA-BSN) [1] surge como artefato de um SAS desenvolvido pelo LADECIC (grupo de pesquisas registrado no CNPQ). Ela é um protótipo que observa os dados biomédicos de um indivíduo, aqui chamado de paciente, e, em tempo real, identifica se ele está em risco ou não.

1.1 Objetivos

Ao iniciar, a SA-BSN carrega diversos parâmetros, incluindo as faixas de risco dos sinais vitais medidos pelos sensores disponíveis. Essas faixas de risco são categorizadas em níveis baixo, médio e alto, correspondendo a diferentes valores dos sinais vitais. Cada faixa é utilizada para calcular uma porcentagem de risco para cada dado coletado por sensor. Esses dados são então analisados em conjunto para determinar o nível de risco geral do indivíduo monitorado, indicando se ele está em situação de emergência ou não.

No entanto, essas faixas de risco permanecem estáticas durante toda a execução. Isso significa que, se o estado de atividade do indivíduo mudar, como ao iniciar a prática de exercícios, seus padrões vitais se comportarão de maneira diferente. Por exemplo, a

frequência cardíaca aumentará sem necessariamente indicar um alto risco. Dessa forma, as leituras dos sinais vitais nesse momento podem sugerir níveis de risco mais elevados que não refletem o risco real do indivíduo monitorado.

O objetivo deste trabalho é permitir que a SA-BSN realize uma verificação autônoma dos estados de atividade do indivíduo monitorado e adapte dinamicamente os parâmetros das faixas de risco para o estado correspondente. Isso evitará que alertas de alto risco sejam gerados de forma errônea.

Para isso, é necessário:

1. Entender como os dados biomédicos utilizados pela SA-BSN são gerados e/ou coletados e configurados.
2. Definir e caracterizar os estados de atividade do paciente compatíveis com a arquitetura da SA-BSN e definir uma estratégia de inferência sobre estados.
3. Estender a SA-BSN com essas novas funcionalidades conforme os estados identificados do paciente em seus diferentes contextos de atividade.

1.2 Contribuição

Este trabalho contribui com a inclusão de novas funcionalidades à SA-BSN para dinamicamente distinguir estados de atividade (contextos) do paciente, e, conseqüentemente, identificar o estado de emergência conforme cada contexto. Através dessa funcionalidade, é possível que a SA-BSN aplique as políticas de adaptação conforme os diferentes contextos identificados do paciente. Para tanto, foi necessária a definição de estados de atividade de acordo com as faixas de risco dos sinais vitais, além da implementação de um controlador baseado em heurística para realizar a adaptação da SA-BSN. Os resultados mostram que o controlador foi capaz de realizar a adaptação no contexto executado no teste, em que sinais vitais reais de um paciente, em diversos estados de atividade foram utilizados.

1.3 Organização do trabalho

Os próximos capítulos deste trabalho foram organizados da seguinte forma. O Capítulo 2 apresenta um referencial teórico sobre os conceitos importantes e que foram utilizados neste trabalho, bem como apresenta as tecnologias utilizadas. O Capítulo 3 apresenta a implementação e o desenvolvimento da solução para o problema apresentado, detalhando as escolhas feitas e a modelagem do processo de inferência utilizado. O Capítulo 4 apresenta como essa implementação foi validada. Por fim, o Capítulo 5 apresenta as conclusões atingidas sobre este trabalho.

Capítulo 2

Referencial teórico e tecnologias adotadas

Neste capítulo a SA-BSN é apresentada tecnicamente, bem como tecnologias utilizadas em sua implementação, e é apresentado, também, aspectos fundamentais sobre o referencial teórico utilizado nesse trabalho.

2.1 SA-BSN

A *Self-Adaptive Body Sensor Network* (SA-BSN) é uma rede de sensores corporais utilizada para o monitoramento médico [1]. Seu desenvolvimento se iniciou por participantes do Laboratório de Engenharia de Software (LES-UnB) e assim é mantido até hoje, sendo utilizada em diversos trabalhos relacionados. Sua criação teve como objetivo a criação de um SAS palco de teste com especificação em CGM [7], combinando teoria de controle com inteligência artificial [8].

A SA-BSN é um artefato de uma aplicação biomédica implementado em ROS [9], cujo objetivo é identificar emergências através do monitoramento contínuo de sinais vitais de um paciente. Ela consegue se adaptar aos níveis de atributos de qualidade desejados com mínima intervenção humana.

A Figura 2.1 apresenta o *Goal Model* utilizado na arquitetura da SA-BSN.

Dentro do atuador BSN existem 4 objetivos:

- *G1: Emergency is detected*: Objetivo principal do sistema, detectar emergências, é realizado pela conclusão de *G2*.
- *G2: Patient status is monitored*: Monitoramento do status do paciente, é realizado pela conclusão de *G3* e *G4*.

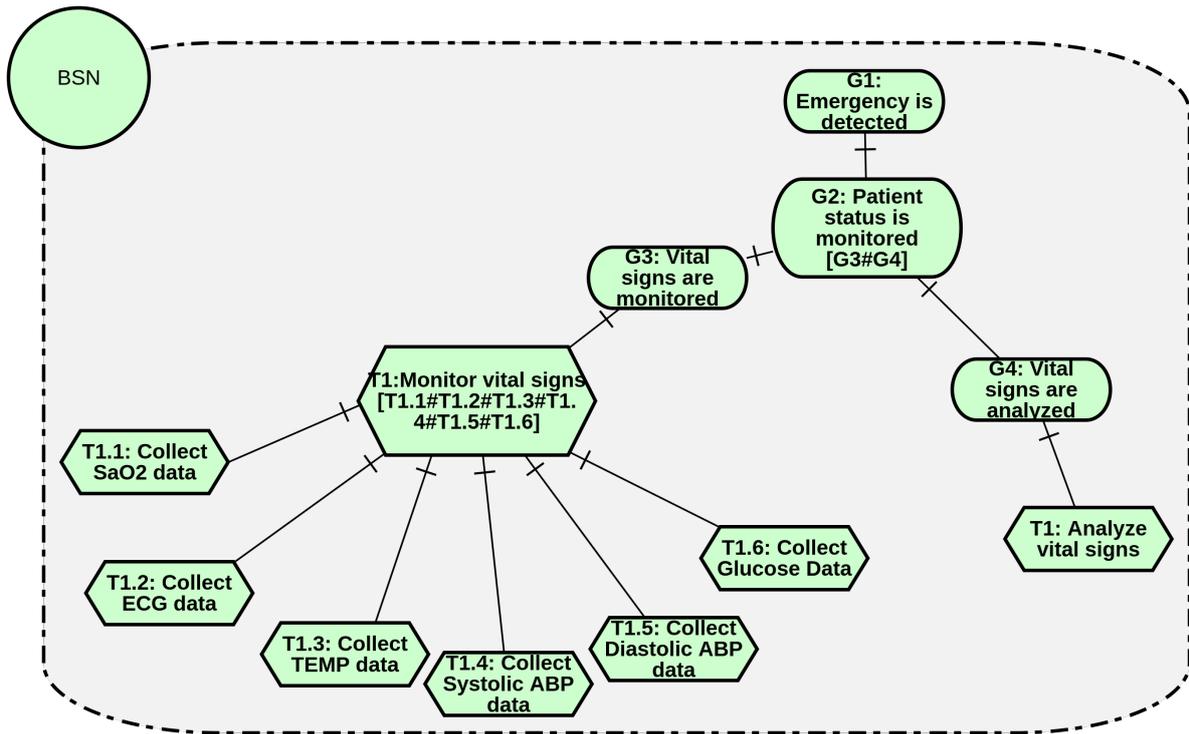


Figura 2.1: *Goal Model* utilizado para representar a SA-BSN

- *G3: Vital signs are monitored*: Sinais vitais são monitorados, é realizado pela conclusão da tarefa *G3T1*.
- *G4: Vital signs are analyzed*: Sinais vitais são analisados, é realizado pela conclusão da tarefa *G4T1*.

Que são alcançados pela realização das tarefas:

- *G3T1: Monitor vital signs*: Sensores responsáveis por coletar os sinais vitais do paciente.
- *G4T1: Analyze vital signs*: Corresponde a *Central Hub* que recebe os dados de todos os sensores e identifica o status do paciente.

A tarefa *G3T1* é dividida em seis sub-tarefas correspondentes a cada um dos sensores disponíveis na SA-BSN, sendo eles de: temperatura, oxigenação, glicose, frequência cardíaca, pressão arterial sistólica e diastólica. Cada tarefa e sub-tarefa é realizada por um nó em ROS.

A Figura 2.2 mostra o gráfico de nós e tópicos. Cada elipse representa um nó da SA-BSN e cada seta indica o tópico correspondente. A origem da seta é o nó publicador e o destino é o nó inscrito.

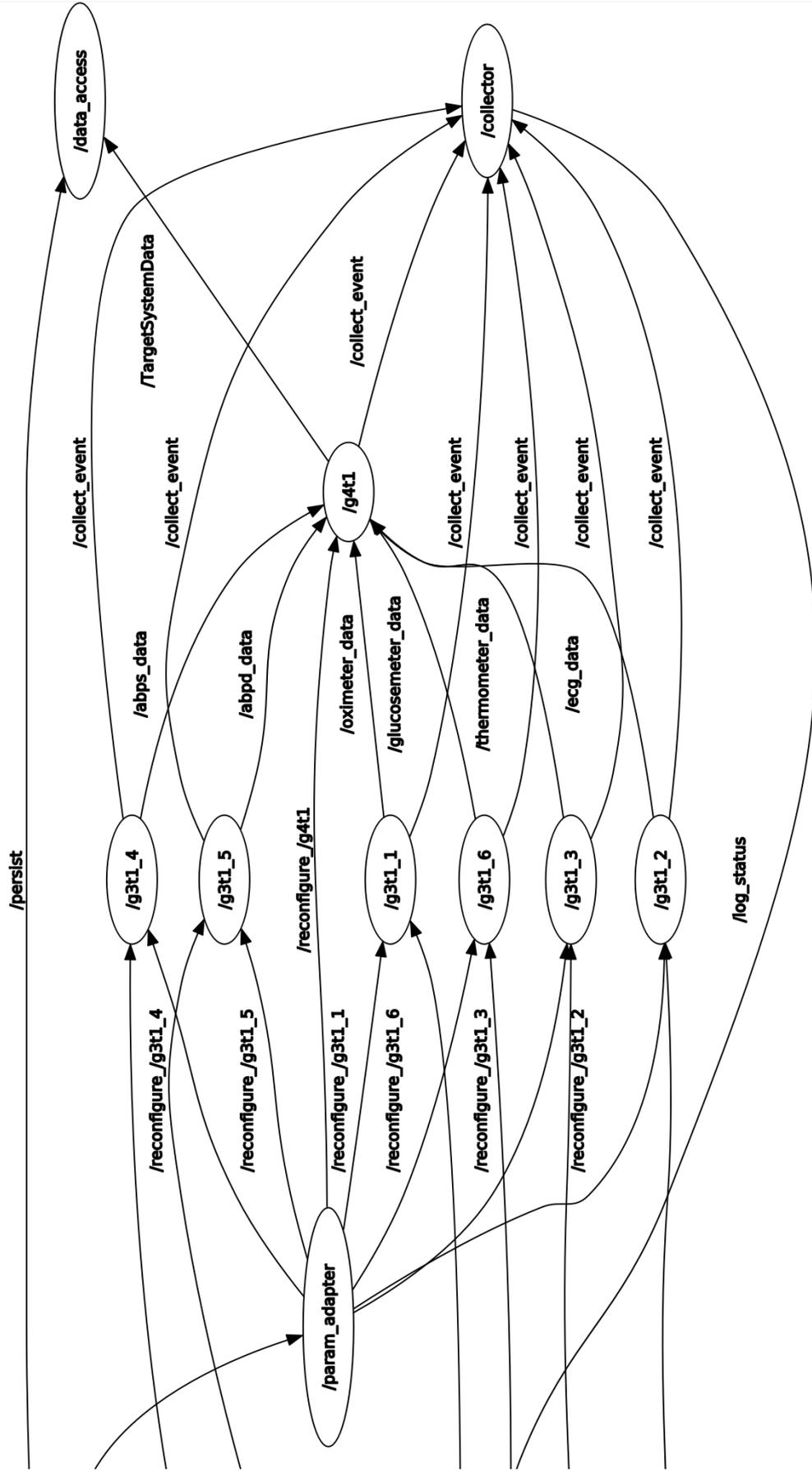


Figura 2.2: Recorte do gráfico de nós e tópicos da SA-BSN. Detalhe para o tópico `/TargetSystemData`.

2.1.1 Visão arquitetural

A arquitetura da SA-BSN segue os princípios de auto-adaptação definidos por Weyns [5]. Ela é composta pelo Sistema Gerenciador (*Managing System*), o Sistema Gerenciado (*Managed System*), o Repositório de Conhecimento (*Knowledge Repository*) e Simulação (*Simulation*). Sua perspectiva de interação entre os sistemas pode ser vista na Figura 2.3.

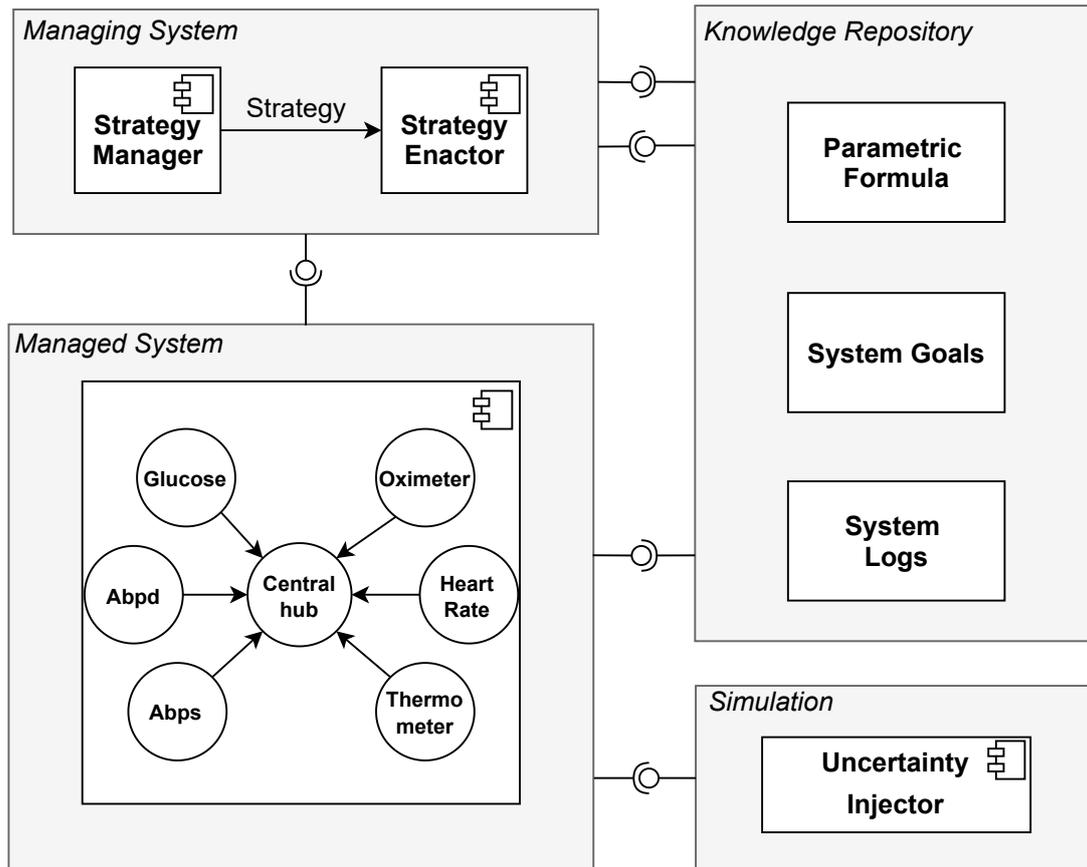


Figura 2.3: Perspectiva arquitetural da SA-BSN como um SAS [1]

1) *Sistema Gerenciador*: Esse módulo compreende o Gerenciador da Estratégia (*Strategy Manager*) e o Executor da Estratégia (*Strategy Enactor*). Nele está implementado o controlador. O Gerenciador de Estratégia estima a confiabilidade do sistema e o custo dado a Fórmula Paramétrica contida no Repositório do Conhecimento. O Executor da Estratégia, onde de fato o controlador está implementado, atua na aplicação dos parâmetros.

2) *Sistema Gerenciado*: Módulo que compreende o sistema que está sendo monitorado. É composto pelos seis sensores, que aferem o sinal vital designado, e pela *Central Hub*, que recebe esses sinais aferidos e indica se o paciente está em estado de risco ou não.

3) *Repositório do Conhecimento*: Possui a fórmula paramétrica, que calcula a confiabilidade e os objetivos do sistema no formato de *Goal Model* (veja Figura 2.1), bem como os *Logs* do sistema.

4) *Simulação*: Composto pelo Injetor de Incertezas (*Uncertainty Injector*), ele é responsável por simular ruídos e provocar falhas no sistema que, caso contrário, não falharia, assim provocando a adaptação.

2.1.2 Cadeias de Markov

A SA-BSN possui uma série de configurações em seu ambiente de execução, uma parte dessas configurações são para os sensores e o gerador de dados, essas configurações correspondem ao perfil do paciente. O perfil é moldado por Cadeias de Markov em Tempo Discreto (DTMC), sendo uma para cada sensor.

A DTMC se trata de modelos matemáticos que descrevem a progressão temporal de um sistema, esse sistema transita por cada estado em que a probabilidade de transição para o próximo estado depende exclusivamente do estado atual, em outras palavras, esse sistema tem uma "memória curta", visto que fatores anteriores ao estado atual são irrelevantes. A escolha por DTMC para representar o ambiente é interessante pois, ao mesmo tempo que o comportamento é aleatório, representando muito bem a variabilidade do ambiente e que o sistema não tem controle sobre este, as probabilidades são definidas e o sistema não se torna complexo com o passar do tempo, facilitando a implementação e análise do sistema.

A Figura 2.4 representa uma cadeia de Markov de três estados de uma máquina, imaginando um modelo onde ela está quebrada, ou em manutenção ou operacional. As probabilidades de transição estão nas setas. Estando no estado operacional, a máquina tem a probabilidade de 70% de ficar no mesmo estado, 20% de ir para o estado Quebrada e 10% de ir para o estado Em manutenção. Da mesma forma, os outros estados têm um conjunto de probabilidades. Nem sempre o estado terá uma probabilidade de transição para todos os estados, por exemplo, o estado Quebrada não saltará para operacional e o estado Em manutenção não saltará para Quebrada.

2.1.3 Configuração dos sensores

No projeto da SA-BSN, cada estado da cadeia significa um nível de risco do paciente e é utilizado para o cálculo da previsão do risco. São cinco estados, sendo dois de alto risco, dois de médio risco e um de baixo risco. Cada estado corresponde a uma faixa de valores de sinais biomédicos, indicando e correspondendo ao risco associado. Esses valores são

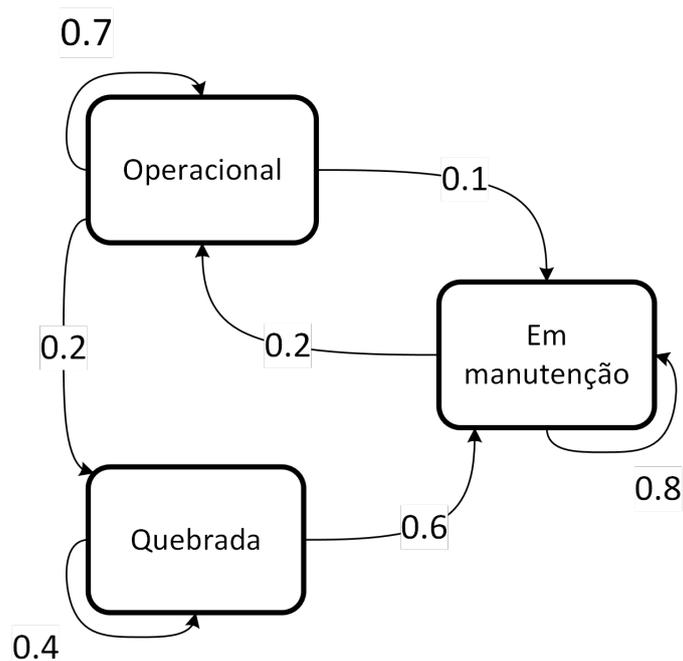


Figura 2.4: Exemplo de um gráfico representando uma cadeia de Markov dos estados de uma máquina

carregados ao servidor de parâmetros na inicialização do sistema e são acessados tanto pelo gerador de valores quanto pelos sensores para o cálculo do risco.

```

<!-- Markov chain for heart frequency -->
<param name="heart_rate_State0" value="72,21,4,2,1" />
<param name="heart_rate_State1" value="14,61,19,4,2" />
<param name="heart_rate_State2" value="1,17,60,20,2" />
<param name="heart_rate_State3" value="0,2,15,70,13" />
<param name="heart_rate_State4" value="0,1,2,20,77" />

<!-- Risk values for heart frequency -->
<param name="heart_rate_HighRisk0" value="0,70" />
<param name="heart_rate_MidRisk0" value="70,85" />
<param name="heart_rate_LowRisk" value="85,97" />
<param name="heart_rate_MidRisk1" value="97,115" />
<param name="heart_rate_HighRisk1" value="115,300" />
  
```

Figura 2.5: Trecho da configuração parâmetros de frequência cardíaca da cadeia de Markov e das faixas de risco.

Na Figura 2.5 os valores dos parâmetros que são carregados estão representados. No primeiro conjunto, está definida a matriz de transição da cadeia de Markov, sendo que o primeiro valor, de cada linha, corresponde à probabilidade de saltar para o estado zero, o segundo valor para o estado 1 e assim por diante, caracterizando as probabilidades em cada estado.

Já no segundo conjunto, estão definidos os valores de risco, nesse caso para a frequência cardíaca. O primeiro estado é o de alto risco 0 e a faixa de valores corresponde de 0 até 70 BPM, o segundo é de médio risco 0, que corresponde de 70 até 85 BPM e assim por diante.

2.1.3.1 Gerador de dados

O gerador de dados está dentro do nó do paciente, ele possui tanto as cadeias de Markov quanto as faixas de risco em memória. O objetivo principal do nó é simular um paciente, provendo os sinais vitais virtuais necessários para os sensores. A comunicação entre os sensores e este nó é feita através de serviços em que o sensor requisita um sinal biomédico e o nó responde com o dado gerado para este sensor. Ao gerar um valor, ele estará dentro da faixa correspondente ao estado atual.

2.1.3.2 Detecção de risco

Cada nó do sensor obtém, do servidor de parâmetros, as faixas correspondentes ao sinal vital designado, essa obtenção é feita quando o nó se inicia. Ao coletar um dado, ele calcula um valor de risco, entre 0 e 100 em porcentagem, atribuído a essa coleta. O risco é calculado por um deslocamento dentro das faixas, assumindo uma lógica que depende do sensor. Para o sensor de oxigenação, o valor mais alto, ou seja, o limite superior da faixa de baixo risco, corresponde ao menor risco calculado. Já para os sensores de pressão (APBS e APBD), o menor valor, ou seja, o limite inferior da faixa de baixo risco, corresponde ao menor risco calculado. Para os outros sensores, o menor risco é o valor que está no centro da faixa de baixo risco.

Depois, esses dados são enviados por um tópico para a *Central Hub*, tarefa *G4T1*. Nesse nó, os dados de risco de todos os sensores são agregados em um risco geral do paciente. Valores de risco alto são considerados um estado de emergência, assim concluindo o objetivo *G1* da *Goal Model* (veja Figura 2.1).

2.1.4 SA-BSN como um Sistema Ciberfísico

Sistemas Ciberfísicos são sistemas compostos por entidades computacionais que têm o objetivo de controlar entidades físicas. A SA-BSN se caracteriza como esse tipo de sistema, já que possui componentes de natureza física (sensores) e componentes de natureza computacional que controlam os componentes físicos (controlador).

Inicialmente, a *Self-Adaptive Body Sensor Network* (SA-BSN) foi implementada utilizando somente sensores virtuais, com seus dados providos através de seu gerador de dados. Porém, em atualizações mais recentes, sensores físicos foram implementados através de

Arduinos, e os dados aferidos foram inseridos na SA-BSN no lugar dos dados virtuais, dando assim um aspecto mais físico ao sistema.

2.2 *Robot Operating System*

A SA-BSN é implementada utilizando o conjunto de *frameworks* de software chamado *Robot Operating System* (ROS). Trata-se de um *framework* de código aberto, cujo desenvolvimento começou em 2007, com a primeira versão lançada em 2010, com o objetivo de ser um "denominador comum" no desenvolvimento de sistemas robóticos, facilitando assim a implementação de diversas funcionalidades independentemente da área de atuação do desenvolvedor [10].

Por se tratar de um *framework* de código aberto, formou-se uma comunidade, e atualmente o ROS é largamente utilizado em aplicações industriais, de pesquisa e acadêmicas. Suas diversas implementações e contribuições recebidas durante os anos o consolidaram como estado da arte para implementações robóticas [11].

Em nível de computação, as entidades padrão são: nós, *master*, servidor de parâmetros, mensagens, tópicos e serviços.

- Nós: Nós em ROS são processos que realizam computação. Eles se comunicam entre si através de tópicos, serviços ou parâmetros. Eles devem operar em uma escala refinada; um sistema robótico normalmente contém vários nós. Podem ser implementados tanto em *C++* quanto em *Python*.
- *Master*: O ROS *master* fornece, para os demais nós, os processos de registro e pesquisa de nomes. Todo sistema ROS ao ser executado necessita de uma instância do ROS *master* e deve compartilhar essa instância, mesmo se estiver sendo executado em máquinas diferentes.
- Servidor de parâmetros: Faz parte do ROS *master*. O servidor de parâmetros permite que todos os nós acessem e modifiquem dados em um repositório comum.
- Mensagens: Nós se comunicam entre si através de mensagens, que são estruturas de dados simples contendo dados fortemente tipados.
- Tópicos: Mensagens são transportadas através de um tópico por meio de um sistema de publicar/inscrever, em que um nó publica uma mensagem em um tópico com determinado nome e outros nós se inscrevem nesse tópico para receber as mensagens.
- Serviços: Embora flexível, nem sempre a arquitetura de tópicos atende às necessidades de um sistema. Serviços surgem como uma alternativa, fornecendo interações

cliente/servidor, onde determinado nó faz uma requisição de dados e outro nó responde.

2.2.1 Tópicos

Tópicos são uma das maneiras de nós trocarmos informações entre si. Eles são barramentos nomeados por onde as mensagens percorrem e, no geral, os nós não sabem com quem estão comunicando, estando apenas "interessados" nos dados transmitidos. É uma comunicação $n:n$, pois podem existir n nós publicadores e n nós inscritos.

Considerando que dois nós desejam comunicar entre si, o nó publicador anuncia para o *master* que deseja publicar em determinado tópico. Já o nó inscrito anuncia para o *master* que deseja se inscrever para receber as mensagens deste tópico. A partir desse momento, cada mensagem publicada será recebida pelo nó inscrito. Tópicos podem ter mais de um publicador e inscrito (veja Figura 2.6).

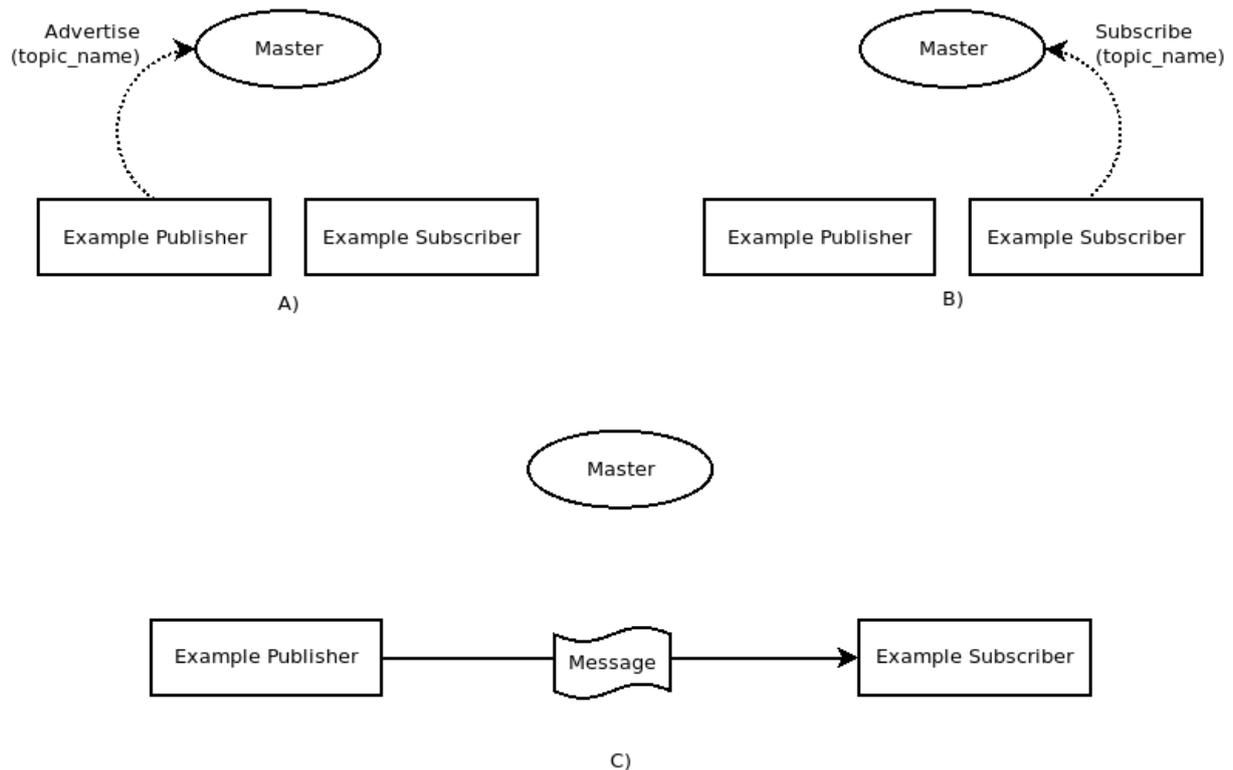


Figura 2.6: Comunicação entre *Master*, *Publisher* e *Subscriber* através de tópicos

2.2.2 Serviços

Ao contrário dos tópicos, que possuem uma comunicação unidirecional, onde o publicador envia a mensagem e o inscrito a recebe a qualquer momento, os serviços partem de um

modelo cliente/servidor. Nesse modelo, existem dois tipos de nó e a comunicação é bidirecional. A qualquer momento, o nó cliente do serviço pode enviar uma requisição, que pode conter dados, para o servidor. Este recebe a requisição, analisa e elabora uma resposta que será enviada para o cliente. Ou seja, ao contrário dos tópicos, em que as mensagens são publicadas a qualquer momento, nos serviços o servidor só poderá enviar uma mensagem ao cliente quando ele requisitar. Além disso, há uma restrição: só pode existir um nó servidor, porém pode haver mais de um cliente, caracterizando uma comunicação $1:n$.

A Figura 2.7 representa uma comunicação através de serviços, onde *A*) é o caminho percorrido pela mensagem de requisição e *B*) é o caminho percorrido pela mensagem de resposta

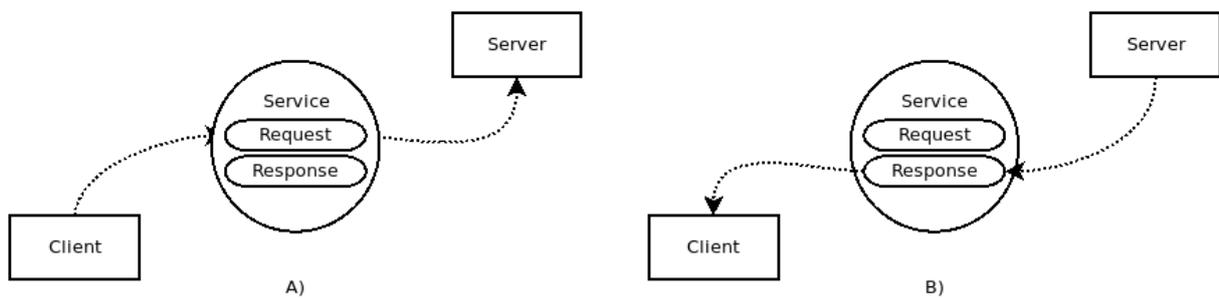


Figura 2.7: Comunicação entre *Client* e *Server* através de um serviço

2.2.3 Servidor de parâmetros

O servidor de parâmetros armazena informações em um dicionário compartilhado e multi-variado acessível por uma API. Normalmente, esses parâmetros são configurações dos nós que são definidas em sua inicialização. Eles podem ser acessados e modificados por vários nós, armazenados em arquivos e carregados ao servidor (veja Figura 2.8). Eles podem representar alguns tipos de dados, sendo: inteiros de 32 bits, *booleanos*, *strings*, *doubles*, *ISO8601 dates*, listas e dados binários codificados em Base64. Além disso, o desenvolvedor pode definir o escopo de cada parâmetro, ou seja, quais nós podem acessá-los.

```
<!-- Exemplo de um parametro definido em um arquivo -->
<param name="publish_frequency" type="double" value="10.0" />
```

Figura 2.8: Como parâmetros podem ser definidos em um arquivo

2.3 Contextual Goal Model

O cenário de engenharia de software evolui muito rapidamente. Nesse sentido, a engenharia de requisitos representa um papel importante na definição das necessidades e

condições que determinado sistema deve atingir. À medida que paradigmas de computação se tornam mais difundidos e universais, a engenharia de requisitos tradicional enfrenta dificuldades em lidar com a variedade de contextos em que um sistema moderno pode operar. Uma das principais dificuldades da engenharia de requisitos tradicional vem da definição de requisitos estáticos ao longo do tempo, assumindo apenas um retrato do cenário do ambiente. Se assim for feito, o processo de definição de objetivos pode falhar em capturar nuances provindas dos diferentes contextos em que o sistema pode operar, resultando na incapacidade do sistema de se adaptar a novas condições operacionais, como é frequentemente visto em cenários de computação móvel e ubíqua.

Isso leva à criação de sistemas mais rígidos, dificultando sua adaptação a mudanças contextuais ou novas necessidades dos interessados. Essa rigidez na modelagem de requisitos resulta em sistemas que, embora funcionais em condições previsíveis, falham em oferecer uma resposta adequada em situações inesperadas ou não previstas. Além disso, os engenheiros de requisitos podem enfrentar dificuldades na incorporação de novos requisitos ou na modificação dos existentes, o que pode ser crítico em sistemas que demandam manutenção contínua.

O contexto, definido como o conjunto de condições que envolvem a operação de um sistema, pode impactar tanto os objetivos dos interessados quanto a maneira como esses são alcançados. Essas condições contextuais podem incluir variáveis como localização geográfica, horário, preferências do usuário, condições ambientais e até mesmo o estado atual de outros sistemas interconectados. Por exemplo, em um sistema de informação para museus, o contexto pode determinar se um visitante está próximo a uma obra de arte específica, se ele possui preferências alimentares que influenciam as recomendações de restaurantes, ou se o museu está prestes a fechar [12]. Cada uma dessas variáveis contextuais pode alterar os objetivos prioritários do sistema e a forma como as funcionalidades são oferecidas ao usuário.

A utilização de *Contextual Goal Model* (CGM) permite incorporar esses fatores contextuais diretamente na modelagem do sistema, possibilitando que ele adapte seu comportamento em tempo de execução com base nas informações contextuais capturadas. Isso significa que, ao invés de operar com um conjunto fixo de regras ou objetivos, o sistema pode reconfigurar suas prioridades e processos conforme o contexto evolui. Essa adaptabilidade é crucial em ambientes dinâmicos, como sistemas de saúde, onde o estado de um paciente pode mudar rapidamente, exigindo que o sistema de monitoramento ajuste seus alertas e notificações em tempo real.

Ao integrar o contexto em diversos pontos do *Goal Model*, o CGM assegura que os objetivos permanecem alinhados com os interessados, mesmo que as condições do ambiente mudem. Isso é particularmente importante em sistemas complexos e interativos, onde a

falta de adaptação ao contexto pode resultar em experiências de usuário insatisfatórias ou até em falhas operacionais. Contudo, a implementação do CGM também apresenta desafios, como a necessidade de um monitoramento contínuo do contexto e a complexidade de decidir quais informações contextuais são relevantes em cada momento. Além disso, há o desafio de lidar com informações que não podem ser diretamente aferidas ou que são de natureza abstrata, como as emoções e intenções dos usuários. Nesses casos, o sistema pode precisar fazer inferências com base em dados disponíveis.

Ainda assim, a capacidade de adaptação em tempo de execução oferecida pelo CGM representa uma evolução significativa em relação à engenharia de requisitos tradicional. Com o CGM, os sistemas podem evoluir em sincronia com o ambiente, garantindo que os objetivos dos interessados sejam atendidos de forma mais eficaz e precisa, além de tornar o sistema mais autônomo, sabendo lidar com as alterações contextuais sem a necessidade de interação humana.

2.3.1 *Goal Model*

Após o entendimento dos desafios da engenharia de requisitos tradicional e a necessidade de incorporar fatores contextuais, a estruturação de um *Goal Model* surge como uma metodologia essencial para garantir que sistemas sejam projetados com flexibilidade e adaptabilidade em mente. A construção de um *Goal Model* é feita de forma que os requisitos do sistema sejam orientados a objetivos. Ele permite elencá-los de uma maneira hierárquica e gráfica, ficando claro quais são os objetivos, como eles interagem entre si, além das tarefas a serem feitas para concluí-los de acordo com os *stakeholders* e as condições contextuais que influenciam a operação do sistema.

2.3.1.1 Estrutura de um *Goal Model*

O *Goal Model* é uma modelagem e possui alguns itens que o caracterizam. A seguinte lista elenca os mais essenciais:

- **Objetivos:** Os objetivos são as intenções ou estados desejados que o sistema deve alcançar. Eles podem variar de intenções gerais e subjetivas, como "Melhorar a experiência do visitante," a metas específicas e claramente alcançáveis, como "Notificar o visitante sobre eventos futuros." Cada objetivo pode ser desdobrado por um conjunto de sub-objetivos ou tarefas que especifiquem o procedimento para o objetivo ser alcançado.
- **Tarefas:** Tarefas são ações concretas que o sistema deve realizar para atingir um objetivo. Podem estar associadas a outras tarefas dentro do mesmo objetivo, sendo

que a realização dessa tarefa, ou do conjunto de tarefas, indica que o objetivo foi alcançado.

- **Atuadores:** São entidades responsáveis pela realização das tarefas, como por exemplo um gerente, um sistema ou um programa. Atuadores operam de forma independente, ou seja, um atuador A realiza suas tarefas de forma independente do atuador B. Porém, podem haver interações entre atuadores. Por exemplo, um braço robótico fixo (atuador A) pode depender de um robô móvel (atuador B) para trazer algum objeto necessário para a realização de alguma tarefa ou objetivo específico.
- **Contextos:** Desempenham um papel fundamental na *Contextual Goal Model*, pois determinam as condições em que determinados objetivos devem ser realizados. Por exemplo, se o museu estiver próximo do horário de fechar, o objetivo "visitantes estarão fora do museu no horário de fechamento" se torna relevante, e tarefas como "alertar visitante sobre horário de fechamento" ou "conduzir visitante até a saída" se tornam necessárias e são realizadas. São comumente definidos por variáveis como localização, horário e estado do sistema.

A interação entre componentes pode ser feita por algumas ligações *links* diferentes. Dentre eles:

- **Links de decomposição ou refinamento:** São utilizados entre objetivos e tarefas ou outros objetivos. Podem ser do tipo:
 - AND: Quando, para a realização do objetivo, é necessária a realização de todas as tarefas ou sub-objetivos associados..
 - OR: Quando é necessária a realização de apenas uma das tarefas ou sub-objetivos.
- **Links de dependência:** São utilizados quando há necessidade de algum recurso de outro atuador.

2.3.2 Análise de contexto

Um passo importante na elaboração de requisitos utilizando CGM é a análise de contexto. Porém, diferentemente da análise de objetivos, que segue uma forma sistemática para descobrir o conjunto de objetivos e tarefas dentro de um atuador, a análise de contexto também segue uma forma sistemática, mas desta vez para descobrir conjuntos alternativos de fatos que um atuador pode verificar para julgar se um contexto se aplica.

Na análise de contexto, *World Predicate* (WP) é definido como um estado parcial do mundo e pode ser classificado em dois tipos:

- Fatos: O WP é um fato para determinado atuador quando ele pode ser verificado diretamente por este atuador.
- Declarações: O WP é uma declaração para determinado atuador quando ele não pode ser verificado por este atuador.

Como a avaliação do *World Predicate* depende das informações de mundo que o atuador tem acesso, o que é fato para um atuador pode ser uma declaração para outro, visto que diferentes atuadores tem diferentes visões de mundo. Porém o fato não é subjetivo, ou seja, se um fato for verdadeiro para um atuador ele será verdadeiro para todos. Por exemplo, "a temperatura na sala é maior que 25°C", é um fato que o atuador pode obter verificando um sensor ou consultando uma base de dados que possui essa informação.

Já as declarações ocorrem quando não podem ser verificados e isso pode ocorrer por dois motivos:

- Falta de informação: ocorre quando o atuador não consegue verificar diretamente por não ter acesso a informação de contexto. No exemplo anterior "a temperatura na sala é maior que 25°C" pode ser uma declaração se o sistema não tiver acesso a um termômetro.
- Natureza abstrata: ocorre quando não existe um critério claro a ser avaliado. Por exemplo, "A sala está fria", por mais que a temperatura esteja relacionada o que é frio ou não parte do julgamento de cada pessoa e a resposta para esse não é clara.

Dependendo da análise de contexto, o atuador não terá acesso a todas informações necessárias, assim alguns WP dessa análise podem ser declarações. Porém caso essas declarações consigam ser avaliadas de outra forma, por exemplo, a declaração anterior "A sala está fria" pode ser apoiada por outros WP, como por exemplo, "Mais da metade das pessoas da sala estão usando casaco", que pode ser um fato e contribui para a afirmação da declaração. Então é definido que uma declaração é apoiada por um conjunto de WP caso esse conjunto dê evidências necessárias para a afirmação dela. Caso a declaração tenha esse conjunto ela é chamada de declaração monitorável. Por fim, o contexto é considerado monitorável caso seja composto por um conjunto de fatos e declarações monitoráveis.

A análise de contexto é feita com o objetivo de obter um contexto monitorável e o conjunto de WP que apoiam ele. Caso não seja, ele não deve ser utilizado na modelagem de objetivos, visto que o atuador nunca terá "certeza" o suficiente para dizer se esse contexto se aplica.

2.3.3 Exemplos

A Figura 2.9 apresenta um exemplo de um *Goal Model* representando sistema que tem como objetivo informar sobre uma peça de arte em um museu.

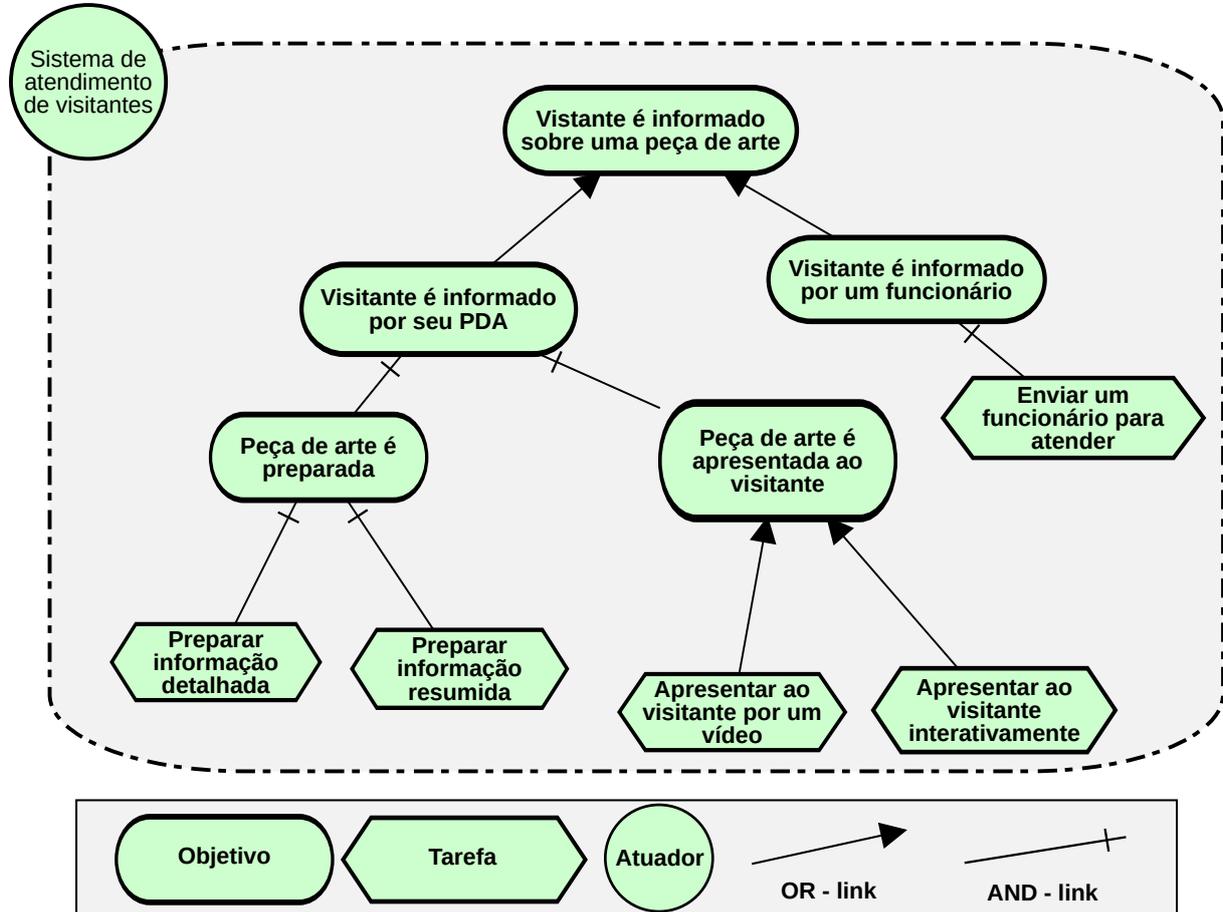


Figura 2.9: Exemplo de um *Goal Model*

2.4 Sistemas Auto-adaptativos

Conforme os computadores se tornam mais potentes e disponibilizam mais recursos para os usuários e desenvolvedores de sistemas, estes se tornam mais complexos e com mais atribuições. Um desafio enfrentado é como gerenciar esses sistemas, visto que a complexidade e a heterogeneidade destes tornavam o gerenciamento humano uma tarefa bastante difícil, ou até impossível, de ser realizada.

A computação autônoma surge como uma solução, porém a elaboração de sistemas que conseguem se auto-gerir também é um desafio. Proposta pela IBM, ela estabeleceu alguns aspectos que um sistema que queira se gerenciar deve ter [13]. Esses aspectos são:

- Auto-configuração: Capacidade de se configurar considerando políticas de alto nível.

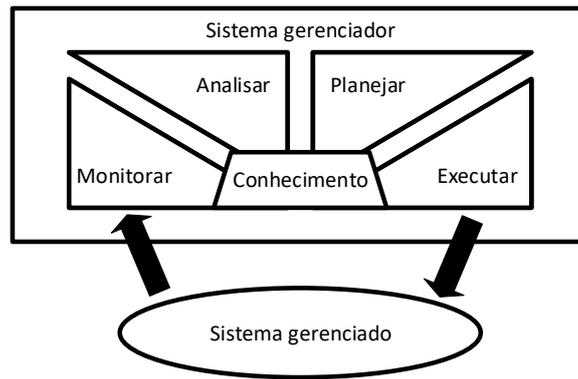


Figura 2.10: Diagrama conceitual de um sistema que utiliza MAPE-K

- Auto-otimização: Capacidade de procurar continuamente oportunidades e melhorar sua performance e eficiência.
- Auto-cura: Capacidade de detectar, diagnosticar e corrigir automaticamente problemas no sistema.
- Auto-proteção: Capacidade de se defender contra ataques e falhas.

Um modelo conceitual proposto para a implementação desses tipos de sistema é o Monitorar, Analisar, Planejar, Executar, Conhecimento (MAPE-K). Nesse modelo o sistema é composto por duas partes distintas, sendo uma responsável por gerenciar e a outra por ser gerenciada. Cada uma das partes do MAPE-K corresponde a uma função principal de um gerenciador

- Monitorar: Coleta dados do sistema e do ambiente.
- Analisar: Interpretação dos dados coletados.
- Planejar: Desenvolvimento de estratégias para o sistema conforme a análise.
- Executar: Aplicação das estratégias elaboradas
- Conhecimento: Base de dados que armazena informações, acessada nas quatro etapas.

A IBM definiu diretrizes para a computação autônoma, Weyns expandiu essas ideias [5]. Ele estabeleceu dois princípios que definem um sistema auto-adaptativo:

- Princípio Externo: O sistema deve ser capaz de lidar com mudanças e incertezas no ambiente, no próprio sistema e em seus objetivos de forma autônoma.

- Princípio Interno: O sistema é composto por duas partes. A primeira interage com o ambiente e lida com as preocupações de domínio (objetivos do sistema). A segunda é um laço de *feedback*, que interage com a primeira parte e performa a adaptação.

O laço de *feedback* atua como parte essencial da adaptação, visto que ele verifica diretamente o ambiente (que é a origem das principais falhas do sistema) e o sistema gerenciado, tem acesso aos objetivos de sistema e gera a adaptação necessária para alcançar esses objetivos definidos pelos *stakeholders*.

Capítulo 3

Controlador de Contextos da SA-BSN

O monitoramento de sinais vitais realizado pela SA-BSN é crucial para a identificação de riscos. Conforme apresentado no *Goal Model* (veja Figura 2.1), o objetivo geral de detecção de emergência é decomposto em dois principais objetivos, sendo eles o monitoramento e análise dos sinais vitais: $G3T1$ e $G4T1$ respectivamente.

A análise de sinais vitais de um paciente pode ser algo muito subjetivo e muitas vezes dados parecidos podem ter significados diferentes. Por exemplo, ao aferir os batimentos de uma pessoa e obter um valor de 130 Batimentos por minuto (BPM), esse valor sem nenhum outro sinal vital associado ou mais informações do estado da pessoa, não é suficiente para indicar se ela está em risco ou não. Caso ela esteja em repouso, pode ser considerado um valor alto. Porém, se ela estiver se exercitando, é esperado que ela possua BPM iguais ou até maiores, não se tratando de um risco.

Atualmente, a SA-BSN utiliza como referência os valores de faixa de risco que são carregados no servidor de parâmetros ao inicializar e assim se mantêm durante toda a execução da SA-BSN. Considerando uma faixa de alto risco que vai de 115 até 300 BPM, o valor 130 estaria dentro dessa faixa e esse sensor contribuiria para a identificação positiva de emergência para o paciente. No entanto, caso essa faixa não corresponda ao estado atual do paciente, os alertas da SA-BSN seriam falsos e a tarefa $G1$: *Emergency is detected* não estaria sendo feita corretamente.

A proposta deste trabalho é, com base na arquitetura existente, adicionar funcionalidades à SA-BSN para permitir que ela identifique, de forma autônoma, diversos contextos de paciente e, assim, ajuste as faixas de risco para cada contexto identificado. Dessa forma, tem-se como objetivo aumentar a precisão da detecção de alertas de risco do paciente conforme seus diferentes contextos. Minha solução consiste na definição de mais

contextos de atividades no formato da SA-BSN, na elaboração de uma heurística capaz de identificar esses contextos e na implementação dessa heurística na SA-BSN.

A Figura 4.1 exemplifica o problema em que a frequência cardíaca, nesse caso, varia bastante, inclusive fora das faixas de baixo e médio risco, porém sabe-se que não se trata de altos riscos verdadeiros.

3.1 Definição de contextos

Em engenharia de requisitos, mais especificamente em *Contextual Goal Model*, contexto é definido como um estado parcial do ambiente que seja relevante para os objetivos do atuador [12]. Ele pode variar de acordo com as capacidades de observação do atuador, o qual nem sempre tem acesso a todas as informações necessárias para definir precisamente um contexto. Além disso, alguns contextos podem não ser relevantes, mesmo que eles possam ser verificados.

Para a SA-BSN, um contexto relevante é o estado de atividade do paciente, visto que esses estados impactam diretamente na análise dos dados e detecção de emergência. O que o atuador da SA-BSN possui para obter dados relevantes para a detecção de contexto é o próprio conjunto de sinais vitais que são aferidos.

É interessante que os contextos sejam definidos de uma forma que aproveite a arquitetura existente da SA-BSN e que ao mesmo tempo não precise alterá-la de forma significativa, seja para evitar um aumento na complexidade, seja para permitir que o sistema opere externamente à detecção de contextos, mantendo seu comportamento original.

3.1.1 Contextos a partir das faixas de risco

Considerando a combinação das faixas de risco para cada um dos seis sensores — Frequência cardíaca, Oxigenação, Temperatura, Pressão arterial sistólica (APBS), Pressão arterial diastólica (APBD) e Glicose — podemos utilizar as leituras dos dados para realizar inferências sobre o contexto do paciente. Por exemplo, uma elevação na frequência cardíaca e na pressão sistólica pode indicar que o paciente está realizando uma atividade de nível mais alto, enquanto níveis estáveis e menores de glicose e temperatura sugerem que o paciente está dormindo. Isso demonstra que a utilização dos dados dos sensores é eficaz para a caracterização do contexto do paciente.

É possível então descrever vários contextos ao descrever o conjunto de faixas de risco para os sensores disponíveis.

A Figura 3.1 possui um exemplo de três contextos diferentes, em que para cada contexto são atribuídas cinco faixas de risco correspondentes. O contexto 0 corresponde aos

```

<!-- Risk values for heart frequency in context 0 -->
<param name="context0_heart_rate_HighRisk0" value="0,70" />
<param name="context0_heart_rate_MidRisk0" value="70,85" />
<param name="context0_heart_rate_LowRisk" value="85,97" />
<param name="context0_heart_rate_MidRisk1" value="97,115" />
<param name="context0_heart_rate_HighRisk1" value="115,300" />

<!-- Risk values for heart frequency in context 1 -->
<param name="context1_heart_rate_HighRisk0" value="0,40" />
<param name="context1_heart_rate_MidRisk0" value="40,50" />
<param name="context1_heart_rate_LowRisk" value="50,70" />
<param name="context1_heart_rate_MidRisk1" value="70,80" />
<param name="context1_heart_rate_HighRisk1" value="80,100" />

<!-- Risk values for heart frequency in context 2 -->
<param name="context2_heart_rate_HighRisk0" value="0,80" />
<param name="context2_heart_rate_MidRisk0" value="80,100" />
<param name="context2_heart_rate_LowRisk" value="100,140" />
<param name="context2_heart_rate_MidRisk1" value="140,160" />
<param name="context2_heart_rate_HighRisk1" value="160,300" />

```

Figura 3.1: Exemplo de parâmetros para descrição de diferentes contextos como parâmetros da SA-BSN.

Contexto	Limite inferior	Limite superior
0	85	97
1	50	70
2	100	140

Tabela 3.1: Valores de baixo risco de frequência cardíaca para diferentes contextos

valores padrão da SA-BSN [14], os outros contextos foram definidos arbitrariamente. Na Tabela 3.1, as faixas de baixo risco para cada um dos contextos estão explicitadas.

Como é possível perceber, o contexto 1 possui valores nominalmente menores que o contexto 0, já o contexto 2 possui valores maiores. Em alto nível, o contexto 0 pode representar um estado de atividade do paciente em que ele está em repouso, o contexto 1 pode representar que esteja dormindo e o contexto 2 pode representar que esteja realizando alguma atividade física.

Da mesma forma que os contextos foram definidos para o sensor de frequência cardíaca, eles devem ser definidos para todos os outros sensores, assim teremos a especificação completa de um contexto. A Figura 3.2 demonstra uma definição de como todas as faixas de risco dos sensores podem ser declaradas como um único contexto.

A descrição das faixas pode variar a cada tipo de contexto que queira ser monitorado pelos interessados e podem ser diferentes para cada paciente que está sendo monitorado, por serem parâmetros a serem implementados na SA-BSN eles são configurados antes da inicialização do sistema.

```

<!-- Risk values for oximeter in context 2 -->
<param name="context2_oxigenation_HighRisk0" value="-1,-1" />
<param name="context2_oxigenation_MidRisk0" value="-1,-1" />
<param name="context2_oxigenation_LowRisk" value="85,100" />
<param name="context2_oxigenation_MidRisk1" value="75,85" />
<param name="context2_oxigenation_HighRisk1" value="0,75" />

<!-- Risk values for heart frequency in context 2 -->
<param name="context2_heart_rate_HighRisk0" value="0,80" />
<param name="context2_heart_rate_MidRisk0" value="80,100" />
<param name="context2_heart_rate_LowRisk" value="100,140" />
<param name="context2_heart_rate_MidRisk1" value="140,160" />
<param name="context2_heart_rate_HighRisk1" value="160,300" />

<!-- Risk values for temperature in context 2 -->
<param name="context2_temperature_HighRisk0" value="0,32.99" />
<param name="context2_temperature_MidRisk0" value="33,36.99" />
<param name="context2_temperature_LowRisk" value="37,38.99" />
<param name="context2_temperature_MidRisk1" value="39,41.99" />
<param name="context2_temperature_HighRisk1" value="42,51" />

<!-- Risk values for diastolic pressure in context 2 -->
<param name="context2_abpd_HighRisk0" value="-1,-1" />
<param name="context2_abpd_MidRisk0" value="-1,-1" />
<param name="context2_abpd_LowRisk" value="0,80" />
<param name="context2_abpd_MidRisk1" value="80,90" />
<param name="context2_abpd_HighRisk1" value="90,300" />

<!-- Risk values for systolic pressure in context 2 -->
<param name="context2_abps_MidRisk0" value="-1,-1" />
<param name="context2_abps_HighRisk0" value="-1,-1" />
<param name="context2_abps_LowRisk" value="0,160" />
<param name="context2_abps_MidRisk1" value="160,170" />
<param name="context2_abps_HighRisk1" value="170,300" />

<!-- Risk values for glucose in context 2 -->
<param name="context2_glucose_HighRisk0" value="25,35" />
<param name="context2_glucose_MidRisk0" value="35,40" />
<param name="context2_glucose_LowRisk" value="40,90" />
<param name="context2_glucose_MidRisk1" value="90,110" />
<param name="context2_glucose_HighRisk1" value="110,200" />

```

Figura 3.2: Exemplo de parâmetros para descrição de diferentes contextos como parâmetros da SA-BSN

3.1.2 Cuidados na definição de contextos

A definição dos contextos deve ser feita com o auxílio de profissionais da saúde, que possuem conhecimento melhor sobre o comportamento de sinais vitais, e/ou por pesquisas que permitem elaborar faixas de risco para contextos específicos para cada perfil de paciente. Caso as faixas de risco estiverem erradas a detecção de risco da SA-BSN pode se tornar inconfiável.

Além disso, estados de paciente diferentes, mas que possuem faixas de risco semelhantes ou muito próximas se tornam difíceis de identificar. E caso existam outros sinais vitais que caracterizam melhor alguns contextos, o atuador da SA-BSN pode ser modificado para que ele tenha acesso e monitore esses sinais vitais, bem como a descrição de contexto ser feita para estes também, dessa forma ampliando as informações de ambiente

disponíveis.

3.2 Inferência

Com os contextos definidos, é necessária a elaboração de um modelo que seja capaz de inferir o contexto, e isso foi feito a partir de uma heurística. O objetivo dela é evitar que alertas falsos sobre o risco do paciente sejam gerados. A heurística foi elaborada a partir de alguns princípios:

1. Evitar falsos alertas de emergência:

- Objetivo: Minimizar alterações de contexto desnecessárias.
- Implementação: O sistema mantém o contexto atual se todas as leituras dos sensores indicarem um risco baixo, assim evitando mudanças de contexto desnecessárias.

2. Contexto de baixo risco:

- Objetivo: Adaptar o contexto do paciente de acordo com as leituras de sinais vitais.
- Implementação: Se todas as leituras dos sensores se enquadrem melhor no intervalo de baixo risco de outro contexto, assume-se que esse seja o estado do paciente e adapta o sistema a esse contexto.

3. Priorizar risco mais alto:

- Objetivo: Garantir a segurança do paciente.
- Implementação: Quando os dados dos sensores se enquadram em múltiplos contextos, o sistema seleciona o contexto que representa o maior risco, priorizando a detecção de situações potencialmente perigosas.

A seleção desses princípios foi realizada para aproveitar a arquitetura existente da SABS, evitando alterações que possam impactar os requisitos estabelecidos, ou alterá-la substancialmente. Além disso, a implementação foi feita de forma modular, permitindo que o sistema opere normalmente mesmo sem a funcionalidade de identificação de contextos ativada. Dessa forma, garantimos a continuidade do funcionamento do sistema, independentemente da ativação dessa nova funcionalidade.

3.2.1 Controlador

A inferência é realizada por um mecanismo que opera de forma análoga a um controlador, analisando continuamente os dados dos sensores e fazendo ajustes nos parâmetros. Ela foi implementada usando o modelo referencial MAPE-K, executando uma série de processos que garantem a auto-adaptação do sistema. Ele opera paralelamente ao controlador original da SA-BSN [1]. Nesse trabalho se refere ao aqui implementado, excetuando-se quando é especificado o controlador original.

O controlador é implementado como um nó ROS e escrito em *C++* utilizando conceitos de Programação Orientada a Objetos (POO). Todos os contextos, como são definidos na Figura 3.2, de todos os sensores são parâmetros utilizados por este nó e eles são previamente definidos e são carregados ao servidor de parâmetros na inicialização deste.

Além disso, outros parâmetros são definidos. Um deles é o limiar que deve ser considerado para o valor de risco ser alto, esse valor deve ser de 0-100 e corresponde à porcentagem de risco do sensor. O outro é o tamanho da fila que será utilizada para calcular a média dos últimos valores lidos, por exemplo, caso a fila tenha tamanho 10, o valor do sinal vital que será utilizado corresponde à média móvel das últimas 10 leituras de dado.

3.2.1.1 Inicialização

Na inicialização do sistema, os valores do contexto são obtidos do servidor de parâmetros e são armazenados em memória. Além disso, é feita a inscrição nos tópicos necessários.

3.2.1.2 Monitorar

O monitoramento dos dados é feito através da inscrição no tópico */TargetSystemData*. Esse tópico é criado pelo nó *G4T1* e em suas mensagens existem os dados mais recentes de todos os sensores, incluindo os riscos associados e os dados brutos (valor lido diretamente pelo sensor), que são os dados de interesse.

Por estar inscrito no tópico, o monitoramento é executado automaticamente quando há publicações de mensagem neste. Os dados de interesse são armazenados e é calculada a média das últimas leituras de sinais vitais a partir do parâmetro definido previamente. Isso é feito para evitar que picos de alteração nos valores, potenciais falhas nestes, gerem uma mudança errônea de contexto.

3.2.1.3 Analisar

Primeiramente é verificado se os sensores estão em alto risco, caso ele esteja em alto risco é verificado se ele está em baixo risco para outro contexto. Por exemplo, frequência cardíaca atual em alto risco, mas em baixo risco para contexto 2 e glicose atual em alto risco, mas

em baixo risco para contexto 1. Caso seja encontrado um, ou mais de um contexto, eles são candidatos a possível contexto que represente o estado atual do paciente.

Essa etapa cumpre os dois primeiros princípios estabelecidos para a heurística.

- Apenas um contexto:

Verifica se os dados dos outros sensores se enquadram em baixo ou médio risco para esse contexto.

- Caso eles estejam é seguido para a próxima etapa em que o sistema é adaptado para esse contexto.
- Caso algum deles não esteja é considerado que o contexto atual representa o estado de atividade do paciente e não ocorre nenhuma adaptação.

- Mais de um contexto:

Verifica se os dados dos outros sensores se enquadram em baixo ou médio risco para esses contextos.

- Caso em todos os contextos pelo menos algum dado se enquadre em alto risco, não é feita nenhuma adaptação.
- Caso somente um contexto tenha os dados de todos os sensores se enquadrando em baixo ou médio risco, esse contexto é selecionado para ser adaptado.
- Caso mais de um contexto tenha os dados de todos os sensores em baixo ou médio risco é feita uma caracterização. Existem três tipos de caracterização:
 - * Centralidade: Para sensores em que o menor risco seja o centro, por exemplo: frequência cardíaca, temperatura, glicose, será avaliado o quão centralizado o dado do sensor está, considerando as faixas de médio e baixo risco, em que o dado mais centralizado recebe valor 0 e o dado mais afastado do centro recebe valor 1.
 - * Limite superior: Para sensores em que limite superior é o valor de menor risco, por exemplo: oxigenação, é avaliado o quão próximo o dado está deste limite, caso ele for igual ao limite superior recebe o valor 0 e caso ele for igual ao limite inferior recebe o valor 1.
 - * Limite inferior: Para sensores em que o limite inferior representa o menor risco, por exemplo: pressão sistólica e diastólica, é avaliado o quão próximo o dado está deste limite, caso ele for igual ao limite inferior recebe o valor de 0, caso for igual ao limite superior recebe o valor 1.

Dessa forma, valores correspondentes a 0 indicam que o dado estaria no menor risco possível caso estivesse nesse contexto, já o valor 1 corresponde ao maior

risco. Será escolhido o contexto em que possua o sensor com o valor mais alto, ou seja, o contexto em que algum sensor individual traria o maior risco.

Essa etapa cumpre o terceiro princípio estabelecido para a heurística.

3.2.1.4 Executar

Com a adaptação certa que irá ocorrer e para qual contexto adaptar, nessa etapa será enviada uma requisição através de um serviço com os novos parâmetros das faixas de risco.

3.2.2 Visão processual

A inferência sobre o estado de atividade do paciente pode ser descrita como um processo que está representado na Figura 3.3.

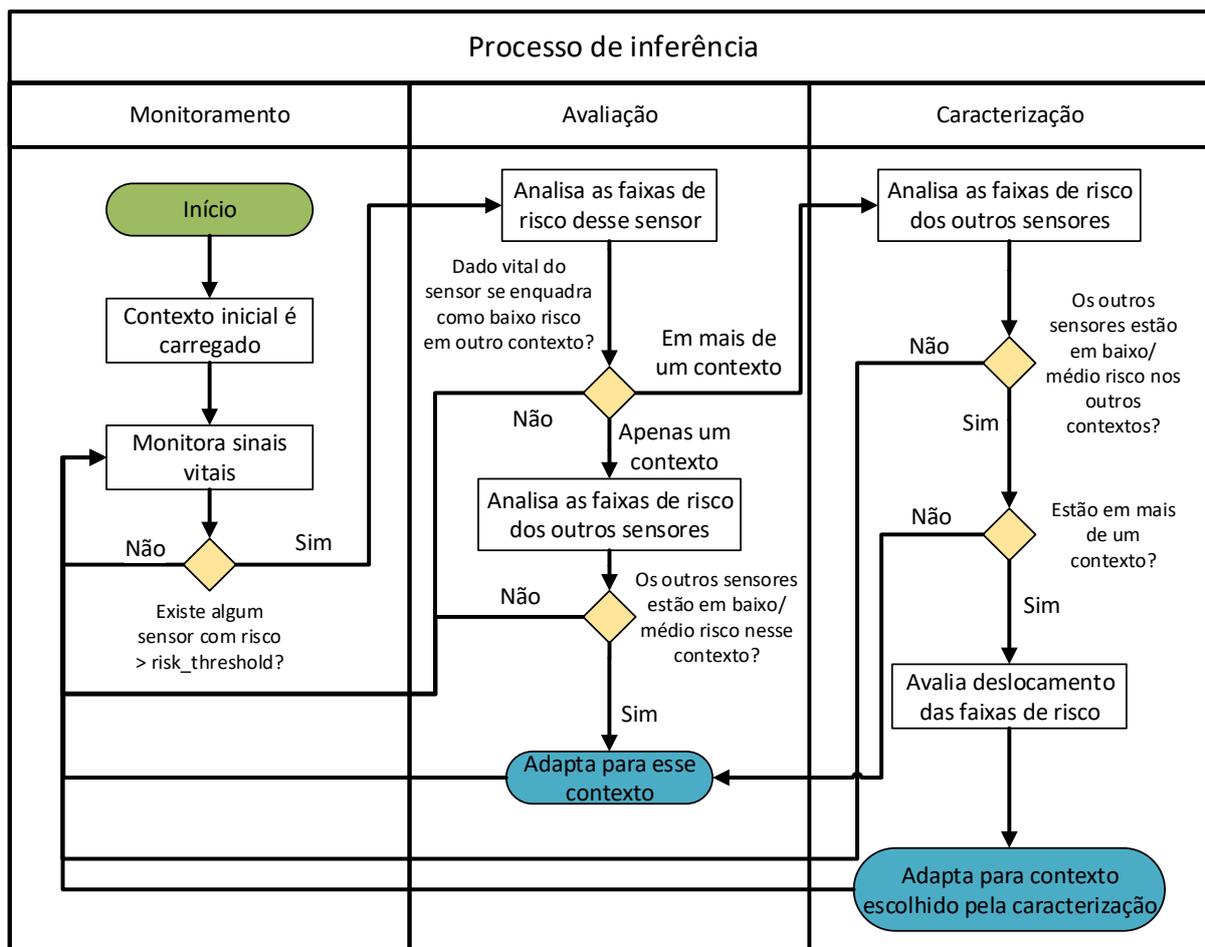


Figura 3.3: Processo do modelo de inferência

O processo foi dividido em três etapas, sendo elas Monitoramento, Avaliação e Caracterização.

Monitoramento: Na primeira etapa, os dados são continuamente monitorados e, caso algum risco seja maior que o limiar pré-definido, é avançado para a próxima etapa.

Avaliação: Nessa etapa, já se sabe que algum sensor está em alto risco. A avaliação ocorre com a verificação se os sinais vitais que estão em alto risco se enquadram como baixo risco de outro contexto. Se eles se enquadram em apenas um contexto, é verificado se todos os sinais vitais se enquadram, também, em baixo ou médio risco neste; caso sim, a adaptação é feita para esse contexto. Caso os sinais vitais estejam em alto risco para mais de um contexto, é seguido para a etapa de caracterização.

Caracterização: Como existe mais de um contexto possível, eles são caracterizados de forma a descobrir qual deles é o mais adequado para realizar a adaptação. Isso é feito através da verificação das faixas de risco (Centralidade, Limite superior e Limite inferior) já descritas.

Neste processo a heurística proposta pode ser comparada às tomadas de decisões, que são os losangos amarelos. Ou seja, a cada momento que alguma decisão é tomada ela é feita de forma a seguir a heurística.

3.2.3 Análise de contexto

A análise de contexto de uma *Contextual Goal Model* indica se uma tarefa deve ou não ser realizada. Na inferência proposta, existe a tarefa de adaptar o sistema para o contexto (estado de atividade do paciente) encontrado, dessa forma é possível criar uma análise de contexto que avalia se existe um estado do paciente a ser adaptado. (veja Figura 3.4).

Considerando C1 como "Adaptação ao estado de atividade necessita ser feita", ela pode ser verificada por uma série de declarações ou fatos sobre os sinais vitais do paciente. (veja Equações (3.1) e (3.2)).

$$C1 \text{ applies if } f1 \ \& \ \alpha \tag{3.1}$$

$$\alpha = (f2 \ \& \ f3) \ || \ (f4 \ \& \ (f5 \ || \ (f6 \ || \ f7 \ || \ f8))) \ \text{supports } d1 \tag{3.2}$$

3.3 Alteração arquitetural

Algumas alterações arquiteturais foram implementadas, elas ocorreram em alguns nós existentes da SA-BSN e em seu *Goal Model*.

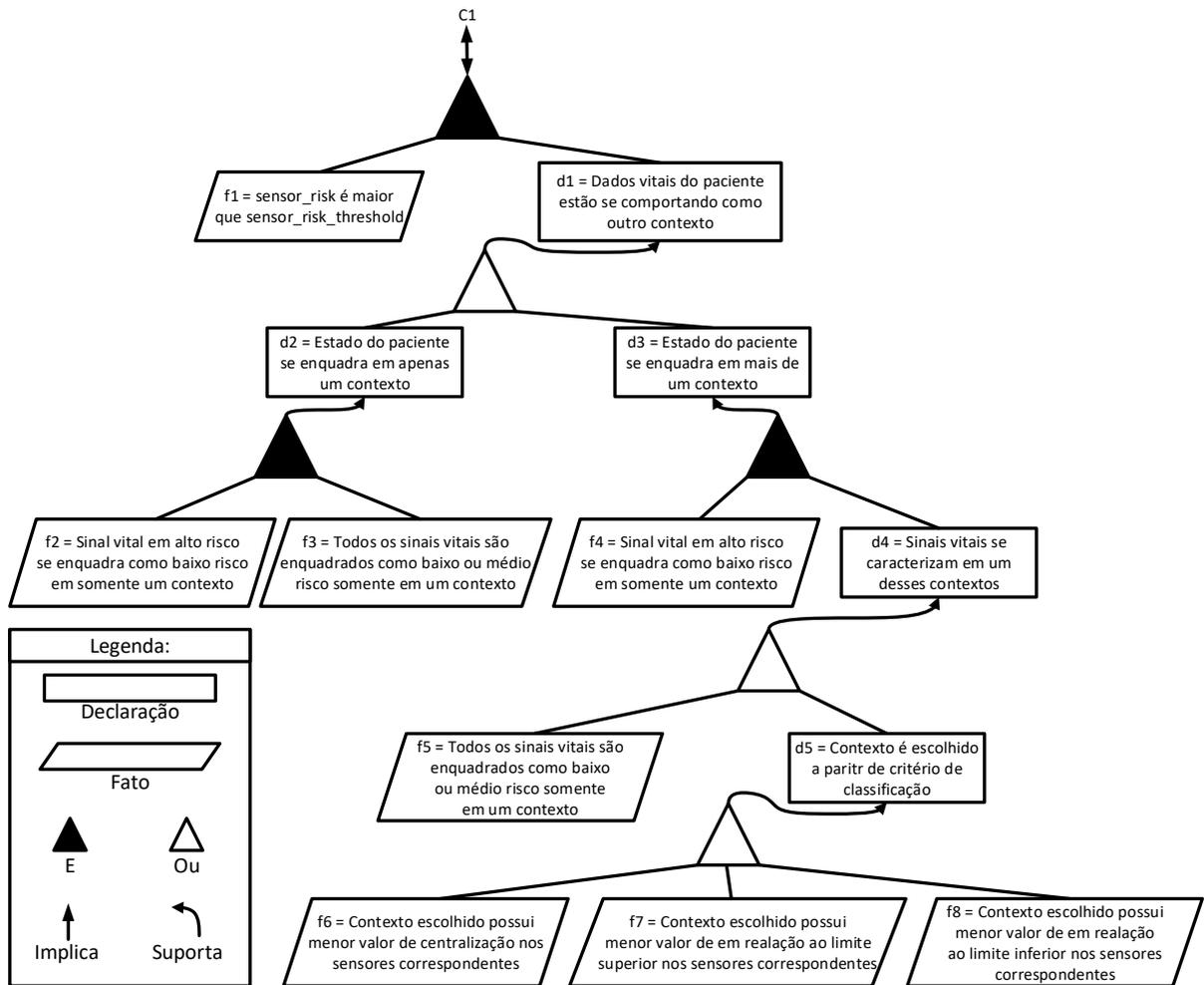


Figura 3.4: Análise do contexto C1, nessa imagem contexto se refere ao estado de atividade

3.3.1 Alteração nos nós existentes

Para realizar a adaptação de fato, algumas alterações nos nós existentes precisaram ser feitas. Elas ocorreram no nó *Patient* e nos nós dos sensores.

3.3.1.1 *Patient*

Esse nó é responsável pela geração de dados quando os sensores são virtuais. Ele carrega do servidor de parâmetros os valores das faixas de risco de cada sensor e as cadeias de Markov que indicam as transições de estado e configura o gerador de dados ao inicializar o nó.

A alteração feita neste nó foi a reconfiguração do gerador para gerar dados a partir das novas faixas de risco e atualizá-las no servidor de parâmetros.

3.3.1.2 Sensores

Os sensores, ao inicializarem, carregam do servidor de parâmetros as faixas de risco e utilizam-nas para avaliar o risco. Caso essas faixas não sejam atualizadas pelo sensor, os riscos serão calculados utilizando os valores originais, mesmo que o servidor de parâmetros tenha sido alterado.

A modificação feita nesses nós foi feita para que os valores utilizados para calcular o risco sejam atualizados, com o valor mais recente do servidor de parâmetros, toda vez que um dado novo, referente ao sinal vital, seja requisitado.

3.3.2 Remodelagem do *Goal Model*

Com a implementação da funcionalidade de verificação dos contextos o *Goal Model* da SA-BSN pode ser atualizado para receber esse novo objetivo. A Figura 3.5 representa a nova proposta.

Houve a inclusão do objetivo *G5: Patient context is verified*, que possui a tarefa *G5T1: Verify context*, correspondente à verificação do contexto do paciente realizada através da inferência proposta neste projeto. Ao inserir este objetivo, assume-se que é do interesse dos envolvidos saber o contexto do paciente, pois ele interfere diretamente na detecção de emergências. Portanto, faz sentido assumir isso.

3.3.3 Adição de novo nó

A implementação da inferência é feita através de um novo nó ROS escrito em C++, seu código fonte está disponível no *github* [15]. O nó é chamado de */context_adaptation* e ele

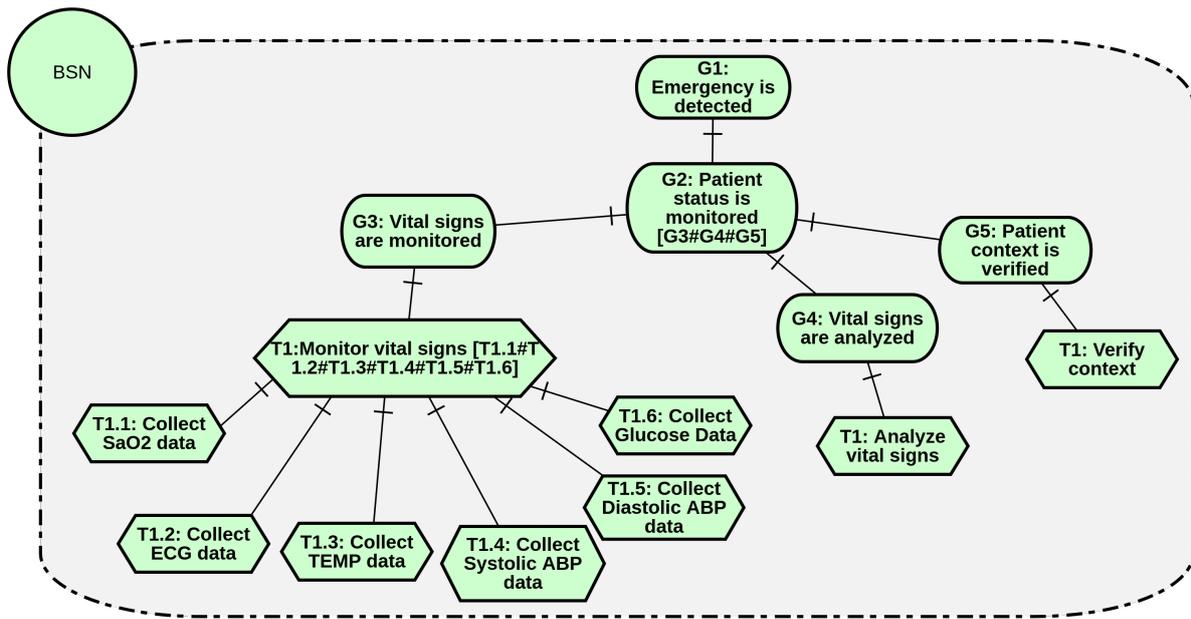


Figura 3.5: Novo *Goal Model* proposto para a SA-BSN

está inscrito no tópico */targetSystemData*, em que recebe os dados de todos os sensores, e é cliente do serviço */setAdaptation*, em que envia os novos parâmetros das faixas de risco.

3.3.4 Requisitos funcionais e não funcionais

Este trabalho manteve todos os requisitos originais da SA-BSN e adicionou mais alguns relacionados à detecção de contexto, sendo eles:

- Requisito funcional: A SA-BSN deve ser capaz de identificar o contexto de estado de atividade do paciente de acordo com os contextos definidos previamente.
- Requisito não funcional: A SA-BSN deve ter seus parâmetros adaptados de acordo com os contextos identificados.

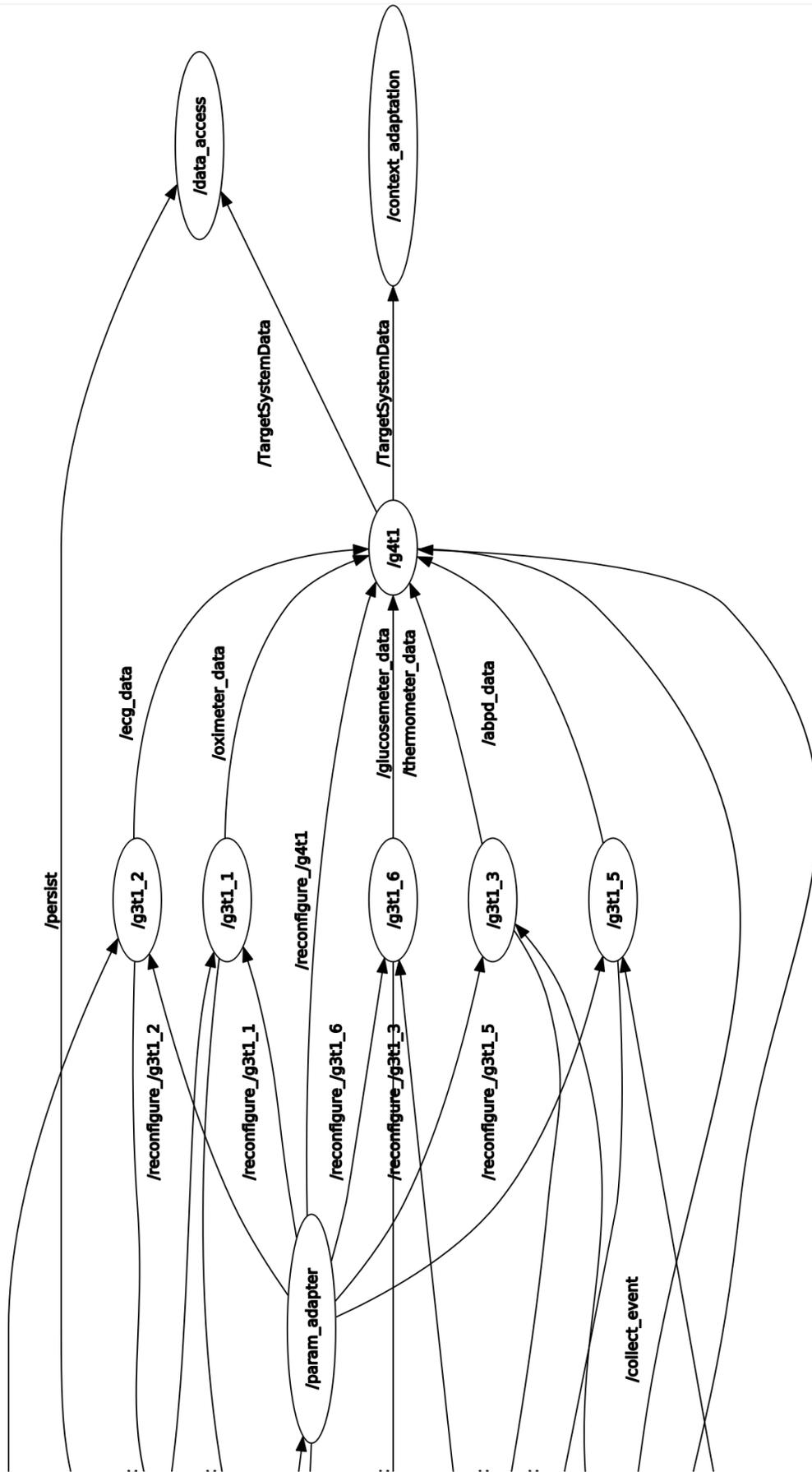


Figura 3.6: Recorte do gráfico de nós e tópicos após adição do nó */context_adaptation*

Capítulo 4

Validação

A heurística foi validada utilizando dados reais de sensores a partir do monitoramento de um voluntário. Esses dados foram também apresentados na Escola de Verão Internacional Robotic Mission Engineering (RoME) 2024, realizada na Universidade de Brasília ¹. Os dados fornecidos foram obtidos através de uma Interface de Programação de Aplicação (API) fornecida por um relógio inteligente. Esse dispositivo realizou o monitoramento e forneceu dados de frequência cardíaca e oxigenação aferidos em um período de vinte e quatro horas. Durante esse período, sabe-se que a pessoa monitorada se manteve saudável e esteve em alguns estados de atividade diferentes.

A Figura 4.1 apresenta os dados de frequência cardíaca obtidos ao longo das vinte e quatro horas e as faixas de baixo (verde) e médio risco (laranja), já a Figura 4.2 apresenta, da mesma forma, os dados para a oxigenação. É possível perceber que durante a execução os valores de frequência cardíaca variaram bastante, enquanto os de oxigenação se mantiveram mais estáveis.

Também é possível perceber que muitos dos valores aferidos para a frequência cardíaca estão fora da faixa de baixo, e até, da de médio risco. Dessa forma caso SA-BSN fosse executada utilizando esses dados como entradas para os sensores eles indicariam um nível de risco alto durante a execução.

4.1 Configuração

Para a realização do experimento foi feito um programa em *python* que atualizaria a entrada de dados da SA-BSN com os dados provindos dos dados reais, de forma em que eles fossem percorridos por volta de 12 minutos. Após isso os *logs* foram analisados para mostrar os valores de risco obtidos durante essa execução. (veja Figura 4.3).

¹<https://lesunb.github.io/RoME/>

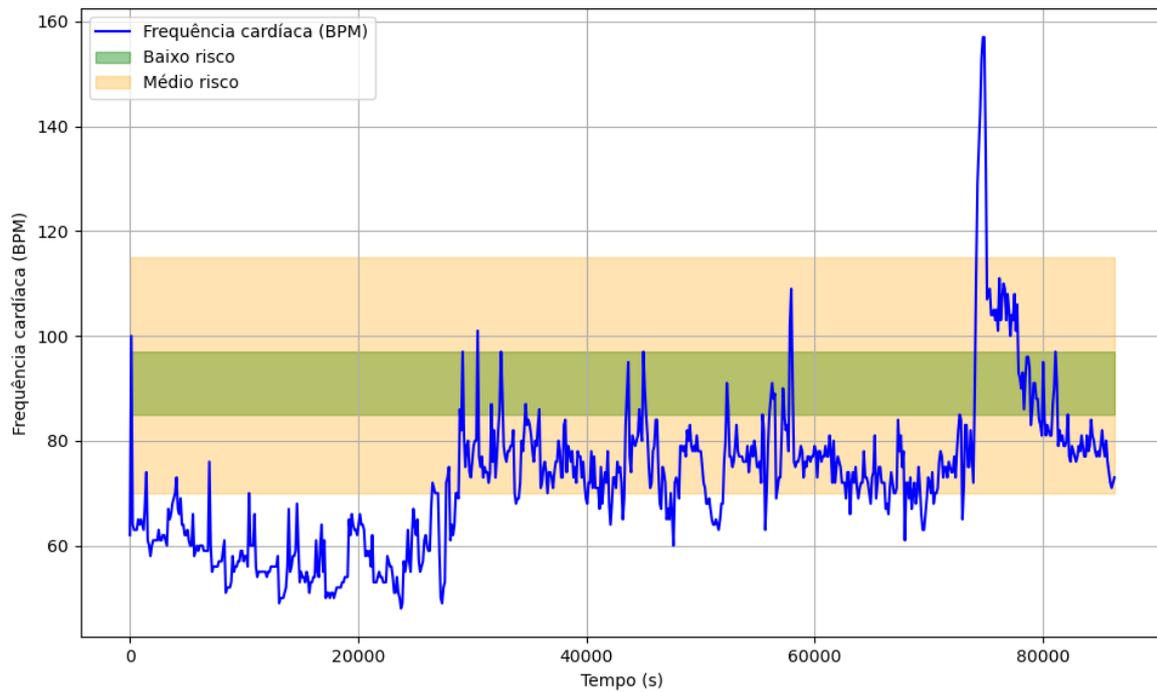


Figura 4.1: Gráfico da frequência cardíaca ao longo do tempo, com marcação das faixas de risco.

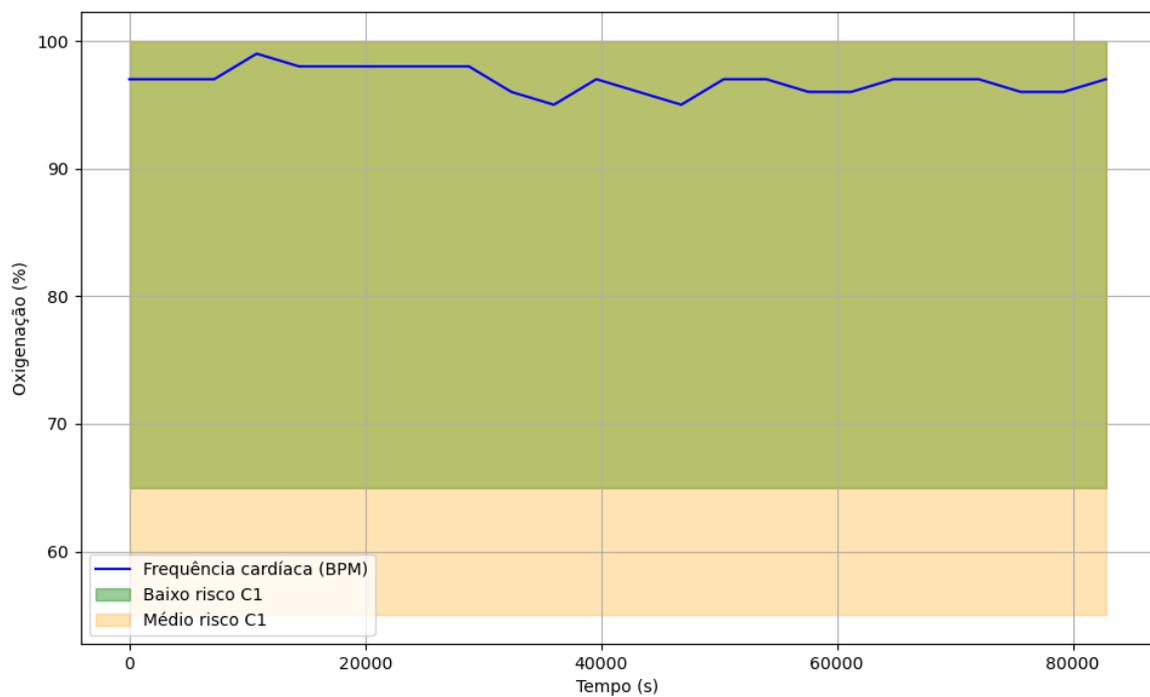


Figura 4.2: Gráfico da oxigenação ao longo do tempo, com marcação das faixas de risco.

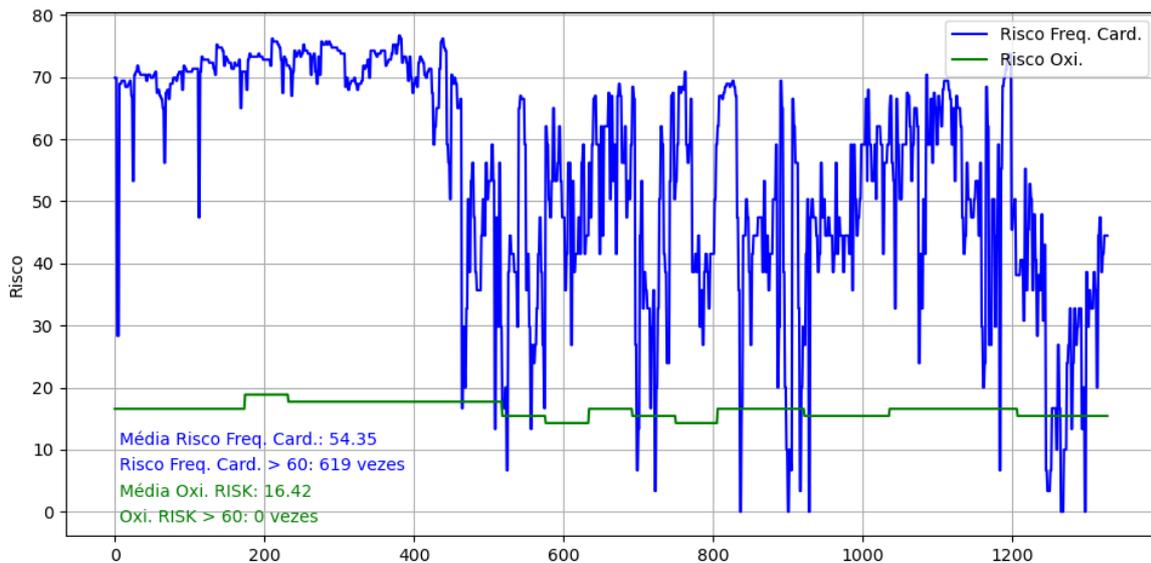


Figura 4.3: Valores de risco durante a execução da SA-BSN

4.1.1 Análise

Observando este gráfico, é possível perceber que durante a maior parte da execução os valores de risco da frequência cardíaca (ECG) estavam altos, com uma média de 54,35%, sendo que em 619 vezes ele foi calculado como maior que 60%. Já a oxigenação (Oxim.) se manteve mais baixa, com uma média de 16,42%, com nenhuma ocorrência de risco maior que 60%. Esses dados são esperados, já que a frequência cardíaca estava variando bastante e fora da faixa de baixo risco, enquanto a oxigenação se manteve estável dentro dela.

4.2 Definição das faixas de risco

Como se sabe que durante esse período o paciente esteve saudável, é possível criar faixas de risco para os sensores disponíveis de forma a contemplar os diferentes valores que eles estiveram. Com isso, foram criados três contextos. Um deles sendo as faixas originais presentes na Figura 4.1, as outras duas correspondem a faixas com valores menores e maiores, respectivamente, de frequência cardíaca, enquanto as faixas de oxigenação foram mantidas as mesmas da original, já que não houve maiores variações.

As Figuras 4.4 e 4.5 apresentam as faixas de risco para o contexto 1 e 2, respectivamente. Já as Tabelas 4.1, 4.2 e 4.3 apresentam essas faixas de risco numericamente para os contextos 0, 1 e 2, respectivamente.

Médio risco 0	70	85
Baixo risco	85	97
Médio risco 1	97	115

Tabela 4.1: Faixas de risco para contexto 0

Médio risco 0	40	55
Baixo risco	55	73
Médio risco 1	73	90

Tabela 4.2: Faixas de risco de BPM para contexto 1

Médio risco 0	90	100
Baixo risco	100	130
Médio risco 1	130	150

Tabela 4.3: Faixas de risco de BPM para contexto 2

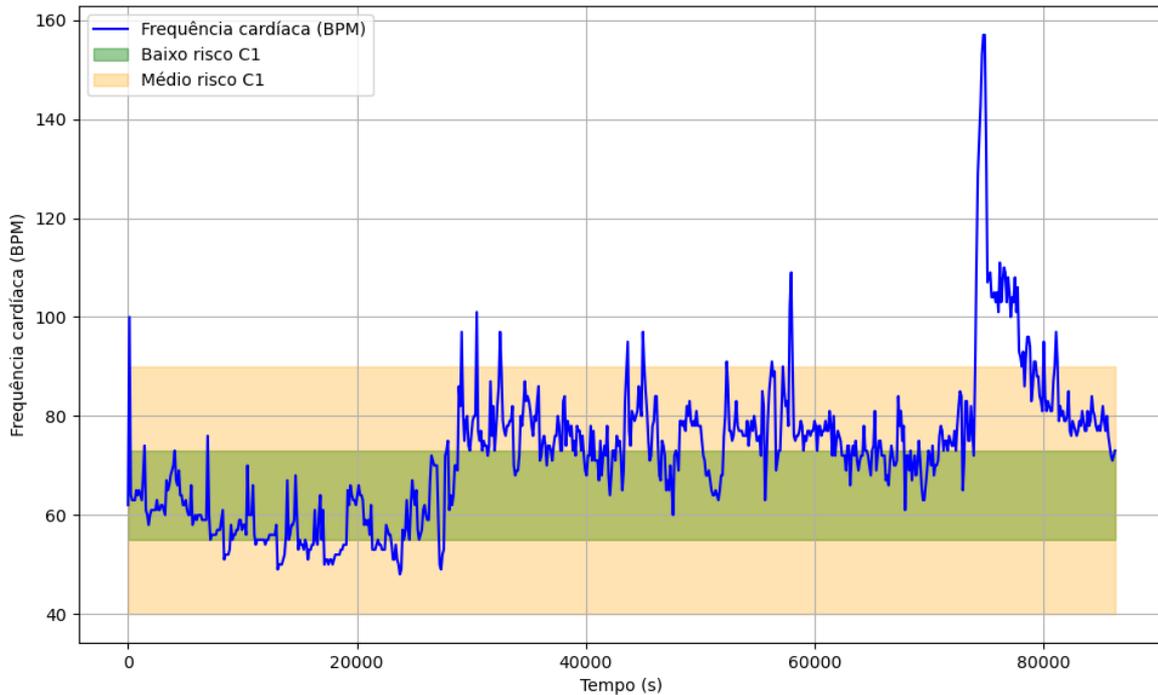


Figura 4.4: Faixas de risco de BPM para contexto 1

4.3 Resultados esperados

Os resultados esperados para um experimento que utiliza dados de frequência cardíaca e oxigenação como entrada, juntamente com um controlador acionado, são que a SA-BSN ajuste o contexto conforme a variação da frequência cardíaca. Assim, quando a frequência cardíaca estiver na faixa de baixo risco, o sistema se adaptará a essa condição, considerando um risco calculado maior que 60%, considerando que os dados de oxigenação

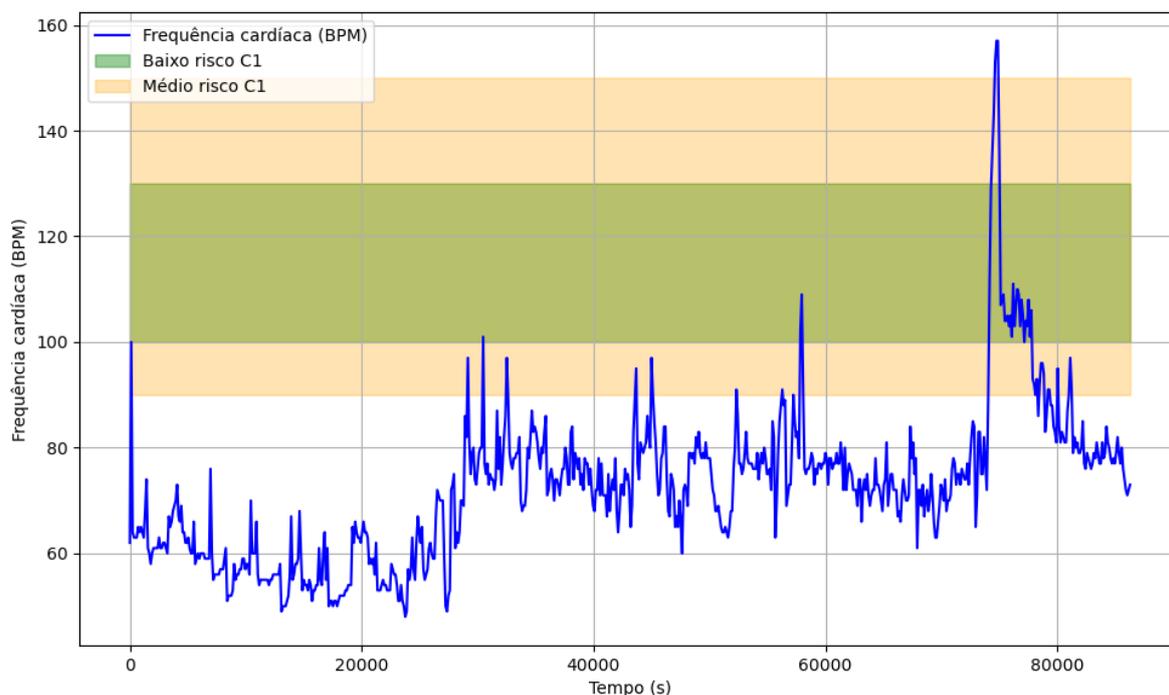


Figura 4.5: Faixas de risco de BPM para contexto 2

permanecem consistentemente em baixo risco.

A Figura 4.6 apresenta um gráfico da frequência cardíaca ao longo do tempo, em que um filtro passa-baixas foi aplicado, sendo uma média móvel simples dos últimos cinco valores, para corresponder ao filtro aplicado pelo controlador. Ele também apresenta as três faixas de baixo risco, uma para cada contexto. Por fim, a cor da linha representa o contexto esperado para aquele momento.

A linha muda de cor toda vez que o valor fica dentro da faixa de baixo risco de outro contexto. Caso ele não fique dentro de nenhuma faixa, o contexto se manterá de acordo com a última faixa, já que um dos critérios para realizar a mudança de contexto é justamente que o dado esteja dentro da faixa de baixo risco de outro contexto.

Dessa forma, é esperado que ocorram onze mudanças de contexto, visualmente representadas como mudanças de cores no gráfico, sendo que uma delas corresponde à primeira mudança, já que o sistema assume o contexto inicial como o contexto 0.

4.4 Experimento

Para a realização do experimento, as faixas de risco foram definidas como parâmetros do nó controlador, e foi feita uma nova execução, desta vez com o controlador iniciado.

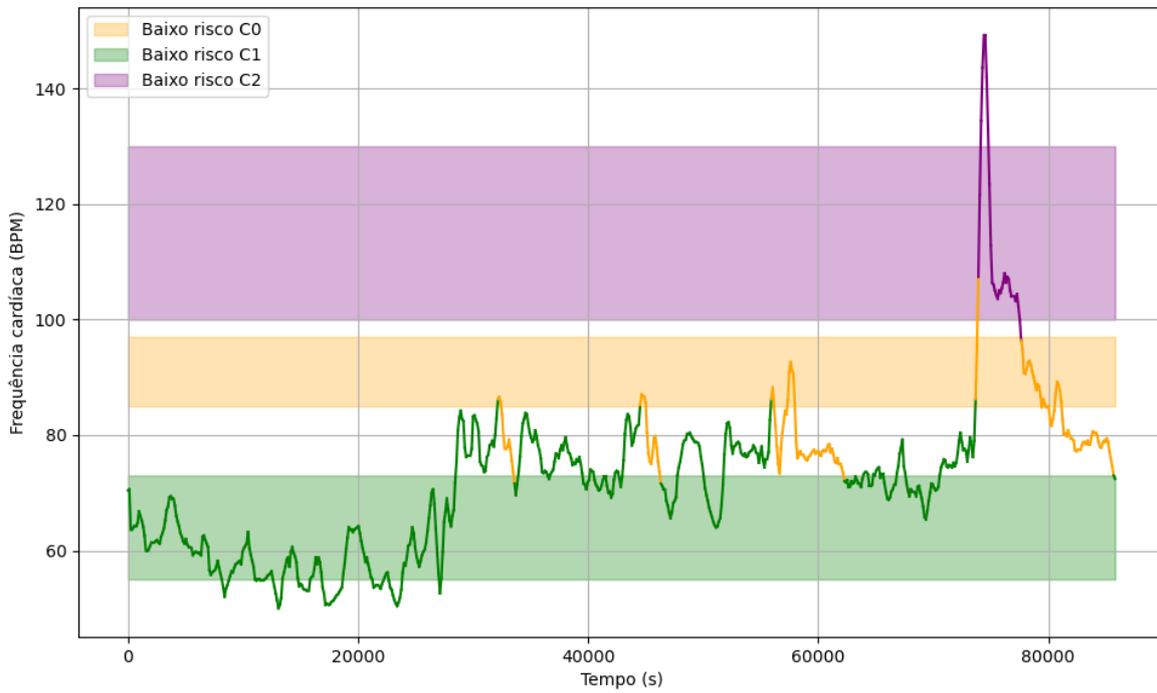


Figura 4.6: Gráfico da frequência cardíaca ao longo do tempo com as faixas de baixo risco e filtro de passa baixas e contextos esperados

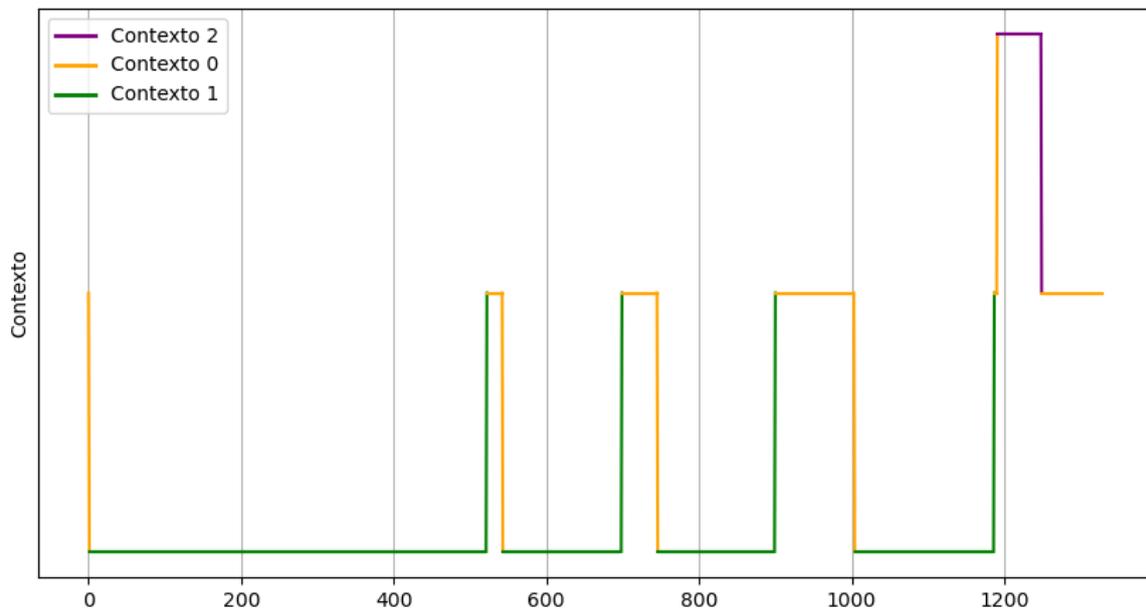


Figura 4.7: Gráfico do contexto durante a execução

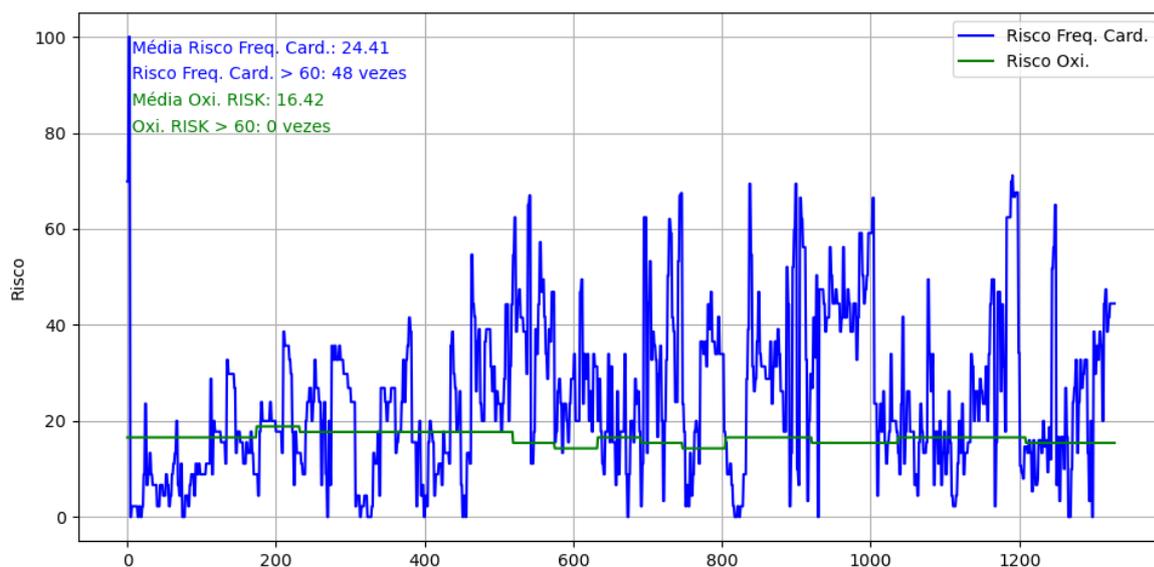


Figura 4.8: Valores de risco durante a execução SA-BSN com o controlador ativado

	Original	Com adaptação	Porcentagem de diferença
Valor médio do risco (%)	54,35	24,41	55,09%
Ocorrências de risco > 60%	619	48	92,25%

Tabela 4.4: Resultados e comparação das execuções

A Figura 4.8 apresenta os valores de risco obtidos através dessa execução. A média do valor de risco para a frequência cardíaca foi de 24,41%, sendo que 48 desses valores foram maiores que 60%. Já os valores de risco da oxigenação permaneceram os mesmos. Isso representa uma redução de 92,25% na quantidade de alertas de risco maiores que 60% e uma redução de 55,09% no valor médio da porcentagem de risco.

A Figura 4.7 apresenta os contextos calculados e aplicados ao longo da execução. Comparando com a Figura 4.6, observa-se que os contextos calculados pelo controlador correspondem aos contextos esperados, exceto pelo último. Era esperado que a execução terminasse no contexto 1, porém, ao observar a Figura 4.8, nota-se que o risco ao final da execução não foi maior que 60%, valor esperado para realizar a adaptação. Ou seja, de fato, não deveria ocorrer essa adaptação.

Além disso, observando a Figura 4.8, é possível identificar picos de alto risco. Alguns desses picos estão associados a mudanças de contexto, já que esse é o critério inicial para realizar alguma adaptação. Logo após essa identificação, o risco abaixa, pois as faixas para o cálculo do risco foram atualizadas e os valores dos sensores se enquadram nessa faixa.

Conclui-se, portanto, que o algoritmo foi capaz de identificar diferentes contextos e aplicar as mudanças de faixas de risco na SA-BSN.

Capítulo 5

Conclusão

5.1 Conclusões gerais

Com os avanços da tecnologia, tanto a computação quanto os sistemas biomédicos se tornam mais onipresentes e acessíveis. Dessa forma, a necessidade de identificação dos estados de atividade passa a ser um passo importante para esses sistemas, onde a computação pode ser aliada para a criação de algoritmos que analisem isso.

Esse trabalho utilizou a SA-BSN como sistema base para a implementação de um algoritmo capaz de identificar diferentes estados de atividade de pacientes e adaptar seus parâmetros para tais. Através da análise dos resultados, chega-se à conclusão que a solução proposta para a verificação dos estados de atividade do paciente foi implementada corretamente pelo algoritmo, seguindo todos os critérios estabelecidos pela heurística proposta. Além disso, a heurística foi capaz de identificar corretamente os contextos, evidenciando ser uma solução válida para esse problema. Assim, a SA-BSN se torna um sistema biomédico mais adaptável, já que, além das adaptações já realizadas originalmente (para atingir uma confiabilidade ou custo desejados [1]), agora ela consegue adaptar seus parâmetros de acordo com os estados de atividade do paciente.

5.2 Limitações

Muitas vezes, a caracterização de estados de atividade depende de muitas informações do ambiente. Por exemplo, é muito mais provável que uma pessoa esteja dormindo se ela estiver em seu quarto com as luzes apagadas, e uma pessoa realizando uma atividade física estará em movimento e provavelmente terá a respiração acelerada. Nem sempre todas as informações de ambiente necessárias estarão disponíveis para o atuador da SA-BSN.

Ao fazer a modelagem dos contextos, as informações de ambiente que o atuador tem acesso devem ser levadas em consideração. Caso algum contexto que se queira especificar

não seja muito bem caracterizado pelos sensores disponíveis, é possível que a SA-BSN não faça uma identificação correta. Além disso, se contextos diferentes acabarem sendo caracterizados de uma forma muito parecida, é possível que o sistema identifique um no lugar do outro.

Além disso, a solução proposta depende da definição prévia dos contextos específicos para cada paciente, não sendo possível a inserção de novos contextos durante a execução e não sendo capaz de identificar o contexto de uma forma mais generalizada para qualquer paciente.

5.3 Trabalhos futuros

Existem alguns trabalhos que podem ser realizados de forma a complementar e ampliar o que foi realizado.

5.3.1 Ampliação das informações de ambiente

Como citado neste capítulo, nem sempre alguns estados serão bem representados pelos conjuntos de sensores atuais disponíveis no atuador da SA-BSN. Dessa forma, trabalhos que visam ampliar essa gama de sensores podem contribuir com o atual, melhorando a modelagem e a caracterização de diferentes contextos.

5.3.2 Criação de faixas de risco

A adaptação é feita a partir de faixas de risco definidas. Dessa forma, trabalhos que procuram caracterizar contextos de diferentes perfis de pacientes por faixas de risco, através de bases de dados existentes ou coletando os próprios dados, atuam em conjunto com o atual, visto que garantiriam que os contextos modelados representem de fato um estado de atividade de um paciente.

5.3.3 Implementação de diferentes algoritmos

O trabalho permitiu à SA-BSN ter seus parâmetros de faixas de risco adaptados por um controlador baseado em uma heurística. Diferentes trabalhos podem criar novos algoritmos para a detecção desses contextos utilizando outras abordagens. Por exemplo, algoritmos de redes neurais poderiam fazer essa identificação com dados treinados a partir de um ou de um conjunto de pacientes, ou com algum treinamento para identificar anomalias independentemente da quantidade de sensores.

Referências

- [1] Gil, Eric Bernd, Ricardo Caldas, Arthur Rodrigues, Gabriel Levi Gomes da Silva, Genáina Nunes Rodrigues e Patrizio Pelliccione: *Body sensor network: A self-adaptive system exemplar in the healthcare domain*. Em *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, páginas 224–230, May 2021. ix, 2, 4, 7, 26, 41
- [2] Pfeiffer, Bruna Ferreira e Cristina Pio de Almeida: *Relevância do uso de smartwatch para detecção e monitoramento de fibrilação atrial: um mapeamento sistemático*. Res. Soc. Dev., 11(16):e136111637774, dezembro 2022. 1
- [3] *Use Crash Detection on iPhone or Apple Watch to call for help in an accident*. <https://support.apple.com/en-us/104959>, 2024. [Accessed 15-08-2024]. 1
- [4] Wyatt, Kirk D, Lisa R Poole, Aidan F Mullan, Stephen L Kopecky e Heather A Heaton: *Clinical evaluation and diagnostic yield following evaluation of abnormal pulse detected using apple watch*. J. Am. Med. Inform. Assoc., 27(9):1359–1363, setembro 2020. 1
- [5] Weyns, Danny: *An introduction to self-adaptive systems: A Contemporary Software Engineering Perspective*. Wiley, 2021. 2, 7, 19
- [6] Knight, J.C.: *Safety critical systems: challenges and directions*. Em *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, páginas 547–550, May 2002. 2
- [7] Félix Solano, Gabriela, Ricardo Diniz Caldas, Genáina Nunes Rodrigues, Thomas Vogel e Patrizio Pelliccione: *Taming uncertainty in the assurance process of self-adaptive systems: a goal-oriented approach*. Em *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, páginas 89–99, 2019. 4
- [8] Caldas, Ricardo Diniz, Arthur Rodrigues, Eric Bernd Gil, Genáina Nunes Rodrigues, Thomas Vogel e Patrizio Pelliccione: *A hybrid approach combining control theory and ai for engineering self-adaptive systems*. Em *2020 IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, páginas 9–19, 2020. 4
- [9] Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng *et al.*: *Ros: an open-source robot operating system*. Em *ICRA workshop on open source software*, volume 3, página 5. Kobe, Japan, 2009. 4

- [10] Wyrobek, Keenan: *The Origin Story of ROS, the Linux of Robotics* — *spectrum.ieee.org*. <https://spectrum.ieee.org/the-origin-story-of-ros-the-linux-of-robotics>, 2017. [Accessed 25-08-2024]. 11
- [11] Joseph, Lentin: *Mastering ROS for robotics programming*. Packt Publishing, Birmingham, England, dezembro 2015. 11
- [12] Ali, Raian, Fabiano Dalpiaz e Paolo Giorgini: *A goal-based framework for contextual requirements modeling and analysis*. *Requirements engineering*, 15:439–458, 2010. 14, 22
- [13] Kephart, J.O. e D.M. Chess: *The vision of autonomic computing*. *Computer*, 36(1):41–50, 2003. 18
- [14] Caldas, Ricardo Diniz: *Self-Organized Mapping for environmental non-determinism simulation at WEKA* — *github.com*. <https://github.com/rdinizcal/markov-sensors>, 2024. [Accessed 03-09-2024]. 23, 45
- [15] Lottermann, Carlos Eduardo Taborda: *GitHub - carloseduardotl/bsn_context: Body Sensor Network (BSN): a prototype for exercising dependable adaptation in healthcare domain*. — *github.com*. https://github.com/carloseduardotl/bsn_context, 2024. [Accessed 06-09-2024]. 31

Anexo I

Parâmetros de inicialização

Este anexo demonstra como diferentes contextos foram implementados na SA-BSN. Cada sensor possui a definição de três contextos (0, 1 e 2), onde todas as faixas de risco do contexto 0 representam o próprio contexto, e assim por diante. O contexto 0 corresponde às faixas de risco implementadas originalmente na SA-BSN, que foram definidas previamente [14]. Os outros contextos foram definidos por este trabalho, representando faixas de risco abaixo (contexto 1) e acima (contexto 2) dos valores normais de frequência cardíaca, possivelmente representando, por exemplo, um contexto de estado de atividade de uma pessoa dormindo e realizando alguma atividade física, respectivamente.

```

<!-- Risk threshold and queue size-->
<param name="adapt_risk_threshold" value="60" />
<param name="number_of_last_readings" value="5" />

<!-- Risk values for oximeter in context -->
<param name="context0_oxigenation_HighRisk0" value="-1,-1" />
<param name="context0_oxigenation_MidRisk0" value="-1,-1" />
<param name="context0_oxigenation_LowRisk" value="65,100" />
<param name="context0_oxigenation_MidRisk1" value="55,65" />
<param name="context0_oxigenation_HighRisk1" value="0,55" />

<param name="context1_oxigenation_HighRisk0" value="-1,-1" />
<param name="context1_oxigenation_MidRisk0" value="-1,-1" />
<param name="context1_oxigenation_LowRisk" value="65,100" />
<param name="context1_oxigenation_MidRisk1" value="55,65" />
<param name="context1_oxigenation_HighRisk1" value="0,55" />

<param name="context2_oxigenation_HighRisk0" value="-1,-1" />
<param name="context2_oxigenation_MidRisk0" value="-1,-1" />
<param name="context2_oxigenation_LowRisk" value="65,100" />
<param name="context2_oxigenation_MidRisk1" value="55,65" />
<param name="context2_oxigenation_HighRisk1" value="0,55" />

<!-- Risk values for heart frequency -->
<param name="context0_heart_rate_HighRisk0" value="0,70" />
<param name="context0_heart_rate_MidRisk0" value="70,85" />
<param name="context0_heart_rate_LowRisk" value="85,97" />
<param name="context0_heart_rate_MidRisk1" value="97,115" />
<param name="context0_heart_rate_HighRisk1" value="115,300" />

<param name="context1_heart_rate_HighRisk0" value="0,40" />
<param name="context1_heart_rate_MidRisk0" value="40,55" />
<param name="context1_heart_rate_LowRisk" value="55,73" />
<param name="context1_heart_rate_MidRisk1" value="73,90" />
<param name="context1_heart_rate_HighRisk1" value="90,100" />

<param name="context2_heart_rate_HighRisk0" value="0,90" />
<param name="context2_heart_rate_MidRisk0" value="90,100" />
<param name="context2_heart_rate_LowRisk" value="100,130" />
<param name="context2_heart_rate_MidRisk1" value="130,150" />
<param name="context2_heart_rate_HighRisk1" value="150,300" />

<!-- Risk values for temperature -->
<param name="context0_temperature_HighRisk0" value="0,31.99" />
<param name="context0_temperature_MidRisk0" value="32,35.99" />
<param name="context0_temperature_LowRisk" value="36,37.99" />
<param name="context0_temperature_MidRisk1" value="38,40.99" />
<param name="context0_temperature_HighRisk1" value="41,50" />

<param name="context1_temperature_HighRisk0" value="0,29.99" />
<param name="context1_temperature_MidRisk0" value="30,33.99" />
<param name="context1_temperature_LowRisk" value="34,35.99" />
<param name="context1_temperature_MidRisk1" value="36,38.99" />
<param name="context1_temperature_HighRisk1" value="39,48" />

<param name="context2_temperature_HighRisk0" value="0,32.99" />
<param name="context2_temperature_MidRisk0" value="33,36.99" />
<param name="context2_temperature_LowRisk" value="37,38.99" />
<param name="context2_temperature_MidRisk1" value="39,41.99" />
<param name="context2_temperature_HighRisk1" value="42,51" />

```

Figura I.1: Exemplo de parâmetros de inicialização do nó controlador. Parte 1/2

```

<!-- Risk values for diastolic pressure -->
<param name="context0_abpd_HighRisk0" value="-1,-1" />
<param name="context0_abpd_MidRisk0" value="-1,-1" />
<param name="context0_abpd_LowRisk" value="0,80" />
<param name="context0_abpd_MidRisk1" value="80,90" />
<param name="context0_abpd_HighRisk1" value="90,300" />

<param name="context1_abpd_HighRisk0" value="-1,-1" />
<param name="context1_abpd_MidRisk0" value="-1,-1" />
<param name="context1_abpd_LowRisk" value="0,80" />
<param name="context1_abpd_MidRisk1" value="80,90" />
<param name="context1_abpd_HighRisk1" value="90,300" />

<param name="context2_abpd_HighRisk0" value="-1,-1" />
<param name="context2_abpd_MidRisk0" value="-1,-1" />
<param name="context2_abpd_LowRisk" value="0,80" />
<param name="context2_abpd_MidRisk1" value="80,90" />
<param name="context2_abpd_HighRisk1" value="90,300" />

<!-- Risk values for systolic pressure -->
<param name="context0_abps_MidRisk0" value="-1,-1" />
<param name="context0_abps_HighRisk0" value="-1,-1" />
<param name="context0_abps_LowRisk" value="0,120" />
<param name="context0_abps_MidRisk1" value="120,140" />
<param name="context0_abps_HighRisk1" value="140,300" />

<param name="context1_abps_MidRisk0" value="-1,-1" />
<param name="context1_abps_HighRisk0" value="-1,-1" />
<param name="context1_abps_LowRisk" value="0,110" />
<param name="context1_abps_MidRisk1" value="110,130" />
<param name="context1_abps_HighRisk1" value="130,300" />

<param name="context2_abps_MidRisk0" value="-1,-1" />
<param name="context2_abps_HighRisk0" value="-1,-1" />
<param name="context2_abps_LowRisk" value="0,160" />
<param name="context2_abps_MidRisk1" value="160,170" />
<param name="context2_abps_HighRisk1" value="170,300" />

<!-- Risk values for glucose -->
<param name="context0_glucose_HighRisk0" value="20,39.99" />
<param name="context0_glucose_MidRisk0" value="40,54.99" />
<param name="context0_glucose_LowRisk" value="55,95.99" />
<param name="context0_glucose_MidRisk1" value="96,119.99" />
<param name="context0_glucose_HighRisk1" value="120,200" />

<param name="context1_glucose_HighRisk0" value="20,30" />
<param name="context1_glucose_MidRisk0" value="30,45" />
<param name="context1_glucose_LowRisk" value="45,85" />
<param name="context1_glucose_MidRisk1" value="85,105" />
<param name="context1_glucose_HighRisk1" value="105,200" />

<param name="context2_glucose_HighRisk0" value="25,35" />
<param name="context2_glucose_MidRisk0" value="35,50" />
<param name="context2_glucose_LowRisk" value="50,90" />
<param name="context2_glucose_MidRisk1" value="90,110" />
<param name="context2_glucose_HighRisk1" value="110,200" />

```

Figura I.2: Exemplo de parâmetros de inicialização do nó controlador. Parte 2/2