Consciência de Situação em Sistemas Ubíquos para Saúde: uma Análise das Técnicas

Douglas Adalberto Scheunemann, UCPEL

Resumo—A consciência de situação aplicada em sistemas ubíquos para saúde possibilita uma série de inferências sobre o estado de pacientes e sobre o próprio sistema, potencializando os recentes avanços em computação móvel e distribuída. Pesquisas apontam que um número expressivo dos *middlewares* para computação ubíqua não possuem funções para detecção de contexto e inferência de situação. Dentre as causas para esta ausência, pode ser citada a complexidade das camadas de software que necessitam ser integradas aos *middlewares* para disponibilizar estas funções. Em sistemas para medicina, a confiabilidade dos resultados das inferências é um fator essencial para que não sejam induzidos erros médicos. Nessa perspectiva, existem desafios ligados às técnicas de inferência de situação que necessitam de tratamento mais aprofundado. Neste trabalho é apresentado um panorama dos principais métodos para consciência de situação em computação ubíqua, e também uma análise sobre alguns casos de aplicação na área da saúde. Ao final, busca-se apontar quais são as tendências e os principais desafios de pesquisa nessa área.

Palavras-Chave—Consciência de situação, Consciência de contexto, Sistemas ubíquos, Saúde.

1 Introdução

Ao longo da história a medicina tem se beneficiado dos avanços tecnológicos promovendo melhores diagnósticos e possibilidades de tratamento, aumentando assim a qualidade de vida das pessoas. Ferramentas de informática são amplamente aplicadas na medicina, seja em etapas ligadas ao diagnóstico de doenças, monitoramento de pacientes ou gerenciamento de informações.

No início da década de noventa, Mark Weiser utilizou o termo "computação ubíqua" para caracterizar uma tendência em sistemas computacionais que acreditava ser promissora. Weiser sugeriu que, no futuro, os computadores passariam a integrar as atividades humanas, sem que fossem percebidos, provendo ambientes e sistemas mais inteligentes. Isso seria possível através da evolução de várias tecnologias, desde camadas de rede sem fio, tipos de sensores a mecanismos de interface [1].

A computação ubíqua, também nomeada computação pervasiva ou ubicomp, é um paradigma de projeto de sistemas computacionais que desperta o interesse de pesquisadores de diversas áreas, inclusive da medicina. Assim como ocorreu para outras áreas da informática no passado, a computação ubíqua vem sendo estudada e aplicada na área da saúde. Existem hoje, na engenharia e na informática, linhas de pesquisa que tratam sobre técnicas computacionais aplicadas à saúde. Para algumas delas, foram estabelecidos termos próprios, como por exemplo *eHealth*, *mHealth*, *telemedicine*, dentre outros. Em alguns casos existe sobreposição entre os termos [2], porém, de maneira geral, rementem para aplicação de técnicas que têm como base conceitos de ubicomp.

A situação de uma entidade pode ser definida a partir da interpretação de um conjunto de elementos de contexto, relacionando-os de forma a prover alguma informação válida em um intervalo de tempo específico [3]. Dessa forma, verifica-se que a situação pode ser obtida a partir de variáveis de contexto, uma vez que sejam aplicadas técnicas de processamento oportunas. A definição de situação citada pode ser convenientemente aplicada na área da saúde, onde as entidades para as quais se define a situação podem ser pacientes, equipamentos ou até mesmo um ambiente clínico.

A aplicação de sistemas conscientes de situação na área da saúde depende essencialmente da confiabilidade das situações inferidas, evitando

• ***Douglas Adalberto Scheunemann**: Programa de Pós-Graduação em Eletrônica e Computação, Centro de Politécnico - CPoli. Universidade Católica de Pelotas - UCPEL. E-mail: douglas.tro.s@gmail.com*

que sejam induzidos erros médicos [4]. Porém, a utilização de técnicas para consciência de situação é um requisito desafiador devido à sua complexidade. Por esse motivo, muitos *middlewares* para computação ubíqua não disponibilizam funções nesta área [5]. Esses aspectos motivam pesquisas para melhorar a confiabilidade e facilitar o uso de mecanismos para consciência de situação.

O principal objetivo deste trabalho é apresentar um panorama das técnicas para consciência de situação. Primeiramente foi feita uma análise das mesmas no âmbito geral da computação ubíqua, e, após, foram identificadas a suas aplicações em trabalhos na área da saúde. Também, foi feita uma revisão das técnicas para consciência de contexto e arquiteturas baseadas em *internet of things* (IoT). Como resultado do estudo são apontadas as tendências e os principais desafios de pesquisa relacionados com a consciência de situação em aplicações na área da saúde.

O trabalho foi dividido em quatro capítulos, descritos a seguir.

No capítulo dois é feita uma revisão sobre consciência de contexto, apresentando a sua definição, técnicas de modelagem e distribuição. Ainda, são feitas considerações sobre a utilização de IoT como arquitetura para sistemas sensíveis ao contexto.

No capítulo três são apresentados aspectos relacionados com a consciência de situação, a sua definição e as principais técnicas encontradas na literatura, baseadas em especificação e aprendizado.

No capítulo quatro são analisados três trabalhos na área da saúde, que disponibilizam funções para consciência de situação.

No capítulo cinco são apresentas as considerações finais do trabalho.

2 Fundamentos sobre consciência de contexto

Uma vez definida a consciência de situação como sendo um produto do processamento de variáveis de contexto [3], primeiramente, faz-se necessário apresentar a definição de contexto e as técnicas aplicadas na coleta de suas variáveis, para então analisar as técnicas que podem ser aplicadas na inferência de situação. Nas subseções 2.1 e 2.2 são analisadas as definições e técnicas relacionadas à modelagem da consciência de contexto. Na subseção 2.3 são apresentados alguns termos e definições sobre *Internet of Things* (IoT), a qual representa um meio para coleta e distribuição de informações de contexto. Por fim, na subseção 2.4 é apresentado o resumo de uma pesquisa sobre as principais ferramentas aplicadas para consciência de contexto em aplicações ubíquas.

2.1 Definição de consciência de contexto

Em relações humanas, diálogos e ações de grupos de indivíduos são moldados pelo contexto. Um exemplo disso é a adaptação do vocabulário de um indivíduo de acordo com o ambiente. Certamente, o vocabulário aplicado em uma conversa entre amigos será diferente do que é aplicado em uma palestra formal. Esta capacidade de adaptação, que ocorre de forma praticamente implícita entre indivíduos, é difícil de ser modelada para interação humana com sistemas computacionais [6]. Desta forma, um dos objetivos do estudo e modelagem de contexto em computação ubíqua é otimizar a interface de humanos com computadores, proporcionando a exibição e interpretação de informações de acordo com o contexto.

Em [6], o contexto é definido como sendo qualquer informação utilizada para caracterizar a situação de uma entidade. Esta pode ser uma pessoa, lugar ou objeto considerado relevante para a interação entre o usuário e a aplicação. O contexto é algo que pode ser afirmado sobre uma entidade, não existindo de forma independente [7]. No trabalho [8], são definidas algumas variáveis indispensáveis para caracterização do contexto, em inglês são os cinco "W's" (*Who*, *What*, *Where*, *When* e *Why*), os quais definem:

- **Quem** (*Who*): entidade que está interagindo com o sistema ou que gerou certa informação ou evento;
- **O quê** (*What*): definição, classificação e interpretação da ação do usuário, gerando uma resposta relevante para a pergunta "O quê o usuário está fazendo?";
- **Onde** (*Where*): localização espacial do usuário ou de ocorrência de um evento;
- **Quando** (*When*): informação sobre o instante de tempo em que o evento ocorreu;
- **Por que** (*Why*): conjunto de informações e eventos que apontam o motivo que levou o usuário a realizar determinada ação.

Uma vez que estão disponíveis no domínio da aplicação as informações de contexto, a mesma pode otimizar o seu desempenho e modo de operação. A capacidade de um sistema de utilizar o contexto para fornecer ou melhorar a relevância das informações por ele geradas, é chamada de **consciência de contexto** [9].

2.2 Modelagem de contexto

Uma etapa fundamental na consciência de contexto é a modelagem ou representação do mesmo. Segundo [5] as principais técnicas de modelagem de contexto são:

- **Modelagem por identificador-valor**: tipo de modelagem baseada em pares de identificador e valor. A vantagem deste método é a sua facilidade de interpretação e gerenciamento para pequenos conjuntos de dados. Sua principal desvantagem é que informações estruturadas não podem ser representadas. No código 1 pode ser visto um exemplo de modelagem com esta técnica.

```
<Tipo=Humano>
<Nome=João>
<Temperatura=38>
<Batimento_Card=65>
<Latitude=40,71>
<Longitude=-74,00>
<Data=05/07/2015 19:04:30>
```

Código 1: Exemplo de modelagem de contexto aplicando técnica de identificador-valor.

- **Modelagem por esquemas de marcação**: neste caso as informações são representadas utilizando marcações (*tags*), sendo possível a estruturação das informações. Podem ser aplicadas, nesse caso, linguagens de marcação como XML e JSON. A seguir, é apresentado um exemplo utilizando XML.

```
<Entidade Tipo=Humano Nome=João>
  <Sensores>
    <Temperatura>38</Temperatura>
    <Batimento_Card>65</Batimento_Card>
  </Sensores>
  <Local>
    <Latitude>40,71</Latitude>
    <Longitude>-74,00</Longitude>
  </Local>
  <Data> 05/07/2015 19:04:30</Data>
</Entidade>
```

Código 2: Exemplo de modelagem de contexto aplicando XML.

- **Modelagem gráfica**: especifica as relações entre as entidades de maneira gráfica. Neste caso pode ser aplicada *Unified Modelling Language* (*UML*). Um exemplo de modelagem utilizando *UML* pode ser visto na figura 1. Esse modelo é equivalente ao exemplo em XML apresentado no código 2. A notação de entidade relacionamento pode ser convertida em uma especificação de bando de dados SQL, por exemplo.

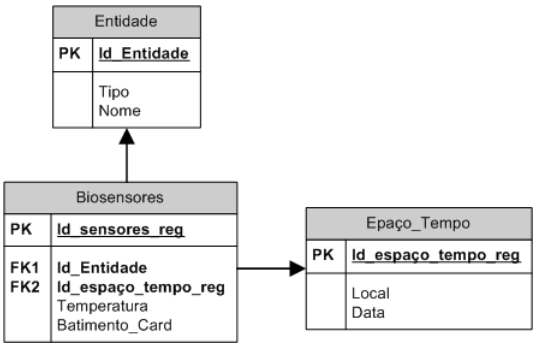

Figura 1: Exemplo de modelagem de contexto utilizando diagrama de entidade relacionamento em *UML*.

- **Modelagem baseada em objetos:** a modelagem de contexto orientada a objetos utiliza conceitos de hierarquia de classes e relações para definir os elementos de contexto. No entanto, esse tipo de modelagem é utilizado para estruturação interna em aplicações e não para compartilhamento de contexto. A propriedade de polimorfismo inerente à orientação por objetos, pode ser útil para modelar contextos de maneira genérica, permitindo a reutilização de código. Na figura 2 tem-se um exemplo deste tipo de modelagem.

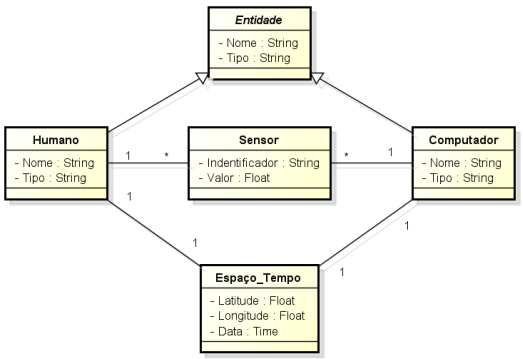

Figura 2: Exemplo de modelagem orientada por objetos, representado por diagrama de classes *UML*.

- **Modelagem baseada em lógica:** fatos, expressões e regras são utilizados para representar as informações de contexto baseadas em lógica. A principal aplicação desse tipo de modelagem é a construção de regras, geração de alarmes, restrições e preferências. Esta

técnica é aplicada geralmente em conjunto com ontologias.

- **Modelagem baseada em ontologia:** a modelagem a partir de ontologia é feita utilizando técnicas de semântica. São construídas relações entre classes e entidades, de forma semelhante à da modelagem por objetos. Porém, técnicas baseadas em ontologia, além de permitir a modelagem hierárquica, possibilitam também a construção de regras e restrições entre as classes e entidades, o que faz com que com que a modelagem de contexto possa ser completamente desassociada do domínio da aplicação [5].
  Na figura 3 pode ser vista a representação gráfica de algumas regras para inferência de contexto, utilizando modelagem por ontologia. A inferência da situação de febre de "João" apresenta dependência de valor com a temperatura do sensor vinculado com ao mesmo.

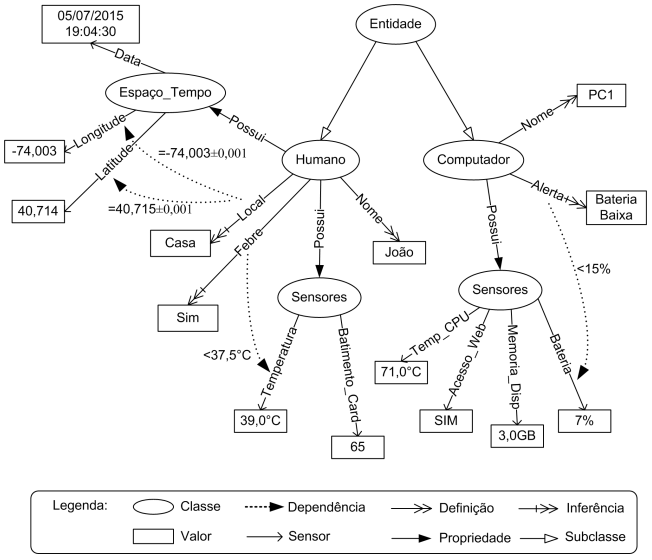

Figura 3: Exemplo de modelagem de contexto baseada em ontologia.

A ontologia é a técnica mais aplicada para modelagem de contexto [5]. Linguagens como OWL e RDF podem ser utilizadas na modelagem computacional por ontologia [5], [10]. Esta técnica é utilizada também na consciência de situação, pois permite agregar propriedades baseadas em raciocínio que podem ser representadas através de relações entre elementos [10].

A possibilidade de utilizar linguagens de marcação como OWL para modelagem de contexto baseado em ontologia, facilita a distribuição de informações entre dispositivos em cenários distribuídos. Desta forma, um servidor responsável por coletar informações de sensores, pode transmitir para os seus clientes, além dos valores das medições, relações de conhecimento, restrições e dependência dos dados. O cliente que recebe as informações representadas em OWL, pode inferir situações, executar validações e correlacionar os dados com informações de outros servidores que compartilham a mesma base semântica de ontologia.

2.3 Arquiteturas baseadas em Internet of Things

O suporte à consciência de situação em aplicações ubíquas depende de uma série de camadas de software e hardware responsáveis por coletar, pré-processar, analisar e gerar inferências. Neste cenário, podem ser utilizadas arquiteturas e aplicações que combinam elementos *desktop*, *web*, *mobile*, *Cloud Computing* e *IoT* [5].

A topologia inerentemente distribuída da computação ubíqua leva ao surgimento de cenários com grande heterogeneidade, ao mesmo tempo que requer conectividade entre os elementos. Arquiteturas *IoT* são convenientes para manipular esse tipo de sistema, pois propiciam a conectividade de diversos tipos sensores e atuadores com dispositivos controladores.

Em aplicações na área da saúde, o sensoriamento é uma etapa fundamental, através da qual é feita a captação de sinais do paciente ou sistema monitorado. Assim é esperado que *IoT* seja uma técnica presente em computação ubíqua para a saúde.

Internet of Things é um paradigma computacional utilizado para definir sistemas e ambientes inteligentes, nos quais é propiciada a interconexão de sensores e atuadores com sistemas de processamento distribuídos, promovendo o compartilhamento de informações entre plataformas através de um framework unificado [11].

Tecnicamente, a *IoT* pode ser definida como uma tecnologia criada sob redes Web, baseada em protocolo TCP/IP, permitindo a comunicação entre dispositivos físicos e virtuais que possuem identidade única, atributos e interfaces inteligentes [12]. Ainda relacionado com a conectividade,

são atribuídas características de generalização espaço-temporais à *IoT*, buscando a conectividade em qualquer lugar e a qualquer momento [5].

As redes de sensores são elementos fundamentais em *IoT*, pois é através delas que são gerados os pontos de acesso para os elementos monitorados ou controlados. Muitas aplicações possuem seu funcionamento baseado em rotinas de coleta, processamento e tomada de decisões baseadas em informações de sensores. Como resposta às decisões, podem ser gerados comandos para atuadores [5]. Na figura 4 tem-se um exemplo de organização arquitetural comumente utilizado em *IoT*.

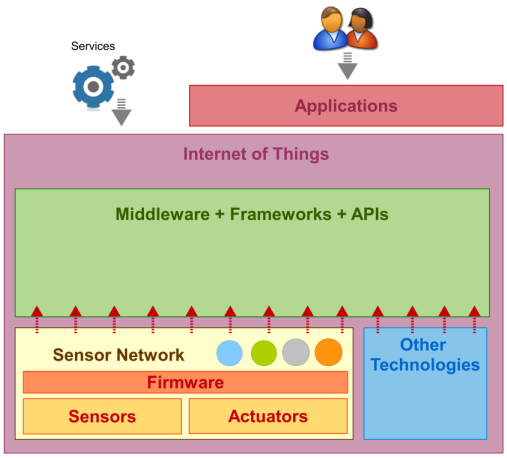

Figura 4: Modelo de camadas arquiteturais comuns em *IoT*. Fonte: [5].

Na parte superior da figura 4 são mostrados os pontos de acesso de alto nível para informações provenientes de sistemas *IoT*. Esse pode ser feito por pessoas através de aplicações em diferentes tecnologias como *desktop* e *mobile*, ou ainda através de serviços acessados por outros sistemas computacionais. Na parte inferior da figura é caracterizada a rede de sensores. Percebe-se a existência de uma camada de firmware, geralmente aplicada para prover um certo grau de inteligência e conectividade para os sensores. Na rede de sensores podem ser feitas também algumas etapas de pré-processamento dos dados coletados.

Como elemento central na figura 4, são simbolizados os *middlewares*, *frameworks* e *APIs*. Um *middleware* pode ser definido como uma camada de software, normalmente composta por várias subcamadas, posicionada entre a camada de aplicação e os elementos arquiteturais de uma dada tecnologia. O *middleware* realiza a decomposição de sistemas complexos em modelos bem definidos e de fácil interpretação, acessados através de protocolos e linguagens de programação padronizadas. Os *frameworks* e *APIs* são camadas de software utilizadas para permitir a programação e o acesso ao *middleware* [12].

Através das redes de sensores distribuídas em *IoT*, são geradas grandes quantidades de dados brutos, os quais necessitam de processamento para que informações relevantes sejam geradas. Em alguns casos, a informação final pode ser obtida a partir da fusão de dados de sensores com informação de contexto. Devido à natureza imprecisa, inerente a algumas variáveis de contexto e dados de sensores, a inferência de contexto por algoritmos clássicos, como os de lógica Booleana, pode ser inviável ou de implementação muito complexa [13], [5]. Isso faz com que outras técnicas de programação sejam aplicadas, conforme dados apresentados na subseção 2.4.

2.4 Pesquisas em consciência de contexto

Em pesquisa realizada por [14], são apontadas as técnicas para consciência de contexto mais utilizadas em trabalhos publicados nos eventos Ubicomp e Pervasive entre os anos de 2004 e 2009. Foram analisados 114 trabalhos e na figura 5 pode ser visto um resumo da contagem das técnicas aplicadas.

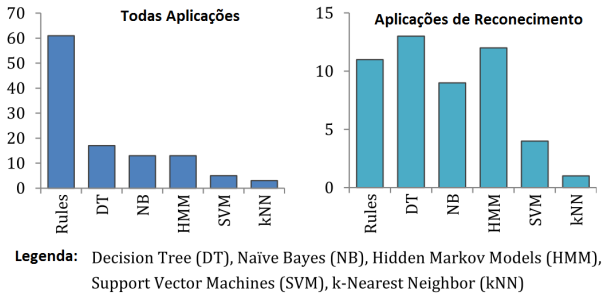

Figura 5: (Esquerda) Contagem de tipos de técnicas para consciência de contexto em 109 aplicações entre os anos de 2003 e 2009. (Direita) Contagem para tipos de classificadores para 50 aplicações de reconhecimento. Fonte: [14].

Nota-se que, quando são considerados todos os tipos de aplicações que utilizam consciência de contexto, as técnicas baseadas em regras ainda são predominantes, porém, quando trata-se de aplicações específicas para reconhecimento de contexto, pode ser destacada a aplicação das

técnicas *Decision Tree* (DT), *Hidden Markov Models* (HMM) e *k-Nearest Neighbor* (kNN). Estas são baseadas em aprendizado e algumas também são empregadas na consciência de situação, como será detalhado na subseção 3.2.

3 Técnicas para consciência de situação

A consciência de situação representa o entendimento do ambiente, incluindo parâmetros da própria aplicação, representados através do contexto e de informações das entidades em intervalos de espaço e tempo, propiciando a interpretação de significado e projeção em ações futuras [15].

A situação é genuinamente um conceito de Ontologia (parte da filosofia que estuda o ser e a existência) composta por entidades, relações intrínsecas e dependentes do contexto. Situações podem ser representadas computacionalmente através de linguagens para ontologia, nas quais, uma situação é gerada a partir da composição de relações entre propriedades de uma ou mais entidades [7].

Na figura 3 pode ser visto um exemplo de inferência para situação de uma entidade baseado na modelagem por ontologia. Trata-se da situação da instância da classe "Humano", definida como "João" que possui a variável de contexto "temperatura", com valor igual a 39,0 ºC. Como o valor da temperatura é maior do que o definido na relação de dependência, a propriedade inferida "Febre" recebe o estado "Sim", logo, pode-se afirmar: "João está com febre".

Devido à simplicidade e limitado número de variáveis, a inferência de situação para o exemplo da figura 3 pode ser facilmente modelado através de programação sequencial utilizando lógica booleana em conjunto com um modelo de ontologia. No entanto, em muitas aplicações, a quantidade de variáveis de contexto e entidades que precisam ser correlacionadas para inferior situações pode ser maior. Aspectos como incerteza de medições, disponibilidade incerta de recursos, dependência espaço-temporal e requisito de auto-aprendizado, fazem com que técnicas mais abrangentes sejam empregadas na inferência de situação.

Segundo [13], a definição de uma situação e o seu refinamento pode ser feito através de relações com outras situações ou variáveis de contexto. Em [13] são definidas ainda os principais tipos de ralações que podem ser estabelecidas entre as situações, conforme descrito a seguir:

- **generalização:** uma situação pode ser considera mais geral do que outra. Por exemplo, a situação "exercitando-se" generaliza a situação "jogando futebol";
- **composição:** uma situação pode ser composta por situações menores. Por exemplo, "jogando futebol" pode ser composto por "correndo" e "chutando";
- **dependência:** quando a ocorrência de uma situação depende da outra. Por exemplo, "comemorando o próprio gol" depende da situação "jogando futebol";
- **contradição:** duas situações que não podem ocorrer para uma mesma entidade, no mesmo local e ao mesmo tempo. Por exemplo, as situações "jogando futebol" e "jogando tênis" são mutuamente exclusivas;
- **sequência temporal:** uma dada situação pode correr antes, depois ou ainda de forma intercalada com outra. Por exemplo, a situação "discutindo com o árbitro" pode ser antecedida por "jogando futebol" e sucedida por "indo para o vestiário".

As técnicas utilizadas para prover consciência de situação aos sistemas podem ser divididas em dois grupos. Aquelas baseadas em especificação e as em aprendizado [13]. Estes grupos podem ainda ser subdivididos em alguns subconjuntos de técnicas. Algumas destas, consideradas promissoras por [13], serão apresentadas a seguir.

3.1 Técnicas baseadas em especificação

As técnicas baseadas em especificação são geralmente aplicadas em sistemas que possuem um número reduzido de sensores, para os quais os dados e relações podem ser interpretados através de regras elaboradas por especialistas [13].

3.1.1 Lógica Formal

A utilização de lógica formal para modelagem de situação depende da capacidade de modularização das situações e discretização de variáveis de contexto [13]. A lógica formal é constituída por um conjunto de operações matemáticas definidas principalmente em teoria de conjuntos e álgebra booleana. Como ferramenta de programação para

lógica formal, pode ser utilizada a linguagem Prolog [16].

No código 3 pode ser visto um exemplo de modelagem de consciência de situação, aplicando lógica formal modelada em Prolog. No mesmo, é feita a configuração do tipo de toque de um telefone celular baseada na situação do usuário. Para as situações reunião (*meeting*) e palestra (*lecture*), o toque é configurado para silencioso. Quando o usuário estiver em um restaurante, os sons de alerta do celular são habilitados.

```
required_phone_mode(quiet) :- my_current_user(E),
    in_meeting_now*>E.
required_phone_mode(quiet) :- my_current_user(E),
    in_lecture*>E.
required_phone_mode(noisy) :- my_current_user(E),
    restaurant*>E.
```

Código 3: Exemplo de modelagem de consciência de situação utilizando a linguagem Prolog. Fonte: [16].

3.1.2 Ontologia

A modelagem por ontologia já foi apresentada neste artigo no item 2.2, como uma técnica para modelagem de contexto. O mesmo exemplo foi utilizado também na introdução desta seção para descrever como as situações podem ser inferidas a partir de variáveis de contexto.

Modelos baseados em ontologias tiram proveito da capacidade de expressar relações complexas. A validade de dados é geralmente expressa através da imposição de restrições. Centrando-se sobre as relações entre entidades, ontologias são adequadas para mapear o conhecimento dentro de uma estrutura de dados facilmente utilizada e interpretada. Além disso, a ampla adoção da ontologia permite a reutilização de trabalhos anteriores e a criação de vocabulários de domínio comuns e compartilhados [8], [13].

Dentre as ferramentas disponíveis para modelagem gráfica de ontologia pode ser citado a *Protégé* desenvolvida na Universidade de Stanford. A mesma permite a geração de código para OWL, a partir de um modelo criado graficamente. Na figura 6 tem-se um exemplo de ontologia gerado nesta ferramenta. A descrição em OWL exportada para a mesma pode ser vista no código 4.

```
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
    <Declaration>
    <Class IRI="#Biosensores"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Humano"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#Batimento_Card"/>
    </Declaration>
```

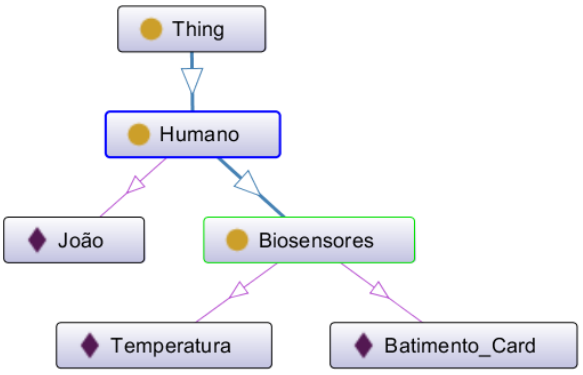

Figura 6: Exemplo de modelo de ontologia criado no software *Protégé*.

```
    <Declaration>
        <NamedIndividual IRI="#João"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#Temperatura"/>
    </Declaration>
    <SubClassOf>
        <Class IRI="#Biosensores"/>
        <Class IRI="#Humano"/>
    </SubClassOf>
    <ClassAssertion>
        <Class IRI="#Biosensores"/>
        <NamedIndividual IRI="#Batimento_Card"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Humano"/>
        <NamedIndividual IRI="#João"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#Biosensores"/>
        <NamedIndividual IRI="#Temperatura"/>
    </ClassAssertion>
</Ontology>
<!-- Generated by the OWL API (version 3.5.1) http://
     owlapi.sourceforge.net -->
```

Código 4: Código OWL gerado através do software *Protégé* para o modelo de ontologia mostrado na figura 6.

3.1.3 Lógica Espaço-temporal

Lógica espaço-temporal é uma área bem estabelecida em inteligência artificial (AI), aplicada na representação e raciocínio sobre as características e restrições de contexto e situações [17]. Em 1994 James F. Allen [17] propôs um método para representação e raciocínio lógico para modelagem de problemas que dependem de informações de espaço e tempo. Allen sugeriu que a representação de tempo e espaço por intervalos poderia ser mais interessante do que a representação por pontos específicos.

No trabalho [18] foram aplicados conceitos de modelagem espaço temporal para ambientes inteligentes. No código 5 pode ser visto um exemplo de modelagem extraído deste trabalho. Cabe ressaltar a aplicação do operador de alto nível *ANDlater*, que representa a sucessão de um evento temporal.

```
IF at_kitchen_on ANDlater
  tdRK_on ANDlater
  no_movement_detected
THEN assume the occupant has fainted
```

Código 5: Lógica espaço temporal aplicada no monitoramento de um indivíduo em um ambiente inteligente. Fonte: [18].

3.1.4 *Lógica Fuzzy*

Em aplicações baseadas em *IoT*, a quantidade e a heterogeneidade de sensores manipulados têm crescido. A incerteza de disponibilidade e diferentes faixas de precisão de sensores agregam incertezas ao sistema. Além disso, na modelagem de contexto e de situação, é necessário gerar inferências de estados a partir de dados de sensores. Em muitos casos, essa classificação não está relacionada com um valor especifico, como falso ou verdadeiro, mas sim com uma faixa de valores [13], [5]. Nesse tipo de cenário, a lógica fuzzy ou difusa tem se mostrado uma técnica promissora para tratar a incerteza de dados e mapear saídas consistentes para ao sistema [19].

A lógica fuzzy, ao contrário da lógica boolena, permite respostas intermediárias entre falso e verdadeiro, ou seja, assume a existência de infinitos valores ente 0 e 1. Assim, conceitos intrínsecos da linguagem humana para representar situações de forma aproximada podem ser modelados. Por exemplo, as afirmações: "está pouco frio" ou "está muito quente", de acordo com o contexto, podem transmitir de maneira satisfatória a informação entre os interlocutores, sem a necessidade de tratar valores numéricos exatos ou escalas de medidas.

Na computação ubíqua, a lógica fuzzy pode ser utilizada como ferramenta na inferência de situação através do mapeamento do grau de pertinência de uma variável para a ocorrência de uma situação [20]. Isso é feito através de variáveis linguísticas, as quais podem assumir valores de um determinado conjunto de termos. A definição dos conjuntos para as variáveis linguísticas pode ser feita com o auxilio de uma ferramenta para ontologia, conforme visto na seção 3.1.2.

Na figura 7 pode ser visto um exemplo do mapeamento de pertinência da variável temperatura para a variável linguística "febre", a qual pode assumir os valores: "Baixa", "Média" e "Alta". Como trata-se de uma técnica baseada em especificação, as faixas de valores são estabelecidas a partir do conhecimento de especialistas. Porém, inferências como "febre um pouco alta" para 38,5 °C e "febre muita alta" para 40 °C, podem ser facilmente geradas.

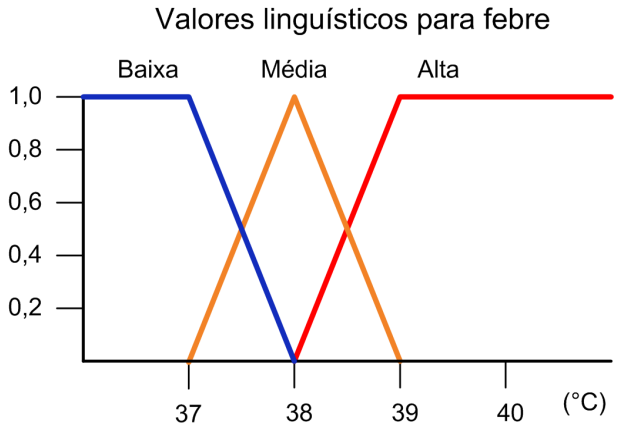

Figura 7: Valores de pertinência da variável temperatura para modelagem da variável linguística "Febre" utilizando lógica fuzzy. Adaptado de [21].

Quando uma dada situação depender de mais de uma variável, a pertinência final é dada pelo somatório da pertinência de cada variável multiplicada por um peso correspondente à sua contribuição global na situação, conforme expresso na equação 1 [20].

$$P = \sum_{i=1}^{N} w_i.\mu(x_i) \tag{1}$$

Onde: x_i representa uma dada variável contínua ou discreta, $\mu(x_i)$ representa a função de pertinência individual da variável, e w_i representa o peso da variável na situação final.

3.2 Técnicas baseadas em aprendizado

As técnicas baseadas em aprendizado permitem que o sistema gere inferências de forma automática, estabelecendo relações entre dados de sensores e estados anteriores do sistema através de um processo de aprendizado de máquina. Essas técnicas são aplicadas em sistemas em que existem muitos dados de entradas e sensores de forma que a inferência de situação baseada em especificação torna-se inviável [13]. Algumas das técnicas baseadas em aprendizado, citadas em [13], são descritas a seguir.

3.2.1 Redes Bayesianas

Rede Bayesiana é um gráfico acíclico dirigido, em que cada nó representa uma variável que pode ser discreta ou contínua, e cada arco é a relação causal entre os nós. Se houver um arco a partir de um nó A para outro nó B, então A é chamado um pai de B, o que implica que a variável B é considerado como diretamente dependente de A [13].

Na figura 8 tem-se uma exemplo de rede de classificação Bayesiana, a um partir da qual pode ser calculada a probabilidade da variável "C" em assumir um dado estado "j" em função das probabilidades condicionais P($X_i|C$), conforme equação 2.

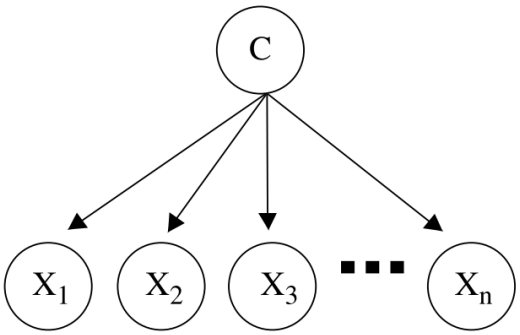

Figura 8: Estrutura de uma rede de classificação Bayesiana. Fonte: [22].

$$P(C_j|X_1...X_n) = \prod_{i=1}^{n} P(X_i|C) \qquad (2)$$

O processo de aprendizado de uma rede Bayesiana consiste em determinar o mapa de probabilidades condicionais $P(X_i|C)$ para cada variável da rede, possibilitando assim que, a partir dos valores de um dado conjunto de variáveis, a probabilidade condicional de outra possa ser determinada seguindo a equação 2 [22].

3.2.2 Modelos Ocultos Markov

Uma cadeia de Markov é um caso particular de processo estocásticos em que estados discretos podem ser identificados e que a a probabilidade de transição do estado atual para um estado futuro depende somente do estado atual [23], [24]. Na figura 9 pode ser visto um exemplo de cadeia de Markov para previsão do tempo baseada em probabilidade.

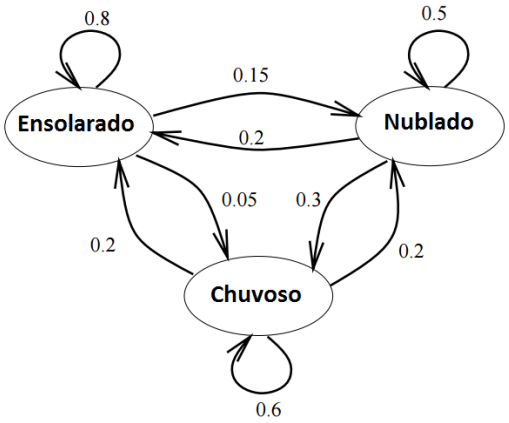

Figura 9: Exemplo de cadeia de Markov para modelagem de previsão do tempo baseada em probabilidade. Adaptado de: [23].

No Modelo Oculto de Markov é assumido que o processo em análise apresenta as propriedades estocásticas do Modelo de Markov, porém este não é diretamente observável. O mesmo pode ser inferido através de um outro conjunto de variáveis estocásticas que produzem uma sequência de observações independentes entre si [24].

3.2.3 Redes Neurais

Redes neurais artificiais são uma técnica subsimbólica, originalmente inspirada por redes de neurônios biológicos. São uma técnica alternativa para o reconhecimento de atividade. As redes neurais podem aprender automaticamente e mapear complexas relações não-lineares. Uma rede neural é composta de muitos neurônios artificiais que estão ligados entre si de acordo com uma arquitetura de rede específica. Na figura 10 pode ser visto um exemplo de classificador baseado em uma rede neural artificial.

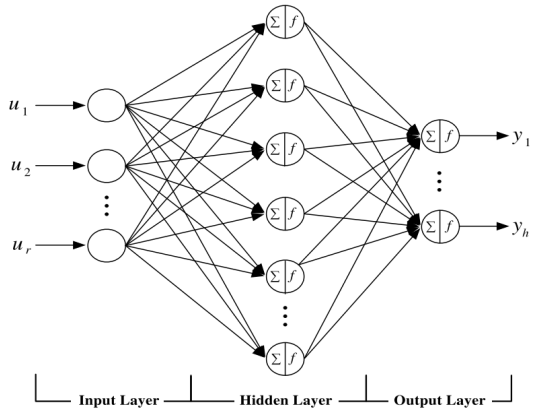

Figura 10: Exemplo de classificador baseado em uma rede neural. Fonte: [25].

O desempenho de redes neurais é afetado pela quantidade de dados de treinamento. Uma rede neural é considerada uma boa opção se há abundância de dados para treinamento e se o domínio do problema é mal compreendido para derivar um modelo aproximado [13].

4 Trabalhos na área da saúde

A consciência de situação e as técnicas de ubicomp podem ser aplicadas na área da saúde tanto em ambientes clínicos, quanto no monitoramento remoto. Em aplicações remotas podem ser destacadas o auto-monitoramento de doenças crônicas, acompanhamento remoto de pacientes e detecção antecipada de distúrbios fisiológicos [20]. O termo *eHealth* tem sido utilizado na literatura para caracterizar uma série de trabalhos relacionados com a aplicação de tecnologias de informática emergentes, como *IoT* e *Cloud Computing* para resolver problemas da área da saúde [26], [2]. Outros termos como *mHealth* e *Telemedicine* também tem sido relacionados com aplicações para saúde, porém, o termo eHealth tem sido mais recorrente nos últimos anos [27].

Segundo estudo realizado por [4], a utilização de sistemas conscientes do contexto e, baseados em reconhecimento de padrões, apresenta ainda uma série de assuntos que precisam ser estudados, principalmente com relação a questões ligadas a confiabilidade das inferências de situações geradas pelo sistema.

O autor [4] afirma que, em alguns casos a aplicação de técnicas de reconhecimento de padrões aumentou o número de erros médicos, o que pode ter contribuído para que o uso deste tipo de aplicações na área da saúde esteja abaixo do esperado.

Nas seções 4.1, 4.2 e 4.3 será apresentado um resumo de três trabalhos relacionados com a consciência de situação aplicada para questões da área da saúde. Buscar-se-á evidenciar as técnicas de inferência de situação e elementos arquiteturais empregados em cada trabalho, isto para auxiliar na determinação de tendências em aplicações desta área.

4.1 Trabalho 1: "Situation-Aware Mobile Health Monitoring"

No trabalho intitulado *"Situation-Aware Mobile Health Monitoring"* [20] é apresentada uma proposta de *framework* para consciência de situação aplicando modelos de raciocínio fuzzy voltados para área da saúde. Neste modelo, as inferências de situação são geradas a partir da fusão de informações de alto nível referentes à atividade atual do paciente com dados de sensores de pressão arterial, batimento cardíaco e temperatura.

A atividade é determinada através de um algoritmo de autoaprendizado baseado em árvores de decisão. O autor justifica que a utilização de informações da atividade do paciente possibilitam uma melhor determinação da situação do mesmo, visto que diversos parâmetros fisiológicos são afetados pela atividade realizada. Um exemplo disso é alteração do nível de batimento cardíaco enquanto o paciente faz uma caminhada.

Por se tratar de monitoramento móvel, é considerado pelo autor o uso de *smartphones* na coleta de variáveis de contexto e no processamento do algoritmo de inferência de situação. Assim, apresenta uma técnica para minimizar o uso bateria do dispositivo, reduzindo a complexidade de processamento do algoritmo de inferência de atividade do paciente. Isso é feito variando o intervalo de processamento de acordo com nível de exatidão necessário na inferência, sendo que o mesmo é função da criticidade da situação atual do paciente. Os blocos que compõem a arquitetura proposta podem ser visto na figura 11.

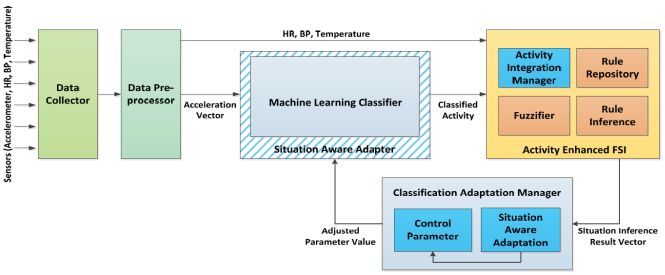

Figura 11: Modelo arquitetural para inferência de situação. Fonte: [20].

No diagrama da figura 11, o bloco "*Activity Enhanced FSI*" é responsável pelo execução da inferência de situação baseado na atividade do usuário em conjunto com seus parâmetros fisiológicos. O bloco de realimentação " *Classification Adapation Manager*" é utilizado para controlar o nível de exatidão no processamento para inferência da atividade do usuário. Esse controle é feito baseado no nível de criticidade da situação atual do paciente. Desta forma, quanto menor o grau de

criticidade, menor será o esforço computacional empregado para a determinação da atividade.

Como estudo de caso, o autor utiliza o monitoramento de pacientes com hipertensão, para os quais é feita a classificação da situação de saúde nos níveis apresentados a seguir:

- S01 – Saudável;
- S02 – Repouso Hipertensão Estágio 1;
- S03 – Repouso Hipertensão Estágio 2;
- S04 – Repouso Hipertensão Estágio 3;
- S05 – Movimento Lento Hipertensão Estágio 1;
- S06 – Movimento Lento Hipertensão Estágio 2;
- S07 – Movimento Lento Hipertensão Estágio 3;
- S08 – Movimento Rápido Hipertensão Estágio 1;
- S09 – Movimento Rápido Hipertensão Estágio 2;
- S10 – Movimento Rápido Hipertensão Estágio 3.

Para realizar a classificação entre os níveis fuzzy de S01 a S10 foram utilizados, além da atividade do paciente, os parâmetros fisiológicos pressão sistólica, pressão diastólica e batimento cardíaco. O nível S01 representa o nível menos crítico de saúde do paciente, enquanto o S10 identifica o mais crítico. Assim, para a situação S01, o processamento da inferência de atividade é desligado, visto que a mesma é irrelevante para o contexto. Conforme a situação do paciente oscila entre os níveis de S01 a A10, o intervalo de processamento da inferência de situação é reduzido, aumentando assim a sua exatidão.

Através de simulações, o autor demostra que o algoritmo de controle e processamento da inferência de atividade produz um aumento do tempo de duração da bateria do *smartphone* de 38,26%. Aponta como trabalhos futuros o teste do sistema com pacientes reais com acompanhamento por profissionais da saúde. Sugere ainda que o trabalho seja expandido, aplicando o *framework* desenvolvido em um ambiente de processamento distribuído, possibilitando, por exemplo, que um médico acompanhe remotamente vários pacientes.

O trabalho resumido nesta seção mostra a elaboração de duas funcionalidades importantes e recorrentes em sistemas ubíquos: a inferência de situação e a sensibilidade ao contexto. Para a inferência de situação foi aplicada lógica fuzzy, técnica esta que é adequada para inferência de situação em modelos probabilísticos multivariáveis.

A sensibilidade do sistema ao contexto foi baseada na realimentação do nível de criticidade da situação do paciente. Foi avaliada a contribuição da realimentação no consumo de energia do sistema, entretanto, o mecanismo de realimentação proposto pode agregar ganhos em outras funções, como por exemplo: otimizar a coleta e armazenamento de dados do paciente e reduzir a transmissão de informações redundantes, caso o sistema seja utilizado de forma distribuída.

4.2 Trabalho 2: "Ontology model-based situation and socially-aware health care service in a smart home environment"

No trabalho *"Ontology model-based situation and socially-aware health care service in a smart home environment"* [28] é apresentada uma proposta de serviço para saúde sensível à situação, distribuído e aplicado em *smart homes*. As autoras utilizam técnicas sustentadas pelos avanços em computação sensível ao contexto, potencializadas pelo uso de *smartphones*, sensores e atuadores com capacidade de conexão sem fio e acesso a Web. Apontam que existem poucos trabalhos relacionados ao uso de tecnologia de *smart homes* aplicadas na área da saúde. Destacam como principais benefícios do modelo de aplicação para saúde proposto, a oportunidade de fornecer serviços altamente customizados e disponibilizados na casa do paciente, o que propicia um melhor convívio social, melhorando a qualidade de vida do paciente e de sua família.

A ontologia foi o método de modelagem de contexto aplicado no trabalho. As autoras destacam que essa técnica tem sido utilizada em diversos trabalhos relacionados com a consciência de contexto e consideram esta tecnologia como ponto chave para o desenvolvimento do trabalho. São apresentadas quatro funções principais para a aplicação:

1) manipulação do grande número de contextos que podem ocorrer em um modelo de *smart home*;
2) utilizar dados de diversos tipos de sensores para determinar situações relacionadas com um usuário especifico;
3) através da análise de dados históricos de contexto e de relações sociais, prover recomendações para bons serviços de saúde;
4) suporte para a integração de forma transparente de uma grande variedade de dispositivos para realizar a interoperabilidade, monitoramento remoto e programação remota.

Na figura 12 pode ser vista a arquitetura proposta no trabalho, a qual foi dividida em três camadas.

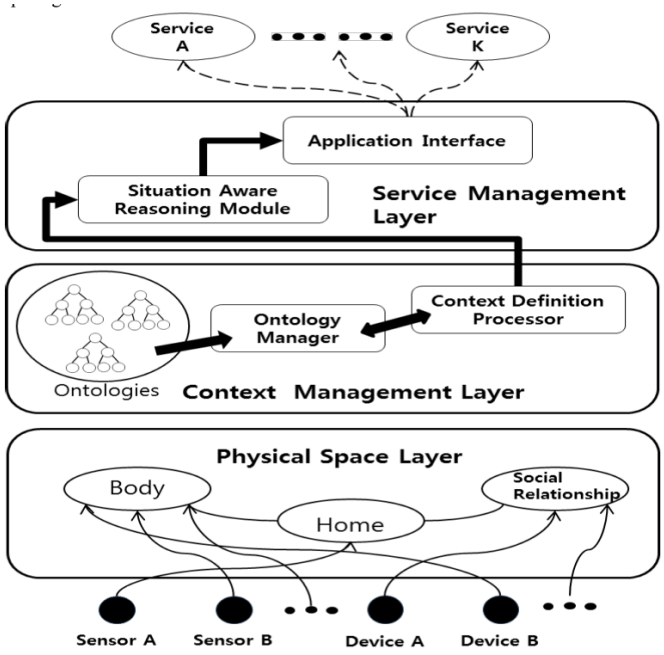

Figura 12: Arquitetura do sistema proposto no trabalho. Fonte: [28].

A primeira camada mostrada na figura 12, *Phisical Space Layer* é responsável pela manipulação e organização dos dados dos sensores. Destaca-se nessa camada que os dados referentes às relações sociais são coletados e propagados no sistema da mesma forma como é feto para dados de sensores físicos.

Na camada *Context Management Layer* é feita a determinação das informações de contexto baseadas em modelos de ontologias, que foram desenvolvidas utilizando a linguagem OWL. As regras de ontologia criadas buscam essencialmente determinar se o estado de saúde do paciente é normal. Os modelos de ontologia propostos no trabalho foram divididos de acordo com os tipos de sensores utilizados, sendo assim foram considerados: sensores biológicos, sensores do ambiente e sensores de relações sociais.

Na terceira camada *"Service Management Layer"* existe um módulo de inferência de situação e uma API para aplicações e serviços. A inferência de situação é feita a partir da interpretação das regras de raciocínio descritas no modelo de ontologia em OWL. A partir das situações inferidas, podem ser gerados via API diversos tipos de eventos, que incluem: notificações de emergência e controle de dispositivos domésticos.

O trabalho em questão apresentou uma abordagem de consciência de situação baseada em especificação, utilizando a técnica de ontologia com suporte da linguagem OWL. Cabe ressaltar que o modelo de arquitetura aplicado é baseado em *IoT*, e está de acordo com o modelo mais geral apresentado na figura 4. A possibilidade de combinar conceito de *eHealth* e *smart home* em um mesmo trabalho, parece uma aplicação promissora e que possibilita vários nichos de pesquisa relacionados com a computação ubíqua.

O trabalho proposto por [29] aplica conceitos semelhantes para o monitoramento remoto de pacientes com demência. Também, foram aplicadas neste trabalho técnicas de ontologia para inferência de situações, mas parte do processamento foi realizado através de *Cloud Computing*.

4.3 Trabalho 3: "Smart home-based health platform for behavioral monitoring and alteration of diabetes patients"

Partindo de números que apontam para um aumento nos casos de diabetes e de pessoas com fatores de risco para desenvolvimento da doença, como a obesidade, é proposto no trabalho *"Smart home-based health platform for behavioral monitoring and alteration of diabetes patients"* [30], um sistema que permita o monitoramento remoto de pacientes com diabetes, avaliando padrões fisiológicos e comportamentais.

São apresentadas cinco funções principais para a aplicação:

1) **flexibilidade e extensão:** permitir a manipulação de tipos heterogêneos de sensores, como câmeras, sensores de temperatura e sensores fisiológicos;
2) **interoperabilidade e transparência:** integração automática em diversos dispositivos, considerando ainda aspectos de interoperabilidade com dispositivos clínicos clássicos;
3) **auto-integração:** permitir o uso de dispositivos *plug-and-paly* de diversos tipos de protocolo.
4) **monitoramento e programação remota:** permitir, por exemplo, que médicos em centros clínicos acessem informações referentes à situação do paciente e façam

customizações em rotinas de coletados de sensores e monitoramento em resposta a estudos, características clínicas ou regulamentares;

5) **geração de modelo e análise:** através da utilização de algoritmos baseados em aprendizado, busca criar modelos comportamentais, reconhecimento de padrões e geração de inferências baseadas nestes padrões.

Na figura 13 pode ser visto um diagrama de como ocorre o fluxo de informação no sistema proposto no trabalho.

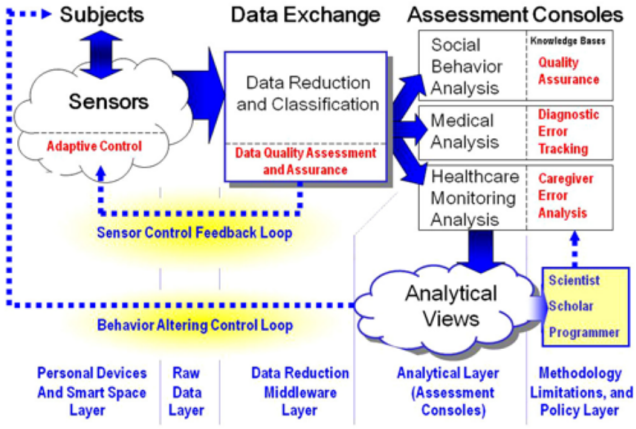

Figura 13: Arquitetura da plataforma proposta no trabalho. Fonte: [30].

Como pode ser visto na figura 13, o sistema foi organizado em cinco camadas. As duas primeiras *Personal Devices and Smart Spaces Layer* e *Raw Data Layer* estão relacionadas com as etapas de acesso e coleta de dados dos sensores. Na camada *Data Reduction Middleware Layer* é utilizado o *middleware Atlas* para criar funções e abstração de sensores, redes e hardware.

As funções agrupadas na camada *Analytical Layer* estão relacionadas com as etapas de análise das informações. Observa-se a aplicação de Modelos Ocultos de Markov para o reconhecimento de atividades. Uma segunda técnica foi aplicada para o reconhecimento de atividades comportamentais e de dieta, onde foram utilizadas imagens de vídeo e processamento digital para detectar e analisar o comportamento de mastigar. Agregando informações de espaço e tempo, esses dados podem ser utilizados para estimar a quantidade e tipo de alimento que um indivíduo consome.

Como resultado, os autores indicam que em um experimento realizado com 20 adultos, o reconhecimento de atividades aplicando Modelos Ocultos de Markov se mostrou correto em 98% dos casos. Para a detecção da atividade de mastigar, o algoritmo foi testado para as situações: mastigar com a boca aberta, mastigar com a boca fechada, falando, fazendo caretas, e nenhum movimento facial. O método se mostrou eficiente em grande parte dos casos, todavia, a mastigação foi confundida com conversa que ocorreu na mesma frequência temporal e padrão de movimento da mandíbula ao mastigar.

5 Considerações Finais

Neste artigo foram explorados aspectos gerais relacionados com a consciência de situação em aplicações ubíquas. Foi feita também uma análise de como as técnicas identificadas foram aplicadas em trabalhos relacionados com a área da saúde.

Na seção 2 foram exploradas as técnicas para consciência de contexto, logo, na seção 4 foi possível identificar a aplicação de algumas delas.

O uso de arquiteturas baseadas em *IoT*, seguindo conceitos arquiteturais mostrados na figura 4, se fez presente nos trabalhos 2 e 3, sendo que uma das propostas de expansão do trabalho 1 é a utilização de soluções distribuídas. Com base nisso é razoável considerar que a aplicação de *IoT* seria uma solução adequada também para trabalho 1.

Outro aspecto relacionado com técnicas para modelagem de contexto que foi observado é o uso de ontologias, o qual se fez presente nos trabalhos 1 e 2. Outros trabalhos como [31] e [32] reforçam que esta técnica é oportuna quando é necessário realizar o mapeamento de contexto em variáveis linguísticas. Em ambos os trabalhos a linguagem OWL foi utilizada na implementação computacional da ontologia.

Diante disso, destacam-se três tendências em modelagem e consciência de contexto:

- **utilização de *IoT* como solução arquitetural;**
- **modelagem de contexto aplicando ontologia;**
- **utilização da linguagem OWL no mapeamento e distribuição de ontologias.**

Relacionado com a consciência de situação, na seção 3 foram analisadas algumas das principais técnicas identificadas na literatura, tanto para

modelos baseados em especificação quanto em aprendizado.

No trabalho 2 o modelo de inferência de situação foi embasado em especificação de regras em um modelo de ontologia, separando-as de acordo com os contextos previstos para a aplicação. No trabalho 3 o tratamento da consciência de situação foi feito com base em aprendizado, sendo utilizada a ferramenta Modelos Ocultos de Markov.

No trabalho 1 foi utilizada uma solução híbrida para a inferência de situação. A classificação de atividade foi feita por meio de um algorítimo de árvores de decisão, baseado em aprendizado. A situação final do paciente foi definida através de lógica fuzzy, baseada em especificação.

Conforme os trabalhos analisados e por meio de afirmações quanto a problemas no reconhecimento de padrões enfatizados em [4], pode-se considerar que a utilização de uma solução híbrida apresenta algumas vantagens, visto que permite combinar a especificação de regras de alto nível estabelecidas pelo conhecimento de profissionais com algoritmos especializados de aprendizado, aplicados em alguns blocos específicos, como é o caso da inferência de atividade do paciente.

De acordo com os dados apresentados neste trabalho, pode ser sugerido como um estudo futuro uma comparação entre os métodos de consciência de situação, buscando, através de simulação ou testes práticos, apontar as principais vantagens de cada método, traçando assim um perfil das aplicações mais adequadas para cada um deles.

Referências

[1] Mark Weiser. The Computer for the 21 century. *Scientific American*, 265(3):94–104, 1991.

[2] Emiel a Boogerd, Tessa Arts, Lucien Jlpg Engelen, and Tom H van de Belt. "What Is eHealth": Time for An Update? *JMIR Research Protocols*, 4(1):e29, 2015.

[3] Ana Marilza Pernas. *Sensibilidade à situação em sistemas educacionais na web*. PhD thesis, Universidade Federal do Rio Grande do Sul, 2012.

[4] C Chibelushi. An Overview and Concerns on IoT in Healthcare Applications ; The Role of AI and Pattern Recognition. *IJICTT*, 1(1):39–52, 2014.

[5] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context Aware Computing for The Internet of Things : A Survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, X(X):1–41, 2013.

[6] Anind K Dey and Gregory D Abowd. Towards a Better Understanding of Context and Context-Awareness. *Computing Systems*, 40(3):304–307, 1999.

[7] Patrícia Dockhorn Costa, Giancarlo Guizzardi, João Paulo a Almeida, Luís Ferreira Pires, and Marten Van Sinderen. Situations in conceptual modeling of context. *Proceedings - 2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOCW2006*, (i), 2006.

[8] Paolo Bellavista, Antonio Corradi, Mario Fanelli, and Luca Foschini. A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys*, 44(4):1–45, 2012.

[9] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58, 2000.

[10] Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. An Ontology-based Context Model in Intelligent Environments. *in Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 270–275, 2004.

[11] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[12] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Sergio Gusmeroli, Harald Sundmaeker, Alessandro Bassi, Ignacio Soler Jubert, Margaretha Mazura, Mark Harrison, Markus Eisenhauer, Pat Doody, Friess Peter, Guillemin Patrick, Gusmeroli Sergio, Bassi Harald, Sundmaeker Alessandro, Jubert Ignacio Soler, Mazura Margaretha, Harrison Mark, Eisenhauer Markus, and Doody Pat. Internet of Things Strategic Research Roadmap. 2009.

[13] Juan Ye, Simon Dobson, and Susan McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1):36–66, 2012.

[14] Brian Y. Lim. Improving Understanding and Trust with Intelligibility in Context-Aware Applications. (May):454, 2012.

[15] Mica R. Endsley. Measurement of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):65–84, 1995.

[16] S W Loke. Logic Programming for Context-Aware Pervasive Computing. (2):1–7, 2004.

[17] James F. Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[18] Juan Carlos Augusto, Jun Liu, Paul McCullagh, Hui Wang, and Jian-Bo Yang. Management of Uncertainty and Spatio-Temporal Aspects for Monitoring and Diagnosis in a Smart Home. *International Journal of Computational Intelligence Systems*, 1(4):361, 2008.

[19] Manoj Sharma. Fuzzy Logic Based Handover. 3(4):21–29, 2012.

[20] Tuan Minh Delir Haghighi, Pari and Perera, Averi and Indrawan-Santiago, Maria and Huynh. Situation-Aware Mobile Health Monitoring. *ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)*, pages 248—-256, 2014.

[21] Geraldo José Carvalho Santos. *Lógica Fuzzy*. PhD thesis, Universidade Estaduas de Santa Cruz, 2003.

[22] Panu Korpipää, Miika Koskinen, Johannes Peltola, Satu Marja Mäkelä, and Tapio Seppänen. Bayesian approach to sensor-based context awareness. *Personal and Ubiquitous Computing*, 7(2):113–124, 2003.

[23] Eric Fosler-lussier. Markov Models and Hidden Markov Models: A Brief Tutorial. *Ca Tr-98-041*, 1198(510):132–141, 1998.

[24] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition, 1989.

[25] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, 29(16):2213–2220, December 2008.

[26] David K Ahern, Jennifer M Kreslake, and Judith M Phalen. What is eHealth (6): perspectives on the evolution of eHealth research. *Journal of medical Internet research*, 8(1):e4, January 2006.

[27] Rashid Bashshur, Gary Shannon, Elizabeth Krupinsk, and Jim Grigsby. The Taxonomy of Telemedicine, 2011.

[28] Haesung Lee and Joonhee Kwon. Ontology model-based situation and socially-aware health care service in a smart home environment. *International Journal of Smart Home*, 7(5):239–250, 2013.

[29] Ederson Bastiani, Giovani R. Librelotto, Leandro O. Freitas, Rafael Pereira, and Miguel B. Brasil. An Approach for Pervasive Homecare Environments Focused on Care of Patients with Dementia. *Procedia Technology*, 9:921–929, 2013.

[30] Abdelsalam Helal, Diane J Cook, and Mark Schmalz. Smart home-based health platform for behavioral monitoring and alteration of diabetes patients. *Journal of diabetes science and technology (Online)*, 3(1):141–148, 2009.

[31] Pierpaolo D'Urso, Livia De Giovanni, and Paolo Spagnoletti. A fuzzy taxonomy for e-Health projects.

[32] Patrice C. Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas. A Possibilistic Approach for Activity Recognition in Smart Homes for Cognitive Assistance to Alzheimer's Patients. *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58, 2011.

Douglas Adalberto Scheunemann Mestrando do Programa de Pós-Graduação em Engenharia Eletrônica e Computação da Universidade Católica de Pelotas, na linha de pesquisa Computação Ubíqua Sistemas Eletrônicos e Computacionais. Também é pós-graduado *Lato Sensu* (2014) em Engenharia de Produção pela Uninter . Possui Graduação (2010) em Engenharia Eletrônica pela Universidade Católica de Pelotas. Tem experiência na área de projeto de equipamentos eletromédicos com ênfase nas etapas de processamento digital de sinais e criação de *middlewares*.