



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

**UM ESTUDO SOBRE APLICAÇÕES
SENSÍVEIS A CONTEXTO PARA
DISPOSITIVOS MÓVEIS**

Lucas de Queiroz Lins Martins

TRABALHO DE GRADUAÇÃO

Recife

13 de Julho de 2017

Universidade Federal de Pernambuco
Centro de Informática

Lucas de Queiroz Lins Martins

**UM ESTUDO SOBRE APLICAÇÕES SENSÍVEIS A CONTEXTO
PARA DISPOSITIVOS MÓVEIS**

*Trabalho apresentado ao Programa de Graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Bacharel em Ciência da Com-
putação.*

Orientador: *Prof. Carlos André Guimarães Ferraz*

Recife

13 de Julho de 2017

Dedico este trabalho aos meu avós Vanilda Queiroz e João Anacleto Lins; à minha mãe, que sempre colocou a educação dos filhos em primeiro lugar; aos meus irmãos e aos meus tios e tias.

AGRADECIMENTOS

A minha mãe, minha tia Gláucia e meu tio Luciano por sempre me questionarem sobre esse trabalho e pelo incentivo dado para a conclusão do curso;

A Mikaela que me aperriou bastante para que eu escrevesse logo esse trabalho, o que fez com que eu escrevesse as primeiras linhas;

O meu orientador, Carlos Ferraz, por todo apoio e por resgatar o interesse na conclusão desta monografia;

Meu irmão, Joao Gabriel, por ter aberto o caminho para a computação;

Agradeço o apoio de todos os outros não citados aqui.

Ubiquitous computing names the third wave in computing, just now beginning.

First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop.

Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.

—MARK WEISER

RESUMO

No cenário atual de popularização de dispositivos móveis e com o aumento da capacidade sensorial e comunicativa desses aparelhos, nos aproximamos de uma larga adoção de aplicações móveis sensíveis a contexto. Com o objetivo de fazer um levantamento e análise dos atuais sistemas sensíveis a contexto focados em dispositivos móveis, este trabalho usou como base artigos que estudaram a sensibilidade a contexto numa época em que a mobilidade era limitada e também artigos recentes sobre o tema. Inicialmente foi abordado o que há de mais novo em arquitetura e infraestrutura para o desenvolvimento de aplicações capazes de trabalhar com uma grande massa de dados e de garantir a privacidade das informações do usuário, temas estes de grande relevância atualmente. Posteriormente é feita uma análise de aplicações móveis, mostrando o que tem sido feito recentemente na área com modelos não tradicionais de sensibilidade a contexto. Em seguida alguns obstáculos para o desenvolvimento da área são apresentados juntos a desafios que uma vez transpostos podem nos colocar cada vez mais perto da era da computação ubíqua.

Palavras-chave: Sensibilidade a Contexto, Computação Sensível a Contexto, Computação Ubíqua, Computação Pervasiva,

ABSTRACT

In the current scenario of popularization of mobile devices and with the increase of the sensorial and communicative capabilities of these devices, we are approaching a wide adoption of context-aware mobile applications. With the objective of making a survey and analysis of current context-aware systems focused on mobile devices, this paper made use of articles that studied context-awareness in a time when mobility was limited as well as recent articles on the subject. Initially, an overview of the newest architectures and infrastructures for the development of applications capable of dealing with a large amount of data and also able to ensure user's information privacy, topics of great relevance at the present time. Subsequently an analysis of mobile applications is done, showing what has been done recently in the area with non traditional context-awareness models. Then some obstacles to the development of the area are presented together with challenges that once transposed can put us ever closer to the era of ubiquitous computing.

Keywords: Context Awareness, Context-Aware Computing, Contextually Aware Applications, Ubiquitous Computing, Pervasive Computing

SUMÁRIO

Lista de Figuras	x
Capítulo 1—Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Organização do documento	2
Capítulo 2—Revisão de literatura	4
2.1 Computação ubíqua	4
2.2 Sensibilidade a contexto	5
2.3 Definição de contexto	6
Capítulo 3—Arquitetura e Infraestrutura	9
3.1 ALAGAR	10
3.2 ALPACA	18
Capítulo 4—Exemplos e Aplicações	21
4.1 Sugestões de acordo com o nível de agitação	22
4.2 Troca de informações por gestos	23
4.3 Direcionamento de conteúdo baseado em perfil	25

Capítulo 5—Problemas e Desafios	27
5.1 De Sensores ao Contexto	27
5.1.1 Percepção limitada	27
5.1.2 Compreendendo os dados	28
5.1.3 Erros de percepção de contexto	29
5.1.4 Capacidade de compreensão	30
5.2 Segurança e Privacidade	30
Capítulo 6—Conclusão	34

LISTA DE FIGURAS

1.1	Crescimento do número de dispositivos por pessoas com o passar do tempo. Adaptado de [1]	2
3.1	Modelo de execução de sistema sensível a contexto. Adaptado de [2] . . .	11
3.2	Arquitetura de sistema sensível a contexto. Adaptado de [2]	14
3.3	Arquitetura em camadas para a proteção de dados sensíveis. Adaptado de [3]	20
3.4	Representação de uma variável de contexto com diversos níveis de ofuscação. Adaptado de [3]	20

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

Nos últimos anos o mercado *mobile* cresceu consideravelmente, mais de 46% de 2013 a 2015[4] e estimava-se que em 2016 esse número aumentasse mais 12.6%, ultrapassando 2 bilhões de usuários de *smartphones* no mundo[4]. Quando se considera todos os outros aparelhos móveis, de dispositivos vestíveis a computadores que podem ser usados como tablets, esse número é muito maior, chegando a superar, em número de dispositivos, a população mundial[5].

Ao mesmo tempo as pessoas estão se tornando cada vez mais *mobile-centric*, deixando de fazer certas atividades que agora podem ser feitas com mais facilidade nesses dispositivos a qualquer momento e lugar como: a realização de operações bancárias, que gera uma diminuição no fluxo de pessoas em agências; o acesso a notícias, que é feito por redes sociais e portais de conteúdo; e a troca de mensagens instantâneas, que é realizada através de aplicativos presentes praticamente em todos os celulares.

Devido ao aumento do tempo que as pessoas passam utilizando esses dispositivos e também à uma série de outros fatores como: a característica móvel desses aparelhos, os diversos sensores embarcados capazes de mensurar variáveis de ambiente e outros dados atreladas ao mesmo; é gerada uma quantidade imensa de informação que pode ser utilizada para diversas aplicações. Estudos recentes preveem um aumento no tráfego global de dados móveis - que está diretamente ligado à geração de informação - de 10 vezes nos próximos 3 anos, alcançando 292 Exabytes anuais em 2019[6].

Nesse cenário, temos a popularização de dispositivos móveis (gráfico 1.1) capazes de perceber o ambiente por meio de diversos sensores, o que nos gera uma gama de

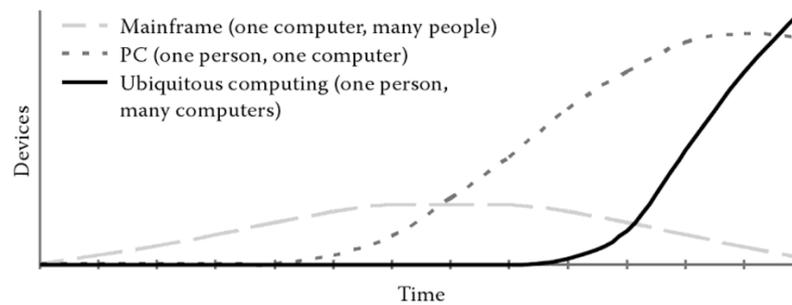


Figura 1.1 Crescimento do número de dispositivos por pessoas com o passar do tempo. Adaptado de [1]

informações nunca tida antes. A partir disso, surge a oportunidade de desenvolver aplicações que possam entender o ambiente no qual o usuário está inserido e prover serviços ou funcionalidades que se adaptem melhor às necessidades do mesmo.

1.2 OBJETIVOS

Esse trabalho se propõe a fazer um levantamento e análise dos sistemas sensíveis a contexto atuais focados em dispositivos móveis, estudar modelos de arquitetura utilizados na construção desses sistemas e fazer uma análise dos problemas e desafios que a área encontra atualmente.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Inicialmente é feito um apanhado geral do que é a computação ubíqua, computação sensível a contexto e termos correlatos no capítulo 2, introduzindo ao leitor um novo modelo computacional que é a própria definição da terceira e nova era da computação.

No capítulo 3 é feita uma análise de como as aplicações sensíveis a contexto podem ser construídas. Duas arquiteturas são estudadas com esse fim, a primeira mostra uma infraestrutura bem genérica capaz de ser utilizada na construção de aplicações em diversas áreas e a outra, focando em privacidade da informação, traz uma arquitetura em camadas na qual a informação perde precisão a medida que passa por essas divisões. Na estrutura

proposta o usuário tem total controle sobre a informação que é compartilhada com as aplicações construídas em cima desse *framework*.

No capítulo 4, é feito um esforço para desmitificar as aplicações sensíveis a contexto, com exemplos simples e práticos e, por fim, três aplicações são analisadas. A primeira utiliza sensores de movimento que aliados com uma camada de inferência pode, a partir do nível de agitação do usuário, fazer sugestões ou se adaptar ao ambiente. A segunda aplicação mostra como é possível trocar informações com gestos. Uma pulseira desenvolvida para tal finalidade ou até um *smartwatch* podem ser usados para fazer transferência de dados. Nesse caso, ao se cumprimentar um outro usuário o sistema faz a sincronia dos gestos e com isso compartilha as informações com a pessoa correta. A terceira e última aplicação faz uso de uma robusta infraestrutura e mostra como é possível utilizar o contexto histórico do usuário para fazer o direcionamento de conteúdo baseado em perfil.

No capítulo 5, alguns pontos que dificultam o desenvolvimento das aplicações sensíveis a contexto são citados. Inicialmente é abordado o tema de inferência de contexto a partir de sensores e alguns problemas relacionados ao mesmo como percepção limitada, compreensão dos dados, erros de percepção e capacidade de compreensão do sistema pelo usuário. No segundo tópico desse mesmo capítulo é mostrada a problemática da segurança da informação e da privacidade do usuário num mundo repleto de dispositivos móveis capazes de sentir o ambiente e conectados à internet.

Por fim no capítulo 6, a conclusão do trabalho, mostra que para o desenvolvimento e crescimento da computação ubíqua e da computação sensível a contexto é necessário que outras áreas da computação que são essenciais para essas se desenvolvam, servindo como base para que possamos chegar, de fato, na terceira e nova era da computação.

CAPÍTULO 2

REVISÃO DE LITERATURA

Neste capítulo será feito um breve apanhado sobre 1) o conceito de computação ubíqua e seus termos correlatos, 2) sistemas sensíveis a contexto e por fim 3) o que é contexto e como se pode categorizá-lo.

2.1 COMPUTAÇÃO UBÍQUA

Em 1988, Mark Weiser criou o termo *Computação Ubíqua* [1] vislumbrando um futuro no qual a computação estaria embarcada em diversos utensílios e equipamentos utilizados no dia-a-dia, tornando-se presente em todos os lugares e a todo momento. Esse termo foi se tornando cada vez mais amplo com outras publicações na área como em Weiser [7] que cita o termo *Computação Pervasiva*, que se refere à integração transparente de dispositivos computacionais utilizados no dia-a-dia, em Han [8] o termo *Ambient Intelligence* é utilizado para representar essa capacidade que, em outras palavras, o ambiente irá adquirir e em Greenfield [9] o termo *Everyware* aparece tornando o conceito ainda mais amplo.

Cada termo utilizado para descrever a computação ubíqua enfatiza uma propriedade diferente desse conceito. Se nos atermos aos objetos utilizados, podemos chamá-la de computação física ou internet das coisas (*IoT*), se o foco for nas funcionalidades adicionadas ao ambiente, podemos chamá-la de *Ambient Intelligence* e dessa forma podemos utilizar outros termos a medida que mudamos o foco na propriedade que está sendo estudada. Ao invés de propor uma única definição para computação ubíqua e para os termos relacionados, em Posland [10] é proposta uma taxonomia de propriedades para a computação ubíqua, pela qual é possível descrever diversos tipos de sistemas e aplicações

ubíquas, evitando a necessidade de criação de novas nomenclaturas.

Em termos gerais o conceito descreve um ambiente no qual os computadores ficassem em segundo plano, dando lugar para as aplicações e funcionalidades providas pelos mesmos, um ambiente que não fosse necessário todo um conhecimento técnico específico para interagir com esses dispositivos, mas onde a integração e interoperabilidade fossem tão simples que não seriam notadas pelos usuários finais, que só perceberiam o resultado desses sistemas.

2.2 SENSIBILIDADE A CONTEXTO

Existem diversos desafios para que a computação ubíqua torne-se realidade, um deles é a construção de sistemas capazes de identificar em que contexto eles estão inseridos e, principalmente, capazes de reagir a essa informação. Esses sistemas são chamados de sistemas sensíveis a contexto.

O termo sensibilidade a contexto (*context-awareness*) foi utilizado pela primeira vez em Schilit e Theimer [11], para se referir a sistemas capazes de prover informações relevantes de contexto e serviços para usuários e aplicações. Um ano depois em Schilit et al [12] a definição foi melhorada e o termo foi definido como a capacidade que sistemas tem de se adaptar ao contexto. Várias outras definições similares foram feitas como em Dey 2000 [13], Ryan 1998 [14] e outros.

Posteriormente foram criadas outras definições para esses sistemas que tornam o conceito mais amplo como em Chen e Kotz 2000 [15], que os define como sistemas que podem inferir um estado interno ou externo utilizando sensores e monitores, podendo variar suas funcionalidades de acordo com o novo estado.

Para deixar um único entendimento sobre a sensibilidade a contexto, pode-se resumir-la como a capacidade que alguns sistemas possuem de inferência e adaptação ao contexto.

2.3 DEFINIÇÃO DE CONTEXTO

Para se poder trabalhar com sistemas sensíveis a contexto é necessário, primeiramente, definir o que é o contexto. Segundo o Michaelis, dicionário de língua portuguesa, Contexto é o conjunto de circunstâncias inter-relacionadas de cuja tessitura se depreende determinado fato ou situação; circunstâncias, conjuntura, situação[16]. Essa definição é um tanto genérica, mas serve para compreender o conceito de forma ampla.

Buscando um entendimento mais preciso e visando aplicar tecnicamente a definição ao desenvolvimento de sistemas sensíveis a contexto é mais interessante olhar definições prévias do termo. Em Dey 99 [17] é dada a seguinte definição para contexto: “Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação em si.”

Em Schilit et al. [11], o autor tenta ser mais objetivo e define três aspectos importantes do contexto: onde o usuário está, com quem o usuário está e que recursos estão por perto. O contexto é definido como sendo o ambiente de execução em constante mudança. No artigo o ambiente é dividido nos seguintes itens:

Ambiente computacional processadores disponíveis, dispositivos acessíveis para entrada de dados pelo usuário e telas para exibição de informação, capacidade da rede, conectividade e custos computacionais

Ambiente do usuário localização, grupo de pessoas por perto e situação social

Ambiente físico iluminação, nível de ruído e temperatura.

Apesar dos esforços em se ter uma definição formal, os exemplos dados acima não são suficientes para se criar uma representação real de contexto. Para o desenvolvimento de sistemas sensíveis a contexto é necessário que essa representação seja objetiva e de forma computacional. Com esse intuito deve ser feita uma categorização do termo.

Não existe um consenso quanto a categorização de contexto, em Morse et al. (2000) é citada uma classificação bem ampla que separa o contexto em 6 dimensões: 1) Quem (faz

uso do sistema), 2) O que (para denotar o tipo do serviço), 3) Como (o serviço precisa de prestado), 4) Onde (prestar o serviço), 5) Quando (prestar o serviço), e 6) Por que (propósito do serviço).

Visando um melhor entendimento desse trabalho o contexto será classificado em 4 grupos: Físico, Social, de Temperamento e de Interação.

Físico O contexto físico é o tipo de contexto mais comum, de forma simplificada ele reflete as variáveis de ambiente onde o usuário se encontra como temperatura, localização, data e hora, por exemplo. Aparentemente ele não parece ser nada mais que um conjunto de parâmetros de um determinado lugar mas pode-se inferir informações mais ricas e próximas do que realmente se entende como contexto. A partir da localização do usuário, pode-se inferir a língua local utilizando-se uma base de países, ao adicionar à essa variável o dia e hora mais as informações de um calendário é possível inferir em que estação do ano o usuário se encontra e a partir disso se ele deve se preparar para o frio ou não. O conjunto dessas informações pode de fato enriquecer o contexto.

Segue abaixo uma lista não exaustiva dos sensores/serviços que podem ser usados para agregar valor ao contexto físico.

- Data e Hora
- Calendário
- Localização
- Clima e Tempo
- Pressão e Temperatura
- Proximidade

Social O contexto social representa informações de relacionamento entre pessoas e a importância delas para o usuário. É considerado um dos contextos mais importantes e um dos fatores para isso é o fato dele ser pouco explorado. O que se deseja ter conhecimento nesse caso é quais pessoas se relacionam com usuário, qual o grau de

parentesco entre essas pessoas, qual o trabalho delas e se elas estão próximas do usuário em questão. Ele também pode ser mais amplo e tratar não só do contexto social do ambiente no qual o utilizador do sistema está inserido mas também da cidade ou país em que ele se encontra, entendendo que no momento existe uma crise política ou econômica. Um exemplo simples desse tipo de contexto pode ser o sistema perceber que o usuário está na presença de sua/seu namorada(o) e sugerir que eles façam algo juntos. Os sensores/serviços que pode ser utilizados para a inferência desse contexto são:

- Redes Sociais
- Sensores de presença
- *Feeds* de notícias

de Temperamento O contexto de temperamento trata do humor do usuário. A informação que tenta-se inferir nesse caso é o estado emocional do mesmo. O usuário pode estar alegre, triste, calmo, estressado etc. Este tipo de análise não é fácil de ser feita mas pode ser inferida a partir de heurísticas sobre as ações da pessoa como também pela coleta de informações de sensores vestíveis (conectados ao corpo) para se monitorar batimento cardíaco, pressão sanguínea e outros parâmetros.

de Interação O contexto de interação reflete como o usuário está se comunicando e interagindo com o sistema em si ou com o resultado do mesmo. O usuário pode estar interagindo através de um celular ou sendo impactado por um anúncio ao assistir TV. Esse tipo de contexto é importante para fazer a comunicação com o utilizar do sistema pelo canal mais adequado. Se o mesmo encontra-se utilizando o celular não faz sentido ele ser notificado em outro aparelho.

CAPÍTULO 3

ARQUITETURA E INFRAESTRUTURA

A criação de aplicações sensíveis a contexto esbarra em vários desafios. Obter dados através de sensores ou agir sobre atuadores para gerar mudanças no ambiente, o que pode ser traduzido como a sensibilidade desses sistemas, é de certa forma a parte simples da criação dessas aplicações. Os maiores desafios aparecem justamente quando se foca no contexto e no suporte necessário para a troca de conhecimento e tomada de decisão que esses sistemas devem possuir: a partir de que variáveis de ambiente físicos ou virtuais, e principalmente de que formas, se pode inferir o contexto? Como se pode representar o contexto inferido? Como, a partir de um contexto já inferido e representado, se pode criar o conhecimento necessário para a aplicação de regras que irão gerar as ações necessárias? Com que dispositivos essas informações serão compartilhadas? Como elas serão distribuídas? É sobre esses e outros desafios que arquiteturas, designs e infraestruturas de sistemas são propostas visando facilitar a criação e o desenvolvimento dessas aplicações capazes de ter conhecimento sobre o ambiente à sua volta e de se adequar ao mesmo de forma desejada.

Em Dey [18] são citadas três razões que ao serem consideradas, excluindo outras inúmeras, são suficientes para tornarem difícil a capacidade de lidar com contextos: (1) a inexistência de princípios de orientação para dar suporte à boas práticas de engenharia de software; (2) a falta de abstrações que projetistas se deparam ao pensar em contexto; e (3) a complexidade de sistemas distribuídos, necessários para criação de boa parte das aplicações sensíveis a contexto. Pode-se entender essas razões como os três princípios básicos a serem trabalhados para se criar uma infraestrutura capaz de possibilitar o desenvolvimento ágil de aplicações sensíveis a contexto.

Em Zaidenberg [19] são definidas algumas necessidades a serem endereçadas por tais

arquiteturas 1) heterogeneidade: ambientes ubíquos ou inteligentes tem que integrar plataformas diferentes, computadores, no termo amplo da palavra, rodando diversos sistemas operacionais como Android, iOS, Windows, MacOS, Linux etc. 2) comunicação: só é possível integrar essa diversidade de sistemas com uma arquitetura distribuída para a computação ubíqua. Isso leva à necessidade de um protocolo de comunicação que todos esses dispositivos espalhados pelo ambiente possam entender ou até mesmo a uma pluralidade de protocolos. 3) descoberta: Além da comunicação, a diversidade de aparelhos implica na necessidade de criação de um mecanismo de descoberta de dispositivos. Já que para se comunicarem eles tem que, em tempo real, achar todos os sensores e atuadores locais acessíveis num determinado momento.

A partir das diferentes necessidades apresentadas é possível perceber que apesar de muitos aspectos terem sido trabalhados nos últimos anos, não há uma abordagem única utilizada pela indústria ou academia que consiga endereçar todos os pontos citados. O objetivo desse capítulo é apresentar algumas arquiteturas e modelos utilizados para solucionar tais problemas e proporcionar um melhor entendimento de como uma aplicação sensível a contexto pode ser construída.

3.1 ALAGAR

Em ALAGAR [2] é definida uma arquitetura baseada em componentes para a criação de um *framework* sensível a contexto que lida com a definição de contexto propriamente dita, sensibilidade, regras de reação, interações e também com a capacidade de adaptação que é necessária às aplicações que podem ser criadas com o mesmo. O *framework* definido é suficientemente genérico para atender aos requisitos de aplicações de diversos domínios como transporte, saúde, segurança e redes sociais. Além disso, foi desenvolvido utilizando-se princípios de engenharia de software teóricos e práticos aplicados na implementação de sistemas robustos.

De acordo com os autores, pode-se resumir o *framework* proposto como um conjunto de métodos e funcionalidades para resolver os problemas listados abaixo:

1. Gerenciamento de sensores para adquirir conhecimento sobre contexto do ambiente

no qual o usuário se encontra e representação e gerenciamento das informações de contexto.

2. Definição de informação semântica e regras para a inferência de situações baseadas nas informações de contexto.
3. Definição de adaptações às situações contextuais e gerenciamento de políticas para regular tais adaptações.
4. Gerenciamento de atuadores para fazer adaptações no ambiente do usuário.

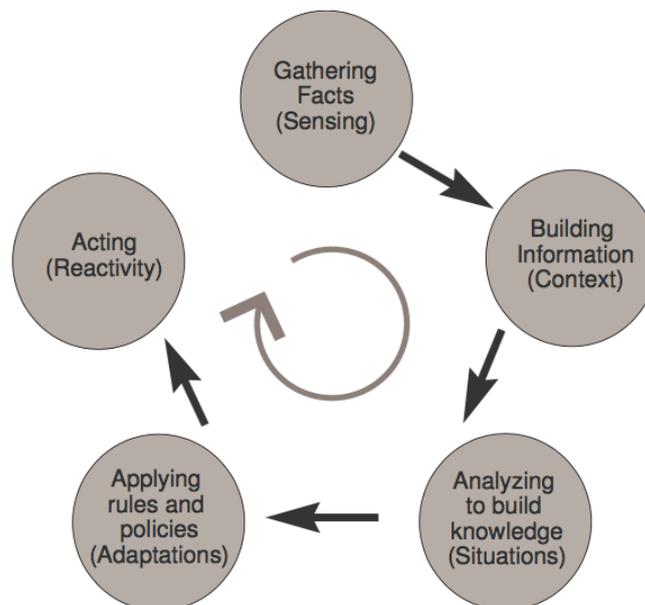


Figura 3.1 Modelo de execução de sistema sensível a contexto. Adaptado de [2]

Na figura 3.1 pode-se ter uma visão do modelo de execução do *framework*: 1) *Sensing*: coleta de dados e fatos de diversos dispositivos e sensores instalados no ambiente, 2) *Building Information*: Extração e combinação de dados relevantes para a construção dos contextos relativos ao momento atual. A construção é baseada em definições de contexto previamente definidas, 3) *Building Knowledge/Awareness*: Análise dos contextos gerados para construir as situações do momento. Essa construção também é baseada em definições

de situações previamente definidas no sistema, 4) *Adapting*: Dedução de adaptações válidas baseadas nas situação atuais. Que pode ser definida como a geração de ações que podem ser tomadas a partir das situações encontradas. E, por último, 5) *reacting*: Atuar sobre o conhecimento obtido do ambiente executando ações de acordo com as especificações. As informações do ambiente, como mostra a figura 3.1, trafegam pelas diversas camadas da arquitetura onde durante a execução do sistema são transformadas de uma forma que possibilita o correto processamento em cada etapa.

O artigo [2] parte de uma definição matemática de contexto, inicialmente citada em Wan [20] que o define como sendo um conjunto de pares que representam um aspecto específico do ambiente. Esses pares são compostos de uma dimensão e um valor associado a mesma, como pode-se visualizar no exemplo de um seminário no qual pode-se ter um apresentador falando em uma sala sobre um assunto específico num horário predeterminado. Esse contexto poderia ser representado como o seguinte conjunto de pares: [APRESENTADOR: *João Gabriel*, LOCAL: *Sala-1*, TÓPICO: *A nova era da computação*, HORÁRIO: *10hrs*].

A definição formal mostrada em Wan [20] é de uma relação, conjunto de pares ordenados, de (d, x) onde d é uma dimensão e x um valor. Sendo $DIM = \{D_1, D_2, \dots, D_n\}$ o conjunto de todas as possíveis dimensões e U o conjunto de todos os possíveis valores associados a uma dimensão. Um contexto c é um subconjunto finito da relação $\{(d, x) | d \in DIM \wedge x \in U\}$. Representado da seguinte forma $c = [(D_1, U_1), (D_2, U_2), \dots, (D_n, U_n)]$.

Em seguida é definido o conceito de situação. Uma situação pode ser resumida como a interpretação e as implicações de um contexto. A partir das variáveis que compõem um contexto, já exemplificado, podemos inferir outras informações de mais alto nível. Se um contexto possui a localização de um usuário descrita em coordenadas globais (lat, lng), tem-se um detalhamento muito bom da sua posição mas uma informação pouco útil para a tomada de decisão. Ao se dar significado à essa dimensão, colocando-se no lugar de coordenadas em que tipo de estabelecimento o usuário se encontra, decisões podem ser tomadas com mais facilidade. Além da interpretação do dado, as situações podem gerar implicações lógicas como em: $\text{climaBom} = \text{quente} \text{ E } \text{ensolarado}$; $\text{climaBom} \text{ IMPLICA}$

sair. Para a criação das regras de interpretação e implicação é definida uma linguagem de expressão de situações de contexto CSEL (context situation expression language) com a qual o usuário do sistema pode definir suas próprias regras.

Associadas às definições de situação são definidas políticas para prevenir que ações indevidas sejam tomadas. Por permitir que um sistema computacional tome decisões que possam impactar diretamente o ambiente do usuário, é necessário a criação de regras que garantam a restrição de acesso e o correto comportamento do sistema. Em ALAGAR [2] as políticas são divididas em duas categorias:

- Políticas de dados: São regras que restringem os valores que certos sensores podem ter. Ao se mensurar a temperatura de uma sala pode-se detectar que o termômetro em questão está apresentando falha devido ao valor que está sendo reportado, como por exemplo um valor fora de um intervalo aceitável de temperatura para a sala. Esse tipo de mecanismo é essencial para evitar que o sistema tente interpretar ou até reagir ao receber informações de sensores danificados.
- Políticas de execução: São regras para controlar o comportamento do sistema. Um exemplo claro é a liberação de uma ação baseada nas permissões do usuário. Se o mesmo é administrador do sistema ele pode ter acesso a um conjunto de informações que outros usuários não podem acessar.

O último conceito definido para se entender a arquitetura em camadas proposta é o de adaptações. Sistemas sensíveis a contexto são sistemas que tem a capacidade de reagir e se adaptar às mudanças do ambiente. E, por isso, a definição desse termo deve ser feita para se ter uma compreensão completa da ferramenta. Na arquitetura em questão uma adaptação é um conjunto de reações que tem duas propriedades: 1) são respostas às mudanças nas situações e 2) podem afetar o ambiente. Uma reação representa uma ação atômica, que não pode ser mais dividida e pode alterar o meio a partir de um atuador. Pela complexidade que as adaptações podem ter devido à diversidade de contextos e situações, uma linguagem capaz de representar políticas e fluxos de ação (AWPEL - Adaptation Workflow and Policy Expression Language) foi desenvolvida para expressar

tais adaptações no *framework*. Com essa ferramenta é possível criar adaptações sequenciais ou em paralelo de alta complexidade. Um exemplo simples seria a resposta a uma ameaça recebida por um sistema de segurança, podendo ser representada pelo seguinte conjunto de ações: 1) ativar o alarme de incêndio, 2) fechar as saídas de áreas críticas e 3) ligar para um número de emergência.

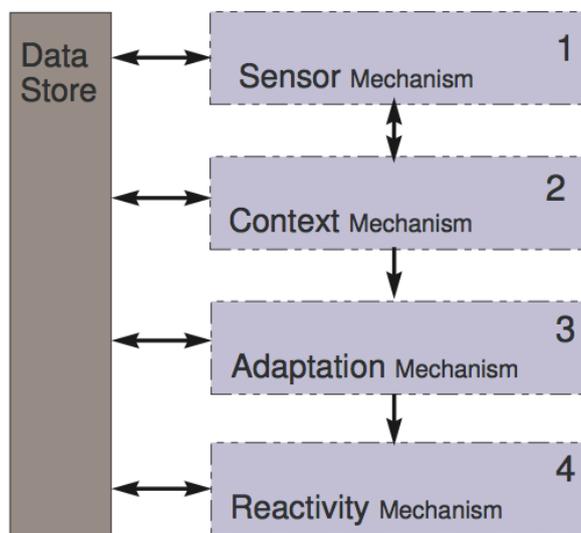


Figura 3.2 Arquitetura de sistema sensível a contexto. Adaptado de [2]

Por fim, depois da contextualização do trabalho é apresentada uma estrutura em alto nível dos componentes que compõem a arquitetura. Na imagem 3.2 pode-se ver que ela é composta por um banco de dados e quatro macro componentes para o gerenciamento e processamento de sensores, contexto, adaptações e reatividade, nessa ordem. O banco de dados utilizado na arquitetura é compartilhado entre as diversas camadas, o que possibilita o acesso às informações repassadas pelo usuário a qualquer momento por qualquer componente e possibilita, também, que camadas separadas possam se comunicar mesmo não sendo vizinhas. A principal função dessa base de dados é manter todas as preferências do usuário e as regras definidas utilizando todo o ferramental disponibilizado pelo *framework*. As camadas, por sua vez, são ordenadas e as informações de contexto devem ser repassadas para cada uma delas respeitando-se a precedência estabelecida na arqui-

tutura para garantir o correto tratamento das informações. Abaixo tem se uma descrição detalhada da estrutura e função de cada componente.

Sensores O primeiro componente é o mecanismo responsável pelo gerenciamento de sensores. Esse componente é dividido em seis elementos: sensores, conectores, controladores, tradutores, verificadores e um último responsável pela sincronização dos dados. Todos esses elementos são independentes o que gera flexibilidade de design, implementação e facilita a manutenção e escalabilidade do código. Para cada sensor tem-se uma pilha de elementos, chamada de camada interna de conexão, composta por um conector, um controlador, um tradutor e um verificador. Todos esses elementos estão muito relacionados com o tipo de dado que o sensor coleta e por isso é necessário uma pilha para cada sensor utilizado.

Pode-se resumir o fluxo da informação da seguinte forma: os sensores coletam os dados que vão ser transmitidos pelos conectores, a transmissão pode ocorrer por diversas tecnologias como WiFi, Bluetooth ou portas serial. Após sua transmissão os dados serão traduzidos para um formato que o sistema possa processar e antes de serem utilizados passam por uma etapa de verificação feita pelos verificadores.

Depois que os dados são recebidos pelo *framework* e passam por essa camada interna de conexão com os sensores, eles são repassados para o elemento de sincronização que é único nesta camada de processamento e é responsável pelo alinhamento e priorização das informações coletadas. Um sistema de localização pode usar vários sensores para determinar a posição dos usuários, o elemento de sincronização ao receber duas coordenadas diferentes para uma mesma pessoa pode usar o dado mais confiável, como um sensor biométrico ao invés de um leitor de cartões digitais, ou também usar uma variável de tempo para saber qual dos sensores tem de fato a última posição do usuário. Essa camada utiliza abstrações de alterações no ambiente para o processamento das informações. Quando algum elemento no ambiente é alterado, isso gera um evento na camada de sensores, cada evento cria o que é chamado de um estímulo no *framework*. Os estímulos podem ser criados por um evento externo ou uma mudança de parâmetro no ambiente, por exemplo: uma

pessoa entrando numa sala gera um evento de entrada com um parâmetro que é a identificação da pessoa. Cada estímulo pode estar associado com várias dimensões de um contexto e está relacionado a um sensor que é capaz de ler tais dimensões do ambiente. Nessa arquitetura os sensores são considerados elementos a parte, como caixas pretas, que tem sua própria implementação de hardware e software, que são desconhecidas.

Contexto O segundo componente é responsável pela criação de contexto e geração de situações. Essa camada é composta por três elementos: geradores de contexto, tradutores de contexto e interpretadores de contexto em situações. O geradores de contexto são ativados toda vez que o sincronizador da camada de contexto faz o processamento das informações e envia novos dados para a próxima camada. Eles são responsáveis por receber os dados coletados e criar os contextos de acordo com a representação citada anteriormente nessa arquitetura. Toda vez que uma dimensão é alterada a camada de sensores é acionada e envia novos dados para os geradores de contexto que sintetizam novos contextos. Antes de falarmos dos tradutores é preciso entender os interpretadores de contexto já que eles são os elementos que criam a necessidade dos tradutores. Esse trabalho utiliza uma definição de contexto que o representa na forma de um conjunto de pares (d, x) onde d é uma dimensão e x um valor. Outras representações de contexto existem e são citadas por trabalhos da área como em Dey [18], Winograd [21], Wrona [22] e Bettini [23]. Os interpretadores acessam os contextos criados e, a partir de uma base de definições de situações, podem inferir que situações são geradas a partir do mesmo. Esses interpretadores podem trabalhar com os diversos tipos de representações de contexto. Em ALAGAR três tipos de interpretadores são exemplificados: baseados em ontologias, baseados em regras e baseados em inferência. A capacidade de trabalhar com diversos interpretadores e poder utilizá-los em conjunto, em partes de um mesmo contexto, confere a essa arquitetura um grau de inovação não tido até o momento da publicação da mesma. Para que esses interpretadores possam processar os diversos tipos de contexto é necessária a criação de tradutores de con-

texto que transformam as diversas representações de contexto para uma única, na qual todos os interpretadores podem trabalhar. O resultado final dessa camada é a criação das situações, que são contextos de alto nível, inferidas das representações de informações do ambiente

Adaptação O terceira camada constitui o mecanismo de adaptações. Ela é composta por três elementos: uma unidade de resolução de adaptações, um motor de fluxos de ação e um verificador de políticas. As situações geradas na camada de contexto são processadas pela unidade de resolução de adaptações junto às definições descritas em WPEL(Workflow and Policy Expression Language), esse processamento gera uma série de reações que satisfazem as definições de adaptações passadas pelo administrador do sistema. Para cada situação processada são geradas uma ou mais reações por esse elemento da camada, que não necessariamente serão executadas. As reações geradas precisam passar, primeiramente, pelo motor de fluxo de ações, que é responsável por aplicar as políticas cadastradas na base de dados compartilhada contra as ações propostas e processar todo o fluxo de ações. A verificação de acordo com as políticas é feita pelo verificador de políticas que é utilizado pelo motor de fluxos. Podemos resumir a camada da seguinte forma: as reações, geradas pela unidade de resolução, mais as situações e as políticas de execução são repassadas como entradas para o motor de fluxo que, por sua vez, cria uma lista de adaptações válidas a serem tomadas fazendo uso do verificador de políticas. Essas adaptações serão enviadas para a próxima camada, responsável por, de fato, executá-las.

Reatividade A quarta e última camada contém os mecanismos de reação responsáveis por aplicar as adaptações no ambiente. As reações recebidas tem as seguintes propriedades 1) independência: podendo ser executadas sem a necessidade de outras. 2) atomicidade: efetuam apenas uma alteração. 3) comunicação: trocam informação com os atuadores e 4) dependência de contexto: precisam de parâmetros do contexto para efetuar certas alterações. Esse último ponto é necessário pois certos atuadores precisam das novas instruções a serem aplicadas no meio, como um controle de temperatura que precisa saber que novo valor deve ser aplicado. A partir

do momento em que essas reações são recebidas pela camada anterior elas são repassadas para um controlador que é capaz de fazer o acionamento dos atuadores que de fato vão alterar o ambiente. Os controladores fazem uso de configurações de atuadores, que ficam armazenadas na base de dados compartilhada, para acionar de maneira adequada os atuadores. Uma porta, por exemplo, pode ter várias configurações de velocidade e tempo de resposta a um comando e esses diferentes parâmetros ficam registrados nas configurações. Os comandos são por fim enviados através de um conector, que implementa os algoritmos de comunicação necessários, e um tradutor, que faz o mapeamento do comando para o atuador em questão. Pela singularidade de cada atuador e visando criar uma camada de abstração para simplificar as interações, se faz necessário um controlador e tradutor específico para cada um dos atuadores utilizados.

O *framework* descrito se mostra capaz de moldar-se a diversas aplicações e ao mesmo tempo é bem completo, olhando-se os aspectos arquiteturais. Ele trata da estrutura de obtenção de contexto por meios de sensores, se preocupando com a praticidade do *framework* ao criar diversos níveis de abstração. Trata também da representação de contexto, interpretação e por fim adaptação do ambiente, o que não só possibilita a criação de aplicações sensíveis a contexto mas também a criação de aplicações que podem agir sobre o ambiente utilizando-se de informações coletadas no mesmo. Alguns outros aspectos não são abordados em ALAGAR [2], como segurança por exemplo, porém em seguida será feito um estudo de uma outra arquitetura que trata desse problema.

3.2 ALPACA

Em ALPACA (Layered Architecture for Privacy-Assertions in Context-Aware Applications), um *framework* centrado em privacidade para o gerenciamento de informação de contexto [3] é introduzido, usando como base uma abordagem com foco em segurança da informação que utiliza um mapeamento conceitual das necessidades de privacidade do usuário para camadas distintas da arquitetura.

O principal objetivo, segundo os autores, é tornar os dispositivos móveis dos usuários em cofres para seus dados pessoais, de forma que o usuário tenha acesso completo e também poderá definir, com um mecanismo de acesso fino, como suas informações sensíveis serão compartilhadas.

No modelo proposto, a arquitetura em camadas possibilita que as informações do usuário percam precisão e fiquem cada vez mais ofuscadas a medida que passam pelas camadas. A ideia é criar diversos níveis de permissão para que cada aplicação criada nesse *framework* trabalhe apenas com o recebimento das informações estritamente necessárias para o seu funcionamento. Além das diversas camadas de proteção das informações sensíveis do usuários, é criado um sistema de permissões granular, justificado pela grande quantidade de aplicações que ou recebem diversas informações ou simplesmente não podem ser executadas. Esse sistema permite ao usuário optar por quais informações serão cedidas e em que nível de precisão essas informações serão repassadas às aplicações. Na figura 3.3 é possível ver as camadas e o nível de ofuscamento que é aplicado a cada informação que trafega pela arquitetura. Já na figura 3.4 pode-se ver como uma única informação pode ser repassada com diversos níveis de precisão e semântica, sendo possível, assim, atender aos requisitos de cada aplicação sem abrir mão da privacidade do usuário.

Adicionalmente é criado um sistema de liberação de contexto baseado em gatilhos que usa a representação de contexto do usuário para, em seu nome, aplicar permissões às aplicações e assim poder disponibilizar de forma adequada as informações sensíveis do usuário e o próprio contexto. Assim, o sistema pode liberar ou restringir o acesso a determinado sensor se certas condições forem atendidas, por exemplo o microfone poderia ser bloqueado ao se entrar numa reunião para impedir que informações confidenciais sejam registradas por aplicações indevidas ou liberado ao se entrar numa sala de aula para se manter um registro da mesma. Essa funcionalidade faz com que esse *framework* não seja apenas utilizado para a criação de aplicações sensíveis a contexto mas também o torna uma dessas aplicações já que, ao se utilizar do contexto do usuário para fazer a liberação das informações, ele também deve ser considerado uma aplicação sensível a contexto.

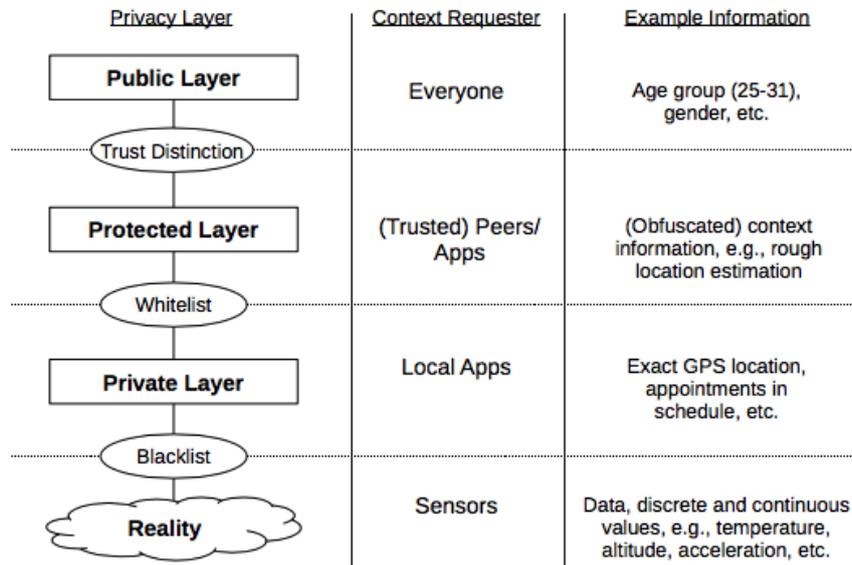


Figura 3.3 Arquitetura em camadas para a proteção de dados sensíveis. Adaptado de [3]

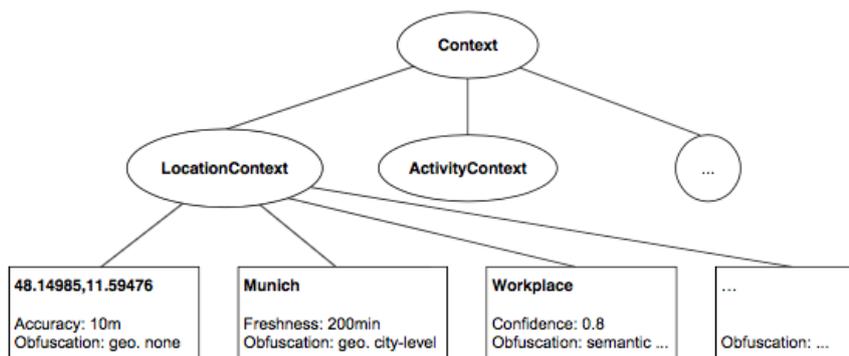


Figura 3.4 Representação de uma variável de contexto com diversos níveis de ofuscação. Adaptado de [3]

CAPÍTULO 4

EXEMPLOS E APLICAÇÕES

Quando se pensa em aplicações sensíveis a contexto é normal imaginar uma gama de aplicações extremamente complexas que são completamente integradas e que resolvem problemas capazes de mudar o mundo. Um ambiente no qual o usuário ao chegar em casa é recebido pela mesma, todas as suas vontades são atendidas e, após análise, é sugerida uma refeição balanceada utilizando-se informações do seu DNA e dos nutrientes atuais no seu organismo. Cria-se praticamente uma realidade utópica onde o ser humano vive em ambientes aparentemente comuns, mas que são capazes de responder a todas as suas perguntas e prover tudo com apenas o dizer de algumas palavras ou até mesmo apenas analisando o seu pensamento.

Teremos aplicações assim no futuro mas não é preciso esperar pois a computação sensível a contexto está mais perto do que é comumente imaginado. Não só podemos criar aplicações sensíveis a contexto hoje como já as temos. Quanto mais simples, menos perceptíveis elas são e mais interessantes e eficientes podem ser.

“As tecnologias mais profundas são as que desaparecem. Elas se integram no tecido da vida cotidiana até ficarem indistinguíveis dessa.” - Mark Weiser [7]

Este capítulo começará abordando aplicações simples que possuem um grande potencial de explicar o que são esses sistemas sensíveis a contexto e por fim algumas outras aplicações mais complexas serão exemplificadas.

Várias aplicações sensíveis a contexto estão no nosso dia-a-dia. Em muitas cidades, as luzes que iluminam as ruas já possuem foto-sensores que conseguem perceber, pela diminuição da intensidade da luz, que anoiteceu e acionam a iluminação da cidade. Quantas portas em estabelecimentos comerciais possuem sensores de presença e abrem automaticamente ao perceber a aproximação de um indivíduo? A maioria dos celulares e alguns

computadores possuem sensores que regulam automaticamente o brilho da tela de acordo com a luminosidade do ambiente, possibilitando que o conteúdo seja melhor visualizado e que o usuário não tenha a visão agredida. Por mais simples que essas aplicações pareçam elas são sensíveis a contexto e bastante úteis.

Ao se agregar mais funcionalidades e, com isso, valor a essas aplicações, facilmente é notado que as mesmas, de fato, reagem a mudanças no contexto. Imagine uma porta automática de um estabelecimento, ela abre para qualquer pessoa que se aproxime. Até então, uma aplicação bem simples. Mas a porta poderia abrir somente para pessoas autorizadas. Ao se aproximar da porta um sistema de autenticação validaria a credencial do usuário e o acesso para o mesmo seria liberado. Esse tipo de aplicação existe em pedágios, os motoristas podem optar por parar nas guaritas e efetuar o pagamento em dinheiro ou utilizar um sistema de credenciais acoplado ao veículo que não só libera o acesso à estrada para as pessoas habilitadas como também faz o débito do valor no sistema e se o usuário possuir fundos suficientes a sua passagem será liberada.

4.1 SUGESTÕES DE ACORDO COM O NÍVEL DE AGITAÇÃO

Em COSTA JUNIOR, J. B. [24], é proposto um *framework* que faz a avaliação da interação e da experiência de uso de Instrumentos Musicais Digitais (DMIs). Com o intuito de provar o *framework* são criados três DMIs para serem estudados em ciclos de iteração usados no aprimoramento do modelo. Uma das aplicações desenvolvidas pelo autor, a *Orquestra anticibernética dos filhos de Asimov*, tem como objetivo permitir que usuários (com foco em leigos em música) possam controlar uma orquestra virtual de frevo através de movimentos do corpo.

O sistema possui três módulos principais:

Rastreamento ou módulo de entrada - Responsável pela captura dos movimentos do corpo do usuário. Este modulo foi desenvolvido com o uso de um *Microsoft Kinect Sensor*¹, sensor de detecção de movimentos capaz de monitorar independentemente

¹Microsoft Xbox Kinect Website. <https://en.wikipedia.org/wiki/Kinect>

os membros de uma ou mais pessoas.

Expressividade ou módulo de saída - Responsável pelo controle de parâmetros musicais da aplicação (ex.: intensidade e articulação de diferentes instrumentos da orquestra)

Mapeamento ou modulo de conexão - Responsável pelo controle dos eventos de entrada para gerar o resultado musical desejado(conexão dos dados da entrada com a saída do sistema).

Para atingir o objetivo de controlar a orquestra, o sistema é capaz de identificar uma variável de contexto de alto nível que pode ser associada ao temperamento do usuário (o nível de euforia do mesmo). Essa variável de contexto pode ser definida por três estados:

Parado O menor estado de euforia, quando o usuário realiza poucos movimentos.

Intermediário Estado de euforia média, quando o usuário realiza movimentos mas estes não são tão significativos quantos o do próximo nível.

Eufórico O maior estado de euforia, quando o usuário realiza vários movimento bruscos de alta magnitude.

Esse DMI desenvolvido é uma aplicação sensível a contexto que é capaz de entender o nível de euforia do usuário e se adaptar a esta situação. Pode-se, utilizando a mesma metodologia, recriar essa aplicação em dispositivos móveis, como um smartwatch ou pulseiras digitais. No lugar de sensores visuais de movimento, seriam usados sensores inerciais para se chegar no nível de agitação dos usuários. Utilizando-se um smartwatch com esses sensores, é possível identificar com facilidade o nível de agitação do usuário e ao associar sua localização à esse contexto, a aplicação é capaz de fazer sugestões inteligentes como selecionar um treino específico ou uma lista de músicas ao perceber que o usuário está num parque fazendo atividades físicas.

4.2 TROCA DE INFORMAÇÕES POR GESTOS

Em 2015 a apple fez o deposito de uma patente[25] na qual uma tecnologia de troca de informações por proximidade utilizando um dispositivo acoplado ao pulso do usuário é

descrita.

A patente[26] descreve um sistema que pode facilitar a troca de informações, não se limitando apenas a informações de contato, entre os dispositivos de usuários diferentes. A troca de dados pode ser parcialmente ou completamente automatizada e será acionada através de um gesto de saudação que pode ser um aperto de mãos, *hand slap*(um outro cumprimento de mãos), ou até um abraço. Ao detectar esse evento de saudação o dispositivo pode selecionar que informações, baseadas no contexto atual, serão transmitidas. Esse sistema permite que o usuário controle como as informações serão compartilhadas. Ao fazer uso de variáveis de contexto pode-se criar regras de compartilhamento para que o sistema possa funcionar automaticamente sem necessidade de interação com usuário para solicitar autorização para o envio dos dados.

A aplicação descrita acima faz uso de sensores de movimento como acelerômetro e giroscópio para gerar um padrão de movimentos que identifica os dois dispositivos e, com uso de tecnologias de comunicação sem fio como bluetooth, NFC e WiFi, faz a troca de informações.

Apesar da apple ter feito uma descrição mais completa, essa aplicação não é nova. O Bump[27], aplicativo para android, fazia troca de informações como contatos, fotos, vídeos e arquivos em geral quando os usuários batiam um telefone no outro. Esse era o evento de pareamento que identificava os dispositivos e permitia que eles posteriormente fizessem a troca de dados.

Uma outra aplicação bem mais parecida com a descrição da patente da Apple é o iBand[28]. O iBand é uma pulseira que faz a troca de informações de contato por meio de um gesto comum, aperto de mãos especificamente, entre os usuário. A ideia do autores é de explorar aplicações na interseção das seguintes áreas: redes sociais e computação ubíqua. O dispositivo utiliza o movimento sincronizado de subida e descida das mãos dos usuários e um sensor infra-vermelho para fazer a troca de identificadores. As informações são armazenadas numa base de dados na nuvem e podem ser acessadas a partir dos IDs compartilhados.

4.3 DIRECIONAMENTO DE CONTEÚDO BASEADO EM PERFIL

A partir de uma tecnologia de localização *indoor*, capaz de posicionar usuários em ambientes fechados nos quais o sinal de GPS não consegue penetrar e fazendo uso de diversos algoritmos e técnicas de inferência, mineração de dados e aprendizagem de máquina. A In Loco[29], empresa de geolocalização e inteligência em dados, construiu uma plataforma que possibilita: a criação e ativação de clusters de usuários baseados em *geo behavior*; a validação de transações de cartões de crédito; a análise de potencial de retorno sobre um determinado ponto geográfico; o estudo do movimento pendular entre cidades (com a identificação de cidades dormitórios); validação de endereço comercial e residencial; análises estatísticas sobre abrangência de redes móveis entre diversas outras aplicações possíveis unindo-se localização precisa, massa de usuários e inteligência em dados.

Fundada por alunos do Centro de Informática(CIn), na Universidade Federal de Pernambuco, e inspirada por pesquisas na área de ubiquidade, a empresa sempre teve como foco o desenvolvimento de aplicações e tecnologias voltadas para essa nova era da computação.

Uma de suas aplicações, criada em cima da infraestrutura de dados e inteligência, é a In Loco Media[29], rede de publicidade *mobile* geolocalizada que, com o uso da plataforma citada acima, realiza a entrega de anúncios baseada no perfil do usuário e em momentos de alta probabilidade de conversão.

O perfil, anteriormente citado, é criado utilizando-se o histórico de localizações e outras informações coletadas nos dispositivos dos usuários. Esse conjunto de informações extraídas de smartphones, ou outros aparelhos que possuam a tecnologia da In Loco embarcada, é enviado para os servidores da empresa onde são processados e passam a compor a representação de comportamento dos usuários. É importante mencionar que apesar da quantidade de dados coletada e da capacidade de inferência da empresa, a In Loco trabalha com o conceito *privacy by design*, considerando a privacidade um tema de suma importância e. por isso, não coleta nenhum dado que possa identificar seus usuários².

²Personally Identifiable Information (PII), ou Sensitive Personal Information (SPI) são informações

A aplicação em questão, tem mudado o mercado de mídia e de ad-tech ao fazer a ativação dos usuários utilizando os seus hábitos no mundo físico e entregando métricas de visitas a pontos de venda para anunciantes. É uma aplicação que faz uso de várias variáveis de dispositivos móveis e do contexto de localização histórico, utilizando uma infraestrutura complexa de processamento de dados em larga escala que é capaz de inferir preferências sobre os usuários.

Esse tipo de plataforma em conjunto com uma grande quantidade de dados abre portas para a criação de diversas aplicações sensíveis a contexto que podem ajudar a chegarmos cada vez mais perto da visão utópica que é comumente atribuída à computação ubíqua.

que sozinhas ou associadas com outras informações podem ser utilizadas para identificar uma pessoa.
https://en.wikipedia.org/wiki/personally_identifiable_information

CAPÍTULO 5

PROBLEMAS E DESAFIOS

Estamos na era da computação móvel, já temos diversas aplicações sensíveis a contexto e possuímos ferramental para o desenvolvimento desses sistemas. Então por que a computação ubíqua ainda não é uma realidade? Neste capítulo, vamos entender as problemáticas que, de certa forma, inibem o desenvolvimento dessa área e fazem com que tais sistemas ainda não estejam sendo usados amplamente.

5.1 DE SENSORES AO CONTEXTO

O objetivo final de um sistema sensível a contexto é que o sistema chegue numa representação do ambiente ao redor que é próxima da percepção do usuário. [30] O desafio é conseguir, a partir de uma série de variáveis, chegar no contexto correto para a situação.

1) Os sensores tem que ser capazes de perceber o ambiente como um todo. 2) Ao mesmo tempo o sistema deve ter a capacidade de entender esses dados gerados e 3) transformá-los no contexto adequado que não seja uma compreensão parcial do ambiente (o que pode gerar contexto errados). 4) E no final o usuário tem que ter um mínimo de entendimento do que está acontecendo para poder fazer uso de tais sistemas.

Abaixo será feito um detalhamento de cada um desses problemas, o que ajudará na compreensão da magnitude dos desafios.

5.1.1 Percepção limitada

Uma pergunta importante para se ter um bom entendimento sobre percepção limitada é como se pode reduzir a diferença entre a percepção do mundo real do usuário e a do sistema. [30] Considerando-se alguns sensores, essa diferença de percepção pode ser

bem pequena ou inexistente, como localização que pode ser traduzida em nome de ruas, cidades ou estados e tempo que pode ser facilmente interpretado junto a um calendário. Por outro lado, outros dados obtidos através de sensores podem ter uma variação de percepção grande, como é o caso da percepção da temperatura do ambiente que obtida por um termômetro difere da sensação térmica percebida por uma pessoa, que pode ter a sensação de calor a 23°C ou de frio na mesma temperatura.

A percepção do usuário do que está ao seu redor é baseada nos sentidos humanos, mas tem relação, ao mesmo tempo, com experiência e memória. Quando um pessoa volta para casa de uma parada de ônibus tarde da noite, ela pode perceber que está escuro, calmo e frio, mas, ao mesmo tempo, ela pode achar a situação assustadora. Uma outra pessoa, que estava ocupada o dia inteiro num lugar repleto de outras pessoas, também pode perceber que está escuro, calmo e frio, mas, ao mesmo tempo, pode achar o ambiente relaxante e libertador. Esse exemplo mostra que utilizar apenas os dados dos sensores não retrata completamente a situação[30].

Não se tem atualmente um sistema que consiga perceber com completude o contexto em que o usuário está inserido. São inúmeras variáveis, inúmeras experiências e inúmeros resultados que podem ser obtidos a partir dessas informações. Fora as particularidades de entendimento de mundo de cada pessoa, um sistema, atualmente, está limitado às suas variáveis de ambiente o que gera uma percepção limitada de mundo para a aplicação. Percepção essa que difere da do usuário.

5.1.2 Compreendendo os dados

Numa foto facilmente se pode distinguir uma pessoa feliz de uma pessoa triste. E isso não se resume a um sorriso ou alguém cabisbaixo. Já para um computador essa capacidade de interpretação de estado emotivo é extremamente difícil de se conseguir. Por mais sofisticadas que estejam as aplicações de processamento de imagens, elas não conseguem captar certas sutilezas, que só outra pessoa pode identificar.

Quando se dirige um carro, novamente, é fácil sentir que o mesmo tem um boa resposta ou que ele possui um bom atrito à pista, mas como um computador pode ter a mesma

sensação ou conhecimento desses fatos?

Esses exemplos mostram que a compreensão dos dados ainda é um desafio para a inferência de contexto e, por conseguinte, para o desenvolvimento de aplicações que tratam esses dados.

Alguns sensores, como os de localização, possuem diferentes meios de obtenção de dados (por exemplo GPS, sinais de torres de celular) e diferentes meios de interpretação (World Geodetic System, WSG84, Post Code), meios esses já bem estabelecidos. Entretanto, para muitos outros sensores praticamente não há um modo único e bem estabelecido de interpretar as informações coletadas[30].

Por isso, mesmo que se consiga uma mensuração completa do ambiente, é necessário que os sistemas sensíveis a contexto tenham a capacidade de entender o que essas variáveis representam e em muitas aplicações isso se mostra bastante difícil de ser implementado, tornando a compreensão e interpretação dos dados mais um desafio para esses sistemas.

5.1.3 Erros de percepção de contexto

Existem vários exemplos de erros de percepção de contexto, como um carro que trava as portas com a chave na ignição, um computador que entra no modo de *screen saver* numa apresentação ou até um celular que muda para o modo *landscape* quando o usuário está deitado sem levar em consideração sua orientação em relação ao celular.

Quando se fala em erros de percepção, não está sendo dito que as aplicações tem que ser perfeitas e não podem nunca errar na inferência de contexto. Mas certas aplicações dificilmente irão ganhar tração, e serem largamente utilizadas, devido a dificuldades de aceitação de certos falsos positivos. Esse é o verdadeiro desafio da percepção de contexto. Por exemplo: um celular que não vai automaticamente para modo avião quando o usuário está prestes a decolar é, de certa forma, aceitável, mas um celular que entra em modo avião sem que o usuário esteja num vôo acaba atrapalhando mais do que ajudando o usuário.

Um outro exemplo seria um sistema que faz a troca de perfis de notificação (silencioso, em casa, na rua, reunião) de um celular baseado em sua localização e agenda para

supostamente silenciar o celular quando o usuário está em reuniões. Esse sistema poderia confundir uma ocasião e silenciar o celular do usuário num momento indesejado fazendo com que o mesmo perdesse uma ligação importante, ou não silenciá-lo numa reunião importante.

Algumas aplicações por mais simples que sejam, são extremamente difíceis de serem melhoradas a ponto de alcançar um estado aceitável. Isso torna esse erros de percepção mais comuns e faz com que esses aplicações sejam não apenas desprezadas, mas realmente ineficientes.

5.1.4 Capacidade de compreensão

O ideal da computação ubíqua é de que os sistemas computacionais estarão presentes em todo o lugar, não serão percebidos pelos usuários e as pessoas irão receber respostas antes mesmo de fazerem as perguntas. Mas em algumas aplicações se o usuário não entender corretamente como o sistema funciona, ele pode acabar fazendo um mau uso do mesmo e não conseguir os resultados desejados.

O usuários tem que conseguir entender minimamente o sistema. Por isso os sistemas computacionais sensíveis a contexto tem que possuir essa capacidade de compreensão para que o usuário possa saber porque algo está acontecendo. Ao ter seu acesso negado por uma porta, o usuário vai se perguntar porque ele não pode entrar, talvez exista uma política de restrição(ninguém pode entrar depois da meia noite). Ele tem que entender para que possa agir corretamente a situação.

A capacidade de compreensão das aplicações sensíveis a contexto torna-se um desafio quando o usuário não consegue fazer uso do sistema por não compreendê-lo corretamente, o que acontece com frequência.

5.2 SEGURANÇA E PRIVACIDADE

Recentemente a rede mundial de computadores tem sido fortemente atacada por grupos de cybercriminosos. Os ataques mais atuais começaram depois que Edward Snowden,

ex-funcionário da Agência Nacional de Segurança dos Estados Unidos (NSA), divulgou em 2013 arquivos confidenciais da agência que revelaram um gigantesco programa de vigilância global [31]. A NSA fazia uso de falhas no Windows 7, Windows Xp e outros sistemas operacionais para fazer o monitoramento ostensivo da população [31]. Em abril de 2017, um grupo de hackers chamado, The Shadow Brokers, lançou na internet várias ferramentas(EternalBlue, DoublePulsar) que podiam explorar as falhas já conhecidas pela NSA. A partir daí hackers se utilizaram das ferramentas e falhas divulgadas e lançam ataques globais(WannaCry, EternalRocks, Petya) através da internet que fizeram grandes empresas e instituições do mundo inteiro pararem [32].

Não é novidade sistemas computacionais serem hackeados e subvertidos. A segurança é um problema para qualquer sistema computacional. O que a torna desafiadora na computação sensível a contexto é que as aplicações e dispositivos que executam esse tipo de software estão cada vez mais populares e coletam cada vez mais dados do ambiente. Praticamente todas as pessoas tem um pequeno dispositivo portátil, com acesso à internet e com uma extensa capacidade sensorial. A Internet das Coisas (IoT) já aumentou a quantidade de aparelhos por pessoa e vai continuar a aumentar esses números. Esses dispositivos possuem várias informações sensíveis sobre o usuário e, se invadidos, por hackers podem ter essas informações comprometidas e espalhadas pela internet, como é o caso de algumas cameras IPs que foram hackeadas e os atacantes obtiveram acesso tanto ao seu hardware, para utilizá-lo para outros fins, quanto às imagens captadas pelo dispositivo.

O escândalo da coleta de dados em larga escala na internet pela NSA não só gerou um problema de segurança, mas também uma preocupação acerca da privacidade do usuário. A população agora não só tem a sensação de que está sendo espionada, ela tem certeza. Os problemas relacionados à privacidade não existem apenas em sistemas que podem ser subvertidos, eles estão presentes normalmente em aplicações que silenciosamente coletam informações de contexto e as enviam para a internet sem o consentimento do usuário.

Um exemplo claro disso é o caso de algumas TVs de grandes marcas que equipadas com microfones faziam a gravação de todo o audio do ambiente e os enviava para servi-

dores na nuvem. Essas mensagens foram interceptadas e, por não estarem fazendo uso de nenhum algoritmo de criptografia, os atacantes puderam facilmente escutar todo o conteúdo transmitido.

Tratando-se de aplicações, para se alcançar altos níveis de qualidade de serviços, os algoritmos usados para aquisição de contexto tipicamente buscam maximizar a resolução, a acurácia e precisão dos dados. Quando se fala em preservar a privacidade do usuário, entretanto, objetivos diferentes e, em parte, as vezes até contraditórios são perseguidos [3].

O foco em otimização dessas aplicações acaba gerando sistemas ricos em contexto mas que tratam pobremente a privacidade do usuário. O mesmo acontece com as infraestruturas disponíveis para o desenvolvimento dessas aplicações.

Focando em funcionalidades e performance, a maioria das arquiteturas existentes para gerenciar informações de contexto tipicamente colocam em produção componentes centrais para a coleta, análise e distribuição dos dados atualizados dos usuários [3]. Esses componentes centrais, muitas vezes, compartilham informações e tem um controle de acesso limitado, negligenciando assim a privacidade, apesar desta continuar sendo um fator crucial para a aceitação de tais sistemas.

O problema da privacidade, como Weiser citou, pode ser reduzido ao controle sobre a informação.

“The problem, while often couched in terms of privacy, is really one of control. If the computational system is invisible as well as extensive, it becomes hard to know what is controlling what, what is connected to what, where information is flowing, how it is being used. . . and what are the consequences of any given action” [33]

De qualquer forma, o controle de acesso e a ciência de fluxo de informação das aplicações melhora os parâmetros de privacidade do sistema. Então, por mais que o problema possa não ser a privacidade em si, ela é o resultado desejado.

A privacidade e segurança das informações do usuários estão intrinsecamente ligadas. Sem a segurança não conseguimos garantir a privacidade, pois os dados podem ser acessados uma vez que o sistema seja invadido. E sem a privacidade a segurança perde um pouco sua importância já que as informações pode ser acessadas por terceiros. É preciso

trabalhar ambos os conceitos juntos para garantir que os dados e informações sensíveis dos usuários estejam protegidos.

CONCLUSÃO

Neste trabalho foi feito um estudo sobre computação sensível a contexto, algumas arquiteturas e modelos estruturais foram discutidos, diferentes tipos de aplicações móveis foram analisados como a troca de dados por gestos, adaptação de acordo com o nível de agitação e o direcionamento de conteúdo baseado em perfil. Por fim foi abordado o problema da segurança, privacidade e alguns outros relacionados à inferência do contexto em si.

Percebe-se, por mais que o estudo da computação ubíqua em alguns meses complete 20 anos, que o tema é de fato bastante atual e continua imensamente inovador. Algumas das aplicações sugeridas no decorrer desses anos ainda são muito complexas e cada nova aplicação na área traz novidade e entusiasmo para os estudos no tema.

No decorrer deste trabalho é possível notar que ainda existe um caminho muito grande a ser percorrido para que o ideal de computação sensível a contexto e computação ubíqua sejam alcançados. Isso se deve, principalmente, à dependência que esses temas tem com outras áreas de estudo. Quando começamos a nos aprofundar nos problemas, percebe-se que outras áreas vão surgindo para que se possa, de fato, inferir o contexto. Para que uma aplicação possa entender o que está à sua volta e tomar uma decisão a respeito do que foi sentido, é necessário o uso de técnicas e algoritmos de outras áreas da computação como agentes autônomos ou inteligência artificial. Da mesma forma, se quisermos entender o que se passa num ambiente repleto de sensores, dados e pessoas é preciso toda uma estrutura de *big data*, análise, inferência e inteligência de dados para compreender, reconhecer padrões e ter a capacidade de tomar decisões em cima de toda essa informação.

Ao analisar a parte que inicialmente parece mais simples como a coleta de dados por sensores, por exemplo, também são encontrados diversos desafios que exigem o uso

de outros campos da computação para que estes possam ser superados. No caso da localização de pessoas ou objetos temos uma área inteira de estudo que vai de GPS, Redes WiFi e sensores de movimento até posicionamento por imagens para que se possa tornar a obtenção desse dado factível.

Quando tivermos um progresso significativo em áreas que dão suporte à essa nova era da computação vamos ser capazes de criar sistemas cada vez mais cientes e sensíveis ao contexto necessário para a aplicação e isso vai impulsionar e desenvolver a área como, até então, apenas imaginado.

A mobilidade que deu início à computação sensível a contexto só é possível hoje pelos avanços em eficiência energética, miniaturização de hardware, conectividade *wireless* entre outros. Um exemplo disso é IoT que só agora se tornou realidade e é um dos primeiros passos para a computação pervasiva.

Em alguns anos teremos o ferramental e a infraestrutura que vai nos possibilitar chegar bem perto do real conceito da computação ubíqua e, logo após isso, estaremos de fato nessa nova era.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. Krumm, Ed., *Ubiquitous Computing Fundamentals*. Boca Raton, FL: Chapman and Hall/CRC, 2010.
- [2] V. Alagar, M. Mohammad, K. Wan, and S. Als Salman Hnaide, “A framework for developing context-aware systems,” *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 1, 2014.
- [3] F. Dorfmeister, S. Feld, and C. Linnhoff-Popien, “Alpaca: A decentralized, privacy-centric and context-aware framework for the dissemination of context information,” *International Journal On Advances in Intelligent Systems*, vol. 7, no. 1 and 2, pp. 223–236, 2014.
- [4] “emarketer: 2 billion consumers worldwide to get smart(phones) by 2016,” <http://web.archive.org/web/20170218201554/https://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>, accessed: 2017-02-18.
- [5] “avelareduarte: Dispositivos móveis 2015 (estatísticas),” <https://web.archive.org/web/20170528215537/http://www.avelareduarte.com.br/fases-projetos/conceituacao/demandas-do-publico/pesquisas-de-usuarios-atividades-2/dados-sobre-o-publico-alvo/dispositivos-moveis-2015-dados-e-fontes/>, accessed: 2017-05-28.
- [6] “Tráfego global de dados móveis crescerá quase 10 vezes entre 2014 e 2019,” <https://web.archive.org/web/20170528214904/http://www.cisco.com/c/pt-pt/about/press/news-archive-2015/20150203.html>, accessed: 2017-05-28.

- [7] M. Weiser, “The computer for the 21st century,” *Scientific American*, vol. 265, pp. 94–104, 1991.
- [8] U. Hansmann, L. Merk, M. S. Nicklous, and T. Stober, *Pervasive Computing: The Mobile World*, 2nd ed. Berlin, Germany: Springer-Verlag, August 2003.
- [9] A. Greenfield, *Everyware: The Dawning Age of Ubiquitous Computing*. Indianapolis, Indiana: New Riders, March 2006.
- [10] S. Poslad, *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons Inc, 2009.
- [11] B. N. Schilit and M. M. Theimer, “Disseminating active map information to mobile hosts,” *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994.
- [12] B. Schilit, N. Adams, and R. Want, “Context-Aware Computing Applications,” *Proceedings of 1st Workshop on Mobile Computing Systems and Applications (MCSA’94), Santa Cruz, California, U.S.*, pp. 85–90, 1995.
- [13] A. Dey, “Providing architectural support for building context aware applications.” *Ph.D. thesis. Department of Computer Science, Georgia Institute of Technology, Atlanta*, 2000.
- [14] N. S. Ryan, J. Pascoe, and D. R. Morse, “Enhanced reality fieldwork: the context-aware archaeological assistant,” in *Computer applications in archaeology*, 1998.
- [15] G. Chen and D. Kotz, “A survey of context-aware mobile computing research,” *TR2000-381. Dept. of Computer Science, Dartmouth College*, 2000.
- [16] Michaelis dicionário brasileiro da língua. [Online]. Available: <http://michaelis.uol.com.br/busca?r=0&f=0&t=0&palavra=contexto>
- [17] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a better understanding of context and context-awareness,” in *Handheld and Ubiquitous Computing*, 1999, pp. 304–307.

- [18] A. K. Dey, G. D. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,” *Human-Computer Interaction*, vol. 16, no. 2-4, pp. 97–166, 2001.
- [19] Z. S., R. P., and J. L. Crowley, “An architecture for ubiquitous applications,” Laboratoire LIG, 681 rue de la Passerelle - Domaine Universitaire - BP 72, 38402 St Martin d’Hères, 2007.
- [20] K. WAN, “Lucx: Lucid enriched with context.” *Ph.D. thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.*, 2006.
- [21] T. Winograd, “Architectures for context.” *Human-Computer Interaction*, vol. 16, no. 2-4, pp. 401–419, 2001.
- [22] K. Wrona and L. Gomez, “Context-aware security and secure context-awareness in ubiquitous computing environments.” *In XXX Autumn Meeting of Polish Information Processing Society Conference Proceedings*, pp. 255–265, 2005.
- [23] C. Bettini, O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.
- [24] J. B. COSTA JUNIOR, “Um framework para avaliação da experiência de uso de instrumentos musicais digitais,” Dissertação de mestrado, Centro de Informática, Universidade Federal de Pernambuco, 2013.
- [25] “Apple watch patent would have you sharing files with a handshake,” <https://web.archive.org/web/20160419155747/http://www.engadget.com/2015/06/29/apple-watch-data-exchange-patent/>, accessed: 2017-07-02.
- [26] “53. (wo2015094220) gesture-based information exchange between devices in proximity,” <https://web.archive.org/web/20160321232321/https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2015094220&recNum=53&maxRec=51457&office=>

- &prevFilter=%26fq%3DOF%3AWO%26fq%3DICF_M%3A”H04W”&sortOption=Data+pub+ordem+inversa&queryString=&tab=PCTDescription, accessed: 2017-07-02.
- [27] “Bump (application),” [https://web.archive.org/web/20161129194950/https://en.wikipedia.org/wiki/Bump_\(application\)](https://web.archive.org/web/20161129194950/https://en.wikipedia.org/wiki/Bump_(application)), accessed: 2017-07-02.
- [28] *Toward wearable social networking with iBand*. ACM, 2005.
- [29] In loco media. [Online]. Available: <https://www.inlocomedia.com>
- [30] “Interaction design foundation: Context-aware computing,” <https://web.archive.org/web/20170108040242/https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction> accessed: 2017-01-08.
- [31] Wikipedia - edward snowden. [Online]. Available: https://en.wikipedia.org/wiki/Edward_Snowden
- [32] Wikipedia - wannacry. [Online]. Available: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
- [33] M. Weiser, R. Gold, and J. S. Brown, “The origins of ubiquitous computing research at parc in the late 1980s.” *IBM Systems Journal*, vol. 38, no. 4, pp. 693–696, 1999.