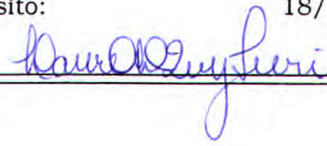


SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC - USP

Data de Depósito:

18/05/2004

Assinatura: _____



Gerenciamento de contexto: uma abordagem baseada em representações extensíveis ¹

Rodrigo Felicio dos Santos ²

Orientador: *Prof. Dr. Edson dos Santos Moreira* ³

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP), como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Abril de 2004

¹ Trabalho realizado com auxílio financeiro do CNPq.

² E-mail: rodrigo @ rodrigofelicio.com.br

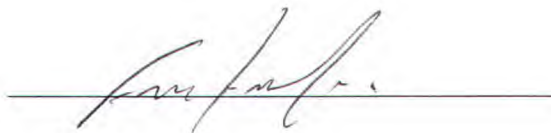
³ E-mail: edson @ icmc.usp.br

A Comissão Julgadora:

Prof. Dr. Edson dos Santos Moreira

A handwritten signature in black ink, consisting of a stylized 'M' followed by a long horizontal line.

Prof. Dr. Francisco José Monaco

A handwritten signature in black ink, appearing to be 'Francisco José Monaco' written in a cursive style.

Prof. Dr. Eleri Cardozo

A handwritten signature in black ink, appearing to be 'Eleri Cardozo' written in a cursive style.

Dedicatória

Aos meus pais, Ivan e Cláudia, pelo intenso amor, carinho, compreensão, exemplo e apoio durante todos esses anos. Ao meu irmão Rafael, pelo companheirismo, afeto e amor. Plena gratidão a vocês.

Agradecimentos

Agradeço a Deus por mostrar que mesmo nos momentos mais difíceis, com perseverança, dedicação, determinação, atitude e um pouco de paciência é possível alcançar nossos objetivos. Obrigado também pela família, namorada e amigos. Sem eles nada faria sentido.

Gostaria de agradecer a todos da minha família, em especial aos meus pais pelo incondicional apoio psicológico, ao meu irmão pelos pagodes e à minha avó Emília pelas coxinhas de mandioca com carne moída (sem elas com certeza haveria menos fósforo pra gastar e, logicamente, o mestrado estaria comprometido!).

Gostaria de fazer um agradecimento especial à Ana Palmira, minha namorada, por ser um exemplo de pessoa, pelo seu companheirismo, amor, carinho, apoio, determinação, alegria, romantismo e principalmente, pela paciência que teve comigo durante todo esse tempo de luta. Muito obrigado princesa, tenho muito orgulho de estar ao seu lado.

Ao meu orientador Edson dos Santos Moreira por permitir a possibilidade dessa pesquisa, pelo apoio, e confiança. Graças a ele fui capaz de descobrir habilidades que não conhecia e desenvolver características que, sem dúvida, serão necessárias por toda vida. Obrigado Edson.

À professora Graça Pimentel pelos ensinamentos adquiridos durante as aulas de Hipermídia. Sem eles eu teria gasto mais tempo para efetuar a minha implementação.

Agradeço imensamente ao amigo e companheiro de trabalho Rudinei Goularte, pelos ensinamentos, confiança, humildade, apoio incessante, confiança e amizade. Obrigado Rudinei, você é um exemplo de profissional e de pessoa para mim. Já que estou falando de ícones dessa fase da minha vida, gostaria de registrar aqui o agradecimento aos meus amigos Kamarão pelas injeções de auto-estima enviadas diretamente do Canadá (valeu kmr, você é outro exemplo de sucesso máximo!) e Renato Bulcão pelas considerações relevantes durante o trabalho, confiança na minha

competência e, principalmente, amizade demonstrada. Valeu Bulcão!
Gostaria de agradecer ao amigo e companheiro de república Marcelo Muniz, por dividir todos os momentos de alegria, preocupação, balada e insanidade, e ao amigo e não menos companheiro de baladas Dalton Matsuo por seus conselhos, suas pérolas e palavras sábias. Devo mencionar aqui que durante o tempo de mestrado essa figura passou por uma transformação sem igual, de modo que posso afirmar sem sombra de dúvida que ele não é aquele japonês que os professores do cursinho mencionavam naquele ditado: "Enquanto vc está festejando, tem um japonês estudando."

Agradeço ao amigo e companheiro de trabalho Francisco Milagres pelas contribuições para este trabalho, pelo apoio em diversos momentos da pesquisa e da vida. Valeu mestre!

Agradeço ainda aos Intermídia Boys: Fernando (Pezão), Fabrício (Fmz), Robson, Tom, Martins, Léo Andrade, Flávia, Juninho, Renan, Andrea, Billy e Elaine pela amizade.

Obrigado também a Jorge Félix pela contribuição na modelagem e troca de conhecimentos e a Rijomar pela disponibilidade em esclarecer minhas dúvidas. Obrigado pelo apoio.

Aos amigos Renato, Helena, Luciana, Marcos e Ana Paula pela amizade e pela revisão deste documento. Obrigado.

Agradeço também ao pessoal do CCE, em especial ao conterrâneo Giuliano pela ajuda durante o desenvolvimento do trabalho.

Agradeço aos perebentos pelas peladas e ao Daniel Dosualdo pelo apoio desde a chegada em São Carlos. Obrigado.

Finalmente, agradeço às amizades que conquistei aqui em São Carlos, aos amigos de Rio Preto e a todos que contribuíram de forma direta ou indiretamente para a realização deste trabalho, para a conclusão desta fase da minha vida, ou simplesmente, pelo meu sucesso.

*“Eu sou a luz do mundo; Aquele que me segue não andar  em trevas,
mas ter  a luz da vida.”*

Jo o 8:12

Sumário

| | |
|--|----------|
| Sumário | xii |
| Lista de Figuras | xiv |
| Lista de Tabelas | xv |
| Resumo | xvii |
| <i>Abstract</i> | xix |
| 1 Introdução | 1 |
| 1.1 Motivações | 2 |
| 1.2 Objetivos do Trabalho | 4 |
| 1.3 Organização da Dissertação | 5 |
| 2 Computação Ciente de Contexto | 7 |
| 2.1 Definição de Contexto | 9 |
| 2.2 Definição de Computação Ciente de Contexto | 11 |
| 2.3 Categorização de Características para Aplicações Cientes de Contexto | 12 |
| 2.4 Classificação de Contexto | 13 |
| 2.5 Requisitos para Desenvolvimento de Aplicações Cientes de Contexto | 15 |
| 2.5.1 Especificação de Informações de Contexto | 15 |
| 2.5.2 Separação de Aquisição e de Utilização de Informações de Contexto | 15 |
| 2.5.3 Interpretação das Informações de Contexto | 16 |
| 2.5.4 Comunicação Distribuída e Transparente | 16 |
| 2.5.5 Disponibilidade Contínua dos Componentes de Captura de Informações de Contexto | 16 |
| 2.5.6 Armazenamento de Informações de Contexto | 17 |
| 2.5.7 Descoberta de Recursos | 17 |
| 2.6 Trabalhos Relacionados | 17 |
| 2.6.1 Active Badge | 18 |

| | | |
|----------|---|-----------|
| 2.6.2 | Cyberguide | 18 |
| 2.6.3 | Context Toolkit | 19 |
| 2.6.4 | Context Fabric | 20 |
| 2.6.5 | Context Kernel | 21 |
| 2.7 | Representação de Contexto | 22 |
| 2.8 | Considerações Finais | 32 |
| 3 | O Serviço de Gerenciamento de Contexto | 35 |
| 3.1 | Representação de Contexto no Serviço de Gerenciamento de Contexto | 36 |
| 3.1.1 | Representação de Contexto de Dispositivos | 39 |
| 3.1.2 | Representação de Contexto de Rede | 42 |
| 3.2 | O Serviço de Gerenciamento | 47 |
| 3.3 | O Coletor Contextual | 48 |
| 3.4 | O Web Service Gerente de Contexto | 49 |
| 3.4.1 | O Protocolo SOAP | 50 |
| 3.4.2 | A Linguagem WSDL | 51 |
| 3.4.3 | O Protocolo UDDI | 51 |
| 3.4.4 | Disponibilidade de Informações Contextuais | 51 |
| 3.5 | Modelagem do Serviço de Gerenciamento de Contexto | 54 |
| 3.6 | A Arquitetura do Serviço de Gerenciamento de Contexto | 56 |
| 3.7 | O Fluxograma do Serviço de Gerenciamento de Contexto | 57 |
| 3.8 | Exemplo de Uso | 58 |
| 3.9 | Considerações Finais | 63 |
| 4 | Conclusões | 65 |
| 4.1 | Discussão Sobre as Contribuições Deste Trabalho | 66 |
| 4.2 | Limitações do Serviço de Gerenciamento de Contexto | 67 |
| 4.3 | Propostas Para Trabalhos Futuros | 67 |
| A | Representações de Contexto | 71 |
| A.1 | Representação de Contexto das Características dos Dispositivos | 71 |
| A.2 | Representação de Contexto de Rede | 71 |
| B | Especificação WSDL do Web Service Gerente de Contexto | 75 |

Lista de Figuras

| | | |
|-----|--|----|
| 1.1 | Infra-estrutura para produção, distribuição e consumo de conteúdo do protótipo de TV Interativa. | 4 |
| 1.2 | O Gerenciador de Serviços do protótipo de TV Interativa. Destaque para o Serviço de Gerenciamento de Contexto. | 5 |
| 2.1 | Arquitetura do <i>Context-Toolkit</i> | 19 |
| 2.2 | Fluxograma do <i>Context Kernel</i> | 21 |
| 2.3 | Componentes que compõem o contexto do Sistema. | 25 |
| 2.4 | Diagrama de classes UML para o tipo <i>ContextType</i> | 27 |
| 2.5 | Diagrama de classes UML do contexto da Infra-Estrutura (<i>InfrastructureType</i>). | 28 |
| 2.6 | (a) O tipo <i>LapTopType</i> criado para representar uma entidade dispositivo (<i>laptop</i>). (b) Instância de um dispositivo do tipo <i>LapTopType</i> | 29 |
| 2.7 | Diagrama de classes UML para os contextos da Aplicação (<i>ApplicationsType</i>) e do Usuário (<i>UsersType</i>). | 30 |
| 2.8 | Diagrama de classes UML referente ao contexto do Sistema (<i>SystemType</i>). | 30 |
| 2.9 | Diagrama de classes UML representando o contexto Primário (<i>PrimaryContextType</i>). | 31 |
| 3.1 | Diagrama hierárquico dos elementos de contexto. Destaque para o elemento <i>How</i> | 38 |
| 3.2 | Diagrama de classes UML do contexto Primário (<i>PrimaryContextType</i>) após a inserção do tipo <i>HowType</i> | 39 |
| 3.3 | Perfil CC/PP. | 40 |
| 3.4 | Diagrama hierárquico do tipo <i>CCPPfeaturesType</i> que encapsula os tipos declarados no padrão CC/PP. | 41 |
| 3.5 | (a) Tipo <i>CCPPfeaturesType</i> . (b) Instância de um dispositivo com característica que suporta o uso de cores (<i>ColorCapable</i>). | 42 |

| | | |
|------|--|----|
| 3.6 | Fórmula utilizada para o cálculo da largura de banda utilizada individualmente pelos <i>hosts</i> | 45 |
| 3.7 | Tipo <i>HostType</i> | 46 |
| 3.8 | (a) Declaração do tipo <i>NetPropertiesType</i> . (b) Instância de rede. | 47 |
| 3.9 | Fluxograma do Coletor Contextual. | 49 |
| 3.10 | Fragmento XML retornado como resposta à requisição do contexto primário <i>How</i> referente ao dispositivo. | 52 |
| 3.11 | Exemplo de Política EPAL. | 55 |
| 3.12 | Diagrama de classes UML representando o Serviço de Gerenciamento de Contexto. | 56 |
| 3.13 | Diagrama UML do <i>Web Service</i> Gerente de Contexto. | 56 |
| 3.14 | A arquitetura de camadas do Serviço de Gerenciamento de Contexto. | 57 |
| 3.15 | Fluxograma do Serviço de Gerenciamento de Contexto. | 58 |
| 3.16 | Fluxograma da interação entre a Aplicação, o Serviço de Gerenciamento de Privacidade e de Contexto. | 59 |
| 3.17 | Tela de apresentação e <i>login</i> da aplicação ciente de contexto. | 60 |
| 3.18 | Instância de representação de contexto de usuário. | 61 |
| 3.19 | Regra de acesso gerada dinamicamente pelo Serviço de Gerenciamento de Privacidade. | 61 |
| 3.20 | Tela de apresentação do conteúdo ao usuário após a verificação da regra de acesso. | 62 |

Lista de Tabelas

| | | |
|-----|--|----|
| 3.1 | Parâmetros SNMP referentes à interface de rede e suas descrições. | 45 |
| 3.2 | Funcionalidades dos elementos do Serviço de Gerenciamento de Contexto. | 48 |
| 3.3 | Exemplos de valores das categorias que compõem o vocabulário EPAL. | 54 |

Resumo

A computação ciente de contexto faz uso da exploração de informações contextuais para prover serviços relevantes aos usuários. Lacunas dessa área são: a carência de representações contextuais extensíveis, e a falta de exploração de outros tipos de informações contextuais, que não o contexto do Usuário. Aplicações que proporcionam acesso ao conteúdo por meio de dispositivos heterogêneos, por exemplo, necessitam imperativamente de informações contextuais referentes a esses dispositivos, classificadas na literatura como informações contextuais referentes à Infra-Estrutura. Com o aumento na quantidade de informações de contexto a ser utilizada, um outro problema emerge: a necessidade de gerenciamento dessas informações. A literatura aponta alguns trabalhos que convergem para a solução desse problema. Porém, todos eles apresentam limitações em relação à representação de contexto.

Este trabalho tem como propósito explorar a representação contextual referente à Infra-Estrutura, sobretudo a representação de contexto de dispositivos e rede e desenvolver um serviço de gerenciamento de informações de contexto. Além disso, o trabalho apresenta como contribuições: a especificação do modelo que representa o contexto da Infra-Estrutura, o desenvolvimento do serviço de gerenciamento contextual, seus componentes, especificações da arquitetura, modelagens e fluxogramas.

Abstract

*T*he Context-aware Computing area has been exploited as a way to provide contextual information helping the development of services to users. Issues in this area are: the need of extensible and uniform contextual representations and the exploitation of other types of contextual information beyond the user context. Applications that provide access to content using heterogeneous devices, for instance, imperatively claim for devices contextual information. This information is classified by literature as Infrastructure contextual information. As the amount of contextual information increases, another issue emerges: the need of managing this information. Many efforts reported in the literature have worked on these issues. However, they present limitations related to the contextual representation.

This work aims to exploit the contextual representation of Infrastructure, such as devices and network. Besides that, this work aims to develop a context information management service. The main contributions are: the model specification, which represents the Infrastructure context, the context information management service, and the specification of its components, architecture, modeling and workflows.

Introdução

Durante a comunicação, pessoas se beneficiam de artifícios como gestos, escrita e fala para expressar opiniões e idéias. Logo, a interação ocorre mais facilmente, pois é enriquecida com informações implícitas que revelam o contexto em que os participantes estão inseridos. Baseado nesse tipo de interação, o pesquisador Mark Weiser vislumbrou dispositivos computacionais integrados ao ambiente no intuito de coletar informações referentes ao contexto e, conseqüentemente, explorá-las de modo a favorecer os usuários. A área de pesquisa introduzida por Weiser, que aborda a interação não intrusiva, é denominada **Computação Ubíqua** (Weiser, 1993).

Um dos focos de pesquisa na área de computação ubíqua é a **Computação Ciente de Contexto**. O propósito é explorar o emprego de informações contextuais que caracterizem a interação usuário-computador e com base nessas informações prover serviços relevantes ao usuário. Por exemplo, de acordo com informações contextuais do usuário, como preferências, um sistema é capaz de prover conteúdo personalizado.

Uma das dificuldades no uso de informações contextuais é a tarefa de definir quais informações de contexto são relevantes em uma determinada interação. Para isso, são apontadas na literatura diretrizes conhecidas como dimensões semânticas, capazes de auxiliar projetistas nessa tarefa (Dey, 2000; Abowd & Mynatt, 2000). A literatura ainda apresenta outro recurso com essa finalidade: o contexto primário, o qual é composto pelos tipos de informações contextuais de maior relevância. O contexto primário pode ser usado como índice para outras fontes de informação contextual (Schilit et al., 1994; Brown et al., 1997; Pascoe, 1998; Dey & Abowd, 2000).

1.1 Motivações

Desde o início das pesquisas em ciência de contexto, o tipo de contexto mais explorado na literatura é o contexto do Usuário (Abowd & Mynatt, 2000; Dey & Abowd, 2000; Dey, 2001). No entanto, alguns sistemas necessitam de informações de contexto adicionais às informações de contexto dos usuários. Esse conjunto de informações adicionais é classificado por alguns autores como informações de Infra-Estrutura, Sistema, Domínio e Ambiente (Schilit, 1995; Dix et al., 2000; Chalmers, 2002; Crowley et al., 2002). Um exemplo de situação do uso de informações de Infra-Estrutura é a utilização de dispositivos heterogêneos por parte do usuário. Nesse caso, é imperativo conhecer o contexto referente à Infra-Estrutura para prover conteúdo orientado aos diversos tipos de dispositivos que podem ser utilizados.

A classificação das informações de contexto em diversas categorias contribui para uma melhor organização dessas informações. No entanto, também traz consigo a necessidade de gerenciamento, exatamente pela grande quantidade de informações que podem ser utilizadas em aplicações cientes de contexto (informações de usuários, infra-estrutura, aplicação, etc.). Exemplos de pesquisas que abordam o gerenciamento contextual são os trabalhos de Salber et al. (1999) com o *Context Toolkit*, o projeto *Context Fabric*, de Hong & Landay (2001) e o *Context Kernel* de Arruda Jr (2003). O primeiro fornece componentes abstratos que podem ser conectados para processar a informação contextual vinda de sensores. O *Context Fabric* fornece serviços de notificação de eventos e consultas a informações contextuais. O *Context Kernel* atua como uma ferramenta de armazenamento de informações contextuais para aplicações cientes de contexto. Os dois primeiros possuem notificação a eventos e captura de informações contextuais provenientes de sensores, características ausentes no *Context Kernel*. Por outro lado, esse último é capaz de prover suporte a aplicações cientes de contexto heterogêneas, por se tratar de um serviço Web (Arruda Jr et al., 2003).

Embora as diferenças entre esses trabalhos sejam evidentes, todos eles têm uma lacuna em comum: a ausência de uma representação contextual extensível. Segundo Abowd et al. (2002), sem representações satisfatórias à aplicação, os desenvolvedores são forçados a implementar esquemas proprietários e limitados para armazenamento e manipulação. Conseqüentemente, essa barreira dificulta o intercâmbio das informações de contexto. Alguns trabalhos tiveram seus esforços voltados para a construção de representações de contexto (Schilit et al., 1994; Kindberg et al., 2000; Crowley et al., 2002; Harter et al., 1999). Porém, a maioria não se preocupou em classificar os tipos de contexto de forma a obter uma representação estruturada e abrangente das

informações contextuais.

Um exemplo de trabalho que possui uma representação extensível e que procura abranger os tipos de informações contextuais, além de possuir uma classificação e organização de contexto coerente é o projeto do protótipo de TV Interativa, desenvolvido pelo grupo de pesquisadores atuante no Laboratório Intermídia do Instituto de Ciências Matemáticas e Computação da Universidade de São Paulo em São Carlos. Desde 1993 esse grupo atua em pesquisa e desenvolvimento de sistemas multimídia e hipermídia distribuídos baseados em padrões, como MPEG (*Moving Picture Experts Group*) (ISO/IEC, 2003) e W3C (*World Wide Web Consortium*)¹, além de desenvolver aplicações que utilizam vídeo (Faria, 2001; Faria et al., 2001; Santos Jr et al., 2001; Santos Jr, 2002; Goularte & Moreira, 2001, 2002; Goularte et al., 2003; Goularte, 2003). O objetivo do protótipo de TV Interativa é utilizar informações de contexto para potencializar o desenvolvimento de serviços avançados, como geração e adaptação automática e personalizada de conteúdo. Para tanto, faz-se necessário um modo padronizado de representação das informações de contexto. Além disso, o protótipo de TV Interativa almeja prover conteúdo aos usuários considerando dispositivos heterogêneos. Portanto, o uso do contexto referente à Infra-Estrutura é imprescindível.

A infra-estrutura para produção, distribuição e consumo de conteúdo do protótipo de TV Interativa é ilustrada na Figura 1.1. Os elementos responsáveis pela captura, autoria, codificação, *streaming* e apresentação (*player* com *plug-in* MPEG-4) representam ferramentas comerciais da empresa Envivio². Os elementos *Servidor de Descrições Multimídia (SDM)* e *Gerenciador de Serviços* representam as aplicações desenvolvidas por Goularte (2003) no contexto do protótipo. As setas contínuas indicam o fluxo de dados e as setas pontilhadas indicam alguns dos trabalhos futuros do projeto.

Para que as aplicações pudessem se beneficiar de informações de contexto, esse autor especificou e desenvolveu um *framework*³ para classificação e organização de contexto, capaz de abordar tipos de contexto mencionados na literatura (Usuário, Infra-Estrutura, Aplicação e Sistema). Tal *framework* permite que desenvolvedores de aplicações cientes de contexto possam utilizar tipos contextuais e entidades pertencentes aos tipos de contexto contidos na representação, ou ainda, se necessário ou conveniente, que os desenvolvedores criem seus próprios tipos e entidades contextuais apenas estendendo as representações de contexto.

Apesar do *framework* organizar de modo estruturado a criação de mode-

¹<http://www.w3c.org/>

²<http://www.envivio.com>

³Estrutura reusável de um programa ou de parte de um programa que pode ser expressa como um conjunto de classes (Johnson, 1992)

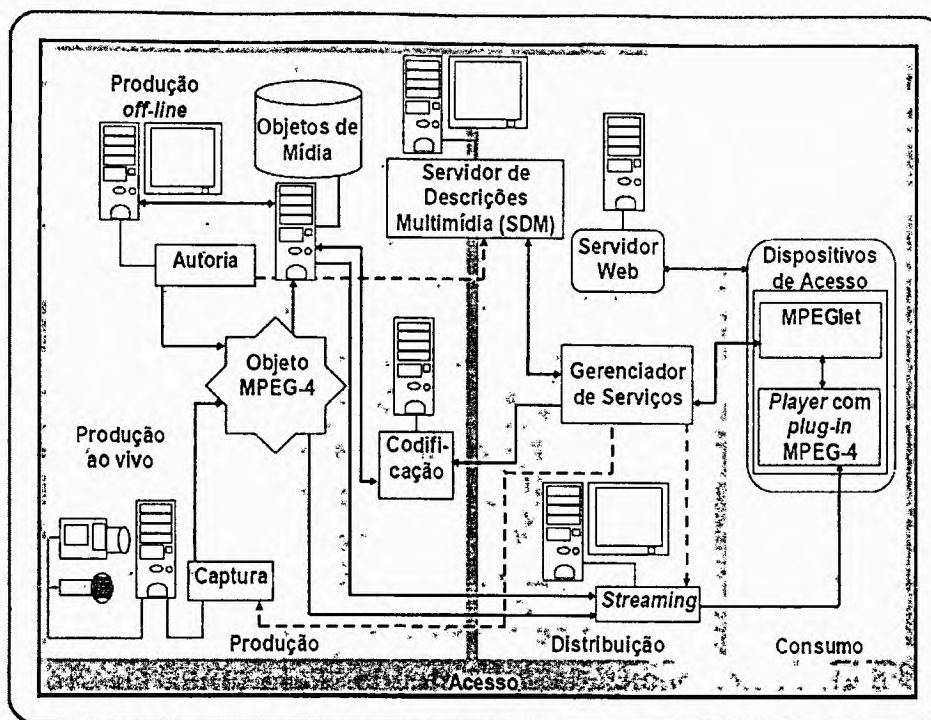


Figura 1.1: Infra-estrutura para produção, distribuição e consumo de conteúdo do protótipo de TV Interativa. Adaptado de (Goularte, 2003).

los de contexto, a definição de um modelo abrangente para o contexto de Infra-Estrutura, sobretudo o contexto referente aos dispositivos e às características da rede, ficou como trabalho futuro (Goularte, 2003). O tratamento do contexto de Infra-Estrutura tem sua relevância evidenciada pela heterogeneidade dos dispositivos que podem ser utilizados pelos usuários para acesso ao conteúdo. Além disso, a utilização de informações de contexto referentes à Infra-Estrutura reforça a necessidade de gerenciamento dessas informações.

1.2 Objetivos do Trabalho

Considerando-se o contexto do protótipo de TV Interativa, este trabalho tem como objetivos desenvolver uma representação de contexto de Infra-Estrutura mais elaborada, condizente com as necessidades do protótipo de TV Interativa abordando a descrição de contexto referente aos dispositivos e rede e ainda, criar um serviço de gerenciamento de informações de contexto levando-se em conta características relevantes de trabalhos mencionados na literatura (Salber et al., 1999; Hong & Landay, 2001; Arruda Jr et al., 2003), mas, tomando como base o modelo de representação contextual de Goularte (2003). Tal serviço faz parte do gerenciador de serviços apresentado na Figura 1.1 e ilustrado de modo mais detalhado na Figura 1.2. A criação do serviço de gerenciamento de contexto é justificada devido à grande quantidade de informações contextu-

ais do Usuário e da Infra-Estrutura que devem ser manipuladas. Vale ressaltar que o gerenciador de serviços é composto também por serviços que tratam do conteúdo provido pelo protótipo e por um serviço de gerenciamento de segurança ⁴.

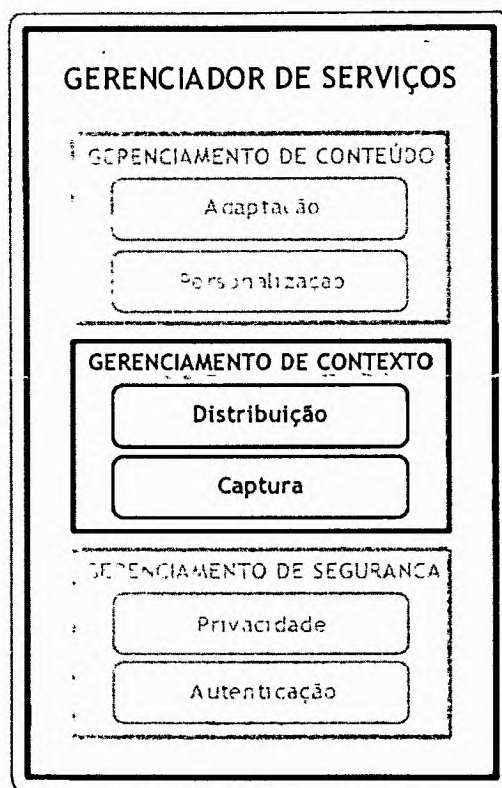


Figura 1.2: O Gerenciador de Serviços do protótipo de TV Interativa. Destaque para o Serviço de Gerenciamento de Contexto.

1.3 Organização da Dissertação

Esta dissertação está organizada em quatro capítulos e dois apêndices. O Capítulo 1 é composto por esta introdução.

O Capítulo 2 apresenta a revisão bibliográfica referente à computação ciente de contexto. Inicialmente é feita uma introdução sobre a área de computação ubíqua e seus principais focos de pesquisa. Em seguida, são abordadas definições, classificações e representações de contexto, bem como requisitos para o desenvolvimento de aplicações cientes de contexto e trabalhos relacionados.

O Capítulo 3 apresenta a representação contextual utilizada no desenvolvimento do projeto. O capítulo ainda aborda estratégias e protocolos utilizados

⁴Milagres, F. G. *Segurança em Computação Baseada em Informações de Contexto*. A ser editado pelo ICMC/USP, 2004.

para a representação, bem como artifícios e definições. O capítulo ainda descreve a implementação do serviço de gerenciamento de contexto. São descritos os componentes, os recursos utilizados, a modelagem, protocolos, a arquitetura, limitações do serviço e um cenário para ilustrar o uso do mesmo.

O Capítulo 4 apresenta as conclusões do trabalho, aborda as contribuições obtidas com o desenvolvimento deste e descreve os trabalhos futuros.

Os Apêndices A e B apresentam, respectivamente, as representações contextuais criadas e a especificação do Gerente de Contexto detalhado no Capítulo 3.

Computação Ciente de Contexto

O termo **Computação Ubíqua** foi introduzido pelo pesquisador Mark Weiser em 1991 para caracterizar ambientes incrementados com recursos computacionais capazes de prover serviços e informações quando e onde sejam desejados. De acordo com Weiser (1993) deve haver integração contínua entre tecnologia e ambiente de modo a auxiliar os usuários em atividades cotidianas. Portanto, o ambiente deve ser dotado de recursos implícitos capazes de se adaptarem aos desejos e objetivos dos usuários. A interação usuário-computador deve ocorrer de modo não intrusivo, o mais semelhante possível à forma natural de interação entre os seres humanos.

No entanto, para alcançar uma interação usuário-computador permanente, Abowd et al. (2002) relatam a necessidade de compreensão dos hábitos das pessoas, proliferação de dispositivos heterogêneos com interações distintas e o gerenciamento dos dispositivos conectados em rede. Com isso, o desenvolvimento de novas aplicações capazes de explorar o uso desses dispositivos vem crescendo nos últimos anos. Assim, três vertentes de pesquisa objetivaram o desenvolvimento de aplicações capazes de melhorar a interação usuário-computador: interfaces naturais, captura e acesso de atividades humanas e aplicações cientes de contexto.

As interfaces naturais correspondem à tentativa de adaptação das interfaces computacionais à forma de interação natural que ocorre entre pessoas. Por meio do suporte a formas comuns de expressão humana, as interfaces naturais facilitam a capacidade de comunicação entre usuários e computadores utilizando ações explícitas ou implícitas durante a comunicação. Desse modo, a interação usuário-computador torna-se não intrusiva, ideal para computação ubíqua (Abowd & Mynatt, 2000).

Sendo assim, projetos voltados para interfaces mais amigáveis investigam técnicas de interação natural. É o caso do projeto desenvolvido por Ishii & Ullmer (1997), no qual objetos do mundo físico servem como meio de manipulação de artefatos eletrônicos, ou o projeto de sensores atrelados a dispositivos computacionais capazes de manipular fisicamente tais dispositivos (Harrison et al., 1998), ou ainda, a pesquisa de Schilit et al. (1994) que consiste na utilização de escrita manual como interação.

O objetivo da área de captura e acesso automatizados de atividades humanas é prover a obtenção e posteriormente o acesso de informações referentes a atividades efetuadas pelos usuários por meio de recursos computacionais. Como exemplo de atividades podem ser citadas reuniões ou palestras. O armazenamento simultâneo da atividade é garantido graças aos diferentes tipos de mídia, os quais também colaboram para o acesso posterior ao conteúdo capturado, preservando as relações entre eles. Exemplos de pesquisas efetuadas na área são projetos de captura e acesso de informações provenientes de vídeos, áudios e lousas eletrônicas durante aulas, o *eClass* (Abowd, 1999), desenvolvido no GATech (*Georgia Institute of Technology*) e durante reuniões, como o projeto Tivoli (Minneman et al., 1995), desenvolvido nos laboratórios da Xerox PARC.

A área de computação ciente de contexto é responsável pelo enriquecimento da interação usuário-computador. O objetivo é alcançar um aproveitamento eficaz da interação por meio de informações contextuais presentes na comunicação. A idéia é prover serviços personalizados ou serviços que se adaptem às necessidades do usuário respeitando-se as condições da infra-estrutura do sistema. Como exemplo de pesquisa relevante na área pode ser citado um dos projetos precursores em computação ciente de contexto, o *Active Badge* (Want et al., 1992), desenvolvido no *Olivetti Research Lab's*. Esse projeto é capaz de rastrear pessoas com crachás diferenciados, por meio de sensores espalhados em todo o ambiente. Além do *Active Badge*, podem ser citados também o *Context Kernel* (Arruda Jr, 2003) e o Protótipo de TV Interativa (Goularte, 2003), ambos desenvolvidos no ICMC-USP. O primeiro é um serviço Web responsável pelo armazenamento e recuperação de informações contextuais utilizando a infra-estrutura fornecida pela Internet por meio de *Web Services* (W3C, 2002), o segundo, o Protótipo de TV Interativa tem por objetivo efetuar a personalização de vídeos baseada em informações de contexto.

No intuito de aprofundar a discussão na área de computação ciente de contexto, vertente abordada por este trabalho, as seções seguintes fornecem uma revisão da literatura relacionada a esse tópico. As Seções 2.1 e 2.2 apresentam as definições de contexto e de computação ciente de contexto, respectivamente. A Seção 2.3 apresenta a categorização das características relevantes

para aplicações cientes de contexto. Na Seção 2.4 são abordadas as classificações de contexto mencionadas na literatura. A Seção 2.5 apresenta os requisitos para o desenvolvimento de aplicações cientes de contexto. Na Seção 2.6 são discutidos alguns trabalhos relacionados ao serviço de gerenciamento de contexto. A Seção 2.7 aborda a representação de contexto e finalmente, a Seção 2.8 descreve as considerações finais deste capítulo.

2.1 Definição de Contexto

À medida que há interação entre pessoas, informações são obtidas tanto de modo explícito, através da fala, por exemplo, quanto de modo implícito, por meio de gestos ou expressões faciais. As informações implícitas, ou seja, informações contextuais, auxiliam a interação, uma vez que as pessoas são capazes de usá-las para incrementar e aperfeiçoar a interação, enriquecendo-a.

Porém, na interação usuário-computador, informações contextuais raramente são usadas. Conseqüentemente, serviços que poderiam ser fornecidos de um modo mais eficiente utilizando-se informações de contexto, por exemplo, a entrega personalizada de conteúdo ou a adaptação de apresentação e até mesmo conteúdo adaptado às condições momentâneas dos recursos disponíveis, não são oferecidos. Contudo, considerando-se a interação usuário-computador, o que pode ser contexto? Quais aspectos devem ser considerados para a classificação de uma informação como sendo referente ao contexto?

A definição pioneira de contexto foi apresentada por Schilit et al. (1994), os quais referem-se ao contexto como localização, identificação de pessoas e objetos próximos à interação, bem como as mudanças nesses objetos. Brown et al. (1997) determinam contexto como localização, identificação de pessoas ao redor do usuário, situação, temperatura, etc. Já Ryan et al. (1997) definem contexto como a localização do usuário, ambiente, identidade e tempo. Salber et al. (1998) referem-se ao contexto como informações pessoais ou referentes a dispositivos que podem colaborar para a melhoria no fornecimento de serviços através de transformações nos mesmos. Como exemplos de informações desse tipo podem ser citados dados emocionais, históricos de localização, e finalmente, dados de foco de atenção do usuário. Por sua vez, Pascoe (1998) determina contexto como o subgrupo de estado conceitual e físico relevante a uma entidade.

Embora as definições apresentadas sejam relevantes, Dey & Abowd (2000) e Dey (2001) relatam que elas são excessivamente específicas. Como são definições baseadas em exemplos, todas são extremamente difíceis de serem aplicadas na prática. Ainda, para esses autores, contexto é todo tipo de infor-

mação importante em uma situação, tanto para uma aplicação quanto para seu conjunto de usuários. Não há a possibilidade de enumerar quais aspectos de situações são importantes, pois os mesmos mudam de situação para situação. Em alguns casos, por exemplo, o ambiente físico pode ser importante, enquanto que em outros essa mesma informação pode ser completamente irrelevante.

Depois de analisar as características relevantes e as lacunas das definições prévias de contexto, Dey (2001) formalizou uma definição mais geral de contexto. Essa definição é amplamente aceita atualmente pela comunidade da área de computação ciente de contexto:

"Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto considerado relevante para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação em si."

A definição de Dey (2001) facilita o trabalho do projetista de decidir quais informações podem ser consideradas contextuais e o torna apto a enumerá-las dado um cenário de aplicação. Se uma determinada informação pode ser usada para caracterizar a situação de um participante em uma interação, então tal informação é contexto (Dey & Abowd, 2000; Dey, 2001). Um exemplo de situação que ilustra o uso de informações de contexto é a consulta a informações de localização e de preferências de usuários por um guia turístico, a fim de adequar o passeio dos turistas a pontos turísticos de interesse dos mesmos.

Para auxiliar o trabalho dos desenvolvedores de aplicações cientes de contexto de eleger quais informações realmente são relevantes para a interação usuário-computador, ou seja, de definir o conjunto de informações contextuais, Dey & Abowd (2000) propõem a categorização de informações contextuais em cinco dimensões semânticas. Tais dimensões, conhecidas como os cinco Ws semânticos, são as diretrizes **quem** (*who*), **onde** (*where*), **o quê** (*what*) e **quando** (*when*) que são relacionadas a uma determinada entidade de modo a obter a razão, o **porquê** (*why*) da situação ocorrida num determinado momento (Abowd & Mynatt, 2000). Geralmente a diretriz *what* pode ser combinada com uma sexta diretriz **como** (*how*) para inferir informações sobre *why*. Considerando-se o domínio de aplicações de captura e acesso, a diretriz **como** (*how*) torna-se relevante por definir o modo de captura dos dados, bem como de acesso aos mesmos (Truong et al., 2001).

Há certos tipos de informações contextuais mencionados na literatura (Schilit et al., 1994; Brown et al., 1997; Pascoe, 1998; Dey & Abowd, 2000; Dey,

2001) chamados de **contexto primário** que, na prática, são mais importantes que outros. Tais tipos de informações de contexto compreendem informações de **localização, identidade, atividade e tempo**. São mais relevantes porque além de responderem às questões quem, onde, o quê e quando, atuam como indicadores para fontes de informações de contexto. Por meio da identidade de uma pessoa, por exemplo, é possível descobrir um conjunto de informações referentes a essa pessoa, como número de telefone, endereço, e-mail, etc.

2.2 Definição de Computação Ciente de Contexto

O termo "computação ciente de contexto" foi utilizado pioneiramente por Schilit & Theimer (1994) para definir "aplicações capazes de efetuar adaptação de acordo com sua localização de uso, grupo de pessoas e objetos nas proximidades e ainda, considerando as mudanças ocorridas com tais objetos no decorrer do tempo". Embora o termo tenha sido disseminado pelos autores citados, a primeira pesquisa em computação ciente de contexto foi o *Active Badge* (Want et al., 1992), desenvolvido em 1992 no *Olivetti Research Lab's*. Desde então inúmeras tentativas de definição de computação ciente de contexto foram feitas.

Além de Schilit & Theimer (1994), outros pesquisadores se aventuraram a definir computação ciente de contexto. Hull et al. (1997) e Pascoe (1998) abordam computação ciente de contexto como habilidade dos dispositivos em perceber, interpretar e reagir a aspectos do ambiente que cerca o usuário e os próprios dispositivos. Outro exemplo de definição é a apresentada por Ryan et al. (1997) que se referem à ciência de contexto como a aptidão de compreender e atuar com base em informações do usuário, seus interesses e atividades, e também em condições do ambiente.

Fundamentados nas definições prévias de computação ciente de contexto, Dey & Abowd (2000) qualificam-nas em dois grupos: definições que requerem a modificação do comportamento de uma aplicação de modo a considerá-la ciente de contexto (Schilit & Theimer, 1994; Ward et al., 1997; Ryan et al., 1997; Brown et al., 1997; Korteum et al., 1998) e definições que requerem que o sistema seja perceptível ao contexto, além de interpretá-lo e ser capaz de produzir uma resposta ao mesmo (Hull et al., 1997; Pascoe, 1998; Salber et al., 1998). Portanto, com o objetivo de suprir as deficiências demonstradas pelas definições anteriores, Dey & Abowd (2000) e Dey (2001) propõem uma definição mais abrangente:

"Um sistema é ciente de contexto se usa contexto para fornecer informação e/ou serviço relevante para o usuário, onde a relevância depende da tarefa do

usuário.”

Efetuando-se uma comparação entre a definição de Dey & Abowd (2000) e Dey (2001) e a definição dos pesquisadores Hull et al. (1997) e Pascoe (1998), pode-se dizer que a primeira é mais genérica que a segunda. Ambos requerem que aplicações cientes de contexto sejam capazes de detectar, interpretar e responder a informações de contexto. Em contrapartida, Dey & Abowd (2000) e Dey (2001) requerem apenas a resposta ao contexto, aspecto que permite detecção e interpretação de informações de contexto por outras entidades computacionais. Finalmente, a definição de Dey & Abowd (2000) e Dey (2001) difere-se das demais por estar focada no usuário, não limitar a ciência à interface das aplicações; não requerer que aplicações executem serviços automaticamente e não requerer a aquisição de contexto em tempo real.

2.3 *Categorização de Características para Aplicações Cientes de Contexto*

A literatura apresenta algumas propostas de categorização das características para aplicações cientes de contexto. As mais citadas são as propostas de Schilit et al. (1994), Pascoe (1998) e de Dey & Abowd (2000). Embora tais propostas utilizem nomeações distintas para a categorização das mesmas características, essas características podem ser agrupadas sem perda semântica em:

- **Percepção contextual:** capacidade de detectar e apresentar informações contextuais ao usuário.
- **Adaptação contextual:** capacidade de executar ou modificar um serviço automaticamente baseado nas informações contextuais presentes. Essa característica descreve casos nos quais o contexto é responsável por uma ação e casos nos quais o contexto é usado para adaptar serviços. Chalmers (2002) classifica esses casos em duas categorias:
 - **Ações disparadas pelo contexto:** dado um determinado conjunto de informações contextuais que atinge valores específicos, uma ação é disparada.
 - **Mediação contextual:** adaptação de serviços e dados de acordo com os limites e preferências impostos pelo contexto. A requisição de serviços ou dados não deve ser necessariamente ocasionada pelo contexto.

- **Descoberta de recursos:** permite que aplicações cientes de contexto sejam capazes de encontrar e consumir recursos e serviços relevantes para um determinado contexto.
- **Associação de informação contextual:** permite associar informação contextual a dados. Por exemplo, em uma reunião, os participantes podem ser associados à ata da mesma.

Uma proposta mais recente organizada por Dey & Abowd (2000) combina ambas as taxonomias de Schilit et al. (1994) e Pascoe (1998), além de considerar suas distinções. Análoga à taxonomia de Pascoe (1998), Dey & Abowd (2000) apresentam aspectos suportados por aplicações cientes de contexto:

- Apresentação de informações e serviços ao usuário;
- Execução automática de serviços;
- Identificação de informações contextuais para recuperação posterior.

Além de apresentar as características citadas, Dey & Abowd (2000) efetuam um comparativo dessas características com as propostas prévias. A primeira característica apresentada pelos autores é uma combinação de características presentes no trabalho de Schilit et al. (1994), seleção aproximada e comando contextual, além da característica de apresentação das informações contextuais ao usuário, proposta por Pascoe (1998). O aspecto de execução automática de serviços está associado à definição de ações focadas em informações de contexto de Schilit et al. (1994) e à adaptação contextual proposta por Pascoe (1998). A terceira tarefa está relacionada à definição de associação de informação contextual apresentada por Pascoe (1998).

A categorização de aspectos para aplicações cientes de contexto é relevante por especificar os tipos de aplicações que as arquiteturas devem suportar. Além disso, a categorização pode ser empregada com o objetivo de orientar projetistas em relação às características consideradas no desenvolvimento de aplicações cientes de contexto.

2.4 Classificação de Contexto

Dey & Abowd (2000) e Dey (2001) empregam contexto para prover, ao usuário, informações e/ou serviços relevantes dependendo da tarefa exigida por ele. Essa definição é referente, portanto, ao contexto do Usuário. No entanto, propostas como a de Schilit (1995), Dix et al. (2000), Chalmers (2002) e Crowley et al. (2002) classificam os conjuntos de informações utilizados em aplicações cientes de contexto em tipos contextuais distintos em adição ao contexto

descrito por Dey & Abowd (2000) e Dey (2001): **Infra-Estrutura, Sistema, Domínio e Ambiente**.

Segundo Schilit (1995) e Dix et al. (2000), o **contexto da Infra-Estrutura** é necessário para representação do estado atual da comunicação entre a aplicação e o dispositivo utilizado para o acesso a ela. Com tal representação, podem-se obter subsídios para que seja diagnosticado qualquer tipo de mudança ocorrida nas condições da comunicação, como transição de dispositivos no ambiente e mudanças de estado devido a falhas.

O **contexto do Sistema** é responsável por prover informações que permitam aos dispositivos estarem cientes dos outros dispositivos que se encontram nas proximidades e, relacionado à ciência dos dispositivos, quais aplicações estão cientes de outras aplicações que também oferecem serviços aos usuários (Dix et al., 2000; Crowley et al., 2002; Chalmers, 2002). Esse aspecto é importante porque os dispositivos podem afetar o funcionamento uns dos outros devido à competição por recursos. Outro fator relevante também é a combinação dos dispositivos, que pode resultar em serviços mais avançados. Os autores também relatam que tal contexto é composto pelo contexto do Usuário acrescido ao estado momentâneo de dispositivos e serviços utilizados. No entanto, tal argumentação não apresenta uma independência clara entre contextos do Sistema e da Infra-Estrutura, pois ambos empregam aplicações e dispositivos.

O **contexto do Domínio** fornece informações referentes à semântica do domínio da aplicação. Para tal é necessário considerar os relacionamentos entre dispositivos e seus usuários e como essas relações podem ser usadas para determinar a natureza das interfaces ou serviços sendo apresentados. Por exemplo, num quarto de hospital, a relação entre a especialidade do médico e o paciente pode determinar o tipo de informação a ser apresentada. Outra particularidade importante do contexto do Domínio é o nível de privacidade e de segurança da informação em situações nas quais informações sobre um usuário podem se tornar disponíveis para outros usuários (Dix et al., 2000; Abowd et al., 2002).

Dix et al. (2000) relatam que o **contexto do Ambiente** deve informar sobre características do ambiente onde uma determinada entidade se encontra. Exemplos de características do ambiente são: luminosidade e temperatura. O contexto do Ambiente abrange tanto ambientes reais quanto virtuais.

Embora as classificações apresentadas nesta seção sejam relevantes, seus autores não evidenciam uma tentativa clara de separação entre os tipos de contextos mencionados, nem sequer o mapeamento das aplicações cientes de contexto já existentes nessas classificações. Por exemplo, Dix et al. (2000) trabalham no desenvolvimento de uma taxonomia para localização e espaço,

características presentes no contexto do Ambiente.

2.5 Requisitos para Desenvolvimento de Aplicações Cientes de Contexto

Para construção de aplicações cientes de contexto e de infra-estruturas capazes de suportar tais aplicações, Dey (2000) apresenta em seu trabalho um conjunto de requisitos que podem ser seguidos durante o desenvolvimento de ambas. Tais requisitos são derivados principalmente de aspectos comuns de aplicações cientes de contexto. O conjunto de requisitos definido por Dey (2000) é descrito nas subseções apresentadas a seguir. São eles: Especificação de Informações de Contexto, Separação de Aquisição e de Utilização de Informações de Contexto, Interpretação das Informações de Contexto, Comunicação Distribuída e Transparente, Disponibilidade Contínua dos Componentes de Captura de Informações de Contexto, Armazenamento de Informações de Contexto e finalmente, Descoberta de Recursos.

2.5.1 Especificação de Informações de Contexto

Um mecanismo capaz de permitir a especificação de tipos de informações de contexto que podem ser requisitados deve estar presente. Tal mecanismo deve permitir que os desenvolvedores expressem informações contextuais em fragmentos simples ou múltiplos, bem como a existência de relacionamentos entre essas informações. Além disso, esse mecanismo deve permitir a notificação da disponibilidade da informação contextual à aplicação como também a notificação ao usuário, por parte da aplicação, da necessidade de modificar a especificação das informações de contexto. Por fim, o mecanismo deve prover suporte a múltiplas especificações originárias de múltiplas aplicações que requisitaram informação de contexto.

2.5.2 Separação de Aquisição e de Utilização de Informações de Contexto

Não há uma padronização relacionada à aquisição e à manipulação de informações contextuais. Essas informações são obtidas e manipuladas sem uma metodologia, no improviso, pois os desenvolvedores preferem utilizar técnicas mais triviais de implementação, abdicando da generalização e do reuso. Técnicas de gerenciamento de eventos como mecanismos de consulta (técnica de *polling*), ou ainda, mecanismos de notificação (técnica de *callback*) podem ser utilizados no intuito de aquisição das informações de contexto. Não há

necessidade que ambos sejam implementados. Contudo, é importante que aplicações cientes de contexto sejam beneficiadas por ambos, por razões de flexibilidade.

Depois de adquiridas as informações de contexto, cabe à aplicação determinar se tais informações foram alteradas e ainda, se as mesmas são relevantes. A relevância desse requisito consiste em separar o processo de aquisição do processo de consumo das informações de contexto. Dessa forma, aplicações podem consumir essas informações sem se preocupar com o modo como tais informações foram adquiridas.

2.5.3 *Interpretação das Informações de Contexto*

O processo de aquisição das informações de contexto é composto por múltiplas camadas. Considerando-se a perspectiva dos desenvolvedores essas camadas devem ser transparentes e, para isso, as informações de contexto devem ser interpretadas antes de sua utilização pelas aplicações. Por exemplo, o nome de um usuário pode ser usado para a interpretação de um determinado gênero de programa de TV. A combinação de mecanismos capazes de interpretar informações contextuais com mecanismos de especificação das mesmas permite a requisição de informações de contexto desejadas por aplicações e sua recuperação na forma interpretada quando necessário.

2.5.4 *Comunicação Distribuída e Transparente*

A captura de informações contextuais é efetuada por sensores que não estão, em sua maioria, conectados ao computador que os utiliza. Esses sensores, geralmente, estão distribuídos fisicamente em um ambiente onde ocorre interação. Sendo assim, há a necessidade de transparência na comunicação distribuída, tanto para sensores quanto para aplicações cientes de contexto. Sem esse aspecto, os desenvolvedores são forçados a especificar e implementar um protocolo de comunicação e um esquema de codificação e decodificação para transmitir as informações de contexto.

2.5.5 *Disponibilidade Contínua dos Componentes de Captura de Informações de Contexto*

Aplicações cientes de contexto são imprevisíveis no sentido de que podem solicitar informações contextuais a qualquer instante. Portanto, componentes responsáveis pela captura de informações de contexto devem estar sempre ativos, prontos para disponibilizá-las quando requisitadas. Além disso, devem exercer seu papel de modo independente das aplicações que os executam.

Desse modo, tais componentes podem atender a várias aplicações que necessitem das mesmas informações contextuais.

2.5.6 Armazenamento de Informações de Contexto

A importância de manter informações históricas é um requisito relacionado à necessidade de disponibilidade contínua dos componentes responsáveis pela captura de informações de contexto. Baseadas em históricos de contexto, as aplicações são capazes de pressupor valores de contexto futuros, ou ainda, estabelecerem tendências. Sem o armazenamento de informações de contexto, essas ações não seriam possíveis. Logo, para manter um histórico contextual, os componentes devem estar aptos a capturar informações contextuais a todo o momento, mesmo quando essas informações não forem requeridas por nenhuma aplicação ciente de contexto. Portanto, tais informações de contexto podem ser utilizadas por uma aplicação que necessite não apenas dos valores momentâneos dessas informações, mas também do histórico das mesmas. Por exemplo, uma aplicação pode utilizar o histórico de localização de um usuário para prever sua próxima localização no ambiente.

2.5.7 Descoberta de Recursos

Dispositivos de captura de informações de contexto devem especificar quais tipos de informações de contexto provêm, bem como sua localização e como aplicações cientes de contexto devem se comunicar com eles. A função do mecanismo de descoberta de recursos é encontrar componentes adequados e fornecer mecanismos de acesso a eles. O mecanismo de descoberta de recursos pode ser combinado com o mecanismo de especificação e com os componentes de captura. O intuito é determinar quais situações podem ser capturadas, ou ainda, se uma determinada requisição de informação de contexto pode ser realizada pela infra-estrutura em execução. Além disso, pode ser utilizado por aplicações para determinar se a informação de contexto está disponível no ambiente.

2.6 Trabalhos Relacionados

Ao construírem as primeiras aplicações cientes de contexto, os desenvolvedores foram forçados a construir também os mecanismos de captura, armazenamento e processamento de informações de contexto dessas aplicações. Porém, há aplicações que utilizam serviços especializados, ou sejam, aplicações baseadas em *toolkits*, infra-estruturas ou *middlewares*.

Os *toolkits* são baseados em *frameworks* e são capazes de oferecer um conjunto de componentes reusáveis para diferentes funcionalidades. As infraestruturas consistem de conjuntos de tecnologias bem definidas, confiáveis e publicamente acessíveis que atuam como suporte a outros sistemas. Os *middlewares* são interfaces que permitem que desenvolvedores de aplicações possam acessar serviços distribuídos sem considerar os protocolos de comunicação utilizados, como dados são gerenciados e se há balanceamento de carga, por exemplo.

Nas subseções a seguir são apresentadas algumas aplicações cientes de contexto que implementam seus serviços de captura, armazenamento e gerenciamento de informações de contexto, bem como exemplos de *toolkit* e infraestrutura capazes de suportar aplicações cientes de contexto.

2.6.1 *Active Badge*

Uma das aplicações pioneiras em ciência de contexto, a *Active Badge* (Want et al., 1992) foi uma solução proposta com o intuito de minimizar o problema de localização de um indivíduo no ambiente. Foi desenvolvido um sistema que combina detecção de posicionamento e comunicação para o monitoramento de indivíduos. O sistema foi usado para auxiliar telefonistas a encontrarem indivíduos no ambiente. Por meio de sensores acoplados em crachás e instalados em escritórios e áreas comuns de um determinado edifício, as telefonistas recebiam uma lista com nomes de usuários e números de telefones referentes aos lugares mais próximos de cada um desses usuário. Quando era detectada a movimentação de um usuário pelo edifício, o sistema atualizava a listagem, garantindo uma lista consistente com os telefones mais próximos desse usuário. Embora tenha sido preciso, o projeto *Active Badge* não teve continuidade devido ao seu alto custo.

2.6.2 *Cyberguide*

O sistema *Cyberguide* (Long et al., 1996; Abowd et al., 1997) atua como um guia turístico no espaço físico de um laboratório de pesquisa. Uma das aplicações cientes de contexto mais comuns, o *Cyberguide* é formado por *handhelds* que se comunicam com sensores posicionados em todo o ambiente. Os *handhelds* apresentam aos usuários um mapa referente à área do laboratório onde ele está destacando locais de interesse para se visitar; além de disponibilizar outras informações relacionadas a esses locais. Com a movimentação do usuário pelo laboratório, o dispositivo altera a interface para o novo posicionamento, atualizando os pontos de interesse e fornecendo, ainda, as respectivas informações sobre os os novos pontos.

2.6.3 Context Toolkit

Consiste da implementação do *framework* conceitual apresentado por Dey (2000). O *Context Toolkit* foi desenvolvido para prover suporte a aplicações cientes de contexto (Salber et al., 1999; Dey et al., 1999). Sua arquitetura é formada por três tipos de componentes de informações contextuais que se comunicam via XML sobre HTTP (*Hypertext Transfer Protocol*). São os *wid-gets*, interpretadores e agregadores, demonstrados na Figura 2.1 e descritos a seguir.

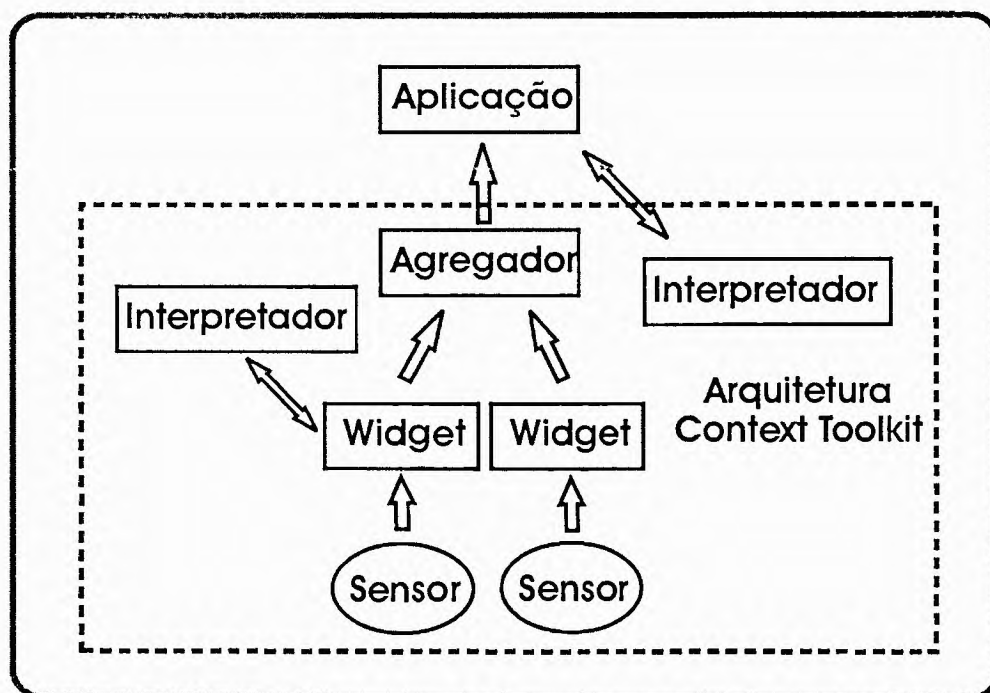


Figura 2.1: Arquitetura do *Context-Toolkit*. Adaptado de (Salber et al., 1999).

Os componentes **wid-gets** têm a função de prover interação entre sensores e componentes ou aplicações por meio da captura e disponibilidade de informações contextuais. As informações capturadas são disponibilizadas via *polling* ou *callback*. Os **interpretadores** são responsáveis pela inferência de informações contextuais de alto nível baseadas em informações de contexto de baixo nível. Essa inferência pode ser o resultado da combinação de um ou mais tipos de informações de contexto. Por exemplo, informações referentes à identificação, localização e tempo podem ser combinadas resultando na inferência de que uma aula está ocorrendo. Por fim, os **agregadores** são semelhantes aos **wid-gets**, porém, são capazes de agregar informações de contexto a uma entidade do mundo real. Por exemplo, agrupar e associar informações de localização e identificação a uma entidade do tipo usuário.

O *Context Toolkit* fornece serviços como descoberta de recursos, captura, armazenamento, conversão, agregação, interpretação, acesso e distribuição

de informações contextuais. Dey (2000) relata em seu trabalho algumas aplicações beneficiadas pelo uso do *Context Toolkit*. Por exemplo, a *Context-Aware Mailing List*, aplicação capaz de enviar mensagens de e-mail a membros de um grupo de pesquisa. O diferencial dessa aplicação é a característica de enviar mensagens apenas às pessoas que estão no laboratório de pesquisa. Outro exemplo é a aplicação *In/Out Board*, a qual apresenta num *display* os pesquisadores que estão ou não no prédio de pesquisas (Salber et al., 1999).

2.6.4 *Context Fabric*

Fornece quatro tipos de serviços com o intuito de reduzir o esforço durante o desenvolvimento de aplicações cientes de contexto: gerenciamento de sensores, criação automática de caminho, serviço de eventos e serviços de consulta. Além dos serviços, o *Context Fabric* também é composto por uma linguagem declarativa de informações contextuais (Hong & Landay, 2001).

O serviço de gerenciamento de sensores é responsável pelo registro e pela gerência dos sensores locais usados pelos outros serviços do *Context Fabric*. Além disso, provê aos serviços de consulta e eventos a padronização dos dados capturados por sensores. O serviço de criação automática de caminho agrega o fluxo de dados de sensores aos componentes de software necessários para refinar os dados originários dos sensores em informações de contexto de alto nível. Se alguns serviços provêm informações referentes à localização em um determinado ambiente, o serviço de criação automática de caminho é o responsável pelo monitoramento das informações provenientes desses serviços para que o evento desejado possa ser diagnosticado.

O serviço de eventos é responsável pelo registro de eventos relevantes às aplicações em um ambiente ciente de contexto por meio de *callback*. O objetivo é notificar de modo assíncrono as aplicações que requisitarem algum evento a ser ocorrido. Desse modo, as aplicações não têm que se preocupar com a obtenção de dados. Finalmente, o serviço de consulta permite que aplicações obtenham informações de contexto via *polling*.

A linguagem declarativa de especificação de informações contextuais denominada CSL (*Context Specification Language*) utiliza o padrão XML (*xtensible Markup Language*) (Bray et al., 2000) para declaração de informações contextuais e eventos por parte das aplicações. O padrão XML é utilizado para a comunicação entre serviços de eventos e de consulta e as aplicações cientes de contexto. A infra-estrutura *Context Fabric* separa a aquisição de informações contextuais do armazenamento, processamento, conversão e acesso das mesmas por parte das aplicações.

2.6.5 Context Kernel

Infra-estrutura de software baseada em *Web Service* (W3C, 2002) que permite a manipulação de informações contextuais modeladas com base nas dimensões de contexto *who*, *where*, *when*, *what* e *how*, descritas na Seção 2.2. Um dos benefícios obtidos com o uso da tecnologia de *Web Service* é a interoperabilidade entre aplicações, devido à comunicação transparente entre aplicações e serviços independentemente da heterogeneidade de tecnologias utilizadas para suas implementações. O *Context Kernel* formaliza as informações contextuais em dimensões primitivas e dimensões derivadas.

As dimensões primitivas são triplas no formato (TIPO, VALOR, QUALIFICADOR). Por exemplo, a tripla (TIPO= data, VALOR = 2004-04-19T09:19:52, QUALIFICADOR = time). As dimensões derivadas são determinadas via regras que contêm no mínimo uma premissa e uma inferência. Premissas são quádruplas no formato dimensão-tipo-valor-qualificador e inferências são pares dimensão-valor. As regras são inferidas pelas aplicações e armazenadas no *Context Kernel* para consultas. Por exemplo, uma regra pode ser baseada nas premissas: (dimensão=where, tipo=latitude, valor=23) e (dimensão=where, tipo=longitude, valor=46). Essas premissas resultam na inferência: (dimensão=what, valor=cidade de São Paulo) (Arruda Jr, 2003; Arruda Jr et al., 2003).

A Figura 2.2 ilustra o fluxograma do *Context Kernel*. O objetivo é prover serviços de registro, armazenamento, consulta e recuperação de informações de contexto (Arruda Jr, 2003; Arruda Jr et al., 2003).

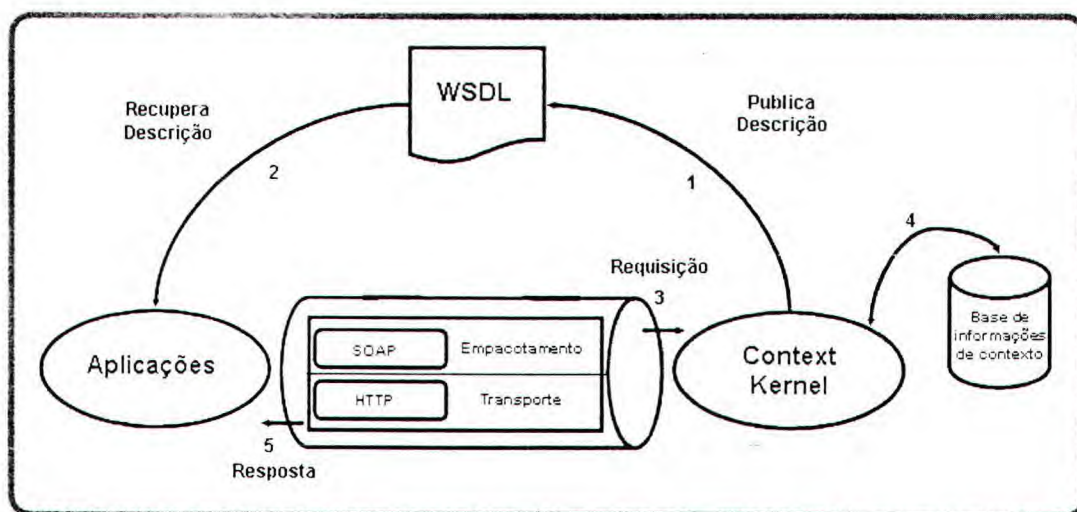


Figura 2.2: Fluxograma do *Context Kernel*. Adaptado de (Arruda Jr, 2003)

O registro de informações é o serviço responsável por registrar aplicações que serão beneficiadas pelo *Context Kernel*. Para tal, devem ser fornecidos

à infra-estrutura o nome da aplicação, sua descrição, última versão, endereço na Web, entre outras informações. A aplicação recebe como resposta dois identificadores, um público e outro privado, necessários a outros serviços fornecidos pelo *Context Kernel*. O serviço de armazenamento permite que informações contextuais sejam armazenadas de forma assíncrona. Essa característica provê liberdade às aplicações cientes de contexto devido à possibilidade de efetuar tal operação à medida que as informações são capturadas. Para que o armazenamento possa ser executado é necessário o uso do identificador privado da aplicação fornecido durante o registro da mesma.

O serviço de recuperação de informações de contexto utiliza o identificador público. Provê a recuperação de informações referentes a uma única aplicação, as quais foram fornecidas durante o registro da aplicação, a recuperação das informações de todas as aplicações cadastradas no *Context Kernel* e, por fim, a recuperação das informações contextuais armazenadas pelas aplicações registradas. Finalmente, o serviço de consultas às informações contextuais da infra-estrutura permite o intercâmbio de informações de contexto entre aplicações. Com o serviço de consultas, uma aplicação é capaz de armazenar suas informações de contexto e disponibilizar seu identificador público para que outras aplicações sejam capazes de consultá-las.

2.7 Representação de Contexto

No desenvolvimento de aplicações cientes de contexto, tão complexo quanto apontar informações relevantes no contexto de uma interação usuário-computador é escolher a melhor forma de representá-las. Segundo Abowd et al. (2002), sem representações de contexto consistentes e satisfatórias, os desenvolvedores são forçados a implementar estruturas limitadas para o armazenamento e a manipulação das informações, prejudicando o intercâmbio das mesmas.

Várias tentativas foram feitas em busca do aprimoramento da representação de contexto. A definição de contexto (Dey, 2000) e as cinco dimensões contextuais (**who**, **where**, **when**, **what** e **how**) apresentadas na seção 2.1 contribuem para o direcionamento da relevância de determinadas informações contextuais obtidas durante a interação usuário-computador. No entanto, mesmo seguindo a definição de contexto e as cinco dimensões, o problema de representação ainda persiste. Há, ainda, a necessidade de analisar os requisitos da aplicação e o modo como a modelagem e a representação das informações contextuais serão efetuadas em cada dimensão.

Alguns trabalhos exploraram a representação de contexto fornecendo apoio às abstrações de informações contextuais que possam ser utilizadas por apli-

cações cientes de contexto. O *Context Toolkit* (Salber et al., 1999), o *Context Fabric* (Hong & Landay, 2001) e o *Context Kernel* (Arruda Jr, 2003) são exemplos de trabalhos desse tipo. Como explicado na Subseção 2.6.3, o *Context Toolkit* é uma ferramenta que fornece componentes abstratos (*widgets*, interpretadores e agregadores) que se conectam para processar informações contextuais provenientes de sensores. O *Context Fabric* utiliza a linguagem própria, CSL, como solução para a declaração de informações contextuais (Hong & Landay, 2001). Já o *Context Kernel* (Arruda Jr, 2003) utiliza dimensões primitivas e derivadas para formalizar a modelagem da representação de informações de contexto.

Outros projetos são focados em modelar as informações contextuais e entregá-las às aplicações cientes de contexto. É o caso do projeto desenvolvido por Schilit et al. (1994), que utiliza servidores dinâmicos de ambiente para gerenciar e distribuir informações de contexto. O projeto *Cooltown* representa contexto por meio de páginas HTML (*Hypertext Markup Language*) que simbolizam entidades contextuais. O problema do uso desse tipo de representação é que as informações são desestruturadas e desenvolvidas para o uso humano, característica que dificulta o uso por aplicações. Já o projeto *Sentient Computing* (Harter et al., 1999) faz uso de um modelo Entidade-Relacionamento para representar contexto. Nesse projeto as informações contextuais são armazenadas numa base de dados relacional. Crowley et al. (2002) representam contexto como um conjunto de variáveis que representam o estado do sistema.

A maioria dos trabalhos mencionados apresentam lacunas como a falta de flexibilidade para compartilhamento de informações de contexto entre aplicações. Outra limitação é a ausência de uma classificação dos tipos de contexto de modo a obter uma representação estruturada e abrangente das informações contextuais. Atentando-se à essas limitações, Goularte (2003) classifica e organiza as informações contextuais com base na argumentação de Santos Jr (2002), de modo a suprir as deficiências encontradas nos trabalhos citados. Segundo Santos Jr (2002), informações contextuais podem auxiliar o gerenciamento de interações em ambientes que incluam mobilidade. O autor ainda levanta questões pertinentes relacionadas à caracterização de um ambiente, à descoberta de serviços disponíveis num determinado ambiente e como oferecê-los considerando-se dispositivos disponíveis e a preferência do usuário.

De acordo com Goularte (2003), efetuando-se todo esse questionamento, pode-se notar que a definição de contexto de Dey (2000) é insuficiente, pois restringe informações contextuais apenas às interações entre usuário e aplicação. Para responder às questões levantadas por Santos Jr (2002), faz-se necessário o conhecimento do contexto da situação das aplicações, do estado

momentâneo da rede e da situação dos diversos tipos de interação que podem ocorrer entre as aplicações e o sistema, além do contexto do Usuário (Santos Jr et al., 2001).

Desse modo, Goularte (2003) aborda em seu trabalho a necessidade de estender a definição apresentada por Dey (2000), para que a mesma possa ser aplicada em ambientes interativos de um modo mais abrangente explorando assim as interações entre aplicações e o sistema ao invés de apenas o contexto do usuário e da aplicação:

*Contexto é qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto **físico ou software** que é considerado relevante para a interação entre um usuário e uma aplicação, **e/ou entre uma aplicação e o sistema**, incluindo usuário, aplicação e **sistema em si.***

De acordo com o autor, considerando-se a definição estendida e a classificação de contexto mencionada na Seção 2.4, é possível classificar os tipos de contexto em: *Infra-Estrutura, Sistema, Usuário, Aplicação, Domínio e Ambiente*. No entanto, para que possa ser utilizada no desenvolvimento de uma representação contextual, essa classificação deve ser organizada.

A Figura 2.3 ilustra a organização proposta por Goularte (2003). É conveniente salientar que os tipos de contextos do Domínio e do Ambiente foram omitidos, embora tenham sido classificados, pois o presente trabalho não anseia explorar tais contextos. Portanto, a organização apresentada por Goularte (2003) será integralmente respeitada.

Os contextos de Usuário e da Aplicação são ilustrados pelos retângulos devido à sua representação de contexto própria. O contexto referente à Infra-Estrutura é composto pelo conjunto da representação das entidades Rede, Dispositivos e Serviços. O contexto do Sistema abrange o conjunto dos contextos do Usuário, Aplicação e Infra-Estrutura. Ambos os contextos de Infra-Estrutura e Sistema são representados por retângulos pontilhados por serem formados por entidades com representações contextuais próprias (Goularte, 2003).

O sistema computacional apresentado na Figura 2.3 é composto por usuários, aplicações e por uma infra-estrutura. Os usuários fazem uso de dispositivos (computadores, *handhelds*, etc.) para acessar aplicações e requisitar alguma tarefa. As aplicações, por sua vez, utilizam os dispositivos aliados aos serviços (compartilhamento de arquivos, servidores de impressão, etc.) proporcionados pela infra-estrutura para efetuar a tarefa requisitada. A rede é responsável pela comunicação entre usuários, dispositivos, aplicações e serviços.

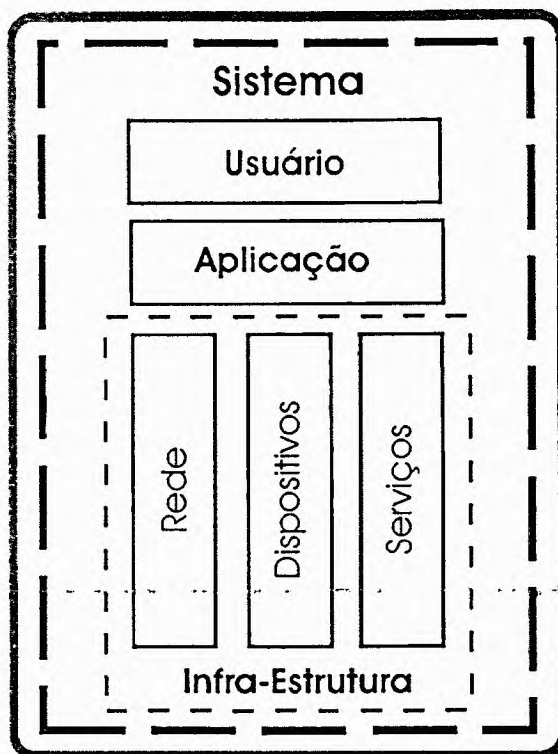


Figura 2.3: Componentes que compõem o contexto do Sistema. Adaptado de (Goularte, 2003).

A abordagem adotada considera o contexto da Infra-Estrutura como sendo composto por representações do estado da rede, dos dispositivos e dos serviços. Cada um desses componentes é uma classe de entidades contextuais, cada qual caracterizada por informações de contexto. Tal abordagem distingue-se das abordagens de Schilit (1995) e de Dix et al. (2000) porque, para esses autores, o contexto da Infra-Estrutura é formado apenas por informações referentes ao estado da rede, comunicando apenas usuários e aplicações. Portanto, os autores não consideram a influência do dinamismo do estado dos serviços e dispositivos utilizados pela aplicação (Goularte, 2003). Por exemplo, o acréscimo de dispositivos conectados à rede influencia diretamente na competição por largura de banda e por serviços.

Considerando-se ainda a discussão efetuada na Seção 2.4 sobre organização de contexto do Sistema, para autores como Dix et al. (2000), Crowley et al. (2002) e Chalmers (2002), tal contexto é formado pela representação do contexto do Usuário e as representações dos estados dos dispositivos e serviços. No entanto, os estados de ambos são considerados atributos do contexto do Sistema. Para Goularte (2003), porém, o contexto do Sistema é composto por representações do Usuário, da Aplicação e da Infra-Estrutura (Figura 2.3), além de classificar serviços e dispositivos como parte do contexto da Infra-Estrutura. Essa organização distinta, contudo, não tem seu significado semântico prejudicado, pois as características dos serviços e dispositivos

ainda permanecem contidas no contexto do Sistema. Outro aspecto relevante da abordagem de Goularte (2003) é que o contexto da Aplicação é classificado como parte do contexto do Sistema. Essa característica não é considerada por outros autores (Dix et al., 2000; Crowley et al., 2002; Chalmers, 2002), embora os mesmos relatem a relevância das informações referentes às aplicações para o contexto do Sistema.

Finalmente, a organização estruturada dos tipos de contexto e a separação coerente entre os contextos do Sistema e da Infra-estrutura presentes no trabalho de Goularte (2003) são características inexploradas nos trabalhos mencionados na Seção 2.4. Tais aspectos auxiliam a compreensão dos diferentes tipos de contexto e, portanto, contribuem para a criação de representações para informações contextuais.

Para solucionar as limitações de estruturação das informações de contexto e da flexibilidade em compartilhá-las entre aplicações distintas, Goularte (2003) apresenta como alternativa a construção de esquemas de descrição por meio de padrões para representação como XML Schema (Fallside, 2001) ou RDF (*Resource Description Framework*) (Lassila & Swick, 1999). Tais esquemas devem ser utilizados para estruturação e organização da modelagem de informações de contexto. Por haver variação das informações contextuais de aplicação para aplicação, torna-se complexo, ou talvez impossível, abranger todas as possibilidades de uso de contexto. Porém, é possível criar um conjunto de esquemas de representação de informações de contexto comuns a um determinado grupo de aplicações, além de prover a possibilidade de criação de representações próprias por parte dos projetistas, caso haja necessidade (Goularte, 2003).

Com esses argumentos, Goularte (2003) apresenta em seu trabalho uma biblioteca extensível de elementos contextuais estruturados. Baseada em um espaço de nomes XML, a biblioteca é composta por esquemas desenvolvidos em XML Schema, padrão de representação escolhido para o modelo devido à integração com o padrão MPEG-7. Esse último padrão utiliza o XML Schema de uma forma estendida como linguagem de definição de dados. Os elementos pertencentes à biblioteca extensível de elementos contextuais estruturados provêm algumas funcionalidades como o **Elemento Raiz** que compõe o topo da hierarquia na representação contextual, os **Tipos de Contexto** e o **Contexto Primário**, o qual é responsável pela caracterização das entidades contextuais.

Elemento Raiz

O Elemento Raiz Context é o elemento que encapsula toda a hierarquia de representação de contexto. Partindo-se desse elemento, dois caminhos podem

ser percorridos: o primeiro descreve o contexto do Sistema (elemento <System>), e o segundo descreve qualquer combinação possível entre os elementos dos contextos do Usuário (elemento <Users>), Aplicação (elemento <Applications>) e da Infra-Estrutura (elemento <Infrastructure>). A divisão relativa ao modelo de organização é necessária porque, baseado na discussão sobre classificação de contexto efetuada na Seção 2.4 e na argumentação de Goularte (2003), há aplicações que consideram o contexto do Sistema e ainda, aplicações em que apenas o contexto do Usuário é importante, por exemplo, ou ainda alguma combinação entre contextos de Infra-Estrutura, Usuário e Aplicação.

A Figura 2.4 ilustra o Diagrama de classes UML (*Unified Modeling Language*) para o tipo abstrato `ContextType`, o qual é estendido pelo elemento raiz. `ContextType` é uma generalização dos tipos de contexto que compõem o conjunto de esquemas Tipos de Contexto: `SystemType`, `UsersType`, `ApplicationsType` e `InfrastructureType`.

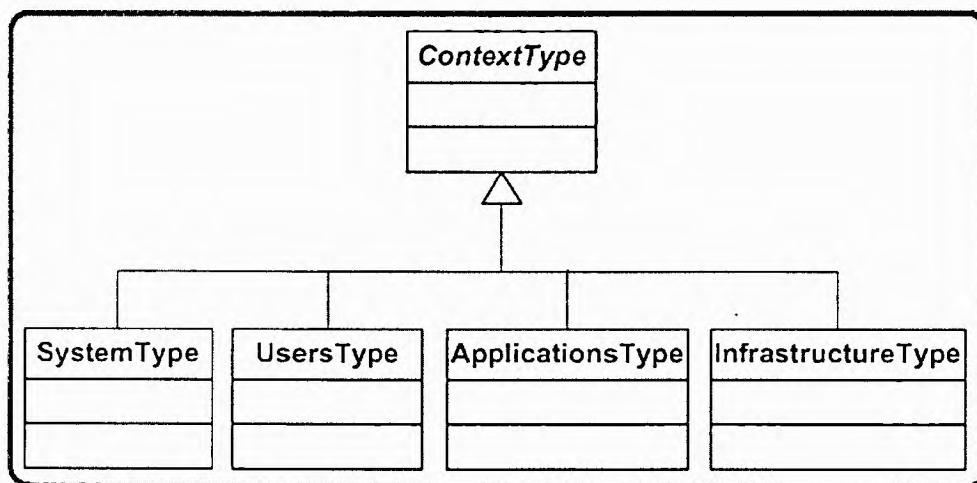


Figura 2.4: Diagrama de classes UML para o tipo `ContextType`. Adaptado de (Goularte, 2003).

Conjunto de Esquemas Tipos de Contexto

Respeitando-se a organização descrita por Goularte (2003) e mencionada na Seção 2.7, o conjunto de esquemas Tipos de Contexto representa os tipos de contexto do Sistema, Usuário, Aplicação e Infra-Estrutura. Vale lembrar que tanto o contexto do Sistema quanto o de Infra-Estrutura são formados por conjuntos de entidades contextuais com características comuns. Desse modo, o contexto de Infra-Estrutura é apresentado a partir do elemento <Infrastructure> que encapsula o conjunto de elementos que compõe o contexto de Infra-Estrutura. O elemento <Infrastructure> é constituído pelos elementos <Network>, <Service> e <Device>.

A hierarquia de representação de tipos é apresentada na Figura 2.5. Na figura é possível notar a representação dos componentes (NetworkType, ServicesType e DevicesType) criados no intuito de alcançar um modo de representação de uma entidade contextual específica, a qual pode ser classificada como pertencente a um dos componentes, além de permitir a inserção de novas entidades criadas pelos desenvolvedores. Justificando-se tais argumentos, em sua abordagem, Goularte (2003) utiliza como artifício em sua abordagem o uso de componentes intermediários relacionados a cada componente. Desse modo, por exemplo, o componente DevicesType representa a classe enquanto que seu respectivo componente intermediário representa uma determinada instância dessa classe (Figura 2.5).

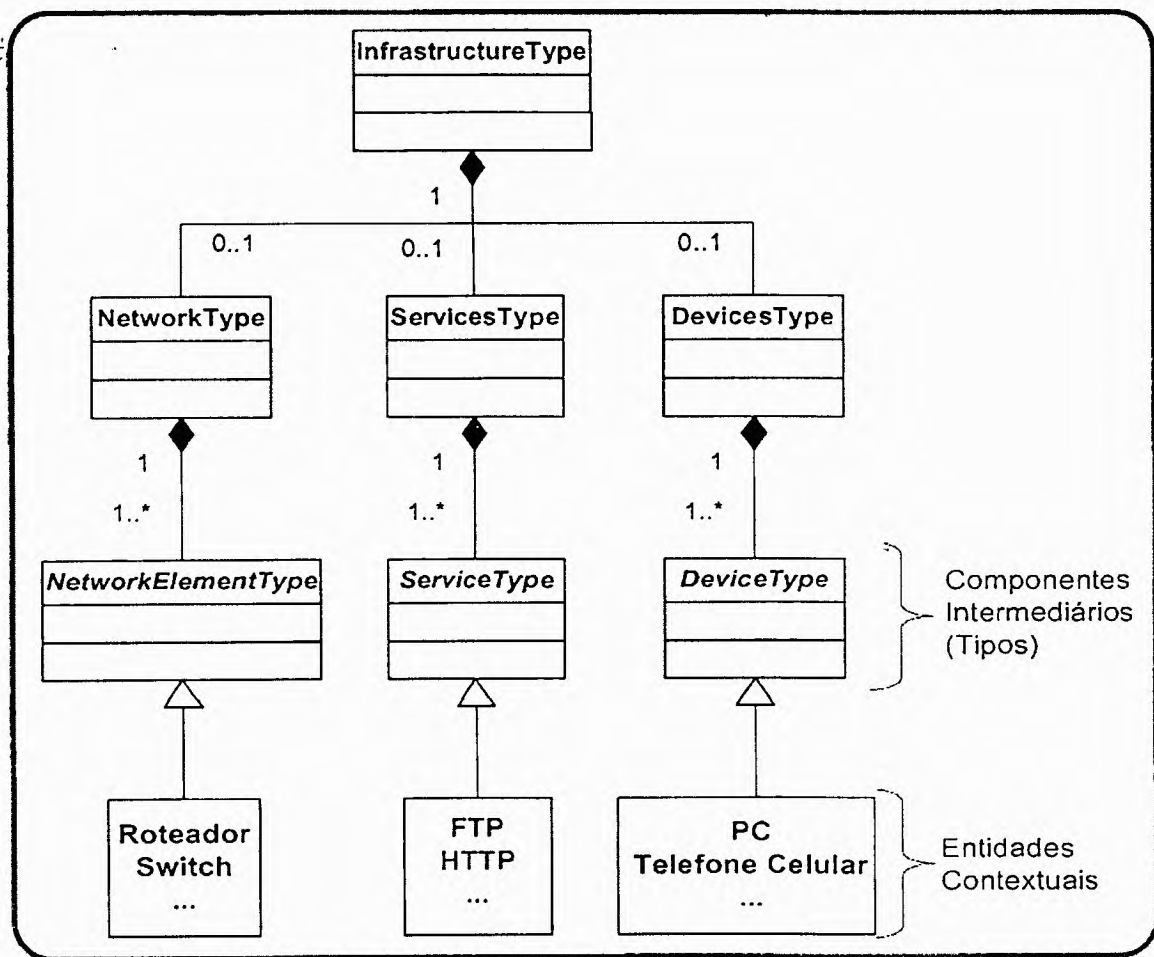


Figura 2.5: Diagrama de classes UML do contexto da Infra-Estrutura (InfrastructureType). Adaptado de (Goularte, 2003).

Para que uma entidade contextual seja representada nessa hierarquia, faz-se necessária a criação de um tipo que estenda o tipo abstrato apropriado. Por exemplo, a Figura 2.6 (a) mostra o tipo LaptopType, derivado por extensão do tipo abstrato DeviceType (linha 4), criado para representar um dispositivo qualquer, no caso, um *laptop*. Para efetuar a instância desse novo tipo faz-se

```

1 <schema targetNamespace="http://www.icmc.usp.br/~rudinei/Context/"
...
2 <complexType name="LapTopType">
3   <complexContent>
4     <extension base="ctx:DeviceType" />
5   </complexContent>
6 </complexType>
...

```

(a)

```

...
1 <Device xsi:type="LapTopType">
2   <Identity xsi:type="DeviceIDType">
3     <DeviceID>dev00345</DeviceID>
4   </Identity>
5   <Location xsi:type="InDoorLocationType">
6     <Floor>1</Floor>
7     <Room>Kitchen</Room>
8   </Location>
9   <Time xsi:type="dateType">
10    <Date>2003-11-20</Date>
11  </Time>
12 </Device>
...

```

(b)

Figura 2.6: (a) O tipo `LapTopType` criado para representar uma entidade dispositivo (*laptop*). (b) Instância de um dispositivo do tipo `LapTopType`.

necessário o uso do elemento `<Device>`. No entanto, por ser um tipo abstrato, `<Device>` não possui um tipo definido, o que leva à necessidade de se indicar qual tipo é implementado por esse elemento. Para isso, utiliza-se o atributo XML `xsi:type`. A instância `LapTopType` é ilustrada no item (b) da Figura 2.6 (Goularte, 2003).

Formados por entidades com representações contextuais próprias, os contextos da Aplicação e do Usuário são representados diretamente. Do mesmo modo, o tipo `ApplicationType` representa o componente intermediário do contexto da aplicação, enquanto o tipo `UserType` representa o componente intermediário do contexto referente ao usuário. Ao contrário da representação contextual da Infra-Estrutura, ambos os tipos `ApplicationType` e `UserType` não são abstratos. Tais tipos representam instâncias de entidades contextuais de forma direta, como apresentado na Figura 2.7.

Já o contexto do Sistema é formado pela representação contextual do Usuário, da Aplicação e da Infra-Estrutura. Portanto, o tipo de representação do contexto do Sistema é composto pelos tipos `InfrastructureType`, `UsersType` e `ApplicationsType`, como mostra a Figura 2.8.

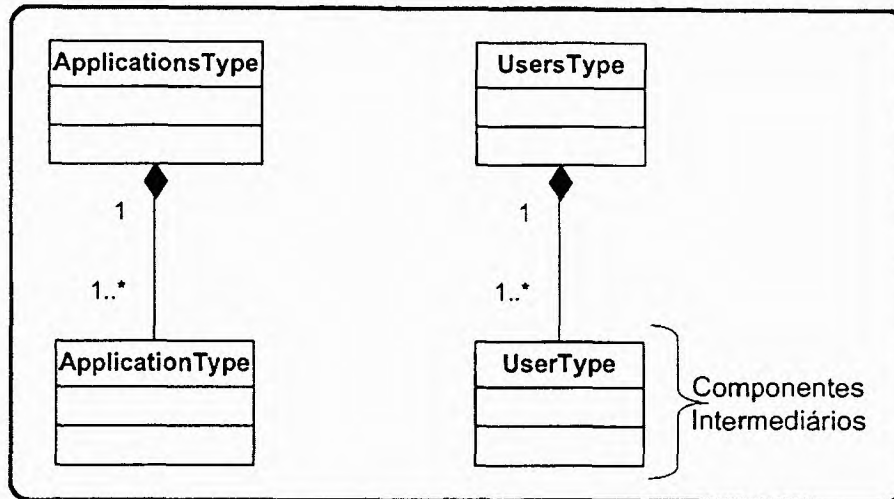


Figura 2.7: Diagrama de classes UML para os contextos da Aplicação (ApplicationsType) e do Usuário (UsersType). Adaptado de (Goularte, 2003).

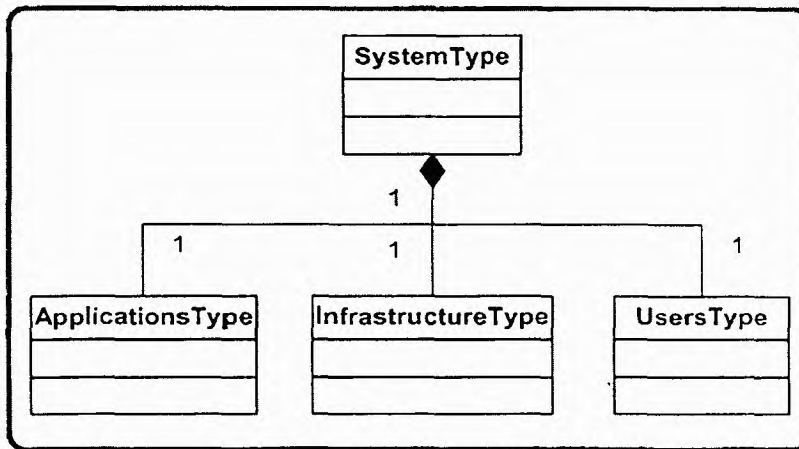


Figura 2.8: Diagrama de classes UML referente ao contexto do Sistema (SystemType). Adaptado de (Goularte, 2003).

Contexto Primário

Partindo da definição de contexto de Dey (2001) abordada na Seção 2.1, a qual especifica que a situação de uma dada entidade é caracterizada por um conjunto de informações de contexto, Goularte (2003) menciona em seu trabalho o uso do contexto primário como solução para indexação de informações contextuais. O contexto primário, mencionado na Seção 2.1, é constituído por tipos de informações contextuais que, na prática, são mais importantes que outros. Compõem esses tipos: informações de localização, identidade, atividade e tempo (Schilit et al., 1994; Brown et al., 1997; Pascoe, 1998; Dey & Abowd, 2000; Dey, 2001).

Goularte (2003) se beneficia do contexto primário para representar informações contextuais por meio do tipo abstrato PrimaryContextType, mostrado na Figura 2.9, constituído pelos tipos abstratos: IdentityType, LocationType,

TimeType e ActivityType. Esses tipos abstratos representam seus correspondentes tipos primários, respectivamente: identidade, localização, tempo e atividade.

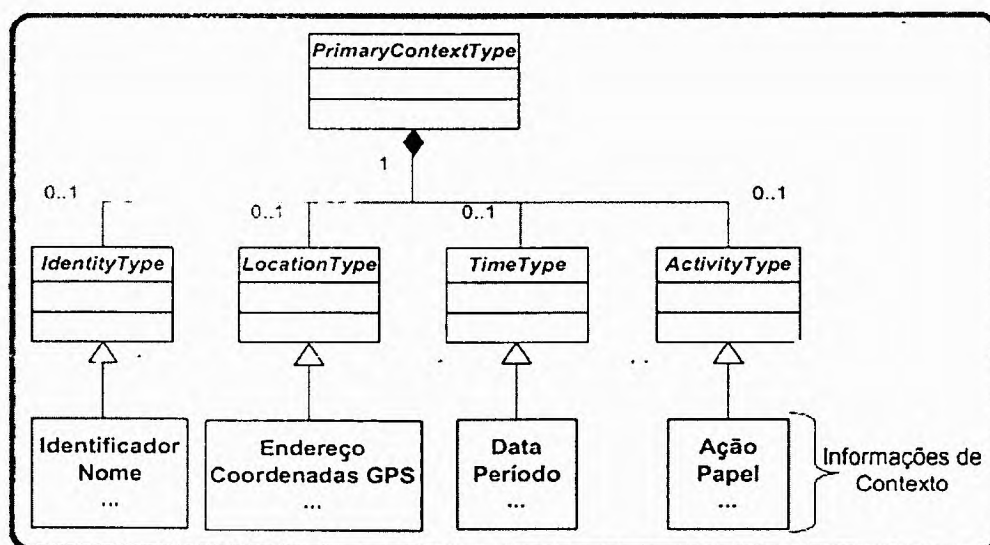


Figura 2.9: Diagrama de classes UML representando o contexto Primário (PrimaryContextType). Adaptado de (Goularte, 2003).

Informações contextuais são representadas por meio de tipos derivados dos componentes de PrimaryContextType. Embora a biblioteca apresentada por Goularte (2003) forneça um conjunto pré-definido de informações de contexto, os desenvolvedores podem criar informações de contexto específicas. Tal característica é permitida porque os componentes de PrimaryContextType são abstratos.

Os componentes intermediários apresentados na Seção 2.7 são derivados por extensão do tipo PrimaryContextType. Como consequência, toda entidade contextual herda automaticamente o contexto primário possibilitando assim o uso das informações contextuais para a caracterização da situação da entidade.

Na Figura 2.6 (b) é apresentado um exemplo de instância válida para o tipo LaptopType. Por ser um tipo derivado de DeviceType, componente intermediário do componente DeviceType, LaptopType, herda todos os elementos que compõem o contexto primário (<Identity>, <Location>, <Time>, <Activity>). É importante ressaltar que a ausência do elemento <Activity> ocorre na instância apresentada porque os elementos do contexto primário são de caráter opcional (Goularte, 2003).

2.8 Considerações Finais

Neste capítulo foi apresentada uma revisão na literatura da área de computação ubíqua. O objetivo de estudo da área é a melhoria das formas de interação usuário-computador baseada no incremento de recursos computacionais capazes de prover serviços e informações quando e onde forem desejadas de modo não intrusivo. Foram abordados conceitos e apresentadas as três vertentes de pesquisas da área: interfaces naturais, captura e acesso de atividades humanas e computação ciente de contexto (a área na qual o presente trabalho está inserido).

A área de computação ciente de contexto é responsável pelo enriquecimento da interação usuário-computador. O objetivo é alcançar um aproveitamento eficaz da interação utilizando-se de informações contextuais presentes na comunicação. Para auxiliar os desenvolvedores de aplicações cientes de contexto a eleger quais informações são relevantes para a interação, a literatura apresenta dimensões contextuais conhecidas como os cinco Ws semânticos apresentados na Seção 2.1.

Foram apresentadas também as classificações de contexto descritas por diversos autores na Seção 2.4 e a categorização de características para aplicações cientes de contexto na Seção 2.5. Para esses autores o contexto pode ser classificado em contexto de Infra-Estrutura, de Sistema, de Domínio e de Ambiente. Além das definições, dimensões, classificações de contexto e categorias de aplicações cientes de contexto, este capítulo apresentou também os requisitos que podem ser seguidos para o desenvolvimento de aplicações cientes de contexto, descritos por Dey (2000). Segundo o autor, tais requisitos são derivados principalmente de aspectos comuns em aplicações cientes de contexto.

No capítulo foram descritos alguns trabalhos importantes nessa área e muito relevantes para este trabalho. As aplicações *Active Badge* (Want et al., 1992) e *Cyberguide* (Long et al., 1996; Abowd et al., 1997) são trabalhos pioneiros na área de computação ciente de contexto. O *Context Toolkit* (Salber et al., 1999; Dey et al., 1999; Dey, 2000), *Context Fabric* (Hong & Landay, 2001) e *Context Kernel* (Arruda Jr, 2003) são exemplos de infra-estruturas que auxiliam o suporte a aplicações cientes de contexto.

O *Context Toolkit* provê suporte à aquisição de informações contextuais provenientes de sensores distribuídos em um ambiente. Seus componentes denominados *widgets* permitem que aplicações sejam capazes de acessar informações de contexto independente do processo de captura das mesmas. Graças às abstrações implementadas pelo *Context Toolkit* foi possível fornecer padronização e generalização às aplicações, aspecto que resultou no reuso

de métodos de desenvolvimento de novas aplicações cientes de contexto.

O *Context Fabric* propõe a linguagem CSL descrita em formato XML para especificação de notificação de eventos e consultas a informações de contexto. Além dessa linguagem, o *Context Fabric* fornece serviços especializados em notificação de eventos e consultas de informações contextuais.

O *Context Kernel* provê serviços capazes de manipular informações de contexto fornecendo registro de aplicações, armazenamento, recuperação e consulta dessas informações. Implementada com tecnologia *Web Service*, essa infra-estrutura modela informações de contexto com base nas dimensões contextuais descritas na literatura. Por se beneficiar do uso de *Web Service*, o *Context Kernel* garante a interoperabilidade entre as aplicações, e também a independência das aplicações na comunicação. No entanto, não efetua captura de informações contextuais, podendo ser classificado, portanto, como um serviço de armazenamento e consulta de informações de contexto. Com relação à representação contextual usada nesse trabalho, considerando-se o modo como foi implementado, o *Context Kernel* trata as informações contextuais isoladamente, ou seja, não há relação entre as informações de contexto.

Embora todos os trabalhos mencionados forneçam suporte ao desenvolvimento de aplicações cientes de contexto, são notórias suas diferenças. O *Context Toolkit* privilegia a interação entre sensores e componentes ou aplicações por meio de abstrações, enquanto que o *Context Fabric* e o *Context Kernel* são focados em uma modelagem de armazenamento de informações de contexto. Além disso, O *Context Fabric* e o *Context Toolkit* provêm notificação de eventos, característica inexistente no *Context Kernel*. No entanto, todos eles são capazes de efetuar consultas a informações de contexto.

O *Context Kernel* por sua vez faz uso do formato XML para comunicação de informações de contexto, assim como o *Context Fabric* e o *Context Toolkit*. Porém, o *Context Kernel* tem a vantagem de utilizar a tecnologia *Web Service* para a troca de mensagens XML. A tecnologia *Web Service* é composta por um conjunto de padrões desenvolvidos pelo Consórcio W3C, característica que contribui para sua difusão. Essa característica não é explorada pelo *Context Toolkit* que, embora utilize XML sobre HTTP especificando o modelo de troca de mensagens, não segue nenhum padrão W3C. Por sua vez, o *Context Fabric* apresenta sua própria linguagem de especificação de informações de contexto utilizada para notificação de eventos e consultas, a linguagem CSL. Por não se tratar de um padrão, essa linguagem torna-se restrita ao *Context Fabric*, um ponto negativo no uso da mesma, pois desenvolvedores deverão se ater ao seu aprendizado.

Apesar das diferenças entre esses trabalhos serem evidentes, todos eles têm uma lacuna em comum: a carência de uma representação de informa-

ções de contexto extensível. A importância de representações contextuais com essas características foi abordada na Seção 2.7. Nessa seção também foi apresentada a alternativa proposta por Goularte (2003) para suprir esse problema, com a modelagem de uma representação de contexto baseada em uma classificação e organização de contexto coerentes. A representação apresentada por Goularte (2003) é base para o desenvolvimento do trabalho apresentado nesta dissertação. No próximo capítulo tal desenvolvimento é descrito em detalhes.

O Serviço de Gerenciamento de Contexto

No capítulo anterior foram apresentados os esforços de pesquisadores no sentido de desenvolver infra-estruturas capazes de gerenciar informações de contexto. Exemplos que evidenciam a convergência para o gerenciamento de informações contextuais são os trabalhos de Salber et al. (1999) com o *Context Toolkit*, o projeto *Context Fabric*, de Hong & Landay (2001) e o *Context Kernel* de Arruda Jr (2003), mencionados na Seção 2.6. Ainda, foi enfatizado também que a infra-estrutura deve possuir um modelo de representação contextual uniforme e extensível para que os desenvolvedores possam, caso seja necessário, criar tipos de informações de contexto e entidades contextuais próprias. Assim, com um modelo de representação de contexto extensível, a flexibilidade dessa representação é garantida (Abowd et al., 2002). Essa característica contribui para o compartilhamento das informações contextuais e para a interoperabilidade das aplicações cientes de contexto beneficiadas pela infra-estrutura. Como discutida na Seção 2.7, a representação de informações de contexto tem sido tema de várias pesquisas, pois representar informações contextuais é tão complexo quanto apontar a relevância desse tipo de informação.

Por outro lado, o tipo de contexto mais explorado na literatura é o contexto do Usuário (Abowd & Mynatt, 2000; Dey & Abowd, 2000; Dey, 2001). Além desse tipo de contexto, alguns sistemas necessitam de informações contextuais de Infra-Estrutura, pois informações desse tipo fornecem subsídios para que a disponibilidade de recursos provenientes do sistema seja conhecida (Schilit, 1995; Dix et al., 2000). Como exemplo de sistema que pode

se beneficiar de informações contextuais de Infra-Estrutura pode ser citado o projeto no qual este trabalho está contextualizado: o protótipo de TV Interativa (Goularte, 2003). A utilização de informações de contexto de Infra-Estrutura reforça a necessidade de um gerenciamento sólido dessas informações e de uma representação contextual uniforme, consistente e extensível.

Portanto, para suprir a carência de abordagens de informações de contexto referentes à Infra-Estrutura e de representação dessas informações, o presente trabalho apresenta neste capítulo soluções para a representação de contexto de Infra-Estrutura e um serviço de gerenciamento de informações contextuais que se beneficia dessa representação de contexto. Vale ressaltar que a representação apresentada aqui é derivada do modelo de representação contextual de Goularte (2003). Esse autor convergiu sua pesquisa para a exploração do contexto do Usuário; não abordando o contexto de Infra-Estrutura referente aos dispositivos e o contexto relacionado às características da rede. O presente trabalho estendeu a representação de contexto dos dispositivos de Goularte (2003) no intuito de preencher a lacuna relacionada à descrição das características específicas dos dispositivos, além de modelar, especificar e elaborar uma entidade de rede como contribuição para o avanço nas pesquisas focadas na representação de contexto referente à Infra-Estrutura.

Este capítulo está dividido em oito seções. A Seção 3.1 aborda a representação de contexto utilizada pelo serviço de gerenciamento de contexto, apresentando as representações contextuais dos dispositivos e de rede, respectivamente. A Seção 3.2 ilustra o serviço de gerenciamento de contexto. As Seções 3.3 e 3.4 descrevem as partes que compõem o serviço de gerenciamento: o Coletor Contextual e o *Web Service* Gerente de Contexto, respectivamente. Na Seção 3.5 é descrita a modelagem do serviço de gerenciamento de contexto, enquanto que a arquitetura do serviço é apresentada na Seção 3.6. Na Seção 3.7 é apresentado o fluxograma desse serviço. Finalmente, a Seção 3.9 apresenta algumas considerações finais sobre o capítulo.

3.1 Representação de Contexto no Serviço de Gerenciamento de Contexto

Na Seção 2.7 foi apresentada a abordagem para classificação e organização de contexto, além da representação de contexto para o projeto do protótipo de TV Interativa proposta por Goularte (2003) na Subseção ???. Como vantagens, podem ser mencionadas: uma classificação dos tipos de contexto; a construção de estruturas de alto nível que organizam a descrição contextual; uma biblioteca extensível, que possibilita a criação estruturada de informações contextuais além de permitir que desenvolvedores criem suas próprias

representações para entidades contextuais e para informações de contexto.

Ainda, o trabalho de Goularte (2003) utiliza informações de contexto com o objetivo de potencializar o desenvolvimento de serviços avançados, como geração e adaptação automática e personalizada de conteúdo. Para tanto, Goularte (2003) se beneficia de informações provenientes do contexto do Usuário, pois há a necessidade de exploração das características das preferências do usuário para o provimento dos serviços mencionados. No entanto, considerando-se uma aplicação como o protótipo de TV Interativa, durante este trabalho foi evidenciada a necessidade não apenas do uso de informações provenientes do contexto do Usuário, mas também do contexto da Infra-Estrutura, mais especificamente os contextos referentes aos dispositivos e à rede. O objetivo é prover ao sistema informações sobre os dispositivos heterogêneos que por ventura possam ser utilizados pelos usuários e, ainda, informações sobre o estado atual da rede.

Para garantir tal característica ao protótipo de TV Interativa, o uso de contexto referente à rede e aos dispositivos pode ser considerado fator determinante. Desse modo, conhecendo o dispositivo empregado pelo usuário para acesso ao sistema, é possível gerar conteúdo orientado ao dispositivo, além de personalizado para o usuário. Essa propriedade concentra-se na relevância em conhecer as características específicas de cada dispositivo que podem ser exploradas, bem como suas limitações. O contexto da rede, por sua vez, é importante uma vez que usuários competirão por largura de banda, por exemplo, aspecto que pode comprometer a entrega de conteúdo compatível com a qualidade suportada pelo dispositivo, ou colaborar para entrega de conteúdo com melhor qualidade.

Para viabilizar a representação de informações contextuais, Goularte (2003) faz uso dos tipos de contexto primário. Como mencionado na Seção 2.1, a literatura (Schilit et al., 1994; Brown et al., 1997; Pascoe, 1998; Dey & Abowd, 2000; Dey, 2001) aponta esses tipos como mais relevantes, na prática. Embora Goularte (2003) utilize as dimensões contextuais *Who*, *Where*, *When* e *What*, representadas pelo contexto primário, o uso de uma dimensão extra faz-se necessário no presente trabalho: a dimensão *How*.

A utilização dessa dimensão é necessária devido à importância de se conhecer as características específicas e as limitações dos dispositivos utilizados na interação entre usuário e aplicação, bem como o estado momentâneo da rede, responsável pela comunicação. É importante ressaltar que houve a tentativa de mapeamento dessas informações para os tipos de contexto primário tradicionais. Contudo, tal mapeamento não foi viável em virtude das informações que provêm o estado da rede e aspectos específicos dos dispositivos não serem suportadas por nenhuma das categorias de contexto primário es-

pecificadas por Goularte (2003). Por exemplo, a resolução de um *handheld* não poderia ser mapeada como informação de identificação (tipo de contexto primário identidade), muito menos de atividade e tampouco de localização ou tempo, mas sim como informação referente a uma característica que informa quais dimensões devem ser adotadas para a apresentação de conteúdo a esse dispositivo.

Portanto, por meio da dimensão contextual *How* foi possível inserir na hierarquia de representação de contexto de Goularte (2003) o tipo de contexto primário **modo** (*How*) que supriu tal carência. Com a inserção desse tipo de contexto primário na hierarquia houve um mapeamento das informações relacionadas às características dos dispositivos e da rede para esse tipo de contexto primário. A Figura 3.1 ilustra a hierarquia de representação de contexto destacando a inserção do tipo de contexto primário *How*.

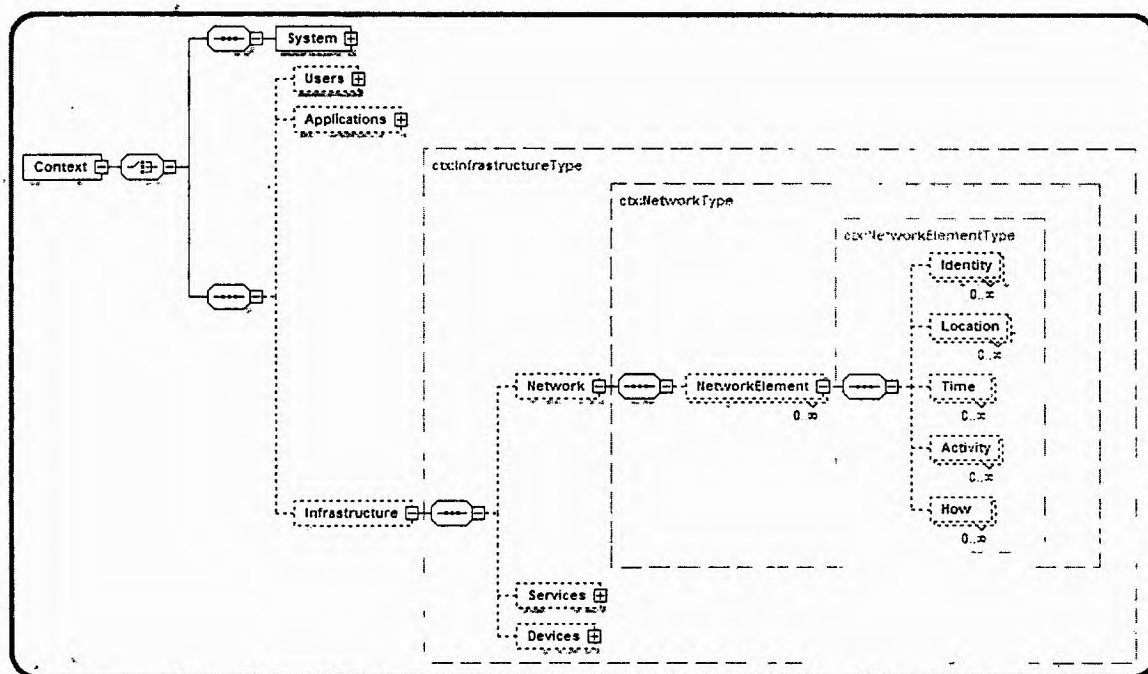


Figura 3.1: Diagrama hierárquico dos elementos de contexto. Destaque para o elemento *How*.

A Figura 3.2 apresenta o diagrama de classes em UML, representando o contexto primário, formado pelos tipos abstratos *IdentityType*, *LocationType*, *TimeType*, *ActivityType* que representam os respectivos tipos de contexto, identidade, localização, tempo e atividade, utilizados por Goularte (2003) (Figura 2.9). O tipo de contexto primário que expressa o **modo** (*How*), é representado pelo tipo abstrato *HowType*, também ilustrado na Figura 3.2. Portanto, com a inserção do tipo *HowType* no conjunto de tipos abstratos que compõe o contexto primário, os componentes intermediários mencionados na Seção ??, que derivam por extensão o tipo *PrimaryContextType*, serão capazes também

de representar informações contextuais de modo (*How*), além das informações já apresentadas por Goularte (2003) (identidade, atividade, localização e tempo).

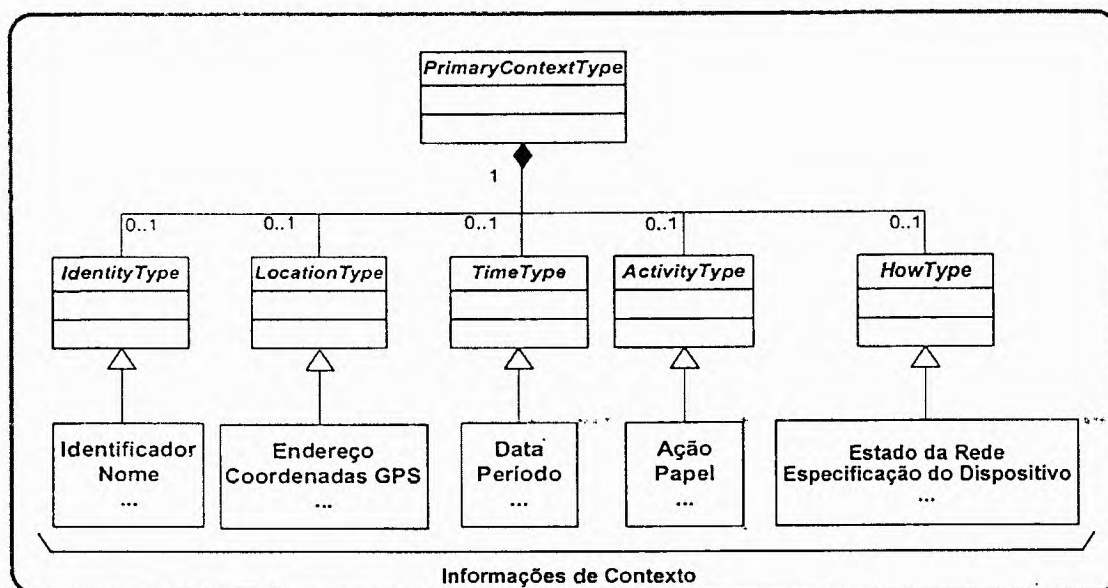


Figura 3.2: Diagrama de classes UML do contexto Primário (PrimaryContextType) após a inserção do tipo HowType.

Com base nas abstrações fornecidas por Goularte (2003) e a extensão do tipo contexto primário (PrimaryContextType) apresentada neste trabalho, foram modeladas representações de informações de contexto para dispositivos e rede.

3.1.1 Representação de Contexto de Dispositivos

Como mencionado, é de alta relevância o conhecimento de informações relacionadas aos aspectos específicos dos dispositivos. Munidos dessas informações, desenvolvedores estão aptos a tratar o conteúdo provido aos usuários de modo orientado ao dispositivo utilizado para o acesso ao conteúdo, tanto do ponto de vista de apresentação quanto de geração desse conteúdo. No entanto, há uma série de informações úteis que caracterizam os dispositivos. Efetuar a descrição e classificação de todas, ou de uma grande parte dessas informações seria um trabalho árduo. Além disso, há a necessidade de especificar as características dos dispositivos de modo uniforme e padronizado. Para preencher essa carência, foi analisado e adotado o padrão CC/PP (*Composite Capabilities/Preference Profiles*) (Klyne et al., 2004) proposto pelo W3C, padrão esse que atendeu às expectativas necessárias para representação das características dos dispositivos.

Com esse padrão é possível especificar um perfil referente a um determi-

nado dispositivo. Um perfil CC/PP é uma descrição das capacidades de um dispositivo ou das preferências de um usuário que podem ser usadas para adaptar a apresentação de conteúdo de acordo com o dispositivo utilizado e com as preferências do usuário (Klyne et al., 2004). Vale destacar que no contexto do presente trabalho, o padrão CC/PP é usado apenas para a descrição dos dispositivos, sendo de responsabilidade da representação de contexto do Usuário a manipulação das informações contextuais referentes às preferências do mesmo.

Um perfil CC/PP contém um determinado número de componentes (Hardware, Sistema Operacional ou Aplicação) e cada componente possui pelo menos uma ou mais propriedades, por exemplo, versão ou nome. O conjunto de propriedades de componentes e seus valores constituem um vocabulário utilizado para validação do perfil CC/PP (Klyne et al., 2004): Um exemplo de perfil é mostrado a seguir na Figura 3.3. As linhas 7 e 8 apresentam as propriedades referentes à resolução (*displayWidth* e *displayHeight*) suportada pelo dispositivo para a apresentação de conteúdo, por exemplo.

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"
  xmlns:ex="http://www.example.com/schema#" >
3   <rdf:Description
4     rdf:about="http://www.example.com/profile#MyProfile">
5     <ccpp:component>
6       <rdf:Description
7         rdf:about="http://www.example.com/profile#TerminalHardware">
8         <rdf:type
9           rdf:resource="http://www.example.com/schema#HardwarePlatform" />
10        <ex:displayWidth>320</ex:displayWidth>
11        <ex:displayHeight>200</ex:displayHeight>
12       </rdf:Description>
13      </ccpp:component>
14     <ccpp:component>
15       <rdf:Description
16         rdf:about="http://www.example.com/profile#TerminalSoftware">
17         <rdf:type
18           rdf:resource="http://www.example.com/schema#SoftwarePlatform" />
19         <ex:name>EPOC</ex:name>
20         <ex:version>2.0</ex:version>
21         <ex:vendor>Symbian</ex:vendor>
22       </rdf:Description>
23      </ccpp:component>
24    </rdf:Description>
25  </rdf:RDF>
```

Figura 3.3: Perfil CC/PP. Adaptado de (Klyne et al., 2004).

Embora o padrão CC/PP seja completo, preenchendo todos os requisitos necessários para a especificação das características de dispositivos, sua integração ao modelo de representação de contexto seria problemática, pois o CC/PP é baseado em RDF, linguagem utilizada para descrição de recursos, seus valores e relações, ou seja, descrição do relacionamento semântico entre recursos e suas propriedades (Lassila & Swick, 1999).

Essa característica do padrão CC/PP foi considerada problemática para a integração, porque todo o modelo de representação de contexto utilizado no protótipo de TV Interativa é baseado na linguagem XML Schema (Fallside, 2001). Essa linguagem é responsável pela especificação das representações de contexto, além da validação semântica e sintática das mesmas. Por ser baseado no padrão RDF, conseqüentemente, o padrão CC/PP utiliza para a validação de descrições de recursos o padrão RDF Schema. Um documento RDF Schema é responsável por fornecer informações sobre a interpretação das declarações de um modelo de dados RDF (Fallside, 2001). Além disso, o RDF Schema também atua em nível semântico, não efetuando a validação sintática da descrição RDF.

A carência de validação sintática do padrão CC/PP tem sido mencionada na literatura. Em seu trabalho Smith & Butler (2002) abordam alguns problemas encontrados devido à ausência desse tipo de validação, por exemplo, perfis que não descrevem recursos RDF válidos, ou ainda documentos mal formados (com erros de sintaxe). Baseado em tais exemplos, o autor relata a necessidade de uma validação sintática para perfis CC/PP, de modo a garantir a coerência desses perfis. Como solução para essa lacuna, Smith & Butler (2002) apresentam o mapeamento do padrão CC/PP para a linguagem de descrição XML Schema, com o objetivo de abolir essa limitação.

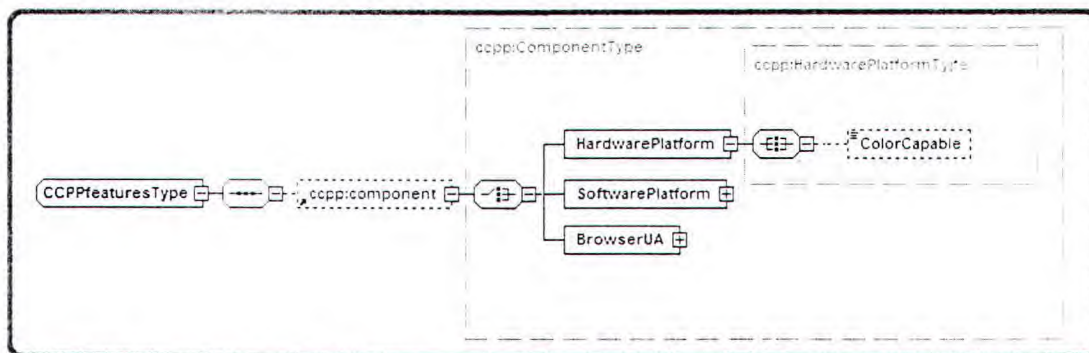


Figura 3.4: Diagrama hierárquico do tipo CCPPfeaturesType que encapsula os tipos declarados no padrão CC/PP.

O mapeamento realizado por Smith & Butler (2002) é usado neste trabalho como solução adequada para integração do padrão CC/PP à modelagem de representação contextual. Para realizar a integração, foi criado o tipo de informação contextual chamado CCPPfeaturesType (Figura 3.4). Esse elemento foi adicionado aos tipos de informações contextuais referentes à representação de dispositivos, no intuito de encapsular todas as informações que compõem o padrão CC/PP.

A Figura 3.5 (a) ilustra a declaração do tipo CCPPfeaturesType na biblioteca de contexto criada por Goularte (2003). Por tratar de características referentes

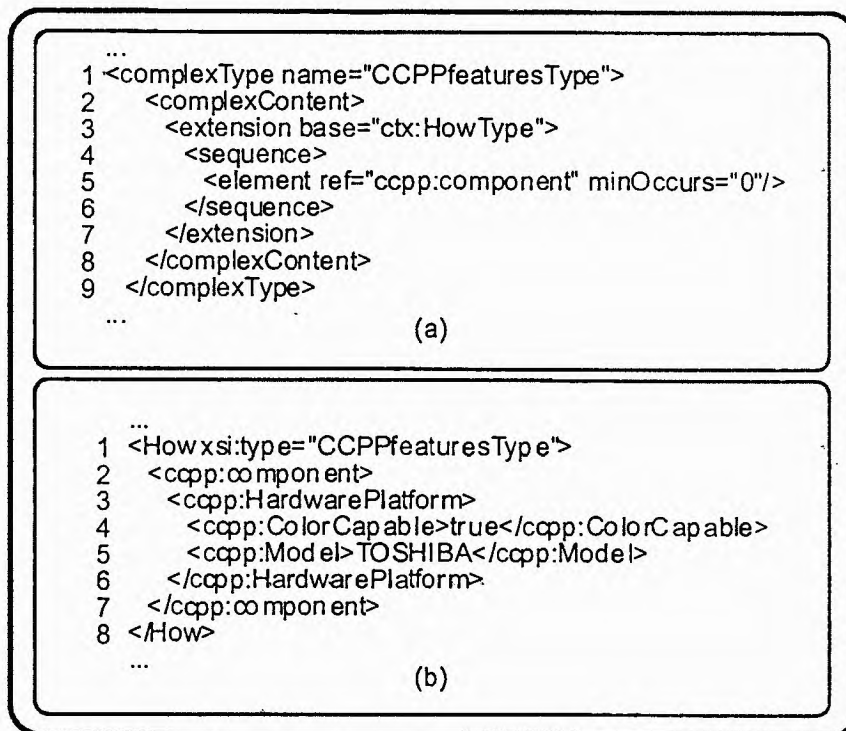


Figura 3.5: (a) Tipo CCppFeaturesType. (b) Instância de um dispositivo com característica que suporta o uso de cores (ColorCapable).

a aspectos do dispositivo, o tipo CCppFeaturesType estende o tipo HowType (linha 3). O tipo CCppFeaturesType é composto por uma seqüência de elementos chamados <component> (linha 5), os quais são do tipo ComponentType, herdado do mapeamento efetuado por Smith & Butler (2002). Esse tipo abrange, por exemplo, o tipo HardwarePlatformType, ilustrado na Figura 3.4. A Figura 3.5 (b), por sua vez, apresenta uma instância de dispositivo que possui, por exemplo, a característica ColorCapable declarada.

Assim, a extensão da representação de contexto referente aos dispositivos foi capaz de suprir a carência de especificação de características peculiares de dispositivos heterogêneos. Além disso, essa extensão provê informações referentes aos tipos primários de contexto tradicionais, abrangendo, portanto, uma ampla variedade de informações contextuais de dispositivos.

3.1.2 Representação de Contexto de Rede

Assim como a representação de informações de contexto referentes aos dispositivos, a representação de informações de contexto de rede também é muito importante. A relevância dessa representação é evidenciada pela necessidade de se conhecer a situação do estado da rede. Com informações referentes a métricas de rede, é possível, por exemplo, adequar a qualidade do vídeo a ser entregue às condições atuais da rede, a qual é responsável pela comunicação

entre o dispositivo do usuário e o sistema. Portanto, além da adaptação de conteúdo aos dispositivos, é necessário manipulá-lo no intuito de adequá-lo ao estado da rede também, pois não há possibilidade de entrega de conteúdo com a qualidade requisitada pelo usuário se a rede está congestionada, por exemplo.

Tendo em vista a exploração de informações contextuais de rede, o presente trabalho apresenta uma modelagem de informações contextuais. No entanto, para apresentar a modelagem da representação, conceitos introdutórios da área de redes de computadores serão descritos, além da apresentação de algumas considerações.

Neste trabalho a definição de redes de computadores considerada é a de Tanenbaum (2003), que define uma rede como uma coleção de computadores autônomos interconectados. Há características que pertencem exclusivamente a redes. É o caso, por exemplo, de aspectos como escala, topologia e velocidade (10Mbps, 1Gbps, etc.). A escala da rede identifica o tamanho da mesma, a qual pode ser classificada como LAN - *Local Area Network*, rede local, MAN - *Metropolitan Area Network*, rede metropolitana ou WAN - *Wide Area Network*, rede de longa distância. A topologia, por sua vez, pode ser classificada como sendo do tipo Barramento (*Bus*), Anel (*Ring*) e Estrela (*Star*), por exemplo.

Ainda, uma rede de computadores é formada por elementos de rede, definidos neste trabalho como qualquer dispositivo que possui um endereço IP (*Internet Protocol*), aspecto justificado pela necessidade de gerenciamento desses dispositivos, mencionada nas seções seguintes. Desse modo, possuem tal classificação, dispositivos como *hosts*, *switches* inteligentes, *gateways* e roteadores.

A medida largura de banda foi utilizada durante a modelagem da representação contextual, como exemplo para ilustração do modelo. Assim, é importante ressaltar a relevância dessa métrica. Partindo do princípio que todo elemento de rede compete por largura de banda, todos são, portanto, capazes de prover valores para medição da mesma. Esses valores podem ser utilizados para o cálculo da quantidade de banda ainda disponível e do uso da largura de banda pelos elementos que compõe a rede. A seguinte situação é capaz de ilustrar esta afirmação: se há apenas um usuário conectado à rede, o protótipo de TV Interativa pode fornecer conteúdo com a máxima configuração aceita pela rede (como um vídeo com um número máximo de *frames* por segundo), usando, portanto, toda a largura de banda disponível naquele momento. No entanto, se mais usuários se conectam ao serviço, todos eles estarão competindo por recursos (largura de banda). Logo, o protótipo poderia reavaliar a largura de banda disponível e reconfigurar o vídeo a fim de

adaptá-lo às condições atuais da mesma.

A capacidade de provimento de valores para medição da largura de banda resulta em uma redução na complexidade de obtenção dessa informação contextual. Contudo, não indica o modo (*How*) como tal informação deve ser acessada. Portanto, é necessário explorar recursos capazes de efetuar o cálculo da largura de banda baseados em valores que podem ser obtidos por meio de análise da rede. Para isso, é apresentada uma fórmula baseada em parâmetros providos pelo protocolo SNMP (*Simple Network Management Protocol*) (CISCO, 2003).

O protocolo SNMP, padrão TCP/IP (*Transmission Control Protocol / Internet Protocol*) para gerenciamento de rede, define uma série de operações, suas funções e descrições. Assim, todo dispositivo deve manter informações de controle e estado próprios para que o gerenciamento desse dispositivo possa ser efetuado. Por exemplo, um roteador mantém informações estatísticas referentes ao estado de suas interfaces de rede, tráfego de pacotes recebidos e/ou enviados, perda de pacotes, mensagens de erro, etc.

Embora o SNMP forneça acesso a informações estatísticas, esse protocolo não especifica exatamente quais informações podem ser acessadas em dispositivos. O padrão responsável por especificar os detalhes de cada tipo de dispositivo é o MIB - *Management Information Base*, base de informação de gerenciamento. O MIB especifica as informações que devem ser fornecidas pelos dispositivos gerenciados, além das operações permitidas em cada dispositivo. Esse protocolo divide o gerenciamento de informações em diversas categorias. Dentre elas são exemplos a categoria de sistema, responsável por fornecer informações referentes ao sistema operacional e a categoria de interfaces que provê informações sobre a interface de rede (Comer, 2000).

O funcionamento do protocolo SNMP consiste de programas servidores que são executados em todas as máquinas gerenciadas e de um programa cliente executado no *host* gerente. Para obter informações referentes a algum *host* da rede que está sendo gerenciado, o gerente envia uma requisição ao programa servidor que está em execução nesse *host* e então o servidor responde ao gerente com a informação solicitada.

A Figura 3.6 apresenta a fórmula utilizada para o cálculo da largura de banda individual dos *hosts*. Os parâmetros utilizados na fórmula e a descrição de seus valores são apresentados na Tabela 3.1.

O cálculo apresentado requer que sejam medidas as variações de *bytes* recebidos (variação do parâmetro *ifInOctets*) e de *bytes* transmitidos pela mesma (variação do parâmetro *ifOutOctets*) interface de rede. Depois de incrementados, esses valores são multiplicados por oito. Há a necessidade dessa multiplicação porque os valores desses parâmetros estão em *bytes* e o parâme-

$$\frac{(\Delta \text{ifInOctets} + \Delta \text{ifOutOctets}) * 8 * 100}{\Delta t * \text{ifSpeed}}$$

Figura 3.6: Fórmula utilizada para o cálculo da largura de banda utilizada individualmente pelos *hosts*. Adaptada de (CISCO, 2003).

tro *ifSpeed* (Tabela 3.1) é dado em número de *bits*. Portanto, para converter o resultado da soma das variações *ifInOctets* e *ifOutOctets* em *bits* é preciso multiplicá-lo por oito. A multiplicação por cem é utilizada apenas para que o resultado seja equivalente ao percentual da largura de banda utilizada (CISCO, 2003). A variação de tempo corresponde ao intervalo em que foram obtidos os valores momentâneos *ifInOctets* e *ifOutOctets* (Figura 3.6).

Tabela 3.1: Parâmetros SNMP referentes à interface de rede e suas descrições.

| Parâmetros | Descrição |
|--------------------|------------------------------------|
| <i>ifInOctets</i> | Número total de bytes recebidos |
| <i>ifOutOctets</i> | Número total de bytes transmitidos |
| <i>ifSpeed</i> | Número de bits por segundo |

Embora seja viável o uso de parâmetros SNMP para o cálculo da largura de banda individual, alternativas podem ser abordadas. É o caso da utilização de Qualidade de Serviço (*Quality of Service - QoS*), por exemplo, por meio do protocolo RSVP (*Resource Reservation Protocol*) (Ribeiro, 2002). Nesse caso, deve haver a solicitação de necessidades de QoS à rede, como largura de banda, por meio desse protocolo, o qual solicita à rede a garantia da QoS conveniente (reserva) à aplicação. Logo, a rede “aceita” eventualmente a solicitação e “tenta garantir” a reserva solicitada (Ribeiro, 2002). Se há “tentativa de garantia” da reserva solicitada, é possível estimar quanto da largura de banda está “reservado” para um cliente. No entanto, vale lembrar que o protocolo SNMP possui uma variedade de parâmetros de sistema e de interface de rede, os quais podem ser utilizados, por exemplo, como fontes para informações contextuais de rede.

O Modelo de Representação de Contexto de Rede

De acordo com as considerações e conceitos apresentados, de modo geral, uma rede é formada por atributos específicos e elementos de rede. Esses elementos têm propriedades comuns entre si. Porém, alguns deles possuem propriedades distintas que caracterizam suas funções. É o caso de roteadores, por exemplo, que possuem tabelas de rotas. São características comuns entre elementos de rede, endereço IP, largura de banda e a presença de interfaces de

comunicação. Por serem consideradas características comuns, tais características foram relacionadas ao tipo `NetworkElementType`. Portanto, entidades para representação de tipos de elementos de rede herdarão as características comuns. Neste trabalho foi construído um tipo básico de elemento de rede: o tipo `HostType`. Esse tipo é utilizado para representar entidades que são estações de trabalho (*hosts*) pertencentes à rede. A Figura 3.7 ilustra a declaração desse tipo.

```
1 <schema targetNamespace="http://www.icmc.usp.br/-rudinei/Context/"
  ...
2   <complexType name="HostType">
3     <complexContent>
4       <extension base="ctx:NetworkElementType"/>
5     </complexContent>
6   </complexType>
  ...
7 </schema>
```

Figura 3.7: Tipo `HostType`.

Além da criação do tipo `HostType` utilizado para representação de uma entidade de rede, houve a necessidade de indexação das propriedades dos elementos de rede no contexto primário, descrito na seção 2.4. Desse modo, tipos de informações contextuais foram criados para suportá-las. Por exemplo, o item (a) da Figura 3.8 ilustra a declaração do tipo `NetPropertiesType`, o qual possui um elemento que descreve qual é o endereço IP do *host* (`<HostAddress>`), característica que provê a identidade do *host*, sendo mapeado, portanto, para o contexto primário `Identity` (linha 6, item (a)). Já a medida largura de banda foi mapeada para o tipo de contexto primário `How`, por informar quanto dessa medida está sendo utilizada por cada *host* e, conseqüentemente, quanto ainda está disponível.

Conforme especificado, o cálculo da largura de banda é efetuado utilizando-se parâmetros obtidos pelo uso do protocolo SNMP (`ifInOctets`, `ifOutOctets` e `ifSpeed`). Desse modo, foram criados dois tipos que suportam os parâmetros usados no cálculo: o tipo `InterfacesType`, e o tipo `BandwidthType`. O tipo `InterfacesType` é mapeado para o tipo de contexto primário `Identity` por proporcionar informações referentes às interfaces. É utilizado devido à possibilidade dos *hosts* possuírem mais de uma interface de rede. Sendo assim, o elemento `<Interface>` foi criado para suportar o elemento `<ifSpeed>` que contém o valor referente ao parâmetro `ifSpeed` (Tabela 3.1) de cada interface de rede. O tipo `BandwidthType` é mapeado para o tipo de contexto primário `How`, por tratar do estado das interfaces de rede. Esse tipo é formado por elementos `<Throughput>`, responsáveis por informar as variações de bytes recebidos e enviados pela interface de rede por meio dos elementos `<ifInOctets>` e `<ifOutOctets>`, respectivamente. O atributo `id` dos elementos `<Throughput>` informa

a seqüência de captura dos valores dos parâmetros `ifInOctets` e `ifOutOctets`.

```

1 <schema targetNamespace="http://www.icmc.usp.br/~rudinei/Context/"
...
2 <complexType name="NetPropertiesType">
3 <complexContent>
4 <extension base="ctx:IdentityType">
5 <sequence>
6 <element name="HostAddress" type="ctx:IPType"/>
...
7 </sequence>
8 </extension>
9 </complexContent>
10 </complexType>
11 </schema>

```

(a)

```

...
1 <Network NetBandwidth="0.125272" Scale="LAN" Topology="Bus" Speed="10Mbps">
2 <NetworkElement xsi:type="HostType">
3 <Identity xsi:type="NetPropertiesType">
4 <HostAddress>192.168.0.1</HostAddress>
5 </Identity>
6 <Identity xsi:type="InterfacesType">
7 <Interface>
8 <ifSpeed>1000000</ifSpeed>
9 </Interface>
10 </Identity>
11 <How xsi:type="BandwidthType">
12 <Throughput id="1">
13 <ifInOctets>360947</ifInOctets>
14 <ifOutOctets>1475691</ifOutOctets>
15 </Throughput>
16 <Throughput id="2">
17 <ifInOctets>363197</ifInOctets>
18 <ifOutOctets>1489246</ifOutOctets>
19 </Throughput>
20 </How>
21 </NetworkElement>
22 </Network>
...

```

(b)

Figura 3.8: (a) Declaração do tipo `NetPropertiesType`. (b) Instância de rede.

As características específicas da rede, como escala (*Scale*), topologia (*Topology*) e velocidade (*Speed*) foram mapeadas como atributos do elemento `<Network>`. Ainda, o atributo largura de banda utilizada da rede (`NetBandwidth`) também foi inserido como atributo desse elemento. Esse valor é fornecido dinamicamente efetuando-se o cálculo da média das larguras de banda individuais, as quais foram obtidas por meio da utilização da fórmula ilustrada na Figura 3.6. A Figura 3.8, item (b), apresenta uma instância válida do modelo de representação contextual descrito. No exemplo, são apresentados os valores dos parâmetros SNMP utilizados para o cálculo da largura de banda do *host* identificado pelo endereço IP `192.168.0.1` (linha 4).

3.2 O Serviço de Gerenciamento

A responsabilidade do serviço de gerenciamento de contexto é manipular informações contextuais. Para tal, houve divisão de tarefas entre dois mó-

dulos: o *Web Service* Gerente de Contexto e o Coletor Contextual. A Tabela 3.2 ilustra os componentes do serviço de gerenciamento de contexto e suas respectivas operações.

Tabela 3.2: Funcionalidades dos elementos do Serviço de Gerenciamento de Contexto.

| Componentes | Descrição |
|---------------------|--|
| Coletor Contextual | Coleta de informações contextuais de rede |
| Gerente de Contexto | Disponibilidade de informações de contexto |

3.3 O Coletor Contextual

O Coletor Contextual é responsável pela captura de informações de contexto. Num primeiro momento, apenas informações contextuais provenientes da rede são capturadas. O módulo coletor ainda efetua a construção da árvore de representação contextual da rede, apresentada na Subseção 3.1.2. Alguns elementos dessa representação possuem valores estáticos, inseridos automaticamente pelo coletor. É o caso dos atributos de rede *Scale*, *Topology* e *Speed*. Os valores dinâmicos são referentes aos parâmetros SNMP utilizados para o cálculo das larguras de banda individuais baseado na fórmula ilustrada na Figura 3.6. Além desses valores, o atributo de rede *NetBandwidth* também é calculado dinamicamente pelo módulo.

A Figura 3.9 ilustra o funcionamento do módulo coletor. O Coletor Contextual é a implementação de um cliente SNMP. Esse cliente efetua requisições periódicas para coletar parâmetros SNMP de todos os elementos da rede, os quais possuem servidores SNMP (passo 1). Depois de efetuadas as coletas (passo 2), que proporcionam valores referentes às variações *ifInOctets*, *ifOutOctets* e o valor referente ao parâmetro *ifSpeed* (Figura 3.6), o módulo coletor efetua o cálculo das larguras de banda individuais dos elementos de rede baseado na fórmula ilustrada na Figura 3.6 (passo 3). Em seguida, os valores obtidos são incrementados e o resultado é dividido pelo número de elementos, proporcionando, portanto, a largura de banda utilizada em toda rede pelos elementos que a compõe. Finalmente, o valor da largura de banda utilizada na rede (*NetBandwidth*) é inserido na representação de contexto (passo 4) (Figura 3.8 item (b), linha 1), bem como os valores dos parâmetros SNMP de cada elemento obtidos durante a coleta. Depois, a representação é armazenada na base de informações contextuais (passo 5).

É importante ressaltar que a escolha do protocolo SNMP ocorreu devido à influência de recursos utilizados há algum tempo no protótipo TV-Interativa. As aplicações responsáveis pela manipulação de vídeo no projeto utilizam uma

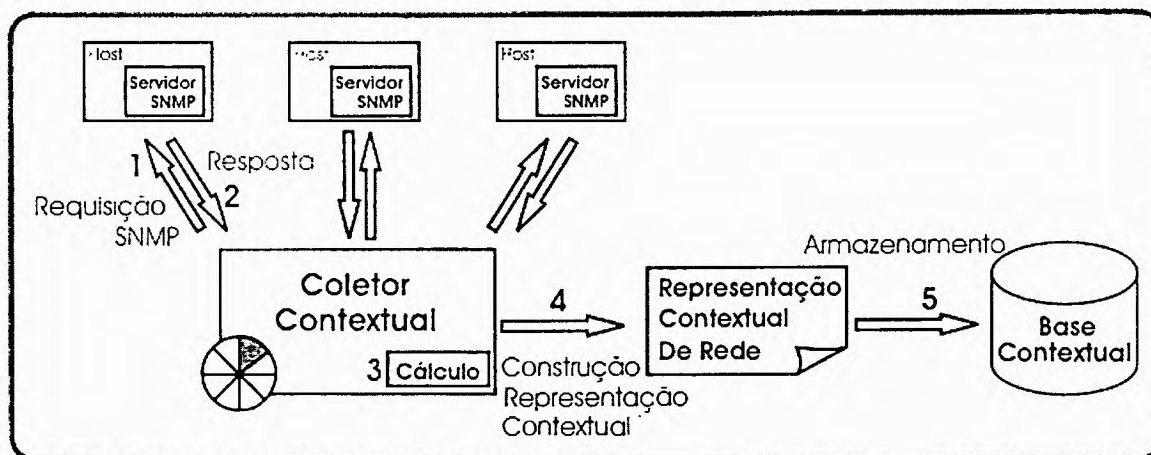


Figura 3.9: Fluxograma do Coletor Contextual.

ferramenta chamada *SNMPview*¹ que utiliza uma biblioteca proprietária das aplicações de manipulação. Essa biblioteca é responsável por cálculos relacionados a algumas medidas de rede, por exemplo, largura de banda. No entanto, por ser uma biblioteca proprietária, não é possível isolá-la da ferramenta *SNMPview*, ou ainda usá-la de modo independente. Portanto, a opção encontrada foi a implementação de um cliente SNMP com o auxílio da fórmula de cálculo de utilização de largura de banda evidenciada na seção 3.1.2. Para tal implementação foi utilizado um pacote de código aberto JAVA que implementa o protocolo SNMP². O pacote provê suporte a operações SNMP básicas definidas nas duas primeiras versões do protocolo.

O módulo coletor mantém a consistência dos dados contidos na árvore de representação contextual de rede. Tal característica é possível devido à periodicidade com que o programa efetua a coleta dos parâmetros SNMP. Garantindo a consistência das informações estruturadas na árvore de representação contextual (Figura 3.8), essa representação é consumida pelo *Web Service Gerente de Contexto*, módulo que também compõe o serviço de gerenciamento de contexto.

3.4 O Web Service Gerente de Contexto

Web Services são sistemas de software identificados por uma URI (*Uniform Resource Identifier*) e definidos por um documento de descrição baseado na linguagem XML. Essa descrição deve permitir que outros sistemas possam identificar o sistema descrito e interagir com o mesmo de acordo com sua definição, utilizando mensagens baseadas em XML conduzidas por protocolos Internet (Bray et al., 2000).

¹<http://www.snmpview.de/>

²http://edge.mcs.drexel.edu/GICL/people/sevy/snmp/snmp_package_introduction.html

Por meio de *Web Services*, aplicações são capazes de expor suas funcionalidades como um serviço que pode ser invocado por outras aplicações que queiram se beneficiar desse serviço. A tecnologia de *Web Service* foi escolhida para a implementação porque fornecem interoperabilidade, flexibilidade e simplicidade a baixo custo de processamento, pois utiliza o protocolo HTTP para transporte e a linguagem XML para representação de informações. Além disso, devido à interoperabilidade fornecida pela tecnologia de *Web Service*, a comunicação entre aplicações e serviços pode ocorrer de forma transparente e independente da heterogeneidade de linguagens de programação, sistemas operacionais, dispositivos, tecnologias de rede e mecanismos de estruturação de dados. *Web Services* são baseados em especificações ainda em desenvolvimento, como o protocolo SOAP (*Simple Object Access Protocol*), o protocolo UDDI (*Universal Description, Discovery and Integration*) e a linguagem WSDL (*Web Service Description Language*), descritos a seguir.

O *Web Service* Gerente de Contexto é parte integrante do serviço de gerenciamento de contexto e tem como funcionalidade disponibilizar informações de contexto referentes às categorias: usuários, dispositivos, rede e segurança. Ainda, o *Web Service* Gerente de Contexto é baseado na modelagem de representação de contexto desenvolvida por Goularte (2003) e estendida neste trabalho.

3.4.1 O Protocolo SOAP

O Protocolo SOAP é utilizado por *Web Services* para troca de mensagens com aplicações (Mitra, 2003). Esse protocolo define um modelo sintático e semântico para o intercâmbio de mensagens XML entre sistemas de software, além de utilizar o protocolo HTTP para o transporte de mensagens.

A estrutura de uma mensagem SOAP é composta por dois elementos básicos: o cabeçalho (header) e o corpo (body) da mensagem. O cabeçalho é responsável por definir, por exemplo, informações de controle e informações referentes ao processamento da mensagem. Essa parte da mensagem SOAP é opcional. O corpo da mensagem é obrigatório, contendo, necessariamente, a informação a ser transmitida. É importante ressaltar que a especificação do protocolo SOAP não define quais os tipos de informações trocadas entre aplicações e *Web Services*. Definir informações para troca de mensagens SOAP é total responsabilidade das aplicações. Essa propriedade de abster-se dos tipos de informações trocadas confere ao protocolo SOAP a característica de ser extensível.

3.4.2 A Linguagem WSDL

A linguagem WSDL é usada para a descrição completa das mensagens que podem ser trocadas. A linguagem WSDL é usada entre aplicações e *Web Services*. É responsável por fornecer meios para que as aplicações saibam quais serviços estão disponíveis pelos *Web Services* e ainda por prover a forma como as aplicações devem invocar tais serviços, quais informações são necessárias para a requisição dos mesmos, bem como o formato das informações contidas na resposta. Esse formato de informações é tipado de acordo com a notação XML Schema. Um serviço consiste de uma coleção de operações descrita por uma interface. O acesso a essas operações pode ser feito por meio de identificadores URI, cada qual associado a uma operação.

3.4.3 O Protocolo UDDI

O protocolo UDDI é responsável pela publicação da definição WSDL na Internet utilizando um repositório de descrições de *Web Services*. Por meio dessa publicação, outros sistemas de software podem encontrar a descrição e, desse modo, utilizar o serviço apresentado pela descrição WSDL. O protocolo UDDI define um modo interoperável de publicar e descobrir dinamicamente informações referentes a *Web Services*, atuando como um serviço de DNS (*Domain Name System*). Por se tratar de um protótipo, o *Web Service Gerente de Contexto* não está publicado na Internet.

3.4.4 Disponibilidade de Informações Contextuais

O *Web Service Gerente de Contexto* formaliza um conjunto de operações baseado na linguagem XML. Esse serviço é responsável por disponibilizar informações contextuais referentes a usuários, dispositivos, rede e privacidade. É importante destacar que embora as informações referentes à privacidade sejam classificadas como informações de segurança, são disponibilizadas pelo *Web Service Gerente de Contexto* por se tratarem de informações contextuais também. No entanto, outro padrão de representação dessas informações é utilizado. Esse padrão é descrito mais adiante.

Para disponibilizar essas informações, o *Web Service* oferece às aplicações cientes de contexto uma interface de acesso às informações de contexto composta por um conjunto de operações. Assim, sete métodos foram implementados de modo a viabilizar a disponibilidade das informações contextuais: os métodos `getUser`, `getFullUser`, responsáveis por informações de contexto do usuário, `getDevice`, `getFullDevice`, que tratam do contexto de dispositivos, `getNetwork` e `getFullNetwork`, os quais provêm informações contextuais de

rede e, finalmente, `getPolicy`, responsável por fornecer políticas de privacidade.

Os métodos `getUser`, `getDevice` e `getNetwork` provêm informações de contexto primário (Identity, Location, Time, Activity ou How). Como resposta, retornam um fragmento XML referente à informação do contexto primário requisitada, caso essa esteja presente na árvore de representação contextual. Esse fragmento é encapsulado pelo elemento XML que identifica o tipo da informação contextual que foi requisitada. Por exemplo, se uma aplicação requisita o contexto primário How referente à árvore de representação contextual de um dispositivo, o resultado da requisição, se tal contexto primário existir na representação, será encapsulado pelo elemento `<device>` (linhas 1 e 11, Figura 3.10). O mesmo processo é efetuado nos métodos `getUser` e `getNetwork`, porém, os elementos que encapsulam os resultados das requisições são, respectivamente, `<user>` e `<network>`.

Para o método `getUser` são exigidos dois parâmetros: a identificação do usuário e o tipo de informação de contexto requerida, no caso do exemplo apresentado na Figura 3.10, a informação How (linhas 3 e 10). Por sua vez, o método `getDevice` recebe como parâmetros a identificação do dispositivo, a identificação do usuário e o tipo de informação de contexto primário. É importante ressaltar que a identificação do usuário é necessária devido à possibilidade de utilização de dispositivos heterogêneos no contexto do protótipo de TV Interativa. Desse modo, além de efetuar a identificação do dispositivo, o *Web Service* Gerente de Contexto deve associá-lo a um usuário. Finalmente, o método `getNetwork` recebe apenas a informação de contexto requisitada, já que há apenas uma representação contextual de rede.

```
1 <device xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2     xmlns:ccpp="http://www.wapforum.org/profiles/UAPROF/ccppschema-20020709#">
3   <How xsi:type="CCPPfeaturesType">
4     <ccpp:component>
5       <ccpp:HardwarePlatform>
6         <ccpp:ColorCapable>true</ccpp:ColorCapable>
7         <ccpp:Model>TOSHIBA</ccpp:Model>
8       </ccpp:HardwarePlatform>
9     </ccpp:component>
10  </How>
11 </device>
```

Figura 3.10: Fragmento XML retornado como resposta à requisição do contexto primário How referente ao dispositivo.

Os métodos `getFullUser`, `getFullDevice` e `getFullNetwork` retornam todas as informações de contexto contidas nas representações contextuais de usuário, dispositivo e rede, respectivamente. O método `getFullUser` recebe como parâmetro a identificação do usuário, enquanto que o método `getFullDevice` requer os parâmetros: identificação do dispositivo e do usuário. O método `getFullNetwork` não possui parâmetros por haver uma única representação

contextual. A Figura 3.8 item (b) ilustra um exemplo de resposta do método `getFullNetwork`.

O método `getPolicy` é responsável por prover as informações referentes às políticas de privacidade. Essas políticas são utilizadas no intuito de prover acesso a conteúdo. Na verdade, ao prover as informações contidas nas políticas de privacidade, o método `getPolicy` fornece subsídios a aplicações cientes de contexto para que essas sejam capazes de tomar decisões relacionadas ao acesso ou à restrição de conteúdo. Esse método recebe como parâmetro a identificação da política requisitada e provê como resposta (Figura 3.11) políticas de privacidade que são descritas no formato da linguagem EPAL (*Enterprise Privacy Authorization Language*).

A linguagem EPAL é utilizada para troca de políticas de privacidade entre aplicações por meio de um formato estruturado. Essa linguagem foi desenvolvida pela IBM (*Instructions Business Machine*) Research no intuito de fornecer subsídios para que empresas possam codificar suas práticas e políticas de privacidade relacionadas à manipulação de dados (Schunter et al., 2003).

Uma política de privacidade EPAL categoriza os dados de uma empresa e as regras que regem o uso dos dados de cada categoria. Como a linguagem EPAL é projetada para abranger políticas de privacidade em diversas áreas, não pode pré-definir os elementos que compõem a política de privacidade. Para isso, a linguagem provê o vocabulário EPAL, um mecanismo para definição dos elementos que são usados para construção da política de privacidade. Esse vocabulário é responsável por definir uma linguagem utilizada para expressar as políticas de privacidade, ou seja, os elementos que compõem a política.

Os tipos de elementos que compõem o vocabulário EPAL são: `<user-category>`, responsável pela definição das categorias de usuários descritas na política de privacidade, `<data-category>`, que define as categorias de dados que terão sua privacidade controlada pelas regras da política, `<purpose>`, representando as finalidades para o acesso aos dados, `<action>`, responsável pela especificação de quais ações são permitidas e negadas de acordo com alguma circunstância em particular, e `<obligation>`, que define obrigações efetuadas pelo sistema de acordo com a ação (*action*) ocorrida, por exemplo, retenção de uma determinada informação por um mês. Exemplos de valores de cada categoria descrita no vocabulário EPAL estão ilustrados na Tabela 3.3.

A política EPAL consiste de uma lista de regras de privacidade baseadas nos elementos definidos no vocabulário EPAL. A Figura 3.11 ilustra um exemplo de política EPAL com uma regra³. Como toda política EPAL é projetada com base num vocabulário EPAL, há uma referência para tal vocabulário (linha

³Milagres, F. G. *Segurança em computação baseada em informações de contexto*. A ser editado pelo ICMC/USP, 2004.

Tabela 3.3: Exemplos de valores das categorias que compõem o vocabulário EPAL.

| Categoria | Descrição |
|------------------|----------------------------|
| <user-category> | Funcionário |
| <data-category> | Modelagem |
| <purpose> | Desenvolvimento do projeto |
| <condition> | Horário de expediente |
| <action> | Acesso |
| <obligation> | Efetuar log do acesso |

8). Uma política EPAL é formada por informações referentes a essa política (linhas 3 a 7), definições de elementos <condition> (linhas 8 a 19 e 20 a 30), e os elementos <rule> (linhas 31 a 39).

Uma regra determina que uma categoria de usuário (<user-category>) possui permissão ou restrição a uma ação (<action>) sobre um determinado dado de uma dada categoria (<data-category>) para um determinado propósito (<purpose>) se e somente se as condições (elementos <condition>) relacionadas a essa ação são satisfeitas, e que, conseqüentemente, determinadas obrigações (elemento <obligation>) devem ser executadas Schunter et al. (2003).

No exemplo apresentado duas condições devem ser satisfeitas: se o funcionário encontra-se nas dependências da empresa (linhas 9, 15 a 17) e se o funcionário está em horário de expediente (linhas 20, 26 a 28). No exemplo da Figura 3.11, o elemento <rule> que descreve a regra é composto por um atributo *ruling* (linha 31) que expressa qual será a decisão tomada se a regra ocorrer, a categoria de usuário (linha 32), a categoria de dados (linha 33), o propósito (linha 34), a ação (linha 35), as condições referentes à ação (linhas 36 e 37) e a obrigação (linha 38).

Atualmente na versão 1.2, a linguagem EPAL está em processo de avaliação do W3C, tendo sido submetida para tal pela IBM Research em 10 de Novembro de 2003. Maiores detalhes podem ser obtidos com o grupo de desenvolvimento da linguagem EPAL (Schunter et al., 2003).

3.5 Modelagem do Serviço de Gerenciamento de Contexto

A Figura 3.12 mostra um diagrama de classes UML que ilustra o serviço de gerenciamento de contexto (*Context Manager Infrastructure*) composto pelo Coletor Contextual (*Contextual Collector*) e pelo Web Service Gerente de Contexto (*Context Manager Service*). Esse último, apresentado de forma mais detalhada

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <epal-policy default-ruling="deny" global-condition="" version="1.0">
3 <policy-information id="politica">
4 <issuer/>
5 <location>c:\temp\politica.xml</location>
6 <version-info/>
7 </policy-information>
8 <epal-vocabulary-ref id="" location="c:\temp\vocabulario.xml" revision=""/>
9 <condition id="local_trabalho">
10 <evaluates-container refid="context_container"/>
11 <xacml:Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of"
12 xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy">
13 <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:IS EQUAL TO">
14 <xacml:EnvironmentAttributeDesignator
15 AttributeId="urn:ibm:epal:attribute:context_container:location"
16 DataType="http://www.w3.org/2001/XMLSchema#string"/>
17 </xacml:Apply>
18 <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
19 empresa
20 </xacml:AttributeValue>
21 </xacml:Condition>
22 </condition>
23 <condition id="horario_expediente">
24 <evaluates-container refid="context_container"/>
25 <xacml:Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of"
26 xmlns:xacml="urn:oasis:names:tc:xacml:1.0:policy">
27 <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:IS EQUAL TO">
28 <xacml:EnvironmentAttributeDesignator
29 AttributeId="urn:ibm:epal:attribute:context_container:time"
30 DataType="http://www.w3.org/2001/XMLSchema#string"/>
31 </xacml:Apply>
32 <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#boolean">
33 true
34 </xacml:AttributeValue>
35 </xacml:Condition>
36 </condition>
37 <rule id="expediente_na_empresa" ruling="allow">
38 <user-category refid="funcionario"/>
39 <data-category refid="modelagem"/>
40 <purpose refid="desenvolvimento_projeto"/>
41 <action refid="access"/>
42 <condition refid="local_trabalho"/>
43 <condition refid="horario_expediente"/>
44 <obligation refid="log_do_acesso"/>
45 </rule>
46 </epal-policy>

```

Figura 3.11: Exemplo de Política EPAL.

na Figura 3.13, faz referência ao documento de descrição WSDL, às representações contextuais de usuários (*User Context Information*), de dispositivos (*Devices Context Information*), de segurança (*Access Rule*) e de rede (*Network Context Information*).

A representação de informações contextuais de segurança é fornecida pelo gerente de segurança, omitido no diagrama por não fazer parte do serviço de gerenciamento de contexto. Já a representação de informações contextuais referentes à rede é provida pelo Coletor Contextual (*Contextual Collector*), o qual é dependente do protocolo SNMP (Figura 3.13).

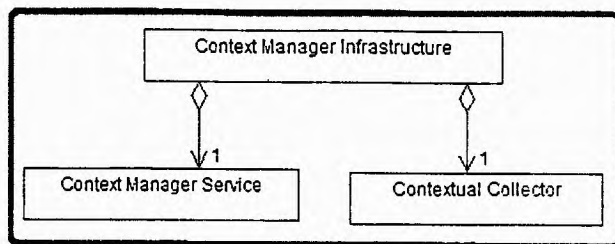


Figura 3.12: Diagrama de classes UML representando o Serviço de Gerenciamento de Contexto.

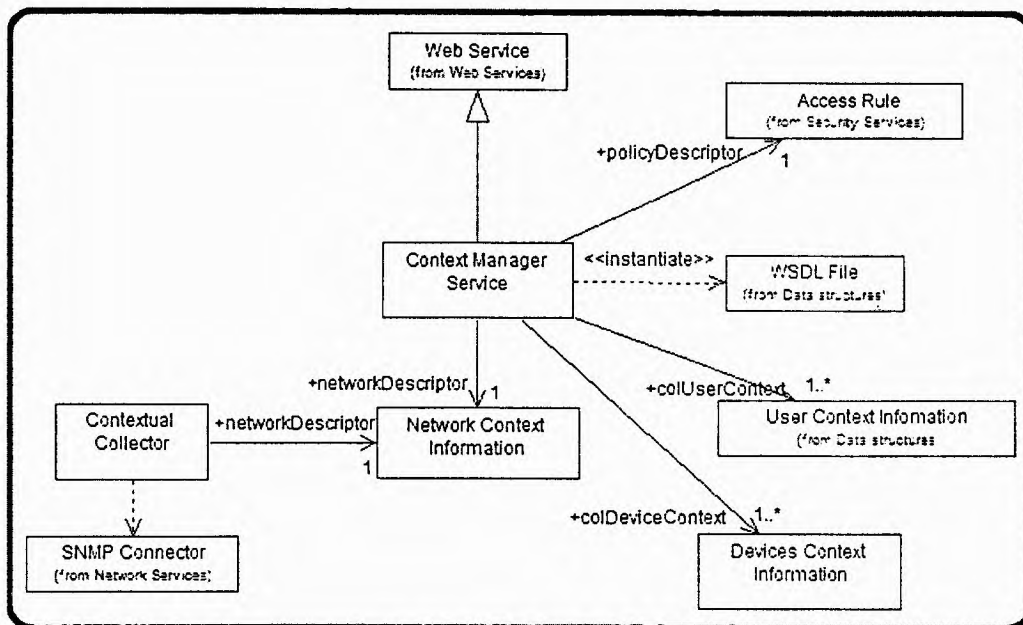


Figura 3.13: Diagrama UML do Web Service Gerente de Contexto.

3.6 A Arquitetura do Serviço de Gerenciamento de Contexto

A arquitetura do serviço de gerenciamento de contexto é dividida em seis camadas, como ilustra a Figura 3.14. A camada de aplicação é composta pelas aplicações que serão beneficiadas pelo *Web Service* Gerente de Contexto. A camada de descrição é responsável pela publicação do documento WSDL que descreve os serviços fornecidos pelo *Web Service*. A camada de transporte representa a troca de mensagens SOAP sobre o protocolo HTTP. Essa camada efetua a comunicação entre as aplicações e o *Web Service* Gerente de Contexto, uma vez que aplicações estão de posse da descrição WSDL, necessária para que as aplicações conheçam as operações fornecidas pelo *Web Service*.

A camada de serviços descreve os serviços fornecidos pelo *Web Service* Gerente de Contexto, que são explicados mais adiante. A camada contextual constitui as informações contextuais providas pelo *Web Service*. Nessa ca-

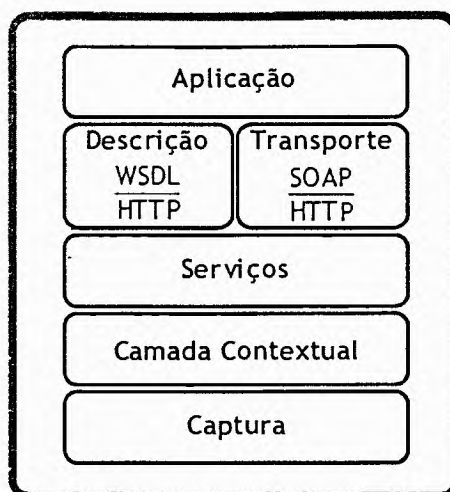


Figura 3.14: A arquitetura de camadas do Serviço de Gerenciamento de Contexto.

mada encontram-se informações referentes aos dispositivos, usuário e rede. Finalmente, a camada de captura é responsável pela obtenção dinâmica e atualização das informações contextuais de rede. Em tal camada está localizado o coletor de informações contextuais de rede, o Coletor Contextual. A plataforma de software utilizada durante a implementação do gerente de contexto inclui o sistema operacional Windows XP, o servidor Apache estendido com o *container* Tomcat e o pacote para desenvolvimento de *Web Services* JWSDP - (*Java Web Services Developer Package*), da Sun Microsystems.

3.7 O Fluxograma do Serviço de Gerenciamento de Contexto

A Figura 3.15 mostra a série de operações necessária para que uma aplicação seja capaz de utilizar as operações do *Web Service* Gerente de Contexto. Primeiramente é efetuada a publicação da descrição WSDL (passo 1). No passo 2, a aplicação consulta as informações contidas na descrição WSDL. Com base nessas informações, as aplicações são capazes de enviar mensagens utilizando o protocolo SOAP para o empacotamento das mesmas (passo 3).

Após receber a requisição, o *Web Service* Gerente de Contexto consulta a base de dados (passo 4) que, por conseguinte, retorna as informações solicitadas ao Gerente de Contexto (passo 5). Em seguida, a resposta, que pode ser um fragmento da árvore de representação contextual ou a árvore toda, é entregue à aplicação que a requisitou (passo 6). Vale ressaltar que mesmo não participando do processo de disponibilidade das informações contextuais às aplicações, o Coletor Contextual é responsável pela consistência das informações contextuais referentes à árvore de representação da rede. A árvore de

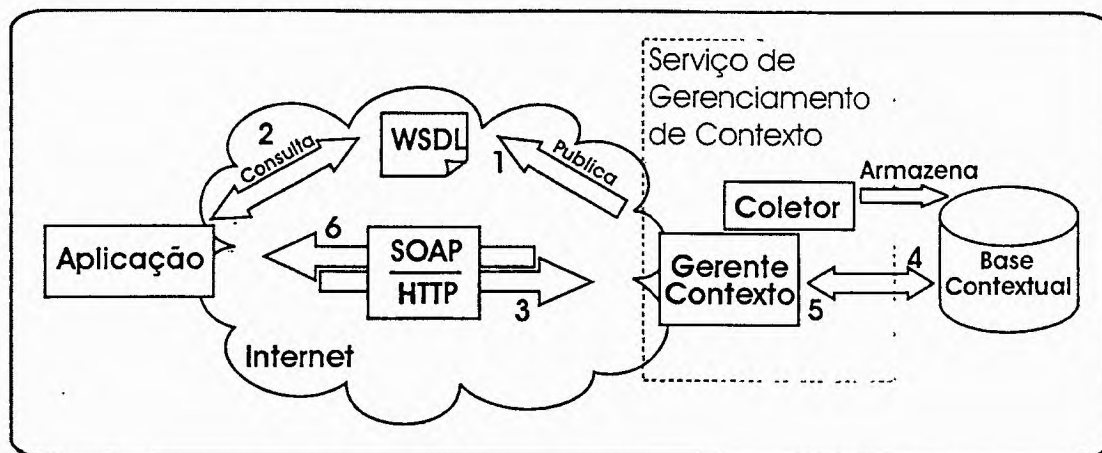


Figura 3.15: Fluxograma do Serviço de Gerenciamento de Contexto.

representação é atualizada periodicamente, de modo a garantir que o Gerente de Contexto sempre forneça informações condizentes com o estado da rede. A base de dados utilizada é a Oracle 9i, capaz de armazenar documentos XML. Porém, atualmente o acesso às informações contextuais é feito lendo-se arquivos referentes às representações de dispositivos, usuários, rede e privacidade.

3.8 Exemplo de Uso

No intuito de prover um exemplo de uso para o serviço desenvolvido neste trabalho, é apresentada nesta seção a implementação de uma aplicação ciente de contexto que faz uso de regras de acesso providas por um serviço de gerenciamento de privacidade. Por sua vez, esse serviço é beneficiado pelo uso das informações de contexto dos usuários providas pelo serviço de gerenciamento de contexto. Desse modo, o serviço de gerenciamento de privacidade é capaz de prover regras de acesso dinâmicas baseadas nas informações contextuais.

A aplicação ciente de contexto implementada é capaz de prover acesso ao conteúdo de um *site*, de acordo com informações de contexto do usuário e com uma política de privacidade desenvolvida na linguagem EPAL, a qual foi apresentada na Seção 3.4.4. Para o provimento do acesso, as informações de contexto utilizadas são referentes à identidade (quem é o usuário (*who*)), localização (*where*) e tempo (o momento em que o usuário está requisitando o acesso (*when*)) do usuário. Beneficiando-se dessas três categorias de informações é possível alcançar um grau de dinamismo, pois as informações relacionadas à localização e tempo não são estáticas.

Com essa idéia em mente, a seguinte situação ilustra um exemplo: um usuário é funcionário de uma determinada empresa. Para obter acesso ao *site*, o funcionário deve ser autorizado, estar nas dependências da empresa e ainda, efetuar o acesso em um intervalo temporal pré-estabelecido, por exem-

plo, em seu horário de expediente (entre 8:00 horas e 19:00 horas). O processo de obtenção, disponibilidade, análise das informações contextuais e construção da regra de acesso (permissão ou restrição ao conteúdo) é todo de responsabilidade do serviço de gerenciamento de informações de contexto e do serviço de gerenciamento de privacidade. Esse último é responsável pela geração de regras de acesso. Para tal são feitas análises das informações de contexto referentes ao usuário e da política de privacidade EPAL referente ao conteúdo. Baseando-se nessa análise o serviço é capaz de tomar decisões e, dessa maneira, permitir ou restringir o acesso ao conteúdo. Vale ressaltar que tal serviço está descrito em um outro trabalho do grupo ⁴.

A Figura 3.16 ilustra a implementação da aplicação ciente de contexto e dos serviços de gerenciamento de contexto e privacidade. Ao requerer o conteúdo (passo 1) o usuário fornece seu login, senha e localização (Figura 3.17). Essas informações são enviadas à aplicação que se encarrega de efetuar a autenticação do usuário no sistema (passo 2). Após a autenticação do usuário, as informações de contexto do mesmo (identificação, localização, e tempo) são coletadas pelo Coletor Contextual (passo 3) para atualização dinâmica (informações de localização e tempo) da representação de contexto do usuário (passo 4), a qual foi previamente construída.

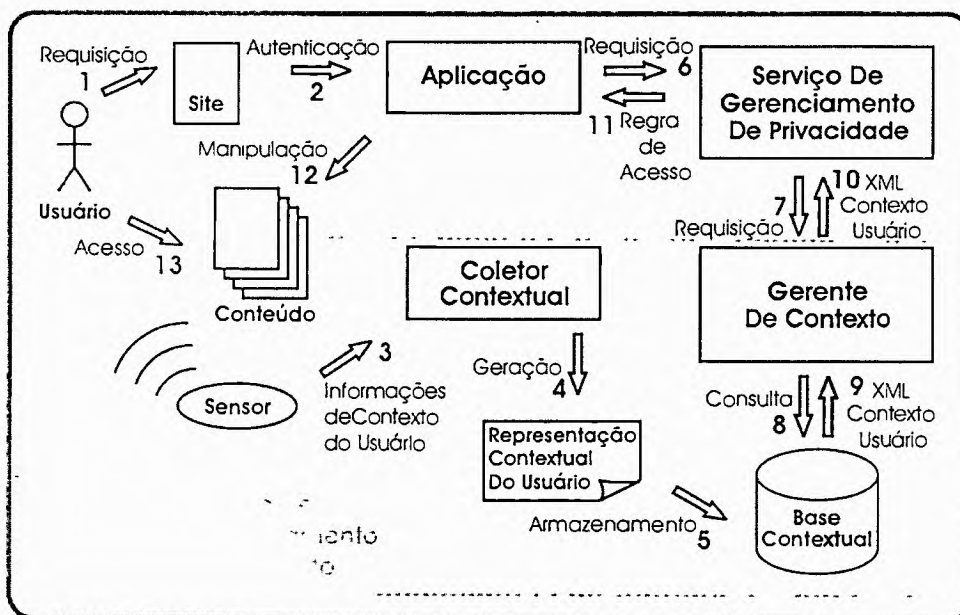


Figura 3.16: Fluxograma da interação entre a Aplicação, o Serviço de Gerenciamento de Privacidade e de Contexto.

É importante ressaltar que a informação contextual referente à localização do usuário é obtida explicitamente, com o usuário apontando sua localização (Figura 3.17). Embora essa alternativa não seja viável e, obviamente, não deve

⁴Milagres, F. G. *Segurança em computação baseada em informações de contexto*. A ser editado pelo ICMC/USP, 2004.

ser utilizada em aplicações cientes de contexto, ela está sendo usada no intuito de simular sensores que, futuramente, serão responsáveis pela obtenção desse e de outros tipos de informações contextuais, como no trabalho de Want et al. (1992). Além disso, é importante destacar também que embora o Coletor Contextual seja responsável pela atualização da representação de contexto do usuário efetuando portanto uma geração **semi-automática** de representação de contexto, seria ideal que esse módulo efetuasse a geração **automática** dessa representação, assim como é efetuado com a representação de contexto de rede descrita na Seção 3.3.

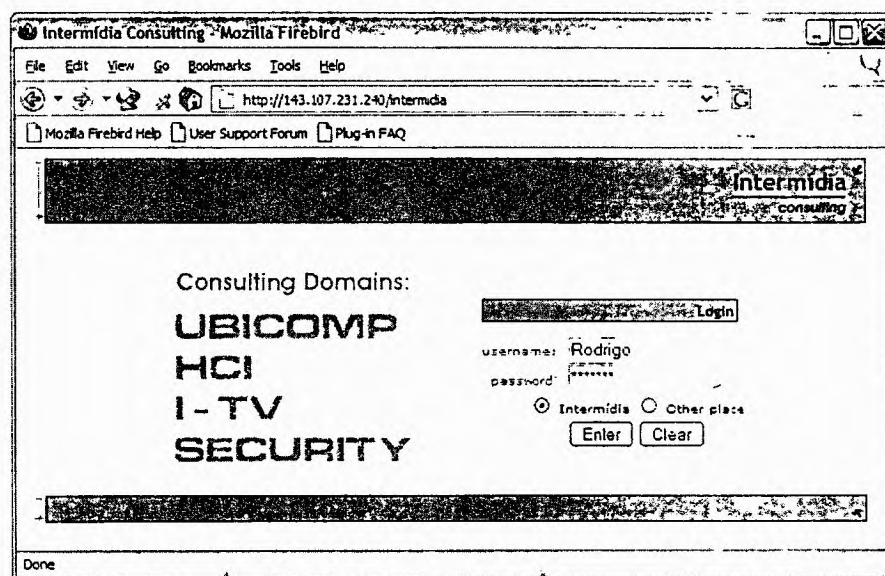


Figura 3.17: Tela de apresentação e *login* da aplicação ciente de contexto.

A instância de representação ilustrada na Figura 3.18 é um exemplo de representação de contexto do usuário atualizada pelo Coletor Contextual. Essa representação de contexto é então adicionada à base contextual para ser disponibilizada posteriormente pelo serviço de gerenciamento de contexto (passo 5).

Após a autenticação do usuário, atualização e armazenamento da representação contextual do mesmo, a aplicação requisita a regra de acesso referente ao usuário autenticado para o serviço de gerenciamento de privacidade (passo 6). Como o serviço de gerenciamento de privacidade necessita das informações de contexto do usuário para gerar a regra de acesso, ele então efetua a requisição da representação contextual do usuário, que foi armazenada na base contextual pelo coletor contextual (passo 5), ao Gerente de Contexto por meio de um dos métodos fornecidos por esse gerente para manipulação de informações de contexto de usuários (passo 7). Por sua vez, o Gerente de Contexto realiza uma consulta à base contextual (passo 8) e recupera a representação de contexto do usuário (passo 9), a qual é enviada como resposta à

requisição efetuada pelo serviço de gerenciamento de privacidade (passo 10). A Figura 3.18 apresenta a representação de contexto que é enviada como resposta ao serviço de gerenciamento de privacidade.

```

1 <Context xmlns:context="http://www.institutoayrtonsenna.com.br/Context/" ... >
2   <Users>
3     <User>
4       <Identity xsi:type="WhoType" ID="funcionario-01" >
5         <Name>
6           <GivenName>Ayrton</GivenName>
7           <FamilyName>Senna</FamilyName>
8         </Name>
9         <Role>analista</Role>
10        </Identity>
11        <Location xsi:type="WhereType">
12          <LocationName>Interlagos</LocationName>
13          <LocationDescription>dependências da empresa</LocationDescription>
14        </Location>
15        <Time xsi:type="context:WhenType">
16          <Date>2004-03-18</Date>
17          <Time>08:00-22:00</Time>
18        </Time>
19      </User>
20    </Users>
21  </Context>

```

Figura 3.18: Instância de representação de contexto de usuário.

Depois de receber a representação de contexto dinâmica do usuário, esse serviço efetua o *parsing* da mesma. Após extrair as informações relevantes para a geração da regra de acesso, o serviço de gerenciamento de privacidade consulta a política EPAL ilustrada na Figura 3.11. Em seguida, é feita uma análise das informações de contexto obtidas com as condições descritas na política EPAL. A partir dessa análise, o serviço de gerenciamento de privacidade dinamicamente gera e envia a regra de acesso apresentada na Figura 3.19 para a aplicação (passo 11) em resposta à requisição efetuada anteriormente (passo 6).

```

<?xml version="1.0" encoding="UTF-8"?>
<xacml:Rule RuleId="1" Effect="Permit">
  <xacml:Description>horário e localização compatíveis</xacml:Description>
</xacml:Rule>

```

Figura 3.19: Regra de acesso gerada dinamicamente pelo Serviço de Gerenciamento de Privacidade.

Por sua vez, a aplicação realiza a análise da regra de acesso e libera ou restringe o acesso ao conteúdo. Em caso de permissão, a aplicação manipula o conteúdo de modo a organizá-lo (passo 12) e, finalmente, libera o acesso do mesmo ao usuário que efetuou a requisição (Figura 3.20) (passo 13).

Embora o contexto em que essa aplicação está inserida seja simples, é possível utilizar informações de contexto dinâmicas com privacidade em cenários mais abrangentes. Por exemplo, em uma casa ciente de contexto, na qual

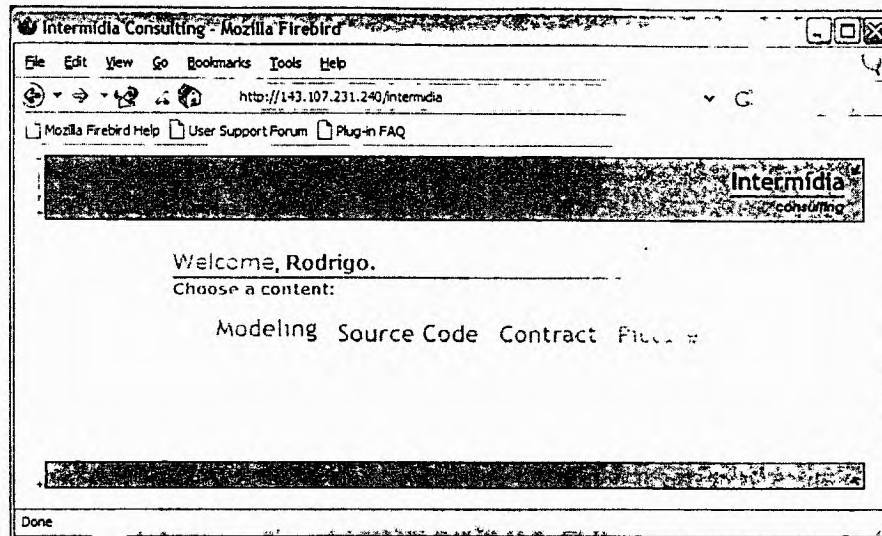


Figura 3.20: Tela de apresentação do conteúdo ao usuário após a verificação da regra de acesso.

habitam pessoas idosas. Cada pessoa possui medicamentos distintos com horários de medicação também distintos. Portanto, devem ter acesso apenas aos seus respectivos medicamentos para que não haja complicações futuras. O controle ao conteúdo, no caso o medicamento, pode ser efetuado de acordo com informações contextuais de identidade (quem é a pessoa que deve ter acesso ao remédio) e de tempo (horário condizente com a prescrição efetuada pelo médico). Se essas condições forem satisfeitas, a casa ciente de contexto pode destrancar o armário de medicamentos da pessoa e liberar o acesso ao remédio.

Um exemplo que considera informações de localização, tempo e identidade é o de um grupo de profissionais que faz uso de um sistema colaborativo ciente de contexto para construção de softwares. Esses profissionais podem exercer funções distintas, como consultoria, desenvolvimento e gerência de projetos. Os profissionais referentes a cada função devem ter acesso a um determinado tipo de conteúdo. Por exemplo, os profissionais responsáveis pelo desenvolvimento do projeto devem compartilhar o conteúdo referente aos códigos do projeto, os modelos, etc. Caso o projeto seja sigiloso, esse conteúdo deve ser acessado durante o horário de expediente e/ou se o profissional estiver nas dependências da empresa, por exemplo. Portanto, podem ser elaboradas representações contextuais de usuários de diferentes áreas profissionais (*Role*), que devem ter acesso ao conteúdo de acordo com a localização e horário de expediente do indivíduo.

É importante destacar que no exemplo de implementação descrito nesta seção, a aplicação se comunica diretamente com o serviço de gerenciamento de privacidade. Porém, por se tratar de um *Web Service*, qualquer aplicação

heterogênea é capaz de obter informações contextuais disponibilizadas pelo Gerente de Contexto de modo direto. O exemplo dá ênfase ao uso de privacidade baseada em informações contextuais, daí o uso indireto do Gerente de Contexto pela aplicação.

3.9 Considerações Finais

No presente capítulo foi apresentado o serviço de gerenciamento de contexto desenvolvido ao longo deste trabalho. Primeiramente, foi descrita a representação de contexto utilizada pelo serviço. Essa representação foi estendida do modelo proposto por Goularte (2003) no projeto de protótipo de TV Interativa, contexto do presente trabalho. Foi abordado o contexto de Infra-Estrutura, sobretudo o contexto referente aos dispositivos e o contexto de rede.

Para suprir a necessidade de representação contextual dos dispositivos, mais precisamente a representação da especificação dos dispositivos, foi utilizado o padrão W3C CC/PP, apresentado na Seção 3.1.1. Como o contexto de rede não havia sido explorado por Goularte (2003), este trabalho efetuou a especificação desse tipo de contexto, criando a entidade de rede *HostType*, bem como a modelagem e a implementação da representação contextual de rede.

Foram detalhados também os componentes que integram o serviço de gerenciamento de contexto: o Coletor Contextual (Seção 3.3) e o *Web Service* Gerente de Contexto (Seção 3.4). O primeiro é responsável pela captura de informações contextuais referentes ao estado atual da rede. Para essa tarefa, foi utilizada como exemplo a métrica largura de banda utilizada, a qual é calculada por meio da fórmula proposta pela empresa CISCO (CISCO, 2003). Essa fórmula faz uso do protocolo SNMP (Comer, 2000) para a computação do resultado. Já o Gerente de Contexto é capaz de disponibilizar informações contextuais por meio dos protocolos que compõem a tecnologia *Web Service*. Essa tecnologia também é descrita nas Subseções 3.4.1, 3.4.2 e 3.4.3. Além dos componentes do serviço de gerenciamento de contexto, são apresentados seus fluxogramas. São descritos também a modelagem, arquitetura e o fluxograma do serviço de gerenciamento de contexto. Por fim, na Seção 3.8 é apresentado um exemplo ilustrando o uso do serviço de gerenciamento de contexto.

A seguir, o Capítulo 4 descreve as conclusões e limitações deste trabalho e aponta trabalhos futuros com o objetivo de preencher as lacunas deixadas por esta pesquisa.

Conclusões

As pesquisas desenvolvidas na área de computação ubíqua contribuem para a convergência ao paradigma no qual a computação é incorporada às diversas atividades humanas de modo transparente e não intrusiva. Uma das áreas responsáveis por essa contribuição é a computação ciente de contexto que se beneficia de informações contextuais para aprimorar a interação usuário-computador (Abowd & Mynatt, 2000).

Trabalhos desenvolvidos nessa área apontaram a carência de projetos para o gerenciamento de informações contextuais baseados em uma representação de contexto extensíveis. Além disso, tópicos pouco explorados nessa área são: o uso de informações contextuais referentes à infra-estrutura do sistema, por exemplo, informações sobre dispositivos heterogêneos utilizados para acesso, bem como informações referentes ao estado da rede e o uso de modelos extensíveis de representação de informações contextuais capazes de prover interoperabilidade às aplicações.

O trabalho apresentado nesta dissertação foi desenvolvido com o objetivo de propor um serviço para aplicações cientes de contexto heterogêneas capaz de gerenciar informações contextuais baseado no modelo de representação contextual extensível desenvolvido por Goularte (2003). Outro objetivo deste trabalho foi estender esse modelo para que informações provenientes da infra-estrutura, mais precisamente de dispositivos e de rede sejam suportadas pelo mesmo. Desse modo, foi implementado um serviço de gerenciamento de informações contextuais capaz de manipular tais informações. Para garantir a interoperabilidade desse serviço entre aplicações heterogêneas foi explorada a plataforma *Web Service*. Essa característica possibilita a utilização do serviço de gerenciamento de contexto não apenas pelas aplicações desenvolvidas

no contexto do projeto protótipo de TV Interativa, mas também por quaisquer aplicações heterogêneas cientes de contexto.

4.1 *Discussão Sobre as Contribuições Deste Trabalho*

Uma das maiores contribuições deste trabalho foi a complementação da pesquisa de Goularte (2003) com a representação de contexto de Infra-Estrutura e também com o Serviço de Gerenciamento de Contexto. Com essas contribuições foi possível agregar valor ao protótipo de TV Interativa, que agora é capaz de manipular informações de contexto além das informações de Usuário. As principais contribuições deste trabalho são destacadas a seguir:

- Extensão do modelo de representação de informações contextuais de dispositivos. No modelo proposto por Goularte (2003) não havia representação relacionada às características específicas dos dispositivos. Para solucionar essa carência, foi utilizado o protocolo CC/PP (Klyne et al., 2004), que atendeu de modo satisfatório as necessidades almejadas por este trabalho.
- Especificação de tipos de informações contextuais responsáveis pela caracterização de entidades de rede. Como um dos objetivos do trabalho foi a exploração de informações contextuais de rede e o modelo utilizado não abordou esse tipo de informação contextual (Goularte, 2003), houve a necessidade de especificar características mínimas presentes nos elementos de rede.
- Desenvolvimento de um serviço de rede capaz de efetuar o cálculo da largura de banda utilizada e gerar a representação contextual da rede de acordo com o modelo de representação. A implementação desse software foi viabilizada pelo uso do protocolo SNMP (Comer, 2000) e pela fórmula apresentada pela CISCO (CISCO, 2003) para o cálculo da utilização de largura de banda.
- Modelo de manipulação de informações contextuais baseado nas informações de contexto primárias (Schilit et al., 1994; Brown et al., 1997; Pascoe, 1998; Dey & Abowd, 2000; Dey, 2001). O contexto primário é utilizado por Goularte (2003) para indexação de informações contextuais. Desse modo, o Gerente de Contexto provê respostas a requisições de acordo com esse modelo, disponibilizando informações de contexto referentes a usuário, dispositivos e rede no mesmo formato.
- Desenvolvimento de um conjunto de operações para disponibilidade de informações contextuais. Esse conjunto de operações disponibiliza in-

formações referentes ao contexto da Infra-Estrutura e do Usuário, além de abranger informações de privacidade, consideradas contextuais.

4.2 Limitações do Serviço de Gerenciamento de Contexto

O serviço de gerenciamento apresentado neste trabalho é capaz de gerenciar informações contextuais e provê-las a aplicações cientes de contexto. No entanto, por se tratar de uma versão inicial, há limitações que precisam ser abordadas:

- **Carência de coletores para categorias de contexto:** embora disponibilize informações contextuais referentes a dispositivos, usuários, rede e privacidade, o serviço de gerenciamento de contexto possui apenas um coletor responsável por obter informações contextuais de rede.
- **Dependência do protocolo SNMP:** o Coletor Contextual está implementado de modo dependente ao protocolo SNMP, pois captura informações provenientes dos servidores existentes em cada *host* pertencente à rede. Em virtude dessa dependência, a abordagem para captura utilizada neste trabalho pode ser realizada apenas em ambientes isolados, pois em um ambiente real, é praticamente impossível garantir a presença de servidores SNMP em pontos de rota intermediários.
- **Ausência de mecanismo de segurança:** não há restrições quanto ao acesso das informações contextuais providas pelo Gerente de Contexto. Portanto, há carência em relação à segurança do Serviço de Gerenciamento de Contexto, pois informações contextuais podem ter caráter privado.

4.3 Propostas Para Trabalhos Futuros

O serviço para gerenciamento de informações contextuais apresentado neste trabalho pode ser considerado passo inicial para o desenvolvimento de uma infra-estrutura para suporte ao contexto. No entanto, considerando-se a modelagem e a implementação desse serviço, foram constatados alguns tópicos para trabalhos futuros:

- Em relação à carência de coletores de informações de contexto, seria interessante se houvessem sensores responsáveis por outras categorias de

informações de contexto mencionadas no trabalho, ou talvez, que o Coletor Contextual fosse incrementado com módulos especializados para as outras categorias. Por exemplo, uma API responsável por coletar informações referentes às características do dispositivo e prover tais informações ao serviço Web Gerente de Contexto dinamicamente.

- A dependência do protocolo SNMP (Comer, 2000), por sua vez, não auxilia na captura de informações contextuais em um ambiente não-controlado por motivos de segurança. Novas alternativas devem ser abordadas com o objetivo de suprir essa limitação. Por outro lado, em um ambiente controlado, por exemplo em uma casa ciente de contexto, o uso do protocolo SNMP pode ser considerado satisfatório.
- Esforços devem ser designados também para preencher a carência de tipos de representação de informações contextuais de rede. Embora o tipo `HostType` tenha sido desenvolvido para esse propósito, em uma rede mais complexa com elementos como *gateways* ou roteadores, esse tipo não seria satisfatório devido às especialidades desses elementos. Outra carência relacionada à rede é a de informações contextuais. Alguns atributos de rede como topologia e escala, por exemplo, foram abordados. Parâmetros SNMP podem ser utilizados para prover um número maior de informações contextuais, pois esse protocolo provê informações de diferentes categorias, por exemplo, sistema e interface.
- Mesmo suprindo as necessidades momentâneas de aplicações integradas ao serviço de gerenciamento de informações de contexto, provendo informações contextuais referentes a um conjunto de categorias, uma série de melhorias podem ser realizadas no serviço. Por exemplo, suporte à manipulação de eventos. Aplicações cientes de contexto que utilizem o serviço poderiam ser informadas quando uma determinada informação contextual estiver disponível (por meio de *triggers*), serem informadas dos limites e preferências impostos pelo contexto (mediação contextual), ou ainda, receberem informação contextual periodicamente (via *polling*).
- Outros trabalhos futuros relevantes relacionados ao aperfeiçoamento do serviço são a implantação de mecanismos de segurança e o tratamento de erros. Qualquer aplicação ciente de contexto pode ser beneficiada pelas informações fornecidas pelo Web Service Gerente de Contexto. Portanto, faz-se necessário a implementação de um serviço de registro capaz de apontar quais aplicações podem acessar as informações providas pelo serviço.

- É importante efetuar uma análise de desempenho para averiguar o custo-benefício das tecnologias usadas para implementação e, caso necessário, buscar novas alternativas para a melhoria do serviço.
- Finalmente, há a necessidade de definição de relacionamentos entre representações de contexto. Por exemplo, a representação de contexto proposta por Goularte (2003) não especifica a relação entre um usuário e seus possíveis dispositivos heterogêneos (não há uma marcação que indique quais são os dispositivos de cada usuário), embora haja relação entre as informações de contexto referentes a uma entidade, por exemplo um usuário. Uma alternativa é o uso de RDF (Lassila & Swick, 1999) para representação dos relacionamentos entre as entidades referentes às representações relacionadas ou a recomendação W3C XLink, a qual define ligações em documentos XML (DeRose et al., 2001).

Representações de Contexto

A.1 Representação de Contexto das Características dos Dispositivos

```
<?xml version="1.0" encoding="UTF-8"?> <schema
targetNamespace="http://www.icmc.usp.br/~rudinei/Context/"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ctx="http://www.icmc.usp.br/~rudinei/Context/"
xmlns:ccpp="http://www.wapforum.org/profiles/UAPROF/ccppschem-20020709#"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="http://www.icmc.usp.br/~rudinei/Context/" schemaLocation="context.xsd"/>
  <import namespace="http://www.wapforum.org/profiles/UAPROF/ccppschem-20020709#"
  schemaLocation="RDF_uaprof_vocab.xsd"/>

  <complexType name="CCPPfeaturesType">
    <complexContent>
      <extension base="ctx:HowType">
        <sequence>
          <element ref="ccpp:component" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

A.2 Representação de Contexto de Rede

```
<complexType name="NetworkType">
  <sequence>
    <element name="NetworkElement" type="ctx:NetworkElementType"
    minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

```

</sequence>
<attribute name="Scale" type="ctx:ScaleType" default="LAN"/>
<attribute name="Topology" type="ctx:TopologyType" default="Bus"/>
<attribute name="Speed" type="ctx:SpeedType" default="100Mbps"/>
<attribute name="NetBandwidth" type="double"/>
<complexType name="NetBandwidthAverageType">
  <sequence>
    <element name="NetworkBandwidth" type="decimal" maxOccurs="unbounded"/>
  </sequence>
</complexType> -->
</complexType>
<!--Network Attributes===== -->
<simpleType name="ScaleType">
  <restriction base="string">
    <enumeration value="LAN"/>
    <enumeration value="MAN"/>
    <enumeration value="WAN"/>
    <enumeration value="WLAN"/>
  </restriction>
</simpleType>
<simpleType name="TopologyType">
  <restriction base="string">
    <enumeration value="Bus"/>
    <enumeration value="Ring"/>
    <enumeration value="Star"/>
  </restriction>
</simpleType>
<simpleType name="SpeedType">
  <restriction base="string">
    <enumeration value="10Mbps"/>
    <enumeration value="100Mbps"/>
    <enumeration value="1Gbps"/>
  </restriction>
</simpleType>
<!-- =====>
                Network Definitions
<!-- =====>
<complexType name="NetworkElementType" abstract="true">
  <complexContent>
    <extension base="ctx:DimensionsType"/>
  </complexContent>
</complexType>
<complexType name="NetPropertiesType">
  <complexContent>
    <extension base="ctx:IdentityType">
      <sequence>
        <element name="HostAddress" type="ctx:IPType"/>
        <element name="SubnetMask" type="ctx:IPType" minOccurs="0"/>
        <element name="Gateway" type="ctx:IPType" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="BandwidthAverageType">
  <complexContent>
    <extension base="ctx:HowType">
      <sequence>
        <element name="Throughput" type="ctx:ThroughputType" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>

```

```

</complexType>
<complexType name="ThroughputType">
  <complexContent>
    <extension base="ctx:HowType">
      <sequence>
        <element name="ifInOctets" type="positiveInteger"/>
        <element name="ifOutOctets" type="positiveInteger"/>
      </sequence>
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="InterfacesType">
  <complexContent>
    <extension base="ctx:IdentityType">
      <sequence>
        <element name="Interface" type="ctx:InterfaceType" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="InterfaceType">
  <sequence>
    <element name="ID" type="ctx:ifIDType" minOccurs="0"/>
    <element name="Description" type="string" minOccurs="0"/>
    <element name="MACAddress" type="ctx:MACAddressType" minOccurs="0"/>
    <element name="ifSpeed" type="positiveInteger"/>
  </sequence>
</complexType>
<simpleType name="ifIDType">
  <restriction base="integer">
    <pattern value="{0-50}"/>
  </restriction>
</simpleType>
<simpleType name="MACAddressType">
  <restriction base="string"/>
</simpleType>
<!-- ===== Network Entity ===== -->
<complexType name="HostType">
  <complexContent>
    <extension base="ctx:NetworkElementType"/>
  </complexContent>
</complexType>
<!-- ===== -->

```

Especificação WSDL do Web Service Gerente de Contexto

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions name="ContextManager"
targetNamespace="http://localhost:8080/cm/ContextManager"
xmlns:tns="http://localhost:8080/cm/ContextManager"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns2="http://java.sun.com/jax-rpc-ri/internal"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types>
    <schema targetNamespace="http://java.sun.com/jax-rpc-ri/internal"
xmlns:tns="http://java.sun.com/jax-rpc-ri/internal"
xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <simpleType name="text_xml">
        <restriction base="string" /></simpleType></schema></types>
  <message name="ContextManagerInterface_getDevice">
    <part name="String_1" type="xsd:string" />
    <part name="String_2" type="xsd:string" />
    <part name="String_3" type="xsd:string" /></message>
  <message name="ContextManagerInterface_getDeviceResponse">
    <part name="result" type="ns2:text_xml" /></message>
  <message name="ContextManagerInterface_getFullDevice">
    <part name="String_1" type="xsd:string" />
    <part name="String_2" type="xsd:string" /></message>
  <message name="ContextManagerInterface_getFullDeviceResponse">
    <part name="result" type="ns2:text_xml" /></message>
  <message name="ContextManagerInterface_getFullNetwork" />
  <message name="ContextManagerInterface_getFullNetworkResponse">
    <part name="result" type="ns2:text_xml" /></message>
  <message name="ContextManagerInterface_getFullUser">
    <part name="String_1" type="xsd:string" /></message>
```

```

<message name="ContextManagerInterface_getFullUserResponse">
  <part name="result" type="ns2:text_xml"/></message>
<message name="ContextManagerInterface_getNetwork">
  <part name="String_1" type="xsd:string"/></message>
<message name="ContextManagerInterface_getNetworkResponse">
  <part name="result" type="ns2:text_xml"/></message>
<message name="ContextManagerInterface_getPolicy">
  <part name="String_1" type="xsd:string"/></message>
<message name="ContextManagerInterface_getPolicyResponse">
  <part name="result" type="ns2:text_xml"/></message>
<message name="ContextManagerInterface_getUser">
  <part name="String_1" type="xsd:string"/>
  <part name="String_2" type="xsd:string"/></message>
<message name="ContextManagerInterface_getUserResponse">
  <part name="result" type="ns2:text_xml"/></message>
<portType name="ContextManagerInterface">
  <operation name="getDevice" parameterOrder="String_1 String_2 String_3">
    <input message="tns:ContextManagerInterface_getDevice"/>
    <output message="tns:ContextManagerInterface_getDeviceResponse"/></operation>
  <operation name="getFullDevice" parameterOrder="String_1 String_2">
    <input message="tns:ContextManagerInterface_getFullDevice"/>
    <output message="tns:ContextManagerInterface_getFullDeviceResponse"/></operation>
  <operation name="getFullNetwork" parameterOrder="">
    <input message="tns:ContextManagerInterface_getFullNetwork"/>
    <output message="tns:ContextManagerInterface_getFullNetworkResponse"/></operation>
  <operation name="getFullUser" parameterOrder="String_1">
    <input message="tns:ContextManagerInterface_getFullUser"/>
    <output message="tns:ContextManagerInterface_getFullUserResponse"/></operation>
  <operation name="getNetwork" parameterOrder="String_1">
    <input message="tns:ContextManagerInterface_getNetwork"/>
    <output message="tns:ContextManagerInterface_getNetworkResponse"/></operation>
  <operation name="getPolicy" parameterOrder="String_1">
    <input message="tns:ContextManagerInterface_getPolicy"/>
    <output message="tns:ContextManagerInterface_getPolicyResponse"/></operation>
  <operation name="getUser" parameterOrder="String_1 String_2">
    <input message="tns:ContextManagerInterface_getUser"/>
    <output message="tns:ContextManagerInterface_getUserResponse"/></operation></portType>
<binding name="ContextManagerInterfaceBinding" type="tns:ContextManagerInterface">
  <operation name="getDevice">
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
  <operation name="getFullDevice">
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
  <operation name="getFullNetwork">
    <input>
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

```

```

        use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
<operation name="getFullUser">
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
<operation name="getNetwork">
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
<operation name="getPolicy">
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
<operation name="getUser">
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </input>
    <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            use="encoded" namespace="http://localhost:8080/cm/ContextManager"/>
    </output>
    <soap:operation soapAction=""/></operation>
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/></binding>
<service name="ContextManager">
    <port name="ContextManagerIFPort" binding="tns:ContextManagerIFBinding">
        <soap:address location="REPLACE_WITH_ACTUAL_URL"/></port></service></definitions>

```


Referências Bibliográficas

- Abowd, G. (1999). Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal* 38(4), 508–530.
- Abowd, G., C. Atkeson, J. Hong, S. Long, R. Kooper, & M. Pinkerton (1997). Cyberguide: A Mobile Context-aware Tour Guide.
- Abowd, G., E. D. Mynatt, & T. Rodden (2002). The Human Experience. *IEEE Pervasive Computing* 1(1), 48–57.
- Abowd, G. D. & E. D. Mynatt (2000). Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7(1), 29–58.
- Arruda Jr, C. R. E. (2003). Context Kernel: Um Web Service Baseado nas Dimensões de Informação de Contexto. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação (ICMC-USP). 85p.
- Arruda Jr, C. R. E., R. F. Bulcão Neto, & M. G. C. Pimentel (2003). Open Context-aware Storage as a Web Service. In *Proceedings of the International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC) as part of the ACM/IFIP/USENIX International Middleware Conference*, pp. 81–87.
- Bray, T., J. Paoli, C. M. Sperberg McQueen, & E. Maler (2000). Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation. Disponível em: <<<http://www.w3.org/TR/REC-xml>>> (Acessado em 22 de Março de 2004).
- Brown, P. J., J. D. Bovey, & X. Chen (1997). Context-aware Applications: From the Laboratory to the Marketplace. *IEEE Personal Communications* 4(5), 58–64.
- Chalmers, D. (2002). *Contextual Mediation to Support Ubiquitous Computing*. Tese de Doutorado, Imperial College of Science, Technology and Medicine, Universidade de Londres.

- CISCO (2003). How To Calculate Bandwidth Utilization Using SNMP. Disponível em: <<http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a008009496e.shtml>> (22 de Março de 2004).
- Comer, D. E. (2000). *Internetworking With TCP/IP*. Prentice Hall PTR.
- Crowley, J. L., J. Coutaz, G. Rey, & P. Reignier (2002). Perceptual Components for Context-aware Computing. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, pp. 117-134.
- DeRose, S., E. Maler, & D. Orchard (2001). XML Linking Language (XLink) Version 1.0. Disponível em: <<<http://www.w3.org/TR/xlink>>> (Acessado em 16 de Maio de 2004).
- Dey, A., G. D. Abowd, & D. Salber (1999). A Context-based Infrastructure for Smart Environments. In *Proceedings of the International Workshop on Managing Interactions in Smart Environments*, pp. 114-128.
- Dey, A. K. (2000). *Providing Architectural Support for Building Context-Aware Applications*. Tese de Doutorado, Georgia Institute of Technology.
- Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing* 5(1), 4-7.
- Dey, A. K. & G. Abowd (2000). Towards a Better Understanding of Context and Context-awareness. In *Proceedings of the Workshop on the What, Who, Where, When, and How of Context-Awareness as part of the Conference on Human Factors in Computing Systems*.
- Dix, A., T. Rodden, N. Davies, J. Trevor, A. Friday, & K. Palfreyman (2000). Exploiting Space and Location as a Design Framework for Interactive Mobile Systems. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7(3), 285-321.
- Fallside, D. C. (2001). XML Schema Part 0: Primer, W3C Recommendation. Disponível em: <<<http://www.w3.org/TR/xmlschema-0/>>> (Acessado em 19 de Março de 2004).
- Faria, G. B. (2001). O Uso de MPEG-7 na Personalização da TV Interativa. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação (ICMC-USP). 101p.
- Faria, G. B., J. B. Santos Jr, R. Goularte, & E. S. Moreira (2001). Uso de Perfis em Aplicações de Televisão Interativa Conscientes de Contexto. In *Anais do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMídia)*. 139-154.

- Goularte, R. (2003). *Personalização e Adaptação de Conteúdo Baseadas em Contexto para TV Interativa*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC-USP).
- Goularte, R. & E. S. Moreira (2001). Managing Access to a Video Server Through a Metadata Base. *International Journal of Computers & Applications* 23(1), 25-29.
- Goularte, R. & E. S. Moreira (2002). Representação de Objetos de Mídia para Aplicações Conscientes de Contexto em TV Interativa. In *Anais do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMídia)*. 150-165.
- Goularte, R., E. S. Moreira, & M. G. C. Pimentel (2003). Structuring Interactive TV Documents. In *ACM Symposium On Document Engineering*. 42-51.
- Harrison, B. L., K. P. Fishkin, A. Gujar, C. Mochon, & R. Want (1998). Squeeze me, Hold me, Tilt me! An Exploration of Manipulative User Interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 17-24.
- Harter, A., A. Hopper, P. Steggles, A. Ward, & P. Webster (1999). The Anatomy of a Context-Aware Application. In *Mobile Computing and Networking*, pp. 59-68.
- Hong, J. I. & J. A. Landay (2001). An Infrastructure Approach to Context-aware Computing. *Human-Computer Interaction (HCI) Journal* 16, 2-3. 21p.
- Hull, R., P. Neaves, ., & J. Bedford-Roberts (1997). Towards Situated Computing. In *Proceedings of the International Symposium on Wearable Computers (ISWC)*, pp. 146-153.
- Ishii, H. & B. Ullmer (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 234-241.
- ISO/IEC (2003). The Mpeg Home Page. Disponível em: <<<http://www.chiariglione.org/mpeg/index.htm>>>. (Acessado em 21 de Março de 2004).
- Johnson, R. E. (1992). Documenting Frameworks Using Patterns. In *Proceedings of the Conference on Object-oriented programming systems, languages, and applications*, pp. 63-76.

- Kindberg, T., H. Morris, J. Schettino, B. Serra, M. Spasojevic, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, & V. Krishnan (2000). People, Places, Things: Web Presence for the Real World. *Mobile Networks and Applications* 7(5), 365–376.
- Klyne, G., F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, & L. Tran (2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation. Disponível em: <<<http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>>>. (Acessado em 18 de Março de 2004).
- Korteum, G., Z. Segsll, & M. Bauer (1998). Context-aware, Adaptive Wearable Computers as Remote Interfaces to “Intelligent” Environments. In *Wearable Computers*, pp. 58–65.
- Lassila, O. & R. R. Swick (1999). Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation. Disponível em: <<<http://www.w3.org/TR/REC-rdf-syntax/>>>>. (Acessado em 18 de Março de 2004).
- Long, S., R. Kooper, G. D. Abowd, & C. G. Atkeson (1996). Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. In *Mobile Computing and Networking*, pp. 97–107.
- Minneman, S., S. Harrison, B. Janssen, G. Kurtenbach, T. Moran, I. Smith, & B. van Melle (1995). A Confederation of Tools for Capturing and Accessing Collaborative Activity. In *Proceedings of the third ACM International Conference on Multimedia*, pp. 523–534. ACM Press.
- Mitra, N. (2003). Simple Object Access Protocol (SOAP) 1.2 Part 0: Primer, W3C Recommendation. Disponível em: <<<http://www.w3.org/TR/soap12-part0/>>>. (Acessado em 20 de Março de 2004).
- Pascoe, J. (1998). Adding Generic Contextual Capabilities to Wearable Computers. In *Proceedings of the International Symposium on Wearable Computers (ISWC)*, pp. 92–99.
- Ribeiro, R. J. M. (2002). Projeto de um Serviço de Redes Ativas para QoS. Dissertação (Mestrado), Instituto de Ciências Matemáticas e de Computação (ICMC-USP).
- Ryan, N., J. Pascoe, & D. Morse (1997). Enhanced Reality Fieldwork: The Context-aware Archaeologist Assistant. *Computer Applications & Quantitative Methods in Archaeology* 0, 269–274.

- Salber, D., A. Dey, & G. Abowd (1998). *Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm*: Tech.
- Salber, D., A. K. Dey, & G. D. Abowd (1999). The Context Toolkit: Aiding the Development of Context-enabled Applications. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 434–441.
- Santos Jr, J. B. (2002). *Modelagem de Ambientes Interativos Conscientes de Contexto: Uma Abordagem Baseada nos Padrões MPEG-4 e MPEG-7*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC-USP).
- Santos Jr, J. B., R. Goularte, E. S. Moreira, & G. B. Faria (2001). The Modeling of Structured Context-aware Interactive Environments. *Transactions of the SDPS - Journal of Integrated Design and Process Science* 5(4), 77–93.
- Schilit, B. & M. Theimer (1994). Disseminating Active Map Information to Mobile Hosts. *IEEE Network* 8(5), 22–32.
- Schilit, B. N. (1995). *A System Architecture for Context-Aware Mobile Computing*. Tese de Doutorado, Graduate School of Arts and Sciences, Columbia University.
- Schilit, B. N., N. Adams, & R. Want (1994). Context-aware Computing Applications. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 85–90.
- Schunter, M., P. Ashley, S. Hada, G. Karjoth, & C. Powers (2003). Enterprise Privacy Authorization Language (EPAL 1.2). Disponível em: <<<http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>>>. (Acessado em 07 de Maio de 2004).
- Smith, C. & M. H. Butler (2002). Validating CC/PP and UAProf Profiles. Disponível em: <<<http://www.hpl.hp.com/techreports/2002/HPL-2002-268.pdf>>> (21 de Março de 2004).
- Tanenbaum, A. S. (2003). *Computer Networks* (4 ed.). Prentice Hall PTR.
- Truong, K. N., G. D. Abowd, & J. A. Brotherton (2001). Who, What, When, Where, How: Design Issues of Capture & Access Applications. In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, pp. 209–224.
- W3C (2002). Web Services Activity. Disponível em: <<<http://www.w3.org/2002/ws>>> (Acessado em 16 de Maio de 2004).

Want, R., A. Hopper, V. Falco, & J. Gibbons (1992). The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)* 10(1), 91–102.

Ward, A., A. Jones, & A. Hopper (1997). A new location technique for the active office.

Weiser, M. (1993). Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM* 6(7), 75–84.