

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação
Programa de Pós-Graduação em Ciência da Computação

**Computação Ciente de Contexto Aplicada
ao Monitoramento de Condições Críticas em
Ambientes Físicos**

Taciana Novo Kudo

São Carlos – SP
Maio – 2004

**Computação Ciente de Contexto Aplicada
ao Monitoramento de Condições Críticas em
Ambientes Físicos**

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação
Programa de Pós-Graduação em Ciência da Computação

**Computação Ciente de Contexto Aplicada
ao Monitoramento de Condições Críticas em
Ambientes Físicos**

Taciana Novo Kudo

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Departamento de Computação, da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, área de concentração em Processamento de Imagens e Sinais - PIS.

São Carlos - SP
Maio - 2004

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

K95cc

Kudo, Taciana Novo.

Computação ciente de contexto aplicada ao monitoramento de condições críticas em ambientes físicos / Taciana Novo Kudo. -- São Carlos : UFSCar, 2005.
103 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2004.

1. Teoria da informação. 2. Computação ubíqua. 3. Computação ciente de contexto. 4. Segurança crítica. I. Título.

CDD: 003.54 (20^a)

*"As barreiras parecem grandes quando nos
sentimos pequenos.
Quando nos posicionamos adequadamente,
todas as portas se abrem".*

Zibia Gasparetto

Dedico este trabalho:

A Deus, Pai presente em todos os momentos
felizes e tristes da minha vida.

Aos meus pais Jogi Kudo e Norma Sueli Novo
Kudo, por tudo o que sempre fizeram e fazem
pela minha formação.

À minha irmã Priscila por cada palavra de
incentivo pronunciada nos momentos em que
mais precisei.

Ao meu querido esposo Renato por toda
paciência, estímulo e força que me
impulsionaram durante
todo o mestrado.

Agradecimentos

Primeiramente gostaria de agradecer a Deus por toda a força e sabedoria que me guiaram durante toda a minha vida e também durante todo o programa de mestrado.

À minha família que apesar da saudade sempre souberam me apoiar e nunca negaram uma palavra de consolo nos momentos de desânimo.

Ao meu esposo Renato que com muito amor acompanhou todos os obstáculos que surgiram durante o mestrado.

À minha orientadora Prof^a. Dr^a. Regina Borges de Araujo pela paciência, dedicação e amizade que nos uniram durante o desenvolvimento deste trabalho.

A todos os meus colegas do laboratório (Simone, Christmas, Matheus, Freire, Marcelo, Fernando, Goiano, Gislaine, Richard, Diego e demais alunos de iniciação científica) que me acolheram com carinho desde o início do mestrado tendo sempre uma palavra de ânimo.

A todos os meus colegas de mestrado das turmas de 2001, 2002 e 2003 (que são muitos, por isso é melhor não citar nomes) que comemoraram comigo cada conquista nas fases de disciplina, concessão de bolsa, qualificação e desenvolvimento do projeto.

À minha amiga Adriana Herden que apesar da distância sempre estamos em sintonia.

A todos os funcionários (principalmente a D.Ofélia, D.Vera, Cristina e Mirian), professores e coordenador do Departamento de Computação da UFSCar pelos incentivos constantes.

Ao CNPQ pela bolsa de mestrado concedida.

E mais uma vez a Deus, pois sem Ele nada teria sentido.

Resumo

A computação ciente de contexto estuda aplicações que adaptam seu comportamento com base em informações (ou contexto) capturadas de um ambiente físico e/ou computacional, como a localização de um usuário. O monitoramento de condições críticas em ambientes físicos é um dos domínios que exploram as facilidades da computação ciente de contexto, no qual os contextos são obtidos via sensores dispersos em ambientes para auxiliar a prevenção, o combate e a pós-avaliação de situações de emergência, como incêndios, rebeliões em prisões, explosões, vazamento de substâncias tóxicas, etc. Para que serviços do tipo interpretação de contexto e localização de pessoas ou objetos possam ser utilizados por uma grande variedade de dispositivos e por qualquer tipo de aplicação da classe de aplicações de monitoramento de condições críticas, verifica-se a necessidade de um *middleware* que forneça serviços independentes de plataforma, sistema operacional e/ou linguagem de programação.

Este trabalho apresenta uma abordagem para a aplicação de computação ciente de contexto no monitoramento de condições críticas em ambientes físicos, através da especificação de um *middleware* e de três de seus principais serviços, a saber: Serviço de Interpretação de Contextos, Serviço de Notificação de Eventos e Serviço de Localização. O Serviço de Interpretação de Contexto notifica e interpreta os contextos obtidos de sensores. Essas interpretações são baseadas nas políticas de segurança de cada empresa, e para que possa atender de maneira flexível a critérios específicos de monitoramento foi criada uma Interface de Configuração e Monitoramento. O Serviço de Notificação de Eventos utiliza o paradigma da publicação/subscrição baseada em tópicos. A localização de pessoas e/ou objetos é realizada através de um Serviço de Localização. A comunicação entre os serviços do *middleware* e as aplicações utiliza mensagens XML via mecanismo *publish/subscribe*. Um estudo de caso para utilização do *middleware* em questão é mostrado tendo como cenário o Museu de Aviação *Asas de um Sonho* da Companhia Aérea TAM.

Abstract

Context-aware computing investigates applications that adapt their behavior according to information (or context) captured from a physical and/or computational environment such as the location of a user. The monitoring of critical conditions in physical environments is one of the domains that take advantage of the context-aware computing facilities, where context is captured from sensors spread over environments in order to aid the prevention, the combat and the evaluation of emergency situations such as fire, rebellions in penitentiaries, explosions and leaks of toxic substances. Related works report the need for a *middleware* that provides services, e.g. for context interpretation and location of people and objects, which can be used by a myriad of devices and any kind of applications for monitoring of critical conditions, regardless platform, operating system and/or programming language.

This work presents an approach exploiting *context-aware computing in the monitoring of critical conditions in physical environments*, which includes the specification of a *middleware* and its following main services: the *Context Interpreter Service*, the *Event Notification Service* and the *Location Service*. The *Context Interpreter Service* notifies and interprets context gathered from sensors. As those interpretations follow the security policies of each business, an *Interface for Configuration and Monitoring* was also developed as a means to allow particular criteria of monitoring for a business. The *Event Notification Service* uses the *publish/subscribe* paradigm based on topics. The *Location Service* handles the location of people and objects in physical environments. *Middleware* components and applications communicate through *XML-encoded messages* via *publish/subscribe* mechanism. A case study for that approach uses the *Asas de um Sonho* aviation museum of the TAM air company as scenario.

Sumário

1. INTRODUÇÃO.....	1
1.1 CONSIDERAÇÕES INICIAIS	1
1.2 MOTIVAÇÃO	3
1.3 OBJETIVOS	5
1.4 ESTRUTURA DA DISSERTAÇÃO.....	6
2. COMPUTAÇÃO UBÍQUA.....	7
2.1 CONSIDERAÇÕES INICIAIS	7
2.2 HISTÓRICO	7
2.3 TEMAS DE PESQUISA DA COMPUTAÇÃO UBÍQUA	9
2.3.1 INTERFACES NATURAIS	9
2.3.2 CAPTURA E ACESSO AUTOMATIZADOS.....	10
2.3.3 CIÊNCIA DE CONTEXTO	10
2.4 DESAFIOS DA COMPUTAÇÃO UBÍQUA	11
2.5 APLICAÇÕES DE COMPUTAÇÃO UBÍQUA.....	13
2.5.1 ENSINO	16
2.5.2 REUNIÃO	17
2.5.3 GUIA TURÍSTICO	18
2.5.4 RESIDÊNCIA	20
2.5.5 SEGURANÇA EM AMBIENTES FÍSICOS	22
2.6 MONITORAMENTO DE CONDIÇÕES CRÍTICAS EM AMBIENTES FÍSICOS	23
2.7 CONSIDERAÇÕES FINAIS.....	25
3. COMPUTAÇÃO CIENTE DE CONTEXTO	27
3.1 CONSIDERAÇÕES INICIAIS	27
3.2 DEFINIÇÕES DE CONTEXTO	28
3.3 DIMENSÕES DE CONTEXTO.....	29
3.4 TIPOS DE CONTEXTO	31
3.5 REQUISITOS DE SOFTWARE DE COMPUTAÇÃO CIENTE DE CONTEXTO.....	32
3.5.1 ESPECIFICAÇÃO DE INFORMAÇÃO DE CONTEXTO	32
3.5.2 SEPARAÇÃO DE AQUISIÇÃO E UTILIZAÇÃO DE INFORMAÇÃO DE CONTEXTO	32
3.5.3 INTERPRETAÇÃO DE INFORMAÇÃO DE CONTEXTO.....	33
3.5.4 COMUNICAÇÃO DISTRIBUÍDA E TRANSPARENTE.....	33
3.5.5 DISPONIBILIZAÇÃO CONTÍNUA DE COMPONENTES DE CAPTURA DE INFORMAÇÃO DE CONTEXTO	34
3.5.6 ARMAZENAMENTO DE INFORMAÇÃO DE CONTEXTO	34
3.5.7 DESCOBERTA DE RECURSOS	35

3.6 TRABALHOS RELACIONADOS.....	35
3.6.1 CONTEXT TOOLKIT	35
3.6.2 CONTEXT FABRIC.....	38
3.6.3 GAIA	40
3.6.4 IROS	42
3.7 CONSIDERAÇÕES FINAIS.....	45

4. COMPUTAÇÃO CIENTE DE CONTEXTO APLICADA AO MONITORAMENTO DE CONDIÇÕES CRÍTICAS EM AMBIENTES FÍSICOS 47

4.1 CONSIDERAÇÕES INICIAIS	47
4.2 PROJETO DE UM SISTEMA DE CAPTURA E ACESSO DE CONDIÇÕES CRÍTICAS EM AMBIENTES FÍSICOS.....	49
4.3 PROPOSTA DE UM MIDDLEWARE PARA APLICAÇÕES DE MONITORAMENTO DE CONDIÇÕES CRÍTICAS.....	51
4.3.1 DESCRIÇÃO DO <i>MIDDLEWARE</i>	51
4.3.2 A COMUNICAÇÃO ENTRE OS MÓDULOS DO MIDDLEWARE	53
4.3.3 O SERVIÇO DE NOTIFICAÇÃO DE EVENTOS.....	57
4.3.4 O SERVIÇO DE INTERPRETAÇÃO DE CONTEXTO	59
4.3.5 O SERVIÇO DE LOCALIZAÇÃO.....	61
4.4 INFORMAÇÕES DE INTERESSE DE APLICAÇÕES DE MONITORAMENTO DE CONDIÇÕES CRÍTICAS.....	64
4.4.1 A DEFINIÇÃO DA ENTIDADE TÓPICO NUM MODELO DE PUBLICAÇÃO/SUBSCRIÇÃO BASEADO EM TÓPICO	64
4.4.2 DEMONSTRANDO INTERESSE EM TÓPICOS	66
4.4.3 NOTIFICAÇÃO DE EVENTO PARA UM TÓPICO	67
4.5 INTERFACES DE CONFIGURAÇÃO E MONITORAMENTO DE CONDIÇÕES CRÍTICAS.....	67
4.6 CONSIDERAÇÕES FINAIS.....	68

5. CONCLUSÕES..... 69

5.1 CONTRIBUIÇÕES GERADAS.....	70
5.2 LIMITAÇÕES E TRABALHOS FUTUROS	71
5.3 CONCLUSÕES.....	72

REFERÊNCIAS BIBLIOGRÁFICAS 74

APÊNDICE A	83
APÊNDICE B.....	88

Lista de Figuras

Figura 2. 1 - Exemplos de dispositivos Active Badges (a) e um mapa (b) indicando a localização e dados das pessoas que carregam esses dispositivos consigo. Adaptado de [WAN, 1992].	13
Figura 2. 2 - Dispositivo PARCTab. Adaptado de [SCH, 1993].	15
Figura 2. 3 - Sala de aula com lousas eletrônicas, projetores e suporte multimídia (a) e um hiperdocumento Web (b) gerado automaticamente a partir do material apresentado em sala. Adaptado de [ABO, 1999].	16
Figura 2. 4 - Reunião na sala iRoom com vários dispositivos controlados por software (ex: grandes lousas eletrônicas). Adaptado de [PON, 2003].	18
Figura 2. 5 - Dispositivo móvel com dados sobre a localização do usuário (a) e interface web com o mapa do campus (b). Adaptado de [BUR, 2002].	19
Figura 2. 6 - Vista externa da casa (a) e aplicação que monitora a atividade diária de pessoas idosas que vivem sozinhas (b). Adaptado de [MYN, 2001].	21
Figura 2. 7 - Arquitetura de sistema de computação ubíqua para manutenção de aviões. Adaptado de [LAM, 2004].	22
Figura 2. 8 - Arquitetura do Siren utiliza a API InfoSpace para unificar as facilidades de armazenamento, comunicação e feedback adaptativo em regras de contexto. Adaptado de [JIA, 2004a].	24
Figura 3. 1 - Context Toolkit – adaptada de [DEY, 2000].	37
Figura 3. 2 - Arquitetura do Context Fabric – adaptada de [HON, 2001b].	39
Figura 3. 3 - Exemplo de um caminho complexo que tenta calcular a probabilidade de estar tendo uma reunião – adaptada de [HON, 2001b].	40
Figura 3. 4 - Arquitetura da Infra-estrutura GAIA – adaptada de [ROM, 2002].	41
Figura 3. 5 - Arquitetura da Infra-estrutura iROS – adaptada de [JOH, 2002].	43
Figura 4. 1 - Visão Geral do Sistema	50
Figura 4. 2 - Aplicação, Middleware e Ambiente Físico (rede de sensores)	53
Figura 4. 3 - Interação entre Aplicação, Serviços do Middleware e Rede de Sensores.	55
Figura 4. 4 - Interação entre Aplicação, Serviços de Evento, Interpretação e Localização e Rede de Sensores	58
Figura 4. 5 - Exemplo de Esquema XML para subscrição de Monitoramento de condição crítica.	59
Figura 4. 6 - Esquema XML de subscrição para Localização	62

Figura 4. 7 - Exemplo de como os arquivos XML são utilizados e como o arquivo publish.xml pode ser usado por qualquer aplicação independente de dispositivo e/ou mídia	63
Figura 4. 8 - Publish/Subscribe baseado em Tópicos – adaptado de [EUG, 2000].....	65
Figura 4. 9 - Exemplo de Publish/Subscribe baseado em Tópicos para um tópico de temperatura (a), e exemplo de Publish/Subscribe baseado em Tópicos para um tópico de temperatura e um sub-tópico de temperatura>60 (b).....	66
Figura 4. 10 - XML para subscrição no serviço de eventos do middleware	67
Figura A. 1 - Interface Inicial	83
Figura A. 2 - Parte 1 – Manutenção de Entidades e Sensores.....	84
Figura A. 3 - Interface de Inserção de Entidades.....	84
Figura A. 4 - Interface de Inserção de Sensores	84
Figura A. 5 - Parte 2 – Entidades e Sensores	84
Figura A. 6 - Parte 3 – Situações e Tipos de Sensores	85
Figura A. 7 - Parte 4 – Entidades e Situações a serem monitoradas.....	85
Figura A. 8 - Parte 5 – Entidades que contém outras entidades	86
Figura A. 9 - Monitoramento Periódico.....	86
Figura A. 10 - Monitoramento Orientado a Eventos	87
Figura A. 11 - Monitoramento Baseado em Consulta	87
Figura B. 1 - Interface Inicial	88
Figura B. 2 - Interface de Manutenção de Entidades e Sensores	89
Figura B. 3 - Interface de Inserção de Entidades.....	89
Figura B. 4 - Interface de Inserção de Sensores	89
Figura B. 5 - Interface de Entidades e Sensores.....	91
Figura B. 6 - Interface de definição de situações a serem monitoradas e tipos de sensores....	92
Figura B. 7 - Interface de definição de entidades e situações a serem monitoradas.....	93
Figura B. 8 - Interface de definição de entidades que contém outras entidades	94
Figura B. 9 - Interface de Monitoramento Periódico.....	95
Figura B. 10 - Interface de Monitoramento Orientado a Eventos	96
Figura B. 11 - Interface de Monitoramento Baseado em Consulta	97
Figura B. 12 - Formato de uma subscrição ao Serviço de Eventos	98
Figura B. 13 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 1 .	98
Figura B. 14 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 2 .	98
Figura B. 15 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 3 .	98
Figura B. 16 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 4 .	99

Figura B. 17 - Exemplo de XML para subscrição orientada a eventos – explosão no setor1..	99
Figura B. 18 - Exemplo de XML para subscrição orientada a eventos – curto-circuito no setor3	99
Figura B. 19 - Formato de uma subscrição ao Serviço de Eventos	100
Figura B. 20 - Exemplo de XML para subscrição periódica – temperatura do setor 1 a cada 10 minutos.....	100
Figura B. 21 - Exemplo de XML para subscrição orientada a eventos – temperatura do setor1, quando temperatura>60.....	101
Figura B. 22 - Exemplo de XML para subscrição baseada em consulta – Qual a temperatura do setor1?	101
Figura B. 23 - Esquema XML de subscrição para Localização	102
Figura B. 24 - Esquema XML de subscrição para Localização - Onde está pessoa X?	102
Figura B. 25 - Esquema XML de subscrição para Localização - Quem está no ambiente X?	102
Figura B. 26 - Esquema XML de subscrição para Localização - Tem alguém no ambiente Y?	103

Lista de Tabelas

Tabela 2. 1 - Eras da Computação. Adaptado de [WEI, 1996]	8
Tabela 3. 1 - Tabela comparativa de formas de interpretação de contexto	44
Tabela B. 1 - Entidades e Tipos de Sensores do Museu	89
Tabela B. 2 - Entidades - Tipo de sensor - N.Serial	91
Tabela B. 3 - Tabela de entidades - situações e sensores.....	93
Tabela B. 4 - Tabela de entidades que contém outras entidades	94

1. Introdução

1.1 Considerações Iniciais

O surgimento de novas áreas de pesquisa tem sido incentivado por inúmeros avanços tecnológicos, tais como a miniaturização eletrônica, o aumento contínuo do poder de processamento, a evolução das tecnologias de sensores e de dispositivos móveis, a melhoria nas tecnologias de comunicação sem fio e o desenvolvimento de arquiteturas de software flexíveis [LYY, 2002].

Para Frank Stajano [STA, 2002] a Computação Ubíqua engloba a visão de um mundo no qual o custo do poder computacional e das comunicações digitais tornam-se tão baratos a ponto de poderem ser embutidos em todos os objetos que nos cercam no dia-a-dia. O pesquisador Mark Weiser [WEI, 1991] define a Computação Ubíqua como uma dessas emergentes áreas de pesquisa que estuda os aspectos sócio-tecnológicos provenientes da integração de tecnologia a ambientes físicos de interação no sentido de fornecer informações e serviços a usuários de forma contínua e transparente.

Segundo Abowd e Mynatt [ABO, 2000] a computação ubíqua envolve três temas principais de pesquisa: interfaces naturais, captura e acesso automatizados e ciência de

contexto. Interfaces naturais exploram formas comuns de expressão humana (ex: fala e escrita) e utilizam ações implícitas dos usuários para facilitar a interação humano-computador, como superfícies eletrônicas públicas (ex: lousa) e privadas (ex: *tablet*) com sistemas de reconhecimento de escrita. Aplicações de captura e acesso automatizados envolvem o registro automático de informações de experiências ao vivo e a disponibilização dessas informações para acesso posterior, como diálogos de uma reunião e *slides* de uma palestra [ABO, 2000]. A computação ciente de contexto, por sua vez, estuda aplicações que adaptam seu comportamento segundo informação (ou contexto) percebida do ambiente físico e/ou computacional de interação, como a localização do usuário, os objetos próximos a este e as mudanças ocorridas ao longo do tempo [SCH, 1994]. Este trabalho tem como foco a computação ciente de contexto, em especial os contextos obtidos por sensores dispersos em um ambiente físico de interação.

A computação ciente de contexto tem sido utilizada em vários domínios de aplicação, como sala de aula [ABO, 1999], sala de reuniões [JOH, 2002], guias turísticos [ABO, 1997], residências [BRU, 2000] e na área de segurança como um todo, por exemplo: segurança industrial [POO, 2002] e segurança na aviação [LAM, 2004]. O monitoramento de condições críticas em ambientes físicos de interação [MIC, 2003] [JIA, 2004a] pode ser considerado um domínio específico de aplicações de computação ciente de contexto, pois devido ao desenvolvimento de sensores mais poderosos e mais baratos, as aplicações de monitoramento de granularidade fina começaram a tornar-se viáveis e importantes na ajuda à prevenção, combate e avaliação de situações de emergência. No primeiro caso (prevenção), informações podem ser capturadas fazendo com que condições críticas possam ser evitadas, já no segundo caso (combate), podem ajudar com informações mais detalhadas para tomadas de decisão no gerenciamento de equipes de resgate e salvamento, e no terceiro caso (avaliação), auxiliar

perícias por parte da seguradora ou até mesmo do próprio responsável pela segurança da empresa.

1.2 Motivação

O monitoramento de condições críticas de ambientes físicos apresenta algumas características próprias: (a) necessidade de captura precisa de dados de nós sensores; (b) necessidade de monitoramento constante e consistente do ambiente; e (c) necessidade de notificação imediata das mudanças que ocorrem no ambiente, mesmo na presença de falhas de nós sensores. A área de computação ubíqua oferece algumas facilidades que podem auxiliar o monitoramento de condições críticas e atender os seus requisitos.

Atualmente, existe uma grande ênfase no desenvolvimento de serviços especializados em ambientes distribuídos para computação ciente de contexto. Nesse sentido, destacam-se os *middlewares*. Um *middleware* corresponde a uma camada de software intermediária entre o sistema operacional e as aplicações que permite abstrair detalhes de baixo nível quanto ao acesso a serviços distribuídos, como protocolos de comunicação, gerenciamento de dados e balanceamento de carga.

É fato que existem diversos projetos voltados ao desenvolvimento de *middlewares* para computação ubíqua [ROM, 2002], porém a maioria deles são voltados para os domínios de aplicações de sala de aula e sala de reuniões, não se preocupando diretamente com as necessidades apresentadas no monitoramento de condições críticas em ambientes físicos. Para que essas necessidades sejam devidamente atendidas, o desenvolvimento de um *middleware* com serviços específicos para monitoramento de condições críticas faz-se necessário.

Uma outra motivação para este trabalho surgiu a partir da avaliação de alguns sistemas existentes usados por equipes de combate ao fogo, para monitoramento de situações de emergência. Estes sistemas apresentam limitações, principalmente em termos de funcionalidade e interface de acesso, que poderiam ser superadas com as novas tecnologias de redes de sensores e de software ciente de contexto. Em um cenário de combate a incêndios, um *middleware* para ciência de contexto pode ser utilizado para fornecer dados a uma aplicação no sentido de auxiliar o monitoramento de um ambiente em situações de emergência, e caso ocorra alguma situação de emergência (como incêndio, inundação, curto-circuito, etc) os bombeiros poderão ser auxiliados no momento do combate.

O ambiente físico de interação (ex: prédio, casa, *shopping center* ou fábrica) precisa conter sensores especializados, tais como aqueles para medição de: temperatura, pressão, umidade, presença de substâncias tóxicas, presença de fumaça, e outros sensores que possam capturar alterações nas condições físicas do ambiente de modo a monitorá-lo e auxiliar no combate em uma situação de emergência.

Durante a fase de monitoramento, informações como temperatura, pressão, presença de fumaça e presença de chama são capturadas e interpretadas para verificar se indicam qualquer grau de probabilidade de iniciar alguma situação de emergência.

Se um incêndio vier a acontecer, no momento do combate ao incêndio, por exemplo, dados do ambiente, e a localização de pessoas poderão ser capturados para verificar se alguém corre perigo de morte. Todas essas informações combinadas poderão servir para planejar estratégias de combate e salvamento. As informações do ambiente podem ser visualizadas de diferentes tipos de dispositivos e por múltiplos usuários, incluindo: (a) os bombeiros para traçar estratégias de combate e salvamento, (b) o engenheiro de segurança da empresa para

verificar se existe alguma prioridade de salvamento, (c) a seguradora da empresa para verificar a ocorrência dos fatos, bem como as suas causas, etc., em situações de perícia.

Além de auxiliar no combate ao incêndio em tempo real, todas as informações capturadas do mundo real, podem ser armazenadas em um repositório e serem acessadas posteriormente. Por exemplo, permitir assistir a um combate a incêndio que ocorreu em um determinado dia e local para efeito de treinamento, ou para avaliação de desempenho individual e/ou coletivo da equipe.

1.3 Objetivos

Um dos objetivos deste trabalho é a especificação de um *middleware* para monitoramento de condições críticas em ambientes físicos que são munidos de um grande número de sensores que capturam informações de contexto do ambiente, e a especificação de três serviços: Serviço de Interpretação de Contextos, Serviço de Notificação de Eventos e Serviço de Localização. No serviço de interpretação de contexto, as interpretações são baseadas nas políticas de segurança de cada empresa. Para isso, o responsável pela segurança da empresa pode configurar o ambiente, estabelecendo, por exemplo, o que proteger, contra quais situações proteger, etc., para fins de monitoramento. Para que o serviço de interpretação de contexto possa atender de maneira flexível as necessidades de monitoramento dos ambientes, foi criada uma **Interface de Configuração e Monitoramento** pela qual o responsável pode definir critérios de monitoramento específicos para sua empresa. O *middleware* como um todo está sendo desenvolvido por alunos de mestrado e de iniciação científica do LRVNet – Laboratório de Realidade Virtual em Rede do Departamento de Computação da Universidade Federal de São Carlos.

Assim, os objetivos deste trabalho resumem-se a:

1. Especificação de um *middleware* para aplicações de monitoramento de condições críticas de ambientes físicos de interação e os seguintes serviços: Serviço de Interpretação de Contexto, Serviço de Notificação de Eventos e Serviço de Localização;
2. Especificação de interfaces XML para comunicação entre os módulos do *middleware* e/ou aplicações;
3. Especificação e implementação de uma interface simples de configuração e monitoramento de condições críticas pela qual o responsável de segurança define os critérios de monitoramento.

1.4 Estrutura da Dissertação

Este trabalho está organizado da seguinte maneira: o **Capítulo 2** descreve a computação ubíqua, alguns de seus domínios de aplicação com ênfase no monitoramento de condições críticas em ambientes físicos, e os principais temas de pesquisa da área de computação ubíqua: interfaces naturais, captura e acesso automatizados, bem como computação ciente de contexto. O **Capítulo 3** discorre sobre a Computação Ciente de Contexto, onde este trabalho está engajado, e sobre os diversos projetos voltados ao desenvolvimento de *middlewares* de computação ubíqua. O **Capítulo 4** apresenta a aplicação da Computação Ciente de no Monitoramento de Condições Críticas em Ambientes Físicos. O **Capítulo 5** apresenta as conclusões deste trabalho e propostas de extensões futuras. Por fim apresentam-se as Referências Bibliográficas, o Apêndice A (telas da Interface de Configuração e Monitoramento) e o Apêndice B (Estudo de Caso).

2. Computação Ubíqua

2.1 Considerações Iniciais

Segundo o Dicionário Aurélio¹, ubiqüidade é a propriedade do ser “que está ao mesmo tempo em toda a parte”. Um dos principais objetivos da computação ubíqua é tornar os serviços computacionais tão intrínsecos a um determinado ambiente a ponto de torná-lo transparente para seus usuários.

Este capítulo tem como objetivo dar uma visão geral sobre a área da computação ubíqua posicionando-a no histórico da computação, apresentando seus três principais temas de pesquisa e descrevendo os principais domínios de aplicação explorados com enfoque especial no emprego de computação ubíqua no monitoramento de condições críticas em ambientes físicos de interação.

2.2 Histórico

A primeira referência de computação ubíqua data de 1991 quando o pesquisador Mark Weiser [WEI, 1991] descreve sua visão de um mundo em que a tecnologia torna-se intimamente

¹ Ferreira, A.B. H. “Dicionário Aurélio Eletrônico Século XXI”, Versão 3.0, Lexicon Informática Ltda, 1999.

ligada ao ambiente físico, fornecendo informações e serviços a usuários quando e onde fossem necessários, auxiliando-os na realização de suas atividades diárias de maneira transparente. Weiser afirma que a computação ubíqua ajudará a organizar e mediar interações sociais onde quer que seja e sempre que as situações puderem acontecer.

Weiser e Brown [WEI, 1996] sugerem uma divisão da história da computação na era do *mainframe* e na era dos computadores pessoais (PCs). A era do *mainframe* (até 1970) traz a idéia de um único recurso computacional compartilhado por muitas pessoas. Já a era do computador pessoal (a partir de 1980) é caracterizada pela relação de um computador para cada pessoa. Em 1990, os avanços da Internet e da computação distribuída caracterizam uma fase de transição para uma nova era. Essa nova era, chamada de Computação Ubíqua, caracteriza-se pela disponibilidade de vários recursos computacionais para uma única pessoa em um ambiente onde esses recursos encontram-se embarcados, e auxiliam os usuários em suas tarefas de maneira transparente. A **Tabela 1** ilustra as eras da computação segundo a visão de Weiser e Brown [WEI, 1996].

Tabela 2. 1 - Eras da Computação. Adaptado de [WEI, 1996]

	Nº Computadores	Nº Pessoas
Era dos <i>mainframes</i>	1	N
Era dos PCs	1	1
Internet e Computação Distribuída	Fase de transição	
Era da Computação Ubíqua	N	1

Segundo Lyytinen e Yoo [LYY, 2002], a computação ubíqua vem sendo acelerada pelos recentes avanços nas tecnologias de comunicações sem fio e de baterias, aumentos contínuos de poder de processamento e aparecimento de arquiteturas de software flexíveis.

2.3 Temas de Pesquisa da Computação Ubíqua

Abowd e Mynatt [ABO, 2000] afirmam que a computação ubíqua envolve três temas principais de pesquisa: interfaces naturais, captura e acesso automatizados e ciência de contexto. Cada um desses temas é apresentado a seguir.

2.3.1 Interfaces Naturais

A computação ubíqua inspira o desenvolvimento de aplicações que suportam interação física entre humanos e computadores cada vez mais distante do paradigma *teclado-mouse-tela* e cada vez mais perto da maneira natural de como os humanos interagem com o mundo físico. Os humanos falam, gesticulam e usam técnicas de escrita para se comunicarem. Essas ações naturais devem ser usadas como contribuição explícita ou implícita em sistemas de computação ubíqua.

Interfaces que apóiam formas humanas mais naturais de comunicação (como escrita, fala e gestos) estão começando a complementar ou substituir elementos do paradigma de interfaces gráficas tradicionais. Essas interfaces facilitam o aprendizado e a usabilidade dos sistemas sem modificar a estrutura das tarefas que são executadas. Como exemplos de pesquisas em interfaces naturais podem-se citar as técnicas de interação baseada em voz de Witbrock e Hauptmann [WIT, 1998] e em escrita de Schilit et al. [SCH, 1998].

2.3.2 Captura e Acesso Automatizados

Grande parte do tempo dos seres humanos é gasta em registrar eventos dos quais participam para posteriormente lembrarem de informações importantes desses eventos. Devido à dificuldade dos seres humanos de registrar todas as informações de seu interesse, faz-se uso de recursos como anotações e gravações de áudio e/ou vídeo para posterior acesso. Com a difusão dos recursos multimídia nos ambientes computacionais, o ser humano passou a utilizar o computador como forma de capturar e armazenar informações relevantes por meio de documentos tradicionais, de hiperdocumentos ou de mídias de áudio e vídeo.

A utilização de recursos computacionais para suportar captura e acesso de informações faz com que o ser humano possa focar sua atenção nas atividades que desempenham com mais eficiência. Uma vez estabelecido o aumento de potencial de utilização desses recursos para o registro de informações, aplicações surgem para tornar mais eficiente o processo de captura e acesso de experiências ao vivo. Essas aplicações permitem o registro simultâneo de diversos fluxos (ex: áudio, vídeo e escrita), o que possibilita uma série de facilidades de acesso ao conteúdo capturado.

2.3.3 Ciência de Contexto

A computação ciente de contexto estuda aplicações que adaptam seu comportamento segundo informação (ou contexto) percebida do ambiente físico e/ou computacional de interação. É considerado contexto qualquer informação relevante que possa ser utilizada para caracterizar entidades de uma interação usuário-computador, como localização e identificação do usuário, sua proximidade de outros usuários e a tarefa que este desempenha ao longo do tempo [SCH, 1994] [DEY 2001].

Pelo fato desse trabalho focar especialmente a computação ciente de contexto no que se refere à interpretação de contextos, o capítulo seguinte apresenta a computação ciente de contexto de maneira mais detalhada.

2.4 Desafios da Computação Ubíqua

A computação ubíqua é uma área de pesquisa intrinsecamente interdisciplinar, colhendo contribuições de diversas áreas, como Interação Humano-Computador, Engenharia de Software, Sistemas Distribuídos, Redes de Computadores, Inteligência Artificial e outros. Em virtude de suas próprias características, conforme apresentado na seção anterior, a computação ubíqua impõe certos desafios que essas referidas áreas têm buscado solucionar.

Quanto ao desenvolvimento de interfaces humano-computador, é necessário manipular dados provenientes de interfaces naturais como tipos de dados de primeira classe no desenvolvimento de sistemas interativos [ABO, 2000]. Configura-se como desafio o desenvolvimento de ferramentas de construção de interfaces multimodais, ou seja, ferramentas com suporte operacional não apenas para dados textuais e primitivas originadas de interações via mouse e teclado, mas também com operações para a manipulação de linguagem natural e outras formas de entrada de dados, como vídeo, escrita, gestos e informações oriundas de sensores.

O tratamento de erros na entrada de dados não é um problema recente, mas o uso de interfaces naturais acentua erros já existentes e gera novos tipos de erros que estão ligados, principalmente, ao reconhecimento de linguagem natural e escrita manual. Pesquisas para tratamento de erros em interfaces baseadas em reconhecimento têm buscado melhorar a tecnologia de reconhecimento visando eliminar ou reduzir erros, descobrir um erro antes do

sistema ou do usuário tratar um erro e, por fim, desenvolver uma biblioteca de técnicas de correção de erros para entrada de dados baseada em reconhecimento [ABO, 2000].

A multiplicidade de dispositivos de captura e acesso móveis ou não e a heterogeneidade de perfis de usuários em ambientes de computação ubíqua fazem surgir desafios quanto à definição de padrões para o fornecimento e a adaptação de serviços e de informação. É necessário que sistemas de computação ubíqua tenham mecanismos de descrição das características físicas e funcionais desses dispositivos, bem como dos perfis e preferências de seus usuários para que um serviço ou informação seja adaptado em conformidade [BAN, 2002].

Para construir sistemas de computação ubíqua em geral, tem-se discutido bastante na literatura a definição de um conjunto de requisitos ou características que sirvam de orientação durante o ciclo de vida desses sistemas [DEY, 2000]. Comunicação distribuída e transparente, tolerância a falhas e mecanismos de descoberta de recursos são alguns desses requisitos. Desenvolvedores de sistemas de computação ubíqua têm também identificado a necessidade de dividir responsabilidades entre aplicações e infra-estruturas de software especializadas quanto ao armazenamento, consulta, processamento e recuperação de informações de um ambiente de interação [KIN, 2002]. Essa divisão de responsabilidades facilita, dentre outras coisas, a modularização e a reutilização de componentes de software para o desenvolvimento de novos sistemas.

Em um ambiente de interação instrumentado com câmeras, microfones e sensores dois aspectos têm adquirido bastante relevância em pesquisas de computação ubíqua: a segurança e a privacidade das informações capturadas por esses dispositivos [JIA, 2002]. Há discussões sobre o armazenamento ou não dessas informações (ex: diálogos gravados), por quanto tempo essas informações devem ser mantidas, quem tem direitos para acessá-las,

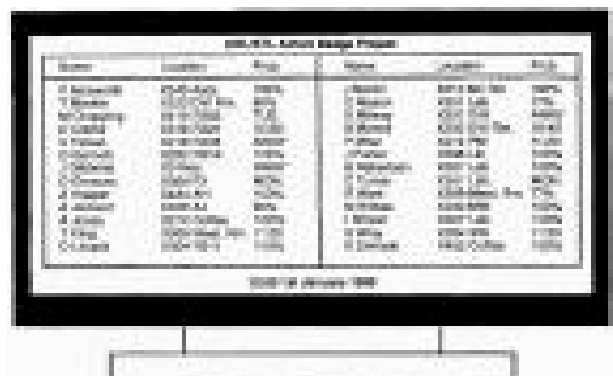
quando estas devem ser capturadas, o que pode ser disponibilizado para acesso público e a possibilidade ou não de identificar as pessoas às quais essas informações estão associadas. Sensores podem identificar onde pessoas estão, o que estão fazendo e quando chegam e saem de casa. Ao serem combinadas, essas informações expõem a privacidade dessas pessoas.

2.5 Aplicações de Computação Ubíqua

Pesquisas em computação ubíqua têm sido aplicadas em vários domínios de aplicação. As primeiras demonstrações de como a computação ubíqua pode auxiliar usuários em suas tarefas diárias foram aplicadas nos laboratórios da *Olivetti Research Labs*, por meio do dispositivo *Active Badge* (Figura 2.1a) [WAN, 1992], e nos laboratórios da *XeroxPARC*, por meio do dispositivo *PARCTab* (Figura 2.2) [SCH, 1993]. Ambos os sistemas eram sistemas cientes de localização de usuários, possibilitando a implementação de facilidades como transferências automáticas de ligações telefônicas e atualizações automáticas de mapas com a informação de localização de usuários em um prédio (Figura 2.1b).



(a)



(b)

Figura 2. 1 - Exemplos de dispositivos Active Badges (a) e um mapa (b) indicando a localização e dados das pessoas que carregam esses dispositivos consigo. Adaptado de [WAN, 1992].

O *Active Badge System* é um sistema para localizar e identificar pessoas e objetos dentro da área onde é instalado. O sistema apresenta como componentes principais:

- *active badge*: dispositivo que transmite sinais infravermelhos, cada um associado a uma pessoa ou objeto a localizar, que permite ao portador mover-se dentro da área controlada pelo sistema;
- sensores: dispositivos receptores de raios infravermelhos associados a uma localização, normalmente espalhados por setores de um recinto fechado, como salas e corredores de um prédio.

Cada indivíduo com seu *active badge* transmite um sinal de raios infravermelhos, que consiste em um código de identificação único desse dispositivo que é detectado a cada dez segundos pelos sensores. Essa informação é enviada pela rede e associada com a informação contida em uma base de dados para posterior processamento.

Este sistema pode ser também utilizado para controle de acesso a áreas de segurança, controle de dispositivos de forma automática (ex: redirecionamento de chamadas telefônicas), auxiliar em emergências médicas e controle de visitantes.

O *PARCTab*, por sua vez, é um protótipo de pesquisa desenvolvido pela Xerox Parc com o objetivo de explorar o potencial e o impacto de computadores móveis em um ambiente de escritório. O sistema *PARCTab* consiste de um dispositivo portátil do tamanho de um Palm com potencial de comunicação sem fio, através de transceptores infravermelho, com aplicações em estações de trabalho.

O projeto do sistema *PARCTab* foi guiado pelos seguintes princípios [SCH, 1995]:

- *Alta portabilidade* – o dispositivo foi projetado para ser transportado e usado o tempo todo. Seu tamanho, peso e capacidades de hardware são adequados para promover computação casual, atendendo necessidades instantâneas de serviço,

como a impressão de documento na impressora mais próxima do local onde está o usuário do dispositivo, por exemplo;

- *Conectividade constante* – o sistema assume que o dispositivo está sempre conectado à infra-estrutura de rede;
- *Relatório de localização* – o software do sistema sempre sabe onde está cada dispositivo *PARCTab*.

O *PARCTab* tem um conjunto de fontes para textos, funções que enviam imagens para sua tela, geram sons e permitem interação com canetas sobre sua tela (**Figura 2.2**).



Figura 2. 2 - Dispositivo PARCTab. Adaptado de [SCH, 1993].

Há três tipos de componentes de software no sistema: *gateways*, agentes e aplicações. Os *gateways* enviam e recebem pacotes de dados via sinais infravermelhos. Cada dispositivo é representado por um agente responsável por rastrear sua localização. As aplicações são adaptadas para acomodarem-se à baixa largura de banda de comunicação e à pequena tela.

Um grande número de aplicações *PARCTab* foi construído para explorar os problemas no acesso da informação ubíqua, o uso de interfaces baseadas em canetas e o controle do ambiente, conforme apresentado a seguir:

- *Controle do ambiente* – um dispositivo pode ser usado como controle remoto, por exemplo, para controlar as luzes e temperatura de um ambiente;

- *Acesso à informação* – previsão do tempo (obtida da Web), temperatura local e velocidade do vento;
- Dicionários, calendários e controle de migração, dentre outras.

A próxima seção apresenta domínios de aplicação tradicionais da computação ubíqua, como ensino, reuniões, guias turísticos, residências e segurança de ambientes.

2.5.1 Ensino

Basicamente, aplicações de computação ubíqua que dão suporte ao ensino têm dois objetivos: (a) reduzir a sobrecarga que alunos têm ao anotar informações passadas pelo professor em sala de aula, (b) auxiliar professores na tarefa de produção de material didático.

Como exemplos dessas aplicações, tem-se o projeto *eClass* [ABO, 1999], formalmente conhecido por *Classroom 2000*. Esse projeto teve início em 1995 no Instituto de Tecnologia da Geórgia nos EUA e compreende um conjunto de tecnologias de hardware e software para a captura de aulas presenciais (**Figura 2.3a**) e posterior disponibilização do material capturado sob a forma de hiperdocumentos multimídia customizados (**Figura 2.3b**).

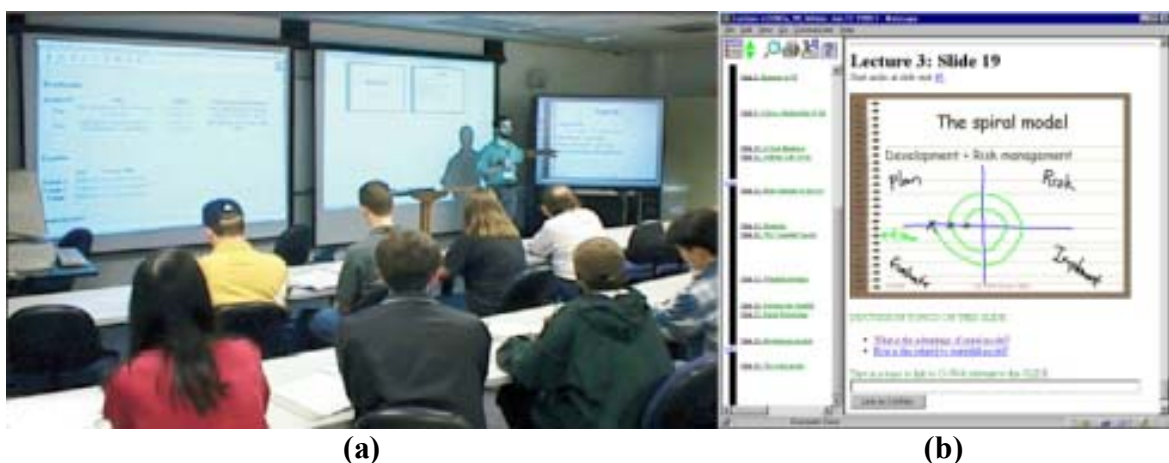


Figura 2. 3 - Sala de aula com lousas eletrônicas, projetores e suporte multimídia (a) e um hiperdocumento Web (b) gerado automaticamente a partir do material apresentado em sala. Adaptado de [ABO, 1999].

O *Lecture Browser* [MUK, 1999] é um projeto da Universidade de Cornell nos EUA que difere do *eClass* ao utilizar técnicas de visão computacional e combinação de mais de uma fonte de vídeo para produzir seus hiperdocumentos multimídia resultantes da aula capturada.

O *Authoring on the Fly* [HÜR, 1999] é um projeto de suporte ao ensino da Universidade de Freiburg na Alemanha que difere de outros sistemas dessa natureza em função do poderoso modelo de sincronização das mídias capturadas da aula, das diversas formas de representação do material capturado e das funcionalidades fornecidas para acesso posterior a esse material.

2.5.2 Reunião

Basicamente, aplicações de computação ubíqua que dão suporte a reuniões formais ou informais têm dois objetivos: (a) registrar o andamento de um projeto e (b) identificar as contribuições individuais de cada membro do grupo.

O sistema *DUMMBO* (*Dynamic, Ubiquitous, Mobile Meeting Board*) [BRO, 1998] é um projeto do Instituto de Tecnologia da Geórgia nos EUA voltado para a captura de atividades de uma reunião informal. Este sistema registra os desenhos feitos em uma lousa eletrônica compartilhada e a voz de cada membro da reunião, além de disponibilizar essa informação sob a forma de hiperdocumentos multimídia sincronizados.

O sistema *TeamSpace* [RIC, 2001] é um projeto colaborativo entre o Instituto de Tecnologia da Geórgia, IBM e Boeing com foco na captura de atividades envolvidas em uma reunião formal local ou distribuída. Esse sistema registra uma parcela maior dos artefatos apresentados durante uma reunião, incluindo slides, anotações, itens de agenda, vídeo e áudio de cada membro da reunião.

O projeto *iRoom* (*Interactive Room*) [JOH, 2002] da Universidade de Stanford nos EUA tem foco nas interações com dispositivos computacionais (ex: computadores pessoais, laptops, PDAs e superfícies eletrônicas) em um espaço de trabalho conectado a uma rede sem fio (**Figura 2.4**). Um de seus principais desafios envolve o suporte à movimentação de dados e de controle entre esses vários dispositivos de interação de forma transparente, de forma a não interromper o processo de colaboração.



Figura 2. 4 - Reunião na sala iRoom com vários dispositivos controlados por software (ex: grandes lousas eletrônicas). Adaptado de [PON, 2003].

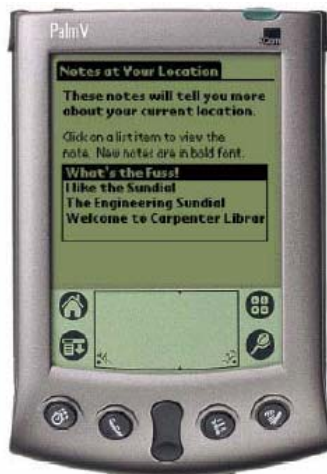
2.5.3 Guia Turístico

Guias turísticos eletrônicos trabalham normalmente com informações de localização dos usuários e dispositivos portáteis para conferir maior mobilidade aos usuários. Os principais objetivos dessa classe de aplicações são: (a) registrar os locais visitados e (b) identificar a posição atual do usuário dentro de um espaço de interação (ex: campus universitário, museu).

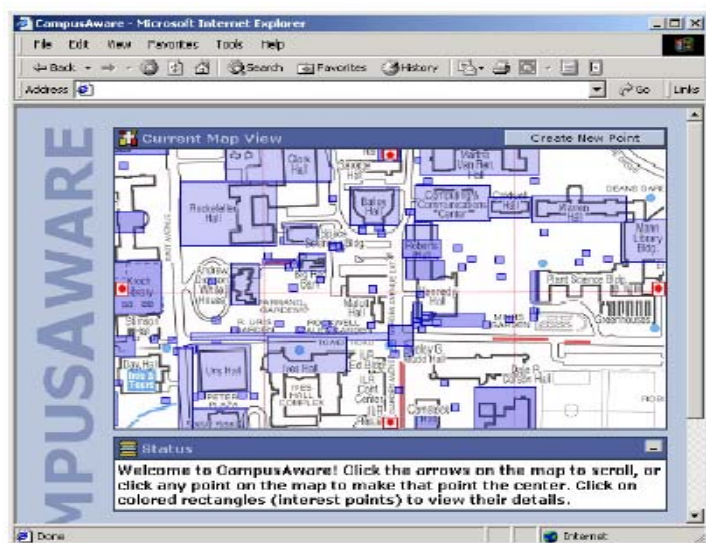
O sistema *CyberGuide* [ABO, 1997] é um guia turístico eletrônico ciente de localização que fornece serviços a usuários, como a posição atual de um usuário, o conjunto

de locais visitados e que direção esse usuário está tomando. As informações de localização baseiam-se no sistema de posicionamento global (GPS – *Global Positioning System*).

O *CampusAware* [BUR, 2002] também é um sistema de turismo que utiliza informações de localização para fornecer serviços aos usuários. O sistema permite que o usuário faça anotações em seus dispositivos portáteis sobre os locais que está visitando (Figura 2.5a), fornecendo também informações de como encontrar determinados locais de interesse e a posição atual do usuário (Figura 2.5b). O *CampusAware* possui um repositório central com três bancos de dados para informações contextuais e sociais, como as anotações dos usuários e locais visitados. Assim como o sistema *CyberGuide*, o *CampusAware* utiliza o sistema GPS para obtenção de informações de localização em ambientes abertos.



(a)



(b)

Figura 2.5 - Dispositivo móvel com dados sobre a localização do usuário (a) e interface web com o mapa do campus (b). Adaptado de [BUR, 2002].

O sistema *Rememberer* [FLE, 2002] é parte do projeto *Cooltown* [KIN, 2001] dirigido pela *Hewlett-Packard Laboratories*. Seu objetivo é capturar experiências pessoais e estimular discussões ou outras formas de interação pessoal através de dispositivos portáteis

sem fio. Para capturar informação de localização em recintos fechados, esse sistema utiliza identificação por rádio-frequência ou RFID (*Radio Frequency Identification*) [AIM, 2003]. Um exemplo de aplicação desse sistema é em visitas a museus, onde câmeras fotográficas registram interações dos usuários no museu e essas informações são disponibilizadas na ordem em que os locais foram visitados, permitindo ainda que o usuário faça anotações sobre esses lugares.

2.5.4 Residência

Basicamente, aplicações de computação ubíqua que abrangem o domínio doméstico têm por objetivos: (a) conhecer as atividades dos moradores de uma casa e (b) fornecer serviços que aumentem a qualidade de vida deles.

Aware Home [KID, 1999] é um projeto desenvolvido pelo Instituto de Tecnologia da Geórgia nos EUA (**Figura 2.6a**). Suas metas são: (a) investigar que tipos de serviços podem ser fornecidos quando um sistema é ciente das atividades dos moradores da casa (**Figura 2.6b**), (b) construir modelos de comportamento humanos para auxiliar os computadores nas tomadas de decisão e (c) apoiar pessoas com mais idade a morar sozinhas, ajudando-as a manter uma dieta saudável, lembrando-as de tomar remédios nos horários exigidos, ou mesmo mantendo seus familiares informados sobre o bem-estar delas. A **Figura 2.6b** mostra uma aplicação que monitora o cotidiano de pessoas idosas ou doentes que moram sozinhas. Metáforas aplicadas a interfaces usuário-computador representam o grau de atividade dessas pessoas, sem invadir sua privacidade com informações de áudio e/ou vídeo.



Figura 2. 6 - Vista externa da casa (a) e aplicação que monitora a atividade diária de pessoas idosas que vivem sozinhas (b). Adaptado de [MYN, 2001].

EasyLiving [BRU, 2000] é um projeto da *Microsoft Research* que se preocupa com o desenvolvimento de arquiteturas e tecnologias para ambientes inteligentes, focando particularmente em uma sala de estar residencial. O ambiente contém um computador, telas eletrônicas, caixas de som, sofás, mesa de café, entre outros itens. Serviços são fornecidos para melhorar o ambiente, como a automatização do controle de luz, tocar música conforme a localização e a preferência do usuário, e ainda transferir automaticamente o conteúdo de uma tela para outra.

Adaptive House [MOZ, 1998] é um projeto da Universidade de Colorado e tem como objetivo desenvolver uma casa que se programe observando o estilo de vida e os desejos de seus habitantes e aprendendo a se antecipar às necessidades dos mesmos. O sistema desenvolvido no *Adaptive House* é chamado *ACHE* (*Adaptive Control of Home Environments*), e foi desenvolvido basicamente para controlar o sistema de AVAC (Aquecedor, Ventilador, Ar-condicionado), iluminação e água.

A próxima seção apresenta algumas aplicações de monitoramento de condições críticas no ambiente físico. Estas serão tratadas com mais detalhes por constituírem o foco do presente trabalho.

2.5.5 Segurança em Ambientes Físicos

Basicamente, aplicações de computação ubíqua voltadas para segurança de ambientes físicos têm como objetivos: (a) proteção de patrimônio contra roubos e intrusões, (b) prevenção de acidentes e (c) detecção de falhas. Exemplos de aplicações são: pilotagem e manutenção de aeronaves, combate a incêndios, monitoramento de manobras militares e segurança industrial.

Reliable Wireless Networks for Industrial Systems [POO, 2002] é um projeto da *Ember Corporation* que utiliza redes de sensores sem fio em uma fábrica de tratamento de água. A idéia é conectar sensores nas tubulações de água enviando dados a uma sala de controle da fábrica para que possam identificar e prevenir vazamentos. Questões como confiabilidade, redundância e comunicação distribuída oferecidas pela rede de sensores sem fio são altamente desejáveis nesse tipo de situação.

Na área de aviação, Lampe et al. [LAM, 2004] propõem a utilização de tecnologias de computação ubíqua para melhorar o processo de manutenção de aeronaves. Qualidade, segurança, documentação e os altos custos gerados quando aeronaves estão fora de operação demandam uma execução eficiente do processo de manutenção das aeronaves.

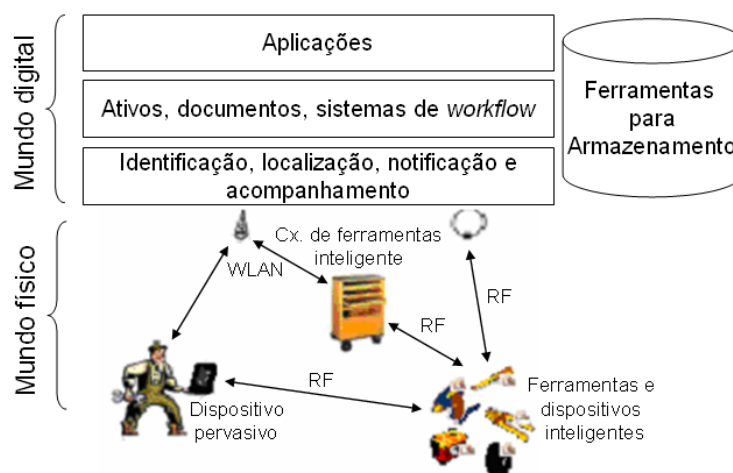


Figura 2. 7 - Arquitetura de sistema de computação ubíqua para manutenção de aviões. Adaptado de [LAM, 2004].

A **Figura 2.7** apresenta a arquitetura de um sistema de computação ubíqua para manutenção de aviões, composto de duas partes: o mundo digital com suporte de armazenamento, aplicações, serviços e dados, e o mundo físico com ferramentas munidas de identificação por rádio frequência. Todas essas ferramentas se comunicariam com uma espécie de caixa de ferramentas inteligente por rede sem fio descrevendo sua localização, status de operação, dentre outras informações.

O presente trabalho focaliza o domínio de aplicações de monitoramento de condições críticas em ambientes físicos de interação, apresentado com maior nível de detalhes na próxima seção.

2.6 Monitoramento de Condições Críticas em Ambientes Físicos

Existem algumas aplicações de computação ubíqua que têm por objetivo monitorar as condições do ambiente e auxiliar nas decisões que precisam ser tomadas de maneira rápida e eficaz quando situações anormais são detectadas. Exemplos de projetos dessa natureza incluem o *Siren* [JIA, 2004a] e o Posto de Comando do Futuro (*The Command Post of the Future*) [DAR, 2004].

Siren é um projeto desenvolvido pelas Universidades da Califórnia, Stanford e Washington nos EUA, voltado especificamente para o suporte às situações de incêndio que fornece comunicação entre bombeiros e mensagens de alerta [JIA, 2004a]. As mensagens são codificadas utilizando XML e enviadas sobre HTTP. *Siren* trata apenas os seguintes contextos: (a) localização relativa – em que localização relativa significa que a informação é do tipo "Sala tal"; (b) temperatura do ambiente; e (c) nível de oxigênio restante.

São gerados apenas cinco tipos de mensagens de alerta: “lugar perigoso”, “perigo para si mesmo”, “próximo a um lugar perigoso”, “outros em perigo” e “instruções”. Essas informações são apresentadas no PDA de cada bombeiro com o objetivo de manter uma comunicação entre os bombeiros e apresentar informações sobre o próprio bombeiro, e se algum outro bombeiro estiver próximo, passar informações sobre o ambiente. Outra característica é que *Siren* não se preocupa com a persistência dos dados, apenas com a utilização desses em tempo-real.

A **Figura 2.8** apresenta a arquitetura do *Siren*: um gerente de armazenamento implementa a abstração de espaço de informação; um gerente de comunicação manipula passagens de mensagens entre dispositivos em uma rede *peer-to-peer*; e uma máquina de regras de contexto que controla o *feedback* do usuário ciente de situação. Todos esses componentes e as aplicações interagem por meio da *API InfoSpace*, uma interface de programação baseado em espaços de tuplas. Aplicações que utilizam os componentes do *Siren* coletam informações de sensores carregados por bombeiros ou embutidos pelo ambiente.

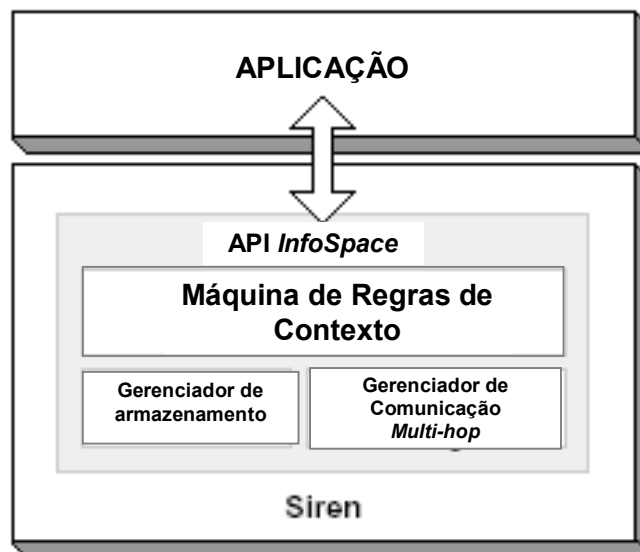


Figura 2. 8 - Arquitetura do Siren utiliza a API InfoSpace para unificar as facilidades de armazenamento, comunicação e feedback adaptativo em regras de contexto. Adaptado de [JIA, 2004a].

O Posto de Comando do Futuro (*The Command Post of the Future*) [DAR, 2004] consiste de um conjunto de projetos de investigação em situações de campo de batalha. Esses projetos focalizam o desenvolvimento de tecnologias avançadas que possibilitem a criação de *displays* interativos do ambiente, integrados com múltiplos sensores para o comandante aperfeiçoar suas decisões de forma rápida e com qualidade (o que pode ser chamado de ciência situacional). Esses *displays* podem auxiliar em tomadas de decisões facilitando a interação multimodal, a visualização de informações do ambiente e a formação de estratégias baseadas em conhecimento.

O domínio de aplicação de monitoramento de condições críticas diferencia-se dos demais domínios (ex: sala de aula e sala de reuniões) no que se refere à consistência, precisão e entrega dos dados do ambiente sendo monitorado, já que no monitoramento de condições críticas vidas humanas podem estar correndo perigo.

2.7 Considerações Finais

Uma tendência das pesquisas em computação ubíqua é a união de esforços de aplicações de captura e acesso automatizados com aplicações de computação ciente de contexto, como é o caso do sistema *Rememberer* [FLE, 2002].

Um dos objetivos deste capítulo foi dar uma visão geral das áreas de aplicações da computação ubíqua, referenciando alguns esforços em cada área e apresentando em particular o uso da computação ubíqua no monitoramento de condições críticas em ambientes físicos que é o domínio enfatizado de maneira especial neste trabalho.

O projeto *Siren* [JIA, 2004a] tem como foco aplicações de combate a incêndios com o objetivo de auxiliar os bombeiros no momento do combate. Porém, não se preocupa com a persistência dos dados para acesso posterior (ex: para perícia e treinamento). O projeto utiliza a comunicação por PDAs, no qual a visualização pode ser um pouco inviável quando a situação é de emergência, em que as ações têm que ser tomadas de maneira rápida e o ambiente pode estar sujeito a presença de fumaça densa e altas temperaturas. Já o projeto “Posto de Comando do Futuro” [DAR, 2004] tem como objetivo o desenvolvimento de tecnologias avançadas que auxiliem em tomadas de decisões facilitando a interação multimodal, a visualização de informações do ambiente e a formação de estratégias baseadas em conhecimento.

O próximo capítulo descreve de maneira mais detalhada a computação ciente de contexto, foco principal deste trabalho.

3. *Computação Ciente de Contexto*

3.1 *Considerações Iniciais*

Os seres humanos são capazes de trocar idéias e refletir sobre novas idéias de maneira apropriada devido a alguns fatores, como: o vocabulário que compartilham, o entendimento de como as coisas funcionam e um entendimento implícito das situações do cotidiano [DEY, 2000]. Quando as pessoas se comunicam, elas são capazes de utilizar informações de situação (ou *contexto*) para acrescentar algo à conversação. Infelizmente, esta capacidade não é encontrada na comunicação entre humanos e computadores. Computadores não entendem a linguagem humana, não entendem como as coisas funcionam e não podem obter de maneira independente informações sobre situações, pelo menos não tão facilmente como os seres humanos.

Na computação tradicional, geralmente, os usuários fornecem informações aos computadores por meio de teclado e mouse utilizando, portanto uma maneira explícita de entrada de dados. Entretanto, conforme visto no Capítulo 2, a computação ubíqua prioriza o desenvolvimento de aplicações cujas interfaces humano-computador exploram as formas naturais de interação humana (ex: fala, gestos e escrita). Para que os usuários não precisem

fornecer informações de maneira explícita ao computador, as aplicações precisam ser capazes de capturar contextos por meio de sensores físicos e lógicos e fornecer serviços e informações sem que o usuário precise solicitá-los.

Este capítulo tem como objetivo discorrer sobre a Computação Ciente de Contexto, definindo contexto, apresentando requisitos de softwares cientes de contexto e descrevendo trabalhos relacionados que tratam a aquisição e a interpretação de informações de contexto.

3.2 Definições de Contexto

Na literatura encontram-se muitas definições para contexto. Schilit e Theimer [SCH, 1994] definem contexto como localização, identidade de pessoas e objetos próximos e mudanças nesses objetos. Em uma definição similar, Brown et al. [BRO, 1997] definem contexto como localização e identidade de pessoas próximas do usuário, adicionando as informações de hora, período e temperatura do ambiente. Ryan et al. [RYA, 1998] definem contexto como localização e identidade do usuário e de objetos, informações físicas do ambiente e tempo.

Dentre as definições existentes, a mais abrangente é a dada em [Dey, 2001], onde *“contexto é qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade, onde uma entidade é uma pessoa, um lugar, ou um objeto considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação”*.

Por ser a mais abrangente e referenciada na literatura, essa definição será utilizada no decorrer deste trabalho quando forem usadas as palavras *contexto* ou *informação de contexto*.

3.3 Dimensões de Contexto

Como visto no Capítulo 2, sistemas pioneiros em ciência de contexto utilizavam informações de localização e identificação de pessoas, via sinais infravermelhos, para prover serviços úteis a usuários, como controle de luzes e temperatura de um dado ambiente e transferência automática de chamadas telefônicas.

Para obter informações de localização, aplicações mais recentes usam as facilidades providas pelo sistema GPS (*Global Positioning System*) para prover navegação de carros em cidades e estradas. O sistema GPS também tem sido utilizado em sistemas de guias turísticos inteligentes capazes de escolher um vídeo ou áudio para ser exibido de acordo com informações de localização dadas através do GPS [ABO, 1997].

Porém, há muitas outras informações contextuais além de localização e identificação de pessoas e objetos. A maioria dos sistemas ciente de contexto ainda não incorpora noções de tempo, história, outras pessoas além do usuário, bem como outras informações disponíveis nos ambientes. Abowd & Mynatt sugerem cinco dimensões para especificação e modelagem de informações de contexto [ABO, 2000]:

- **Who** (identificação): seres humanos realizam suas atividades e recordam de fatos passados com base na presença das pessoas que os vivenciaram. Sendo assim, o sistema deve prover informações de contexto de todas as pessoas envolvidas em uma determinada atividade;

- **Where** (localização): assim como as informações de identificação, informações de contexto de localização têm sido muito utilizadas em sistemas cientes de contexto. As pesquisas mostram um interesse particular na utilização dessa informação contextual associada a outras dimensões, por exemplo, a dimensão temporal para explorar a mobilidade de usuários ao longo do tempo;
- **When** (tempo): informações temporais têm sido usadas para indexar registros capturados ou para informar por quanto tempo um usuário esteve em uma determinada localização. Porém, o conhecimento dessas informações permite fazer inferências ajudando a interpretar as atividades humanas e estabelecer padrões de comportamento. Por exemplo, se uma pessoa idosa passa muito tempo em um banheiro, de forma incomum segundo seu histórico de atividades, há fortes indícios de ela tenha sofrido uma queda e precise de ajuda;
- **What** (atividade): o objetivo é obter informação, normalmente via sensores, que possibilite interpretar o que o usuário está fazendo. Em sistemas onde várias atividades podem ser desenvolvidas, identificar o que um usuário está fazendo em um determinado momento pode ser uma tarefa complexa;
- **Why** (intenção): tão difícil quanto perceber o que o usuário está fazendo, é entender o porquê de sua ação. Interpretar informações de contexto que possam caracterizar o estado de uma pessoa talvez seja o maior desafio da computação ciente de contexto.

Normalmente, as informações de contexto que exprimem a atividade e a intenção do usuário são obtidas por combinação de outras informações de contexto (ex: localização +

identificação + tempo), ou via sensores como aqueles para medição de temperatura corporal, entonação vocal, batimento cardíaco e pressão arterial.

3.4 Tipos de Contexto

Segundo Henricksen et al. [HEN, 2002], contexto pode ser classificado como:

- Estático: contexto estático corresponde ao grupo de informações que permanecem fixas durante o tempo de vida da entidade, como números de CPF e RG de uma pessoa. Todas as outras classes de contexto são dinâmicas.
- Dinâmico: informações de contexto dinâmicas podem ser de três tipos – sentido (do inglês *sensed*), explícito e interpretado.

As informações de contexto dinâmicas sentidas são aquelas capturadas por meio de sensores físicos e lógicos, como localização de uma pessoa e nível de umidade de um ambiente. Informações de contexto dinâmicas explícitas são aquelas fornecidas explicitamente pelo usuário, como sua senha de acesso ou itens de sua agenda particular. A classe de informações dinâmicas interpretadas refere-se aos contextos obtidos de um ou mais contextos por meio de uma regra de interpretação que varia de uma simples condição até algoritmos de Inteligência Artificial mais complexos. Temperatura anormal ou relação de proximidade de entidades são exemplos de contexto dinâmico interpretado.

3.5 Requisitos de Software de Computação Ciente de Contexto

Dey sugere sete requisitos que um software de computação ciente de contexto deve possuir para facilitar sua utilização [DEY, 2000], conforme descrito nas seções seguintes.

3.5.1 Especificação de Informação de Contexto

Provavelmente um dos requisitos mais importantes de um software ciente de contexto é a existência de mecanismos que permitam que uma aplicação especifique quais contextos ela tem interesse (ex: monitoramento da localização do usuário) e que ações devem ser tomadas quando determinado contexto for obtido. Por exemplo, ao monitorar a localização de um usuário, uma aplicação pode transferir chamadas telefônicas para o telefone celular do mesmo quando este não estiver em seu local de trabalho.

3.5.2 Separação de Aquisição e Utilização de Informação de Contexto

Não existe uma maneira padrão de se adquirir e manipular informações de contexto. De maneira geral, desenvolvedores optam pelas técnicas mais fáceis de implementação, abdicando da generalização e do reuso. Para a aquisição de informações de contexto, infra-estruturas de software e aplicações geralmente utilizam gerenciamento de eventos (por ex: *publish/subscribe*) ou mecanismos de consulta.

Uma vez que as informações de contexto foram adquiridas, a aplicação deve determinar, entre outras coisas, a ocorrência e a relevância de modificações nessas informações. Por isso é importante separar a aquisição e a utilização de informações de contexto. Além disso, as aplicações podem utilizar as informações de contexto sem se preocupar com os detalhes de como elas foram adquiridas.

3.5.3 Interpretação de Informação de Contexto

Existe a necessidade de estender os mecanismos de notificação e de consulta de sistemas cientes de contexto para permitir que estes recuperem contextos de ambientes distribuídos. Informações de contexto podem passar por muitas camadas de software antes de serem entregues à aplicação devido à necessidade de abstração (interpretação) de mais alto nível. Por exemplo, uma aplicação de captura de informações de reunião precisa ser notificada quando uma reunião estiver para acontecer: em um nível mais baixo, informações de localização e identificação podem ser interpretadas para determinar onde várias pessoas se encontram. Em um nível mais alto, esta informação pode ser combinada com registros da agenda dessas pessoas para determinar se a reunião está acontecendo.

Para o desenvolvedor da aplicação, o uso de várias camadas deve ser transparente. Para que isso aconteça, o contexto deve ser interpretado antes de ser usado pela aplicação. Para que a interpretação seja facilmente reutilizada por várias aplicações, ela precisa ser fornecida por um *framework* [DEY, 2000]. Caso contrário, as aplicações deverão re-implementar o mecanismo de interpretação que lhes é necessário.

3.5.4 Comunicação Distribuída e Transparente

Sensores utilizados para capturar informações de contexto não estão, em sua maioria, alocados no mesmo computador no qual as aplicações cientes de contexto estão sendo executadas. Geralmente, os sensores encontram-se fisicamente distribuídos no ambiente físico. Devido a isso, a comunicação distribuída deve ser transparente tanto para os sensores

quanto para as aplicações no sentido de que desenvolvedores de aplicações não precisem especificar e implementar um protocolo de comunicação e um esquema de codificação e decodificação para a transmissão de informações de contexto.

3.5.5 Disponibilização Contínua de Componentes de Captura de Informação de Contexto

Devido ao fato de uma aplicação poder solicitar uma informação de contexto a qualquer momento, é necessário que os componentes que capturam contexto sejam executados de forma independente das aplicações, e também que estejam sempre disponíveis. Portanto os componentes devem ser executados continuamente para permitir que aplicações os consultem sempre que necessário. Em uma conferência, por exemplo, componentes de captura de áudio podem ser requisitados continuamente para o registro de fluxos das inúmeras atividades executadas em paralelo (ex: palestras e tutorais).

3.5.6 Armazenamento de Informação de Contexto

Devido à necessidade da disponibilidade constante dos dados, faz-se necessário manter históricos de informações de contexto. Um histórico de contexto pode ser usado para estabelecer tendências e prever futuros valores de contexto. Sem o armazenamento persistente de contextos, esse tipo de análise não poderia ser realizado. Componentes de captura devem adquirir informações de contexto mesmo quando nenhuma aplicação está interessada nas informações de contexto disponíveis naquele momento, pois essas informações de contexto capturadas podem ser utilizadas por uma aplicação que necessite do histórico de um dado de contexto, por exemplo, o histórico de localização de um usuário para prever sua localização futura.

3.5.7 Descoberta de Recursos

Para que uma aplicação possa se comunicar com dispositivos de captura de contexto (ex: sensores), esta deve saber que tipos de informações o dispositivo pode fornecer, qual a sua localização e qual o modo de comunicação suportado. Para que a aplicação não precise se preocupar com esses detalhes, a infra-estrutura de software subjacente precisa fornecer suporte à descoberta de recursos. Assim que uma aplicação é iniciada, ela deve especificar o tipo de informação de contexto de seu interesse. Com isso, o mecanismo de descoberta de recursos se responsabiliza por encontrar componentes adequados e fornecer mecanismos de acesso.

3.6 Trabalhos Relacionados

3.6.1 Context Toolkit

O **Context Toolkit** [SAL, 1999] é um *toolkit* desenvolvido pelo Instituto de Tecnologia da Geórgia que se preocupa com dados de contexto capturados por meio de sensores em ambientes de computação ubíqua. Seu objetivo principal é auxiliar o desenvolvimento de soluções para tratar as dificuldades originadas da natureza da informação de contexto e, então, facilitar a construção de aplicações cientes de contexto.

O Context Toolkit possui componentes básicos de comunicação conhecidos como *BaseObject*, *context widgets*, *context aggregators*, *context interpreters*, *context services* (referentes aos *widgets*, agregadores, interpretadores e serviços, respectivamente) e objetos de descoberta de recursos, conhecido como descobridores (*discoverers*).

A classe *BaseObject* fornece uma infra-estrutura básica para a comunicação dos componentes distribuídos no Context Toolkit, facilitando a comunicação *peer-to-peer* usada pelos componentes do *toolkit*. A classe *BaseObject* contém métodos para comunicação com *widgets*, agregadores, interpretadores, serviços e descobridores.

Quando uma aplicação é iniciada, ela se comunica com o descobridor para localizar os componentes que são relevantes para suas funcionalidades. Aplicações e agregadores precisam se inscrever aos *widgets* para receber contextos de seus interesses. As inscrições são feitas utilizando o método `subscribeTo` fornecido pelo *BaseObject* que envia uma mensagem `addSubscriber` para um *widget*. No método `subscribeTo` são definidos os dados que as aplicações ou agregadores precisam receber e em quais condições. Isto ajuda a reduzir a comunicação na rede, o que é importante em uma arquitetura distribuída por razões de desempenho.

Os *widgets* capturam contexto dos sensores e os torna disponíveis tanto para os agregadores, quanto para as aplicações. As aplicações não adquirem contexto somente de *widgets*, pois podem se inscrever também nos agregadores. *Widgets*, agregadores e aplicações ainda podem solicitar que um contexto seja transformado utilizando os interpretadores. A **Figura 3.1** mostra como aplicações e componentes no Context Toolkit podem interagir.

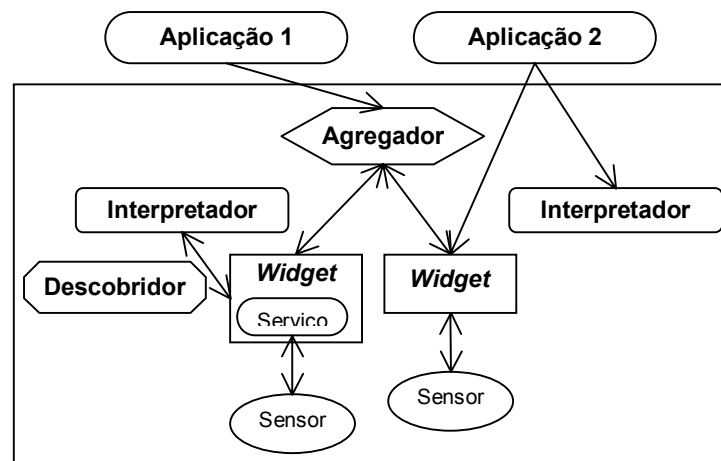


Figura 3. 1 - Context Toolkit – adaptada de [DEY, 2000].

Uma limitação do Context Toolkit é que o processamento de contexto não é feito de forma automática, pois o desenvolvedor deve especificar manualmente qual caminho os dados de contexto deverão seguir, ou seja, o desenvolvedor é quem define se o contexto deve passar por um agregador, por um interpretador, e assim por diante.

Interpretação de Contexto no *Context Toolkit*

Os interpretadores de contexto são responsáveis por realizar as interpretações. Os interpretadores abstraem informações de contexto cruas ou de baixo-nível em informações de alto-nível. Um exemplo de interpretação é a inferência de que uma reunião está acontecendo, a partir de informações de localização, identidade e áudio de uma sala de conferência. Os interpretadores podem ser simples (ex: utilizar tabelas de busca) ou complexos (ex: utilizar inferências complexas baseadas em técnicas de Inteligência Artificial). Os interpretadores de contexto do *Context Toolkit* são executados pelas aplicações.

3.6.2 Context Fabric

Context Fabric [HON, 2001b] é uma infra-estrutura desenvolvida pela Universidade da Califórnia – Berkeley. Uma de suas características é sua arquitetura em camadas, onde cada camada possui diferentes responsabilidades e fornece conjuntos de serviços específicos.

O Context Fabric é composto por quatro serviços básicos: serviço de evento de contexto (*context event service*), serviço de consulta de contexto (*context query service*), serviço de criação automática de caminho (*automatic path creation*) e serviço de gerenciamento de sensores (*sensor management service*). Existe também uma linguagem por meio da qual aplicações podem especificar as informações de contextos necessárias chamada linguagem de especificação de contexto (*Context Specification Language*). A linguagem de especificação de contexto é a interface pela qual os serviços de evento de contexto e consulta de contexto são acessados (**Figura 3.2**).

Utilizando a linguagem de especificação de contexto, aplicações podem especificar quais eventos estão interessadas e quais desejam ser notificadas de forma assíncrona quando este evento ocorrer; para tanto elas devem subscrever-se no serviço de evento de contexto.

A subscrição necessita de duas partes:

- 1^a) especificação de informações relacionadas ao *subscriber*, por exemplo, como enviar um evento ao *subscriber*, o nome do evento a ser enviado e dados de sensores e contextos relevantes para a situação;
- 2^a) descrição do evento de contexto que deseja receber.

O serviço de consulta de contexto fornece interface para que aplicações possam consultar de forma síncrona uma informação de contexto. As consultas têm o mesmo formato das subscrições de eventos.

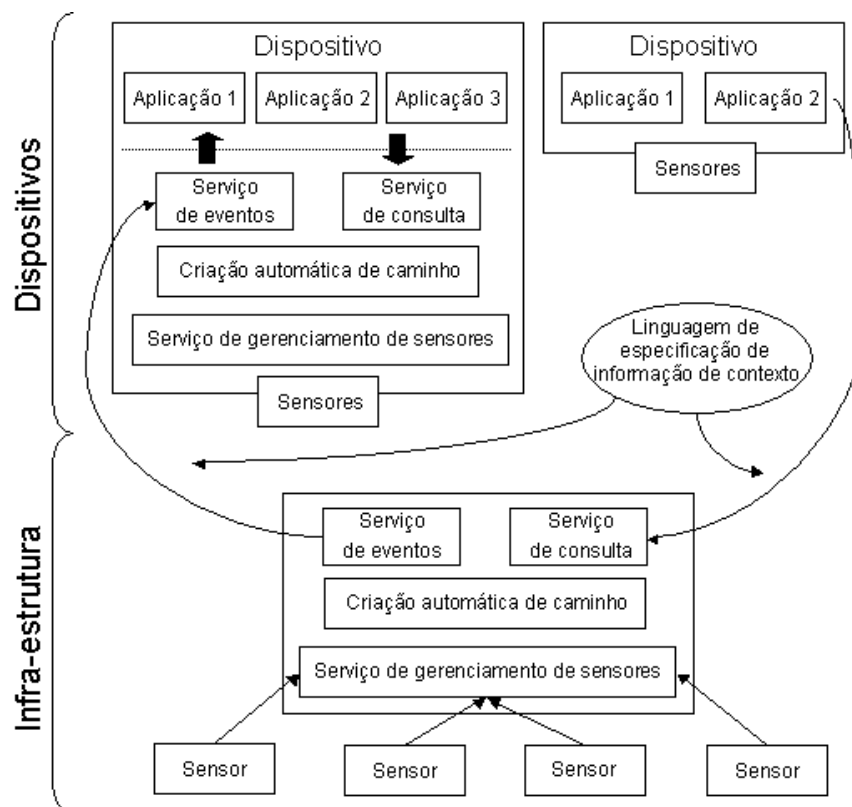


Figura 3. 2 - Arquitetura do Context Fabric – adaptada de [HON, 2001b].

Interpretação de Contexto no *Context Fabric*

Para refinar, combinar e interpretar dados de sensores de baixo nível como dados de contexto de alto nível é utilizada a criação automática de caminho: um serviço que agrupa os contextos necessários e monta o fluxo de dados de sensores e componentes de softwares.

Uma vantagem apontada por [HON, 2001b] na utilização da criação automática de caminho é o fato do desenvolvedor da aplicação não precisar se preocupar com detalhes específicos de sensores e serviços, tendo que se preocupar apenas com a formulação das consulta de

contexto. A **Figura 3.3** apresenta um exemplo do uso da criação automática de caminho no Context Fabric com o objetivo de responder à questão: “Está tendo uma reunião na sala agora?”. Neste exemplo as entradas dos sensores são de áudio e vídeo. Existem alguns componentes de software que fazem a transformação de formatos de dados quando necessário (como de *au* para *wav*), e outros que verificam a ocorrência de algum evento (como o que verifica se o vídeo em *mpeg* está em reprodução, ou se o som em *wav* está audível). A interpretação é feita no componente de software responsável pela fusão dos dados, a partir do qual uma questão pode ser respondida.

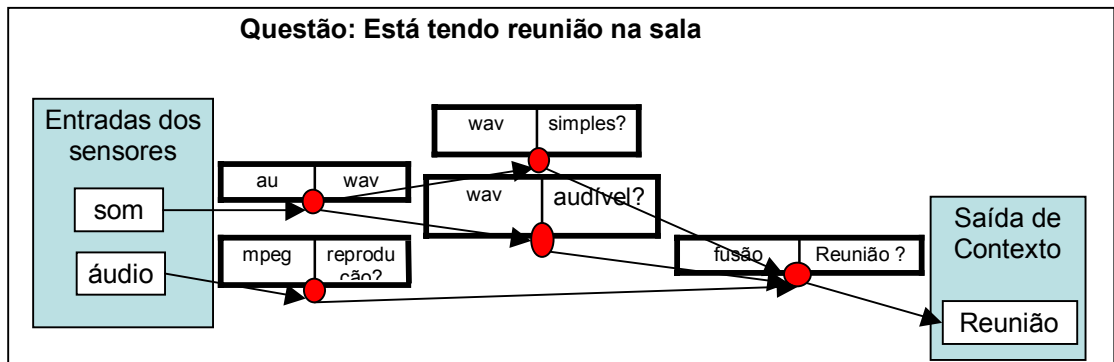


Figura 3.3 - Exemplo de um caminho complexo que tenta calcular a probabilidade de estar tendo uma reunião – adaptada de [HON, 2001b].

3.6.3 GAIA

A infra-estrutura **Gaia** [ROM, 2002] foi desenvolvida pela Universidade de Illinois e trata um espaço ativo (*ActiveSpace*) e seus dispositivos de forma análoga a um sistema operacional tradicional. Espaço Ativo é a abstração usada para se referir a qualquer ambiente de computação ubíqua que possa ser gerenciado pela infra-estrutura Gaia [HES, 2002].

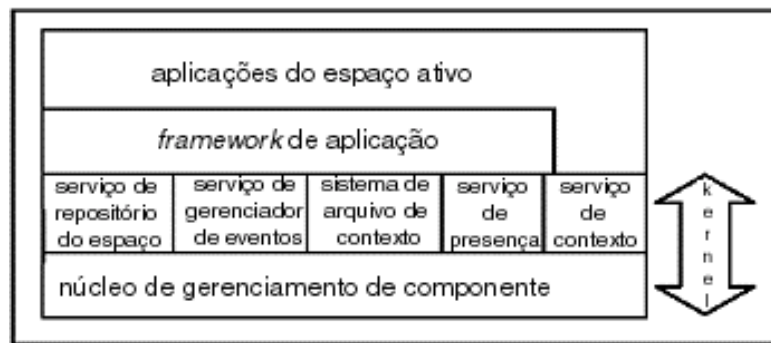


Figura 3. 4 - Arquitetura da Infra-estrutura GAIA – adaptada de [ROM, 2002].

A **Figura 3.4** apresenta a arquitetura da infra-estrutura Gaia que é composta por três blocos principais: *kernel* do Gaia, *framework* de aplicação e aplicações. As aplicações do Gaia são baseadas em componentes distribuídos e móveis, portanto requerem suporte para execução e gerenciamento de componentes remotos; o núcleo de gerenciamento de componente é responsável por essa tarefa. O *kernel* do Gaia é composto por cinco serviços básicos: serviço de presença, serviço gerenciador de eventos, serviço de contexto, serviço de repositório do espaço e sistema de arquivos de contexto. Quanto ao enfoque deste trabalho, serão descritos apenas os serviços de presença, gerenciador de eventos e de contexto.

O serviço de presença é responsável por detectar entidades físicas (pessoas e dispositivos) e digitais (serviços e aplicações) no espaço ativo. O gerenciador de eventos é responsável por criar canais de eventos e fazer a manutenção dos serviços de evento. A implementação atual do gerenciador de eventos utiliza o *CORBA Event Service*, que define duas funções para objetos: função de produtor e função de consumidor [BOR, 2001]. Os produtores produzem dados de eventos e os consumidores processam dados de eventos. Os dados de eventos são trocados entre produtores e consumidores utilizando requisições *CORBA*. Existem duas abordagens para iniciar a comunicação de eventos: modelos *push* e *pull*. O modelo *push* permite que um produtor de eventos inicie a transferência dos dados de

evento para os consumidores. O modelo *pull* permite que os consumidores de eventos solicitem eventos de um produtor. Portanto, o consumidor toma a iniciativa no modelo *pull*, enquanto que no modelo *push* quem toma a iniciativa é o produtor.

O serviço de contexto permite que aplicações pesquisem e se inscrevam para receber informações de contexto. A infra-estrutura de contexto é composta por componentes chamados de provedores de contexto (*context providers*) que fornecem informações sobre o contexto atual. Existe um componente que deduz que tipo de atividade acontece em uma sala específica (ex: reunião ou aula) baseado, por exemplo, em quem está na sala ou que aplicação está sendo executada.

Interpretação de Contexto no GAIA

O modelo de contexto utilizado é baseado em lógica de primeira ordem e álgebra booleana. As regras e consultas são expressas utilizando lógica de primeira ordem, e mecanismos similares aos do Prolog são usados para inferir propriedades que envolvem contextos.

3.6.4 iROS

iROS (*Interactive Room Operating System*) [JOH, 2002] é uma infra-estrutura construída pela Universidade de Stanford para dar suporte a aplicações dentro de um espaço físico utilizando também a abordagem de sistema operacional. Espaço de trabalho interativo (*Interactive Workspace*) é uma abstração que consiste de um ambiente com alta tecnologia onde pessoas podem trabalhar de maneira colaborativa. Na **Figura 3.5** encontra-se a

arquitetura do iROS composta por três subsistemas: memória de eventos (*EventHeap*), memória de dados (*DataHeap*) e *ICrafter*.



Figura 3. 5 - Arquitetura da Infra-estrutura iROS – adaptada de [JOH, 2002].

A memória de eventos (*Event Heap*) [JOH, 2002a] armazena e encaminha mensagens conhecidas como eventos por meio de espaços de tuplas, ou seja, os dados são enviados na forma de tuplas, onde cada tupla é um conjunto ordenado de valores com campos *nome-tipo-valor*. Os produtores colocam as tuplas em um espaço de tuplas, e os consumidores recuperam essas tuplas usando um modelo de tupla cujos parâmetros são especificados com valores onde uma correspondência exata é desejada, ou com variáveis quando os valores desejados resultam da comparação das tuplas. Uma extensão feita dos espaços de tupla foi o acréscimo de expiração de eventos, permitindo que eventos que não são consumidos sejam removidos automaticamente. As aplicações podem se conectar com a memória de eventos através de várias APIs em Java e C++.

A memória de dados (*Data Heap*) permite que qualquer aplicação armazene dados em um repositório associado ao ambiente local. Os dados são armazenados com um número arbitrário de atributos que o caracterizam e pode ser recuperado por uma consulta que especifique os atributos que devem ser compatíveis. Usando atributos em vez de localizações,

aplicações não precisam saber qual sistema de arquivo físico específico está sendo usado para armazenar os dados. Os formatos de dados também são armazenados na memória de dados, e supõe-se que transformações apropriadas sejam feitas para que os dados sejam transformados automaticamente para o melhor formato suportado pelas aplicações que irão recebê-los.

O *ICrafter* consiste de um *framework* para a criação de serviços que podem ser controlados por aplicações ou diretamente por usuários. Serviço é qualquer entidade de software ou hardware controlada por um usuário através da rede (ex: um projetor ou um navegador web). O *ICrafter* fornece também um gerador de interfaces para serviços, que permite que usuários selecionem um serviço para controlar e retornar a melhor interface para o dispositivo de acesso do usuário.

Interpretação de Contexto no iROS

A interpretação fica a cargo da aplicação. iROS não oferece serviço de interpretação de contexto [PON, 2003].

A **Tabela 3.1** mostra uma comparação das formas de interpretação de contexto utilizadas nos sistemas descritos acima.

Tabela 3.1 - Tabela comparativa de formas de interpretação de contexto

	Forma de interpretação de contexto
<i>Context Toolkit</i>	Interpretação simples (ex: tabelas de busca) ou complexa (ex: inferências baseadas em técnicas de Inteligência Artificial). Os interpretadores são definidos pelas aplicações
<i>Context Fabric</i>	Técnica de criação automática de caminho
GAIA	Mecanismos de inferência similares aos utilizados na linguagem Prolog
iROS	A interpretação fica a cargo da aplicação. iROS não oferece serviço de interpretação de contexto.

3.7 Considerações Finais

Com base nos requisitos de infra-estruturas de software para ciência de contexto, é possível realizar um estudo comparativo entre os trabalhos apresentados na seção anterior. O Context Toolkit prioriza o suporte ao desenvolvimento de aplicações que obtêm informações de contexto via sensores distribuídos em um ambiente. Seus *widgets* permitem que aplicações acessem diferentes tipos de contexto sem se ater aos detalhes de como estas foram obtidas. O Context Fabric vai mais além ao propor serviços especializados em notificação de eventos e em consulta de informações de contexto, bem como o uso da linguagem XML para a especificação de consultas e notificação de eventos relacionados a informações de contexto.

As infra-estruturas Gaia e iROS utilizam abstrações similares para o tratamento de contextos distribuídos. Gaia considera espaços ativos como agrupamentos de entidades, representadas em XML, do ambiente em que se encontra um usuário. Seu ponto forte é relacionar a mobilidade de contexto e de aplicações entre diferentes espaços ativos com a mobilidade de um usuário. iROS emprega a abstração de espaços de trabalho interativos. Porém, iROS não trata a mobilidade de usuário entre espaços, pois seu foco é o suporte à colaboração dentro desses espaços de trabalho.

Os projetos apresentados nesse capítulo enfocam o auxílio a aplicações voltadas basicamente para o domínio de sala de aula e sala de reuniões. Porém, os ambientes que precisam ser monitorados apresentam algumas características específicas, como já foi apresentado no Capítulo 2. Portanto, torna-se necessária a criação de um *middleware* que tenha seus serviços voltados às necessidades desse tipo de ambiente. O próximo capítulo apresenta uma proposta de aplicação da Computação Ciente de Contexto no Monitoramento

de Condições Críticas em Ambientes Físicos, a especificação de um *middleware*, e os serviços de interpretação de contexto, notificação de eventos e localização, que fazem parte desse *middleware*.

4. Computação Ciente de Contexto Aplicada ao Monitoramento de Condições Críticas em Ambientes Físicos

4.1 Considerações Iniciais

O monitoramento de condições críticas consiste em supervisionar situações que podem levar a um incidente ou acidente caso essas situações não sejam controladas. As condições críticas são aquelas consideradas anormais para determinados domínios de aplicação, como por exemplo: temperatura ou pressão acima de um certo limite ou existência de fumaça em um determinado local. Vale ressaltar que existe uma diferença entre “*security*” e “*safety*”. O termo “*security*” está relacionado com perdas materiais, ou seja, financeiras. Já o termo “*safety*” é um pouco mais abrangente e trata de perdas relacionadas a situações que colocam em risco vidas humanas. Neste trabalho quando o termo segurança for utilizado, estaremos nos referindo a segurança tanto material quanto às relacionadas à vida de seres humanos.

Para que um plano de segurança seja eficiente ele deve analisar o processo como um todo, incluindo pessoal, equipamento, e componentes ambientais. Um plano de segurança completo normalmente inclui os seguintes sub-planos:

1. Plano preventivo: usado para minimizar riscos e pode conter, por exemplo, uma lista de perigos em potencial, a prioridade desses perigos, possíveis resoluções para cada um deles e uma lista de ações para amenizar estes riscos;
2. Plano de análise: usado para analisar possíveis falhas em componentes ou processos, podendo seguir normas existentes, como as especificadas na FMEA (*Failure Modes and Effects Analysis*) americana [FME, 2004];
3. Plano de comunicação: usado para descrever os relatórios que devem ser feitos quando ocorrer um incidente de segurança. Indica o grau de severidade do incidente sendo relatado, o método a ser usado para relatar e a lista de quais entidades devem ser notificadas.

Um plano de segurança permite a quantificação dos riscos, o que promove a própria assegurabilidade e o potencial comercial do projeto [FMEA, 2004]. Devido à diversidade de objetivos e de especialidades das empresas (ex: manipulação e disposição de materiais químicos, tóxicos, radioativos, infecciosos e explosivos) é quase impossível ter um mesmo plano de segurança que valha para toda e qualquer empresa. Entretanto, um plano pode ter partes adaptadas para necessidades específicas de uma empresa ou instituição.

Este trabalho trata do aspecto de política de segurança relacionada ao monitoramento de condições críticas em ambientes que são monitorados por meio de sensores. O trabalho não pretende automatizar todo o processo relacionado aos aspectos de monitoramento do

ambiente, mas sim, aqueles em que decisões são tomadas a partir de informações capturadas e interpretadas de uma rede de sensores. A proposta deste trabalho é a de prover uma estrutura de monitoramento de granularidade fina em ambientes que podem estar sujeitos a situações de emergência, portanto informações mais refinadas são coletadas seguindo uma política de monitoramento baseada no plano preventivo de riscos, conforme descrito acima.

4.2 Projeto de um Sistema de Captura e Acesso de Condições Críticas em Ambientes Físicos

O projeto do *Sistema de Captura e Acesso de Condições Críticas em Ambientes Físicos* está sendo desenvolvido por três alunos de mestrado do LRVNet – Laboratório de Realidade Virtual em Rede, da UFSCar – Universidade Federal de São Carlos [KUD, 2003] [MIC, 2003]. Esse projeto é composto de três partes cuja visão geral é apresentada na **Figura 4.1**. Cada uma dessas partes é descrita a seguir.

1. Especificação de uma estrutura para captura de contextos de redes de sensores sem fio, projeto e simulação de protocolos de comunicação;
2. Especificação de um *middleware* para aplicações de monitoramento de condições críticas de ambientes físicos de interação e os seguintes serviços: Serviço de Interpretação de Contexto, Serviço de Notificação de Eventos e Serviço de Localização. Esta parte é o foco do presente trabalho (parte sombreada da **Figura 4.1**)

3. Especificação de uma estrutura de visualização e reprodução de ambientes virtuais 3D para monitoramento de condições críticas em ambientes físicos em tempo-real e posteriormente.

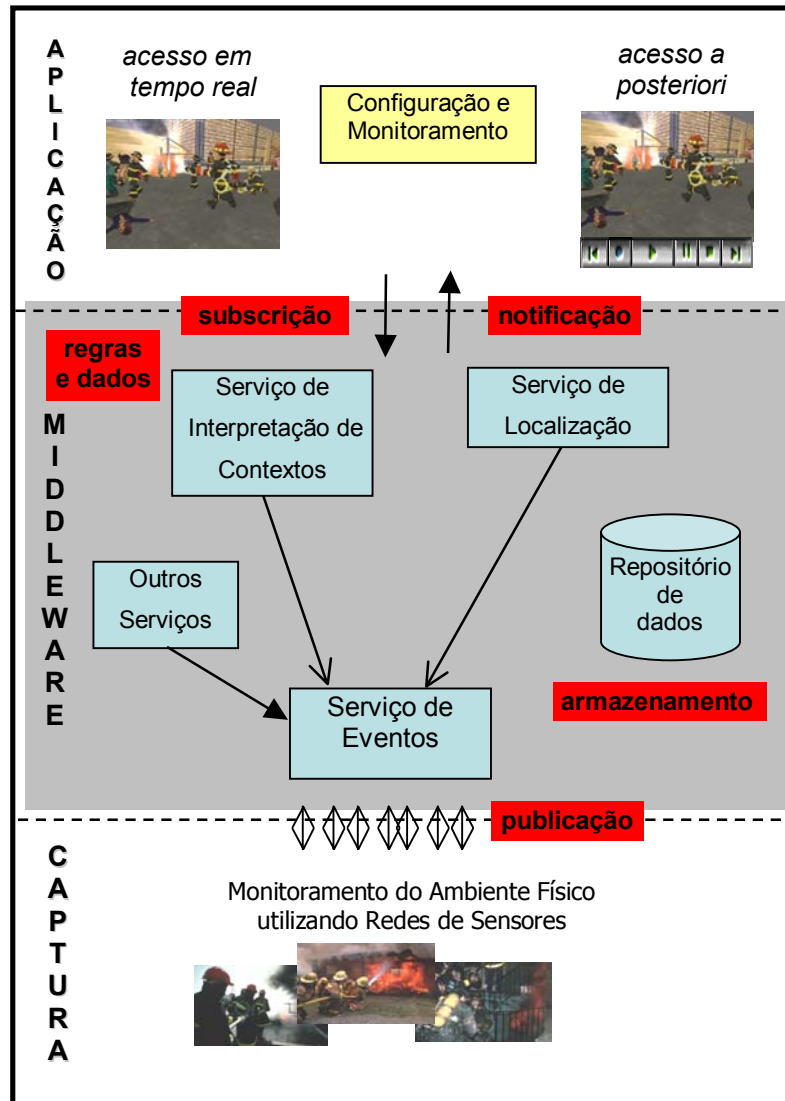


Figura 4.1 - Visão Geral do Sistema

Para que as partes de captura de dados e interpretação de contexto possam ser utilizadas por uma grande variedade de dispositivos e por qualquer tipo de aplicação, verifica-

se a necessidade de um *middleware* que forneça serviços independentes de plataforma, sistema operacional e/ou linguagem de programação.

A próxima seção apresenta a proposta de um *middleware* e seus respectivos serviços para o monitoramento de condições críticas.

4.3 Proposta de um Middleware para Aplicações de Monitoramento de Condições Críticas

Esta seção apresenta a proposta de um *middleware* para Aplicações de Monitoramento de Condições Críticas, a descrição deste *middleware* com os serviços, a comunicação entre os módulos do *middleware*, e os serviços de notificação de eventos, interpretação de contexto e localização.

4.3.1 Descrição do *Middleware*

O *middleware* para aplicações de monitoramento de condições críticas está em fase de especificação e até então conta com os seguintes serviços, conforme mostra a **Figura 4.2**.

- **Serviço de descoberta de recursos:** responsável por descobrir a localização de recursos no ambiente. Por exemplo, quando uma aplicação deseja se comunicar com um nó destino (*sink*) de uma rede de sensores para receber informações diretamente da rede de sensores, a aplicação consulta o serviço de recursos para saber a localização e o protocolo de comunicação utilizado por aquele nó destino;

- **Serviço de notificação de eventos:** responsável por receber subscrições, registrar estas subscrições nos eventos desejados, encaminhá-las como subscrições à rede de sensores, além de notificar entidades participantes quando forem satisfeitas as condições especificadas por estas nas subscrições;
- **Serviço de localização:** responsável por receber subscrições de consulta sobre localização de pessoas e objetos. Dependendo da tecnologia envolvida nos sensores de localização, as tarefas deste serviço podem ser mais ou menos complexas;
- **Serviço de interpretação de contexto:** responsável por receber as notificações do serviço de eventos e interpretar essas informações utilizando regras fornecidas pela aplicação;
- **Serviço de adaptação de conteúdo:** responsável por adaptar conteúdos em função das capacidades dos dispositivos, do meio de comunicação e da aplicação;
- **Repositório de dados:** responsável pelo armazenamento dos dados de contexto puros e interpretados.

Este *middleware* tem sido especificado de forma a atender os requisitos de uma estrutura de computação ubíqua (citados no capítulo 3), como descoberta de recursos, separação de aquisição e interpretação de contextos, adaptação de conteúdo, comunicação distribuída e transparente e disponibilidade contínua dos componentes de captura.

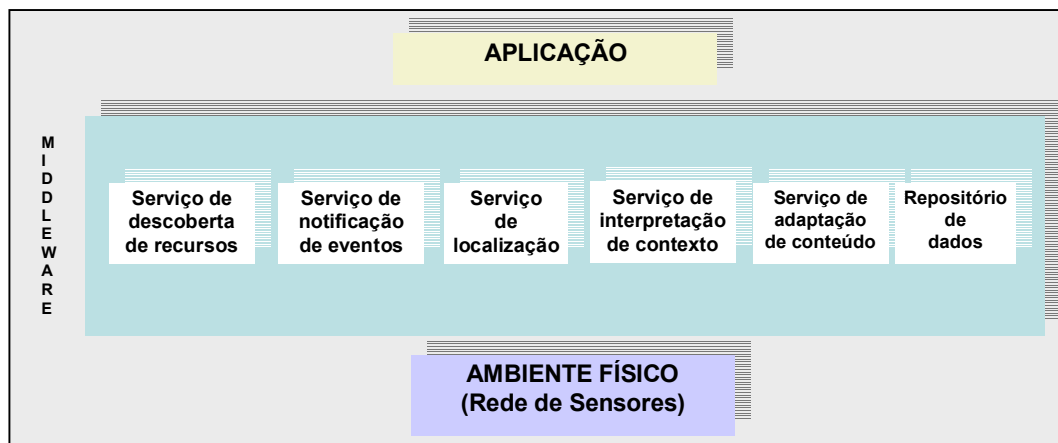


Figura 4. 2 - Aplicação, Middleware e Ambiente Físico (rede de sensores)

O foco deste trabalho é a especificação dos módulos de serviço de interpretação de contextos, serviço de localização e serviço de notificação de eventos. Para que a interpretação de contexto seja feita, é necessária a entrada dos dados que gerem as regras de interpretação (ex: condições de monitoramento) por meio de uma interface de configuração e monitoramento. Depois que esses dados são fornecidos, é necessário emitir as subscrições ao serviço de eventos, para que se saiba quando e o quê enviar ao serviço de interpretação de contexto. Quando o interesse da aplicação é na localização de pessoas e/ou objetos, o serviço de localização deve emitir subscrições ao serviço de eventos para que eventos capturados através de rastreamento de sensores possam ser notificados ao serviço de localização que, de posse dessa informação, pode gerar localizações mais precisas ou mais próximas do contexto requerido pela aplicação.

4.3.2 A Comunicação entre os Módulos do Middleware

O mecanismo de comunicação utilizado entre as aplicações e os serviços do *middleware* é baseado no mecanismo de *Publish/Subscribe*. Neste mecanismo, os *subscribers* (entidades interessadas em receber eventos) podem definir seus interesses em cada evento e serem notificados quando um evento de seu interesse for gerado por um *publisher*. Um evento é

propagado, através de um serviço de notificação de eventos, para todos os *subscribers* interessados naquele evento, de forma assíncrona, diminuindo assim o tráfego de dados desnecessários. Vários paradigmas de comunicação podem ser usados para promover a interação entre as entidades que produzem eventos e os destinos que consomem estes eventos. Exemplos de tais paradigmas incluem troca de mensagens, invocações remotas, notificações, espaços compartilhados e enfileiramento de mensagens. O problema básico com estes paradigmas é que eles falham em promover um desacoplamento total entre as entidades participantes, tornando o sistema menos flexível e menos escalável. Eugster et al. [EUG,2000] fazem um excelente estudo destes paradigmas e os compara com o paradigma de publicação/subscrição, o qual tem recebido crescente atenção por oferecer desacoplamento entre produtores e consumidores no **tempo** (publicadores e subscritores não precisam estar ativos na interação ao mesmo tempo), **espaço** (publicadores e subscritores não precisam conhecer um ao outro) e **fluxo** (publicadores e subscritores não precisam estar sincronizados para interagir). Os modelos mais conhecidos de publicação/subscrição são [EUG, 2000]:

1. *Baseado em tópico ou assunto* – as subscrições identificam apenas classes de eventos pertencentes a um dado canal ou assunto, em que tópicos representam palavras-chave, do tipo: cotação de ações, ou jogos do campeonato brasileiro. Participantes podem publicar notificações e subscrever para recebê-las. São mais simples de implementar, mas menos flexíveis e expressivos;
2. *Baseado em conteúdo* – eventos são classificados de acordo com propriedades, especificadas pelos *subscribers*, que os eventos podem ter – apenas os eventos que satisfazem estas propriedades serão entregues aos *subscribers*. Isto dá às entidades subscritoras muito mais opções de subscrição, além de filtragem de informação, o que reduz o tráfego da rede, porém aumenta a complexidade do sistema;

3. *Baseado em Tipo* – os eventos são objetos, como, instâncias de tipos nativos em uma linguagem de programação orientada a objetos, o que leva a maior segurança em relação a tipos, além de melhor encapsulamento de eventos. No modelo baseado em tipo, o *subscriber* de um tipo específico de objeto só vai receber eventos daquele tipo e de seus sub-tipos.

Neste trabalho, o mecanismo de publicação/subscrição baseado em tópicos é usado para enviar e receber notificações de diferentes tipos de interesse. Este modelo foi escolhido por ser simples de implementar, o que pode representar menor latência na entrega dos eventos, um requisito importante em ambientes de monitoramento de condições críticas. A **Figura 4.3** mostra a interação entre a aplicação e os serviços do *middleware*, através do paradigma da publicação/subscrição.

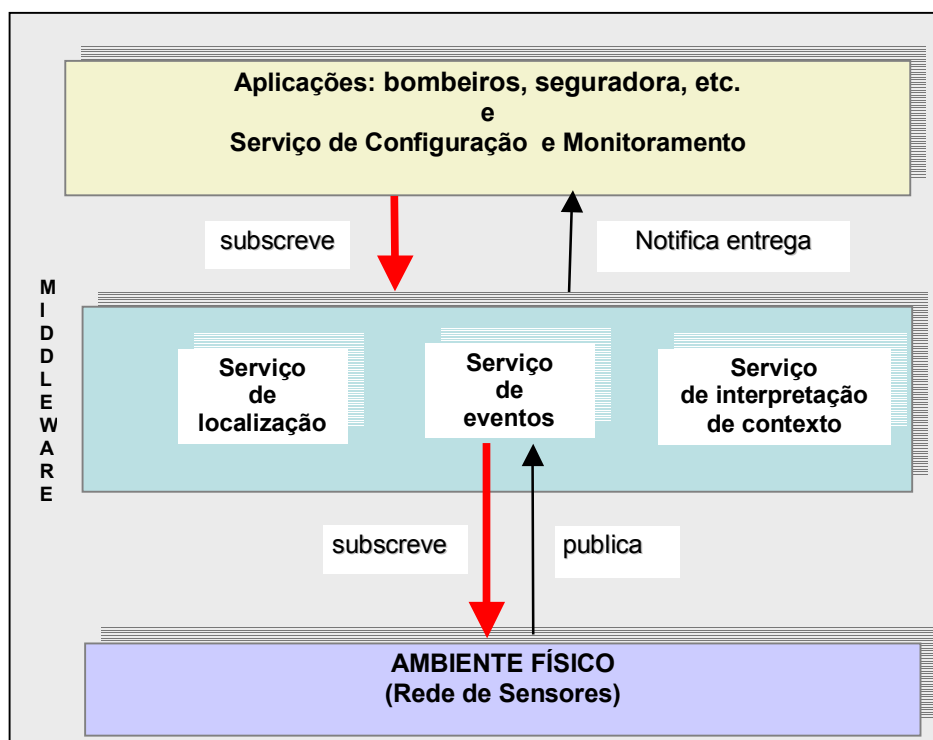


Figura 4.3 - Interação entre Aplicação, Serviços do Middleware e Rede de Sensores

Na **Figura 4.3**, aplicações podem demonstrar interesse em eventos do ambiente, através da emissão de subscrições para o *middleware*, em tópicos específicos, do tipo média de temperatura do ambiente, umidade de uma área do ambiente, situação de incêndio, rebelião, explosão, localização de pessoas e/ou de objetos, etc.

Para isto, é importante que um sistema de monitoramento de condições críticas forneça suporte para os seguintes tipos de monitoramento:

1. periódica: o usuário precisa especificar um intervalo de tempo em que tem interesse para receber informações do ambiente, no caso, relacionado a uma entidade e a um tipo de sensor. Por exemplo, se o usuário deseja receber a cada 10 minutos informações sobre a temperatura do setor1, as informações de contexto envolvidas são: entidade = setor1; tipo de sensor = temperatura; unidade de tempo = minutos; valor = 10.
2. orientada a eventos: o usuário precisa especificar uma condição em que tem interesse para receber informações, sempre relacionado a uma entidade e a um tipo de sensor. Por exemplo, se o usuário deseja somente receber dados sobre a temperatura do setor1 quando a temperatura estiver acima de 50: entidade = setor1; tipo de sensor = temperatura; operador = >; valor = 50.
3. baseada em consulta: o usuário define a entidade e o tipo de sensor de seu interesse. Por exemplo, para consultar a temperatura do setor1: entidade = setor1; tipo de sensor = temperatura.

As próximas seções descrevem alguns dos serviços do *Middleware* para aplicações de monitoramento de condições críticas, a saber: Serviço de Notificação de Eventos, Serviço de Interpretação e Serviço de Localização.

4.3.3 O Serviço de Notificação de Eventos

O serviço de notificação de eventos é especificado como uma coleção de tópicos (seção 4.4) sobre os quais entidades participantes mostram interesse. As aplicações inscrevem-se para receber eventos, e são notificadas assim que os eventos se tornam disponíveis (esquema de “*push*”). Da mesma maneira, vários serviços do *middleware* também inscrevem-se para receber eventos, como mostra a **Figura 4.4**. A subscrição dos serviços de interpretação de contextos e de localização em eventos de interesse que são publicados por sensores de interesses da aplicação, podem ser divididos em três tipos de informação:

1. **consultas** (sobre localização ou estado atual da informação de um sensor);

A aplicação emite consultas do tipo: “Onde está Pedro?” para o serviço de localização que, para atender a esta solicitação inscreve-se para receber eventos de sensores que detectam, por exemplo, a força do sinal emitido por uma etiqueta ou crachá de RF fixada no corpo da pessoa “Pedro”. A seção sobre o Serviço de Localização descreve o atendimento a essas consultas com maior detalhe;

2. **monitoramento de condições críticas** (notificar a ocorrência de situações de emergência do tipo incêndio, detecção de gases tóxicos etc);

A aplicação de monitoramento do ambiente físico emite interesse em receber indicações sobre a ocorrência de situações de emergência. Estas situações são expressas na forma de expressões lógicas (fornecidas pelo usuário, através de uma

interface de configuração e monitoramento) do tipo: “*Presença de Fogo AND Presença de Fumaça AND Temperatura > 80°C*”, que indicam uma situação de incêndio. O serviço de interpretação analisa esta expressão e subscreve-se em cada um dos eventos envolvidos na expressão acima, e que leva à condição de incêndio. Assim que esses eventos são publicados pelos respectivos sensores, o Serviço de Interpretação é notificado. Quando todos os eventos correspondentes à expressão estão disponíveis, o serviço de interpretação processa a expressão lógica e notifica imediatamente a aplicação caso o resultado seja verdadeiro;

3. **Informação capturada diretamente dos sensores**, sem passar pelos serviços de Interpretação ou de Localização, do tipo informações periódicas (notificação da pressão de uma caldeira a cada 10 minutos, ou de notificação quando temperatura do setorA for $> 50^{\circ}\text{C}$).

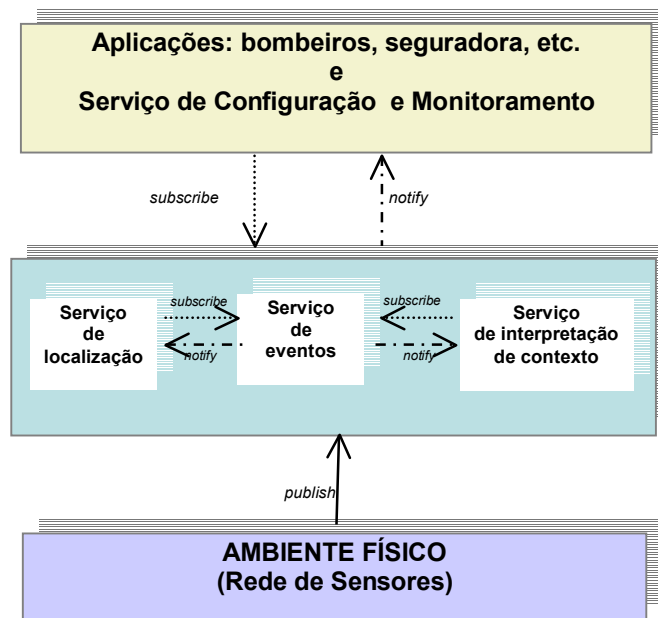


Figura 4. 4 - Interação entre Aplicação, Serviços de Evento, Interpretação e Localização e Rede de Sensores

A seção 4.4 descreve com mais detalhes as abstrações utilizadas no serviço de notificação de eventos, bem como as operações sobre estas abstrações.

4.3.4 O Serviço de Interpretação de Contexto

Quando a informação desejada pela aplicação é simples, esta pode se inscrever diretamente no evento que gera esta informação. Casos mais complexos, do tipo consultas sobre localização ou monitoramento de condições críticas, são emitidos para os serviços correspondentes. Em situações de monitoramento, que envolve o processamento de expressões lógicas, o serviço de interpretação de contexto é acionado. A **Figura 4.5** mostra a descrição do esquema XML correspondente à subscrição de monitoramento para o serviço de interpretação.

```
<subscribe>
  <subscription_id></subscription_id>
  <subscription_type></subscription_type>
  <subscription_value></subscription_value>
  <entity_type></entity_type>
  <entity_value></entity_value>
  <text></text>
  <value></value>
</subscribe>
```

Figura 4. 5 - Exemplo de Esquema XML para subscrição de Monitoramento de condição crítica

Na **Figura 4.5**, temos:

- **subscription_id** - associa uma subscrição a uma determinada notificação;
- **subscription_type** - indica qual o tipo de subscrição desejada (periódica, orientada a eventos e baseada em consulta);

- **subscription_value** - indica o tipo de subscrição relacionada ao serviço solicitado (de uma lista a ser descrita em cada um dos serviços abaixo);
- **entity_type** - indica o tipo de entidade sobre a qual se requer informação (Ex: person; object; environment);
- **entity_value** - refere-se à entidade sobre a qual se requer informação;
- **text e value** - descrevem informação dependente do tipo de subscrição (periódica, orientada a eventos, baseada em consulta).

Um motor de interpretação, parte do Serviço de Interpretação é responsável por gerar, a partir das informações necessárias para o processamento da expressão, subscrições de interesse em informações publicadas pelos sensores correspondentes. Cada uma dessas subscrições é identificada unicamente. A seção 4.4 descreve a forma como as subscrições são feitas. Uma estrutura no serviço de interpretação é instanciada e identificada através de `subscribe_id`, que é ligado à condição sendo monitorada (por exemplo, incêndio). Nesta estrutura são colocados os valores entregues pelo serviço de notificação (publicados por sensores), através de notificação de eventos. Cada evento notificado é munido de um identificador único que emparelha com o identificador da subscrição correspondente. O motor de interpretação processa a expressão (**Figura 4.5**) quando todos os eventos correspondentes à expressão são entregues. O resultado do processamento é entregue para a aplicação, através de notificação.

Subscrições para monitoramento do ambiente podem ser solicitadas pela aplicação a qualquer momento. As situações de monitoramento são definidas com base nas condições que o responsável pela segurança definiu, através da interface de configuração e monitoramento.

Exemplos de condições seriam: incêndio, explosão, curto-circuito, etc. Para cada situação a ser monitorada, o responsável de segurança também define a expressão lógica correspondente, cuja interpretação, pelo motor de interpretação do serviço de interpretação, identifica os sensores envolvidos no monitoramento daquela situação. Por exemplo, para o monitoramento da condição de incêndio, são necessárias subscrições aos eventos dos seguintes sensores: temperatura, fumaça e chama.

4.3.5 O Serviço de Localização

O rastreamento de sensores para a localização de pessoas e/ou objetos é um serviço importante em aplicações de monitoramento de condições críticas, uma vez que a localização de pessoas, durante um incêndio por exemplo, pode minimizar o número de vítimas quando esta informação é utilizada por equipes de resgate e de combate ao fogo na tomada de decisões. Ou seja, a localização de pessoas é importante para traçar estratégias de salvamento.

Os métodos mais comuns de localização usam medir a distância de uma fonte emissora (por exemplo, uma pessoa usando um crachá ou etiqueta RFID) móvel ou não, aos sensores fixos. Essas medidas podem ser baseadas na **Potência do Sinal** (através de leituras de RSSI - *Received Signal Strength*); no **Tempo de Chegada** (tri-lateração hiperbólica, ToA ou Diferença de tempo de chegada - TDoA); ou ainda no **Ângulo de Chegada** (AoA, triangulação). A informação de localização (por exemplo, as posições x, y e z mais provável da fonte emissora) é publicada para o Serviço de Notificação de eventos que notifica o Serviço de Localização. O serviço de localização, a partir de um motor de localização, mapeia esta informação em uma estrutura que traduz a posição para uma localidade conhecida (sala1, setorA, ala3, etc).

Basicamente, três consultas podem ser feitas a uma rede de sensores para a localização de pessoas e/ou objetos em ambientes:

1. Onde está pessoa X?
2. Quem está no ambiente X?
3. Tem alguém no ambiente Y?

Estas consultas podem ser feitas pelas aplicações ao serviço de localização do *Middleware*. A **Figura 4.6** mostra a descrição do esquema XML correspondente à subscrição de consultas de localização para o serviço de localização.

```
<subscribe>
  <subscription_id></subscription_id>
  <subscription_type></subscription_type>
  <subscription_value></subscription_value>
  <entity_type></entity_type>
  <entity_value></entity_value>
</subscribe>
```

Figura 4. 6 - Esquema XML de subscrição para Localização

Na **Figura 4.6**, temos:

- **subscription_id** - associa uma subscrição a uma determinada notificação;
- **subscription_type** – indica qual o tipo de subscrição desejada (periódica, orientada a eventos e baseada em consulta);
- **subscription_value** – indica o tipo de subscrição (Onde está pessoa X?; Quem está no ambiente X?; Tem alguém no ambiente Y);
- **entity_type** – indica o tipo de entidade sobre a qual se requer informação (Ex: *person; object; environment*);

- **entity_value** - refere-se à entidade sobre a qual se requer informação.

Um motor de interpretação, parte do Serviço de Localização é responsável por gerar, a partir dos dados recebidos da aplicação, através da subscrição de localização, subscrições em sensores de localização. É responsável também por mapear, ao receber as publicações dos sensores de localização, a posição recebida.

A vantagem dos contextos serem fornecidos à aplicação no formato XML é o fato de não ficar atrelado a nenhum tipo de mídia, sendo que a aplicação pode utilizá-lo no formato que desejar (textual, 2D, 3D, etc), e também em qualquer tipo de dispositivo (PC, notebook, PDAs, celulares) como pode ser visto na **Figura 4.7**. Quando os serviços estiverem distribuídos, a comunicação é feita via HTTP/XML.

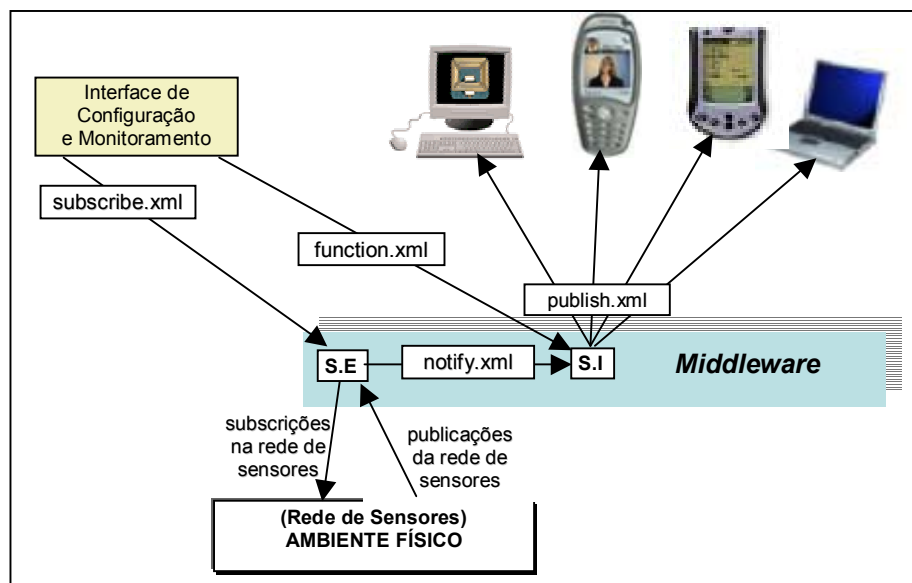


Figura 4.7 - Exemplo de como os arquivos XML são utilizados e como o arquivo **publish.xml** pode ser usado por qualquer aplicação independente de dispositivo e/ou mídia

As próximas seções descrevem o conceito de Tópico adotado neste trabalho, bem como as operações sobre os tópicos para as subscrições e publicações de informações de interesse.

4.4 Informações de Interesse de Aplicações de Monitoramento de Condições

Críticas

As aplicações de monitoramento de condições críticas têm interesse em informações capturadas do ambiente que possam sinalizar, em especial, condições anormais, sejam estas de tipo simples, como temperatura ou pressão muito altas, ou complexas como um incêndio ou explosão, ou ainda a localização de vítimas em situações de incêndio.

Uma lista de interesses dessa classe de aplicações inclui: qualquer tipo de informação para a qual exista um sensor capaz de capturá-la (“senti-la”); condições críticas inerentes a um ambiente específico de aplicação (os interesses de uma aplicação de monitoramento de condições críticas em uma indústria de produtos químicos certamente serão diferentes em um museu ou em um avião). Desta forma, um sistema de suporte à classe de aplicações de monitoramento de condições críticas em diferentes cenários de aplicação, deve ter, além de uma interface de configuração que se molde às políticas de segurança para monitoramento de ambientes com características diferentes, um esquema de subscrição e notificação de eventos eficiente, um serviço de interpretação de contextos que corresponda aos anseios da aplicação e um serviço de localização de baixa latência.

4.4.1 A definição da entidade Tópico num modelo de publicação/subscrição baseado em Tópico

Um tópico é representado por uma abstração que representa, em sua essência, um motor de notificação [EUG, 2000]. No modelo clássico de interação *Publish/Subscribe*, os Tópicos costumam ser vistos como grupos, em que a subscrição para um tópico T é vista como uma

associação a um grupo T. Entretanto, eles têm uma natureza mais dinâmica, uma vez que os participantes podem subscrever em vários tópicos diferentes enquanto os grupos, normalmente têm um conjunto disjunto de membros.

Em um modelo de *Publish/Subscribe* baseado em Tópicos, um participante mostra interesse em um tópico, subscrevendo no tópico. A **Figura 4.8** mostra a dinâmica do modelo *Publish/Subscribe* baseado em Tópicos. O *subscriber* é notificado dos eventos correspondentes ao sub-tópico e tópico.

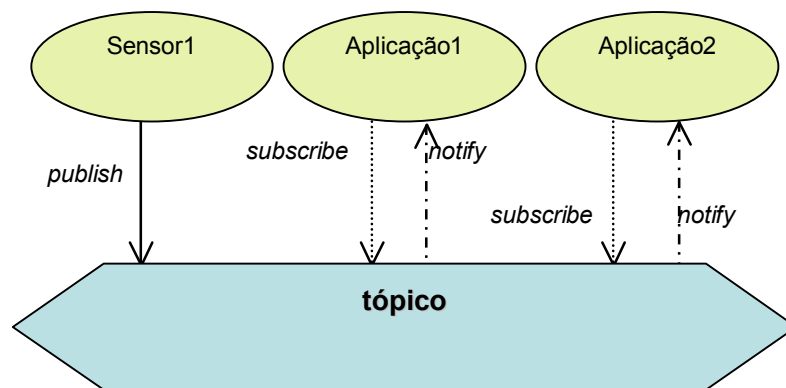


Figura 4.8 - Publish/Subscribe baseado em Tópicos – adaptado de [EUG, 2000].

Os tópicos podem ser estruturados hierarquicamente em sub-tópicos, por exemplo, um tópico pode ser derivado ou especializado de um outro tópico, como mostra a **Figura 4.9 (a)** e **(b)**.

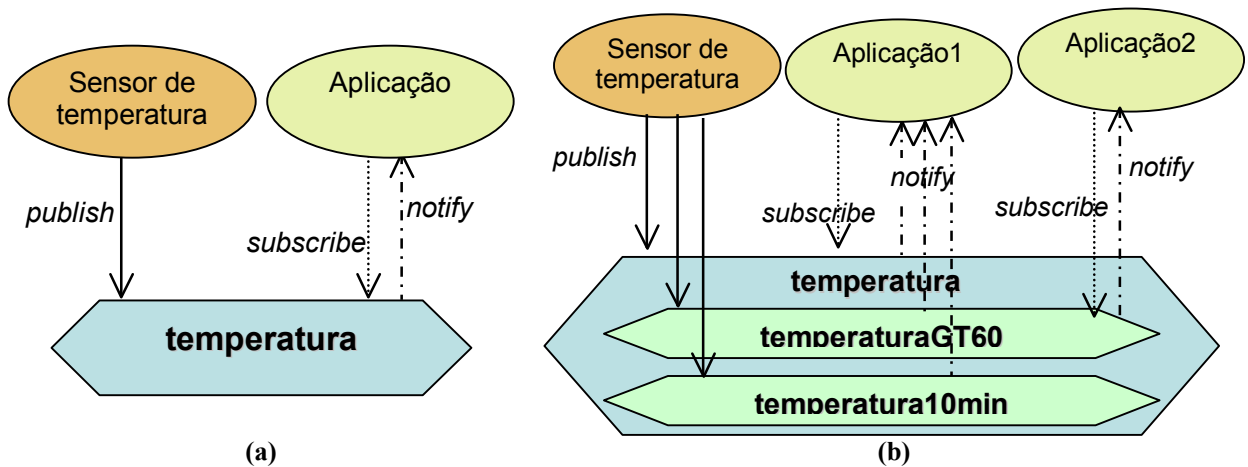


Figura 4. 9 - Exemplo de Publish/Subscribe baseado em Tópicos para um tópico de temperatura (a), e exemplo de Publish/Subscribe baseado em Tópicos para um tópico de temperatura e um sub-tópico de temperatura>60 (b)

4.4.2 Demonstrando Interesse em Tópicos

Conforme descrito no texto acima, os tipos de interesse com os quais uma rede de sensores é configurada (por exemplo, por um especialista em segurança) depende: do tipo do ambiente a ser monitorado (e a existência de sensores para isso); da política de segurança da empresa interessada no monitoramento; e das políticas de segurança ditadas por órgãos de controle (como governos, corpo de bombeiros, etc). A partir dos dados que o especialista fornece pela Interface de Configuração, é gerado um conjunto de subscrições em diferentes tópicos. Novos tópicos podem ser definidos, através da interface de configuração e monitoramento do sistema. Alguns exemplos de tópicos incluem: temperatura, pressão, umidade, localização, incêndio, explosão, rebelião. A subscrição da aplicação nestes tópicos dá-se através da estrutura mostrada na **Figura 4.10**. Essas subscrições são emitidas ao serviço de eventos do *middleware*.

```
<subscription>  
  <topic = "temperatura"/>  
  <subtopic = "temperatura/temperatura10min"/>  
</subscription>
```

Figura 4. 10 - XML para subscrição no serviço de eventos do middleware

Na **Figura 4.10**, temos um exemplo de subscrição ao tópico “temperatura” e ao subtópico “temperatura10min”.

As subscrições aos tópicos são emitidas para a rede de sensores que capturam informações do ambiente físico. Quando estas coincidirem com os critérios descritos nas subscrições, são publicadas pelos sensores e notificadas ao serviço de notificação que decide quem deve ser notificado (serviço de localização, serviço de interpretação ou aplicação).

4.4.3 Notificação de Evento para um Tópico

Quando um evento é publicado por um sensor sobre um Tópico, por exemplo, um sensor de temperatura que captura informação a cada 10 minutos, todos os participantes que se inscreveram naquele tópico são notificados da chegada do novo evento.

4.5 Interfaces de Configuração e Monitoramento de Condições Críticas

Com o objetivo de fornecer ao usuário uma maneira de entrar com informações importantes para configuração das situações que devem ser monitoradas no ambiente físico foi construída uma Interface de Configuração e Monitoramento (**as telas são apresentadas no Apêndice A**). Através desta interface o usuário (responsável pela segurança) deverá fornecer as seguintes informações para a configuração:

- o que deve ser protegido (será considerado uma entidade para o sistema);

- quais os tipos de sensores existentes no ambiente, relacionados com cada entidade em particular e o número de identificação de cada sensor;
- contra quais situações as entidades devem ser protegidas (ex: incêndio, explosão, curto-circuito), quais os tipos de sensores responsáveis pela detecção dessas situações (ex: temperatura, chama, fumaça, corrente-elétrica), e as condições de monitoramento (ex: $t > 40^\circ$).

O monitoramento de condições físicas do ambiente corresponde a informações que são capturadas pelos sensores espalhados no ambiente, como temperatura, fumaça, chama e umidade. Esse tipo de monitoramento pode ser feito utilizando um dos tipos de subscrição definidos para a comunicação entre os serviços do *middleware* e as aplicações.

4.6 Considerações Finais

A implementação do modelo de subscrição e publicação é trabalho de outro aluno de mestrado. Vários mecanismos podem ser utilizados para que uma entidade participante possa se inscrever em um evento. Alternativas vão desde a utilização de plataformas do tipo CORBA até arquiteturas como JXTA da SUN.

5. Conclusões

Este trabalho apresentou a especificação de um *middleware* para monitoramento de condições críticas em ambientes físicos, um domínio de aplicação da computação ciente de contexto que tem recebido bastante atenção recentemente. Dentre os serviços que compõem esse *middleware*, este trabalho destaca: o Serviço de Interpretação de Contexto cuja responsabilidade é a interpretação de contextos provenientes de sensores dispersos em um ambiente físico de interação; o Serviço de Notificação de Eventos responsável por receber subscrições, registrá-las nos eventos desejados, encaminhá-las como subscrições à rede de sensores e gerar notificações quando as condições especificadas nas subscrições forem satisfeitas; e o Serviço de Localização que dependendo da tecnologia envolvida nos sensores de localização executam tarefas mais ou menos complexas para atender às subscrições referentes à localização de pessoas e/ou objetos. Destaca-se também a Interface de Configuração e Monitoramento por meio da qual o responsável pela segurança de uma empresa pode definir políticas de segurança específicas para a mesma, que por sua vez são utilizadas pelos serviços destacados acima.

A fundamentação teórica deste trabalho advém da necessidade de construção de *middlewares* cientes de contexto que atendam às necessidades de sistemas de monitoramento de condições críticas. A arquitetura do *middleware* proposto tem influência de trabalhos descritos na literatura, como Context Toolkit, Context Fabric, Gaia e iROS. A comunicação entre os serviços do *middleware* e destes com as aplicações dá-se segundo um mecanismo de *publish/subscribe* baseado em tópicos e as mensagens são trocadas no formato XML para fins de interoperabilidade.

5.1 Contribuições Geradas

O presente trabalho traz as seguintes contribuições:

1. Melhor compreensão das necessidades de um *middleware* de computação ubíqua e conhecimento de suas necessidades, como: a) separação de aquisição e interpretação de contextos; b) mecanismo eficaz de comunicação entre os serviços do *middleware* e as aplicações; c) forma de interpretação de contextos, assim como de localização;
2. Especificação de um *middleware* para aplicações de monitoramento de condições críticas de ambientes físicos de interação;
3. Especificação de um Serviço de Interpretação de Contexto baseado em lógica clássica e em um motor de interpretação de contexto a partir de dados provenientes de subscrições da aplicação;
4. Especificação de um Serviço de Notificação de Eventos Baseado em Tópicos;
5. Especificação de um Serviço de Localização que atende as três principais consultas de localização de pessoas e/ou objetos: Onde está pessoa X? Quem está na entidade SetorA? Tem alguém na entidade AlaB?

6. Especificação e implementação de uma interface simples de configuração e monitoramento de condições críticas pela qual o responsável de segurança define os critérios de monitoramento;
7. Especificação de interfaces XML para a comunicação entre os componentes do *middleware* e/ou aplicações.

5.2 Limitações e Trabalhos Futuros

- A utilização de lógica clássica no Serviço de Interpretação de Contexto pode ser considerada uma limitação, pois no monitoramento de condições críticas algumas situações não são nitidamente definidas e não podem ser precisamente descritas (ex: temperatura “alta”, “probabilidade” de incêndio). Na lógica clássica os objetos e/ou situações são classificados em categorias muito bem definidas. O problema é como definir exatamente que temperatura é considerada alta, ou quando existe a probabilidade de incêndio e com que grau de certeza? Uma solução como trabalho futuro seria a utilização de Lógica *Fuzzy* no Serviço de Interpretação de Contexto. Com a utilização da lógica *fuzzy* o responsável de segurança teria que definir alguns conjuntos ao invés de determinar apenas um valor máximo permitido.

Um exemplo seria: Grau de possibilidade de incêndio: baixo, médio, grande

Situação incêndio depende de:

- temperatura: baixa, normal, alta
- nível de fumaça: baixo, normal, alto
- nível de chama: baixo, normal, alto

Regra:

Se (temperatura = alta) e (nível_fumaca = alto) e (nível_chama = alto) então grau_probabilidade_incendio=alto

- Outra limitação é o fato da Interface de Configuração e Monitoramento ter sido feita de maneira simples e não muito interativa, pela qual o responsável de segurança deve

entrar com os dados de forma textual, o que pode ficar um pouco cansativo para usuários que não são do meio computacional. Como trabalho futuro é proposta a Construção de uma Interface Gráfica de Configuração e Monitoramento. Por meio dessa interface ficaria mais fácil, por exemplo, para o usuário entrar com os tipos de sensores existentes no ambiente. A idéia é: a partir de uma planta em 2D do ambiente, fornecer uma interface do tipo “arrasta e copia” pela qual o usuário definiria a existência e a localização de cada entidade e de cada sensor, inclusive o tipo de cada sensor;

Outros trabalhos futuros:

- Implementação e avaliação dos serviços de localização e interpretação de contexto quanto à funcionalidade, latência, confiabilidade;
- Implementação e avaliação do serviço de notificação e avaliação do mesmo quanto ao desempenho e entrega de eventos (taxa de perdas).

5.3 Conclusões

A computação ubíqua tem um grande potencial no monitoramento de condições críticas de ambientes físicos por permitir que dados sejam capturados dos ambientes de forma constante e minuciosa através de sensores e interpretados de forma a detectar situações de perigo. A computação ubíqua pode auxiliar também nas decisões que precisam ser tomadas de maneira rápida e eficaz quando situações anormais são detectadas, pois devido às aplicações de Captura e Acesso Automatizados os dados puros e/ou interpretados podem ser acessados em

tempo-real ou posteriormente para planejamento de estratégias de combate e salvamento, treinamento de equipe e perícias.

O presente trabalho é uma contribuição no sentido de explorar a utilização da computação ubíqua no monitoramento de condições críticas fornecendo contextos para as aplicações, de maneira independente de plataforma, dispositivo (PC, notebook, PDAs, celulares) e sistema operacional.

Referências Bibliográficas

- [ABO, 1997] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., Pinkerton, M. “Cyberguide: A mobile context-aware tour guide”. *Baltzer/ACM Wireless Networks*, p. 421-433, 1997.
- [ABO, 1999] Abowd, G.D. “Classroom 2000: An experiment with the instrumentation of a living educational environment”. *IBM Systems Journal*, v. 38, Outubro/1999.
- [ABO, 2000] Abowd, G.D.; Mynatt, E.D. “Charting Past, Present, and Future Research in Ubiquitous Computing”. *ACM Computer Human Interaction*, v.7, p.29-58, Março/2000.
- [AIM, 2003] AIM (2003). Radio Frequency Identification (RFID) home page. Disponível em <http://www.aimglobal.org/technologies/rfid/>. *The Association for Automatic Identification and Data Capture Technologies*.
- [ARN, 2001] Arnstein, L., Sigdursson, S., Franza, B. “Ubiquitous Computing in the Biology Laboratory”. *JALA Manuscript*, Arnstein, Outubro/2001.
- [BAN, 2002] Banavar, G, Bernstein, A. “Software Infrastructure and Design Challenges”. *Communications of the ACM*, v.45, n.12, pag.92-96, 2002.
- [BOR, 2001] Borthakur. B. “The Event Service”. *RFC - Request for Comments: 0003*.

Referências Bibliográficas

- University of Illinois at Urbana-Champaign, Outubro/2001.
- [BRO, 1997] Brown, P.J., Bovey J.D., Chen X. “Context-aware applications: From the laboratory to the marketplace”. *IEEE Personal Communications*, p.58-64. Outubro/1997.
- [BRO, 1998] Brotherton, J.A., Abowd, G.D., Truong, K.N. “Supporting Capture and Access Interfaces for Informal and Opportunistic Meetings”. *Georgia Institute of Technology Technical Report: GIT-GVU-99-06*, Atlanta, GA, 1998.
- [BRU, 2000] Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S. A. “Easyliving: Technologies for Intelligent Environments”. *Proceedings of the Handheld and Ubiquitous Computing Second International Symposium (HUC’2000)*, p.12-29, 2000.
- [BUR, 2002] Burrell, J., Gay, G. K., Kubo, K., Farina, N. “Context-aware computing: A test case”. *Proceedings of the International Conference on Ubiquitous Computing*, p.1-15, 2002.
- [CRU, 2004] Cruz, A. J. O. “Apostila de Lógica Nebulosa”. *Universidade Federal do Rio de Janeiro – Departamento de Ciência da Computação*. Disponível em: <http://equipe.nce.ufrj.br/adriano/fuzzy/bibliogr.htm>, último acesso: Abril/2004.
- [DAR, 2004] DARPA Information Exploitation Office, Command Post of the Future. Disponível em: <http://dtsn.darpa.mil/ixo/programdetail.asp?progid=18>, último acesso: Março/2004.
- [DEY, 2000] Dey, A.K. “Providing Architectural Support for Building Context-Aware

Referências Bibliográficas

- Applications”. *PhD Thesis*, Georgia Institute of Technology, 2001.
- [DEY, 2001] Dey, A.K. “Understanding and Using Context”. *Personal and Ubiquitous Computing*, p.4-7, 2001.
- [EUG, 2000] Eugster,P.T., Guerraoui,R., Sventek,J. "Distributed Asynchronous Collections: Abstractions for Publish/Subscribe Interaction?". *ECOOP 2000, LNCS 1850*, p.252-276, 2000.
- [EUG, 2003] Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M. “The Many Faces of Publish/Subscribe”. *ACM Computing Surveys*, p.114-131, Junho/2003.
- [FLE, 2002] Fleck, M., Frid, M., Kindberg, T., O’Brien-Strain, E., Rajani, R., Spasojevic, M. “From informing to remembering: Ubiquitous systems in interactive museums”. *IEEE Pervasive Computing*, p.13-21, 2002.
- [FME, 2004] Site da FMEA - FAILURE MODES AND EFFECTS ANALYSIS americana. Disponível em: <http://www.fmeainfocentre.com/>, último acesso Abril/2004.
- [FU, 2001] Fu, X., Shi, W., Akkerman, A., Karamcheti, V. “CANS: Composable, Adaptive Network Services Infrastructure”. *USENIX Symposium on Internet Technologies and Systems – USIT*, Março/2001.
- [HEN, 2002] Henricksen,K., Indulska,J., Rakotonirainy, A. “Modeling context information in pervasive computing systems”. *Proceedings of Pervasive 2002: Pervasive Computing: First International Conference, volume 2414 - Lecture Notes in Computer Science - Springer Verlag.*, p. 167-180, Zürich, Switzerland, Agosto/2002.

Referências Bibliográficas

- [HER, 2002] Herrtwich, R. G. “Ubiquitous Computing in the Automotive Domain”. *Proceedings of the Pervasive Computing - First International Conference*, p. 15, Agosto/2002.
- [HES, 2002] Hess, C.K., Roman, M., Campbell, R.H., “Building Applications for Ubiquitous Computing Environments”. *Proceedings of the Pervasive Computing – First International Conference*, p. 16-29, Agosto/2002.
- [HON, 2001a] Hong, J.I., Landay, J.A. “An infrastructure approach to context-aware computing”. *Human-Computer Interaction*, p.287-303, 2001.
- [HON, 2001b] Hong, J.I. “Context Fabric: Infrastructure Support for Context-Aware Systems”. *Phd Thesis*, Universidade da Califórnia – Berkeley, 2001.
- [HÜR, 1999] Hürst, W., Müller, R. “A Synchronization Model for Recorded Presentations and its Relevance for Information Retrieval”. *Proceedings of ACM Multimedia*, 1999.
- [JIA, 2002] Jiang, X. , Landay, J.A. “Modeling Privacy Control in Context-aware Systems Using Decentralized Information Spaces”. *IEEE Pervasive Computing*, v.1, n.3, 2002.
- [JIA, 2004a] Jiang, X., Chen, N.Y., Hong, J., Wang K., Takayama, L., Landay, J.A. “Siren: Context-aware Computing for Firefighting”. *Submetido ao Pervasive’ 2004*, 2004.
- [JIA, 2004b] Jiang, X., Hong, J., Takayama, L.A., Landay, J.A. “Ubiquitous Computing for Firefighters: Field Studies and Prototypes of Large Displays for Incident Command”. *CHI’2004*.
- [JOH, 2002] Johanson, B., Fox, A., Winograd, T. “The Interactive Workspaces Project:

Referências Bibliográficas

- Experiences with Ubiquitous Computing Rooms”. *Pervasive Computing Magazine Special Issue on Systems*, 2002.
- [JOH, 2002a] Johanson, B. and A. Fox. “The Event Heap: An Coordination Infrastructure for Interactive Workspaces”. *Proceedings of the 4th IEEE Workshop on Mobile Computer Systems and Applications (WMCSA-2002)*. Callicoon, New York, USA. 2002
- [KIC, 2000] Kiciman, E., Fox, A. “Using Dynamic Mediation to Integrate COTS Entities in a Ubiquitous Computing Environment”. *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K)*, Alemanha, 2000.
- [KID, 1999] Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., Macintyre, B., Mynatt, E., Starner, T. E., Newstetter, W. “The Aware Home: A Living Laboratory for Ubiquitous Computing Research”. *Proceedings of the Second International Workshop on Cooperative Buildings - CoBuild*, 1999.
- [KIN, 2000] Kindberg T. et al. “People, Places, Things: Web Presence for the Real World”. *Proceedings of the Third Workshop Mobile Computing Systems and Applications*, IEEE CS Press. P.19-28, Los Alamitos, Calif., 2000.
- [KIN, 2001] Kindberg, T., Barton, J. “A Web-based Nomadic Computing System”. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, v.35, nº 4, p. 443-456, Março/2001.
- [KIN, 2002] Kindberg, T., Fox, A. “System Software for Ubiquitous Computing”. *IEEE Pervasive Computing*, v.1, n.1, pag.70-81, 2002.
- [KNI, 2002] Knight, J.C. “Safety Critical Systems: Challenges and Directions”. *ACM -*

Referências Bibliográficas

- ICSE'02*, p.547-550, Florida, Maio/2002.
- [KUD, 2003] Kudo, T. N., Micheloti, G. C., Pazzi, R. W. N., Araujo, R. B. “Visualização Situacional: A Intersecção das Tecnologias de Computação Ubíqua e Realidade Virtual para o Mapeamento de Experiências do Mundo Real no Mundo Virtual”. *WEBMIDIA - Simpósio Brasileiro de Sistemas Multimídia e Web*, Salvador – BA, 2003.
- [LAM, 2004] Lampe, M., Strassner, M. , Fleisch, E. “A Ubiquitous Computing Environment for Aircraft Maintenance”. *ACM - SAC'04*, p. 14-17, Nicosia, Cyprus, Março/2004.
- [LYY, 2002] Lyytinen, K., Yoo, Y. “Issues and Challenges in Ubiquitous Computing”. *Communications of the ACM*, p.62-65, Dezembro/2002.
- [MAO, 2002] Mao, Z. M., Katz R.H. “Achieving Service Portability using Self-adaptive Data Paths”. *IEEE Communications Magazine special Issue on Service Portability and Virtual Home Environment*, Janeiro/2002.
- [MIC, 2003] Micheloti, G. C., Kudo, T. N., Pazzi, R. W. N., Araujo, R. B. “Um Sistema de Captura e Acesso para Ambientes de Segurança Crítica”. *Workshop em Segurança (Safety Workshop): Sistemas Computacionais em Aplicações Críticas (em conjunto com o First Latin-American Symposium on Dependable Computing)*, São Paulo-SP, 2003.
- [MOZ, 1998] Mozer, M. C. “The Neural Network House: An Environment that Adapts to its Inhabitants”. *In American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, p.110-114, 1998.
- [MUK, 1999] Mukhopadhyay, S., Smith, B. “Passive Capture and Structuring of Lectures”.

Referências Bibliográficas

- Proceedings of ACM Multimedia*, 1999.
- [MYN, 1998] Mynatt, E.D., Back, M., Want, R., Baer, M., Ellis, J.B. “Designing audio aura”. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 566-573, Abril/1998.
- [MYN, 2001] Mynatt, E.D., Rowan, J., Craighill, S. and Jacobs, A. (2001). "Digital family portraits: Providing peace of mind for extended family members". *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 333-340.
- [PAL, 1998] Palanque, P., Paternò, F., Fields, B. “Designing User Interfaces for Safety Critical Systems”. *SIGCHI’98 - CHI 98 Workshop*, Outubro/1998.
- [POL, 2004] Site da Polícia Militar do Estado de São Paulo – Corpo de Bombeiros. Disponível em: <http://www.polmil.sp.gov.br/ccb/pagina15.html>, último acesso Abril/2004.
- [PON, 2003] Ponnekanti, S., Johanson, B., Kiciman, E. Fox, A. “Portability, Extensibility and Robustness in iROS”. IEEE International Conference on Pervasive Computing and Communications, 2003.
- [POO, 2002] Poor, R., Hodges, B. “Reliable Wireless Networks for Industrial Systems”. *Technical Report Ember Corporation*, Boston, Massachusetts, USA, 2002.
- [REZ, 2003] Rezende S.O. “Sistemas Inteligentes: Fundamentos e Aplicações”. Ed. Manole, p.169-201, Barueri-SP, 2003.
- [RIC, 2001] Richter, H., et al. “Integrating Meeting Capture within a Collaborative Team

Referências Bibliográficas

- Environment”. *Proceedings of the International Conference on Ubiquitous Computing (UbiComp-2001)*, Atlanta, GA, 2001.
- [ROM, 2002] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt K. “A middleware infrastructure for Active Spaces”. *IEEE Pervasive Computer*, v.1, n.4, p. 74-83, Outubro-Dezembro/2002.
- [RYA, 1998] Ryan N., Pascoe J., Morse D. “Enhanced reality fieldwork: the context-aware archaeological assistant”. *Computer Applications and Quantitative Methods in Archaeology*. Em Gaffney, Leusen e Exxon, Editors, *Computer Applications in Archeology*.
- [SAL, 1999] Salber, D., Dey, A.K., Abowd, G.D. “The Context Toolkit: Aiding the Development of Context-Enabled Applications”. *CHI’99*, ACM Press, p.434-441, Pittsburgh, PA, Maio/1999.
- [SCH, 1993] Schilit, B. N., Adams, N., Gold, R., Tso, M., and Want, R. “The ParcTab mobile computing system”. *Proceedings of the Workshop on Workstation Operating Systems*, p.34-39, 1993.
- [SCH, 1994] Schilit, B.N., Theimer M.M. “Disseminating active map information to mobile hosts”. *IEEE Network*, p.22-32. Setembro-Outubro/1994.
- [SCH, 1995] Schilit, B., Roy, W. “The Xerox PARCTAB”. Disponível em: <http://sandbox.parc.xerox.com/parctab/>, 1995.
- [SCH, 1998] Schilit, B.N., Golovchinsky, G., Price, M.N. “Beyond Paper: Supporting Active Reading with Free form Digital Ink Annotations”. *In Proceedings of the Conference on Human Factors in Computing Systems*, p.249-256, 1998.
- [STA, 2002] Stajano, F. “Security for Ubiquitous Computing”. *Wiley Series in*

Referências Bibliográficas

Communications Networking & Distributed Systems. John Wiley and Sons, Ltd. Fevereiro/2002.

- [WAN, 1992] Want, R., Hopper, A., Falcão, V., Gibbons, J. “The Active Badge Location System”. *ACM Transactions on Information Systems*, p.91-102, Janeiro/1992.
- [WEI, 1991] Weiser, M. “The Computer for the 21st Century”. *Scientific American*, New York, p.94-104, Setembro/1991.
- [WEI, 1996] Weiser, M., Brown, J. “The Coming Age of Calm Technology”. *Xerox PARC*, Outubro/1996.
- [WIT, 1998] Witbrock, M.J., Hauptmann, A.G. “Speech Recognition for a Digital Video Library”. *Journal of the American Society of Information Science*, 49(7), p.619-632, 1998.
- [XML, 2004] Site do W3C (XML - *Extensible Markup Language*). Disponível em: <http://www.w3.org/XML/>, último acesso Abril/2004.

Apêndice A

Interface de Configuração e Monitoramento

Neste apêndice são apresentadas as telas da Interface de Configuração e Monitoramento do Sistema. As mesmas telas são apresentadas com mais detalhes e com exemplos no **Apêndice B**.

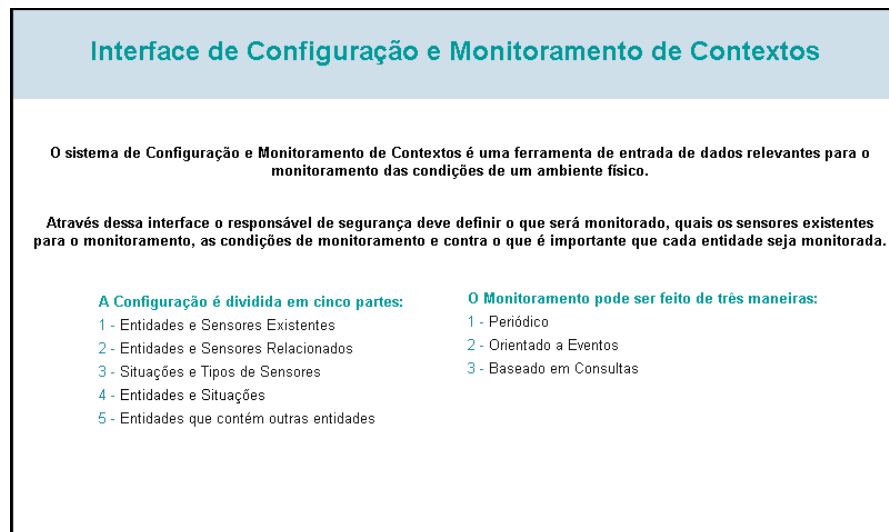


Figura A. 1 - Interface Inicial

A configuração do sistema de monitoramento é dividida em cinco partes:

- entidades e sensores existentes (**Figura A-2, Figura A-3 e Figura A-4**)
- entidades e sensores relacionados (**Figura A-5**)
- situações e tipos de sensores (**Figura A-6**)
- entidades e situações (**Figura A-7**)

- entidades que contém outras entidades (Figura A-8)

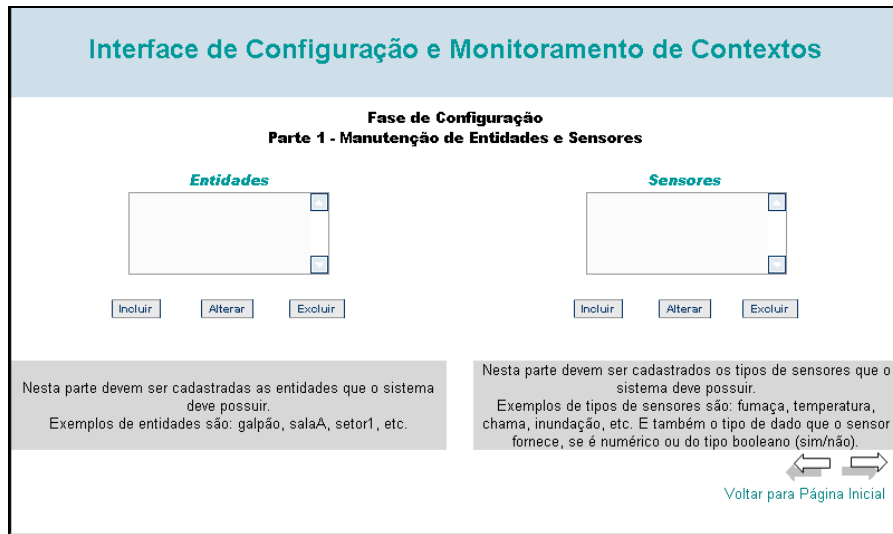


Figura A. 2 - Parte 1 – Manutenção de Entidades e Sensores

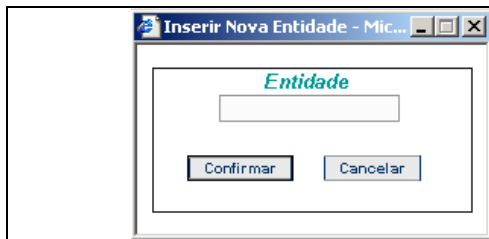


Figura A. 3 - Interface de Inserção de Entidades



Figura A. 4 - Interface de Inserção de Sensores



Figura A. 5 - Parte 2 – Entidades e Sensores

Interface de Configuração e Monitoramento de Contextos

Fase de Configuração
Parte 3 - Situações e Tipos de Sensores

Situação

Escolha os tipos de sensores

.....

 todos

As situações que serão monitoradas no ambiente devem ser especificadas e também os sensores responsáveis pelo monitoramento.

[Voltar para Página Inicial](#)

Figura A. 6 - Parte 3 – Situações e Tipos de Sensores

Interface de Configuração e Monitoramento de Contextos

Fase de Configuração
Parte 4 - Entidades e Situações a serem monitoradas

Escolha uma Entidade

Escolha a(s) situação(ões)

.....

 Todas

Cada entidade deve ser relacionada com as situações das quais devem ser protegidas.
Ex: Setor1: incêndio e explosão.

[Voltar para Página Inicial](#)

Figura A. 7 - Parte 4 – Entidades e Situações a serem monitoradas

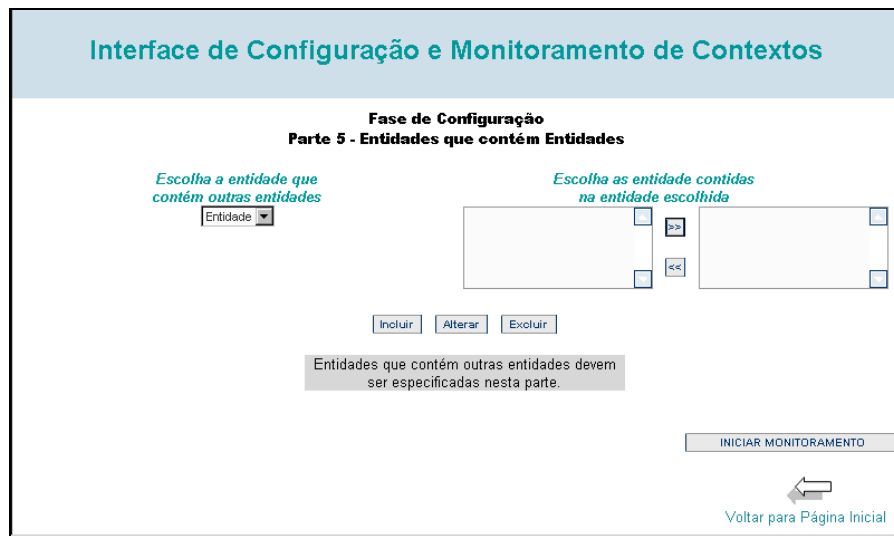


Figura A. 8 - Parte 5 – Entidades que contém outras entidades

O monitoramento pode ser feito de três maneiras:

- periódico (**Figura A-9**)
- orientado a eventos (**Figura A-10**)
- baseado em consultas (**Figura A-11**)

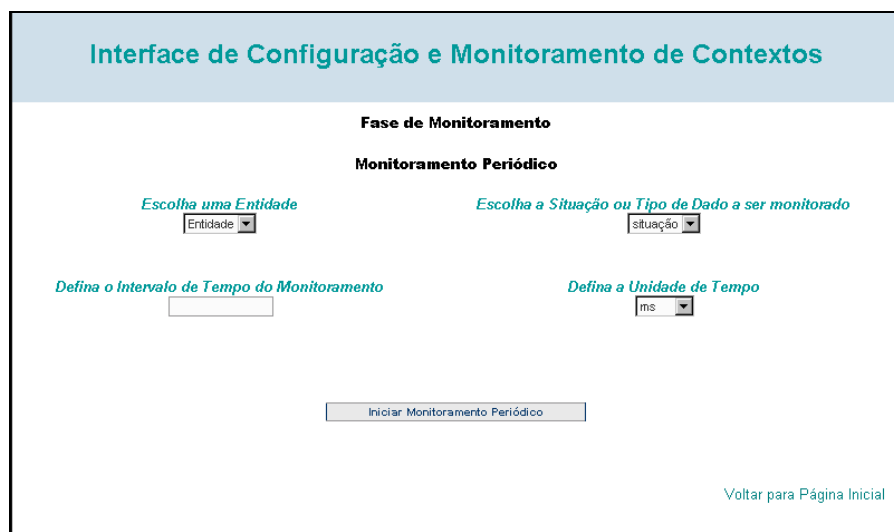


Figura A. 9 - Monitoramento Periódico

The screenshot shows a web interface titled "Interface de Configuração e Monitoramento de Contextos". Under the heading "Fase de Monitoramento", the sub-heading is "Monitoramento Orientado a Eventos". There are two dropdown menus: "Escolha uma Entidade" with a label "Entidade" and "Escolha um Evento (situação)" with a label "Evento". Below these is a button labeled "Iniciar Monitoramento Orientado a Eventos". In the bottom right corner, there is a link that says "Voltar para Página Inicial".

Figura A. 10 - Monitoramento Orientado a Eventos

The screenshot shows a web interface titled "Interface de Configuração e Monitoramento de Contextos". Under the heading "Fase de Monitoramento", the sub-heading is "Monitoramento Baseado em Consulta". There are two dropdown menus: "Escolha uma Entidade" with a label "Entidade" and "Escolha a Situação ou Tipo de Dado a ser monitorado" with a label "Situação". Below these is a button labeled "Iniciar Monitoramento Baseado em Consulta". In the bottom right corner, there is a link that says "Voltar para Página Inicial".

Figura A. 11 - Monitoramento Baseado em Consulta

Apêndice B

Estudo de Caso: Museu de Aviação Asas de um Sonho – TAM

1. Interfaces de Configuração

A configuração do sistema de monitoramento do Museu é feita com base nos dados fornecidos pelo responsável de segurança através da Interface de Configuração e Monitoramento.

A **Figura B.1** mostra a tela principal da Interface de Configuração e Monitoramento onde encontra-se os *links* para as fases de configuração e para as opções de monitoramento.

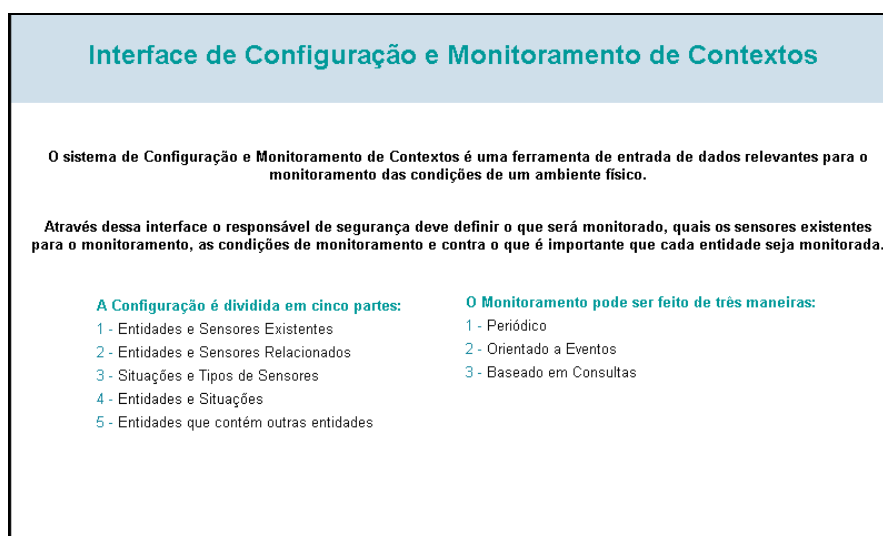


Figura B. 1 - Interface Inicial

Através da segunda tela (**Figura B.2**) é feita a entrada dos dados das entidades que deverão ser monitoradas e os tipos de sensores existentes no ambiente.

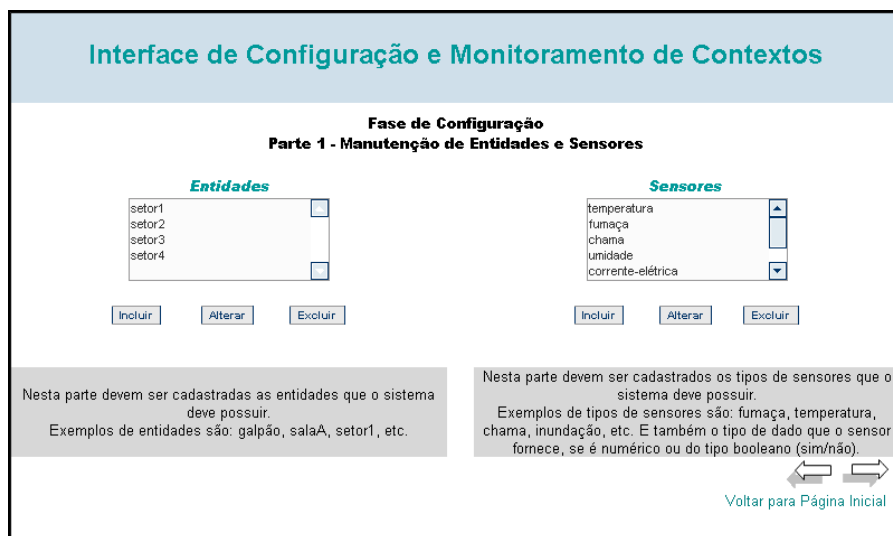


Figura B. 2 - Interface de Manutenção de Entidades e Sensores

A inclusão de entidades e sensores são feitas pelas interfaces apresentadas na **Figura B.3** e **Figura B.4** respectivamente.

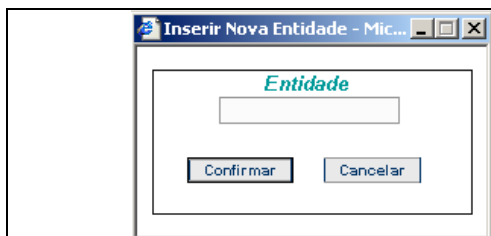


Figura B. 3 - Interface de Inserção de Entidades

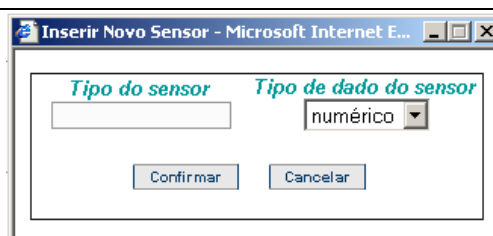


Figura B. 4 - Interface de Inserção de Sensores

No caso do Museu de Aviação as entidades e os tipos de sensores são (**Tabela B.1**):

Tabela B. 1 - Entidades e Tipos de Sensores do Museu

Entidades	Sensores
Setor1	Temperatura
Setor2	Fumaça
Setor3	Chama
Setor4	Umidade
14 BIS (Réplica)	Corrente-elétrica
Aeronca 7AC (1950)	Inundação
Aeronca C3 (1936)	Pressão
Aeronca Chief (1951)	
American Flea Ship (1939)	

Apêndice B – Estudo de Caso

Bucker Jungman 1 (casa) (1960)
Bucker Jungman 2 (casa) (1960)
Cap 4 (1948)
Cessna (l-19) bird dog (1952)
Cessna (l-19) bird dog (1953)
Cessna (l-19) bird dog (1953)
Cessna 140-a (1949)
Cessna 170-b (1951)
Cessna 180-c (1963)
Cessna 195b (1950)
Cessna 210 (1957)
Curtiss Robin B (1928)
De Havilland Chipmunk (1951)
De Havilland Dove (1950)
Dmoiselle 9 (réplica)
Douglas dc-8 (1963)
Emb-110-c Bandeirantes (1971)
F4u-1 Corsair (1943)
Fairchild f24 (1944)
Fairchild pt-19 (1944)
Fleet ii (1931)
Focke-Wulf fw-44j (1943)
Focke-Wulf 190
Girocôptero Bensen (1960)
Grumman p-16 Tracker (1954)
l-049 - Constellation (1946)
Messerschmitt me-109-g
Mig 15 (1953)
Mig 17 (1959)
Mig 21 (1962)
Miles Hawk (1936)
NA - t6 - D Texan (1941)
Nord Norécrin (1948)
Piper pa 12 (1952)

Através da tela apresentada na **Figura B.5** é feito o relacionamento entre as entidades e os sensores que monitoram essa entidade. Nesse momento o n° serial de cada sensor deve ser fornecido.

Interface de Configuração e Monitoramento de Contextos

Fase de Configuração
Parte 2 - Entidades e Sensores

Escolha uma Entidade Escolha um tipo de Sensor Nº Serial:

setor1 temperatura

Todos os sensores relacionados a cada entidade devem ser especificados com o nº serial de cada sensor.

[Voltar para Página Inicial](#)

Figura B. 5 - Interface de Entidades e Sensores

As entidades, os tipos de sensores e o nº serial dos sensores referentes ao museu são apresentados na **Tabela B.2**:

Tabela B. 2 - Entidades - Tipo de sensor - N.Serial

Entidade	Tipo do Sensor	Nº Serial
Setor1	temperatura	1212312
Setor1	temperatura	4545579
Setor1	temperatura	0239387
Setor1	fumaça	6626372
Setor1	chama	21312321
Setor1	pressão	532874
Setor2	temperatura	621536
Setor2	temperatura	048869
Setor2	fumaça	6457438
Setor2	fumaça	0381928
Setor2	chama	6473278
Setor2	chama	6473900
Setor3	temperatura	468697
Setor3	fumaça	7438989

Apêndice B – Estudo de Caso

Setor3	chama	54546566
Setor3	corrente-elétrica	63246778
Setor4	temperatura	6237432
Setor4	fumaça	8435843
Setor4	chama	6327328

Por meio da tela quatro (**Figura B.6**) devem ser definidas as situações que deverão ser monitoradas e os tipos de sensores responsáveis por monitorar cada situação.

Interface de Configuração e Monitoramento de Contextos

Fase de Configuração
Parte 3 - Situações e Tipos de Sensores

Situação

Escolha os tipos de sensores

- temperatura
- fumaça
- chama
- umidade
- corrente-elétrica
- inundação
- pressão
- todos

Incluir Alterar Excluir

As situações que serão monitoradas no ambiente devem ser especificadas e também os sensores responsáveis pelo monitoramento.

[Voltar para Página Inicial](#)

Figura B. 6 - Interface de definição de situações a serem monitoradas e tipos de sensores

Através da tela apresentada na **Figura B.7** devem ser definidas as entidades e contra quais situações elas devem ser monitoradas.

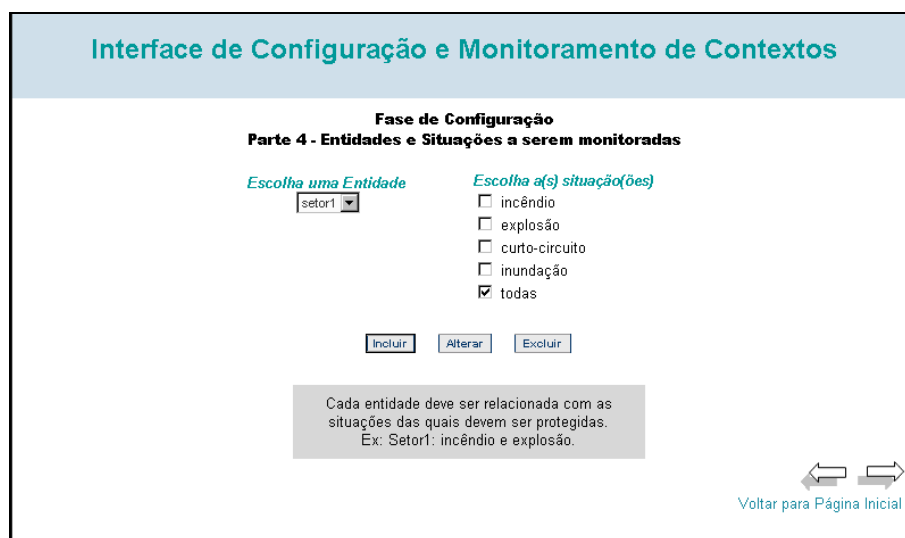


Figura B. 7 - Interface de definição de entidades e situações a serem monitoradas

As situações que devem ser monitoradas no Museu são apresentadas na **Tabela B.3**:

Tabela B. 3 - Tabela de entidades - situações e sensores

Entidade	Situação	Tipo de Sensores
Setor 1	Incêndio	temperatura>40, fumaça, chama
Setor 2	Incêndio	temperatura>40, fumaça, chama
Setor 3	Incêndio	temperatura>40, fumaça, chama
Setor 4	Incêndio	temperatura>40, fumaça, chama
Setor 1	Explosão	chama, pressão>100
Setor 3	Curto-circuito	corrente elétrica>100

Através da tela apresentada na **Figura B.8** devem ser definidas as entidades que contém outras entidades, por exemplo Setor1 contém aeronave 14 BIS (Réplica), como pode ser visto na **Tabela B.4**. Pois, quando uma entidade estiver em situação de emergências, as entidades que ela contém também deverão ser monitoradas.

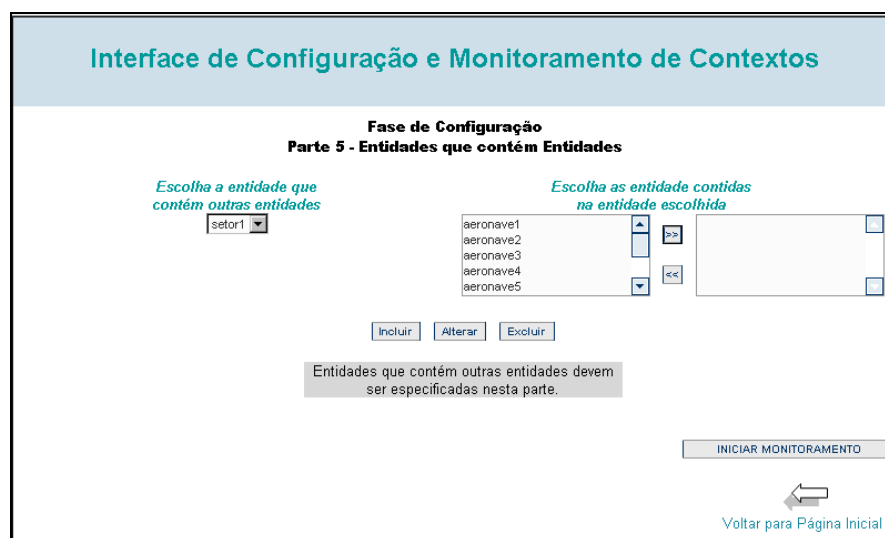


Figura B. 8 - Interface de definição de entidades que contém outras entidades

As entidades que contém outras entidades no museu são:

Tabela B. 4 - Tabela de entidades que contém outras entidades

Entidade que contém	Entidades que estão contidas em outras entidades
Setor1	14 BIS (Réplica) Aeronca 7AC (1950) Aeronca C3 (1936) Aeronca Chief (1951) American Flea Ship (1939) Bucker Jungman 1 (casa) (1960)
Setor2	Bucker Jungman 2 (casa) (1960) Cap 4 (1948) Cessna (1-19) bird dog (1952) Cessna (1-19) bird dog (1953) Cessna (1-19) bird dog (1953) Cessna 140-a (1949) Cessna 170-b (1951) Cessna 180-c (1963)
Setor3	Cessna 195b (1950) Cessna 210 (1957) Curtiss Robin B (1928) De Havilland Chipmunk (1951) De Havilland Dove (1950) Dmoiselle 9 (réplica) Douglas dc-8 (1963) Emb-110-c Bandeirantes (1971) F4u-1 Corsair (1943) Fairchild f24 (1944)

	Fairchild pt-19 (1944) Fleet ii (1931) Focke-Wulf fw-44j (1943)
Setor4	Focke-Wulf 190 Girocóptero Bensen (1960) Grumman p-16 Tracker (1954) I-049 - Constellation (1946) Messerschmitt me-109-g Mig 15 (1953) Mig 17 (1959) Mig 21 (1962) Miles Hawk (1936) NA - t6 - D Texan (1941) Nord Norécrin (1948) Piper pa 12 (1952)

2. Interfaces de Monitoramento

O Monitoramento do ambiente pode ser feito de três maneiras (**Figura B.9**, **Figura B.10**, **Figura B.11**):

- Periódico
- Orientado a eventos
- Baseado em consulta

Interface de Configuração e Monitoramento de Contextos

Fase de Monitoramento

Monitoramento Periódico

Escolha uma Entidade

Escolha a Situação ou Tipo de Dado a ser monitorado

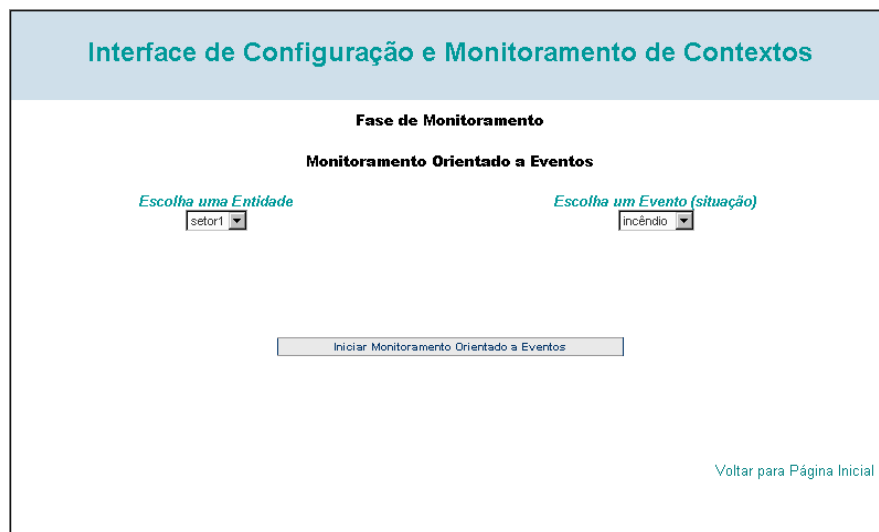
Defina o Intervalo de Tempo do Monitoramento

Defina a Unidade de Tempo

[Voltar para Página Inicial](#)

Figura B. 9 - Interface de Monitoramento Periódico

Para o monitoramento periódico (**Figura B.9**) o usuário deve especificar: a entidade de interesse, a situação (ex: incêndio, explosão) ou o tipo de dado (ex: temperatura, pressão), o intervalo de tempo (ex: 10) e a unidade de tempo (ex: segundos, minutos).



A interface de configuração e monitoramento de contextos, intitulada "Interface de Configuração e Monitoramento de Contextos", apresenta a seguinte estrutura:

- Fase de Monitoramento**
 - Monitoramento Orientado a Eventos**
 - Dois menus suspensos para seleção:
 - Escolha uma Entidade*: com o valor "setor1" selecionado.
 - Escolha um Evento (situação)*: com o valor "Incêndio" selecionado.
 - Um botão centralizado com o texto "Iniciar Monitoramento Orientado a Eventos".
 - Um link no canto inferior direito com o texto "Voltar para Página Inicial".

Figura B. 10 - Interface de Monitoramento Orientado a Eventos

Para o monitoramento orientado a eventos (**Figura B.10**) o usuário deve especificar: a entidade e o evento (situação) de seu interesse.

The screenshot displays a web interface titled "Interface de Configuração e Monitoramento de Contextos". Under the heading "Fase de Monitoramento", there is a sub-section "Monitoramento Baseado em Consulta". This section contains two dropdown menus: "Escolha uma Entidade" with the value "setor1" and "Escolha a Situação ou Tipo de Dado a ser monitorado" with the value "incêndio". Below these is a button labeled "Iniciar Monitoramento Baseado em Consulta". In the bottom right corner, there is a link that says "Voltar para Página Inicial".

Figura B. 11 - Interface de Monitoramento Baseado em Consulta

Para o monitoramento baseado em consulta (**Figura B.11**) o usuário deve especificar: a entidade e a situação (ex: incêndio, explosão) ou o tipo de dado (ex: temperatura, pressão) de seu interesse.

A diferença entre os três tipos de monitoramento é na parte de notificação de eventos. No monitoramento periódico, a notificação é feita obedecendo ao intervalo de tempo especificado, no monitoramento orientado a eventos, a notificação é feita quando as condições especificadas forem atendidas, e no monitoramento baseado em consulta a notificação é feita independente de tempo ou condições.

3. Subscrições ao Serviço de Interpretação

Em situações de monitoramento, que envolve o processamento de expressões lógicas, o serviço de interpretação de contexto é acionado. A **Figura B.12** mostra a descrição do esquema XML correspondente à subscrição de monitoramento para o serviço de interpretação.


```
<subscribe_IS>
  <subscription_id></subscription_id>
  <entity_type></entity_type>
  <entity_value></entity_value>
  <situation></situation>
  <expression></expression>
</subscribe_IS>
```

Figura B. 12 - Formato de uma subscrição ao Serviço de Eventos

Exemplos de subscrições ao serviço de eventos neste estudo de caso são:

```
<subscribe_IS>
  <subscription_id>1234</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor1</entity_value>
  <situation>incendio</situation>
  <expression>temperatura>40 AND fumaca AND chama</expression>
</subscribe_IS>
```

Figura B. 13 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 1

```
<subscribe_IS>
  <subscription_id>154</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor2</entity_value>
  <situation>incendio</situation>
  <expression>temperatura>40 AND fumaca AND chama</expression>
</subscribe_IS>
```

Figura B. 14 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 2

```
<subscribe_IS>
  <subscription_id>174</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor3</entity_value>
  <situation>incendio</situation>
  <expression>temperatura>40 AND fumaca AND chama</expression>
</subscribe_IS>
```

Figura B. 15 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 3

```
<subscribe_IS>
  <subscription_id>4174</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor4</entity_value>
  <situation>incendio</situation>
  <expression>temperatura>40 AND fumaca AND chama</expression>
</subscribe_IS>
```

Figura B. 16 - Exemplo de XML para subscrição orientada a eventos – incêndio no setor 4

```
<subscribe_IS>
  <subscription_id>1231</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor1</entity_value>
  <situation>explosao</situation>
  <expression>pressao>100 AND chama</expression>
</subscribe_IS>
```

Figura B. 17 - Exemplo de XML para subscrição orientada a eventos – explosão no setor1

```
<subscribe_IS>
  <subscription_id>151</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor3</entity_value>
  <situation>curto-circuito</situation>
  <expression>corrente_eletrica>150</expression>
</subscribe_IS>
```

Figura B. 18 - Exemplo de XML para subscrição orientada a eventos – curto-circuito no setor3

4. Subscrições ao Serviço de Eventos

A partir dos dados fornecidos pelo engenheiro de segurança do museu, as subscrições ao serviço de eventos são criadas. O formato da subscrição ao serviço de eventos é (Figura B.19):

```
<subscribe>
  <subscription_type></subscription_type>
  <subscription_id></subscription_id>
  <entity_type></entity_type>
  <entity_value></entity_value>
  <sensor_type></sensor_type>
  <text></text>
  <value></value>
</subscribe>
```

Figura B. 19 - Formato de uma subscrição ao Serviço de Eventos

As subscrições são diferenciadas para cada tipo de monitoramento: Periódico, Orientado a Eventos e Baseada em Consulta, segue com mais detalhes:

4.1 Subscrição Periódica

Exemplos de subscrições para **monitoramento periódico** neste estudo de caso são:

```
<subscribe>
  <subscription_type>1</subscription_type>
  <subscription_id>111</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor1</entity_value>
  <sensor_type>temperatura</sensor_type>
  <text>minutos</text>
  <value>10</value>
</subscribe>
```

Figura B. 20 - Exemplo de XML para subscrição periódica – temperatura do setor 1 a cada 10 minutos

Nesta subscrição *1* indica que ela é do tipo periódica, *111* é o id da subscrição, *ambiente* é o tipo de entidade, *setor1* é a entidade de interesse, *temperatura* é o tipo de dado de interesse, *minutos* representa que a unidade de tempo a ser atendida é minutos, e *10* é o intervalo de tempo. Portanto a notificação sobre a temperatura do setor1 seria feita a cada 10 minutos.

4.2 Subscrição Orientada a Eventos

Exemplos de subscrições para **monitoramento orientado a eventos** neste estudo de caso:

```
<subscribe>
  <subscription_type>2</subscription_type>
  <subscription_id>251</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor1</entity_value>
  <sensor_type>temperatura</sensor_type>
  <text>></text>
  <value>60</value>
</subscribe>
```

Figura B. 21 - Exemplo de XML para subscrição orientada a eventos – temperatura do setor1, quando temperatura>60

Nesta subscrição, 2 indica que ela é do tipo baseada em consulta, 251 é o id da subscrição, *ambiente* é o tipo de entidade, *setor1* é a entidade de interesse, *temperatura* é o tipo de dado a ser consultado.

4.3 Subscrição Baseada em Consulta

Exemplos de subscrições para **monitoramento baseado em consulta** neste estudo de caso:

```
<subscribe>
  <subscription_type>3</subscription_type>
  <subscription_id>25</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>setor1</entity_value>
  <sensor_type>temperatura</sensor_type>
  <text></text>
  <value></value>
</subscribe>
```

Figura B. 22 - Exemplo de XML para subscrição baseada em consulta – Qual a temperatura do setor1?

Nesta subscrição, 3 indica que ela é do tipo baseada em consulta, 25 é o id da subscrição, *ambiente* é o tipo de entidade, *setor1* é a entidade de interesse, *temperatura* é o tipo de dado a ser consultado.

5. Subscrições ao Serviço de Localização

Basicamente, três consultas podem ser feitas sobre a localização de pessoas e/ou objetos em ambientes:

4. Onde está pessoa X?
5. Quem está no ambiente X?
6. Tem alguém no ambiente Y?

Estas consultas podem ser feitas pelas aplicações ao serviço de localização do *Middleware*. A **Figura B.23** mostra a descrição do esquema XML correspondente à subscrição de consultas de localização para o serviço de localização.

```
<subscribe_LS>
  <subscription_id></subscription_id>
  <entity_type></entity_type>
  <entity_value></entity_value>
  <situation></situation>
</subscribe_LS>
```

Figura B. 23 - Esquema XML de subscrição para Localização

Para cada uma dessas consultas apresenta-se um exemplo nas **Figuras B.24, B.25 e B.26**:

```
<subscribe_LS>
  <subscription_id>111</subscription_id>
  <entity_type>pessoa</entity_type>
  <entity_value>pessoaX</entity_value>
  <situation>ONDE</situation>
</subscribe_LS>
```

Figura B. 24 - Esquema XML de subscrição para Localização - Onde está pessoa X?

```
<subscribe_LS>
  <subscription_id>568</subscription_id>
  <entity_type>ambiente</entity_type>
  <entity_value>ambienteX</entity_value>
  <situation>QUEM</situation>
</subscribe_LS>
```

Figura B. 25 - Esquema XML de subscrição para Localização - Quem está no ambiente X?

```
<subscribe_LS>  
  <subscription_id>68</subscription_id>  
  <entity_type>ambiente</entity_type>  
  <entity_value>ambienteY</entity_value>  
  <situation>PRESENÇA</situation>  
</subscribe_LS>
```

Figura B. 26 - Esquema XML de subscrição para Localização - Tem alguém no ambiente Y?