

**LUIS GUSTAVO NARDIN**

**USO DE SISTEMAS MULTIAGENTES EM  
SISTEMAS SENSÍVEIS AO CONTEXTO**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para conclusão do curso de  
Especialização em Engenharia de  
Software

São Paulo  
2005

**LUIS GUSTAVO NARDIN**

**USO DE SISTEMAS MULTIAGENTES EM  
SISTEMAS SENSÍVEIS AO CONTEXTO**

Monografia apresentada à Escola  
Politécnica da Universidade de São  
Paulo para conclusão do curso de  
Especialização em Engenharia de  
Software

Área de Concentração:  
Engenharia de Software

Orientador:  
Prof. Dr. Jaime Simão Sichman

São Paulo

2005

## **FICHA CARTOGRÁFICA**

Nardin, Luis Gustavo

Uso de Sistemas Multiagentes em Sistemas Sensíveis ao Contexto. São Paulo, 2005. 55p.

Dissertação (MBA) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia.

1. Sistemas Multiagentes 2. Sistemas Sensíveis ao Contexto I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais.

## **AGRADECIMENTOS**

Ao meu orientador, Prof. Dr. Jaime Simão Sichman, por suas diretrizes claras e objetivas.

À minha família, por ter me apoiado no avanço de mais uma etapa da vida.

## **RESUMO**

O paradigma de agentes e os sistemas multiagentes vêm se apresentando como alternativa de grande potencial no desenvolvimento de sistemas sensíveis ao contexto. Esta monografia apresenta uma noção de contexto, uma visão geral de sistemas sensíveis ao contexto e um possível emprego de técnicas de sistemas multiagentes na arquitetura de sistemas sensíveis ao contexto. Este trabalho é completado com a apresentação de um possível uso de sistemas multiagentes em sistemas sensíveis ao contexto, aplicado a uma academia de ginástica.

## **ABSTRACT**

Software agents and multi-agent systems technologies can be used in the development of context-aware systems. This work presents a notion of context, a general overview of context-aware systems and some possible uses of multi-agent systems technologies on context-aware systems design. This work is concluded showing a possible use of multi-agent systems technologies on context-aware system design, applied to a Fitness Center.

## SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

1.	INTRODUÇÃO .....	1
1.1.	OBJETIVO .....	3
1.2.	MOTIVAÇÃO .....	3
1.3.	METODOLOGIA .....	4
1.4.	ORGANIZAÇÃO DO DOCUMENTO.....	4
2.	ELEMENTOS DE BASE.....	5
2.1.	CONTEXTO .....	5
2.1.1.	Definição .....	5
2.1.2.	Categorização .....	7
2.1.3.	Aquisição .....	8
2.1.4.	Modelos de Representação.....	9
2.2.	SISTEMAS SENSÍVEIS AO CONTEXTO.....	12
2.2.1.	Definição .....	12
2.2.2.	Categorização .....	12
2.2.3.	Arquitetura .....	13
2.3.	SISTEMAS MULTIAGENTES.....	17
2.3.1.	Introdução .....	17
2.3.2.	Taxonomia.....	20
2.3.3.	Interação entre Agentes.....	21
2.3.4.	Coordenação, Cooperação e Negociação.....	23
2.3.5.	Aprendizagem de Agentes .....	24
3.	UTILIZAÇÃO DE SISTEMAS MULTIAGENTES EM SISTEMAS SENSÍVEIS AO CONTEXTO.....	27
3.1.	Abordagem.....	28
3.2.	Definição e Representação do Contexto.....	30
3.3.	Arquitetura .....	32
3.3.1.	Gerenciador de Contexto .....	32

3.3.2.	Agente Aprendiz Sensível ao Contexto .....	33
4.	EXEMPLO DE APLICAÇÃO .....	36
4.1.	Cenário.....	36
4.2.	Representação do Contexto .....	39
4.3.	Arquitetura .....	40
4.3.1.	Fontes de Contexto .....	41
4.3.2.	Gerenciador de Contexto .....	42
4.3.3.	Agente <i>Trainer</i> .....	42
5.	CONCLUSÕES .....	48
	ANEXO A – ESPECIFICAÇÃO DA ONTOLOGIA DA APLICAÇÃO.....	49
	LISTA DE REFERÊNCIAS .....	51

## LISTA DE TABELAS

Tabela 1 – Taxonomia de Multiagentes (GARCIA; SICHMAN, 2002) .....	21
Tabela 2 – Classificação dos Atos de Fala (SEARLE, 1969) .....	22
Tabela 3 – Mapeamento de Camadas entre Arquiteturas .....	29
Tabela 4 – Descrição da classe #ContextAttribute (BUCUR; BEAUNE; BOISSIER, 2005c).....	31
Tabela 5 – Tabela de Atributos de Contexto da Academia de Ginástica.....	40
Tabela 6 – Atributos das Subclasses da Ontologia de Domínio.....	50

## LISTA DE FIGURAS

Figura 1 – Dimensões da Computação Ubíqua (LYYTINEN; YOO, 2002) .....	2
Figura 2 – Modelo de Contexto (BAZIRE; BRÉZILLON, 2005) .....	6
Figura 3 – <i>Framework</i> Conceitual de Camadas baseado (AILISTO et al., 2002) .....	15
Figura 4 – Arquitetura Global de Sistemas Sensíveis ao Contexto (BUCUR; BEAUNE; BOISSIER, 2005c) .....	28
Figura 5 – Arquitetura do Context Manager (BUCUR; BEAUNE; BOISSIER, 2005c).....	33
Figura 6 – Arquitetura do Agente Aprendiz (BUCUR; BEAUNE; BOISSIER, 2005c) .....	34
Figura 7 – <i>Personal Digital Assistant</i> .....	37
Figura 8 – Esteira Ergométrica com Monitor LCD.....	38
Figura 9 – Ontologia de Domínio da Academia de Ginástica.....	39
Figura 10 – Arquitetura da Aplicação de Academia de Ginástica .....	41
Figura 11 – Diagrama de Seqüência da Sugestão de Exercícios.....	45
Figura 12 – Diagrama de Seqüência do Monitoramento de Saúde .....	47

## LISTA DE ABREVIATURAS E SIGLAS

ACL	Agent Communication Language
AmI	Ambient Intelligence
CM	Context Manager
FIPA	Foundation for Intelligent Physical Agents
GPS	Global Positioning System
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
IRAS	Instantiated Relevant Set
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
PDA	Personal Digital Assistant
RAS	Relevant Attribute Set
RDP	Resolução Distribuída de Problemas
SGML	Standard Generic Markup Language
SIAG	Sistema de Informação de Academia de Ginástica
SMA	Sistemas Multiagentes
UML	Unified Modeling Language

## 1. INTRODUÇÃO

O fantástico aumento da velocidade dos processadores, da abrangência da comunicação sem fio, do desenvolvimento de tecnologias de armazenamento e apresentação, e a redução dos dispositivos eletrônicos possibilitou a integração de computadores em diversas partes do nosso ambiente.

Nos últimos 50 anos, estas evoluções motivaram a mudança do paradigma de computação do *desktop*, tradicionalmente estático, para um novo paradigma, altamente dinâmico, caracterizado (i) pelo emprego de dispositivos portáteis e multifuncionais, e (ii) pela constante mudança no ambiente, como consequência da mobilidade do usuário.

Tornar a computação parte inseparável de nossa experiência diária e simultaneamente fazer com que ela desapareça, segundo Mark Weiser (1991), é a forma fundamental de se medir o progresso da computação.

Esta visão do futuro, onde estaremos cercados por dispositivos eletrônicos, sensíveis e reativos às pessoas é denominada *Ambiente Intelligence* (AmI) (BUCUR; BEAUNE; BOISSIER, 2005a). Estudos nesta área têm como objetivo a inserção deste novo paradigma em nosso cotidiano.

Segundo Shadbolt (2003), a visão de AmI envolve a convergência de diversas áreas da computação, tais como Computação Ubíqua<sup>1</sup>, Sistemas Inteligentes e Sistemas Sensíveis ao Contexto<sup>2</sup>.

A noção de Computação Ubíqua, cujo termo foi introduzido por Mark Weiser (1991), integra os avanços da Computação Móvel e da Computação Pervasiva<sup>3</sup>, sendo cada uma delas responsável por uma dimensão da Computação Ubíqua, conforme apresentado no gráfico da Figura 1 (LYYTINEN; YOO, 2002).

---

<sup>1</sup> Do inglês *Ubiquitous Computing*.

<sup>2</sup> Do inglês *Context-Aware Systems*.

<sup>3</sup> Do inglês *Pervasive Computing*.

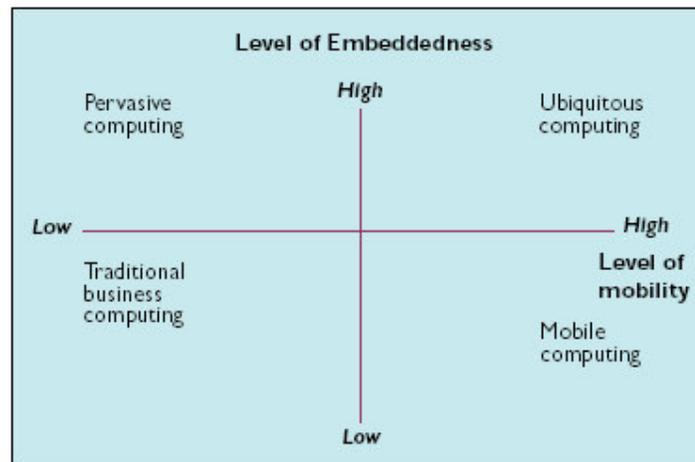


Figura 1 – Dimensões da Computação Ubíqua (LYYTINEN; YOO, 2002)

A Computação Móvel está fundamentalmente relacionada com a mobilidade dos serviços computacionais, o que foi possível com a redução dos dispositivos computacionais e com a possibilidade de acesso aos recursos computacionais através das tecnologias de rede. Entretanto, uma importante limitação deste modelo computacional é que ele não se altera conforme o usuário se move. Isto se deve ao fato do dispositivo não obter informações do contexto em que está inserido, e portanto, não se ajustar ao mesmo. A única possibilidade restante é o próprio usuário realizar a tarefa de ajuste e configuração.

A dimensão de Computação Pervasiva tem por objetivo tornar o computador “invisível”. Isto implica em fazer com que o computador tenha a capacidade de obter informações do ambiente ao qual está inserido, e utilizá-las para se adaptar. De forma recíproca, o ambiente deve ser capaz de detectar a presença de novos dispositivos e utilizar esta informação para realizar alguma ação. Esta interação e dependência mútuas resultam na capacidade dos computadores agirem com “inteligência”.

O maior dos desafios da Computação Ubíqua é a integração da mobilidade (Computação Móvel) com as funcionalidades disponibilizadas pela Computação Pervasiva (LYYTINEN; YOO, 2002). Assim, levar em conta o contexto, especialmente a sua disponibilização para uso, é o pré-requisito obrigatório para a evolução de sistemas de Computação Ubíqua (ABOWD; MYNATT, 2000).

Agentes inteligentes e sistemas multiagentes<sup>4</sup> (SMA) têm se tornado o paradigma da computação predominante nesta área, dentre outras razões, pelo fato dos agentes reduzirem a complexidade destas aplicações através da descentralização e delegação de tarefas.

### **1.1. OBJETIVO**

O presente trabalho tem como objetivo apresentar a aplicabilidade das tecnologias de sistemas multiagentes no desenvolvimento de sistemas sensíveis ao contexto. Adicionalmente, também será esboçado o uso de uma arquitetura de sistema sensível ao contexto na definição de uma aplicação para academia de ginástica.

### **1.2. MOTIVAÇÃO**

Sistemas sensíveis ao contexto precisam estar cientes de seu ambiente e recursos disponíveis, estar habilitados a detectar mudanças neste ambiente, e poder ajustar dinamicamente suas funcionalidades e comportamento, o que implica uma certa “inteligência”.

Tais sistemas apresentam uma grande complexidade em seu desenvolvimento, havendo uma tendência na atualidade em dividi-los em diferentes camadas ou módulos, para simplificar o seu desenvolvimento.

O paradigma de agentes, como será apresentado, é uma promessa na implementação de sistemas nesta área, tendo assim um papel importante no desenvolvimento da visão apresentada de AmI.

A intersecção representada pelo encontro das necessidades do desenvolvimento de sistemas sensíveis ao contexto e das possibilidades encontradas nos SMA é o que motiva este estudo.

---

<sup>4</sup> Do inglês *Multi-Agent System* (MAS).

### **1.3. METODOLOGIA**

Os métodos utilizados na elaboração desta monografia englobam a seleção de literatura e informações disponíveis sobre (i) os elementos de base, ou seja, Contexto, SMA e sistemas sensíveis ao contexto, (ii) aplicação das tecnologias de SMA no desenvolvimento de sistemas sensíveis ao contexto, e (iii) ilustração do possível uso de SMA no desenvolvimento de um sistema sensível ao contexto.

### **1.4. ORGANIZAÇÃO DO DOCUMENTO**

Este documento consiste de 5 capítulos e um anexo. O capítulo 2 apresenta uma introdução às noções de contexto, sistemas sensíveis ao contexto e sistemas multiagentes, que constituem a base fundamental deste trabalho. A utilização de tecnologias de SMA em um sistema sensível ao contexto é exibida no capítulo 3. O capítulo 4 ilustra o uso de uma arquitetura de sistema sensível ao contexto na implementação de uma aplicação para academia de ginástica, identificando seus componentes essenciais, suas características e sua distribuição. As conclusões do trabalho, bem como algumas perspectivas futuras, são apresentadas no capítulo 5. Finalmente, apresenta-se no anexo A a ontologia desenvolvida para a aplicação detalhada no capítulo 4.

## **2. ELEMENTOS DE BASE**

Neste capítulo, é fornecida uma introdução teórica dos elementos básicos da elaboração deste trabalho: contexto, sistemas sensíveis ao contexto e SMA.

### **2.1. CONTEXTO**

Nesta seção, apresenta-se uma noção de contexto: sua definição, algumas tentativas de categorização, modos de aquisição e modelos de representação.

#### **2.1.1. Definição**

Contexto vem sendo objeto de investigação científica há vários anos em algumas áreas científicas, como Linguística e Psicologia Cognitiva. Na área da Ciência da Computação, os estudos sobre o tema são mais recentes.

Apesar de já ter havido grandes avanços no entendimento e formalização sobre contexto, o termo possui uma grande variedade de significados na literatura, dependendo do propósito da aplicação particular e/ou do ponto de vista da comunidade científica em questão.

Na literatura, muitas definições podem ser encontradas conforme mencionado acima, mas a definição mais referenciada e a que será utilizada neste trabalho é a apresentada em (ABOWD et al., 1999):

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

Assim, pode-se considerar contexto como sendo o conjunto de atributos relevantes para um determinado propósito (ou finalidade), onde propósito é o objetivo no qual o contexto é utilizado em um determinado momento.

Na tentativa de estimular a discussão da comunidade em direção a uma definição única para contexto, Bazire e Brézillon (2005) coletaram na Web e analisaram um conjunto de definições de contexto de diferentes disciplinas como Psicologia, Ciência da Computação, Economia, Filosofia e outras.

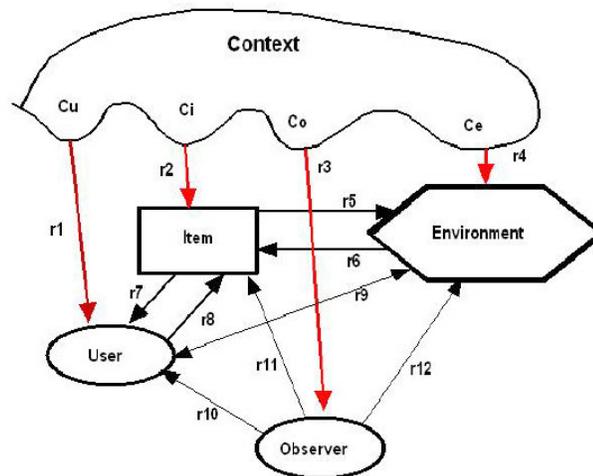


Figura 2 – Modelo de Contexto (BAZIRE; BRÉZILLON, 2005)

Desta coleta e análise, concluíram que o contexto pode ser analisado através de seis componentes essenciais: restrição, influência, comportamento, natureza, estrutura e sistema<sup>5</sup>, assim dando origem a um modelo de comparação de definições de contexto ilustrado na Figura 2.

O modelo de contexto apresentado é formado por um ou mais componentes: item, usuário, ambiente e observador<sup>6</sup>. Quando estes componentes são considerados no contexto, eles são denominados de Ce (Contexto do Ambiente), Ci (Contexto do Item), Co (Contexto do Observador) e Cu (Contexto do Usuário).

<sup>5</sup> Do inglês *constraint, influence, behavior, nature, structure e system*.

<sup>6</sup> Do inglês *item, user, environment e observer*.

Cada componente do modelo pode se relacionar com um ou mais componentes. Este relacionamento é representado no modelo através das arestas uni ou bidirecionais, apresentadas na Figura 2 acima, e são denominados de  $r_i$ , onde  $i$  é um número inteiro entre 1 e 12.

O modelo de contexto proposto sustenta a hipótese de que a razão pela qual as definições sobre contexto divergem é porque elas não focam os mesmos componentes e as mesmas relações.

Mapeando a definição de contexto utilizada neste trabalho, tem-se que o foco da definição está na entidade. Como a entidade é uma pessoa, um lugar ou um objeto, conclui-se que a definição leva em consideração o contexto do item  $[C_i]$  (objeto), do usuário  $[C_u]$  (pessoa) e do ambiente  $[C_e]$  (lugar). As relações a serem consideradas neste caso são  $r_1, r_2, r_4, r_5, r_6, r_7, r_8$  e  $r_9$ , que representam os relacionamentos dos contextos com as entidades e entre as entidades.

Freqüentemente usadas como sinônimo de contexto, informações de contexto são as informações que caracterizam o contexto.

### **2.1.2. Categorização**

Segundo a análise realizada por Mostéfaoui, Pasquier-Rocha e Brézillon (2004), não existe uma categorização genérica de contexto. A relevância de cada uma das informações de contexto é diferente, dependendo do domínio da aplicação e de sua utilização. Categorizar o contexto baseado em um conjunto fixo de informações de contexto não é viável, porque dependendo da aplicação haverá necessidade de se incluir ou retirar informações do conjunto.

Apesar de não existir uma categorização genérica, a categorização de contexto é extremamente importante, devido à grande diversidade existente de contextos. A categorização facilita a compreensão das informações de contexto, a escolha dos tipos de informações necessárias para cada sistema, e permite a busca de outras informações de contexto relevantes.

Em (ABOWD et al., 1999), as informações de contexto são divididas em quatro categorias: Localização<sup>7</sup> (a posição da entidade, proximidade, etc.), Identidade<sup>8</sup> (cada entidade possui uma identificação única), Atividade<sup>9</sup> (propriedades intrínsecas de uma entidade como temperatura e iluminação de uma sala, etc.) e Tempo<sup>10</sup> (utilizada para determinar precisamente uma situação, ordenar eventos, etc.). Estas são denominadas categorias primárias da informação de contexto, que caracterizam a situação de uma entidade específica.

A categorização primária, além de fornecer as informações da situação de uma entidade específica, também serve de índice para outras informações de contexto, chamadas de informações de contexto secundárias. Neste caso, as informações primárias são utilizadas como índices para a determinação das informações de contexto secundárias para uma mesma entidade.

Esta categorização sugerida define uma hierarquia de dois níveis, onde no primeiro nível estão os quatro tipos de informação de contexto identificados acima, e todos os outros tipos são colocados no segundo nível.

Por exemplo, a data de aniversário de uma pessoa é categorizada como uma informação de contexto secundária, pois sua obtenção pode ser feita através da informação de contexto primária, identidade do usuário. A identidade do usuário neste caso é utilizada como índice para obtenção da data de aniversário.

### **2.1.3. Aquisição**

Em (MOSTÉFAOUI; PASQUIER-ROCHA; BRÉZILLON, 2004) são descritas três formas para se adquirir informações de contexto:

---

<sup>7</sup> Do inglês *location*.

<sup>8</sup> Do inglês *identity*.

<sup>9</sup> Do inglês *activity*.

<sup>10</sup> Do inglês *time*.

- *Sensor* – a informação de contexto é obtida a partir de sensores físicos. Os sensores físicos são equipamentos que captam informações, como temperatura, umidade do ar, ritmo de batimentos cardíacos dentre outras;
- *Derivada* – a informação de contexto é computada em tempo de execução, como o tempo de corrida em uma esteira que é obtida pela expressão  $(tf - ti)$ , onde  $tf$  é o tempo final e  $ti$  é o tempo inicial de corrida;
- *Explícita* – ocorre quando a informação de contexto é obtida explicitamente do usuário do sistema, como normalmente é o caso das configurações de preferências pessoais (velocidade máxima de corrida, música preferida, etc).

A aquisição de informações de contexto por sensores é complexa, pois as informações podem ser provenientes de fontes heterogêneas e distribuídas, e/ou serem dinâmicas.

#### **2.1.4. Modelos de Representação**

Um modelo de representação de contexto é necessário para definir e armazenar informações de contexto de uma forma que possam ser processadas por um computador. Os modelos de representação de contexto são utilizados na representação, armazenamento e troca de informações de contexto.

Estes modelos podem ser classificados, segundo Strang e Linnhoff-Popien (2004), em seis tipos distintos:

#### **Modelos de Pares Atributo-Valor**

Uma informação de contexto é descrita como um conjunto de pares atributo-valor, no qual o atributo descreve uma propriedade do contexto. Este é o modelo mais simples de ser implementado, mas é limitado devido a ausência de definição de tipo e inter-relacionamento entre contextos.

### **Modelos Baseados em Esquema**

O contexto é representado como estruturas de dados hierárquicas, compostas de pares atributos e conteúdo. Normalmente, são descritas por linguagens derivadas do *Standard Generic Markup Language* (SGML). Este modelo possui uma estrutura bem definida e permite a composição de informações de contexto.

### **Modelos Gráficos**

O contexto é representado por meio de linguagens gráficas tais como extensões de *Unified Modeling Language* (UML). Modelos baseados em UML são capazes de modelar ricas composições e inter-relacionamentos entre informações de contexto, mas são limitados quanto à modelagem do comportamento dinâmico e temporalidade.

### **Modelos Orientados a Objetos**

O contexto é representado por meio de modelos de objetos, utilizando-se as propriedades de encapsulamento, reutilização e abstração.

### **Modelos Baseados em Lógica**

Neste modelo, o contexto é representado por meio de fatos, expressões e regras que especificam como uma informação de contexto pode ser inferida ou derivada a partir de outra. Este modelo baseia-se em uma formalização matemática muito precisa.

### **Modelos Baseados em Ontologias**

Os contextos e seus inter-relacionamentos são descritos por meio de ontologias.

Uma ontologia engloba uma visão específica do mundo relativa a um domínio particular. Tal visão do mundo, também chamada de conceitualização, é geralmente concebida como um conjunto de conceitos, tais como entidades, atributos e processos, com suas respectivas definições e relacionamentos. Uma ontologia pode ter várias formas, mas deve incluir necessariamente um

vocabulário de termos e as respectivas definições de significado (USCHOLD; GRUNINGER, 1996).

Devido a sua capacidade de descrever contextos complexos, as ontologias são particularmente aplicáveis na descrição de informações utilizadas no dia-a-dia, em estruturas utilizáveis por computadores.

Alguns requisitos e objetivos de uma ontologia são (KORPIÄÄ et al., 2003):

- *Simplicidade* – As expressões e relações utilizadas devem ser as mais simples possíveis para simplificar o trabalho dos desenvolvedores de sistemas;
- *Flexibilidade e Extensibilidade* – A ontologia deve ser capaz de incorporar de forma simples a inclusão de novos elementos e relações;
- *Generalidade* – A ontologia não deve se limitar a um tipo único de contexto, mas sim, servir de base a diferentes tipos de contexto;
- *Expressividade* – A ontologia deve permitir descrever o maior número possível de estados de contexto em detalhes arbitrários.

### **Modelos Baseados em Entidade-Associação-Atributo**

A representação da informação de contexto é dada por uma extensão do modelo de Pares Atributo-Valor, onde as informações de contexto são estruturadas para uma entidade, e cada entidade representa um objeto físico ou conceitual (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002).

Segundo Strang e Linnhoff-Popien (2004), as fraquezas e qualidades de cada um desses seis modelos de contexto podem ser julgados com base em seis requisitos necessários à Computação Ubíqua: composição distribuída da informação, validação

parcial da informação, riqueza e qualidade da informação, incompletude e ambigüidade, nível de formalidade e aplicabilidade.

Dos seis modelos apresentados, segundo Strang e Linnhoff-Popien (2004), somente os modelos orientados a objetos e modelos baseados em ontologias atendem a todos os requisitos.

## **2.2. SISTEMAS SENSÍVEIS AO CONTEXTO**

Uma vez definida a noção de contexto, nesta seção apresenta-se uma visão geral de sistemas sensíveis ao contexto: definição do termo, categorização e arquiteturas de sistemas sensíveis ao contexto.

### **2.2.1. Definição**

O termo sistema sensível ao contexto foi mencionado pela primeira vez por Schilit e Theimer (1994), definido como aplicações que *“se adaptavam de acordo com sua localização, grupo de pessoas e objetos próximos, bem como com as mudanças destes objetos com o tempo”*.

Da mesma forma que não existe uma definição única para contexto, também não existe uma definição única para sistema sensível ao contexto. Entretanto, para os projetistas de sistemas, é essencial o entendimento das características de um sistema sensível ao contexto (SCHILIT; ADAMS; WANT, 1994) (ABOWD et al., 1999).

Será adotada a definição de Abowd et al. (1999) que diz que *“os sistemas sensíveis ao contexto são aqueles que utilizam informações de contexto para disponibilizar informação e/ou serviços relevantes ao usuário, onde a relevância depende da tarefa que o usuário está realizando”*.

### **2.2.2. Categorização**

Os sistemas sensíveis ao contexto podem ser classificados, segundo Abowd et al. (1999), segundo características que devem ser disponibilizadas por um sistema sensível ao contexto. As três categorias identificadas em Abowd et al. (1999) são:

apresentação de informação e serviços, execução automática de um serviço e anexação de informação de contexto para recuperação futura.

### **Apresentação de Informação e Serviços**

As aplicações desta categoria apresentam as informações de contexto para o usuário, ou então utilizam estas informações para propor a seleção de ações por parte do usuário. Por exemplo, sugerir os serviços de Monitoramento de Saúde quando um aluno da academia inicia alguma atividade física.

### **Execução Automática de um Serviço**

As aplicações desta categoria executam um comando, ou reconfiguram o sistema no lugar do usuário, de acordo com as mudanças do contexto. Por exemplo, reajustar o equipamento de ginástica do aluno devido à identificação de batimento cardíaco acima do limite prescrito pelo médico.

### **Anexação de Informação de Contexto para Recuperação Futura**

As aplicações desta categoria estendem o contexto com dados relevantes, capturados em uma determinada situação. Por exemplo, registrar os atributos selecionados como relevantes pelo próprio aluno no momento de selecionar sua próxima atividade física.

### **2.2.3. Arquitetura**

Os sistemas sensíveis ao contexto podem ser estruturados de diferentes formas. Segundo Baldauf, Dustdar e Rosenberg (2004), a estrutura a ser utilizada depende de diferentes requisitos como localização dos sensores, número de possíveis usuários, recursos disponíveis nos dispositivos utilizados ou facilidade de extensão do sistema.

Vários métodos podem ser utilizados na aquisição de informações de contexto. O método utilizado na aquisição de informação de contexto é um dos requisitos mais importantes na definição da arquitetura de um sistema sensível ao contexto. As informações de contexto, conforme descrito na seção 2.1.3, podem ser provenientes

de sensores, derivadas ou explícitas. Chen (2004) apresenta três métodos para a aquisição de informação de contexto:

### **Acesso Direto ao Sensor**

Neste método, a aplicação adquire o contexto através do acesso direto ao sensor físico, sendo mais utilizado em aplicações instaladas em dispositivos com sensores embutidos. Os controladores de acesso ao *hardware* são embutidos na aplicação, não sendo comumente utilizado em sistemas com sensores distribuídos.

O benefício deste método é que as aplicações podem ter um maior controle sobre as operações de baixo nível com os sensores, e podem ter um melhor conhecimento de como os diferentes dados são coletados e computados.

Entretanto, para que a aplicação se comunique com todos os sensores, é necessário que se implemente um protocolo de comunicação específico para cada um deles. Isto implica em uma maior dificuldade no desenvolvimento e manutenção da aplicação.

### **Infra-estrutura de *Middleware***

O método baseado em *middleware* introduz uma arquitetura em camadas na estruturação de sistemas sensíveis ao contexto, cujo intuito é isolar as aplicações dos detalhes das operações de baixo nível com sensores.

Este método simplifica a reutilização de código e facilita a integração de novos sensores, já que não há necessidade de se alterar cada uma das aplicações que acessam os sensores.

Apesar dos ganhos apresentados, este método impõe um maior consumo de recursos computacionais aos dispositivos onde estiver sendo executado o *middleware*, assim gerando um problema para dispositivos com recursos computacionais limitados, como telefones celulares e dispositivos embutidos.

### Servidor de Contexto

Um servidor de contexto é uma entidade que disponibiliza contextos para diferentes sistemas sensíveis ao contexto em um ambiente distribuído. A idéia é realizar as operações de aquisição e gerenciamento de contextos a partir de um dispositivo com maior disponibilidade de recursos computacionais, diminuindo assim a carga de utilização de recursos computacionais nos dispositivos remotos.

Através desses servidores, sistemas que não possuem capacidade de sensoriamento podem obter informações de contexto, e se tornarem sistemas sensíveis ao contexto.

A arquitetura em camadas é a forma mais comum de estruturação de sistemas sensíveis ao contexto na atualidade. Esta arquitetura facilita a reutilização e compartilhamento de componentes de *software* e *hardware* entre sistemas. Como consequência, a complexidade do desenvolvimento deste tipo de sistema se reduz.



**Figura 3 – *Framework* Conceitual de Camadas baseado (AILISTO et al., 2002)**

O *framework* conceitual de camadas apresentado neste trabalho, conforme mostra Figura 3, consiste de cinco camadas: física, dados, semântica, inferência e

aplicação<sup>11</sup> (AILISTO et al., 2002). Cada uma das camadas tem como função aumentar o nível de abstração dos dados coletados conforme se sobe na pilha de camadas.

### **Camada 1 – Camada Física**

Esta primeira camada é composta por uma coleção de sensores, onde sensores não são somente dispositivos físicos, mas qualquer tipo de *hardware* ou *software* gerador de informação. Os dados gerados pelos sensores são fornecidos em sua forma bruta.

### **Camada 2 – Camada de Dados**

Esta camada é responsável pela coleta dos dados da camada física, realizando o acesso a camada inferior através de controladores de acesso ao *hardware* ou através de APIs específicas.

### **Camada 3 – Camada Semântica**

Esta camada é responsável por transformar os dados de um formato bruto em informações com mais significado. Um exemplo é transformar a localização de uma pessoa das coordenadas retornadas por um dispositivo GPS para o nome da sala onde esta pessoa se encontra.

### **Camada 4 – Camada de Inferência**

Esta camada utiliza informações da camada semântica, informações históricas e regras de inferência para raciocinar sobre as necessidades dos usuários e disponibilizar serviços que os atendam.

Esta camada pode ser acessada de duas formas:

- *Síncrona* – os dados desta camada são coletados através de chamadas de método remoto;

---

<sup>11</sup> Do inglês *Physical, Data, Semantic, Inference e Application*.

- *Assíncrona* – o cliente se registra nesta camada informando todos os eventos que o interessam. Quando um destes eventos ocorre, o cliente é notificado ou uma chamada de método do cliente é executada.

### **Camada 5 – Camada de Aplicação**

Esta camada implementa a aplicação do cliente, que utiliza informação do contexto. A implementação da reação aos diferentes eventos e contextos é realizada nesta camada.

De forma mais abstrata, um sistema sensível ao contexto pode ser dividido em duas partes: (1) sensoriamento, para a captura das informações de contexto (Camada Física, Camada de Dados e Camada Semântica); e (2) racionalidade, para realização de operações de inferência (Camada de Inferência e Camada de Aplicação).

## **2.3. SISTEMAS MULTIAGENTES**

Nesta seção, apresenta-se uma visão geral de SMA, propondo uma taxonomia de classificação e detalhando os aspectos de utilização e aprendizagem.

### **2.3.1. Introdução**

Os SMA são estudados especialmente dentro de um ramo da Inteligência Artificial (IA) chamado de Inteligência Artificial Distribuída (IAD). A IAD baseia-se no comportamento social como modelo de inteligência, com ênfase em ações e interações entre agentes (SICHTMAN et al., 1992). A IAD estuda aspectos sociológicos e etológicos, enquanto a IA tradicional estuda aspectos de origem psicológica.

O domínio da IAD pode ser dividido em duas abordagens distintas, denominadas: Resolução Distribuída de Problemas<sup>12</sup> (RDP) e Sistemas Multiagentes (SMA).

---

<sup>12</sup> Do inglês *Distributed Problem Solving* (DPS).

A abordagem RDP utiliza como ponto de partida a existência de um problema, a partir do qual, concebe-se e cria-se um sistema. Portanto, a RDP se propõe a solucionar um problema, previamente definido e especificado, através do projeto de uma sociedade de agentes.

Na abordagem SMA, estuda-se o comportamento de um grupo organizado de agentes autônomos que interagem através de um ambiente compartilhado, e sobre o qual estes agentes atuam, alterando seu estado. O termo autônomo designa aqui o fato de que os agentes têm uma existência própria, independente da existência de outros agentes.

Em SMA, não existe um problema específico a ser resolvido. Assim, o objetivo é estudar modelos genéricos a partir dos quais pode-se conceber agentes, organizações e interações, de modo a poder instanciar tais conceitos quando se deseja resolver um problema particular (ALVARES; SICHTMAN, 1997).

Segundo Álvares e Sichertman (1997), entende-se como agente cada entidade ativa do sistema. O ambiente de um sistema é o conjunto das entidades passivas visíveis aos agentes. Entende-se como interação a troca de informações entre agentes, podendo a troca ser realizada diretamente entre os agentes ou indiretamente através do ambiente. Ao conjunto de agentes que convergem em um mesmo ambiente denomina-se sociedade. Entende-se como organização às restrições aplicadas aos agentes pertencentes a uma sociedade para que possam atingir seus objetivos globais mais facilmente.

As principais características dos SMA são as seguintes (SYCARA, 1998):

- Cada agente possui informação ou capacidade incompleta, assim tendo uma visão limitada do todo;
- Não existe um controle centralizado;
- Os dados estão descentralizados;
- A computação é assíncrona.

Os SMA podem ainda ser subdivididos em duas abordagens, denominadas: Sistemas Multiagentes Reativos e Sistemas Multiagentes Cognitivos.

Os Sistemas Multiagentes Reativos trabalham com o desenvolvimento de sistemas que utilizam um grande número de agentes simples para a resolução de um determinado problema. Segundo Ferber e Gasser (1991), as principais características desta abordagem são:

- Não há representação explícita do conhecimento: o conhecimento dos agentes é implícito (as suas regras de comportamento) e sua manifestação se externa através do seu comportamento e dos demais agentes;
- Não há representação do ambiente: o comportamento (resposta) de cada agente é baseado no que ele percebe (estímulo) a cada instante. Não há uma representação interna explícita do ambiente;
- Não há memória das ações: os agentes reativos não mantêm nenhum tipo de histórico de suas ações, ou seja, o resultado de uma determinada ação passada não influencia diretamente na decisão de uma ação futura;
- Organização etológica: a forma de organização dos SMA reativos é similar à observada por animais que vivem em grandes comunidades;
- Grande número de membros: em geral, os SMA reativos possuem um grande número de agentes, com populações que podem chegar à ordem de milhares de membros.

Os Sistemas Multiagentes Cognitivos trabalham com poucos agentes que realizam tarefas mais complexas. Segundo Feber e Gasser (1991), as principais características desta abordagem são:

- Representação explícita do ambiente e dos outros agentes da sociedade;

- Podem manter um histórico das interações e ações passadas e, graças a esta memória, são capazes de planejar suas ações futuras;
- Seu sistema de percepção, que permite examinar o ambiente, e o de comunicação, que permite a troca de mensagens entre agentes, são distintos. A comunicação entre agentes é realizada de modo direto, através do envio e recebimento de mensagens;
- Seu mecanismo de controle é deliberativo. Os agentes cognitivos raciocinam e decidem em conjunto sobre quais ações devem ser executadas, que planos seguir e que objetivos devem alcançar;
- Considerando a última asserção, os modelos de organização dos SMA cognitivos são modelos sociológicos, como as organizações humanas;
- Um SMA cognitivo contém, usualmente, poucos agentes, na ordem de algumas dezenas, no máximo.

A criação de mecanismos genéricos de interação e coordenação dos agentes para que o sistema funcione de forma adequada e eficiente é um dos grandes desafios da área de SMA, e serão detalhados mais adiante nesta seção.

### **2.3.2. Taxonomia**

Considerando que os SMA são sistemas computacionais especiais, pode-se classificá-los segundo alguns eixos (GARCIA; SICHTMAN, 2002), conforme mostra a Tabela 1.

<b>Eixo</b>	<b>Descrição</b>
Perspectiva	Um SMA pode ter uma perspectiva de simulação social (objetivo é o estudo das interações sociais) ou de resolução social (objetivo é o desenvolvimento de técnicas para resolver problemas de forma cooperativa e distribuída).
Abertura	Um SMA pode ser aberto (número de agentes altera-se dinamicamente) ou não (número de agentes é estático).
Granularidade	Um SMA pode ter baixa (poucos agentes) ou alta granularidade (muitos agentes).
Composição	Um SMA pode ser composto por agentes idênticos (composição homogênea) ou por agentes diferentes (composição heterogênea).
Interação	Existem diferentes padrões de interação de agentes, conforme (GARCIA; SICHMAN, 2002).

**Tabela 1 – Taxonomia de Multiagentes (GARCIA; SICHMAN, 2002)**

### **2.3.3. Interação entre Agentes**

A interação entre agentes em um SMA é frequentemente associada com alguma forma de comunicação. Os agentes interagem quando buscam atingir seus objetivos em um ambiente compartilhado.

Como um agente não permite que outro altere seu estado interno, existe a necessidade da comunicação entre agentes através de ações de comunicação. Essas ações possibilitam que um agente influencie outros, sem que haja acesso direto ao estado interno dos mesmos.

Os estudos na área de comunicação estão, em sua maioria, relacionados com estudos da filosofia da linguagem, e mais especificamente com a Teoria dos Atos de Fala. De acordo com essa teoria, a linguagem humana é entendida como ação, no sentido de que ela muda o estado do mundo de maneira análoga a uma ação física (WOOLDRIDGE, 1996).

Austin (1962) observou que algumas sentenças em linguagem natural não poderiam ser analisadas em termos de condições de verdade, em contraposição aos trabalhos de semântica lógica. As sentenças deste tipo, que não podem ser analisadas como verdadeiras ou falsas, são chamadas de performativas. Ele identificou três aspectos dos atos de fala:

- locução<sup>13</sup> é o enunciado emitido pelo falante através de um meio físico;
- ilocução<sup>14</sup> é a intenção associada ao enunciado do falante;
- perlocução<sup>15</sup> é a ação resultante ou efeito da locução.

Considere os falantes A e B. O falante A ao dizer a B “reduza a velocidade de corrida”, emite uma mensagem (locução), com a intenção de solicitar uma ação de B (ilocução), e como resultado desta ação lingüística, B reduz a velocidade de corrida (perlocução).

Uma extensão do trabalho de Austin (1962) feita por Searle (1969), classificou os atos de fala ilocucionários em cinco tipos diferentes, conforme mostra a Tabela 2.

Representativas	O falante comunica que acredita na verdade da expressão (por exemplo, através da conclusão).
Diretivas	O falante tem a intenção de provocar o ouvinte a realizar uma ação (por exemplo, requisição).
Comissivas	O falante se compromete com a realização futura de uma ação (por exemplo, promessa).
Expressivas	O falante expressa um estado psicológico (por exemplo, agradecimento).
Declarativas	Têm como efeito imediato a mudança do estado do mundo (por exemplo, declaração de casado).

**Tabela 2 – Classificação dos Atos de Fala (SEARLE, 1969)**

<sup>13</sup> Do inglês *locutionary act*.

<sup>14</sup> Do inglês *illocutionary act*.

<sup>15</sup> Do inglês *perlocution act*.

A teoria de atos de fala influenciou o desenvolvimento de diversas linguagens de comunicação entre agentes, dentre as quais as duas mais conhecidas são KQML-KIF (KQML, 2005) (KIF, 2005) e FIPA-ACL (FIPA, 2005).

A comunicação habilita a coordenação das ações e comportamentos dos agentes em um ambiente. Tal coordenação gera como resultado um sistema mais coerente. Para isto, é necessário que todos compartilhem uma conceitualização comum do domínio. Uma conceitualização nomeia e descreve as entidades que possam existir no domínio, descrevendo também os relacionamentos entre essas entidades. Um vocabulário para representar e comunicar o conhecimento sobre o domínio também é fornecido. Assim, as ontologias são também muito úteis na comunicação entre agentes.

#### **2.3.4. Coordenação, Cooperação e Negociação**

A coordenação de atividades de um conjunto de agentes é talvez uma das tarefas mais complexas da sociedade humana (GARCIA; SICHMAN, 2002). Existem duas formas de coordenação do comportamento e atividades de agentes, a cooperação e a negociação.

Cooperação significa que dois ou mais agentes estão tentando atingir conjuntamente um mesmo objetivo, e estão conscientes deste fato (GARCIA; SICHMAN, 2002). A estratégia mais utilizada na implementação dos protocolos de cooperação é a decomposição do problema em tarefas e sua posterior distribuição.

Negociação é o estabelecimento de um compromisso entre agentes divergentes em relação a certos aspectos para execução de uma atividade em comum. Em geral, qualquer negociação é composta por quatro componentes (WOOLDRIDGE, 1996):

- Domínio de negociação, que representa o conjunto das possíveis propostas que um agente pode fazer;

- Protocolo, que define as propostas legais que um agente pode fazer;
- Coleção de estratégias, uma para cada agente, a qual determina que proposta o agente fará. Normalmente, a estratégia utilizada é conhecida somente pelo agente, não sendo compartilhada;
- Regra, que determina quando um negócio foi fechado, e ao que se refere o compromisso estabelecido.

A negociação é um processo iterativo, que consiste de diversas rodadas, e em cada rodada cada um dos agentes faz uma proposta. O processo de negociação pode ocorrer de forma automática, sendo necessário para isto a criação de protocolos de interação, que permitam pré-estabelecer etapas definidas na troca de mensagens entre os agentes, com o intuito de fazer com que o processo de negociação convirja e tenha um bom término.

O processo de negociação é complexo, e um dos motivos que o torna complexo é o número de participantes envolvidos, e como estes agentes interagem. Existem três possibilidades de negociação: um-para-um, muitos-para-um e muitos-para-muitos (WOOLDRIDGE, 1996).

### **2.3.5. Aprendizagem de Agentes**

A aprendizagem em SMA se tornou uma área de pesquisa de grande importância no ramo da Inteligência Artificial Distribuída nos últimos anos. Isto ocorreu devido os SMA serem tipicamente muito complexos, e seu dinamismo e comportamento serem de difícil especificação. Assim, os agentes de um SMA são induzidos a possuir a habilidade de se adaptarem, através da aprendizagem, para melhorar seu desempenho futuro.

Segundo Weiss e Dillenbourg (1999), a aprendizagem multiagente pode ser caracterizada a partir de duas perspectivas: (1) aprendizagem mono-agente<sup>16</sup>; e (2) aprendizagem colaborativa<sup>17</sup>. A perspectiva de aprendizagem colaborativa não será explorada neste trabalho.

A aprendizagem mono-agente permite a caracterização da aprendizagem em três classes de mecanismos (WEISS; DILLENBOURG, 1999):

- **Multiplicação**

Este mecanismo sugere que se cada agente em um SMA possui um mecanismo de aprendizagem, então o sistema como um todo aprende. Neste caso, existem múltiplos agentes aprendizes, mas cada um deles aprende independentemente dos demais. As interações dos agentes não interferem no processo de aprendizagem individual. Os algoritmos de aprendizagem podem ser diferentes para cada um dos agentes.

- **Divisão**

No mecanismo de divisão, a aprendizagem é dividida entre diversos agentes. A divisão pode se dar pelos aspectos funcionais do algoritmo e/ou de acordo com as características dos dados a serem processados para obtenção dos efeitos de aprendizagem desejados. A interação é necessária para que sejam consolidados os resultados obtidos dos diversos agentes. A divisão do algoritmo de aprendizagem e a definição das interações são definidas na fase de projeto do sistema e não faz parte do processo de aprendizagem. Esta divisão reduz a carga de processamento em cada agente.

- **Interação**

Os mecanismos de aprendizagem interativa são baseados no fato dos agentes interagirem durante a aprendizagem, e não somente para troca de informações que servem de entrada e saída ao processo de aprendizagem. Neste caso, a

---

<sup>16</sup> Do inglês *single-agent learning*.

<sup>17</sup> Do inglês *humam-humam collaborative learning*.

interação é uma atividade mais dinâmica que afeta os processos intermediários da aprendizagem, e não está relacionada somente com a troca de dados. Os agentes influenciam a forma de aprendizagem de outros agentes.

No próximo capítulo, mostra-se como a tecnologia dos SMA pode ser utilizada para conceber e implementar sistemas sensíveis ao contexto.

### **3. UTILIZAÇÃO DE SISTEMAS MULTIAGENTES EM SISTEMAS SENSÍVEIS AO CONTEXTO**

O desenvolvimento de sistemas sensíveis ao contexto é considerado uma tarefa complexa e desafiadora (SALBER; DEY; ABOWD, 1999), sendo o conceito de contexto o componente principal e mais importante na realização desta tarefa.

A definição de contexto apresentada na seção 2.1 indica dois dos maiores problemas enfrentados na atualidade pelos sistemas sensíveis ao contexto: (1) a quantidade de informação de contexto que deve ser gerenciada (o trecho da definição “qualquer informação [...] utilizada para caracterizar a situação de uma entidade”, dá uma idéia da complexidade das tarefas de definição e gerenciamento das informações de contexto); e (2) a relevância da informação (“...informação que pode ser utilizada para...”, “...que é considerada relevante...”), onde não se especifica na definição como escolher dentre todas as informações de contexto quais são as relevantes para uma determinada situação (BUCUR; BEAUNE; BOISSIER, 2005c).

A abordagem usual dos projetistas de sistemas sensíveis ao contexto para solucionarem estes problemas é trata-los ainda na fase de projeto. Assim, todas as informações que farão parte do contexto são definidas nesta fase e programadas diretamente na aplicação (CHEN et al., 2001). Ainda nesta fase, define-se qual a relevância de cada uma das informações de contexto para a aplicação.

O maior problema da abordagem apresentada está relacionado com as mudanças ocorridas do paradigma de computação do *desktop*, tradicionalmente estático, para um novo paradigma, altamente dinâmico, caracterizado pelo emprego de dispositivos portáteis e multifuncionais, e pela constante mudança no ambiente, como consequência da mobilidade do usuário. Neste novo paradigma, é muito mais difícil, se não impossível, se prever as mudanças e novas configurações que podem ocorrer. Considerando este fato, uma abordagem estática do contexto em sistemas sensíveis ao contexto não é apropriada.

Uma nova abordagem, baseada em SMA e proposta por Bucur, Beaune e Boissier (2005c), foca o uso de tecnologias de SMA para dinamizar os sistemas sensíveis ao contexto, e transferir para o usuário ou para a aplicação a tarefa da determinação da relevância das informações de contexto em uma determinada situação.

Na literatura, encontram-se diversas arquiteturas para sistemas sensíveis ao contexto baseadas em SMA (CHEN et al., 2001) (LECH; WIENHOFEN, 2005). Entretanto, mesmo utilizando-se um SMA, a responsabilidade de determinação da relevância das informações de contexto continua a cargo do projetista do sistema.

Neste trabalho, será utilizada a abordagem proposta por Bucur, Beaune e Boissier (2005c) para apresentar a utilização de SMA em sistemas sensíveis ao contexto, detalhada nas seções que se seguem.

### 3.1. Abordagem

Bucur, Beaune e Boissier (2005c) propõem que os sistemas sensíveis ao contexto devem: (1) separar o gerenciamento do contexto do raciocínio sobre/com o contexto, e (2) adaptar, dinamicamente, a seleção de informações de contexto relevantes de acordo com a tarefa que está sendo realizada. Esses dois itens direcionam para uma arquitetura em camadas (Figura 4), e cria uma clara distinção entre o gerenciamento de contexto (*Layer 1*) e o raciocínio sobre/com o contexto (*Layer 2*).

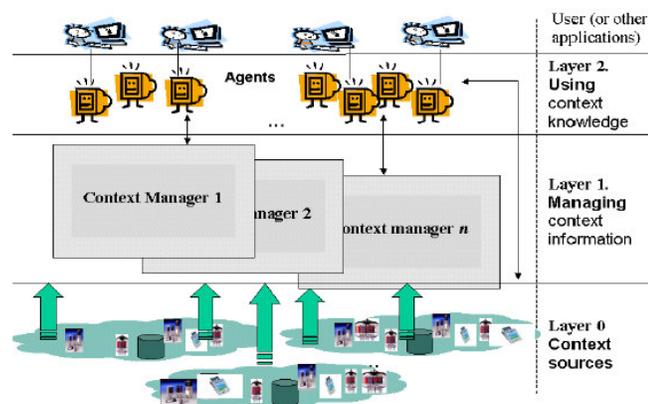


Figura 4 – Arquitetura Global de Sistemas Sensíveis ao Contexto (BUCUR; BEAUNE; BOISSIER, 2005c)

O *Layer 1 – Context Manager* (Camada de Dados e Camada Semântica) realiza as operações de aquisição e gerenciamento de contexto, isolando da camada de inferência tais tarefas.

O *Layer 2 – Using Context Knowledge* (Camada de Inferência) é representado por um conjunto de agentes heterogêneos que raciocinam sobre/com o contexto para:

- Solucionar o mesmo problema em contextos diferentes;
- Solucionar problemas diferentes em determinado contexto.

Para tornar os sistemas sensíveis ao contexto dinâmicos, cada um dos agentes desta camada implementa capacidades de raciocínio, que os habilitam a selecionar de forma dinâmica e adaptável as informações de contexto relevantes e a tomarem decisões apropriadas. As capacidades de raciocínio são as técnicas de aprendizagem individual de cada agente ou as técnicas de compartilhamento de conhecimento em um ambiente multiagentes.

Mapeando a arquitetura de camadas apresentada na Figura 4 para o *framework* conceitual de camadas de Ailiston et al. (2002) tem-se a correlação mostrada na Tabela 3.

<b>Arquitetura Global de Sistemas Sensíveis ao Contexto</b>	<b><i>Framework</i> Conceitual de Camadas</b>
<i>Layer 2 – Using Context Knowledge</i>	Camada de Inferência
<i>Layer 1 – Managing Context Information</i>	Camada de Dados e Camada Semântica
<i>Layer 0 – Context Sources</i>	Camada Física

**Tabela 3 – Mapeamento de Camadas entre Arquiteturas**

Antes de descrever mais detalhadamente a arquitetura apresentada, será apresentada uma definição e representação de contexto que permite que os agentes interroguem o

Gerenciador de Contexto (*Context Manager*) e raciocinem sobre/com o contexto obtido.

### 3.2. Definição e Representação do Contexto

Contexto, conforme definido na seção 2.1.1, é composto por um conjunto de atributos de contexto e um propósito (finalidade).

A finalidade ( $f$ ) é o objetivo no qual o contexto é utilizado em um determinado momento. Ela pode ser considerada a informação mais interessante para um agente em um determinado momento, e auxilia a definir como o agente deve agir.

Cada atributo de contexto ( $a$ ) que define o contexto contém uma informação que define um elemento do contexto, como “NomePessoa”, “PesoPessoa” e “BatimentoCardiacoPessoa”. Cada atributo de contexto possui no mínimo um valor a cada momento para cada entidade. Uma entidade é uma instância de uma “pessoa, objeto ou lugar” (conforme definido na seção 2.1.1), mas também pode ser uma atividade, organização ou conceito. Por exemplo, o atributo de contexto “EquipamentosDisponiveisSala” define os equipamentos disponíveis em determinada sala.

Cada atributo de contexto ( $a$ ) pode conter um conjunto discreto de valores que é denominado  $Va$ . Também é associada a cada atributo de contexto uma função de instanciação denominada *valueOf*, que é definida como o produto cartesiano  $A$  e  $Pa$  para  $P(Va)$ , onde  $A$  é o conjunto de todos os atributos de contexto,  $Pa$  é o conjunto de parâmetros para se computar os valores de  $a$ , e  $P(Va)$  é o conjunto potência de  $Va$ .

Como nem todos os atributos são relevantes para uma finalidade, é definido o predicado *isRelevant(a,f)*, que define que o atributo de contexto ( $a$ ) é relevante para a

finalidade ( $f$ ). O subconjunto de atributos de contexto ( $A$ ) para uma determinada finalidade ( $f$ ) é definido por  $RAS^{18}(f) = \{(a \in A) \mid isRelevant(a,f) = true\}$ .

A instanciação do atributo de contexto ( $a \in A$ ) é o par  $(a, v)$ , onde  $v$  é o conjunto de valores  $v \in P(Va)$  de  $a$  em um determinado momento. O conjunto de atributos de contexto instanciados é definido por  $I = \{(a,v) \mid (a \in A) \wedge valueOf(a) = v\}$ .

O conjunto de atributos de contexto relevantes instanciados para determinada finalidade ( $f$ ) é definido por  $IRAS^{19}(f) = \{(a,v) \mid a \in RAS(f) \wedge (a,v) \in I\}$ .

A representação de contexto é baseada nos modelos Entidade-Associação-Atributo e Ontologia, onde a representação é feita através de propriedades, classes e instâncias.

Os atributos de contexto são representados por uma classe, denominada `#ContextAttribute`, que possui sempre o mesmo tipo de informação: o nome do atributo, número e lista de tipo das entidades necessárias para sua instanciação, o  $Va$  (domínio de valores), e se o atributo pode ter diversos valores ou somente um.

Nome da Propriedade	Tipo da Propriedade	Domínio	Faixa	Múltiplos Valores
Name	DataType	#ContextAttribute	String	Não
NoEntities	DataType	#ContextAttribute	Integer	Não
EntitiesList	Object	#ContextAttribute	#Entity	Sim
ValueType	Object	#ContextAttribute	#Entity	Não
MultipleValue	DataType	#ContextAttribute	Boolean	Não

**Tabela 4 – Descrição da classe #ContextAttribute (BUCUR; BEAUNE; BOISSIER, 2005c)**

A classe `#ContextAttribute`, descrita na Tabela 4, é uma descrição geral das características de todos os atributos de contexto. Cada atributo de contexto

<sup>18</sup> RAS é *Relevant Attribute Set*.

<sup>19</sup> IRAS é *Instantiated Relevant Attribute Set*.

posteriormente será representado como uma instância desta classe, com as devidas restrições em cada uma das propriedades.

A ontologia para a representação de contexto é dada por dois componentes: Ontologia de Domínio (*Domain Ontology*) e Ontologia de Contexto (*Context Ontology*). A ontologia de domínio define todos os conceitos que serão utilizados no sistema. A ontologia de contexto contém todos os atributos de contexto que serão gerenciados.

### 3.3. Arquitetura

A arquitetura em camadas proposta, conforme Figura 5, é composta de agentes aprendizes sensíveis ao contexto<sup>20</sup> que auxiliam um usuário e interagem com um *Context Manager* (CM) local. Cada CM gerencia uma ontologia de domínio e um conjunto de atributos de contexto, disponibilizando aos agentes informações relacionadas ao contexto no qual o agente se encontra naquele determinado momento. É tarefa do CM computar os valores dos atributos de contexto, enquanto os agentes são responsáveis por: (1) aprender a reconhecer informações de contexto relevantes; e (2) agir (tomar ações) baseados em tais informações.

#### 3.3.1. Gerenciador de Contexto

O Gerenciador de Contexto (*Context Manager* – CM) possui duas funções: (1) Permitir aos agentes que entram no sistema, obter quais são os atributos de contexto definidos na ontologia e os que são gerenciados pelo CM; e (2) computar os valores (IRAS) de atributos relevantes (RAS) passados por um agente.

O CM é formado por 3 componentes: (1) *Context Knowledge Base* (CKB) que contém a ontologia de domínio e todos os atributos de contexto; (2) *Instantiation Module* que computa os valores (IRAS(*f*)) de um conjunto de atributos relevantes (RAS(*f*)); e (3) *Dependencies Module* que computa os valores para contextos derivados considerando as possíveis relações entre atributos de contexto.

---

<sup>20</sup> Do inglês *Context-based Learning Agent*.

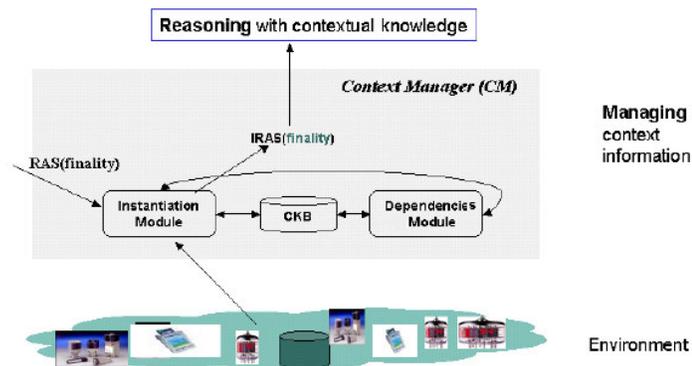


Figura 5 – Arquitetura do Context Manager (BUCUR; BEAUNE; BOISSIER, 2005c)

### 3.3.2. Agente Aprendiz Sensível ao Contexto

Um agente aprendiz sensível ao contexto, conforme mostrado na Figura 6, é constituído por dois componentes principais: (1) Módulo de Seleção (*Selection Module*) que disponibiliza os atributos de contexto relevantes para uma finalidade ( $RAS(f)$ ); e (2) Módulo de Decisão (*Decision Module*) que raciocina sobre os valores das informações de contexto relevantes, computadas pelo CM e selecionadas pelo Módulo de Seleção, para tomar decisões. O conjunto de instâncias de informações de contexto ( $IRAS(f)$ ) auxilia o agente a tomar decisões.

Ambos os módulos possuem métodos de aprendizagem associados, individuais e/ou cooperativos, para melhorar seu comportamento. O método de aprendizagem individual refere-se ao aprendizado do agente utilizando somente informações relacionadas ao seu usuário. Já o método de aprendizagem cooperativo é realizado através do compartilhamento do conhecimento entre agentes distintos.

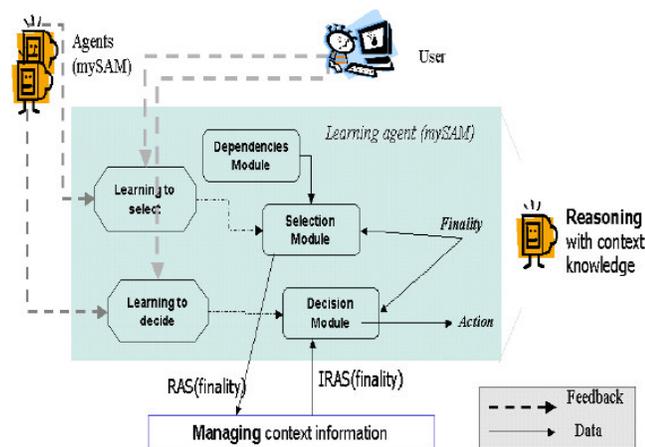


Figura 6 – Arquitetura do Agente Aprendiz (BUCUR; BEAUNE; BOISSIER, 2005c)

### Aprendizagem para Seleção

Os métodos de aprendizagem relacionados com o Módulo de Seleção têm por finalidade aprender dentre o universo total de atributos de contexto disponíveis quais são aqueles relevantes para uma determinada situação. A aprendizagem auxilia na restrição destes atributos, diminuindo assim o esforço de processamento do sistema e tornando-o mais eficiente.

No caso de aprendizagem individual, os usuários determinam quais são os atributos de contexto relevantes para determinada situação. Quando o agente se encontra em situação idêntica, os atributos de contexto são apresentados ao usuário, que pode então remover ou incluir novos atributos, o que também é registrado pelo agente. Após a seleção dos atributos de contexto que são relevantes ( $RAS(f)$ ), o Módulo de Seleção solicita ao CM a instanciação dos atributos ( $IRAS(f)$ ) para que o Módulo de Decisão possa tomar uma ação.

No caso de aprendizagem multiagentes, pode ser utilizado o método de compartilhamento de conhecimento entre agentes. Quando um agente não sabe determinar quais atributos são relevantes para uma determinada situação, ele solicita auxílio aos outros agentes, que respondem baseados em suas experiências individuais. Os atributos de contexto recebidos pelo agente solicitante geram um conjunto de atributos que é apresentado ao usuário para refinamento. Como a lista de

atributos de contexto relevantes para situação é um subconjunto do conjunto global, a seleção pelo usuário se torna uma tarefa mais ágil e fácil.

### **Aprendizagem para Tomada de Decisão**

Os métodos de aprendizagem relacionados com o Módulo de Decisão têm por finalidade melhorar o processo de tomada de decisão dos agentes, tornando-os cada vez mais personalizados ao usuário.

No caso de aprendizagem individual, pode-se utilizar qualquer método de aprendizagem disponível em IA para agentes inteligentes.

No caso de aprendizagem multiagentes, alguns problemas são encontrados como a necessidade dos agentes de (1) terem o mesmo entendimento de como utilizar as informações de contexto e o conhecimento, (2) se comunicarem utilizando a mesma linguagem e codificação da informação, e (3) compartilharem ao menos parte da ontologia de domínio para que entendam o conteúdo das mensagens trocadas. O auxílio à tomada de decisão pode ocorrer de duas formas distintas: (1) compartilhamento da solução – o agente solicitante recebe como resposta dos outros agentes da sociedade somente suas decisões, mas não como raciocinaram para chegar àquela decisão; ou (2) compartilhamento do conhecimento – o agente solicitante recebe como resposta dos outros agentes tanto suas decisões como também como raciocinaram para tomá-la. Na aprendizagem multiagentes, pode-se utilizar qualquer método de aprendizagem disponível em IA para SMA (WEISS; DILLENBOURG, 1999) (PANAIT; LUKE, 2005).

## 4. EXEMPLO DE APLICAÇÃO

Neste capítulo, ilustra-se o uso da arquitetura de sistemas sensíveis ao contexto apresentada no capítulo anterior em uma aplicação para academia de ginástica, sendo apresentados o cenário, o contexto e a arquitetura (componentes, características e distribuição) da aplicação.

### 4.1. Cenário

Suponha-se que Maria matricula-se em uma academia de ginástica denominada “Gym”. A política da academia é:

- prover atendimento personalizado aos alunos;
- ter número reduzido de funcionários;
- ser uma academia de vanguarda.

Para auxiliar a academia em sua política, é disponibilizado aos alunos um agente assistente pessoal, denominado *Trainer*, que deve ser carregado em um PDA<sup>21</sup>. Cada agente *Trainer* auxilia na sugestão e acompanhamento das atividades físicas realizadas por um determinado aluno, além de também auxiliá-lo com a configuração dos equipamentos de ginástica. Para cada aluno, também é disponibilizado um *kit* de sensores como sensor de batimentos cardíacos, pressão arterial, temperatura corporal, etc.

A academia de ginástica está equipada com um sistema de informação denominado SIAG (Sistema de Informação de Academia de Ginástica). Tal sistema é capaz de disponibilizar informações relacionadas ao ambiente da academia (como temperatura, umidade, etc), mapa detalhado da área (como banheiros, equipamentos, salas, etc), localização dos funcionários, equipamentos disponíveis e outras informações relacionadas ao contexto da academia.

---

<sup>21</sup> *Personal Digital Assistant.*

Maria entra pela primeira vez na academia após sua matrícula, momento no qual carregou o agente *Trainer* em seu PDA. O *Trainer* ao detectar que Maria entrou no contexto da academia, apresenta uma lista dos serviços disponíveis.

Inicialmente, o único serviço disponível para Maria em seu PDA (Figura 7) é Sugestão de Exercícios.



**Figura 7 – Personal Digital Assistant**

Maria seleciona o serviço de Sugestão de Exercícios. O agente *Trainer* desconhece os atributos relevantes para o objetivo de Maria. Ele interage com os outros agentes e aprende com a experiência dos outros agentes que os atributos relevantes para esta situação são características e preferências pessoais (IMC, gordura corporal, tempo disponível para realização de atividades, etc), prescrição médica (exercícios recomendados, tempo médio por exercício, etc) e equipamentos disponíveis (esteira, bicicleta, etc). Baseado nestes atributos de contexto, o agente *Trainer* sugere alguns exercícios para Maria em ordem de relevância, e disponibiliza informações extra como equipamentos atualmente disponíveis, total de equipamentos daquele tipo disponíveis na academia, número de pessoas com o mesmo objetivo realizando o mesmo exercício, número médio de calorias/hora que se perde realizando este exercício. Maria seleciona o exercício em que perderá o maior número de calorias/hora. O *Trainer* sugere alguns equipamentos deste tipo, disponíveis no momento, e indica a sua localização no mapa. Maria caminha até um dos equipamentos (Figura 8).



**Figura 8 – Esteira Ergométrica com Monitor LCD**

Antes de iniciar a atividade, Maria seleciona o serviço de Monitoramento de Saúde, que foi disponibilizado em seu PDA quando se aproximou do equipamento. O *Trainer* desconhece as regras de monitoração de saúde para uma pessoa com as características físicas de Maria. Ele então, através da interação com outros agentes de mesmo tipo, aprende que deve notificar a aluna quando ela estiver em situação de risco, sendo os atributos a considerar: (1) o batimento cardíaco ultrapassar 195 batimentos/segundo, (2) a temperatura corporal ultrapassar 40°C, (3) não houver nenhum funcionário próximo, ou (4) o tempo de permanência no equipamento ultrapassar o recomendado. A lista de atributos é apresentada para Maria, que adiciona um limite máximo de perda de calorías de 200 calorías/hora: caso a perda de calorías ultrapasse este limite, deseja ser notificada.

Esta mesma lista de atributos possui uma sugestão de configuração do equipamento como velocidade, inclinação, seqüência de músicas e/ou possíveis vídeos a serem apresentados. Ela faz algumas alterações com relação à seqüência musical e solicita o início da atividade. Durante a atividade, Maria resolve alterar as configurações do equipamento para aumentar a velocidade e inclinação da esteira. Como ela desconhece este equipamento, solicita que o agente *Trainer* ajuste a configuração conforme os parâmetros preenchidos.

No decorrer do período, o agente *Trainer* detecta que o número de calorias e os batimentos cardíacos de Maria estão fora dos limites estabelecidos. Conseqüentemente, notifica Maria sobre o fato e já lhe sugere alterações na velocidade e inclinação da esteira. Maria aceita as sugestões e o equipamento é alterado conforme o sugerido. Neste mesmo momento, o Personal Trainer de Maria é notificado sobre o fato.

#### 4.2. Representação do Contexto

O contexto nesta arquitetura é representado através de uma ontologia. A ontologia para a representação de contexto é dada por dois componentes, conforme explicado anteriormente na seção 3.2: Ontologia de Domínio (*Domain Ontology*) e Ontologia de Contexto (*Context Ontology*).

##### Ontologia de Domínio

Na ontologia de domínio da aplicação de academia de ginástica apresentada na Figura 9, é definida uma superclasse denominada #Entity, sendo #Sala, #Atividade, #Equipamento, #Exercicio, #Pessoa, #Vídeo, #Musica, etc, suas subclasses e estando relacionadas entre si. Cada uma das classes descritas é composta por atributos, por exemplo, o atributo *nome* na classe #Pessoa.

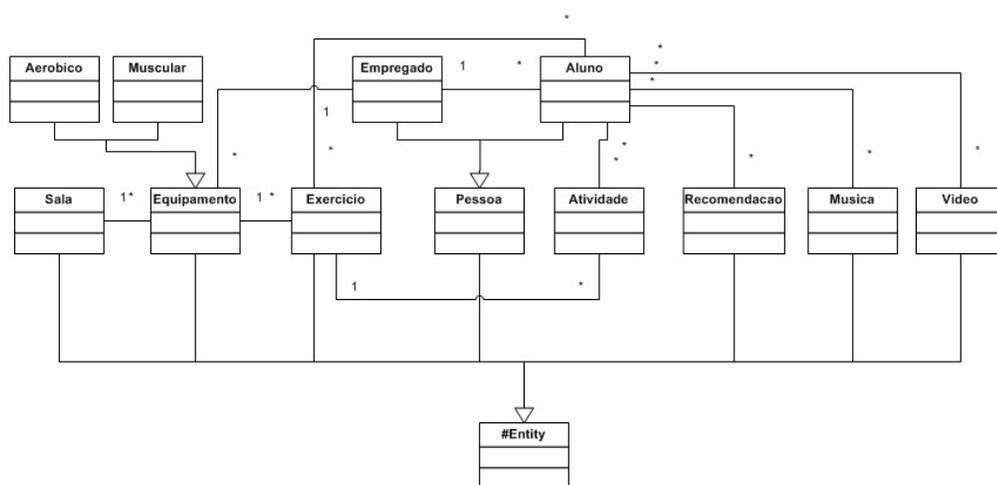


Figura 9 – Ontologia de Domínio da Academia de Ginástica

### Ontologia de Contexto

De acordo com a definição do conceito de “atributo de contexto” apresentada na seção 3.2, o atributo de contexto encapsula todas as informações necessárias para definir e instanciar uma informação de contexto. É definida uma classe denominada #ContextAttribute que define uma descrição comum para todos os atributos de contexto. O conjunto dos atributos de contexto definidos através da classe #ContextAttribute é a Ontologia de Contexto.

Na Tabela 5 abaixo é apresentada uma lista dos principais atributos de contexto da aplicação da academia de ginástica.

<b>Pessoa</b>	<b>Atividade</b>
IsPersonalTrainerOf(Pessoa) -> Pessoa StatusPessoa(Pessoa) -> String NomePessoa(Pessoa) -> String PesoPessoa(String) -> Float BatimentoCardiacoPessoa(Pessoa) -> Float	AtividadeInicia(Atividade) -> Time AtividadeDuracao(Atividade) -> Time AtividadeTermina(Atividade) -> Time
<b>Sala</b>	<b>Ambiente</b>
PessoasSala(Sala) -> Pessoa* EquipamentoSala(Sala) -> Equipamento* EquipamentosDisponiveisSala(Sala) -> Equipamento*	TemperaturaAmbiente(Ambiente) -> Float UmidadeAmbiente(Ambiente) -> Float
<b>Equipamento</b>	
EquipamentoDisponivel -> Equipamento* EquipamentoProximo(Equipamento, Equipamento) -> Boolean VelocidadeCorrida(Equipamento) -> Float	

**Tabela 5 – Tabela de Atributos de Contexto da Academia de Ginástica**

\* Indica que são múltiplos os resultados.

### 4.3. Arquitetura

Dada a representação do contexto de uma academia de ginástica apresentada na seção anterior, uma possível arquitetura de sistema sensível ao contexto conforme descrito na seção 3, seria composta por: (1) Fontes de Contexto; (2) Gerenciador de Contexto, que disponibiliza as informações de contexto aos agentes; e (3) diversos agentes denominados *Trainer*, conforme mostra a Figura 10.

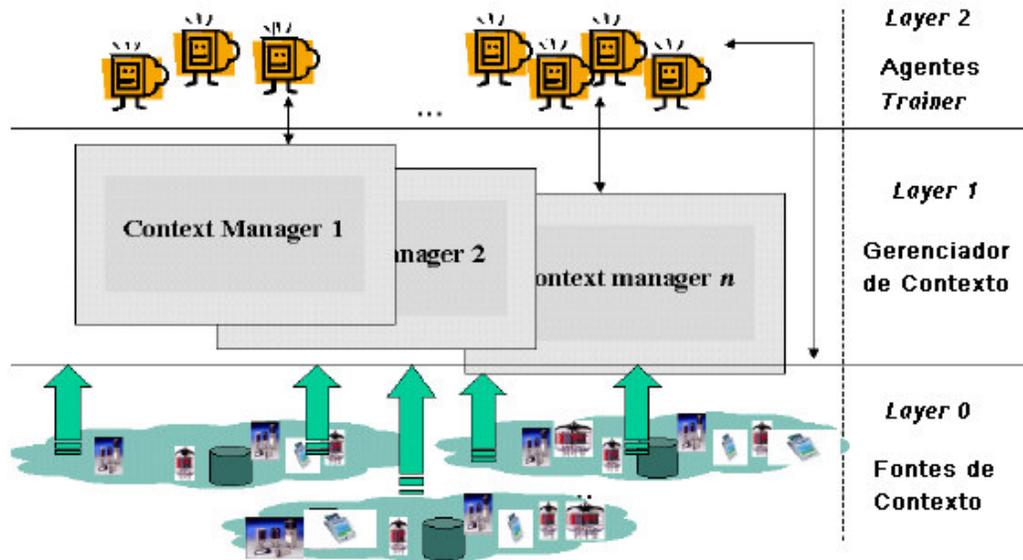


Figura 10 – Arquitetura da Aplicação de Academia de Ginástica

#### 4.3.1. Fontes de Contexto

Esta primeira camada é composta por uma coleção de sensores distribuídos, que geram dados em sua forma bruta para posterior processamento pelo Gerenciador de Contexto.

Para a aplicação de academia de ginástica, os sensores coletam dados do ambiente como temperatura e umidade, dados dos alunos como pressão, batimentos cardíacos, temperatura corporal, etc., e dados dos equipamentos de ginástica como velocidade, distância percorrida, calorias perdidas, etc.

Todos estes dados são transmitidos ao Gerenciador de Contexto para que sejam disponibilizados ao agente *Trainer*, o qual é responsável pela tomada de decisões e disponibilização de serviços aos alunos.

### 4.3.2. Gerenciador de Contexto

Conforme descrito na seção 3.3.1, o gerenciador de contexto deveria ser composto de três módulos: *Context Knowledge Base* (CKB), *Instantiation Module* e *Dependencies Module*.

O módulo CKB do CM é composto de duas ontologias: (1) a ontologia de domínio, definida como uma hierarquia sendo #Entity a raiz; e (2) a ontologia de contexto, definida como subclasses da classe #ContextAttribute.

O *Instantiation Module* é responsável por computar os valores (IRAS( $f$ )) dos atributos relevantes (RAS( $f$ )) passado por um agente *Trainer*.

O *Dependencies Module* não é considerado nesta arquitetura, pois não são tratados os casos de relação entre atributos de contexto para a geração de atributos de contexto derivados.

O Gerenciador de Contexto é o componente intermediário da arquitetura proposta, neste caso denominado de SIAG (Sistema de Informação de Academia de Ginástica). Ele é o componente responsável pela gestão dos contextos adquiridos a partir dos sensores distribuídos no ambiente da academia, nos equipamentos e nos alunos, e responsável por responder as solicitações dos agentes *Trainer* para a computação dos valores dos atributos determinados como relevantes.

Este componente fica localizado em uma máquina servidora com capacidade para processar todas as informações de contexto coletadas a partir dos sensores distribuídos na academia e relacioná-las com a ontologia de domínio.

### 4.3.3. Agente *Trainer*

O agente *Trainer* é o componente da arquitetura que utiliza as informações de contexto disponibilizadas pelo Gerenciador de Contexto para prover serviços aos alunos da academia. O agente *Trainer* fica instalado no PDA de cada aluno e

comunica-se com outros agentes e com o Gerenciador de Contexto através de mensagens.

Este agente, com o tempo, adapta-se às características pessoais de seu usuário, assim provendo informações e auxiliando-o na tomada de decisões. A adaptação ocorre através da aprendizagem do comportamento e preferências do usuário e através da troca de experiência com os outros agentes do sistema, utilizando-se ainda de informações de contexto disponibilizadas pelo Gerenciador de Contexto.

Para a troca de experiência com outros agentes do sistema, o agente *Trainer* dispara mensagens solicitando atributos relevantes a determinado objetivo (REQUEST\_ATTRIBUTE) e solicitando as regras de comportamento dos agentes em determinadas situações (REQUEST\_RULE). Como resultado, os outros agentes informam os atributos que consideram relevantes para o objetivo passado (INFORM\_ATTRIBUTE) ou disponibilizam a regra de comportamento que consideram relevante para o objetivo passado (INFORM\_RULE), respectivamente. No caso da requisição de regra de comportamento, o agente pode se recusar a informar seu comportamento por considerar esta ação uma quebra de segurança, assim recusando-se a informar sua regra de conhecimento (REFUSE\_REQUEST).

Além da comunicação com os outros agentes do sistema, o agente *Trainer* necessita manter-se em constante comunicação com o Gerenciador de Contexto para a obtenção dos valores dos atributos de contexto relevantes na tomada de decisão. Esta solicitação carrega consigo os atributos de contexto relevantes que devem ser instanciados (REQUEST\_VALUES) e recebe como resultado os valores dos atributos de contexto (INFORM\_VALUES). O agente *Trainer* também pode registrar alguns atributos de contexto no Gerenciador de Contexto (REGISTER\_VALUES\_CHANGE), assim quando o valor dos atributos de contexto se altera, o agente é notificado (INFORM\_VALUES\_CHANGE) para tomar alguma ação.

O agente *Trainer* é responsável por prover serviços aos alunos da academia. Os serviços disponibilizados são: (1) auxiliar o aluno na definição dos exercícios mais adequados ao seu perfil físico e objetivo (Sugestão de Exercícios), e (2) monitoração da saúde do aluno (Monitoramento de Saúde).

### **Serviço de Sugestão de Exercícios**

Quando o aluno seleciona pela primeira vez o serviço de Sugestão de Exercícios, o agente *Trainer* não sabe quais são os atributos de contexto relevantes para o objetivo do aluno. Assim, o agente *Trainer* dispara uma mensagem a todos os agentes do ambiente solicitando os atributos relevantes para o objetivo do aluno. Este método de aprendizagem dos atributos de contexto relevante foi denominado aprendizagem por Interação (Aprendizagem Cooperativa) na seção 2.3.5. Após receber os atributos mais relevantes sugeridos pelos outros agentes, o agente *Trainer* realiza a união dos conjuntos de atributos e apresenta o resultado ( $RAS(f)$ ) para o aluno. O aluno tem a opção de incluir ou retirar os atributos que considera necessário e solicita os valores dos atributos relevantes definidos. O agente *Trainer* solicita ao Gerenciador de Contexto (*Context Manager*) o cálculo dos valores dos atributos relevantes. O Gerenciador de Contexto recebe como parâmetro o conjunto de atributos relevantes ( $RAS(f)$ ) e gera como resultado um conjunto contendo os valores dos atributos ( $IRAS(f)$ ). Recebido o conjunto de valores dos atributos de contexto ( $IRAS(f)$ ), o agente utiliza-se de uma técnica para tomada de decisão como o *Case-Based Reasoning* (CBR) ou *Classification Based on Association* (CBA), e disponibiliza uma lista de exercícios ordenados por relevância para que o aluno decida qual sua preferência.

Sendo executada pela segunda vez pelo mesmo aluno, a fase de aprendizagem para seleção não é mais executada, pois o agente já é capaz de determinar os atributos relevantes para a situação. Assim, os atributos relevantes definidos anteriormente são apresentados para alteração. Caso não haja nenhuma alteração, os valores dos atributos são calculados pelo Gerenciador de Contexto, processados pelo agente *Trainer* e como resultado é apresentada uma lista de sugestões de exercícios ao aluno

ordenada baseado na relevância. A seqüência de mensagens trocadas entre os diversos elementos da arquitetura é mostrada na Figura 11.

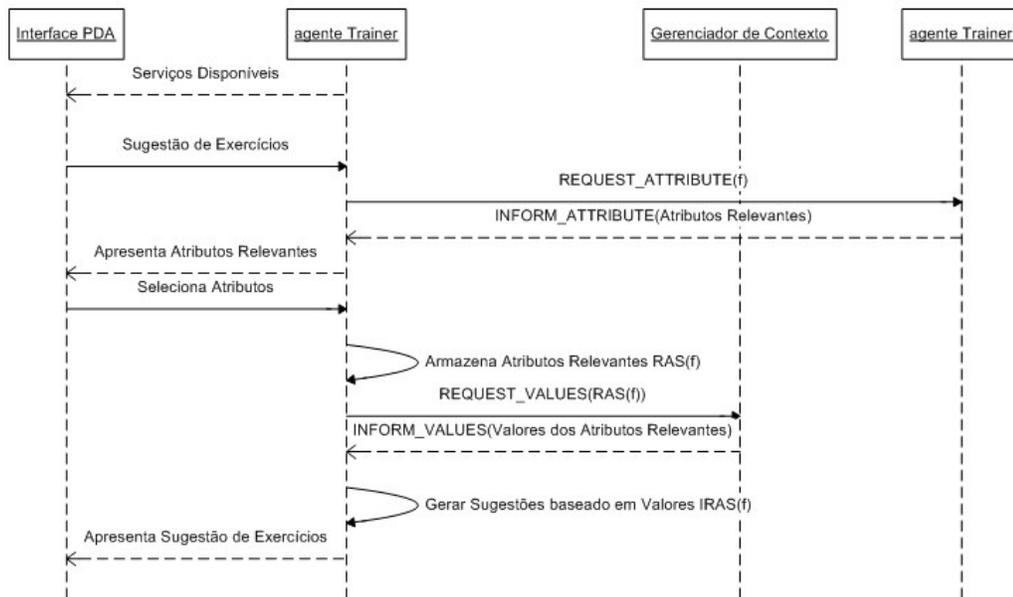


Figura 11 – Diagrama de Seqüência da Sugestão de Exercícios

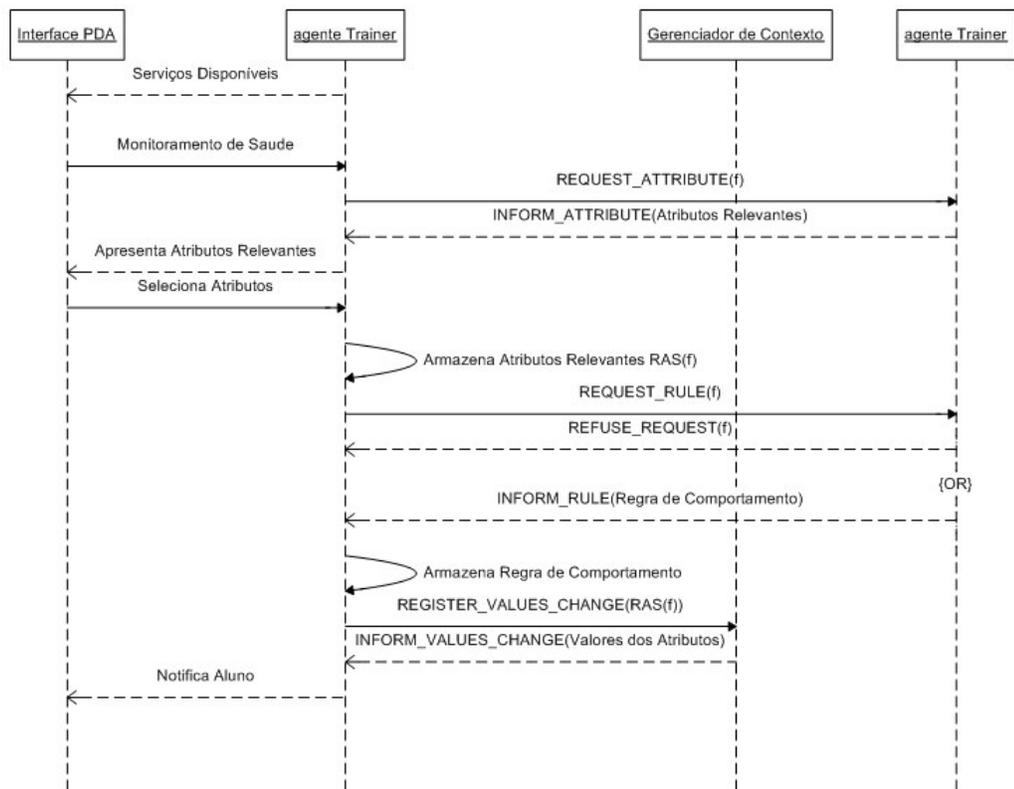
### Serviço de Monitoramento de Saúde

Quando o aluno seleciona pela primeira vez o serviço de Monitoramento de Saúde, o agente *Trainer* não sabe quais são os atributos de contexto relevantes para o objetivo do aluno. Assim, o agente *Trainer* dispara uma mensagem a todos os agentes do ambiente solicitando os atributos relevantes para o objetivo do aluno. Este método de aprendizagem dos atributos de contexto relevante foi denominado aprendizagem por Interação (Aprendizagem Cooperativa) na seção 2.3.5. Após receber os atributos mais relevantes sugeridos pelos outros agentes, o agente *Trainer* realiza a união dos conjuntos de atributos e apresenta o resultado ( $RAS(f)$ ) para o aluno. O aluno tem a opção de incluir ou retirar os atributos que considera necessário (aprendizagem para seleção baseado em *feedback*), e solicita os valores dos atributos relevantes definidos. O agente *Trainer* solicita ao Gerenciador de Contexto (*Context Manager*) o cálculo dos valores dos atributos relevantes. O Gerenciador de Contexto recebe como parâmetro o conjunto de atributos relevantes ( $RAS(f)$ ) e gera como resultado

um conjunto contendo os valores dos atributos ( $IRAS(f)$ ). Recebido o conjunto de valores dos atributos de contexto ( $IRAS(f)$ ), o agente solicita, devido ao seu desconhecimento, aos outros agentes a regra de comportamento para o objetivo do aluno. Os agentes respondem com suas regras (conhecimento) e o agente *Trainer* seleciona a que considerar mais relevante. Após a seleção da regra de comportamento, utiliza-se de técnicas como o *Case-Based Reasoning* (CBR) ou *Classification Based on Association* (CBA) para o monitoramento da saúde do aluno.

Caso detecte-se no decorrer da atividade física, através da notificação feita pelo Gerenciador de Contexto da alteração de algum dos valores dos atributos relevantes ( $IRAS(f)$ ), que o comportamento do aluno não se encontra dentro dos padrões desejados, o agente *Trainer* notifica o usuário sobre a situação e lhe sugere alterações nas configurações do equipamento.

Quando o Monitoramento de Saúde é executado pela segunda vez pelo mesmo aluno, as fases de aprendizagem para seleção e para tomada de decisão não são mais executadas, pois o agente já é capaz de determinar os atributos relevantes e como tomar decisões nesta situação. Assim, tanto os atributos relevantes definidos anteriormente, quanto a regra de comportamento obtida, são utilizados para os monitoramentos posteriores. A seqüência de mensagens trocadas entre os diversos elementos da arquitetura é mostrada na Figura 12.



**Figura 12 – Diagrama de Seqüência do Monitoramento de Saúde**

## **5. CONCLUSÕES**

O uso de sistemas multiagentes no desenvolvimento de sistemas sensíveis ao contexto é uma possibilidade viável e permite a redução da complexidade destes últimos devido à descentralização e delegação de tarefas inerentes aos SMA.

Os SMA têm uma maior aderência na camada de inferência dos sistemas sensíveis ao contexto, pois é nesta camada que eles podem tirar vantagens das informações de contexto e ainda obter experiência de outros agentes para auxiliar a tomada de decisões.

O esboço da arquitetura de um sistema sensível ao contexto aplicado a uma academia de ginástica possibilitou apresentar a real utilização das técnicas de SMA no desenvolvimento de tais sistemas.

## ANEXO A – ESPECIFICAÇÃO DA ONTOLOGIA DA APLICAÇÃO

Neste anexo, será apresentada uma tabela (Tabela 6) com todos os atributos das subclasses da ontologia de domínio da aplicação apresentada no capítulo 4.

SubClasse	Atributo	Tipo
Atividade	data	date
	duracao	float
	horaInicio	time
	hasAluno	Aluno
	hasExercicio	Exercicio
Equipamento	id	int
	funcionamento	boolean
	livre	boolean
	locateAtSala	Sala
	proximo	Equipamento*
	responsavel	Empregado*
Aerobico	id	int
	funcionamento	boolean
	livre	boolean
	locateAtSala	Sala
	proximo	Equipamento*
	responsavel	Empregado*
	capacidadeMaxima	float
Muscular	id	int
	funcionamento	boolean
	livre	boolean
	locateAtSala	Sala
	proximo	Equipamento*
	responsavel	Empregado*
Exercicio	needEquipamento	Equipamento
	objetivo	string
	perdaCalorias	float
Musica	nomeMusica	string
	tipoMusica	string
Pessoa	numIdentificacao	int
	nome	string
	idade	int
	email	string
Aluno	numIdentificacao	int
	nome	string
	idade	int
	email	string
	altura	float

	batimentosCardiacos	float
	objetivo	string
	percentualGordura	float
	peso	float
	personalTrainer	Empregado
	prefExercicio	Exercicio*
	prefMusica	Musica*
	prefVideo	Video*
hasRecomendacao	Recomendação*	
Empregado	numIdentificacao	int
	nome	string
	idade	int
	email	string
	salario	float
	responsavelEquipamento	Equipamento*
Recomendacao	exercicioRecomendado	Exercicio
	tempoRecomendado	int
Sala	hasEquipamento	Equipamento*
	umidade	float
	latitude	float
	longitude	float
	temperatura	float
Video	nomeVideo	string
	tipoVideo	string

Tabela 6 – Atributos das Subclasses da Ontologia de Domínio

## LISTA DE REFERÊNCIAS

(ABOWD et al., 1999) ABOWD, G. D. et al. Towards a Better Understanding of Context and Context-Awareness. *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing.*, Karlsruhe, 1999. London: Ed. Lecture Notes In Computer Science, v.1707. Springer-Verlag, p.304-307.

(ABOWD; MYNATT, 2000) ABOWD, G. D.; MYNATT, E. D. Charting Past, Present and Future Research in Ubiquitous Computing. In: *ACM Transactions on Computer-Human Interaction, Special Issue on HCI in the new Millenium*, p.29-58, March 2000.

(AILISTO et al., 2002) AILISTO, H. et al. Structuring Context Aware Applications: Five-Layer Model and Example Case. In: Workshop on Concepts and Models for Ubiquitous Computing. *UBICOMP2002*: Göteborg, September 2002.

(ALVARES; SICHTMAN, 1997) ALVARES, L. O.; SICHTMAN, J. S. Introdução aos Sistemas Multiagentes. In: *Jornada de Atualização em Informática*, 16. Congresso da SBC, 17. 1997, Brasília, Anais..., Brasília: SBC, p.1-37, 1997.

(AUSTIN, 1962) AUSTIN, J. L. *How Do Things with Words*. London: Oxford University Press, 1962.

(BALDAUF; DUSTDAR; ROSENBERG, 2004) BALDAUF, M.; DUSTDAR, S.; ROSENBER, F. *A Survey on Context-Aware Systems*. Technical Report TUV-1841-2004-24, 2004.

(BAZIRE; BRÉZILLON, 2005) BAZIRE, M.; BRÉZILLON, P. Understanding Context Before Using It. In: *The Fifth International and Interdisciplinary Conference on Modeling and Using Context*. CONTEXT-05, France, 6-8 July 2005.

(BUCUR; BEAUNE; BOISSIER, 2005a) BUCUR, O.; BEAUNE, P.; BOISSIER, O. Providing Context-Awareness Using Multi-Agent Systems in Ambient Intelligence. In: *The Fifth International and Interdisciplinary Conference on Modeling and Using Context*. CONTEXT-05, France, 6-8 July 2005.

(BUCUR; BEAUNE; BOISSIER, 2005b) BUCUR, O.; BEAUNE, P.; BOISSIER, O. Representing the Context in a Multi-Agent System for Decision Making. In: *Proceedings of the 4th International Joint Conference on Autonomous Agents & Multi Agent Systems, Workshop notes Ambient Intelligence – Agents for Ubiquitous Environments*, p.17-28, 2005.

(BUCUR; BEAUNE; BOISSIER, 2005c) BUCUR, O.; BEAUNE, P.; BOISSIER, O. Steps Towards Making Contextualized Decisions: How to Do What You Can, With What You Have, Where You Are. *Research Report 2005-700-013*. Ecole Nationale Supérieure des Mines – Saint-Etienne. September 2005. Submitted to International Journal of Pervasive Computing and Communications, August 2005.

(CHEN et al., 2001) CHEN, H. et al. Creating Context-Aware Software Agents. In: *First GSFL/JPL Workshop on Radical Agent Concepts*. NASA Goddard Space Flight Center, Greenbelt, MD, USA, p.26-28, September 2001.

(CHEN, 2004) CHEN, H. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. 129p. Tese (Doutorado) – University of Maryland. Baltimore County, 2004.

(FERBER; GASSER, 1991) FERBER, J.; GASSER, L. Intelligence Artificielle Distribuée. In: *International Workshop on Expert Systems & Their Applications*, 10, Avignon, Cours n. 9, France: [s.n], 1991.

(FIPA, 2005) *The Foundation for Intelligent Physical Agents*. Disponível em <<http://www.fipa.org>>. Acesso em: Novembro 2005.

(GARCIA; SICHMAN, 2002) GARCIA, A. C. B.; SICHMAN, J. S. Agentes e Sistema Multiagentes. In: Solange Oliveira Rezende. *Sistemas Inteligentes – Fundamentos e Aplicações*. Barueri: Manole, p.269-306, 2002.

(HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002) HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY, A. Modeling Context Information in Pervasive Computing Systems. In: F. Mattern e M. Naghshineh. *Pervasive 2002*. Heidelberg: Springer-Verlag Berlin, LNCS, v.2414, p.167-180, 2002.

(KIF, 2005) *Knowledge Interchange Format*. Disponível em: <<http://logic.stanford.edu/kif/dpans.html>>. Acesso em: Novembro 2005.

(KQML, 2005) *Knowledge Query and Manipulation Language*. Disponível em: <<http://www.cs.umbc.edu/kqml/>>. Acesso em: Novembro 2005.

(KORPIPÄÄ et al., 2003) KORPIPÄÄ, P. et al. Managing Context Information in Mobile Devices. In: *IEEE Pervasive Computing*, v.2, n.3, p.42-51, July-September 2003.

(LECH; WIENHOFEN, 2005) LECH, T. C.; WIENHOFEN, L. W. M. AmbieAgents: A Scalable Infrastructure for Mobile and Context-Aware Information Services. In: *AAMAS '05*, Netherlands. p.605-631, July 25-29, 2005.

(LYYTINEN; YOO, 2002) LYYTINEN, K.; YOO, Y. Issues and Challenges in Ubiquitous Computing. In. *Communications of the ACM*, v.12, n.12, p.62-65, December 2002.

(MOSTÉFAOUI; PASQUIER-ROCHA; BRÉZILLON, 2004) MOSTÉFAOUI, G. K.; PASQUIER-ROCHA, J.; BRÉZILLON, P. Context-Aware Computing: A Guide for the Pervasive Computing Community. In: *Proceedings of the IEEE/ACS International Conference on Pervasive Services*, 2004.

(PANAIT; LUKE, 2005) PANAIT, L.; LUKE, S. Cooperative Multi-Agent Learning: The State of the Art. In: *Autonomous Agents and Multi-Agent Systems*, v.11, p.387-434, 2005.

(SALBER; DEY; ABOWD, 1999) SALBER, D.; DEY, A. K.; ABOWD, G. D. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: *Proceedings of the 1999 Conference on Human Factors in Computing Systems*. p.434-441, 1999.

(SHADBOLT, 2003) SHADBOLT, N. Ambient Intelligence. In: *IEEE Intelligent Systems*, p.2-3, July/August 2003.

(SCHILIT; ADAMS; WANT, 1994) SCHILIT, B. N.; ADAMS, N.; WANT, R. Context-Aware Computing Applications. In: *Proceedings of the 1<sup>st</sup> International Workshop on Mobile Computing Systems and Applications*, p.85-90, 1994.

(SCHILIT; THEIMER, 1994) SCHILIT, B. N.; THEIMER, M. M. *Disseminating Active Map Information to Mobile Hosts*. IEEE Networks, 8(5), p.22-32, 1994.

(SEARLE, 1969) SEARLE, J. R. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press, 1969.

(SICHMAN; DEMAZEAU; BOISSIER, 1992) SICHMAN, J. S.; DEMAZEAU, Y; BOISSIER, O. When Can Knowledge-Based Systems be Called Agents? In: *Anais do IX SBIA*, p.172-185, Rio de Janeiro, 1992.

(STRANG; LINNHOF-POPIEN, 2004) STRANG, T.; LINNHOF-POPIEN, C. A Context Modeling Survey. In: *First International Workshop on Advanced Context Modeling, Reasoning and Management*, England, September 2004.

(SYCARA, 1998) SYCARA, K. P. Multiagent Systems. In: *American Magazine*, American Association of Artificial Intelligence (AAAI), p.79-92, 1998.

(USCHOLD; GRUNINGER, 1996) USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, Methods and Applications. In: *The Knowledge Engineering Review*, v.11, n.2, p.93-136, 1996.

(WEISER, 1991) WEISER, M. The computer for the 21<sup>st</sup> century. In: *Scientific American*, p.94-1994, September 1991.

(WEISS; DILLENBOURG, 1999) WEISS, G.; DILLENBOURG, P. What is “multi” in Multi-Agent Learning?. In: *Dillenbourg (Ed) Collaborative-Learning: Cognitive and Computational Approaches*, p.64-80, 1999.

(WOOLDRIDGE, 1996) WOOLDRIDGE, M. J. *An Introduction to Multiagent Systems*. England: John Wiley & Sons Ltd. England, 1996.