



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada

Mestrado Acadêmico

Jezer Machado de Oliveira

Global: Uma Infra-Estrutura Descentralizada para
Ambientes de Educação Ubíqua

São Leopoldo, 2010

UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA

JEZER MACHADO DE OLIVEIRA

**Global: Uma Infra-Estrutura Descentralizada para
Ambientes de Educação Ubíqua**

Dissertação de mestrado submetida à avaliação
como requisito parcial para obtenção do grau
de Mestre em Computação Aplicada

Prof. Dr. Jorge Luis Victoria Barbosa
Orientador

São Leopoldo, Fevereiro de 2010

AGRADECIMENTOS

Primeiramente, agradeço a meus pais, que tiveram ao meu lado durante todo esse trabalho ajudando de todas as formas possíveis. E que durante toda a minha vida sempre me deram apoio tanto na parte pessoal quanto na minha educação.

Agradeço aos colegas de mestrado que sempre colaboraram com trocas de ideias e sugestões para melhoria desse trabalho. Ao orientador Jorge Barbosa, por me convidar para participar desse projeto e pelo apoio durante todo ele. Por fim, agradeço ao CNPQ que me disponibilizou a bolsa de estudos que viabilizou a realização do mestrado.

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Motivação	12
1.2	Definição do Problema	13
1.3	Objetivos e Contribuição	14
1.4	Organização do Trabalho	14
2	CONCEITUAÇÃO	15
2.1	Computação Ubíqua	15
2.2	Educação Ubíqua	16
2.3	Agentes e Sistema Multi-agentes	18
2.4	Metodologia Prometheus	19
2.5	Considerações sobre o Capítulo	20
3	TRABALHOS RELACIONADOS	21
3.1	Japelas	21
3.2	Framework de interação social com suporte a aprendizagem ubíqua	22
3.2.1	Modelo de aprendizagem ubíqua	23
3.2.2	Suporte a interação social entre aprendizes	24
3.3	Ambiente de aprendizagem ubíqua para a aprendizagem colaborativa P2P ..	24
3.3.1	Gerência de contexto	25
3.3.2	Estrutura	25
3.4	GlobalEdu	26
3.4.1	Arquitetura	26
3.4.2	<i>Middleware</i> de suporte à execução	27
3.5	Avaliação dos Ambientes Pesquisados	28
3.6	Considerações sobre o Capítulo	29
4	INFRA-ESTRUTURA PROPOSTA	30
4.1	Visão Geral	30
4.1.1	Conectividade	31
4.1.2	Contexto	33
4.1.3	Perfil do Usuário	36
4.1.4	Perfis	37
4.1.5	Objetos de Aprendizagem	38

4.1.6	Comunicação	40
4.1.7	Persistência	41
4.1.8	Interface com Usuário.....	41
4.1.9	Proxies	41
4.1.10	Restritores	42
4.2	Modelagem em Prometheus.....	43
4.2.1	Diagrama geral do sistema.....	43
4.2.2	Detalhamento dos agentes	45
4.3	Considerações sobre o Capítulo.....	51
5	ASPECTOS DE IMPLEMENTAÇÃO.....	52
5.1	Estrutura geral do Global	52
5.2	Comunicação entre Agentes	54
5.3	Restrições.....	56
5.4	Persistência	57
5.5	Interface com o usuário.....	57
5.6	Global Agent Message Viewer	58
5.7	Considerações sobre o Capítulo.....	59
6	AVALIAÇÃO	60
6.1	Aplicação que implementa um ambiente de aprendizagem ubíquo existente	60
6.1.1	Arquitetura Original.....	60
6.1.2	Especialização do Global.....	62
6.1.3	Simulação.....	63
6.1.4	Conclusão.....	66
6.2	Integração com o CoolEdu	67
6.2.1	Arquitetura do CoolEdu.....	68
6.2.2	Requisitos de ambiente	68
6.2.3	Especialização do Global.....	69
6.2.4	Simulação.....	72
6.2.5	Conclusão.....	74
6.3	Global Social Network.....	74
6.3.1	Gerencia de perfil pessoal.....	74
6.3.2	Gerência de comunidades	75
6.3.3	Gerência de amigos.....	77
6.3.4	Conclusão.....	77

6.4	Considerações sobre o Capítulo.....	78
7	CONSIDERAÇÕES FINAIS	79
7.1	Contribuições	79
7.2	Trabalhos Futuros	81
	REFERÊNCIAS	82

Lista de Figuras

Figura 1 - Taxonomia da computação ubíqua (Adaptado de Satyanarayanan (2001))	15
Figura 2 - Metodologia Prometheus (Padgham and Winikoff, 2002)	19
Figura 3 - Estrutura Japelas (Yin et al., 2005)	21
Figura 4 - Modelo de aprendizagem ubíquo proposto por Zhang (2005)	23
Figura 5 - Encontro, Comunicação e Colaboração (Zhang 2005)	24
Figura 6 - Arquitetura GlobalEdu (Barbosa, 2007)	27
Figura 7 - Infra-estrutura do Global	30
Figura 8 - Estrutura do Contexto	35
Figura 9 - Estrutura PAPI	36
Figura 10 - Objetos de Aprendizagem	39
Figura 11 - Aquisição de Objeto de Aprendizagem	40
Figura 12 - Estrutura da comunicação	41
Figura 13 - Restrições	42
Figura 14 - Exemplo de restrições	43
Figura 15 - Diagrama geral do Global	44
Figura 16 - Protocolo de HandShake	45
Figura 17 - Detalhamento do agente de Conectividade	46
Figura 18 - Detalhamento do agente de Contexto	47
Figura 19 - Detalhamento do agente de Perfil do Usuário	48
Figura 20 - Detalhamento do agente de Perfis	48
Figura 21 - Protocolo de Verificação de atualização de perfil	49
Figura 22 - Detalhamento do agente de Objetos de Aprendizagem	49
Figura 23 - Detalhamento do agente de Comunicação	50
Figura 24 - Classe Global	52
Figura 25 - Estrutura dos Agentes	53
Figura 26 - Diagrama da classe de mensagens ACL	54
Figura 27 - FIPA-ACL: Especificação por String	55
Figura 28 - Restritores	56
Figura 29 - Persistência	57
Figura 30 - Global Agent Message Viewer	59
Figura 31 - Contexto interativo	61
Figura 32 - Nova estrutura	63

Figura 33 – Alunos A e B ingressam no contexto 1 (Sala 1)	64
Figura 34 - Alunos C, D e E ingressam no contexto 1 (Sala 1).....	65
Figura 35 - Alunos A e C deixam o contexto 1 (Sala 1) e ingressam no 2 (Sala 2).....	66
Figura 36 - CoolEdu	68
Figura 37 - Requisitos de ambiente CoolEdu.....	69
Figura 38 - Adaptação dos Agentes do CoolEdu	70
Figura 39 - Perfil pessoal da integração com CoolEdu	72
Figura 40 - Perfil Social	75
Figura 41 - Telas Perfil Social.....	75
Figura 42 - Informações das Comunidades	76
Figura 43 - Telas da Gerência de Comunidade	76

Lista de Tabelas

Tabela 1 - Legendas Prometheus (Padgham and Winikoff, 2002).....	20
Tabela 2 - Comparativo entre sistemas.....	28
Tabela 3 - Parâmetros FIPA-ACL.....	32
Tabela 4 - Especialização do Global para proposta de Hahn	67
Tabela 5 - Comparativo das funcionalidades	77
Tabela 6 - Comparativo com Global	80

RESUMO

Este trabalho apresenta o Global uma infra-estrutura descentralizada de educação ubíqua, baseada em sistemas multi-agentes. O Global disponibiliza agentes de *software* que executam tarefas comuns ao processo de aprendizagem ubíqua. A partir da extensão desses agentes ou adição de novos ele é especializado para a criação de ambientes de educação ubíqua.

Além de apresentar a arquitetura geral da infra-estrutura, descrevendo seus agentes e componentes auxiliares, é apresentada a modelagem do Global na metodologia Prometheus, uma metodologia para especificação de sistemas multi-agentes.

A dissertação também descreve o protótipo desenvolvido a partir da especificação do Global e apresenta três aplicações desenvolvidas utilizando o protótipo, com finalidade de avaliar o Global.

Palavras-chave: Computação Ubíqua, Educação Ubíqua, Infra-estrutura, Sistema Multi-Agentes.

ABSTRACT

This work presents Global, a decentralized, ubiquitous learning infrastructure based on multi-agent systems. Global provides software agents that execute tasks related to ubiquitous learning. By extending these agents, or adding new agents, Global can be customized to create ubiquitous learning environments.

We present a complete scheme of the general architecture, describing its agents and auxiliary components. We also fully model the system through the Prometheus methodology, which is used to specify multi-agent systems.

We developed a prototype based on the Global specification. Furthermore, we present three applications developed using this prototype, with the goal to perform an evaluation of our approach.

Keywords: Ubiquitous Computing, u-learning, Infrastructure, Multi-agent system, Decentralization.

1 INTRODUÇÃO

A computação ubíqua (Weiser, 1999), também conhecida como computação pervasiva (Satyanarayanan, 2001), é considerada uma extensão da computação móvel, herdando características como mobilidade e localização, com a adição de características próprias como sensibilidade a contexto e invisibilidade. Graças à evolução e proliferação de dispositivos eletrônicos portáteis com alta capacidade de processamento como *Smartphones*, *hand-helds*, *notebooks* e *tablet PCs*, que integram: sistemas de comunicação baseados em tecnologias sem fio tais como *Bluetooth* (Haartsen et al., 1998), *WiFi* (Ieee, 2010) e 3G (IMT-2000, 2010); sistema de localização mais baratos e eficientes (Hightower & Borriella, 2001), como GPS (Qualcomm, 2010) , AGPS (Global Locate, 2010) e triangulação de antena (Hightower et al., 2006)); e tecnologias de captura e reconhecimento de imagens (DENSO, 2010) foram desenvolvidos nos últimos anos uma gama de *softwares* ubíquos nas mais diversas áreas, como comércio (Galaxhi-Janaqi & Nah, 2004) e educação (Yau, 2003).

Neste contexto, a educação vem sendo considerada uma importante área de aplicação, pois a capacidade de um aluno, equipado com um dispositivo móvel, aprender enquanto se desloca por um ambiente, utilizar essas informações para contextualizar e agregar conhecimento a esse aprendizado, traz inúmeros benefícios ao processo de aprendizagem. Atualmente, existem diversas propostas para a organização de ambientes de educação ubíqua, como SmartClassroom (Yau, 2003), PERKAM (El-bishouty et al., 2007), LOCAL (Barbosa et al., 2008), JAPELAS (Yin et al., 2005) e GlobalEdu (Barbosa, 2007), assim como os trabalhos propostos por Chen (2007), Yang (2006) e Zhang (2005).

1.1 Motivação

As diversas propostas de ambientes de educação ubíqua, em sua maioria, utilizam estruturas centralizadas (Yang, 2006). Contudo, tal modelo não se mostra adequado a alguns conceitos da computação ubíqua definidos por Satyanarayanan (2001), que entre suas características requer redes móveis, alta disponibilidade, escalabilidade localizada e acesso móvel à informação. Em contrapartida modelos descentralizados assimilam mais facilmente tais características.

Em modelos centralizados tradicionais a comunicação é feita em uma infraestrutura pré-configurada através de um intermediário (servidor), que é responsável pela entrega da mensagem, o que dificulta a criação de redes móveis. Já em modelos descentralizados a comunicação é feita de forma direta entre usuários. Integrando tal

característica ao uso de algoritmos de roteamento descentralizados (redes *Ad Hoc* (Royer & Toh, 1999)) é possível a criação de redes móveis. Caso um servidor fique inoperante ou inacessível, os elementos gerenciados por ele também ficam ou pelo menos perdem parte de suas funcionalidades. Quando um elemento em um modelo descentralizado fica inoperante ou inacessível somente funcionalidades que dependem dele ficam comprometidas, garantindo uma maior disponibilidade.

Escalabilidade localizada é a capacidade de priorização de interação com usuários mais próximos. Em ambientes descentralizados baseados em redes *Ad Hoc* a própria topologia da rede favorece a escalabilidade localizada, limitando o número de saltos entre os nodos essa escalabilidade pode ser controlada. Em ambientes centralizados, para fazer essa gerência é necessária a participação do servidor o que reduz a escalabilidade. Em modelos descentralizados, devido à ausência de um servidor, as informações são armazenadas no próprio dispositivo garantindo o acesso móvel à informação.

Além disso, no desenvolvimento de ambientes de educação ubíqua para dispositivos móveis, outros dois fatores são relevantes:

- Mesmo com o avanço da tecnologia dos dispositivos móveis, as ferramentas e APIs para desenvolvimento ainda são bastante limitadas em comparação ao desenvolvimento de aplicações *desktop*, aliando tal fato à variedade de tecnologias, como comunicação *wireless* e o sistema de localização, necessárias para o desenvolvimento de aplicações ubíquas torna tal processo oneroso;
- *Softwares* de educação ubíqua apresentam funcionalidades comuns, entre elas se destacam gerência de perfis e comunicação entre usuários.

1.2 Definição do Problema

Dado o número de questões que devem ser tratadas para o desenvolvimento de aplicações descentralizadas de educação ubíqua, uma infra-estrutura de *software* pode ser utilizada para reduzir a complexidade envolvida no projeto destas aplicações, reduzindo os detalhes que devem ser considerados por desenvolvedores (Saha & Mukherjee, 2003).

Baseado-se nas referências estudadas pode-se afirmar que não existe uma infra-estrutura para modelos de educação ubíqua que seja descentralizada e agregue funcionalidades comuns a ambientes de educação ubíqua.

A partir dessas premissas, surge a seguinte questão de pesquisa:

- Existe a possibilidade de criação de uma infra-estrutura descentralizada para educação ubíqua ?

1.3 Objetivos e Contribuição

Este trabalho tem como objetivo definir, implementar e avaliar uma infra-estrutura descentralizada de educação ubíqua, baseado em sistemas multi-agentes, denominada Global (Oliveira et al., 2009).

O Global disponibiliza agentes de *software* (Shoham, 1990) que executam tarefas comuns ao processo de aprendizagem ubíqua. Sendo que sua customização para criação de ambientes de educação ubíqua é feita através da extensão desses agentes ou adição de novos.

A escolha de sistemas multi-agentes tem como vantagem uma maior capacidade de abstração. Além disso, cada agente é modelado de forma independente dos demais, simplificando assim a adição de agentes com novas funcionalidades ou a remoção de agentes com funcionalidades desnecessárias, que é uma das propostas do Global.

A seguir são apresentados os objetivos específicos deste trabalho:

- Especificar e modelar o Global utilizando uma metodologia de engenharia de *software* orientada a agentes;
- Desenvolver um protótipo;
- Validar a infra-estrutura a partir do protótipo implementado.

1.4 Organização do Trabalho

Este trabalho está organizado em seis capítulos. O capítulo 2 descreve conceitos necessários para o entendimento desse trabalho. Em seguida, o capítulo 3 apresenta trabalhos relacionados com o trabalho proposto. O capítulo 4 apresenta a infra-estrutura proposta e sua modelagem na metodologia Prometheus. O capítulo 5 apresenta os aspectos de implementação. O capítulo 6 descreve a avaliação da infra-estrutura. O capítulo 7 apresenta as considerações finais da dissertação.

2 CONCEITUAÇÃO

Este capítulo visa elucidar conceitos de computação ubíqua, educação ubíqua, agentes e sistema multi-agentes, bem como apresentar a metodologia Prometheus, uma metodologia para especificação de sistemas multi-agentes que será utilizada no trabalho.

2.1 Computação Ubíqua

A computação ubíqua, também conhecida como computação pervasiva, pode ser considerada uma extensão da computação móvel, pois herdou algumas de suas características com a adição de características próprias, como sensibilidade a contexto e invisibilidade. Da mesma forma que a computação móvel pode ser considerada uma extensão da área de sistemas distribuídos com adição de características como redes móveis e localização (Satyanarayanan, 2001).

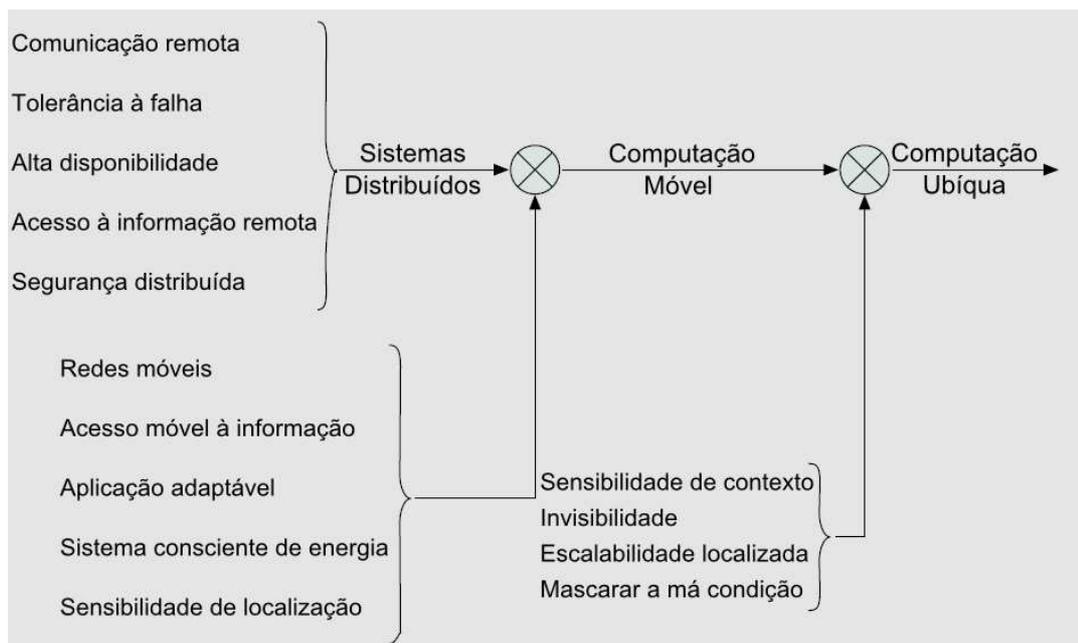


Figura 1 - Taxonomia da computação ubíqua (Adaptado de Satyanarayanan (2001))

Entre as características desejadas em sistemas ubíquos, tanto herdadas de sistemas distribuídos e de computação móvel, como as suas próprias (Figura 1), se destacam:

- **Redes móveis:** capacidade de trocar informação entre dispositivos móveis sem a necessidade de uma infraestrutura de comunicação centralizada, por exemplo, redes *Ad Hoc* e comunicação ponto a ponto (P2P);
- **Acesso móvel à informação:** capacidade de operar mesmo desconectado, pois as informações ficam armazenadas no próprio dispositivo;

- Sensibilidade de localização: capacidade de localização física no ambiente, posicionamento do dispositivo no espaço;
- Segurança distribuída: garantia de privacidade da informação, autenticação, autenticidade dos usuários, segurança na troca de informação;
- Sensibilidade ao contexto: capacidade de alterar o comportamento do sistema através de informações percebidas do ambiente que o dispositivo móvel se encontra, tais como localização física e sensores de temperatura;
- Escalabilidade localizada: capacidade de reduzir a interação entre objetos que se encontram distantes entre si aumentando, assim, a escalabilidade da aplicação;
- Mascarar a má condição: capacidade de se adaptar a alterações no ambiente de execução, como redução da capacidade de processamento e redução ou perda de conectividade;
- Invisibilidade: busca distrair o mínimo o usuário, evitando interações desnecessárias. Em alguns momentos a aplicação pode alertar o usuário de possíveis problemas em outros mascarar as falhas.

2.2 Educação Ubíqua

A educação ubíqua, também conhecida como *u-learning*, consiste na aplicação da computação ubíqua no processo de aprendizagem sendo uma evolução da educação móvel (M-learning) (Jones & Jo, 2004).

Essa aplicação permite a construção de programas de aprendizagem relacionados com questões dinâmicas do contexto do aprendiz, possibilitando que o contexto seja vinculado com os objetivos pedagógicos tendo como foco o aprendiz e sua interação com o meio. Na educação ubíqua é idealizado que os processos educacionais ocorram em qualquer tempo e em qualquer lugar de forma contínua e contextualizada (Bomsdorf, 2005).

Ogata (2004) (baseado nos trabalhos de Chen (2002) e Curtis (2002)) define como características principais da educação ubíqua ideal:

- Permanência: os alunos nunca perdem o seu trabalho a menos que seja propositalmente excluído. Além disso, todo o processo de aprendizagem é registrado de forma contínua todos os dias;
- Acessibilidade: os alunos têm acesso a seus documentos, dados e vídeos em qualquer lugar. Essa informação é fornecida com base em seus pedidos. Portanto, a aprendizagem envolve auto-direcionamento;

- Imediatismo: onde quer que os alunos estejam, eles podem obter qualquer informação imediatamente. Assim, os alunos podem resolver problemas rapidamente. Caso contrário, o aluno pode gravar as questões e procurar a resposta mais tarde;
- Interatividade: os alunos podem interagir com professores ou colegas. Assim, o conhecimento torna-se mais disponível;
- Situando as atividades educacionais: a aprendizagem pode ser incorporada em nossa vida diária. Os problemas encontrados, bem como os conhecimentos necessários são todos apresentados em sua forma natural e autêntica.

Barbosa (2007) após estudos na área destaca quatro principais elementos que caracterizam um ambiente ideal para o suporte à educação ubíqua:

- Aprendiz: o sistema deve conhecer os aprendizes e associá-los a um modelo que os representem e, assim, proporcionar maior qualidade nas informações disponibilizadas para todos. Este modelo procura representar o que o sistema julga conhecido pelo aprendiz e que ele tem interesse em aprender;
- Mobilidade: o aprendiz tem a capacidade de se mover entre vários ambientes, aprendendo de forma constante, independente de onde esteja. Desta forma, é importante identificar esta mobilidade e permitir que os recursos educacionais do aprendiz, para construção de sua aprendizagem, o “acompanhem” em seu percurso, de forma adaptada, mantendo a continuidade do processo. Conhecendo a localização do aprendiz e sua mobilidade, procura-se identificar o que melhor se adapta as suas condições de aprendizagem e a percepção dos elementos que compõem seu contexto de interesse;
- Conteúdo: o sistema deve permitir que o aprendiz aprenda qualquer coisa, em qualquer tempo, com qualquer dispositivo. Dessa forma, é desejável que o sistema suporte a representação de conteúdo genérico, independente de domínio. Com o auxílio do sistema, o aprendiz pode escolher os conteúdos que mais se identificam com seus objetivos e se adaptam ao seu contexto. Os conteúdos devem estar acessíveis de qualquer lugar, disponíveis em vários formatos, considerando a heterogeneidade de dispositivos e redes de comunicação;
- Ciência do contexto: a mobilidade do aprendiz traz a possibilidade de aprendizagem em diferentes contextos. O sistema deve definir, representar e gerenciar os elementos que representam o contexto do aprendiz. Com isso, os elementos que integram as

diversas localizações onde se encontra o aprendiz podem ser relacionados com seus objetivos educacionais. Para isso, é importante representar elementos de contexto genéricos, que possam ser utilizados para representar o maior número de localizações possíveis. Ainda, devido à dinamicidade dos ambientes ubíquos, é importante que o sistema ofereça mecanismos de assistência ao aprendiz no ambiente. Esses mecanismos agem pró-ativamente, percebendo, filtrando e fornecendo informações ao aprendiz conforme seu perfil, em direção a uma interação mais significativa.

2.3 Agentes e Sistema Multi-agentes

Um agente de *software* é uma entidade de *software* que funciona de maneira contínua e autônoma em um ambiente particular com a habilidade de conduzir atividades de forma extensível, inteligente e sensível a mudanças no ambiente (Bradshaw, 1997).

Com intuito de aumentar o nível de abstração (Gomes, 2005), o conceito de agentes de *software* propõe uma extensão do modelo orientado a objeto, no qual agentes são sistemas integrados que internamente incorporam um maior número de capacidades de diferentes áreas (inteligência artificial, banco de dados, linguagens de programação) e externamente estabelecem comunicação com outras aplicações através de uma linguagem independente das características específicas da estrutura de cada aplicação (Serain, 2002).

Apesar de não existir uma definição formal de agente, se assume que as características que todo agente de *software* deve ter são (Miller, 1998):

- Reatividade: agentes percebem seu ambiente (físico ou computacional) e respondem de maneira adequada às mudanças que nele ocorrem;
- Autonomia: capacidade de agir sem interferências externas de usuários ou de outros sistemas;
- Interatividade: os agentes são capazes de se comunicar com outros agentes e com o ambiente onde se encontram;
- Pró-atividade: agentes não agem simplesmente em resposta ao seu ambiente, eles são capazes de exibir um comportamento direcionado a objetivos.

Os sistemas multi-agentes são compostos de vários agentes que se comportam de forma autônoma e interagem entre si, permitindo assim, a coordenação, a cooperação, a competição e a negociação entre eles. O principal conceito dos sistemas multi-agentes é que a partir do comportamento individual de cada agente é que o grupo de agentes alcançará o seu

grau de comportamento inteligente. Assim, nestes sistemas não existe a necessidade de cada agente ser inteligente para que se alcance o comportamento global inteligente (Russell, 2004).

2.4 Metodologia Prometheus

Prometheus (Padgham & Winikoff, 2002) é uma metodologia detalhada de especificação para sistemas multi-agentes que vem sendo desenvolvida por Lin Padgham e Michael Winikoff (RMIT University, Austrália) em colaboração com o Grupo de *Software Orientado a Agentes* (AOS). A metodologia Prometheus é dividida em três fases, ilustradas na Figura 2.

Na fase de especificação do sistema (*System Specification*) deve-se especificar o que o sistema deve ser capaz de fazer (*System Goals*), desenvolver cenários de caso de uso que ilustrem a operação do sistema, identificar as funcionalidades e especificar uma interface com o ambiente em termos de ações e percepções.

A partir das saídas da fase de especificação, a fase de desenvolvimento da arquitetura (*Architectural Design*) determina quais serão os agentes e como será a interação entre eles.

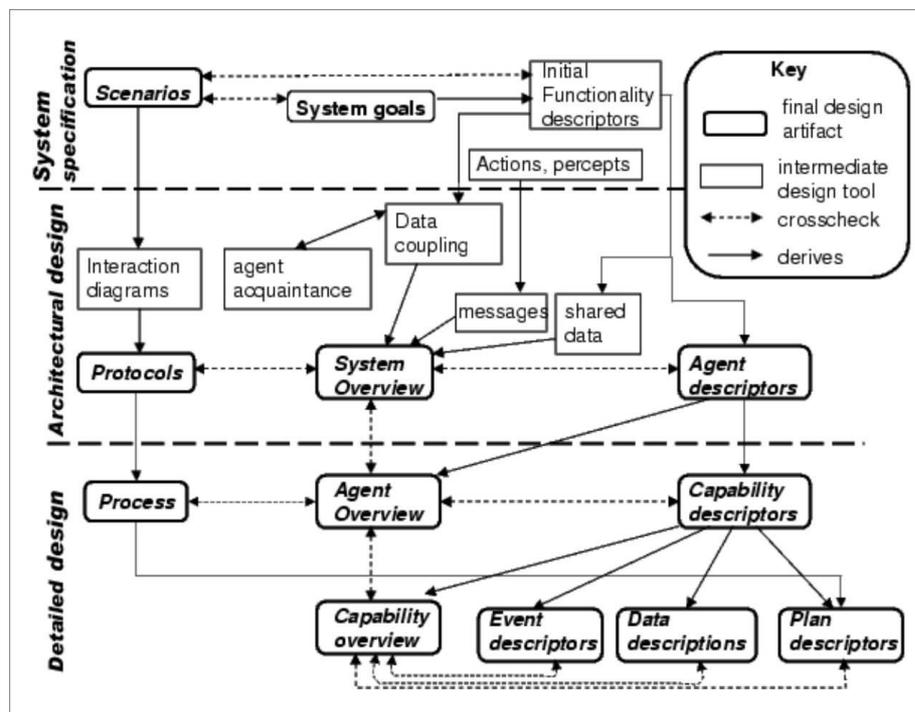


Figura 2 - Metodologia Prometheus (Padgham & Winikoff, 2002)

Na fase de projeto detalhado (*Detailed Design*), desenvolve-se a estrutura interna de cada agente e define-se como cada um realizará sua tarefa dentro do sistema. Refina-se cada agente definindo capacidades, eventos internos, planos e dados acessados.

A Tabela 1 mostra o significado dos símbolos utilizados na especificação.

Tabela 1 - Legendas Prometheus (Padgham & Winikoff, 2002)

Símbolo	Significado	Símbolo	Significado
	Agente		Mensagem
	Ação		Dados
	Percepção		Protocolo
	Ligação		Capacidade

2.5 Considerações sobre o Capítulo

Este capítulo apresentou conceitos de computação ubíqua, educação ubíqua, agentes e sistema multi-agentes, bem como apresentou a metodologia Prometheus que será utilizada no trabalho. No próximo capítulo serão apresentados trabalhos relacionados à proposta do Global, fazendo uma comparação de suas características.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta quatro ambientes de educação ubíqua, que são descentralizados ou apresentam alguma característica descentralizada. Por fim, é feita uma avaliação dos ambientes estudados.

3.1 Japelas

Japelas (Yin et al., 2005) é um sistema que possibilita o aprendizado de expressões de tratamento na língua Japonesa. Os estudantes, portando dispositivos móveis, são assistidos na tarefa de identificar as expressões adequadas para cada contexto. Japelas utiliza tecnologias de localização e perfis de usuário, mas a abordagem empregada é voltada para uma aplicação específica, não apresentando, por exemplo, uma estrutura que facilite adição de novos idiomas. Existem duas propostas mais recentes de ambientes de educação ubíqua baseadas no Japelas, CLU (Ogata & Yano, 2004) e ULE (Li et al., 2005), que apresentam estruturas que facilitam a extensão de seus componentes. No entanto, ambas as propostas tem um modelo cliente/servidor.

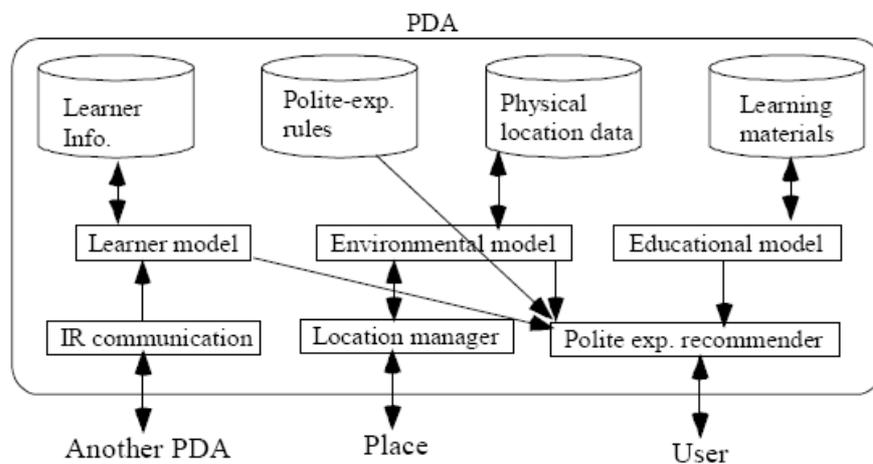


Figura 3 - Estrutura Japelas (Yin et al., 2005)

A Figura 3 mostra a estrutura do Japelas, que é dividida em seis módulos:

- Modelo do aprendiz (*Learner model*): este módulo tem o perfil do aluno contendo nome, idade, sexo, ano de escola, amigos e parentes. Antes de utilizar este sistema, cada aluno cadastra esses dados. Além do método explícito de entrada de dados, o

sistema detecta interesses de aprendizado do aluno de acordo com seu histórico de utilização;

- Modelo do ambiente (*Environmental model*): este módulo possui os dados de salas em uma área. A sala é detectada pelo gerente de localização usando *tag's* RFID (Feldhofer et al., 2004). A localização é usada para determinar o nível de formalidade, por exemplo, salas de reuniões estão propensas a situações mais formais. Se o aluno entra numa sala de reunião, são fornecidas expressões mais formais e as distâncias sociais são levadas em conta;
- Modelo educacional (*Educational model*): este módulo gerencia expressões como conteúdo pedagógico. Professores cadastram as expressões básicas e alunos e professores podem acrescentar ou modificar expressões durante a utilização do sistema;
- Comunicação IR (*IR Communication*): a comunicação por IR (infravermelho) (Ghassemlooy & Hayes, 2000) não precisa de nenhuma infra-estrutura fixa ou configuração. Além disso, simplifica a designação do alvo da comunicação. Em vez de inserir o nome do alvo, os utilizadores podem simplesmente apontar para a pessoa;
- Gerenciador de Localização (*Location manager*): com etiquetas RFID, este módulo detecta a localização do aluno, por exemplo, loja, sala privada e casa. Etiquetas RFID são anexadas nas portas de entrada das salas para identificá-las;
- Recomendador de expressões (*Polite exp. recommender*): com base nas regras de expressões, este módulo fornece a expressão adequada para a situação.

O ambiente é descentralizado, visto que o seu único elemento computacional é o PDA e a interação com outros PDAs é feita de forma direta, usando comunicação por IR.

3.2 Framework de interação social com suporte a aprendizagem ubíqua

Zhang (2005) propôs um *framework* de aprendizagem ubíqua e interação social entre aprendizes. Além disso, ele também define um modelo para a construção de interação social na aprendizagem em ambiente ubíquo, que inclui três funções principais: encontro, comunicação e colaboração.

3.2.1 Modelo de aprendizagem ubíqua

A Figura 4 mostra o modelo de aprendizagem que é dividido em duas partes, centro de suporte à aprendizagem (*learning support center side*) e aprendizado individual (*individual learner side*). O centro de suporte à aprendizagem é composto pela base de dados e três módulos de suporte à aprendizagem:

- Módulo computacional de suporte a aprendizagem (*Computer supported learning module*): esse módulo abrange a função de fornecimento estável de conteúdo e a função de gestão da aprendizagem;
- Módulo de apoio à aprendizagem (*Learning support module*): proporciona aprendizado personalizado, tal como a política de apoio ao aprendizado individual;
- Módulo social (*Social module*): fornece as competências sociais e princípios para o apoio à construção da aprendizagem social comunitária.

As mídias utilizadas no aprendizado individual podem ser PDAs, telefones celulares, computadores e dispositivos embarcados. Os aprendizes podem se comunicar através de uma rede de aprendizado.

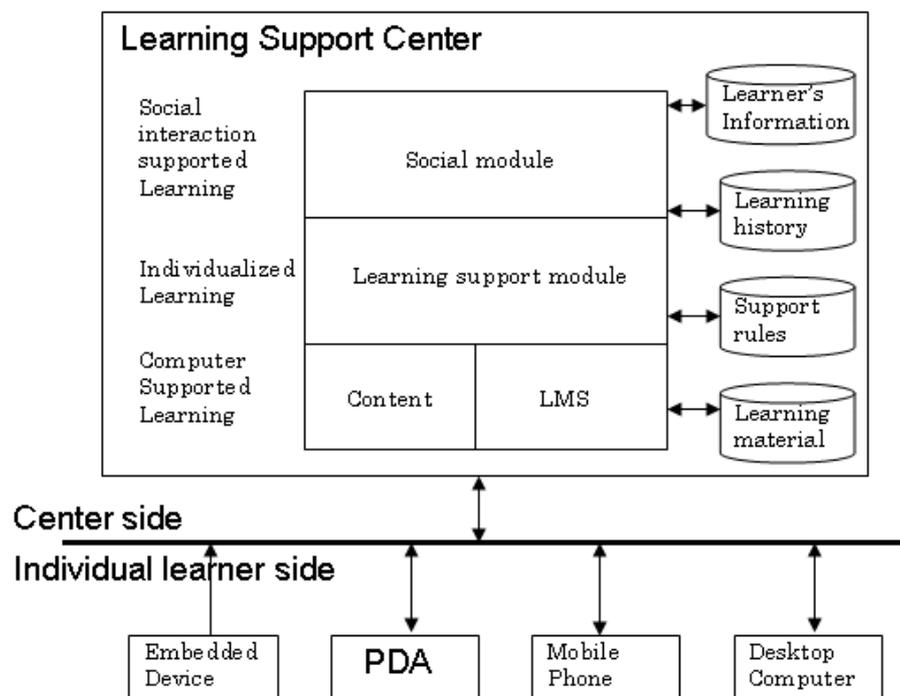


Figura 4 - Modelo de aprendizagem ubíquo proposto por Zhang (2005)

3.2.2 Suporte a interação social entre aprendizes

O modelo de interação social utilizado nesse *framework* é o modelo da “função de encontro” proposto por Jin (2003), que é dividido em três partes (Figura 5):

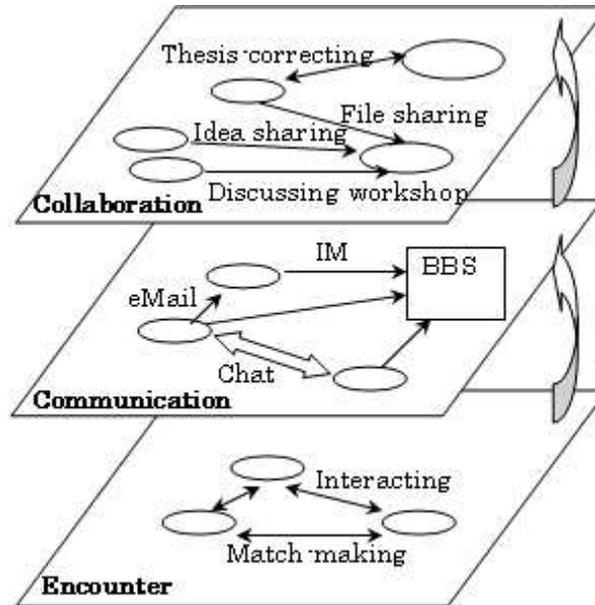


Figura 5 - Encontro, Comunicação e Colaboração (Zhang 2005)

- Encontro (*Encounter*): inicialmente os aprendizes precisam saber da presença de outros aprendizes. Mas um aprendiz pode não estar interessado em todos os aprendizes *on-line*. Então no processo de encontro, grupos são formados dinamicamente segundo vários atributos, como o perfil do aluno, interesses, e histórico de aprendizagem;
- Comunicação (*Communication*): suportada por e-mail, espaço de discussão coletivo, e comunicadores instantâneos;
- Colaboração (*Collaboration*): fornece aos aprendizes um ambiente colaborativo.

Apesar do *framework* criado por Zhang propor aprendizagem ubíquo, ele não se enquadra na definição de ubiqüidade proposta por Satyanarayanan (2001), pois não possui suporte à mobilidade ou sensibilidade ao contexto.

3.3 Ambiente de aprendizagem ubíqua para a aprendizagem colaborativa P2P

Yang (2006) propôs um ambiente de aprendizagem ubíqua para a aprendizagem colaborativa ponto a ponto (P2P). O modelo realiza o apoio ao aprendizado através da criação de comunidades virtuais onde instrutores e alunos descobrem, acessam e compartilham recursos, se comunicam e participam de discussões em grupo.

3.3.1 Gerência de contexto

O ambiente considera a sensibilidade ao contexto como um modelo interativo entre aprendizes e serviços, com base nisso são propostas duas ontologias, uma para descrever os aprendizes outra para descrição dos serviços. Com base nessas duas ontologias um modelo de interatividade através do casamento de padrões realiza raciocínio semântico para realizar descoberta e acesso aos serviços orientados a contexto. Além disso, o ambiente propõe outras duas formas de aquisição de contexto: através da indicação direta do aprendiz e por sensores, como de posição física (GPS) e dados (RFID). A detecção do contexto é dividida em duas partes: lado cliente e lado servidor. No lado cliente são coletadas as informações, já o lado do servidor é responsável por processar essas informações para definir o contexto do aprendiz. Os contextos são divididos em correntes e passados, onde são registrados os contextos que o aprendiz esteve. Essa informação pode ser utilizada para obter preferências, portfólios e padrões comportamentais.

3.3.2 Estrutura

O ambiente é dividido em três subsistemas: subsistema de acesso a conteúdo e adaptação ponto a ponto, subsistema de gerência de anotação personalizada e subsistema multimídia de grupos de discussão em tempo real.

O subsistema de acesso a conteúdo e adaptação ponto a ponto, por intermédio de uma rede P2P, acessa e disponibiliza acesso aos recursos pedagógicos mantidos por cada nodo na rede. Caso esse subsistema não encontre o recurso requisitado por um vizinho na rede, ele solicita para seus próprios vizinhos (redes *Ad Hoc*). O sistema também permite que um nodo seja mediador, uma referência para aquisição de certo tipo de recurso. Os recursos pedagógicos são descritos por perfis contextuais, que contém informações como papel e características do recurso.

As anotações no subsistema de gerência de anotação não são apenas lembretes de coisas a fazer, mas também conceitos e pensamentos. O documento anotado pode ser um arquivo PDF, Word e quaisquer páginas em HTML e as anotações podem ser criadas em formato de texto ou voz. Existem sete tipos de anotações no subsistema, questão, explicação, comentário, favorito, esboço, desenho, e link.

No subsistema multimídia de grupos de discussão em tempo real, o serviço de mensagem é projetado do ponto de vista da colaboração em grupo, disponibilizando serviços como, fóruns de discussão, *instant messenger*, filtro de mensagens e sincronização de mensagens dentro de grupos de colaboração. Cada usuário tem a liberdade de iniciar um

grupo de interesse especial (*special interest group* - SIG) e é livre para se inscrever para participar de qualquer SIG iniciado por outros colegas na rede. O subsistema também disponibiliza recursos de conferência com áudio e vídeo e agenda pessoal ou de grupo de serviço.

O ambiente não é totalmente descentralizado, necessitando de intervenção de um servidor centralizado para determinar o contexto do aprendiz.

3.4 GlobalEdu

O GlobalEdu (Barbosa, 2007) segue uma proposta pedagógica construtivista e interacionista (Moll & Silveira, 1999; Becker, 2001), tratando o conhecimento como um processo em permanente construção, que se transforma na interação do sujeito com o meio e a partir da ação deste sobre o objeto de conhecimento. Seu objetivo é propiciar as condições necessárias para que o aprendiz construa seu conhecimento de qualquer lugar e em qualquer tempo, de forma adaptada ao seu perfil e contexto. Para tanto, o processo educacional considera que o aprendiz (sujeito) é parte do processo e, como ser autônomo responsável pela sua aprendizagem (objeto).

3.4.1 Arquitetura

A arquitetura do GlobalEdu provê uma organização em Camadas (Figura 6). A Camada de Aplicação é representada por um Agente Pedagógico (AP) (Barbosa et al., 2006), cujo objetivo é auxiliar a interação do aprendiz no ambiente. Na Camada de Sistema, a arquitetura propõe um conjunto de módulos necessários para auxiliar o processo educacional do aprendiz no ambiente ubíquo, manipulando o contexto do aprendiz, seu perfil e seus conteúdos, além de elementos que auxiliem a execução do AP e dos recursos que ele manipula. Os módulos são organizados em Educacionais e de Suporte. Os Módulos Educacionais são responsáveis pelo armazenamento e gerenciamento das informações educacionais manipuladas no GlobalEdu, executando tarefas em dispositivos estáveis da arquitetura. Assim, executam funções específicas relacionadas ao perfil do aprendiz, do conteúdo e do contexto, notificando o AP sempre que uma informação relevante está disponível. Os Módulos de Suporte são responsáveis pelos elementos que auxiliam a execução do AP e pelos demais módulos da arquitetura. Um módulo de suporte especial é o de Comunicação, responsável pela comunicação do GlobalEdu com o *middleware* de execução, abstraindo a complexidade de comunicação com a Camada de Execução. Esse método também provê uma maior portabilidade do GlobalEdu. Na Camada de Execução tem-

se o *middleware* de suporte à execução. O modelo considera que o ambiente de execução provê os elementos necessários para o seu funcionamento.

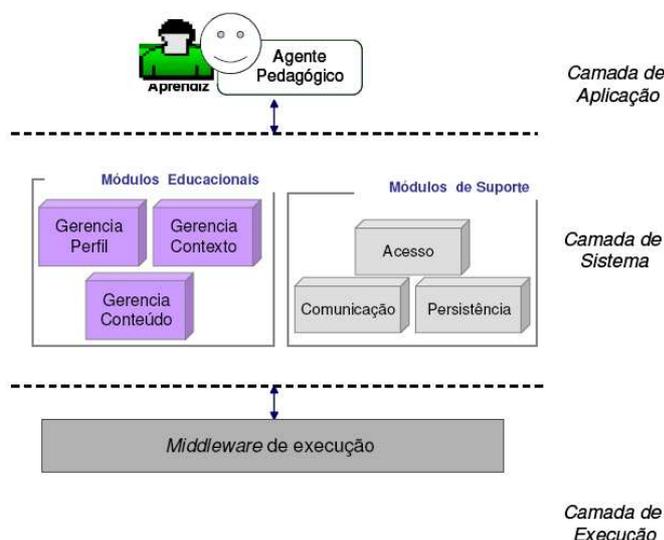


Figura 6 - Arquitetura GlobalEdu (Barbosa, 2007)

3.4.2 *Middleware* de suporte à execução

É desejável que a camada de execução suporte os seguintes requisitos:

- **Percepção do contexto físico:** é necessário que o ambiente de execução forneça os elementos de contexto físico relevantes para o sistema, como a localização do aprendiz no contexto e a situação da sua conectividade no momento;
- **Suporte à mobilidade do aprendiz:** o ambiente de execução deve, transparentemente, dar suporte a mobilidade do aprendiz portanto seu AP, através dos diferentes locais de uma região geográfica ou entre regiões geográficas;
- **Suporte à comunicação e migração entre servidores distribuídos:** o ambiente deve auxiliar a comunicação entre os servidores GlobalEdu, permitindo a busca de recursos, como objetos de aprendizagem e outros elementos para suporte à execução do AP;
- **Acesso ao sistema em escala global:** o AP acessa os recursos da rede independente de sua localização física ou forma de deslocamento. Considera-se uma conectividade intermitente, com da possibilidade de comunicação em escala global.

Das características destacadas, a percepção do contexto físico e suporte à mobilidade do aprendiz são fundamentais para o funcionamento dos elementos propostos pelo GlobalEdu. Caso o sistema não suporte os demais itens, considera-se que o GlobalEdu poderá ser aplicado

em ambientes considerados de acesso local, abrangendo, nesse caso, uma localização caracterizada como uma Região Geográfica.

O GlobalEdu adota um modelo cliente-servidor, mas prevê comunicação P2P entre os APs, mas essa capacidade em última instância depende do *Middleware* de execução.

3.5 Avaliação dos Ambientes Pesquisados

Nessa seção será feito um comparativo entre os trabalhos apresentados, como pode ser visualizado na Tabela 2.

Tabela 2 - Comparativo entre sistemas

	Japelas	Framework de interação social de Zhang	Ambiente de aprendizagem ubíqua para a aprendizagem colaborativa P2P	GlobalEdu
Objetivo	Apoiar a aprendizagem de expressões de tratamento na língua Japonesa.	Apoiar a aprendizagem dos alunos focando em suas habilidades sociais em um ambiente ubíquo.	Criação de comunidades virtuais, onde instrutores e alunos, descobrem, acessam e compartilham recursos.	Dar as condições necessárias para que o aprendiz construa seu conhecimento de qualquer lugar e em qualquer tempo, de forma adaptada ao seu perfil e contexto.
Sensibilidade de localização	Tag's RFID.	Não possui sistema de localização.	Combinação de tecnologias como triangulação de antenas.	Sim, a forma depende do <i>middleware</i> de execução.
Comunicação entre dispositivos	IR	P2P, não especificado.	P2P e cliente servidor.	Depende do <i>middleware</i> de execução.
Redes móveis	Parcial, não apresenta algoritmo de roteamento.	Sim	Sim	Depende do <i>middleware</i> de execução.
Segurança distribuída	Não especificado.	Não especificado.	Não especificado.	Depende do <i>middleware</i> de execução.
Sensibilidade ao contexto	Sim	Não	Sim	Sim
Escalabilidade localizada	Sim	Sim	Sim	Depende do <i>middleware</i> de execução.
Acesso móvel à informação	Sim	Sim	Parcial, pois parte da informação fica no servidor.	Não, as informações ficam armazenadas no servidor.
Extensível	Não, focado apenas no tratamento de expressões do idioma japonês.	Parcialmente, apesar de ser um <i>framework</i> só permite a personalização dos tutores.	Não	Parcialmente, também permite a customização do tutor.

O Japelas possui um objetivo bem específico, apoiar a aprendizagem de expressões de tratamento na língua Japonesa, não apresentando estruturas que facilitem a adição de novas funcionalidades. É um ambiente descentralizado, mas apresenta um sistema de comunicação

limitada entre os dispositivos, por IR (infravermelho), não apresentando estrutura de roteamento de mensagens, servindo basicamente para identificar qual formalismo que deve ser aplicado ao usuário.

O *framework* de interação social de Zhang permite a customização do tutor que pode ser adaptado para uma área específica, mas não apresenta estruturas que facilitem a adição de novas funcionalidades. É um ambiente descentralizado, mas não apresenta duas características essenciais a sistemas ubíquos, sensibilidade de localização e contexto.

O ambiente de aprendizagem ubíqua para a aprendizagem colaborativa P2P, proposto por Yang, não apresenta estruturas que facilitem a adição de novas funcionalidades. Não é um ambiente totalmente descentralizado, necessitando de intervenção de um servidor para determinar o contexto do aprendiz, o que afeta a escalabilidade da aplicação. Além disso, mantém parte das informações no servidor, limitando o acesso móvel à informação.

O GlobalEdu adota um modelo cliente-servidor no qual toda a comunicação entre os módulos do sistema é via servidores. Os APs se comunicam com os servidores (em primeira instância) e entre eles. Sendo que, esta última funcionalidade, depende do desenvolvimento desta capacidade no *Middleware*.

3.6 Considerações sobre o Capítulo

Esse capítulo apresentou quatro ambientes de educação ubíqua, que são descentralizados ou apresentam alguma característica descentralizada, traçando uma análise de suas características, limitações e contribuições.

O próximo capítulo apresenta o Global, uma infra-estrutura descentralizada de educação ubíqua, que busca contornar as limitações das propostas estudadas.

4 INFRA-ESTRUTURA PROPOSTA

Nesse capítulo é apresentada a arquitetura do Global. A Seção 4.1 apresenta uma visão geral da infra-estrutura descrevendo seus agentes e componentes auxiliares. A Seção 4.2 mostra a modelagem na metodologia Prometheus, descrevendo o diagrama geral do sistema e o detalhamento dos agentes.

4.1 Visão Geral

O Global é formado por dez componentes (Figura 7). Interface com Usuário, Persistência, Restritores e Proxies são componentes de suporte representados por API's. Contexto, Conectividade, Perfil do Usuário, Perfis, Objetos de Aprendizagem e Comunicação são agentes de *software*. Esses agentes são executados de forma isolada, sendo que cada um disponibiliza um grupo específico de funcionalidades para o apoio à aprendizagem. Cada instância do Global, isto é, cada cópia sendo executada em um dispositivo, também tem uma instância desses agentes em execução. O Global não exige a execução de todos os agentes, sendo que a ausência de um dos agentes compromete apenas as funcionalidades que dependem dele. Deve-se ressaltar que a execução parcial desses agentes, por exemplo, remoção do agente de Contexto, pode afetar as características de ubiquidade do ambiente.

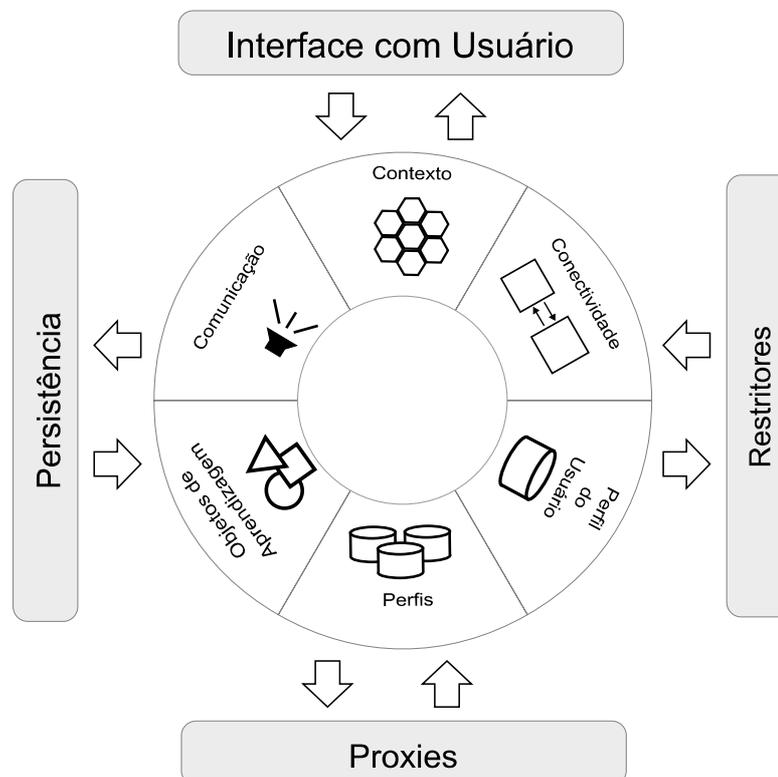


Figura 7 - Infra-estrutura do Global

O agente de Conectividade é responsável pela troca de mensagens entre agentes, sua função é encontrar a melhor forma para entregar a mensagem, sendo de forma direta entre agentes de uma mesma instância do Global, ou pela escolha de um Proxy que tenha acesso ao dispositivo do agente destinatário, caso sejam agentes de instâncias diferentes.

O agente de Contexto é responsável pela gerência dos contextos no Global, mapeando a movimentação do usuário pelos contextos e alertando os demais agentes sobre essas alterações. Os agentes de Perfil do Usuário e Perfis são responsáveis, respectivamente, pela gerência do perfil do usuário do dispositivo em execução e dos perfis de outros usuários com os quais tem ou teve contato. O agente de Objetos de Aprendizagem armazena, disponibiliza e adquire os objetos de aprendizagem utilizados pelos usuários. Por fim, agente de Comunicação é responsável pela troca de mensagens entre os usuários.

Os quatro componentes de suporte, disponibilizando funcionalidades básicas para os agentes. Interface com Usuário é responsável pelo acesso dos usuários às funcionalidades do Global e seus agentes, disponibilizando componentes necessários para entrada de dados do usuário e retornos oriundos do sistema. O componente de Persistência é uma API para gravação e recuperação das informações que devem ser conservadas mesmo quando a aplicação é finalizada. Restritores é uma estrutura genérica para gerência das características de ambiente baseada em restrições, que tem a finalidade de mapear as características lógicas e físicas para definição de contextos. Os Proxies são uma camada de abstração para troca de informações em redes *Ad Hoc*. A seguir são descritos os componentes do Global de forma mais detalhada.

4.1.1 Conectividade

Toda a comunicação entre os agentes do Global é feita pelo agente de Conectividade, através de troca de mensagens, sendo baseadas no padrão FIPA-ACL (*Foundation of Intelligent Physical Agents - Agent Communication Language*) (FIPA-ACL, 2010). A Tabela 3 mostra a estrutura das mensagens.

Tabela 3 - Parâmetros FIPA-ACL

Parâmetro	Tipo de comunicado	Dado comunicado
Performative	tipo de comunicado	Denota o tipo de ação comunicativa
Sender	participante na comunicação	Identidade do agente que enviou a mensagem
Receiver	participante na comunicação	Identidade dos destinatários da mensagem
reply-to	participante na comunicação	Identidade dos destinatários do retorno da mensagem
Content	conteúdo da mensagem	Denota o conteúdo da mensagem
Language	descrição do conteúdo	Denota a linguagem em que o conteúdo é expresso
Encoding	descrição do conteúdo	Denota a codificação do conteúdo
Ontology	descrição do conteúdo	Denota a ontologia do conteúdo
Protocol	controle da comunicação	Denota o protocolo de interação que o agente está usando na mensagem ACL
conversation-id	controle da comunicação	Identifica a sequência de mensagens que juntas formam uma conversação
reply-with	controle da comunicação	Identifica a resposta esperada
in-reply-to	controle da comunicação	Identifica qual mensagem está sendo respondida
reply-by	controle da comunicação	Hora/data limite até a qual o agente emissor esperará por uma resposta

Na troca de mensagens entre agentes de uma mesma instância do Global, ou seja, quando os agentes operam em um mesmo dispositivo, o agente de Conectividade executa a entrega de forma direta ao destinatário. Nas trocas de mensagens entre agentes de instâncias diferentes cabe ao agente de Conectividade encontrar a melhor forma para entregar a mensagem. Para tanto é necessário que o agente escolha um entre os Proxies disponíveis que tem acesso ao dispositivo do agente destinatário e envie essa informação através desse Proxy

para o agente de Conectividade da instância do destinatário da mensagem. Esse agente, por sua vez, encaminha ao agente destinatário. O modelo de entrega de mensagens é o de melhor esforço (*best-effort*) (Comer & Yavatkar, 1989), no qual não é dada a garantia de entrega da mensagem, visto que a presença do usuário na rede pode ser intermitente.

Cada usuário no modelo é representado por um identificador único, formado por um identificador do usuário e um domínio, similar ao *e-mail*. Por exemplo, “carlos@mobilab.unisinos.br”, significa que o usuário se chama “carlos” e o seu domínio é “mobilab.unisinos.br”. A identificação dos agentes dentro do dispositivo funciona de forma similar, onde o nome do agente é concatenado à identificação, por exemplo, caso uma mensagem seja destinada a “contexto:carlos@mobilab.unisinos.br”, o destinatário é o agente de Contexto do usuário carlos do domínio “mobilab.unisinos.br”.

Cada domínio deve disponibilizar um *webservice* com um serviço de autenticação com acesso na *web*, caso um dispositivo queira comprovar a autenticidade de outro, isto é, confirmar a identificação e domínio do usuário. Essa autenticação é baseada em chave pública e privada (Ferguson & Schneier, 2003), na qual a chave privada fica registrada no dispositivo e o serviço presente no domínio disponibiliza a chave pública. Essa autenticação é um processo opcional, pois o dispositivo não tem sempre disponibilidade de acesso à *web*, contudo por estar presente no mesmo ambiente do outro dispositivo pode garantir sua autenticidade.

4.1.2 Contexto

Contextos são fragmentos de informação que podem ser usadas para caracterizar a situação de um participante em uma interação (Dey, 2001). O Global define dois grupos distintos de características: de ambiente e relação.

As características de ambiente são percepções que a aplicação pode coletar do ambiente, tais como posição no espaço e no tempo. Um contexto pode ser definido como uma área no espaço, uma data, um intervalo de tempo ou data, um dia na semana. Já as características de relação definem característica no modelo que podem ser utilizadas como agrupadores de usuários, tais como status (por exemplo: professor, aluno) ou áreas de conhecimento e interesse.

Devido às inúmeras tecnologias de percepção de ambientes (por exemplo, GPS, câmeras digitais, RFID e sensores de calor) e a gama de características que podem ser consideradas como relação, o Global utiliza uma estrutura genérica para gerência das características de contexto baseada em restrições e Restritores. Cada contexto possui uma lista

de restrições. Por sua vez, cada restrição é associada a um Restritor que através da descrição da restrição determina se ela é satisfeita ou não, pelo ambiente ou relação em um dado instante. Cabe ao agente de Contexto associar as restrições de um contexto aos seus respectivos Restritores e monitorar suas mudanças de estado. Um contexto satisfaz sua característica, somente se todas as suas restrições são satisfeitas, isto é, um usuário se encontra em um contexto somente quando todas as restrições do contexto são avaliadas como verdadeiras.

Além disso, contextos podem ser derivados de outros, uma vez que dentro de um contexto podem ser formados sub-contextos apenas pela adição de novas características (restrições) a ele. Também podem ser centralizados em um usuário, no qual só esse usuário controla a gerência do contexto ou descentralizado quando qualquer usuário do contexto pode gerenciá-lo.

Os contextos no Global podem usar uma ou mais dessas características para definir os mais variados tipos de situação. Por exemplo, uma turma representa um contexto centralizado no professor, que possui duração de seis meses. O professor pode criar um contexto derivado da turma, dependente de posição física e tempo, que representa uma aula, definindo a presença na aula através do tempo que o aluno esteve no contexto. Durante a aula o professor passa um trabalho em grupo, os alunos criam contextos descentralizados, que tem como tempo de duração o tempo restante da aula, para representar cada grupo desse trabalho, para trocar informações e materiais.

A Figura 8 mostra os metadados do contexto (*Context*):

- **Id:** é um identificador único que representa o contexto;
- **Description:** descrição textual do contexto;
- **Ontology:** descreve qual estrutura que representa a informações no contexto;
- **Owner:** id do criador do contexto;
- **Centered:** se contexto é centralizado no criador;
- **Restrictions:** lista de restrições do contexto;
- **Information:** árvore de informações do contexto.

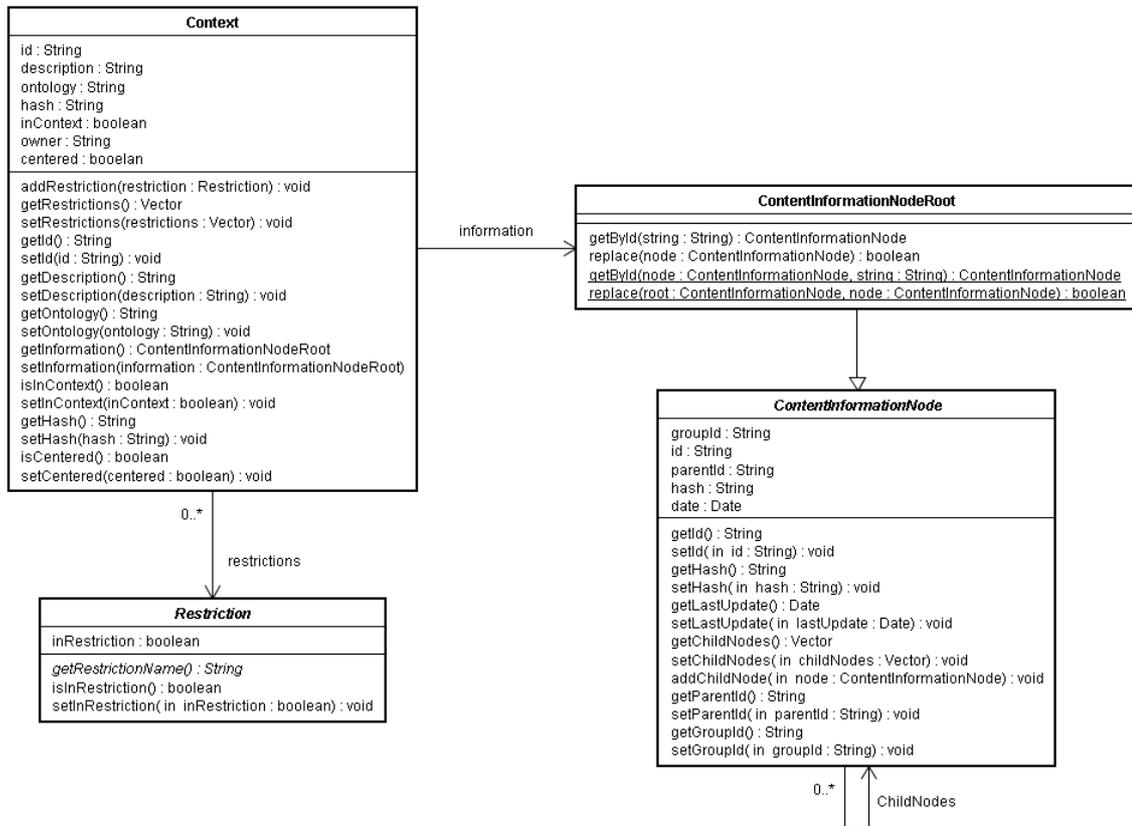


Figura 8 - Estrutura do Contexto

Cada contexto possui uma área destinada ao armazenamento de informações, que são compartilhadas com todos os usuários do contexto, caso o contexto não seja centralizado no criador, qualquer de seus participantes pode adicionar ou remover informações dele. Levando em conta a natureza descentralizada do Global e que usuários podem sair e entrar nos contexto a qualquer momento, a sincronização dessas informações se torna complexa.

Para sincronização dessas informações é utilizada uma estrutura de árvore, a *Global Information Tree*, similar ao DOM (*Document Object Model*) (W3C-DOM, 2010), onde cada nodo (*ContentInformationNode*) da árvore contém parte das informações do contexto, além disso cada nodo mantém a data da sua última atualização e um *hash* que é formado pela suas informações somadas as informações dos nodos filhos. Quando um usuário entra em um contexto ele compara as *hash* dos seus nodos com os nodos dos outros usuários, atualizando somente os nodos mais antigos.

Além da criação, atualização e manutenção dos contextos, o agente de Contexto também serve como uma ponte para os demais agentes obterem informação sobre os contextos, como: lista de contextos disponíveis, usuários ativos no contexto e nodos de informação do contexto. Através da estrutura de subscrição (*subscription*), do modelo FIPA-ACL, outros agentes também podem pedir notificação de mudança de estado dos contextos.

O agente de Contexto também registra todas as mudanças de contexto do usuário, inclusive com o horário de entrada e saída. Tendo como base essas informações, é criado um *tracking* (histórico do deslocamento do usuário pelos contextos), que pode ser consultado por outros agentes.

4.1.3 Perfil do Usuário

O agente de Perfil do Usuário é responsável por gerenciar e disponibilizar informações do usuário aos demais agentes do Global.

O Global não exige a utilização de um padrão específico de perfil, ao invés disso ele utiliza a estrutura do *Global Information Tree* para armazenar as informações do perfil deixando a possibilidade de que cada especialização da infra-estrutura utilize o modelo de perfil que considerar mais adequado. Dois tipos de nodos são utilizados para construção da árvore de informação do perfil, *PublicProfileNode* (nodo público) e *PrivateProfileNode* (nodo privado). Nodos privados podem ser acessados somente por agentes que estão na mesma instância do agente de Perfil do Usuário e nodos públicos podem ser acessados por qualquer agente.

Apesar do Global não exigir um padrão específico de perfil ele implementa por padrão o modelo PAPI (PAPI, 2010). A escolha foi baseada em duas características do padrão:

- Flexibilidade: o PAPI pode ser estendido e todos os seus componentes são opcionais;
- Modularidade: os campos do perfil podem ser tratados de forma separada.

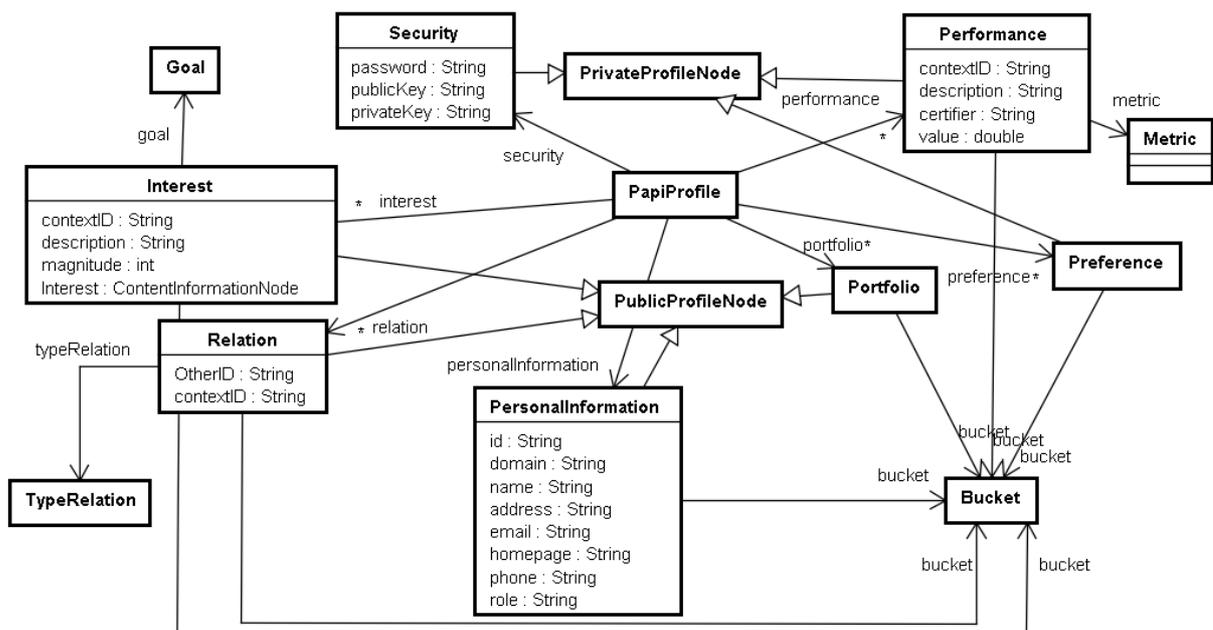


Figura 9 - Estrutura PAPI

A Figura 9 mostra a organização da estrutura do perfil. Bucket são ternas formadas pelo nome do agente, chave e informação (*ContentInformationNode*), servindo para extensão de elementos do perfil. Outros agentes podem utilizar o Bucket para guardar informações de seu interesse no perfil do usuário.

O perfil do usuário é dividido em sete grupos:

- Informações Pessoais (*Personal Information*): armazena informações básicas do usuário;
- Informações de Segurança (*Security*): armazena as credenciais do aluno, como senha chave pública e privada;
- Informações sobre as Relações (*Relation*): define uma lista com relações entre os usuários do Global. É composto pelo ID do usuário da relação e tipo da relação, podendo ser associada a um contexto ou ser uma relação global;
- Informações de Preferências (*Preference*): descreve as preferências do aluno, com o objetivo de facilitar a sua interação com o sistema;
- Informações do Desempenho (*Performance*): relaciona a história do aluno, listando, seu desempenho e avaliações, indicando as áreas de interesse gerais e específicas, métrica de avaliação, valor recebido pelo usuário e quem certificou a avaliação. Também pode ser associada a um contexto ou ser um desempenho global;
- Informações sobre o Portfólio (*Portfolio*): armazena um conjunto de referências para os trabalhos feitos pelo aluno;
- Informações de Interesses (*Interest*): o grupo de interesses, não é parte do padrão PAPI e foi integrando do padrão LIP (IMS, 2010). A seção interesses armazena informações sobre tópicos relacionados a áreas de interesse do usuário.

4.1.4 Perfis

Como o Global é um modelo descentralizado, não existe um local que centralize todas as informações dos usuários. Surge então a necessidade de gerenciar perfis de outros usuários no próprio dispositivo do usuário. Essa é a função do agente de Perfis, que gerencia o acesso ao repositório de perfis de outros usuários.

O agente de Perfis possui quatro configurações básicas para atualização dos perfis, que podem ser usadas dependendo da necessidade da aplicação ou capacidade do dispositivo:

- Requisição por demanda: quando um agente requisita uma informação de perfil, ele contata o agente Perfil do Usuário do usuário que requisitou para pedir a informação;

- Requisição por demanda com *cache*: similar ao *cache* de dados dos navegadores *web*. Quando uma informação sobre um usuário é requisitada e essa informação está disponível ele contata o agente Perfil do Usuário desse usuário perguntado se existe atualização na informação, caso não exista ou ele não consiga contatar o agente, ele utiliza a informação existente;
- Armazenamento de informação: o agente de contexto mantém as informações dos outros usuários armazenados no dispositivo, toda a vez que um usuário fica acessível ele atualiza as informações defasadas;
- Armazenamento de informação com atualização em tempo real: além de manter as informações no dispositivo e atualizar as informações quando o usuário fica acessível, o agente de Perfis, através da estrutura de subscrição (*subscription*), requisita ao agente Perfil do Usuário desse usuário que informe as alterações em seu perfil, desta forma mantendo os dados o mais atualizados o possível.

Como os perfis utilizam o *Global Information Tree*, a atualização também é feita somente nos nodos da árvore que sofreram alteração. O agente de perfis só tem acesso a informação dos nodos derivados do *PublicProfileNode* (nodo público).

4.1.5 Objetos de Aprendizagem

O agente de Objetos de Aprendizagem tem a finalidade de armazenar, adquirir e distribuir objetos de aprendizagem. A especificação dos metadados desses objetos no Global segue a norma IEEE/LTSC/LOM (LOM, 2010). A ampla aceitação desse padrão pode ser conferida em (Rigaux & Spyratos, 2007).

A Figura 10 mostra a estrutura dos metadados, sendo que suas categorias são descritas a seguir:

- Geral (*General*): agrupa informações gerais que descrevem o objeto;
- Ciclo de vida (*lifeCycle*): agrupa informações que descrevem as características relacionadas ao histórico e estado atual dos objetos;
- Técnica (*Technical*): agrupa os requisitos e características técnicas do objeto;
- Educacional (*Educational*): agrupa as características educacionais e pedagógicas;
- Direitos (*Rights*): agrupa os direitos de propriedade intelectual e as condições de uso do objeto;
- Meta-Metadados (*MetaMetadata*): agrupa informações sobre a instância de metadados em si (em vez do objeto de aprendizagem, a instância de metadados que o descreve);

Considerando a descentralização do Global, foi adotado um modelo distribuído para obtenção do objeto de aprendizagem de outros usuários, baseado em redes P2P, onde uma rede *bitTorrent* (Rajagopalan & Shen, 2006) é criada entre os usuários acessíveis pelos Proxies de comunicação, que possuem ou querem um determinado objeto de aprendizagem.

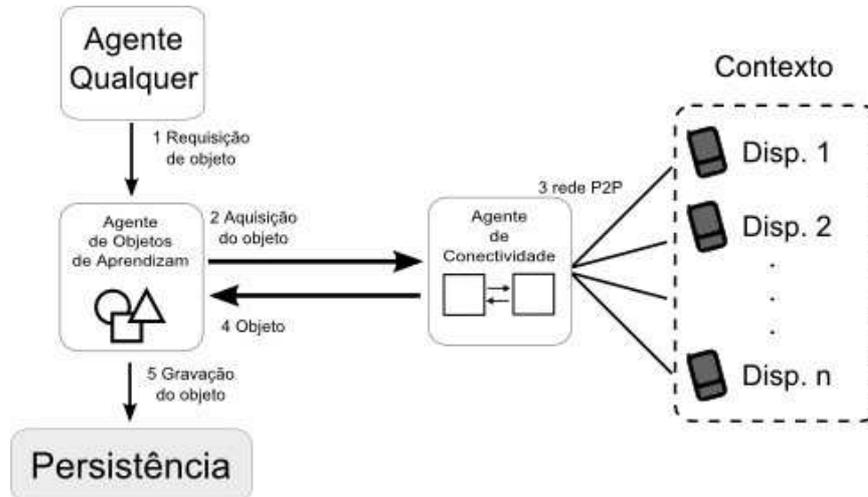


Figura 11 - Aquisição de Objeto de Aprendizagem

A Figura 11 mostra como o processo de aquisição de objetos funciona. Um agente faz a requisição de aquisição de objeto (passo 1). O agente inicia a aquisição do objeto através do agente de Conectividade (passo 2). O agente de Conectividade cria uma rede *bitTorrent* P2P descentralizada para fazer o *download* do objeto (passo 3). A rede *bitTorrent* retorna as informações para o agente de Objetos de aprendizagem (passo 4). Por fim a informação é persistida no dispositivo através do componente de Persistência (passo 5).

4.1.6 Comunicação

O agente de Comunicação gerencia o recebimento e o envio de mensagens. O agente pode ser utilizado tanto pelo usuário, através da interface, quanto por outros agentes do Global. As mensagens são notificações textuais que ficam armazenadas em uma “caixa de entrada” para consulta posterior. O agente de Comunicação suporta tanto o envio de mensagens para um usuário específico, quanto envio de mensagens para um contexto específico onde todos aqueles que estiverem presentes no contexto recebem a notificação. Caso o destinatário da mensagem não esteja disponível o agente armazena a mensagem até que ele esteja. A Figura 12 mostra os metadados da comunicação.

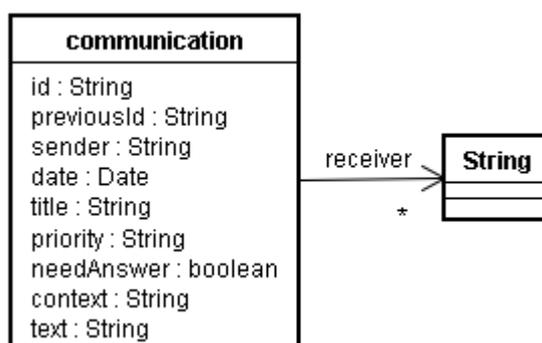


Figura 12 - Estrutura da comunicação

4.1.7 Persistência

O componente de Persistência é responsável por manter a persistência das informações quando a aplicação é finalizada. Esse componente disponibiliza uma API para gravação e recuperação de informação, abstraindo as operações de I/O presentes no resto da aplicação, que passam a utilizar somente essa API. A API é similar a um banco de dados orientado a objetos (OODBMS), onde a informação é armazenada na forma de objetos. O componente ainda disponibiliza uma estrutura de exportação e importação de dados, com a finalidade de realizar *backup* ou migrações, caso o usuário queira trocar de dispositivo, mantendo as mesmas configurações e informações já existentes no dispositivo anterior.

4.1.8 Interface com Usuário

O componente Interface com Usuário é o responsável por disponibilizar os meios para acesso do usuário às funcionalidades do sistema. Esse componente tem como características ser modular e configurável o que facilita a customização do sistema aos mais variados tipos de dispositivos, tanto pela variação de tamanho de *display* como pelas disponibilidades de interação: botões, teclado, *touchscreen* e *multi-touch*.

4.1.9 Proxies

Um Proxy representa um meio de comunicação possível entre dispositivos, por exemplo, um Proxy pode ser implementado para realizar a troca de mensagens utilizando conexão *bluetooth* entre dispositivos e outro Proxy pode ser implementado para realizar troca de mensagens através de um servidor *web*, acessado pelo dispositivo por uma rede Wi-Fi ou 3G. Cada Proxy é responsável por identificar novas conexões, perda de conexões, manter a lista de usuários disponível e gerenciar tanto o envio de mensagens como o seu recebimento.

4.1.10 Restritores

Como dito anteriormente, o Global utiliza uma estrutura genérica para gerência das características de ambiente baseada em restrições e Restritores, sendo que um contexto possui uma lista de restrições. Os Restritores tem a função de monitorar o ambiente e identificar mudanças de estado nas restrições (a restrição é satisfeita ou não) e notificar seus ouvintes (*listeners*).

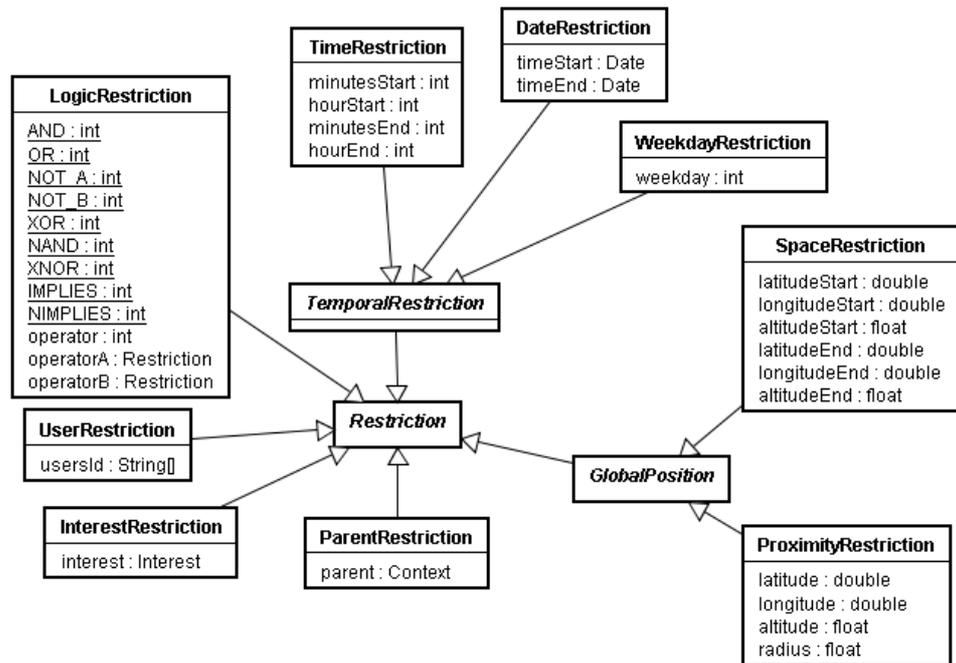


Figura 13 - Restrições

A Figura 13 mostra as restrições definidas pelo Global. Os Restritores dessas restrições são associados na inicialização do agente de Contexto. Desta forma os Restritores podem ser adaptados de forma dinâmica dependendo dos recursos do dispositivo, por exemplo, as restrições do tipo **GlobalPositionRestriction** podem ser associadas a um Restritor de localização por GPS caso o dispositivo tenha esse recurso ou por triangulação de antena caso não tenha.

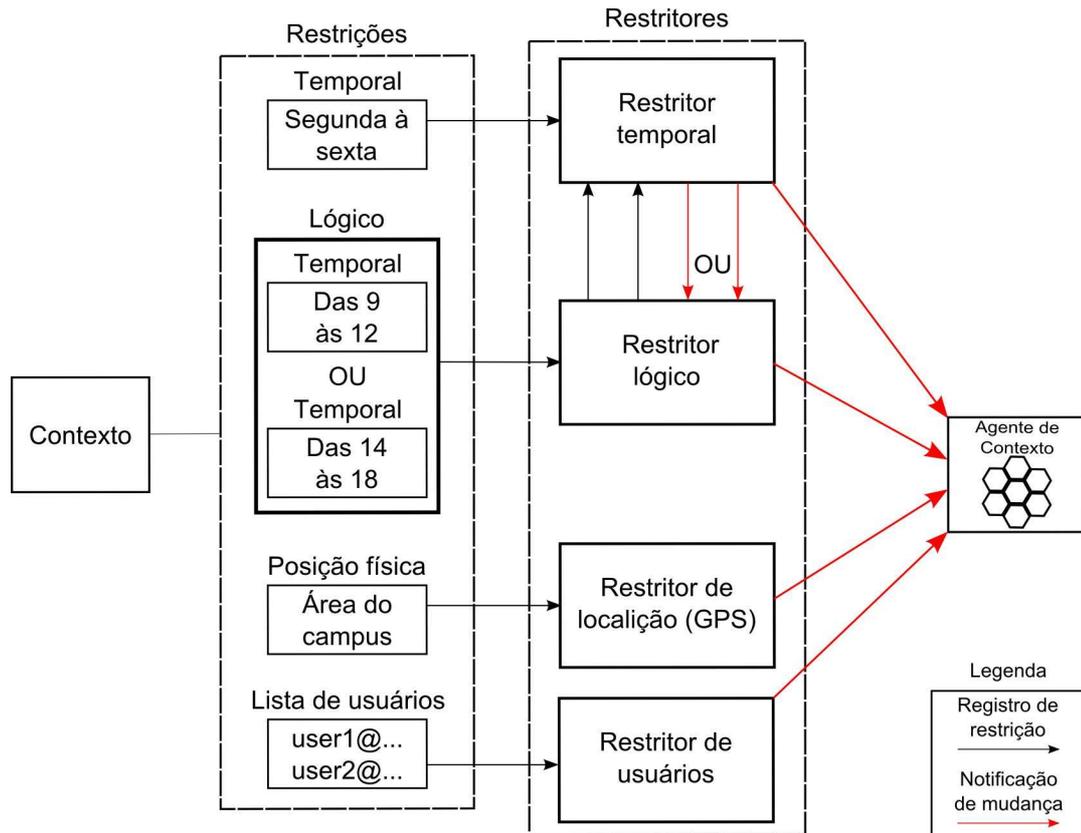


Figura 14 - Exemplo de restrições

A Figura 14 mostra um exemplo das restrições, onde um contexto possui uma restrição temporal para representar os dias da semana, uma restrição lógica de operação “ou” formada por duas restrições temporais para representar os horários de atuação, uma restrição física que representa a área da atuação do contexto e uma restrição que representa os usuários que podem participar do contexto. A vantagem da gerência de contextos por Restritores é que a adição de novos Restritores é simples, bastando criar a restrição e o Restritor, fazendo seu registro no agente de Contexto, não sendo necessária nenhuma outra alteração no Global.

4.2 Modelagem em Prometheus

Essa seção apresenta a especificação do Global, utilizando a metodologia Prometheus, focando na arquitetura (Subseção 4.2.1) e detalhamento do projeto (Subseção 4.2.2).

4.2.1 Diagrama geral do sistema

A Figura 15 mostra o diagrama geral do Global, que apresenta a comunicação entre os agentes, que é realizada através de troca de mensagens, tanto por mensagens únicas, como por protocolos de comunicação. Os protocolos descrevem um fluxo da troca de mensagens entre

os agentes. A Figura 16 mostra um diagrama de troca de mensagens, neste caso o protocolo **handShake**. O diagrama geral mostra uma instância do Global, sendo que os agentes (exceto o agente Conectividade_1) estão sendo executados em um único dispositivo. O agente Conectividade_1 representa um agente de Conectividade genérico executando em outro dispositivo, isto é, representa uma saída para envio de mensagens para outras instâncias do Global. Como dito anteriormente todas as comunicações são feitas através do agente de Conectividade. Para facilitar a compreensão dos diagramas e fluxogramas, somente as comunicações para agentes de outras instâncias (enviadas para o agente Conectividade_1) passam pelo agente de Conectividade. Protocolos com final “Ret Protocol”, por exemplo **HandShake Ret Protocol**, representam protocolos de retorno, isto é, o Global deve responder quando ele é outra instância na execução.

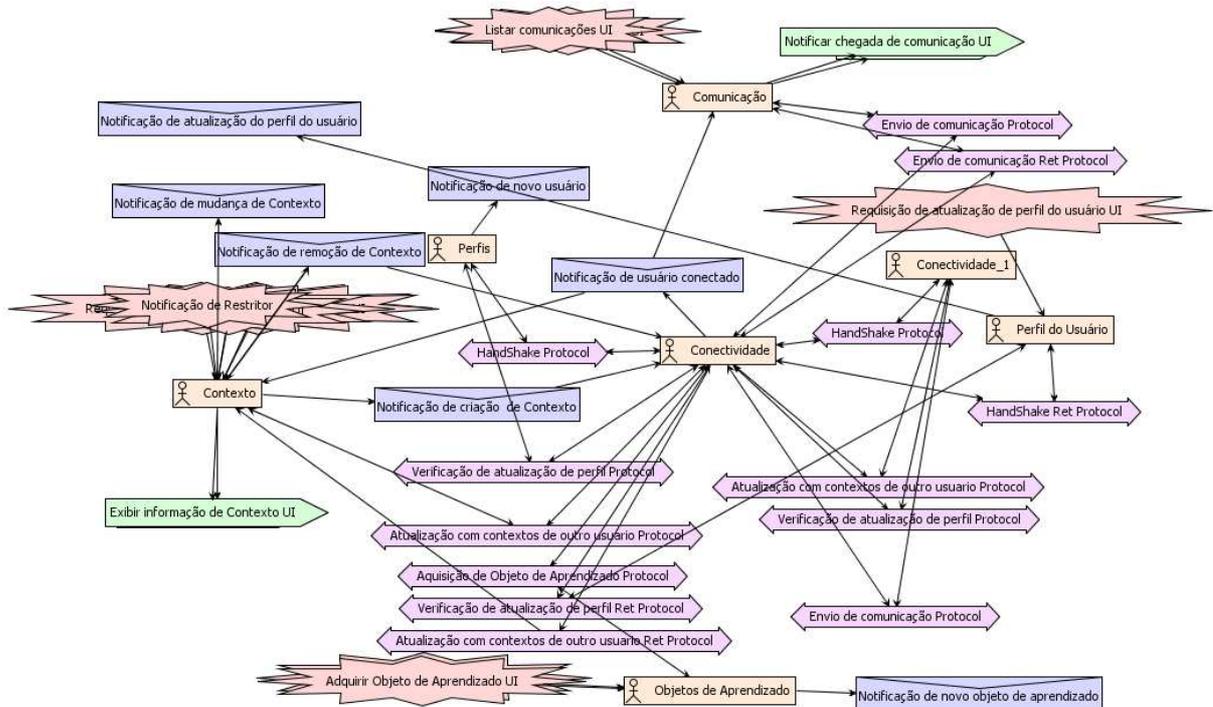


Figura 15 - Diagrama geral do Global

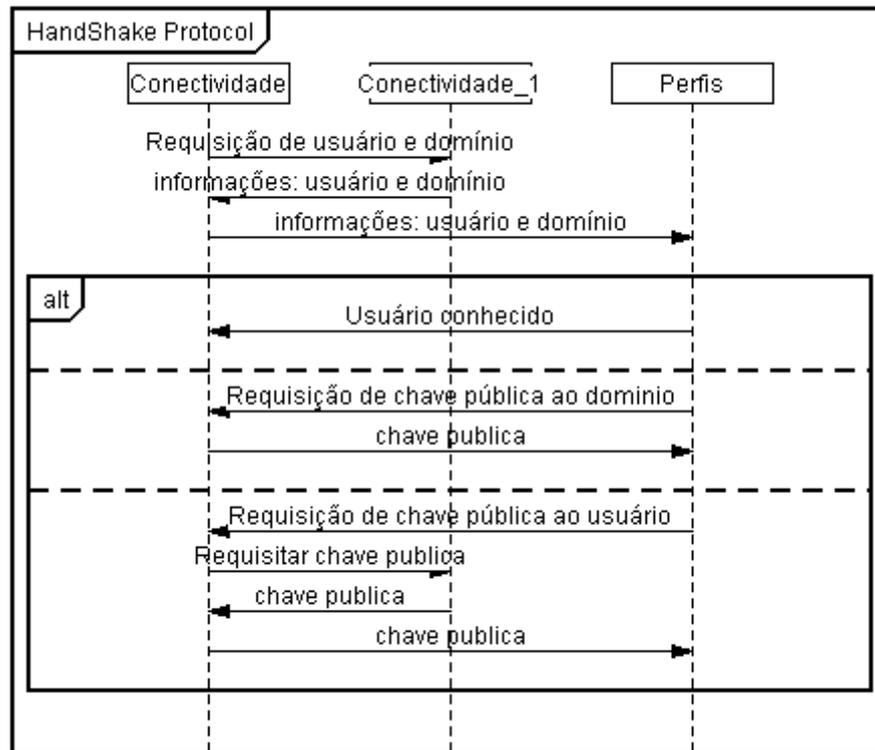


Figura 16 - Protocolo de HandShake

Os agentes do Global disponibilizam uma estrutura de subscrição, na qual qualquer agente pode fazer uma requisição de notificação (*listener*) de um determinado evento para outro agente, por exemplo, a criação de um contexto ou adição de um novo objeto de aprendizagem. A partir dessa requisição toda vez que o agente identificar o evento requisitado ele envia uma notificação, através de uma mensagem, aos agentes requisitantes. Essa estrutura é bastante útil a agentes que desempenham funcionalidades passivas, que não necessitam ficar se comunicando o tempo todo com os demais agentes para saber se tiveram alguma alteração, bastando aguardar o retorno por parte do agente que gerencia a informação.

4.2.2 Detalhamento dos agentes

Nessa subseção é apresentado o detalhamento de cada agente, definindo suas capacidades e seus protocolos de comunicação.

4.2.2.1 Conectividade

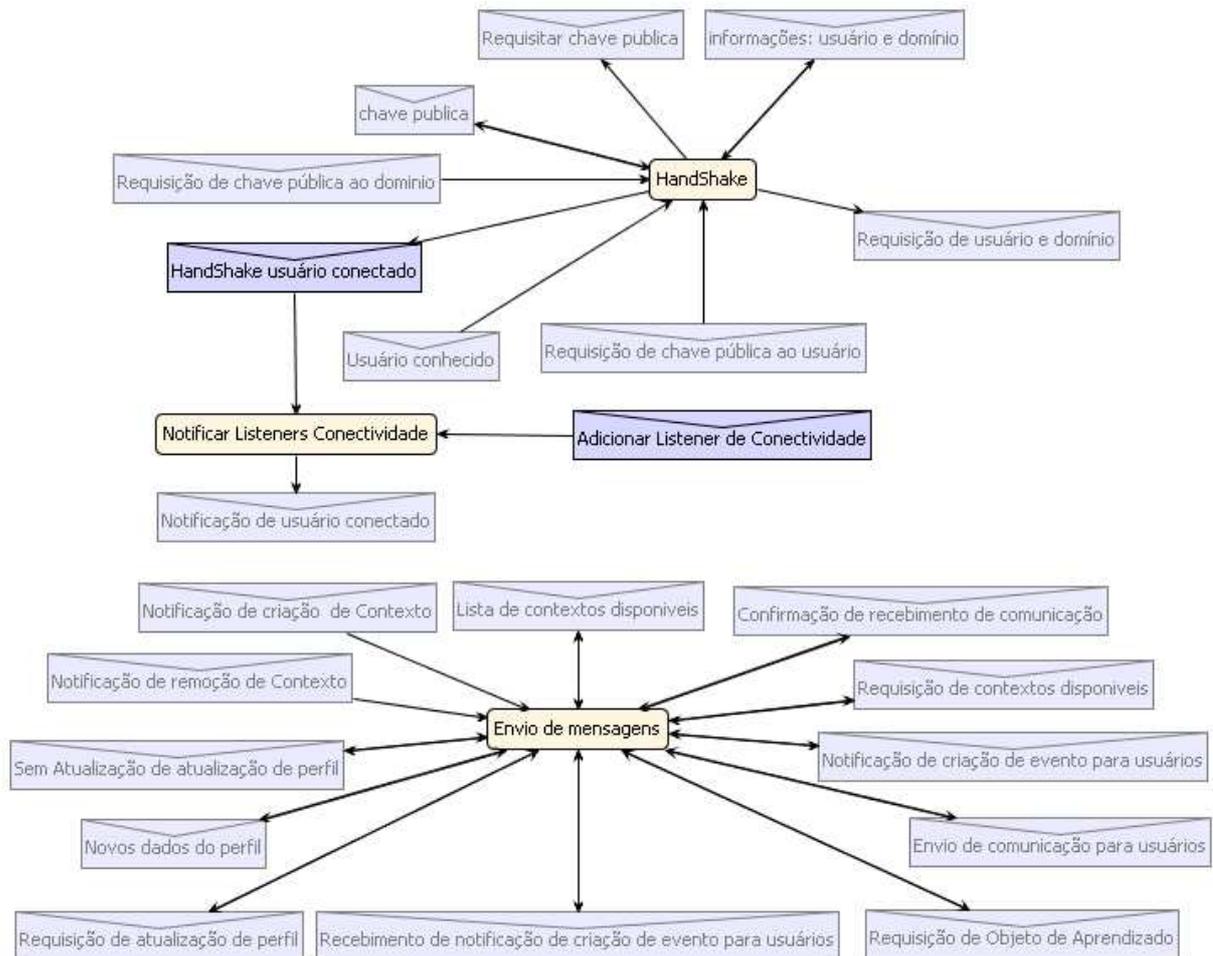


Figura 17 - Detalhamento do agente de Conectividade

A Figura 17 mostra o detalhamento do agente de **Conectividade**, onde são definidas três capacidades para o agente:

- **HandShake:** o **HandShake** é o processo de primeiro contato com outra instância do Global. **HandShake Protocol** (Figura 16) mostra como é feito esse primeiro contato. Após a obtenção da identificação de usuário e domínio, o agente de Perfis é contatado, caso o usuário não seja conhecido, o agente de Perfis retorna como deve ser feita a requisição de chave pública, se de forma direta com o usuário ou por requisição ao domínio. Após o processo de **handShake**, o agente notifica a capacidade de **Notificar Listeners Conectividade**;
- **Notificar Listeners Conectividade:** capacidade de notificar usuários conectados (usuários que podem receber mensagens) e registrar agentes para receber essa notificação;

- **Envio de mensagens:** capacidade de enviar mensagens aos agentes fora da instância em execução, escolhendo um entre os Proxies disponíveis que tem acesso ao agente destinatário.

4.2.2.2 Contexto

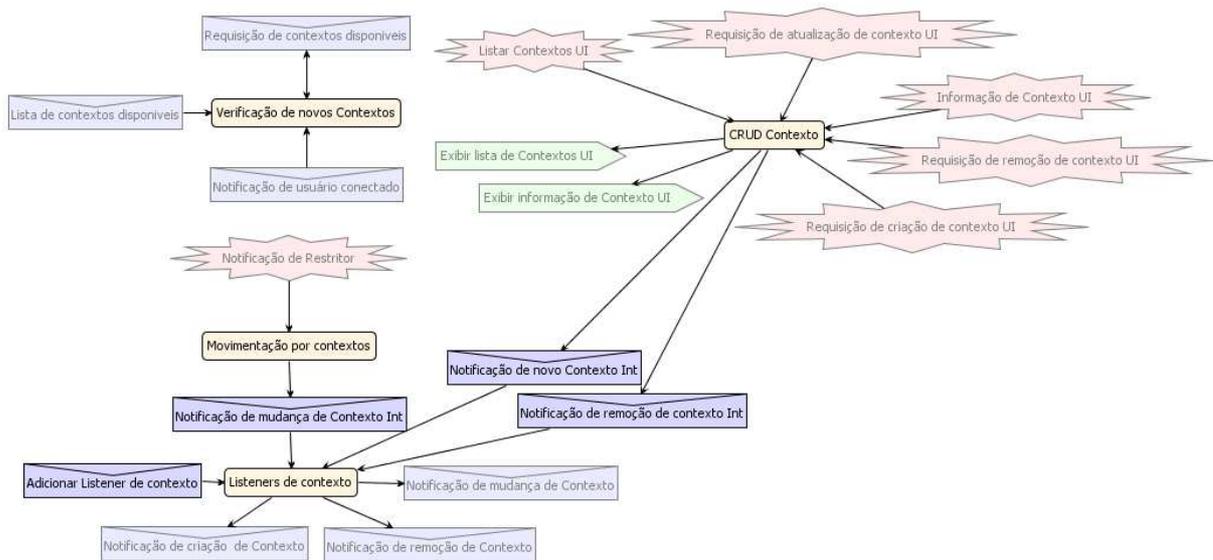


Figura 18 - Detalhamento do agente de Contexto

A Figura 18 mostra o detalhamento do agente de **Contexto**, onde suas capacidades são:

- **CRUD Contexto:** CRUD (*Create, Retrieve, Update e Delete*) são as operações de criação, recuperação, atualização e remoção de contextos. Cada vez que um contexto é alterado o **Listeners de contexto** é notificado, e por sua vez notifica os demais agentes registrados;
- **Movimentação por contextos:** através dos Restritores essa capacidade identifica mudanças nos contextos existentes e notifica o **Listeners de contexto**;
- **Listeners de contexto:** registra pedidos de notificação e gera notificações de criação, mudança e remoção de contextos. Também notifica os agentes de Contexto dos usuários participantes do contexto que sofreu alteração;
- **Verificação de novos Contextos:** quando um usuário se torna acessível, essa capacidade contata esse usuário para buscar novos contextos disponíveis ou atualizar contextos já existentes;
- **Histórico de movimentação:** capacidade de recuperar o histórico do deslocamento do usuário pelos contextos (*tracking*).

4.2.2.3 Perfil do Usuário

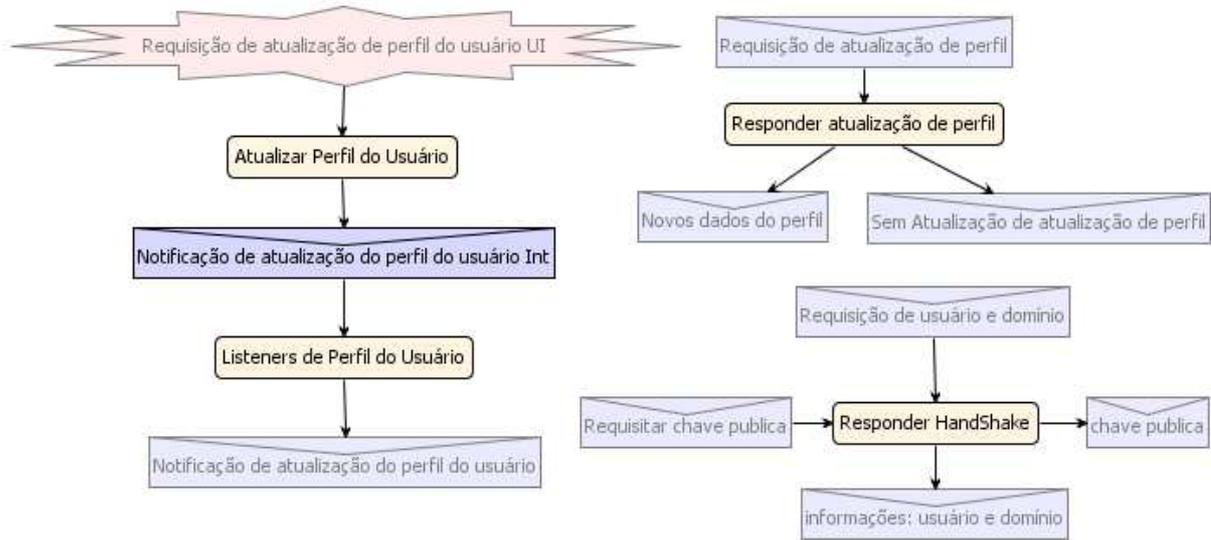


Figura 19 - Detalhamento do agente de Perfil do Usuário

A Figura 19 mostra o detalhamento do agente de Perfil do Usuário. Suas capacidades são:

- **Atualizar Perfil do Usuário:** capacidade de alterar informações do perfil do usuário, notificando o **Listeners de Perfil do Usuário**;
- **Listeners de Perfil do Usuário:** registra pedidos de notificação e gera notificações de mudança de dados do perfil do usuário;
- **Responder atualização de perfil:** capacidade que representa o retorno da requisição de atualização de perfil feito por outra instância do Global;
- **Responder HandShake:** representa a parte de responsabilidade do agente Perfil do Usuário no processo de **HandShake** descrito pelo protocolo **HandShake Protocol**.

4.2.2.4 Perfis

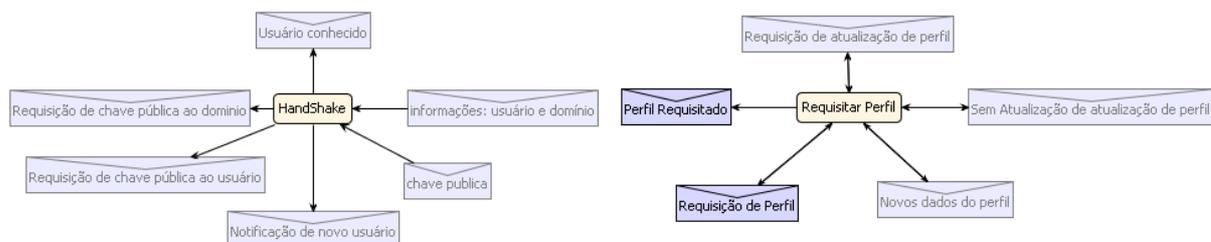


Figura 20 - Detalhamento do agente de Perfis

A Figura 20 mostra o detalhamento do agente de **Perfis**. A seguir são descritas suas capacidades:

- **HandShake**: representa a parte de responsabilidade do agente de Perfis no processo de **HandShake** descrito pelo protocolo **HandShake Protocol**;
- **Requisitar Perfil**: capacidade de requisitar informações de perfis de outros usuários, caso tal informação esteja disponível ele contata o agente Perfil do Usuário desse usuário perguntado se existe atualização na informação. O protocolo de **Verificação de atualização de perfil** (Figura 21) descreve esse processo.

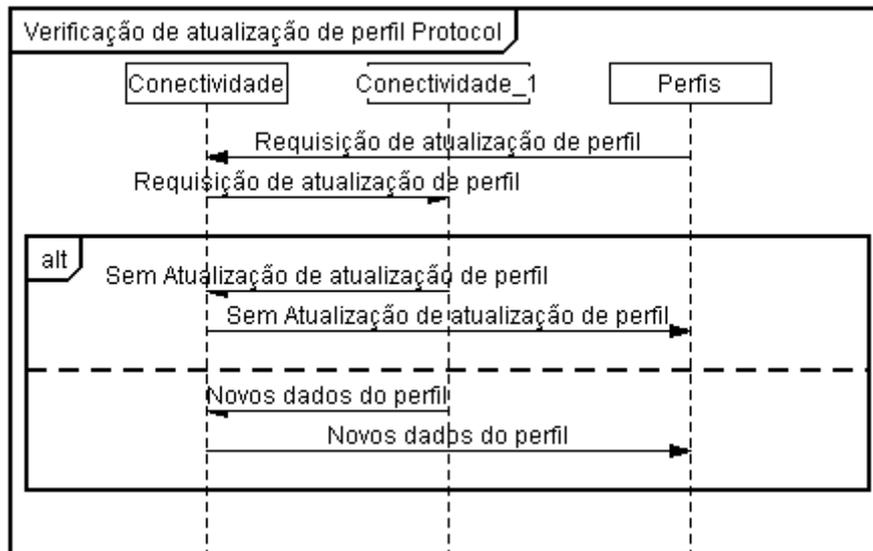


Figura 21 - Protocolo de Verificação de atualização de perfil

4.2.2.5 Objetos de Aprendizagem

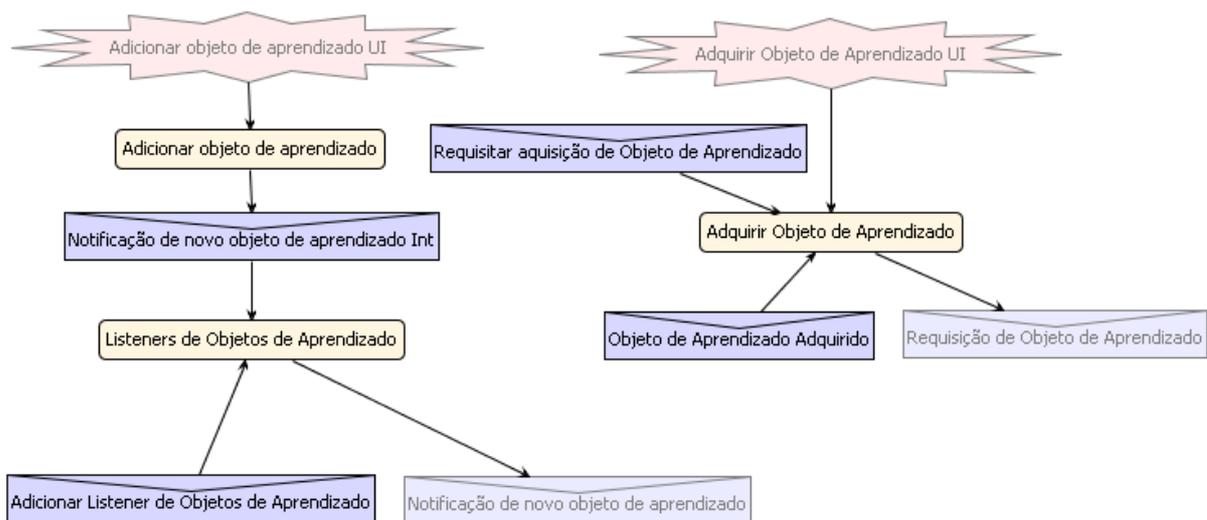


Figura 22 - Detalhamento do agente de Objetos de Aprendizagem

A Figura 22 mostra o detalhamento do agente de Objetos de Aprendizagem. Suas capacidades são:

- **Receber comunicação:** capacidade de receber comunicação, notificando o usuário através da interface com o usuário e confirmando o recebimento para o remetente;
- **Listar comunicações:** capacidade de listar, através da interface com o usuário, as comunicações recebidas.

4.3 Considerações sobre o Capítulo

Esse capítulo apresentou o Global, mostrando uma visão geral da infra-estrutura descrevendo seus agentes e componentes auxiliares. Também foi mostrada a modelagem na metodologia Prometheus, descrevendo o diagrama geral do sistema e o detalhamento dos agentes.

O próximo capítulo apresenta o protótipo do Global, descrevendo alguns de seus aspectos de implementação.

5 ASPECTOS DE IMPLEMENTAÇÃO

O protótipo do Global foi desenvolvido em J2ME (Java 2 - *Micro Edition*), MIDP 2.0 (*The Mobile Information Device Profile*) (JSR-118, 2010) e CLDC 1.1 (*Connected Limited Device Configuration*) (JSR-139, 2010), podendo ser usado em qualquer *smartphone* que suporte essa configuração. O protótipo foi testado nos seguintes equipamentos: Nokia N95, HTC Touch HD e HTC TyTN II.

As seções seguintes descrevem alguns aspectos de implementação do protótipo do Global, não tendo como objetivos mostrar todos os detalhes de implementação e todas as funcionalidades dos agentes e APIs, mas sim, mostrar aspectos e decisões de projeto relevantes ou que trazem ganhos para o desenvolvimento das especializações da infraestrutura.

5.1 Estrutura geral do Global

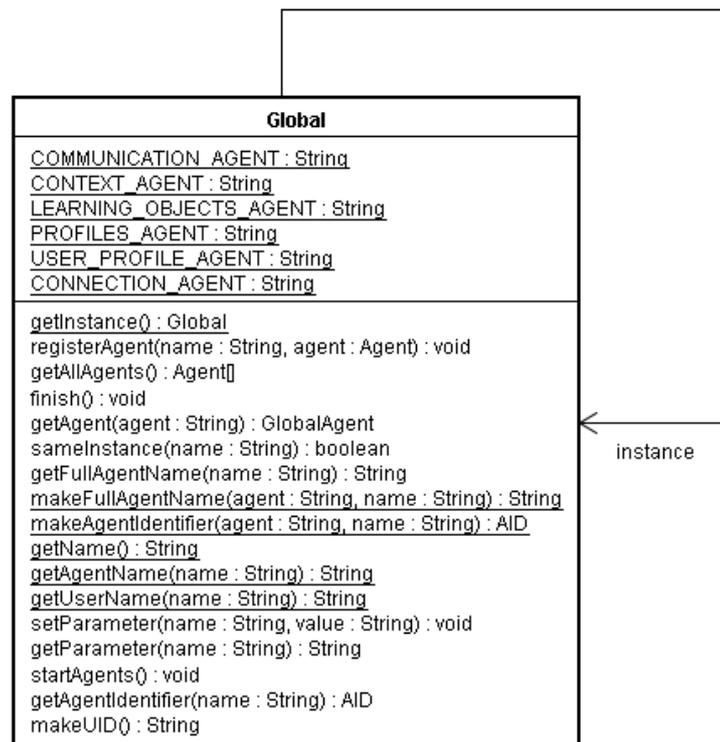


Figura 24 – Classe Global

A Figura 24 mostra o diagrama UML¹ da classe principal do Global, que é responsável pelo registro dos agentes no software. Essa classe é implementada utilizando o padrão de projeto *Singleton* (Gamma et al, 2005) e representa uma instância de execução do Global.

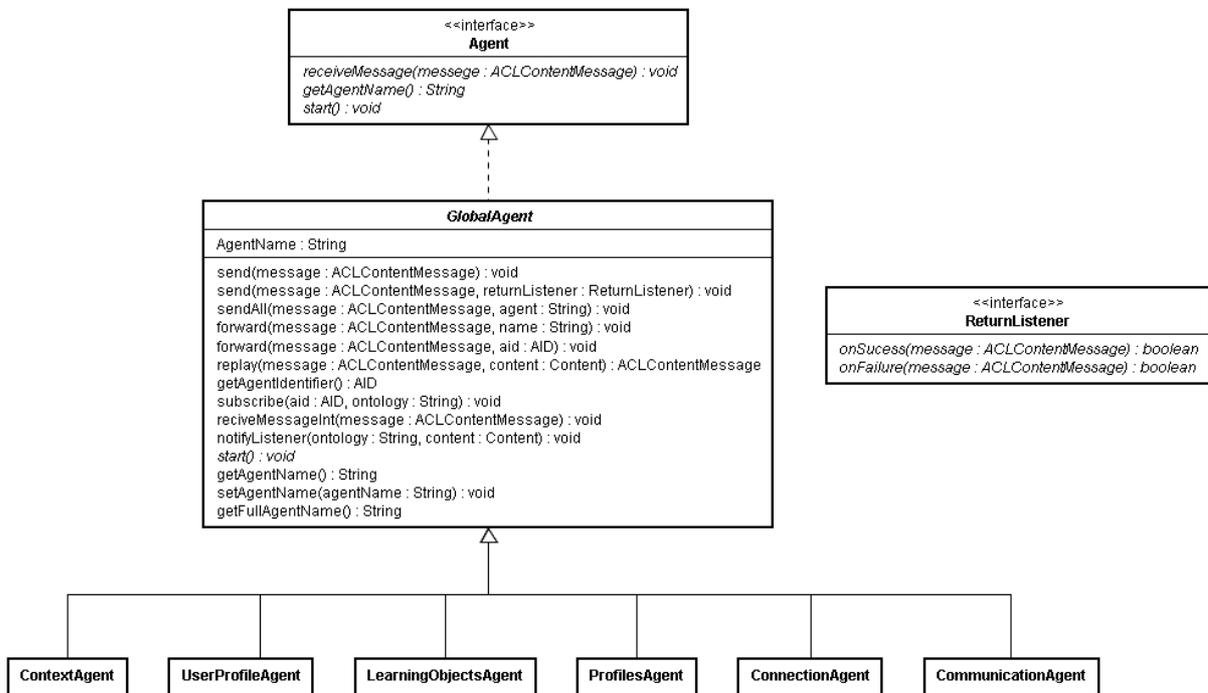


Figura 25 - Estrutura dos Agentes

A Figura 25 mostra a estrutura dos agentes no protótipo. Cada agente é representado por uma classe, que obrigatoriamente deve implementar a interface **Agent**. A classe abstrata **GlobalAgent** disponibiliza uma série de funcionalidades que facilitam a implementação dos agentes. Novos agentes no Global podem ser criados tanto pela extensão da interface **Agent**, utilizada para integração de implementações de agentes já existentes, como pela extensão da classe abstrata, caso se queira aproveitar as facilidades fornecidas pelo Global. A seguir são descritos os principais métodos da interface **Agent** e da classe **GlobalAgent**.

- **send**: envia uma mensagem ACL. Caso um ouvinte **ReturnListener** seja mandado por parâmetro, se houver sucesso na obtenção da resposta, o método **onSucess** é executado de forma assíncrona com a mensagem de retorno, se o tempo de resposta expirar ou não houver sucesso na obtenção da resposta, o método **onFailure** é executado com o erro;

¹ Os diagramas UML dessa seção representam somente as classes, atributos e métodos considerados relevantes e necessários para entendimento da implementação do protótipo.

- **sendAll**: envia uma mensagem para todas os agentes do tipo especificado que estão acessíveis;
- **forward**: encaminha a mensagem ao agente identificado;
- **replay**: responde a uma mensagem com conteúdo enviado por parâmetro;
- **getAgentIdentifier**: retorna a identificação única do agente;
- **subscribe**: envia uma mensagem do tipo **SUBSCRIBE** para um agente;
- **notifyListener**: método para notificação dos ouvintes;
- **start**: método abstrato que representa a inicialização do agente. Esse método deve ser implementado pelas especializações da classe;
- **receiveMessage**: método abstrato utilizado para o recebimento. Esse método deve ser implementado pelas especializações da classe, para executar o tratamento das mensagens.

O **GlobalAgent** já implementa a estrutura de subscrição (*subscription*) definida na FIPA-ACL. Quando um agente invocar o método **subscribe** requisitando notificações de determinado tipo, o **GlobalAgent** envia uma mensagem FIPA-ACL do tipo **SUBSCRIBE** para o agente indicado. Caso esse agente também seja uma extensão do **GlobalAgent**, ele automaticamente armazena essa requisição em uma lista de ouvintes. Toda as vezes que o agente invocar o método **notifyListener**, o **GlobalAgent** retorna uma mensagem para essa lista de ouvintes com o conteúdo indicado no parâmetro do **notifyListener**.

5.2 Comunicação entre Agentes

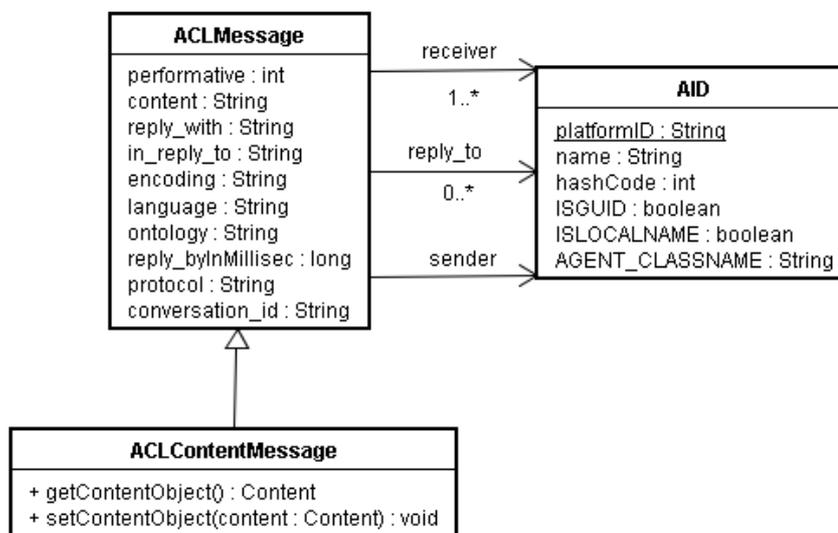


Figura 26 - Diagrama da classe de mensagens ACL

A Figura 26 mostra o diagrama da classe de mensagens. Os campos são descritos na Tabela 3. Os tipos de comunicação (**performative**) são: ACCEPT_PROPOSAL, AGREE, CANCEL, CFP, CONFIRM, DISCONFIRM, FAILURE, INFORM, INFORM_IF, INFORM_REF, NOT_UNDERSTOOD, PROPOSE, QUERY_IF, QUERY_REF, REFUSE, REJECT_PROPOSAL, REQUEST, REQUEST_WHEN, REQUEST_WHENEVER, SUBSCRIBE, PROXY, PROPAGATE e UNKNOWN.

ACLContentMessage é uma classe que representa uma mensagem que transporta conteúdos do tipo **Content**, que é uma interface que permite transportar objetos através de mensagens. As mensagens são enviadas utilizando a especificação por string do FIPA-ACL (FIPA-Representation, 2010), sendo compatível com qualquer outro software que utilize essa especificação de comunicação, por exemplo, JADE (JADE, 2010). A Figura 27 mostra uma mensagem codificada dessa forma.

```
(REQUEST
:sender ( agent-identifier :name context:user2@mobilab.unisinos.br )
:receiver ( set ( agent-identifier :name context:user1@mobilab.unisinos.br ) )
:reply-to ( set ( agent-identifier :name context:user2@mobilab.unisinos.br ) )
:content "/*objeto serializado*/"
:reply-with context:user1@mobilab.unisinos.br1265670264755
:in-reply-to context:user2@mobilab.unisinos.br1265670262987
:language serializedObject
:ontology br.unisinos.mobilab.global.ontology.context.ContextDescriptionList)
```

Figura 27 - FIPA-ACL: Especificação por String

Quando o agente de Conectividade recebe uma mensagem para envio para outra instância do Global, ele percorre a lista de destinatários da mensagem. Para cada destinatário ele procura um Proxy que tem acesso a ele, converte a mensagem para representação por string e envia pelo Proxy. Quando um Proxy recebe uma mensagem ele notifica o agente de Conectividade, enviado a string da mensagem, que converte novamente essa string em um objeto da classe **ACLMessage** e envia ao agente de destino. Desta forma os demais agentes não precisam lidar com conversão de mensagem, podendo tratar a troca de mensagem com um nível de abstração maior.

Dois Proxies foram implementados, um utilizando tecnologia *bluetooth* utilizando a API J2ME (JSR-82, 2010) e ou outro baseado em *WebServices* para Wi-Fi/3G. O Proxy baseado em *WebServices* necessita de um servidor para fazer o roteamento das mensagens, foi

criado com finalidade de avaliar a possibilidade de utilização de múltiplos Proxies. Sendo utilizado como único Proxy em uma execução da infra-estrutura, ela perde parte de suas características de descentralização.

5.3 Restrições

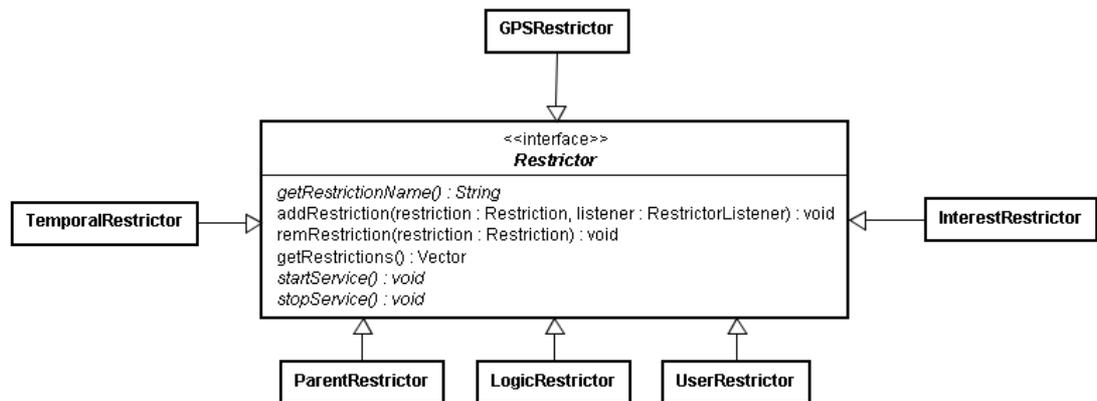


Figura 28 - Restritores

Os Restritores são registrados no agente de Contexto, seguindo o padrão de projeto *Strategy* (Gamma et al, 2005). A associação entre as restrições e o Restritor é feita pelo nome do Restritor. A Figura 28 mostra a interface dos Restritores e os Restritores implementados no protótipo, seguindo o padrão de projeto *observer* (Gamma et al, 2005). Quando um contexto é carregado pelo agente de Contextos, ele adiciona todas as restrições do Contexto nos seus respectivos Restritores. Toda a vez que uma notificação de mudança (através do **RestrictorListener**) é feita pelo Restritor o agente de Contexto verifica se houve mudança no Contexto, notificando os demais agentes interessados.

A seguir são descritos os Restritores implementados no protótipo:

- **GPSRestrictor:** Restritor associado as restrições do tipo **GlobalPositionRestriction** e seus derivados. Utiliza a API padrão para GPS do J2ME (JSR-179, 2010) para monitorar as mudanças de posição física do usuário;
- **TemporalRestrictor:** Restritor associado as restrições do tipo **TemporalRestriction** e seus derivados. Esse Restritor executando uma *thread* que monitora o relógio do sistema, notificando seus ouvintes quando há alguma mudança nos estados das restrições;
- **ParentRestrictor:** Restritor associado as restrições do tipo **ParentRestriction**. Quando uma **ParentRestriction** é adicionada ao Restritor ele busca todas as restrições

do contexto pai definido na restrição, adicionando aos seus respectivos Restritores e monitorando com um ouvinte próprio essas mudanças;

- **LogicRestrictor:** Restritor associado as restrições do tipo **LogicRestriction**. Aplica operações lógicas em outras restrições, como 'ou', 'e' e implicação. Quando um **LogicRestriction** é adicionado o Restritor busca as restrições das operações, adicionando aos seus respectivos Restritores. A cada mudança de estado na restrição, avalia se houve mudança no valor da operação lógica;
- **UserRestrictor:** Restritor associado as restrições do tipo **UserRestriction**. Avalia se o usuário encontra-se na lista de usuários que tem acesso ao contexto;
- **InterestRestrictor:** Restritor associado as restrições do tipo **InterestRestriction**. Compara o interessa da restrição com os interesses do perfil do usuário.

5.4 Persistência

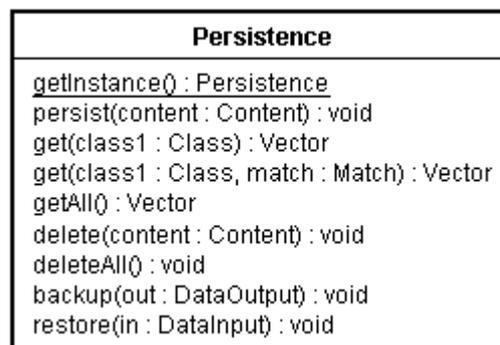


Figura 29 - Persistência

A Figura 29 mostra a API de persistência. A persistência é feita sobre objetos que implementam a interface **Content**. Os objetos são serializados e armazenados utilizando a API RMS (JSR-118, 2010). A recuperação dos objetos persistidos pode ser feita tanto pela recuperação de lista (*Vector*) com todos os objetos de uma classe ou utilizando uma função de *callback* (**Match**) que retorna somente os objetos que são aceitos pela função de combinação. Além disso a API oferece uma função de *backup* através de uma saída de dados genérica (*DataOutput*) e uma função de restauração com uma entrada de dados genérica (*DataInput*).

5.5 Interface com o usuário

Para facilitar o desenvolvimento da interface com usuário e a utilização por suas especializações, foi integrado ao Global uma interface com usuário. Foi utilizado o *framework* LWUIT (*Lightweight UI Toolkit*) (LWUIT, 2010) que é um *framework* de código aberto que

facilita a criação de aplicativos com *designs* mais sofisticados, oferecendo componentes visuais, animações para aplicativos J2ME, sendo sua API baseada na API Swing do Java SE, o que facilita ainda mais a utilização por parte de desenvolvedores que já utilizam essa API.

A API oferece melhorias a componentes já existentes no Java ME, como List, Form, Alert, entre outros e oferece ainda: suporte a touch Screen, diversas fontes, animações, botões, transições de telas animadas, temas (que podem ser incluídos pelos próprios usuários), layouts, utilização de abas (como no Java SE), integração 3D, caixas de diálogo, entre outros.

Além disso, o *framework* disponibiliza uma API de Localização padrão L10n, para facilitar o suporte a múltiplas linguagens.

5.6 Global Agent Message Viewer

Além da implementação do protótipo do Global, também foi desenvolvido o Global Agent Message Viewer, uma ferramenta que é executada em desktop (Java J2SE) que monitora, em tempo real, as mensagens trocadas entre as instâncias em execução do Global. As mensagens são exibidas em forma de árvore, listando também os objetos de seus conteúdos. Além disso, a ferramenta permite exibir todos os objetos armazenados na API de persistência de um determinado agente.

O Global Agent Message Viewer tem a finalidade de facilitar o desenvolvimento e a depuração das especializações do Global. Por meio dele é possível ter uma visão clara do fluxo de mensagens entre as instâncias. A vantagem dele em relação às outras ferramentas de depuração, é que mantém o histórico das trocas de mensagem de todas as instâncias em execução em um só lugar. A Figura 30 mostra uma tela do Agent Message Viewer em execução.

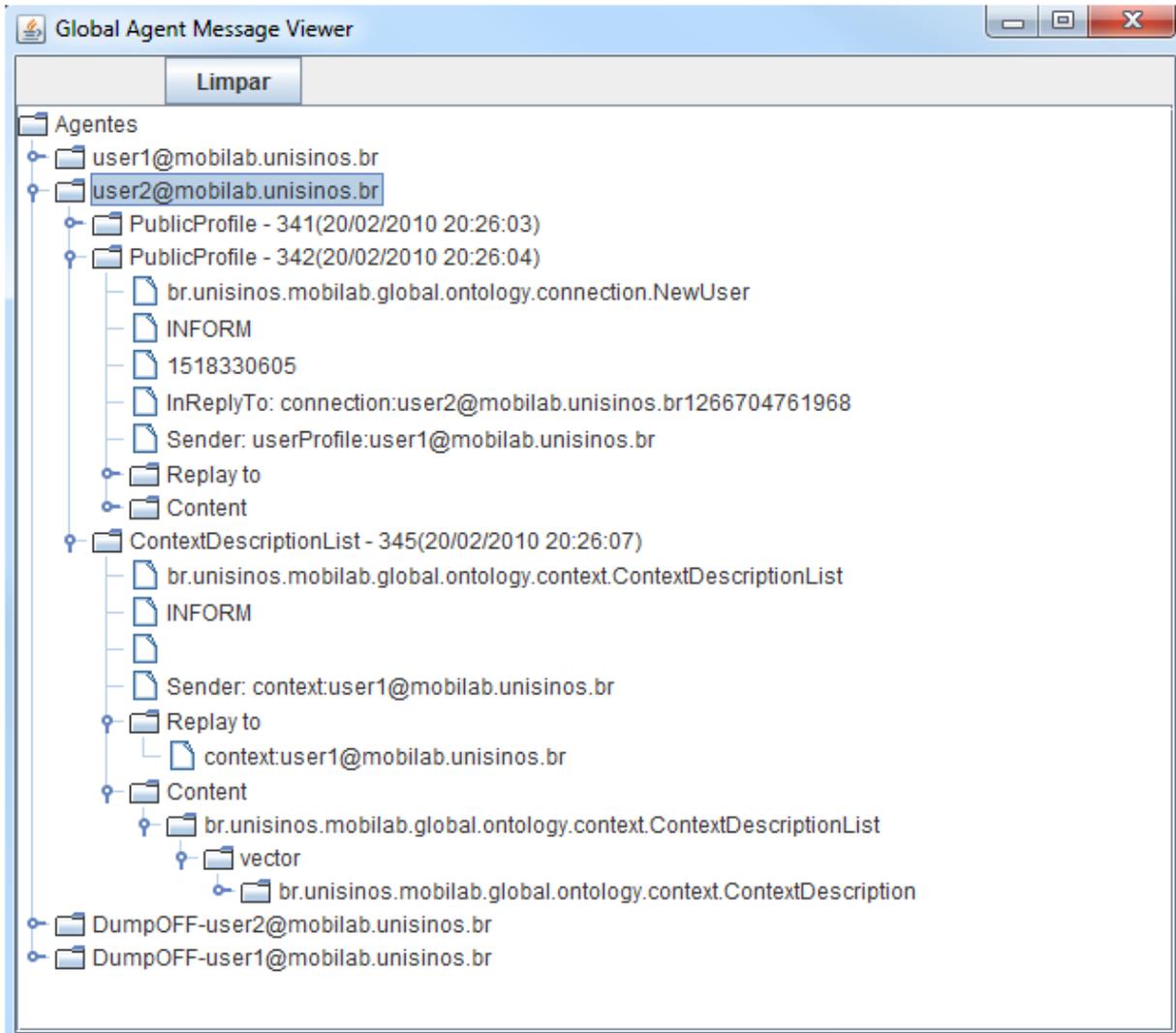


Figura 30 - Global Agent Message Viewer

5.7 Considerações sobre o Capítulo

Esse capítulo apresentou aspectos de implementação do protótipo do Global, foram descritos a organização dos agentes, a comunicação entre os agentes, funcionalidades das APIs do Global e apresentado o Global Agent Message Viewer.

O próximo capítulo apresenta três aplicações desenvolvidas a partir do protótipo do Global para avaliação da infra-estrutura.

6 AVALIAÇÃO

Segundo Edwards et al. (2003) a avaliação de uma infra-estrutura é considerada problemática, uma vez que a mesma não é visível para o usuário final. Sendo que somente é possível avaliar as funcionalidades de uma infra-estrutura construindo aplicações que as utilizem e então avaliá-las, obtendo-se assim, uma avaliação indireta da infra-estrutura.

Considerando esse fato foi proposta a criação de aplicações a partir da infra-estrutura do Global e a avaliação delas tanto no aspecto de desenvolvimento de software, como no de funcionalidade.

Para isso foram criadas três aplicações a partir do protótipo, cada uma com propósito de avaliar diferentes características da infra-estrutura:

- Uma aplicação que implementa um ambiente de aprendizagem ubíqua existente: essa aplicação teve a finalidade de comparar os ganhos trazidos pela utilização da infra-estrutura do Global, em comparação ao desenvolvimento sem a utilização dela. Também sendo possível fazer um comparativo direto entre as funcionalidades das duas aplicações;
- Um ambiente colaborativo de educação: essa aplicação integra um modelo multi-agente de colaboração ao Global e visou avaliar a capacidade da infra-estrutura de integrar novas funcionalidades com a integração de novos agentes;
- Aplicação de rede social ubíqua: essa aplicação teve por objetivo testar a flexibilidade da infra-estrutura para utilização em outras áreas que não a educação.

Nas seções a seguir são descritas as três implementações.

6.1 Aplicação que implementa um ambiente de aprendizagem ubíquo existente

Para realização de avaliação foi escolhida a proposta de arquitetura de assistente pessoal feita por Hahn (2007) (Hahn & Barbosa, 2008).

6.1.1 Arquitetura Original

A arquitetura propõe a organização de espaços de aprendizagem ubíqua na forma de Contextos Interativos, que são representações abstratas de contextos físicos, definidos pela presença de três componentes básicos:

1. Agentes Assistentes, que acompanham e representam os aprendizes no sistema;
2. Gerenciador de Contexto, que delimita e gerencia o Contexto Interativo;

3. Proxy de Comunicação, que oferece a possibilidade de comunicação entre os Agentes e entre estes e o Gerenciador de Contexto.

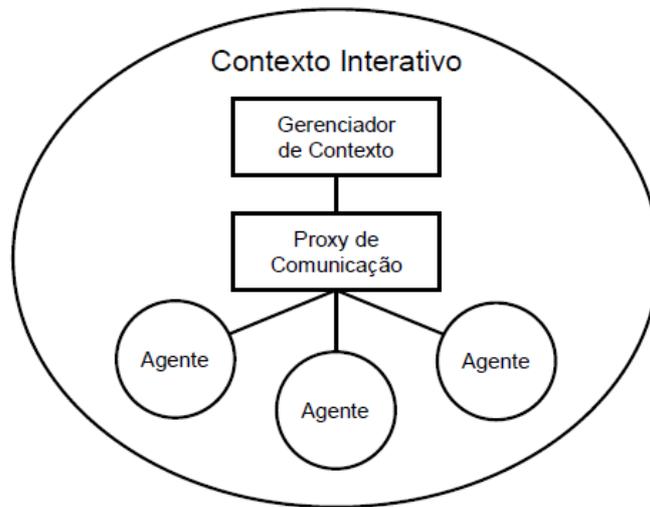


Figura 31 - Contexto interativo

A Figura 31 mostra uma representação do contexto interativo. Os agentes assistentes se deslocam livremente através dos contextos interativos, auxiliados por um sistema de localização. Durante esse deslocamento, os aprendizes interagem entre si e recebem recomendações de objetos de aprendizagem personalizados.

O agente assistente representa o aprendiz na arquitetura. Ele é responsável por acompanhar a mobilidade do aprendiz, buscando informações que descrevam o contexto físico no qual está inserido. O objetivo do agente é mapeado para a busca contínua de oportunidades pedagógicas, de modo a enriquecer a experiência dos aprendizes. Os dados dos perfis são armazenados nesse agente e seguem o padrão PAPI (PAPI, 2010).

Cada contexto interativo é gerenciado por um módulo chamado gerenciador de contexto, que delimita o contexto interativo e representa o ambiente e é responsável por coordenar o ingresso dos agentes no contexto. Quando o agente ingressa em um contexto ele recebe informações contextuais que o descrevem.

O Proxy de Comunicação é responsável por disponibilizar um canal de comunicação para os agentes. Quaisquer agentes presentes em um determinado contexto utilizam o *proxy* de comunicação daquele contexto específico, seja para se comunicar com outros agentes ou com o contexto em si. Conforme se desloca no ambiente, o agente comunica-se com o contexto interativo, buscando informações que reportem seu estado atual. Para comunicar-se,

o remetente (agente ou contexto) contata o *proxy* de comunicação através de mensagens no padrão FIPA-ACL. A interação entre agente e contexto é feita de duas formas:

- Indicação de semelhança entre perfil de usuários;
- Indicação de objeto de aprendizado relevante ao aprendiz.

O índice de semelhança entre perfis é dado pela correspondência de termos entre os assuntos de interesse de dois aprendizes quaisquer. Para cada correspondência verificada, $100/X\%$ (onde X é o número de assuntos de interesse do aprendiz) são acrescidos ao índice de semelhança. O índice de relevância de objetos de aprendizagem é calculado com base na equivalência entre as categorias de interesse de um aprendiz e as palavras-chave associadas a um objeto de aprendizagem. Novamente, para cada correspondência encontrada, $100/X\%$ são acrescidos a este índice (onde X é o número de termos associados ao objeto). Em ambos os casos, o valor máximo é de 100% (o que significa uma correspondência direta de todos os itens comparados).

6.1.2 Especialização do Global

A arquitetura original apesar de ser distribuída é centralizada, pois necessita do Proxy de Comunicação para troca de informação entre os dispositivos dos alunos. As gerências de contexto e material pedagógico também são feitas de forma centralizada, onde cada gerenciador de contexto tem um repositório de objetos associado.

O Global não tem um repositório de objetos de aprendizagem. A gerência dos objetos nos contextos é feita de forma distribuída entre os elementos participantes do contexto. Desta forma é necessário especializar o Global para que possa de alguma forma trabalhar com contextos que centralizem repositórios de objetos de aprendizagem. Para tanto é proposta a criação de uma instância especial do Global formada por especializações dos Agentes de Contexto, Objetos de Aprendizagem e Conectividade. Essa nova instância se comportará como um repositório de objetos.

Além disso, é necessário criar um agente para realizar o processo de indicação de semelhança entre perfil de usuários e indicação de objetos de aprendizagem relevantes ao aprendiz.

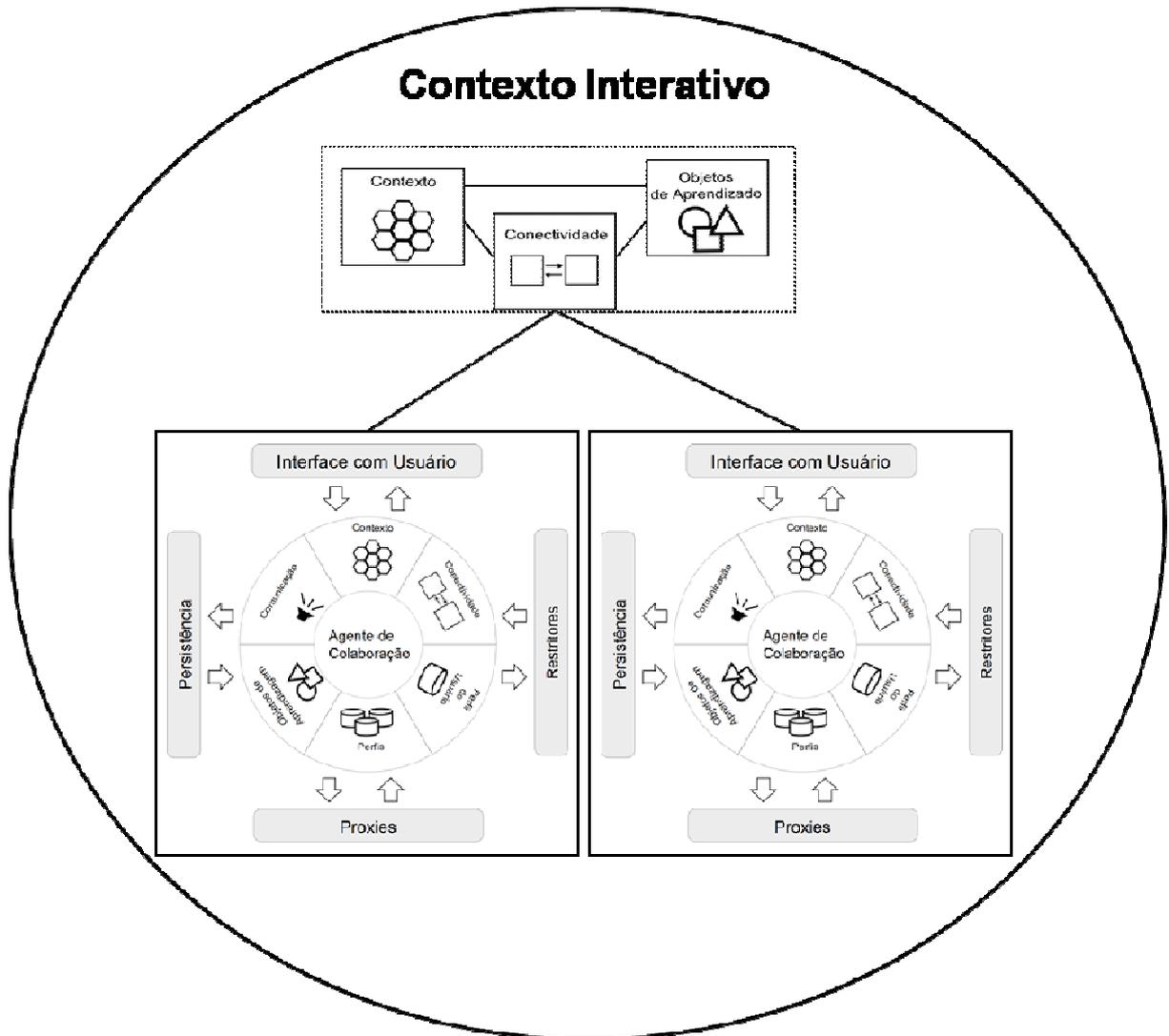


Figura 32 - Nova estrutura

A Figura 32 mostra a nova estrutura. Os agentes assistentes, propostos por Hahn, são formados por uma instância do Global com o agente de Colaboração. Esse agente faz a indicação de usuários e objetos de aprendizagem.

6.1.3 Simulação

Para comparar as funcionalidades das duas implementações é aplicada a mesma simulação utilizada no ambiente original ao ambiente desenvolvido utilizando o Global.

A simulação no Global foi implementada com o emulador da suíte *Sun Java Wireless Toolkit* (SJWT, 2010). Sendo inicializadas sete instâncias do emulador executando o Global, onde cinco representaram os alunos utilizando os agentes assistentes (alunos A, B, C, D e E) e duas representando os contextos interativos da biblioteca (Sala 1 e Sala 2). A movimentação dos aprendizes foi simulada através de um *script* XML, representando os deslocamentos no

espaço. As coordenadas foram obtidas pelo mapeamento de uma área real, sendo as mesmas utilizadas na simulação de Hahn. Nos próximos parágrafos são descritos os passos da simulação.

No ambiente de uma biblioteca, quatro aprendizes (no caso, alunos de mestrado) devem buscar material de suporte para resolver um exercício relacionado à disciplina de Visualização de Dados. O ambiente da simulação é composto de duas salas, mapeadas para dois contextos. Os alunos carregam consigo dispositivos móveis, que são mapeados para agentes aprendizes. Os livros presentes em cada contexto da biblioteca são mapeados como objetos de aprendizagem. A simulação se dá em quatro momentos distintos:

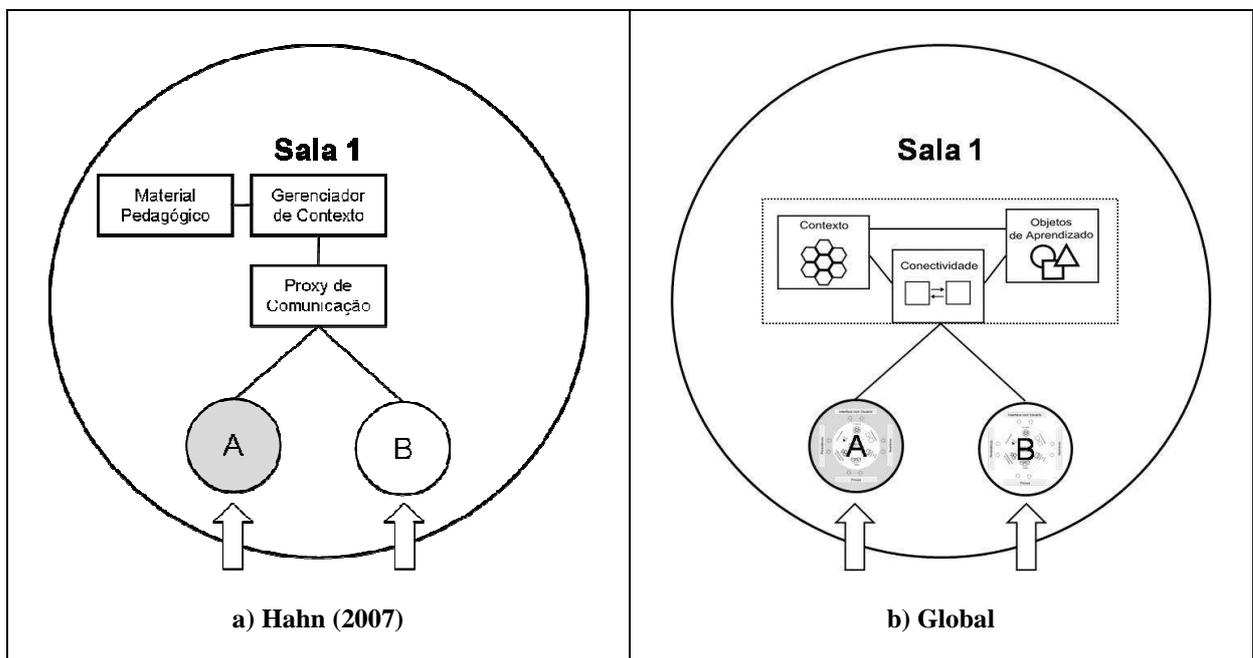


Figura 33 – Alunos A e B ingressam no contexto 1 (Sala 1)

- **Momento 1:** no momento inicial, dois alunos (A e B) chegam até o recinto da biblioteca, Sala 1. A Figura 33a mostra a execução na proposta de Hahn e a Figura 33b mostra a execução na implementação feita com o Global. O aluno A, além do trabalho da disciplina de Visualização de Dados, possui uma tarefa que exige a busca por material da disciplina de Lógica. Com base no contexto interativo atual, são disponibilizados objetos de aprendizagem relacionados à disciplina de Visualização de Dados. Assim sendo, o aluno A adianta-se em procurar material relacionado à Visualização de Dados, enquanto os outros colegas não chegam.

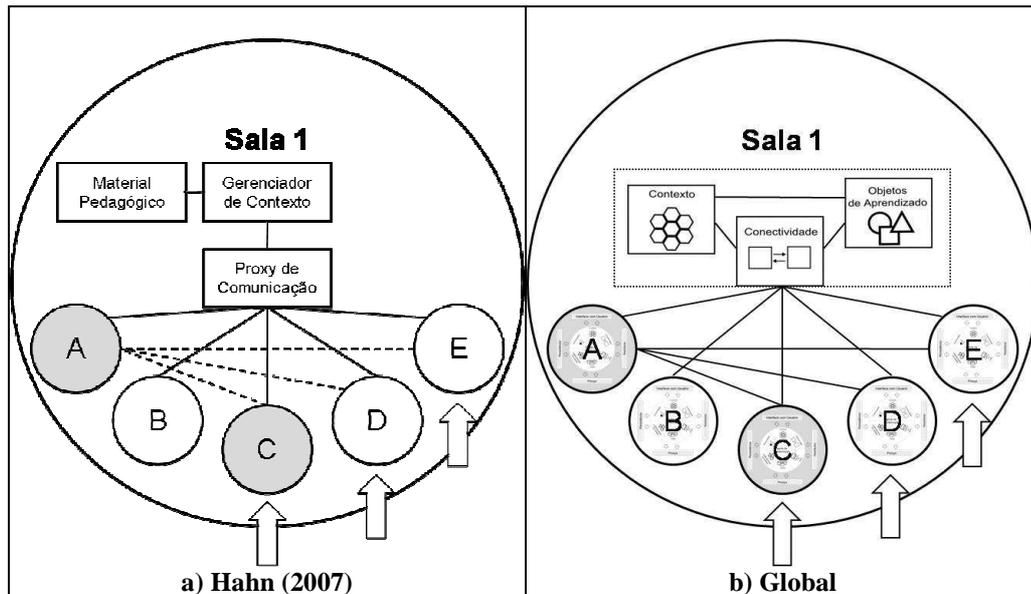


Figura 34 - Alunos C, D e E ingressam no contexto 1 (Sala 1)

- **Momento 2:** três novos alunos C, D e E, da mesma turma de A e de B, chegam ao recinto da biblioteca (Figura 34). Eles entram na mesma sala (mesmo contexto interativo) dos outros aprendizes, e começam a busca de material sobre Visualização de Dados. Todos os objetos de aprendizagem disponibilizados pelo contexto interativo com alguma relevância são recomendados aos alunos. Além disso, os três alunos recém-chegados recebem indicações que o aluno A tem interesse em material de apoio sobre a disciplina de Lógica. O aluno C possui interesse particular sobre o mesmo material;
- **Momento 3:** os alunos C, D e E discutem sobre os materiais recomendados. Durante a discussão, eles chegam a um acordo sobre quais materiais serão utilizados para o exercício;
- **Momento 4:** no momento seguinte (Figura 35), o aluno C procura o aluno A, e ambos discutem sobre a tarefa de Lógica. Eles decidem ir para outra sala da biblioteca, que contém material diretamente relacionado à disciplina. Enquanto o contexto interativo recomenda objetos de aprendizagem contextualizados, os alunos A e C analisam o material e, juntos, discutem a possibilidade da realização da tarefa com o material recomendado.

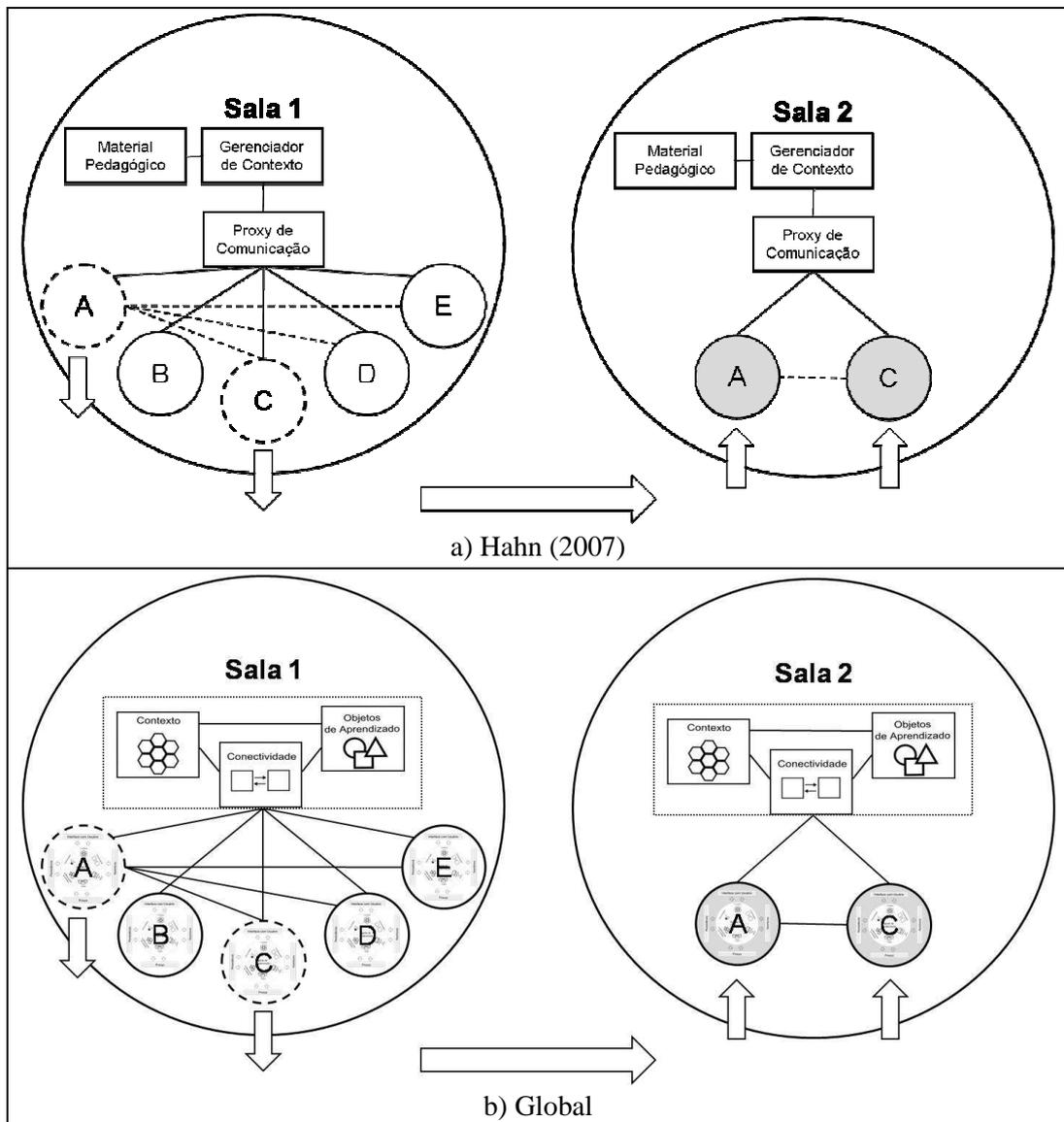


Figura 35 - Alunos A e C deixam o contexto 1 (Sala 1) e ingressam no 2 (Sala 2)

6.1.4 Conclusão

A infra-estrutura do Global se mostrou adequada para implementação da proposta de Hahn. A Tabela 4 mostra a comparação do que foi necessário implementar com o que a infra-estrutura do Global fornece. A infra-estrutura contempla a maioria das necessidades da proposta de Hahn, sendo que foi necessário implementar somente características pontuais como, por exemplo, as características de contextos, e características específicas da proposta como o algoritmo de indicação de semelhança entre perfil de usuários.

Tabela 4 - Especialização do Global para proposta de Hahn

Recurso	Necessário implementação	Fornecido pelo Global
Perfil	Interesses do usuário	Modelo PAPI
Gerência de Contexto	-	Gerência de Contexto
Conteúdo do contexto	Informações como: lista de palavras-chave e lista de objetos de aprendizagem disponíveis no contexto	<i>Global Information Tree</i> para criação do conteúdo dos contextos
Comunicação entre agentes	-	Agente de conectividade e Proxies
Determinação de Semelhança entre perfis	Algoritmo de comparação de perfil	Informações do perfil através do agente de Perfis
Gerência de Objetos de Aprendizagem	Adequação das informações dos objetos de aprendizagem	Gerenciamento de objetos de aprendizagem
Indicação de Objetos de Aprendizagem	Algoritmo de indicação de objetos de aprendizagem	Informações do perfil e objetos de aprendizagem

6.2 Integração com o CoolEdu

Essa avaliação visa testar a capacidade da infra-estrutura integrar novas funcionalidades com a integração de novos agentes, para isso foi utilizado o CoolEdu¹. O CoolEdu é um modelo para colaboração em ambientes descentralizados de educação ubíqua, capaz de auxiliar nos processos educacionais. Ele busca apoiar a educação através do estímulo a colaboração, utilizando para isso serviços de suporte do ambiente de educação ubíqua. Dessa forma, possibilita a utilização desse modelo em diferentes ambientes de educação ubíqua.

¹ Dissertação, do mestrando Solon Andrade Rabello Junior, apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos em Fevereiro de 2010.

6.2.1 Arquitetura do CoolEdu

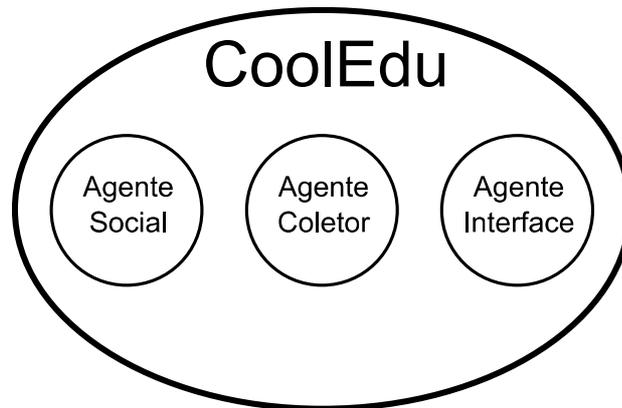


Figura 36 - CoolEdu

Assim como o Global, o CoolEdu também é modelado como um sistema multi-agentes. A Figura 36 mostra a arquitetura de agentes do CoolEdu, sendo dividida em três agentes:

- **Agente Social:** é o agente responsável por promover oportunidades pedagógicas, baseando-se para isso no conceito de zona de desenvolvimento proximal e os conceitos de ZDP de Vygotsky (1988), onde a aprendizagem ocorre pela união de pares de aprendizes, usando o conceito de “par mais capaz”;
- **Agente Coletor:** é o agente responsável pelo gerenciamento dos perfis dos usuários. Suas capacidades englobam atualizar, propagar e coletar perfil, suportando também os eventos de entrada em um novo contexto e a detecção de um novo aprendiz próximo. O agente Coletor também é responsável por informar ao agente Social qualquer alteração no perfil ou contexto;
- **Agente Interface:** é o agente responsável pela comunicação final com o aprendiz. É ele que define qual o modo de comunicação será utilizado, e quando a mensagem será exibida. Essa análise é feita de acordo com o perfil do aprendiz, mais precisamente suas preferências e histórico de utilização, e o contexto atual do aprendiz. De acordo com o ambiente que o aprendiz se encontra, a mensagem pode ser suprimida ou alterada para melhor acesso pelo aprendiz.

6.2.2 Requisitos de ambiente

O CoolEdu foi modelado para executar em ambientes descentralizados de educação ubíqua, sendo responsabilidade do ambiente o fornecimento de serviços de suporte a educação ubíqua. Estes serviços requeridos para a utilização do CoolEdu em um ambiente de educação

ubíqua, compreendem funcionalidades não cobertas por ele, mas que são necessárias a sua operação.

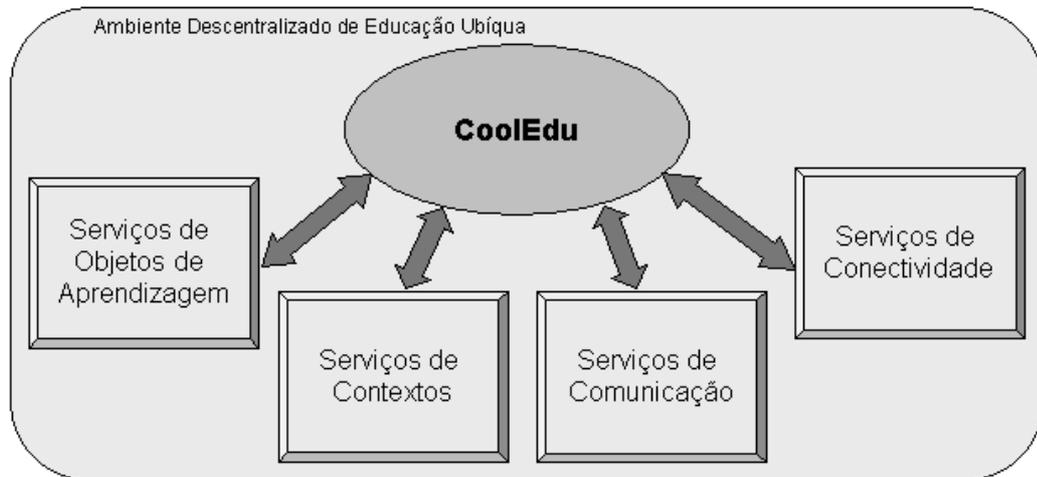


Figura 37 - Requisitos de ambiente CoolEdu

A Figura 37 apresenta serviços que o CoolEdu requer do ambiente ubíquo para seu correto funcionamento. Esses serviços são classificados em: Serviços de Objetos de Aprendizagem, Serviços de Contextos, Serviços de Comunicação e Serviços de Conectividade.

- **Serviços de Objetos de Aprendizagem:** É esperado que os serviços de objetos de aprendizagem armazenem, adquiram e distribuam os objetos de aprendizagem. Além de manipular os objetos de aprendizagem, é requisito do CoolEdu que esses serviços ofereçam metadados sobre os objetos no padrão IEEE/LTSC/LOM (LOM, 2010);
- **Serviços de Contextos:** Os serviços de contextos devem ser responsáveis pela criação e manutenção dos contextos, e monitoração de mudanças de estados no ambiente;
- **Serviços de Conectividade:** O CoolEdu requer que os serviços de conectividade realizem a troca de dados entre diferentes dispositivos. Esses serviços devem ser responsáveis pelo empacotamento e desempacotamento das mensagens no padrão FIPA-ACL.

6.2.3 Especialização do Global

O Global fornece os requisitos de ambiente necessários para execução do CoolEdu, mesmo assim para integração do CoolEdu alguns ajustes foram necessários. A seguir são descritos esses ajustes:

- O Global pode interagir com qualquer agente que utilize o padrão FIPA-ACL, como é o caso do CoolEdu. Mas para facilitar a comunicação entre esses agentes optou-se por

integrar diretamente os agentes na estrutura do Global. Para isso foi usado o padrão de projeto *Adapter* (Gamma et al, 2005), onde uma classe *Adapter* é usada para ligar a interface **Agent** do Global, com um agente do CoolEdu. A Figura 38 mostra como foi feita essa adaptação. Essa abordagem se mostrou necessária devido a adaptações de semântica e conteúdo de algumas mensagens trocas entre os agentes dos dois softwares;

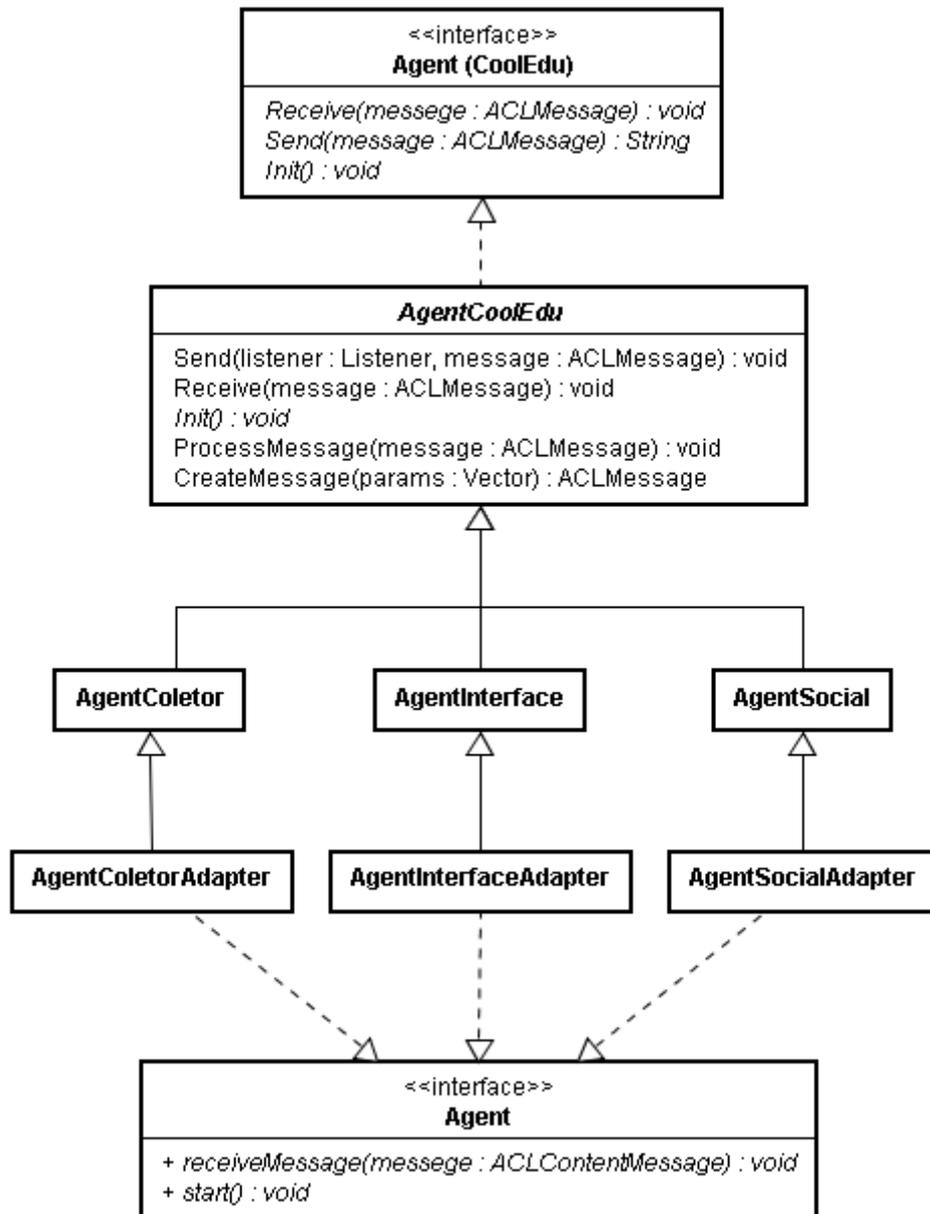


Figura 38 - Adaptação dos Agentes do CoolEdu

- Apesar do Global fornece os requisitos de ambiente necessários para execução do CoolEdu. O CoolEdu requer que as informações sejam entregues para ele pelos serviços de forma automática, sem nenhuma requisição por parte dos agentes do CoolEdu. Mas os agentes do Global só fornecem informações a outros agentes se solicitada ou se o agente se inscreve (*subscription*) pedindo a notificação de algum evento. Logo foi necessário que os *Adapters* dos agentes, no momento da inicialização (método *start*), se inscrevessem para receber as notificações que necessitam;
- O CoolEdu define que o usuário pode possuir dois tipos de relação com um contexto: o usuário está em um contexto ou o usuário faz parte de um contexto. Quando um usuário descobre um novo contexto o agente Social analisa se o contexto é de interesse do usuário, se for, questiona se o usuário quer fazer parte do novo contexto. Esse processo não é aplicado na distribuição de contexto do Global. Para adaptar esse comportamento foi criada uma nova restrição **ActiveContext** que é uma restrição booleana. Quando um novo contexto é descoberto, o agente Social é notificado (pela estrutura de subscrição), depois da análise e questionamento do usuário, caso ele queira participar a restrição **ActiveContext** é marcada como verdadeira, senão como falsa. Dessa forma se o **ActiveContext** esta marcado como verdadeiro o usuário faz parte de um contexto, se todas as outras restrições também são satisfeitas ele também está no contexto;
- Tanto o CoolEdu como o Global tem gerência de perfis. O Global pelos agentes Perfil do Usuário e Perfis, o CoolEdu pelo agente Coletor. Sendo que ambos utilizam o modelo PAPI como perfil de usuário, mas o CoolEdu acrescenta campos, necessário para o seu processo de análise. Para evitar uma gerência dupla de perfis, a classe Adapter do agente Coletor assume o papel dos dois agentes do Global. No processo de registro de agente, na inicialização do Global, ele é registrado três vez, como agente de Coletor, como agente de Perfil do Usuário e como agente de Perfis. A classe Adapter, também faz o de-para (tradução) do formato do perfil do CoolEdu para um perfil PAPI especializado do perfil original do Global, que comporta as novas informações. A Figura 39 mostra a especialização do perfil.

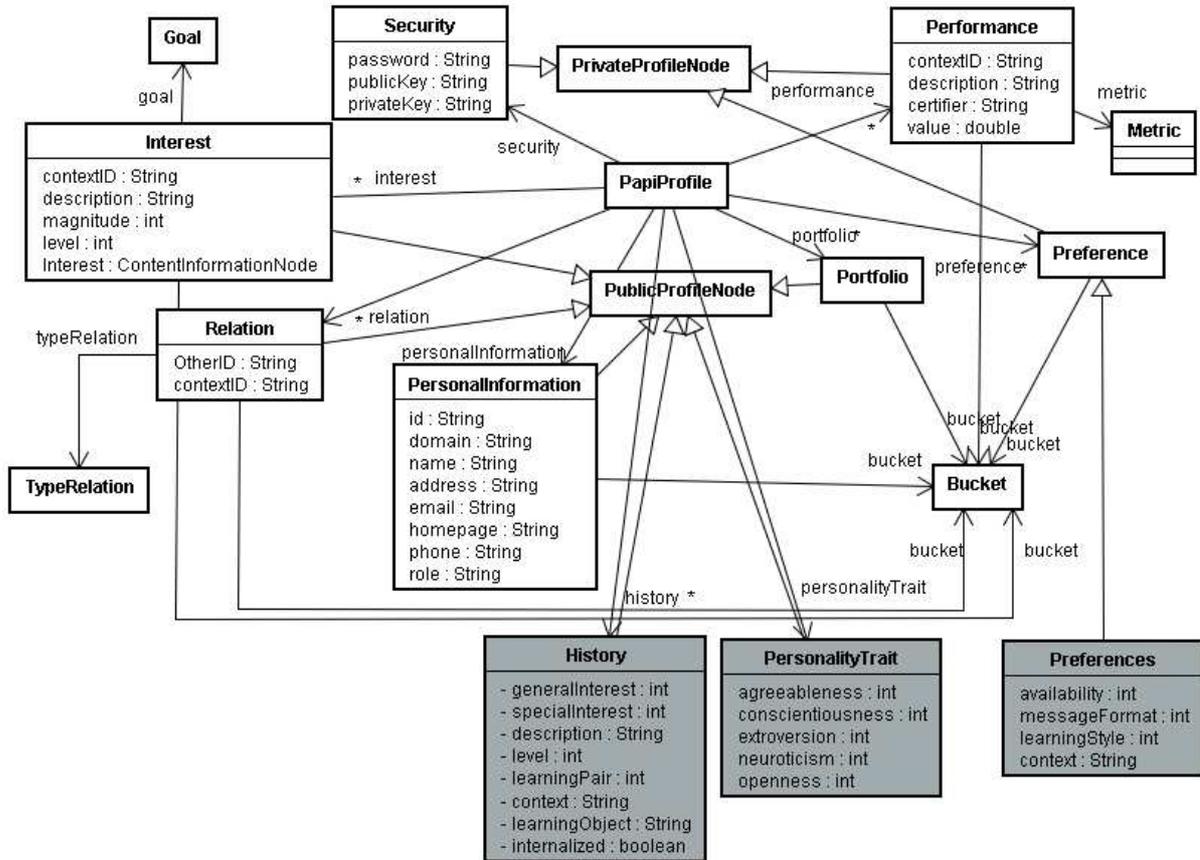


Figura 39 - Perfil pessoal da integração com CoolEdu

6.2.4 Simulação

Para avaliar a integração dos dois trabalhos foi utilizada uma simulação. A simulação foi feita com o emulador da suíte *Sun Java Wireless Toolkit* (SJWT, 2010). Sendo inicializadas seis instâncias do emulador executando o Global integrado ao CoolEdu, que representam os aprendizes (A,B,C,D, E e F). As informações de perfis, e os objetos de aprendizagem que possuem foram pré-carregados na inicialização das instâncias, a movimentação dos aprendizes foi simulada através de um *script* XML, representando os deslocamentos no espaço.

A situação que foi simulada é a seguinte: seis aprendizes, deslocam-se das suas casas até suas respectivas salas de aula. O deslocamento se dá através do metrô. A simulação se dá em cinco momentos distintos:

- **Momento 1:** os usuários se deslocam das suas casas até a estação. Na estação, os usuários entram no contexto TREM, o agente de Contexto (Global) do usuário A notifica o seu agente Social (CoolEdu), que solicita ao agente Coletor (CoolEdu) os perfis dos usuários presentes no contexto, o agente Coletor retorna os perfis dos usuários B, C e D. O agente Social, ao analisar o perfil do usuário A com os outros

perfis, encontra um interesse em comum com os usuários C e D (no caso, o gosto pelo jogo de xadrez) e calcula o índice de interação para ambos os usuários, C e D, chegando respectivamente nos valores 0,76 e 0,84. Usando como critério o maior índice, o agente Social notifica o agente Interface (CoolEdu), que tenta enviar uma sugestão de contato entre os usuários A e D, entretanto o agente Interface do usuário D nega o pedido de envio da mensagem, pois o usuário D configurou o contexto TREM como indisponível para recebimento de sugestões de criação de pares. O agente Social é notificado da negativa, pelo agente Interface, e requisita contato com o segundo da fila, o usuário C. Então o agente Interface do usuário A envia uma mensagem ao agente Interface do usuário C sugerindo que os aprendizes A e C entrem em contato. O agente Social do usuário C analisa o perfil do seu aprendiz e aceita a sugestão. Depois disso, cada instância envia a sugestão para seu respectivo utilizador. Essas sugestões são tratadas de acordo com as preferências configuradas por cada usuário;

- **Momento 2:** depois de trocar algumas mensagens através do agente de Comunicação (Global). O aprendiz C cria um contexto publico com uma restrição por interesse (xadrez) e outra por área (área próxima ao trajeto do TREM). Dessa forma o contexto seria uma forma de ocupar o tempo livre, gasto dentro do TREM, dos usuários com atividades relacionadas ao xadrez;
- **Momento 3:** os agentes de Contextos dos usuários acessíveis, pelo agente de Conectividade (Global) do usuário C, são notificados da criação do novo contexto pelo agente de Contexto do usuário C. O agente de Contexto do usuário D, que havia negado a interação iniciada pelo usuário A anteriormente, notifica o agente Social, que analisa o perfil do seu respectivo aprendiz. Ao constatar que o contexto criado é relacionado aos interesses do usuário D, o agente Social solicita ao agente Interface que notifica o usuário D;
- **Momento 4:** o usuário D, graças à recomendação recebida por parte da sua instância do agente Social, aceita participar do contexto criado anteriormente pelo usuário C. Além disso, o usuário D possui entre seus objetos de aprendizagem um tutorial sobre técnicas avançadas de xadrez. Através do agente de Contexto o usuário D disponibiliza o tutorial no contexto, em questão;
- **Momento 5:** os agentes Coletores dos usuários A e C, notificam seus usuários do novo objeto de aprendizagem no contexto. O usuário C opta por não adquirir o novo

objeto. Já o usuário A aceita o *download* do arquivo correspondente ao novo objeto e através do agente de Objetos de Aprendizagem (Global) inicia a aquisição dele.

6.2.5 Conclusão

Através da simulação executada é possível constatar que a integração do Global com o CoolEdu se mostrou funcional, apesar das adaptações necessárias nos contextos e perfis, a integração com as demais requisições de ambiente presentes no CoolEdu, como serviços de Objetos de Aprendizagem e Conectividade, não apresentaram grandes dificuldades. Mostrando que o Global é capaz de interagir com outros sistemas multi-agentes e até mesmo substituir seus agentes por outras implementações se feito os devidos ajustes.

6.3 Global Social Network

O Global Social Network (GlobalSN) é um protótipo de uma rede social ubíqua desenvolvido utilizando a infra-estrutura do Global. Esse protótipo serviu para avaliar a capacidade de extensão da infra-estrutura para aplicação não somente em ambientes de educação.

O GlobalSN forneça alguns recursos básicos a redes sociais tradicionais (Orkut, 2010 e facebook, 2010) agregando a elas características de ubiquidade, a seguir são descritos os recursos fornecidos pelo GlobalSN:

- Gerência de perfil pessoal: cadastro e edição de dados pessoais;
- Gerência de comunidades: criação comunidades, entrada em comunidades, criação de tópicos e resposta de tópicos;
- Gerência de amigos: adição de amigos, envio e recebimento mensagens.

Para criação do Global Social Network, foram utilizadas as API de Interface com Usuário, Persistência, Restritores, Proxies e os agentes de Contexto, Conectividade, Perfil do Usuário, Perfis e Comunicação. Além desses agentes foi criado um agente de Rede Social, que através da interação com os demais agentes disponibiliza os recursos de rede social.

6.3.1 Gerencia de perfil pessoal

Considerando que o modelo de perfil que o Global tem implementado é o PAPI, que é voltado para o processo de ensino e aprendizado. Para o GlobalSN foi desenvolvido um novo modelo de perfil, que utiliza a estrutura do *Global Information Tree*, o que garante a

utilização transparente do agente Perfil do Usuário e o agente Perfis. A Figura 40 mostra a estrutura do perfil social.

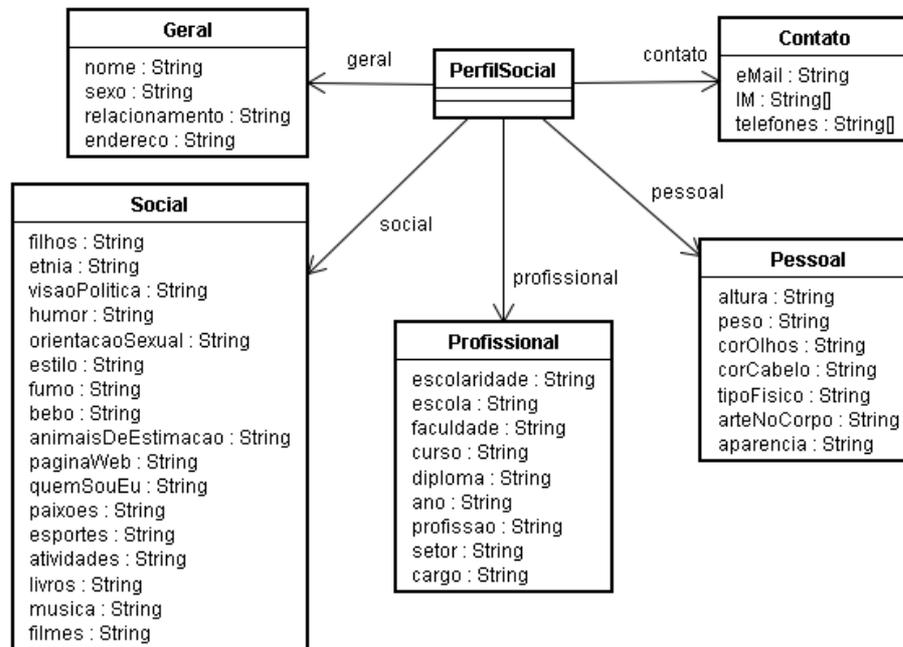


Figura 40 - Perfil Social

Para edição e visualização do perfil também foram criadas interfaces com usuário, utilizando a API do Global. A Figura 41 mostra algumas dessas interfaces.

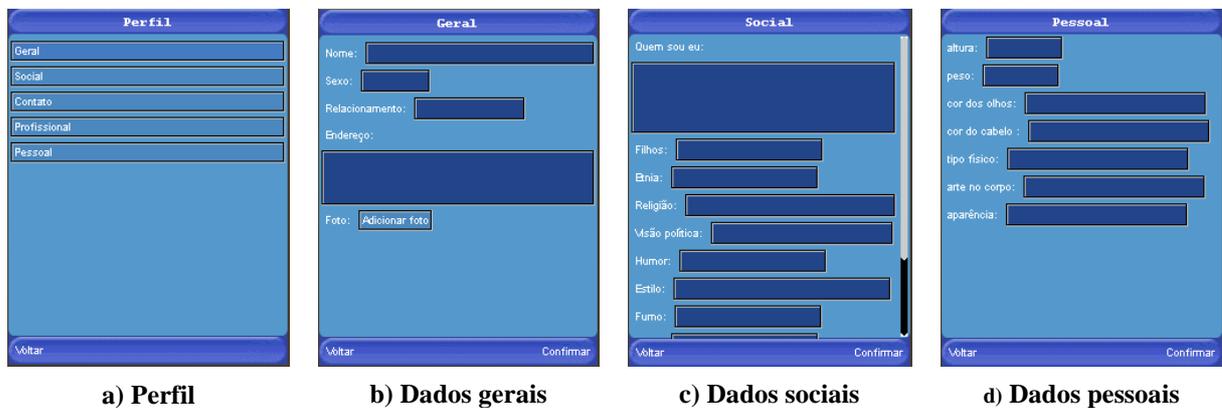


Figura 41 - Telas Perfil Social

6.3.2 Gerência de comunidades

Para gerência de comunidades foi utilizada a estrutura de contexto do Global, onde cada comunidade é um contexto. As vantagens dessa associação é que o agente de Contexto poder ser usado para gerenciar as comunidades e elas adquirem as características de ubiqüidade dos contextos do Global, se tornando comunidades ubíquas.

As informações da comunidade são guardadas na área de armazenamento de informações do contexto, utilizando também a estrutura do *Global Information Tree*, desta forma garantindo a atualização das informações da comunidade no ambiente descentralizado.

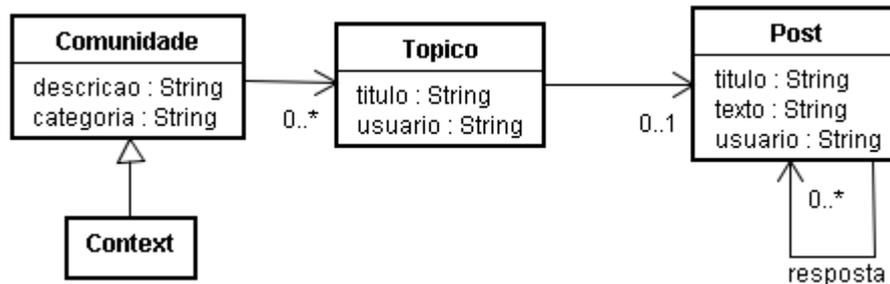


Figura 42 - Informações das Comunidades

A Figura 42 mostra a estrutura das informações das comunidades, onde cada comunidade tem uma descrição, uma categoria e uma lista de tópicos, que são criados pelos participantes. Cada tópico é formado por uma lista de postagens, que podem ser feitas por qualquer participante da comunidade. As postagens por sua vez possuem uma lista de postagem de resposta, que também podem ter uma lista de resposta.

Com a associação das comunidades aos contextos e suas informações, tópicos e postagens ao *Global Information Tree* permitem que suas atualizações sejam gerenciadas pelo agente de Contexto, evitando implementações adicionais para o controle de sincronismo dos tópicos e postagens. As regras de participação das comunidades são gerenciadas da mesma forma que nos contextos, através das restrições e Restritores.

Além disso, foram criadas interfaces com usuário, também utilizando a API do Global, para o cadastro de comunidades, criação de tópicos, postagens e resposta. A Figura 43 mostra algumas dessas interfaces.



a) Lista de comunidades

b) Criação de tópico

c) Lista de tópicos

d) Mensagens do tópico

Figura 43 - Telas da Gerência de Comunidade

6.3.3 Gerência de amigos

A gerência de amigos é feita através de um contexto especial, que todos os usuários têm por padrão. Esse contexto é centralizado no usuário e é formado por uma restrição do tipo lista de usuários. Assim, somente o usuário gerencia quem participa do grupo. O agente de Rede Social é responsável pela interface de requisição de convites para entrada nesse Contexto.

Também foram criadas interfaces com usuário, utilizando a API do Global, para a notificação de pedido de adição como amigo e visualização de perfil. Para envio e recebimento de mensagens é utilizado o agente de Comunicação que já possui todos os recursos para essa tarefa.

6.3.4 Conclusão

A infra-estrutura do Global se mostrou adequada para implementação do Global Social Network, graças ao arcabouço de recursos disponibilizados pelos agentes. Todo o processo de comunicação entre dispositivos e sincronização de informações em ambientes descentralizados foram abstraídos pelo Global. A Tabela 5 mostra as funcionalidades fornecidas pela infra-estrutura do Global e o que foi necessário implementar para garantir recursos propostos no Global Social Network.

Tabela 5 - Comparativo das funcionalidades

Recurso	Necessário implementação	Fornecido pelo Global
Cadastro e edição de dados pessoais	Interface com usuário*, Estrutura do perfil do usuário	Sincronismo de informação do usuário entre dispositivo
Criar comunidades	Interface com usuário*, Estrutura da comunidade	Sincronização das novas comunidades para os demais usuários
Entrar em comunidades	Interface com o usuário*	Estrutura para identificar se o usuário se encontra em um contexto (comunidade)
Criar tópicos	Interface com usuário*, Estrutura do tópico e postagem	Sincronização dos tópicos e postagens
Responder tópicos	Interface com usuário*	Sincronização das respostas
Adicionar amigos	Interface com usuário* Estrutura para solicitação de convite	Estrutura de contexto; Restritor por lista de usuários
Envio e recebimento de mensagens	–	Estrutura e envio, recebimento e armazenamento do agente de Comunicação

* Com o suporte da API de interface fornecida pelo Global.

6.4 Considerações sobre o Capítulo

Esse capítulo apresentou três aplicações desenvolvidas a partir do protótipo do Global para sua avaliação, fazendo um comparativo do ganhos trazidos pela infra-estrutura.

No próximo capítulo serão apresentadas as considerações finais da dissertação.

7 CONSIDERAÇÕES FINAIS

Este capítulo apresenta uma reflexão sobre as principais contribuições do Global, comparando-o com os trabalhos anteriormente estudados. Complementando tal discussão ainda são apresentados os trabalhos futuros.

7.1 Contribuições

Este trabalho apresentou a proposta do Global, uma infra-estrutura descentralizada de educação ubíqua, baseada em sistemas multi-agentes. Para uma maior adequação aos conceitos da computação ubíqua definida por Satyanarayanan (2001), o Global propõe uma estrutura descentralizada baseada em redes *Ad Hoc*.

A Tabela 6 é uma extensão da Tabela 2, mostrando um comparativo entre os trabalhos descritos no Capítulo 3 com o Global. Em comparação com as demais propostas, o Global apresenta todas as características da computação ubíqua analisadas na tabela. Além disso, é uma proposta extensível. A seguir é descrito como o Global garante tais características:

- Sensibilidade de localização: é garantida pela restrição *GlobalPositionRestriction* que representa a localização física do usuário;
- Comunicação entre dispositivos: é garantida pelo componente de *Proxies*, que também é uma arquitetura genérica para comunicação entre dispositivos, sendo que o Global suporta por padrão Bluetooth e Wi-Fi;
- Redes móveis: através de um algoritmo de roteamento presente no *Proxy Bluetooth*, mensagens podem ser roteadas entre dispositivos, formando assim redes móveis;
- Segurança distribuída: garantida pela troca de mensagens criptografadas, e pela estrutura de chaves públicas e privadas propostas pelo agente de *Conectividade*;
- Sensibilidade ao contexto: o agente de *Contexto*, junto com os *Restritores*, consegue mapear características de ambiente, assim criando contextos. Com monitoramento das alterações dessas características, o Global pode alterar seu comportamento, assim garantindo sensibilidade ao contexto;
- Escalabilidade localizada: a tecnologia Bluetooth já opera de forma localizada, limitando o número de saltos da rede *Ad Hoc* é garantida uma maior escalabilidade localizada;
- Acesso móvel à informação: as informações são armazenadas no próprio dispositivo, sendo acessíveis a qualquer momento, assim garantindo acesso móvel à informação;

- Extensível: o Global utiliza uma modelagem multi-agentes. Agentes são entidades que operam de forma autônoma e independente. Um agente não precisa ter conhecimento da estrutura de outro agente para interagir, bastando para tanto o conhecimento do protocolo de comunicação para requisição da funcionalidade desejada. Desta forma em sistemas multi-agentes, agentes poder ser adicionados, removidos ou substituídos sem afetar o sistema como um todo. Esta característica além de trazer uma alta modularidade, facilitando a adição, remoção e alteração das funcionalidades do sistema.

Tabela 6 - Comparativo com Global

	Japelas	Framework de interação social de Zhang	Ambiente de aprendizagem ubíqua para a aprendizagem colaborativa P2P	GlobalEdu	Global
Objetivo	Apoiar a aprendizagem de expressões de tratamento na língua Japonesa.	Apoiar a aprendizagem dos alunos focando em suas habilidades sociais em um ambiente ubíquo.	Criação de comunidades virtuais, onde instrutores e alunos, descobrem, acessão e compartilham recursos.	Dar as condições necessárias para que o aprendiz construa seu conhecimento de qualquer lugar e em qualquer tempo, de forma adaptada ao seu perfil e contexto.	Apoiar a aprendizagem através da análise dos dados de perfis e de contexto em ambientes ubíquos em redes <i>Ad Hoc</i> .
Sensibilidade de localização	tag's RFID.	Não possui sistema de localização.	Combinação de tecnologias com trian-gulação de antenas.	Sim, a forma depende do <i>middleware</i> de execução.	GPS, extensível para outras tecnologias.
Comunicação entre dispositivos	IR	P2P, não especificado.	P2P e cliente servidor.	Depende do <i>middleware</i> de execução.	<i>Bluetooth</i> e WiFi. Extensível para outras tecnologias.
Redes móveis	Parcial, não apresenta algoritmo de roteamento.	Sim	Sim	Depende do <i>middleware</i> de execução.	Sim
Segurança distribuída	Não especificado.	Não especificado.	Não especificado.	Depende do <i>middleware</i> de execução.	Sim
Sensibilidade ao contexto	Sim.	Não	Sim	Sim	Sim
Escalabilidade e localizada	Sim	Sim	Sim	Depende do <i>middleware</i> de execução.	Sim
Acesso móvel à informação	Sim	Sim	Parcial, parte da informação fica no servidor.	Não, as informações ficam armazenadas no servidor.	Sim
Extensível	Não, focado apenas no tratamento de expressões do idioma japonês.	Parcialmente, apesar de ser um <i>framework</i> só permite a personalização dos tutores.	Não	Parcialmente, também permite a customização do tutor.	Sim

Através das três aplicações desenvolvidas foi possível verificar o ganho trazido pela infra-estrutura do Global, tanto na aplicação em ambientes de educação ubíqua, quando em outras áreas, no caso redes sociais.

Uma versão inicial do trabalho também foi submetido para avaliação no 20º Simpósio Brasileiro de informática na Educação, sobre o título "Um Modelo Multi-agente Descentralizado para Ambientes de Educação Ubíqua" (Oliveira et al., 2009), sendo aceito para publicação.

7.2 Trabalhos Futuros

Ao longo da realização deste trabalho foram identificadas idéias e propostas para trabalhos futuros. Algumas destas idéias são apresentadas a seguir:

- Apesar da comunicação entre os agentes utilizar FIPA-ACL, o conteúdo não segue uma padronização, sendo utilizado serialização de objetos no protótipo. Uma proposta para melhoria da infra-estrutura é utilizar um padrão de conteúdo para comunicação de agentes no caso o FIPA RDF (FIPA-RDF, 2010);
- Adição de novas restrições e Restritores, como leitores de RFID (Feldhofer et al., 2004) e QRCODE (DENSO, 2010);
- Adicionar novos Proxies de comunicação;
- Portar o protótipo para outras plataformas móveis como Android (2010) e Iphone OS (2010);
- Criar um agente de interação com usuário, utilizando técnicas mais sofisticadas, como por exemplo agentes emotivos, para interação do usuário com os agentes do Global;
- Criar um manual de utilização da infra-estrutura para facilitar as especializações futuras do Global.

REFERÊNCIAS

- Android, 2010. **Android**. Disponível em: <<http://www.android.com>>. Acesso em: 21 fev. 2010.
- Barbosa, D. N. F.; Barbosa, J. L. V.; Geyer, C. F. R. **Suportando a Educação Pervasiva - consciência do contexto do aprendiz**. Simpósio Internacional de Informática na educação, SBIE, 8., 2006.
- Barbosa, D. **Um Modelo de Educação Ubíqua Orientado à Consciência do Contexto do Aprendiz**. Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.
- Barbosa, J.; Hahn, R.; Rabello, S.; Barbosa, D. **Local: a model geared towards ubiquitous learning**. Proceedings of the 39th SIGCSE technical symposium on Computer science education, pages 432–436, 2008.
- Becker, F. **Educação e construção do conhecimento**. Porto Alegre: Artmed, 2001.
- Bomsdorf, B. **Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education**. Mobile Computing and Ambient Intelligence: The Challenge of Multimedia, 2005.
- Bradshaw, J. **Software Agents**. Cambridge: MIT p. 490, 1997.
- Chen, C.; Li, Y.; Chen, M. **Personalized Context-Aware Ubiquitous Learning System for Supporting Effectively English Vocabulary Learning**. Seventh IEEE International Conference Advanced Learning Technologies, 2007.
- Chen, Y.S.; Kao, T.C.; Sheu, J.P.; Chiang, C.Y. **A Mobile Scaffolding-Aid-Based Bird - Watching Learning System**. Proceedings of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02), pp.15-22, IEEE Computer Society Press, 2002.
- Comer, D. E.; Yavatkar, R. **FLAWS: Performance guarantees in best effort delivery systems**. Proc. INFOCOM, Ottawa. Canada, 1989.
- Curtis, M.; Luchini, K.; Bobrowsky, W.; Quintana, C.; Soloway, E. **Handheld Use in K-12: A Descriptive Account**. Proceedings of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02), pp.23-30, IEEE Computer Society Press, 2002.

DENSO Wave. **QR code**. ISO/IEC18004. Disponível em: <<http://www.denso-wave.com/qrcode>>. Acesso em: 21 fev. 2010.

Dey, A. K. **Understanding and Using Context**. Personal and Ubiquitous Computing, Volume 5, Number 1 / February, 2001.

Edwards, W. K.; Bellotti, V.; Dey, A. K.; Newman, M. W. **The challenges of user-centered design and evaluation for infrastructure**. pages 297–304, Ft. Lauderdale, Florida, USA. ACM, 2003.

El-bishouty, M. M.; Ogata, H.; Yano, Y. **PERKAM: Personalized Knowledge Awareness Map for Computer Supported Ubiquitous Learning**. Educational Technology & Society, 10, 2007.

Facebook, 2010. **Facebook**. Disponível em: <<http://www.facebook.com>>. Acesso em: 21 fev. 2010.

Feldhofer, M.; Dominikus, S.; Wolkerstorfer, J. **Strong authentication for RFID systems using the AES algorithm**. Proc. Workshop on Cryptographic Hardware and Embedded Syst. Lecture Notes in Computer Science, pp. 357–370, 2004.

Ferguson, N.; Schneier, B. **Practical cryptography**. Wiley. ISBN 0471223573, 2003.

FIPA-ACL. **Foundation of Intelligent Physical Agents - Agent Communication Language**. Disponível em: <<http://www.fipa.org/specs/fipa00061>>. Acesso em: 21 fev. 2010.

FIPA-RDF. FIPA00011: **FIPA RDF Content Language Specification**. Disponível em: <<http://www.fipa.org/specs/fipa00011/>>. Acesso em: 21 fev. 2010.

FIPA-Representation. FIPA00070: **FIPA ACL Message Representation in String Specification**. Disponível em: < <http://www.fipa.org/specs/fipa00070/SC00070I.html> >. Acesso em: 21 fev. 2010.

Galanxhi-Janaqi, H.; Nah, F. F. **U-commerce: emerging trends and research issues**. Industrial Management & Data Systems, Volume 104, Número 9, 2004.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA: Addison Wesley, 1995.

Ghassemlooy, Z.; Hayes, A.R. **Digital pulse interval modulation for IR communication systems a review**. International Journal of Communication Systems; 13:519–536, 2000.

Global Locate Inc. **A-GPS Technology**. Disponível em: <http://www.globallocate.com/A-GPS/A-GPS_Frameset.htm>. Acesso em: 21 fev. 2010.

Gomes, E. **Objetos Inteligentes de Aprendizagem: uma abordagem em agentes para objetos de aprendizagem**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, UFRGS. Dissertação de Mestrado, 2005.

Haartsen, J et al. **Bluetooth: Vision, goals, and architecture**. CM Mobile Computing and Communications Review, 1998.

Hahn, R. M. **Uma arquitetura de Assistente Pessoal Orientada a Ambientes de Aprendizagem Ubíqua**. São Leopoldo, RS, Brasil. Trabalho de conclusão de curso – Universidade do Vale do Rio dos Sinos, Unisinos, 2007.

Hahn, R.; Barbosa, J. L. V. **Uma Arquitetura de Assistente Pessoal Orientada a Ambientes de Aprendizagem Ubíqua**. XIX Simpósio Brasileiro de Informática na Educação (SBIE), Fortaleza. Anais do XIX SBIE. Porto Alegre, 2008.

Hightower, J.; Borriella, G. **Location systems for ubiquitous computing**. IEEE Computer, 34(8):57–66, 2001.

Hightower, J; LaMarca, A.;Smith, I. **Practical Lessons from Place Lab**. IEEE Pervasive Computing, vol. 5, no. 3, 2006.

Ieee. **Ieee standard for information part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications**. Disponível em: <<http://standards.ieee.org/getieee802/download/802.11-2007.pdf>>. Acesso em: 21 fev. 2010.

IMS. Global Learning Consortium, Inc. **IMS Learner Information Package - Information Model**. v. 1.0. 2001a. Disponível em: <<http://www.imsglobal.org/profiles/lipinfo01.html>>. Acesso em: 21 fev. 2010.

IMT-2000. **International Mobile Telecommunications-2000 (IMT-2000)**. Disponível em: <<http://www.itu.int/home/imt.html>>. Acesso em: 21 fev. 2010.

iPhone OS. **iPhone OS**. Disponível em: <<http://www.apple.com/iphone>>. Acesso em: 21 fev. 2010.

JADE, 2010. **Java Agent DEvelopment Framework**. Disponível em: <<http://jade.cselt.it>>. Acesso em: 21 fev. 2010.

Jin, Q. **Learning Community Meets Peer-to-Peer Networking: Towards an Innovative Way of Person-to-Person E- Learning Support**. Proc. of KEST (Knowledge Economy and Development of Science and Technology), 2003.

Jones, V.; Jo J. **Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology**. Proceedings of the 21st ASCILITE Conference, 2004.

JSR-118. **Mobile Information Device Profile 2.0**. Disponível em: <<http://www.jcp.org/en/jsr/detail?id=118>>. Acesso em: 21 fev. 2010.

JSR-139. **Connected Limited Device Configuration 1.1**. Disponível em: <<http://www.jcp.org/en/jsr/detail?id=139>>. Acesso em: 21 fev. 2010.

JSR-179. **Location API for J2ME**. Disponível em: <<http://www.jcp.org/en/jsr/detail?id=179>>. Acesso em: 21 fev. 2010.

JSR-82. **Java APIs for Bluetooth**. Disponível em: <<http://www.jcp.org/en/jsr/detail?id=082>>. Acesso em: 21 fev. 2010.

Li, L.; Zheng, Y.; Ogata, H.; Yano, Y. **Ubiquitous Computing in Learning: Toward a Conceptual Framework of Ubiquitous Learning Environment**. Communications, 1(3), 207-215, 2005.

LOM. Ieee/ltsc/lom learning technology standards committee. **draft standard for learning object metadata**. Disponível em: <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf >. Acesso em: 21 fev. 2010.

LWUIT. **Lightweight UI Toolkit (LWUIT)**. Disponível em: <<https://lwuit.dev.java.net> >. Acesso em: 21 fev. 2010.

Miller, E. **An introduction to the resource description framework**. Reading D-Lib Magazine, maio, 1998.

Moll, J.; Silveira, B. M. C. **Construtivismo: desconstituindo mitos e constituindo perspectivas**. In: BECKER, F.; FRANCO, S. R. K. Revisitando Piaget. 2 ed. Porto Alegre: Meditação, 1999.

Ogata, H.; Yano, Y. **Knowledge Awareness Map for Computer-Supported Ubiquitous Language- Learning**. Proc. 2nd IEEE WMTE, 2004.

Oliveira, J. M.; Rabello, S.; Barbosa, J. L. V. **Um Modelo Multi-agente Descentralizado para Ambientes de Educação Ubíqua**. Simpósio Internacional de Informática na educação, SBIE, 2009.

Orkut, 2010. **Orkut**. Disponível em: <<http://www.orkut.com.br>>. Acesso em: 21 fev. 2010.

Padgham, L.; Winikoff, M. **Prometheus: a methodology for developing intelligent agents**. Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, Bologna, Italy, 2002.

PAPI. Draft standard for learning technology. **Public and private information (papi) for learners (papi learner)**. Disponível em: <<http://www.cen-ltso.net/main.aspx?put=230&AspxAutoDetectCookieSupport=1>>. Acesso em: 21 fev. 2010.

Qualcomm. **gpsOne Technology**. Disponível em: <<http://www.cdmatech.com/solutions/products/gpsone.jsp>>. Acesso em: 21 fev. 2010.

Rajagopalan, S.; Shen, C. **A cross-layer decentralized bittorrent for mobile ad hoc networks**. Mobile and Ubiquitous Systems - Workshops. 3rd Annual International Conference on, pages 1–10, 2006.

Rigaux, P.; Spyrtos, N. **Selene report: Metadata management and learning object composition in a self elearning network**. <http://www.dcs.bbk.ac.uk/selene/reports>, 2007.

Royer, E.; Toh, C. **A Review of Current Routing Protocols for Ad Hoc Wireless Networks**. IEEE Pers. Commun, 1999.

Russell, S. J.; **Inteligência Artificial**. CAMPUS, 2ª ed.: 2004. Rio de Janeiro: RJ. ISBN 13:978-85-352-1177-1, 2004.

Saha, D.; Mukherjee, A. **Pervasive computing: a paradigm for the 21st century**. Computer, Los Alamitos, v. 36, n. 3, p. 25–31, Mar, 2003.

Satyanarayanan, M. **Pervasive computing: vision and challenges**. Personal Communications, IEEE [see also IEEE Wireless Communications], v. 8, n. 4, p. 10–17, 2001.

Serain, D. **Middleware and Enterprise Application Integration**. Springer, 2002.

Shoham, Y. **Agent-oriented programming**. Technical Report STAN-CS-1335-90, Computer Science Department, Stanford University, Stanford, CA 94305, 1990.

SJWT. **Sun Java Wireless Toolkit**. Disponível em: <<http://java.sun.com/products/sjwtoolkit/>>. Acesso em: 21 fev. 2010.

Vygotsky, L.; Luria, A.; Leontiev. N.; **Linguagem, desenvolvimento e aprendizagem.** volume 10, editora Ícone, São Paulo, 1988.

W3C-DOM. The World Wide Web Consortium (W3C). **W3C Document Object Model.** Disponível em: <<http://www.w3.org/DOM>>. Acesso em: 21 fev. 2010.

Weiser, M. **The computer for the 21st century.** **SIGMOBILE Mob.** Comput. Commun. Rev., ACM Press, New York, NY, USA, v. 3, n. 3, p. 3–11, July, 1999.

Yang, S. J. **Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning.** Educational Technology & Society, 9, 188-201, 2006.

Yau, S. **SmartClassroom: Enhancing Collaborative Learning Using Pervasive Computing Technology.** II American Society of Engineering Education (ASEE), 2003.

Yin, C.; Ogata, H.; Yano Y.; Oishi Y. **JAPELAS: Supporting Japanese Polite Expressions Using PDA(s).** Towards Ubiquitous Learning, 2005.

Zhang, G.; Jin, Q.; Lin, M. **A Framework of Social Interaction Support for Ubiquitous Learning.** Proceedings of the 19th international Conference on Advanced information Networking and Applications - Volume 2 (March 25 – 30), 2005.