

Modelagem Multiagentes de uma Infraestrutura Descentralizada para Aprendizagem Ubíqua

Jorge Luis Victória Barbosa¹, Jezer Machado de Oliveira¹
Débora Nice Ferrari Barbosa², Sandro José Rigo¹

Universidade do Vale do Rio dos Sinos (UNISINOS)¹
Universidade FEEVALE²

jbarbosa@unisinis.br, jezeroliveira@gmail.com,
deboranice@feevale.br, rigo@unisinis.br

Resumo. *GLOBAL é uma infraestrutura descentralizada para ambientes de aprendizagem ubíqua, baseada em agentes de software. A partir da extensão dos seus agentes ou adição de novos, a infraestrutura pode ser especializada para a criação de ambientes de aprendizagem ubíqua. Este artigo contribui apresentando a modelagem da infraestrutura através de uma metodologia multiagentes. Nesse trabalho, GLOBAL foi completamente modelada usando a metodologia Prometheus. Além disso, o artigo discute como os agentes foram implementados e como foram integrados a um sistema dedicado à colaboração em ambientes de aprendizagem ubíqua.*

Palavras-Chaves: *Computação Ubíqua, Aprendizagem Ubíqua, Infraestrutura, Sistemas Multiagentes.*

Abstract. *GLOBAL is a decentralized ubiquitous learning infrastructure, based on software agents. By extending its agents, or adding new agents, GLOBAL can be customized to create ubiquitous learning environments. This article contributes by presenting the infrastructure modeling through a multiagent methodology. Global was fully modeled using the Prometheus methodology. Moreover, the article discusses as the agents were implemented and how they have been integrated with a system dedicated to support collaboration in ubiquitous learning environments.*

Keywords: *Ubiquitous Computing, Ubiquitous Learning, Infrastructure, Multiagent Systems.*

1. Introdução

Atualmente, os estudos sobre mobilidade em Sistemas Distribuídos vêm sendo impulsionados pela proliferação de dispositivos móveis e pela exploração de novas tecnologias de interconexão baseadas em comunicação sem fio. Este novo paradigma distribuído e móvel é denominado Computação Móvel (Diaz et al., 2010). Além disso, a mobilidade aliada à difusão da comunicação sem fio permitiu aos serviços computacionais serem disponibilizados em contextos específicos – Computação Baseada em Contextos (Bellavista et al., 2012). Neste cenário, pesquisas adicionais relacionadas com a infraestrutura para software (Costa et al., 2008) vêm permitindo o surgimento do suporte computacional contínuo, a qualquer momento e em qualquer lugar – Computação Ubíqua (Caceres et al., 2013). A educação, assim como outras áreas de conhecimento, vem usufruindo dessas novas tecnologias para melhoria de suas

práticas e abordagens – Aprendizagem Ubíqua (Barbosa et al., 2001)(Wagner et al., 2014).

Um sistema computacional ubíquo deve atender certas características, tais como invisibilidade, escalabilidade, adaptação, disponibilidade, tolerância a falhas, entre outras (Satyanarayanan, 2001). No entanto, a maioria dos ambientes de aprendizagem ubíqua adota a abordagem centralizada. Os ambientes ubíquos centralizados não se adéquam, a priori, a duas das características previamente definidas como essenciais em um ambiente ubíquo (Satyanarayanan, 2001): (1) Escalabilidade, pois o processamento e o armazenamento são limitados a poucas máquinas, caracterizando assim um ponto no sistema onde a procura por recursos pode ultrapassar a oferta; e, (2) Disponibilidade, pois em arquiteturas centralizadas qualquer problema de conexão entre os dispositivos e os servidores resulta em indisponibilidade do serviço. Nesse cenário tecnológico foi proposta uma infraestrutura dedicada ao desenvolvimento de ambientes descentralizados de aprendizagem ubíqua, denominada GLOBAL (Oliveira et al., 2012).

Os sistemas multiagentes (Jennings et al., 1998) 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 conceito de sistemas multiagentes estabelece que através do comportamento individual de cada agente, o grupo de agentes alcançará o seu comportamento inteligente.

O desenvolvimento de ambientes ubíquos é complexo. Em função disso, entende-se que a abordagem orientada a agentes de software é adequada para o desenvolvimento deste tipo de sistema (Jennings, 2001). Por sua vez, Prometheus (Padgham et al., 2004) é uma metodologia detalhada de especificação para sistemas multiagentes. A Tabela mostra o significado dos símbolos utilizados nessa especificação.

Este artigo contribui apresentando a modelagem da GLOBAL através da metodologia Prometheus (Padgham et al., 2004). O texto descreve ainda como os agentes foram implementados e integrados a um sistema orientado à colaboração em ambientes de aprendizagem ubíqua, denominado CoolEdu (Rabello et al., 2012). O artigo está organizado em cinco seções. A segunda seção apresenta a modelagem multiagentes da Global. Por sua vez, a terceira descreve como os agentes foram implementados e descreve a integração com o CoolEdu. A seção quatro apresenta considerações finais, principalmente indicando trabalhos futuros. Finalmente, a quinta seção contém as referências bibliográficas.

Tabela 1. Especificação Prometheus (Padgham et al., 2004).

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

2. Modelagem multiagentes da GLOBAL

A GLOBAL é formada por dez componentes (Figura 1) (Oliveira et al., 2012). 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 da GLOBAL

em um dispositivo contém uma instância de cada agente, tornando assim o ambiente completamente descentralizado.

O agente de Conectividade realiza a troca de mensagens entre agentes, encontrando a melhor forma para entregar a mensagem, sendo de forma direta entre agentes de uma mesma instância, ou pela escolha de um Proxy que tenha acesso ao dispositivo do agente destinatário, caso sejam agentes de instâncias diferentes. O agente de Contexto gerencia os contextos, mapeando a movimentação do usuário e alertando os demais agentes sobre as alterações relacionadas. 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 ele tem ou teve contato. O agente de Objetos de Aprendizagem armazena, disponibiliza e adquire os objetos de aprendizagem (OAs) utilizados pelos usuários. Por fim, o agente de Comunicação é responsável pela troca de mensagens entre os usuários.

Os quatro componentes de suporte disponibilizam funcionalidades básicas para os agentes. Interface com Usuário suporta o acesso dos usuários às funcionalidades da 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 (Oliveira et al., 2012). Os Proxies são uma camada de abstração para troca de informações em redes *Ad Hoc*. As próximas seções apresentam a modelagem da GLOBAL através da metodologia Prometheus (Padgham et al., 2004).

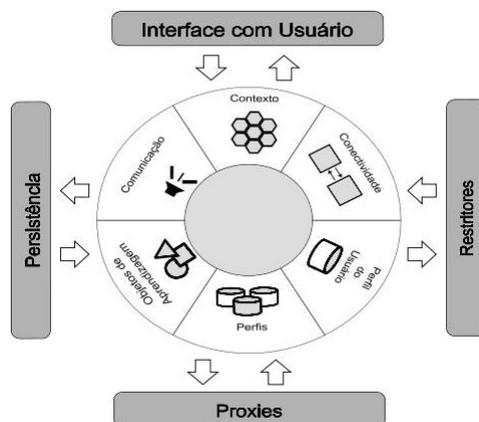


Figura 1. Arquitetura da GLOBAL (Oliveira et al., 2012).

3.1. Diagrama geral da Global

A Figura 2 mostra o diagrama geral da GLOBAL, contendo a comunicação entre os agentes através de troca de mensagens baseada em protocolos de comunicação. Por sua vez, a Figura 3 mostra o diagrama de troca de mensagens do protocolo de *HandShake*. O diagrama geral mostra uma instância da 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 da GLOBAL.

Os agentes 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 permite a agentes que desempenham funcionalidades passivas, não ficarem se comunicando continuamente com os demais agentes para saber se tiveram alguma alteração, bastando aguardar o retorno por parte do agente que gerencia a informação.

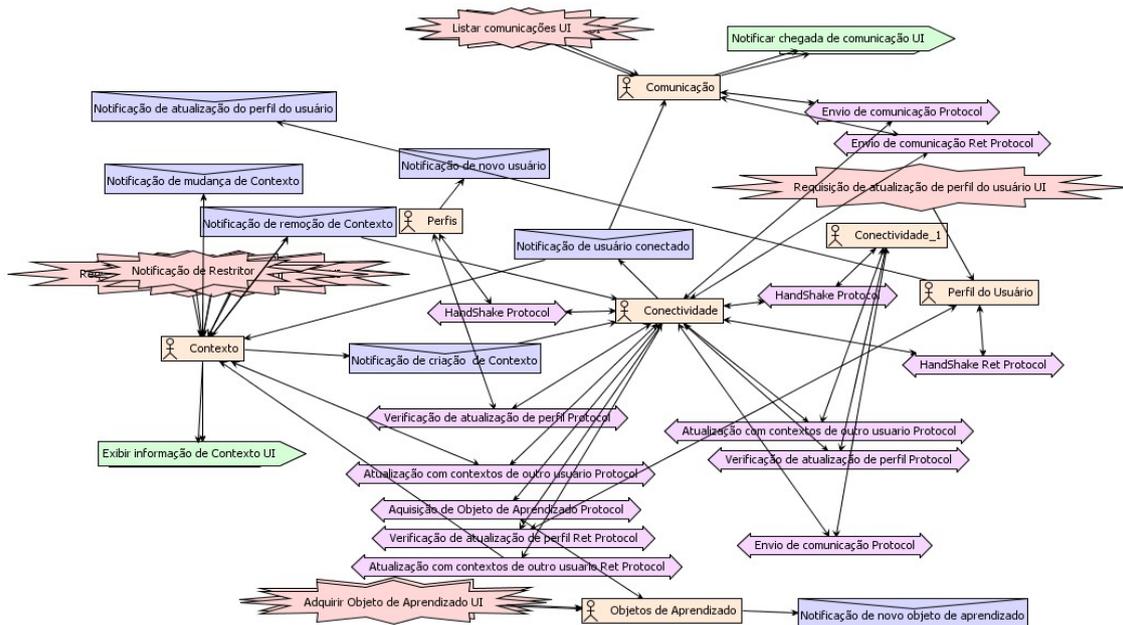


Figura 2. Diagrama geral da Global

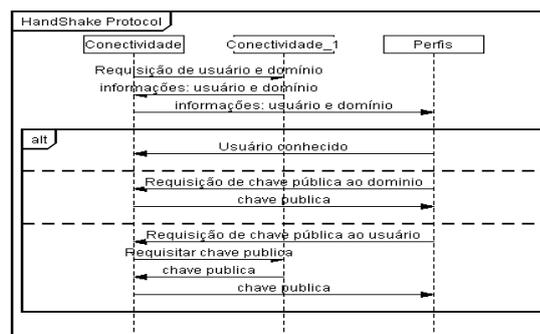


Figura 3. Protocolo de HandShake

3.2. Agente de Conectividade

A Figura 4 apresenta o agente de Conectividade contendo as capacidades:

- *HandShake*: o *HandShake* é o processo realizado no primeiro contato entre instâncias da GLOBAL. O protocolo de *HandShake* (Figura 3) suporta à comunicação nesse 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: Essa capacidade notifica os usuários conectados (usuários que podem receber mensagens) e registra agentes para receber as notificações;
- 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.

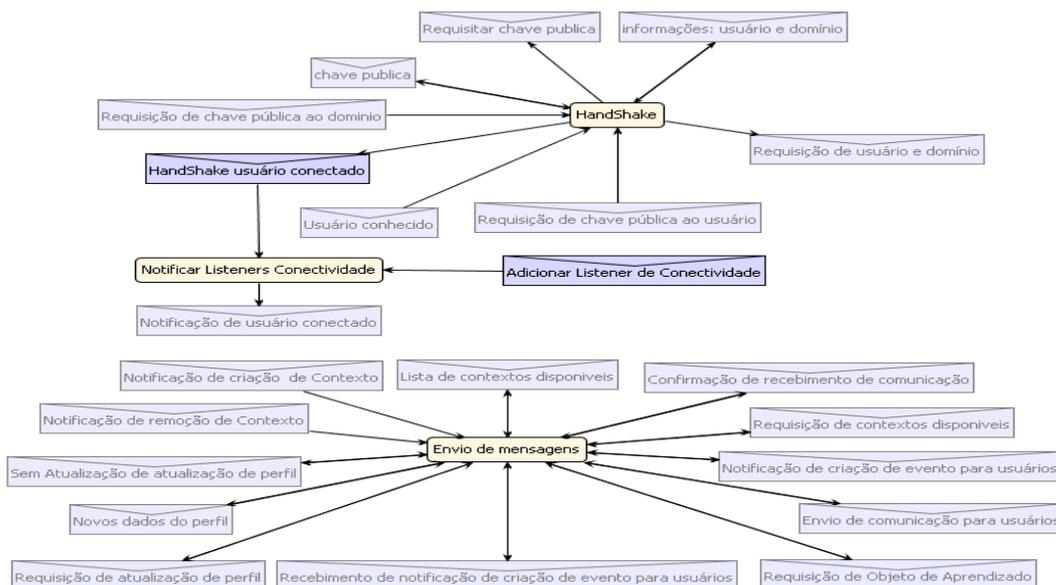


Figura 4. Agente de Conectividade

3.3. Agente de Contexto

A Figura 5 apresenta o agente de Contexto com as seguintes capacidades:

- 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 (Oliveira et al., 2012) 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.

3.4. Agente de Perfil do Usuário

A Figura 6 mostra o agente de Perfil do Usuário, sendo suas capacidades:

- 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 da GLOBAL;
- Responder *HandShake*: representa a responsabilidade do agente Perfil do Usuário no processo de *HandShake* descrito pelo protocolo de *HandShake*.

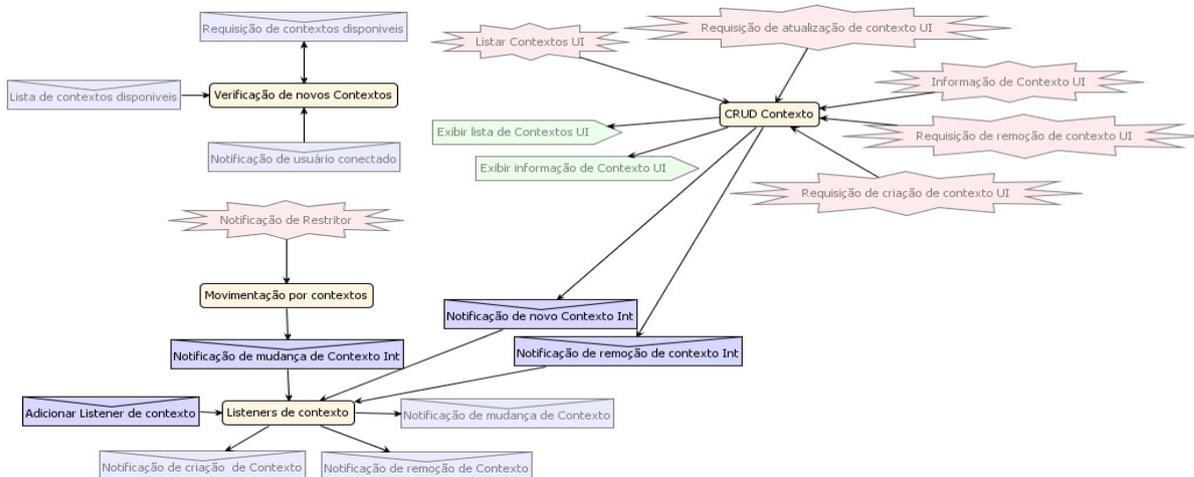


Figura 5. Agente de Contexto

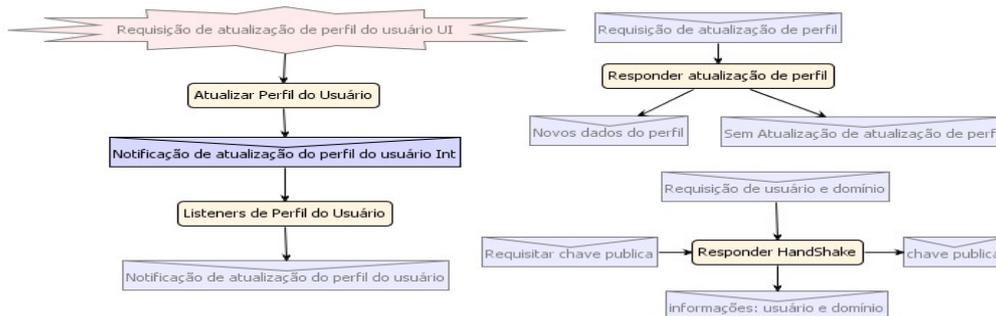


Figura 6. Agente de Perfil do Usuário

3.5. Agente de Perfis

A Figura 7 mostra o agente de Perfis, contendo as seguintes capacidades:

- *HandShake*: representa a parte de responsabilidade do agente de Perfis no processo de *HandShake*, descrito pelo protocolo de *HandShake* (Figura 3);
- Requisitar Perfil: capacidade de requisitar informações de perfis de outros usuários, caso tal informação esteja disponível, ele contata o agente Perfil do Usuário do outro usuário perguntado se existe atualização na informação. O protocolo de Verificação de atualização de perfil (Figura 8) descreve esse processo.

3.6. Agente de Objetos de Aprendizagem

A Figura 9 apresenta o agente de Objetos de Aprendizagem, sendo suas capacidades:

- Adicionar objeto de aprendizagem: quando um objeto de aprendizagem é adicionado na GLOBAL, é enviada uma notificação aos agentes registrados;
- Listeners de Objetos de Aprendizagem: registra pedidos de notificação de adição de objetos de aprendizagem;

- Adquirir Objeto de Aprendizagem: capacidade de adquirir objetos de aprendizagem presentes em outras instâncias da GLOBAL. O agente de Objetos de Aprendizagem envia uma mensagem ao agente de Conectividade, que usa uma rede *bittorrent* para aquisição do objeto.



Figura 7. Agente de Perfis

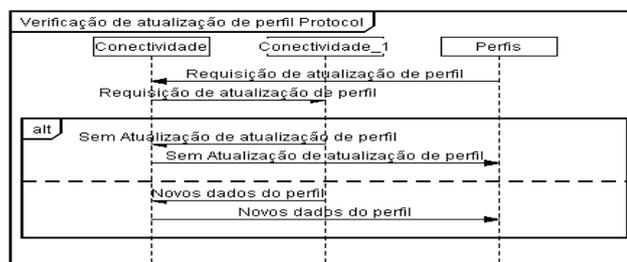


Figura 8. Protocolo de verificação de atualização de perfil

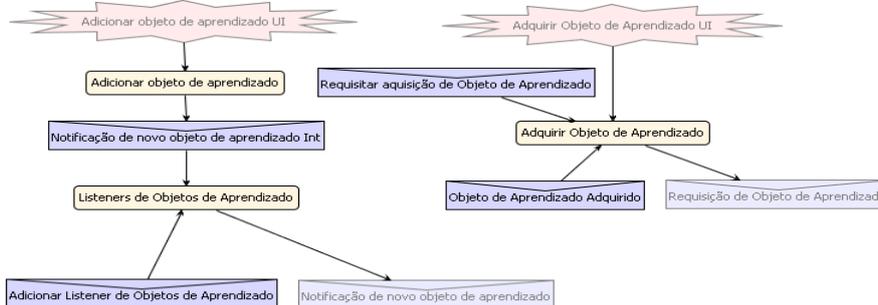


Figura 9. Agente de Objetos de Aprendizagem

3.7. Agente de Comunicação

A Figura 10 mostra o agente de Comunicação, com suas capacidades:

- Criar comunicação: cria uma comunicação e envia aos destinatários, que devem retornar a mensagem confirmado o recebimento;
- Enviar comunicação: envia a comunicação para os usuários que não estavam disponíveis quando a comunicação foi criada;
- Receber comunicação: capacidade de receber comunicação, notificando o usuário e confirmando o recebimento para o remetente;
- Listar comunicações: capacidade de listar, através da interface com o usuário, as comunicações recebidas.

4. Protótipo da Global e sua integração ao CoolEdu

O protótipo da GLOBAL foi desenvolvido em J2ME, MIDP 2.0 e CLDC 1.1, podendo ser usado em qualquer *smartphone* que suporte essa configuração. A Figura 11 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 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 pela GLOBAL.

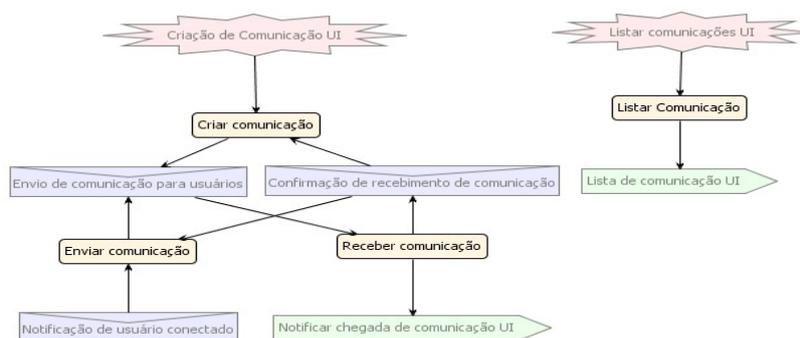


Figura 10. Agente de Comunicação

A *GlobalAgent* implementa a estrutura de subscrição (*subscription*) definida na FIPA-ACL. Quando um agente invocar o método *subscribe* requisitando notificações de determinado tipo, a *GlobalAgent* envia uma mensagem FIPA-ACL do tipo *SUBSCRIBE* para o agente indicado. Caso esse agente também seja uma extensão da *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*.

O CoolEdu (Rabello et al., 2012) é um sistema para colaboração em ambientes descentralizados de aprendizagem ubíqua. Ele apoia a aprendizagem através do estímulo à colaboração, utilizando serviços de suporte de um ambiente de aprendizagem ubíqua. Por sua vez, os agentes da GLOBAL podem interagir com qualquer sistema multiagentes que utilize o padrão FIPA-ACL, como é o caso do CoolEdu. Assim, buscando a criação de um sistema descentralizado de aprendizagem ubíqua e colaborativa, decidiu-se realizar uma integração entre os sistemas. Nesse sentido, foi usado o padrão de projeto *Adapter* (Gamma et al., 2014), onde classes *Adapter* foram usadas para ligar a interface *Agent* da GLOBAL, com os agentes do CoolEdu (Figura 12).

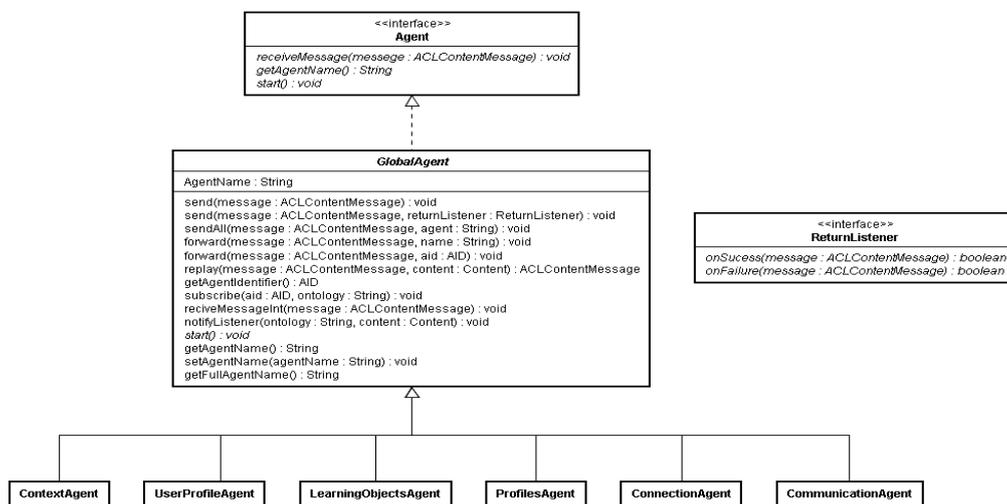


Figura 11. Estrutura dos agentes

Tanto o CoolEdu como a GLOBAL têm gerência de perfis. A GLOBAL usando os agentes Perfil do Usuário e Perfis e o CoolEdu usando um agente denominado Coletor (Rabello et al., 2012). Ambos os sistemas utilizam o modelo PAPI como perfil de usuário, mas o CoolEdu acrescenta campos relacionados ao estímulo à colaboração. Evitando uma gerência duplicada de perfis, a classe *Adapter* do agente Coletor (*AgentColetorAdapter*) assumiu o papel dos dois agentes da GLOBAL. Na inicialização da GLOBAL, o agente Coletor é registrado três vezes, como agente Coletor, como agente Perfil do Usuário e como agente Perfis. A classe *AgentColetorAdapter*, também realiza a tradução do formato do perfil do CoolEdu para o perfil PAPI especializado da GLOBAL, o qual passou a comportar as novas informações necessárias ao estímulo à colaboração pelo CoolEdu.

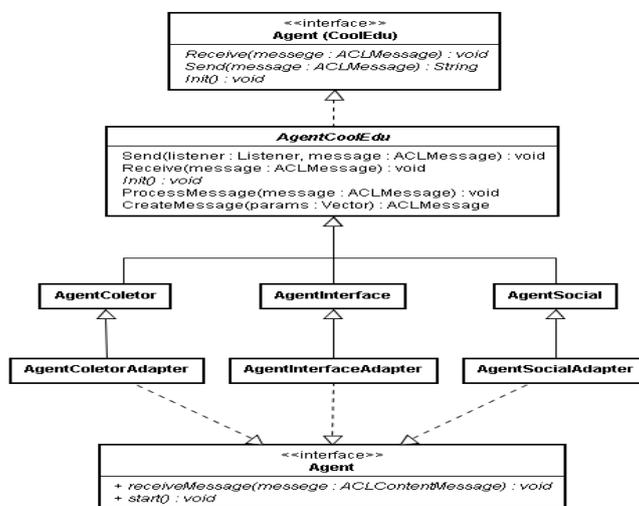


Figura 12. Adaptação dos agentes do CoolEdu

5. Conclusões

A contribuição primária desse artigo consistiu na apresentação de uma modelagem multiagentes para a GLOBAL. Nesse sentido foi usada a metodologia Prometheus. A contribuição secundária foi a discussão de características de implementação que envolveram aspectos de orientação a agentes, focando no protótipo da GLOBAL e na sua integração com um sistema de estímulo à colaboração em ambientes de aprendizagem ubíqua.

A principal constatação alcançada nesse trabalho é que a metodologia Prometheus mostrou-se uma ferramenta adequada para a modelagem da arquitetura de uma infraestrutura descentralizada para suporte à aprendizagem ubíqua. A representação suportada pela metodologia (Tabela 1) permitiu uma padronização durante a modelagem, suportando tanto a especificação quanto a documentação do sistema multiagentes. Destaca-se também a viabilidade do uso de aspectos de orientação a agentes na implementação da GLOBAL e na sua integração com o CoolEdu.

No futuro pretende-se aperfeiçoar a comunicação entre os agentes, pois apesar do uso de FIPA-ACL, o conteúdo das mensagens é baseado na serialização de objetos, não seguindo uma padronização. Uma proposta para melhoria da infraestrutura é utilizar um padrão de conteúdo para comunicação de agentes, como por exemplo, o FIPA RDF. Pretende-se ainda aprimorar o protótipo e sua integração com o CoolEdu, buscando um sistema de colaboração em ambientes de aprendizagem ubíqua que possa ser usado em cenários reais.

6. Referências

- BARBOSA, J.L.V.; HAHN, R.M.; BARBOSA, D.N.F.; SACCOL, A.I.C.Z. A ubiquitous learning model focused on learner interaction. *International Journal of Learning Technology*, v.6, n.1, p.62–83, 2011. DOI: 10.1504/IJLT.2011.040150.
- BELLAVISTA, P.; CORRADI, A.; FANELLI, M.; FOSCHINI, L. A survey of context data distribution for mobile ubiquitous systems. *ACM Computing Surveys*, v.44, n.4, 2012. DOI:10.1145/2333112.2333119.
- CACERES, R.; FRIDAY, A. Ubicomp Systems at 20: Progress, Opportunities, and Challenges. *IEEE Pervasive Computing*, v.11, n.1, p.14-21, 2013. DOI:10.1109/MPRV.2011.85.
- COSTA, C.A.; YAMIN, A.C.; GEYER, C.F.R. Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing*, v.7, n.1, p.64-73, 2008. DOI:10.1109/MPRV.2008.21.
- DIAZ, A.; MERINO, P.; Rivas, F.J. Mobile Application Profiling for Connected Mobile Devices. *IEEE Pervasive Computing*, v.9, n.1, p.54–61, 2010. DOI:10.1109/MPRV.2009.63.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading: Addison-Wesley, 1994. 416p.
- JENNINGS, N.R. An Agent-based approach for building complex software Systems. *Communications of the ACM*, v.44, n.4, p.35-41, 2001. DOI:10.1145/367211.367250
- JENNINGS, N.R.; WOOLDRIDGE, M. *Agent Technology: Foundations, Applications and Markets*. New York: Springer Verlag, 1998. 325p.
- OLIVEIRA, J.M.; RABELLO, S.A.; BARBOSA, J.L.V.; BARBOSA, D.N.F. Uma Infraestrutura Descentralizada para Ambientes de Aprendizagem Ubíqua. *Revista Brasileira de Informática na Educação*, v. 20, n.3 p.85-99, 2012. DOI: 10.5753/RBIE.2012.20.03.85.
- PADGHAM, L.; WINIKOFF, M. *Developing Intelligent Agent Systems: A Practical Guide*. New Jersey: Wiley, 2004. 240p.
- RABELLO, S.A.; OLIVEIRA, J.M.; WAGNER, A.; BARBOSA, J.L.V.; BARBOSA, D.N.F. CoolEdu: a Collaborative Multiagent Model for Decentralized Ubiquitous Education Environments. *IEEE Latin America Transactions*, v.10, n.6, p.2273-2279, 2012. DOI: 10.1109/TLA.2012.6418132.
- SATYANARAYANAN, M. Pervasive computing: vision and challenges. *IEEE Personal Communications*, v.8, n.4, p.10–17, 2001. DOI: 10.1109/98.943998.
- WAGNER, A.; BARBOSA, J.L.V.; BARBOSA, D.N.F. A Model for Profile Management Applied to Ubiquitous Learning Environments. *Expert Systems with Applications*, v.41, n.4, p.2023-2034, 2014. DOI:10.1016/j.eswa.2013.08.098.