

Um modelo baseado em *context-awareness* para disseminação de informações em um Ambiente de Desenvolvimento Distribuído de Software

Ana Paula Chaves*, Igor Scaliante Wiese,
César Alberto da Silva e Elisa Hatsue Moriya Huzita

Universidade Estadual de Maringá,
Av. Colombo, 5790, Maringá, Brasil
chavesana85@yahoo.com.br, igor.wiese@gmail.com,
{apchaves, cesar_as, emhuzita}@din.uem.br
<http://www.din.uem.br/disen>

Resumo Embora as técnicas de CSCW (*Computer Supported Cooperative Work*) e DDS (Desenvolvimento Distribuído de Software) tenham trazido diversas vantagens competitivas, ao seu lado surgiram diversas barreiras impostas pela distância física, como diferenças culturais ou de idiomas, falta de infra-estrutura de comunicação, entre outros. Entre essas barreiras está a dificuldade em disseminar as informações produzidas durante a realização das tarefas para as demais instâncias dos ambientes distribuídos, de forma que todos os indivíduos envolvidos as compreendam de maneira clara e sem ambigüidades. Esse artigo apresenta um modelo baseado em *context-awareness* (percepção de contexto) que se propõe a minimizar o problema da disseminação de informações em um ambiente de desenvolvimento distribuído de software conhecido como DiSEN. O objetivo do modelo é oferecer meios para que as informações contextuais existentes no ambiente possam ser capturadas, representadas, armazenadas e disseminadas para todos os participantes de uma equipe de desenvolvimento trabalhando cooperativamente.

Palavras-chave: Context-awareness, disseminação de informações, comunicação, Desenvolvimento Global de Software

1 Introdução

A necessidade cada vez maior de velocidade e precisão em atividades do desenvolvimento de software tem incentivado pessoas a trabalharem em conjunto de forma cooperativa. Entretanto, quando a realização de atividades envolve mais de uma pessoa, surgem dificuldades como a capacidade de cooperação e coordenação, exigindo novas ferramentas para atender a essa demanda [1].

O CSCW auxilia o trabalho cooperativo utilizando tecnologia computacional. Mas diversas necessidades em CSCW precisam ser satisfeitas para garantir

* Bolsista CAPES – Brasil. Programa de Pós-Graduação, Mestrado em Ciência da Computação – Universidade Estadual de Maringá

que todos os integrantes do grupo compreendam as atividades dos demais, recuperem informações anteriores e armazenem suas ações para serem utilizadas no futuro. Essas necessidades se evidenciam quando os participantes do trabalho cooperativo estão geograficamente distantes.

O desenvolvimento de software com equipes fisicamente distribuídas – DDS – surgiu como uma alternativa para se obter vantagens competitivas em termos de ganho de produtividade, melhoria de qualidade e redução de custos [2,3]. Entretanto, a distância física e os fatores dela derivados como diferenças culturais, de idioma, infra-estrutura de comunicação, entre diversos outros, dificultam a troca de informações e coordenação entre os membros da equipe.

Na tentativa de garantir a comunicação clara e não ambígua entre participantes do trabalho cooperativo, pesquisadores investiram em técnicas de percepção que, combinadas ao gerenciamento de contexto, favoreceram os estudos acerca de sistemas conhecidos como *context-awareness*. Esses sistemas são capazes de capturar, de alguma maneira, informações sobre o contexto das entidades¹ envolvidas no trabalho cooperativo, podendo disseminá-las para os demais indivíduos, relacioná-las a outras informações contextuais já existentes e armazená-las para futura recuperação, ou utilizá-las para executar um serviço automaticamente.

Este artigo propõe um modelo baseado em *context-awareness* para disseminação de informações em um Ambiente de Desenvolvimento Distribuído de Software (ADDS). Esse modelo será aplicado no ambiente DiSEN (*Distributed Software Engineering Environment*) [5], um ADDS que visa disponibilizar ferramentas de apoio à comunicação, à persistência e à cooperação para equipes de desenvolvimento geograficamente dispersas.

A Seção 2 apresenta os trabalhos que envolvem os conceitos de percepção e contexto em sistemas colaborativos. Com base nesses trabalhos, a Seção 3 descreve o modelo proposto. Por fim, a Seção 4 traz algumas considerações a respeito do tema proposto neste artigo e as futuras direções.

2 Trabalhos Relacionados

Diversas áreas de investigação científica, como Inteligência Artificial, Computação Ubíqua e Interface Humano-Computador, têm aplicado os conceitos de percepção e contexto nos respectivos domínios. Em CSCW, dentre as definições existentes, em [1] o contexto foi definido como uma complexa descrição do conhecimento compartilhado sobre circunstâncias físicas, sociais, históricas e outras, dentro das quais ações ou eventos ocorrem.

Devido à importância do conhecimento a respeito do contexto para apoiar as interações dos grupos de trabalho, um *framework* conceitual foi proposto por [1] para identificar e classificar as informações contextuais mais comuns em ferramentas de *groupware*. O *framework* divide as informações em cinco tipos: **Pessoas** (indivíduos e grupos), **Tarefas**, **Relação entre pessoas e tarefas**

¹ Entidade, segundo [4] representa pessoas, objetos, lugares ou outros softwares que interagem com a aplicação.

(interação e plano de execução), **Ambiente** e **Tarefas realizadas** (histórico sobre as tarefas completadas).

Entretanto, esse *framework* não define como essas informações serão apresentadas aos indivíduos do grupo. Para que haja compreensão acerca das informações contextuais, são necessários mecanismos capazes de oferecer percepção às pessoas envolvidas no trabalho cooperativo a fim de diminuir as sensações de impessoalidade e distância, características dos ambientes virtuais [7].

A fim de classificar os requisitos necessários para oferecer apoio adequado à percepção em sistemas de *groupware*, em [8] foi proposto um *framework* que resume esses requisitos em seis questões básicas: “o que” deve ser informado aos usuários, “quando” os eventos ocorrem ou devem ser informados, “onde” as informações são geradas ou apresentadas, “como” apresentá-las, “quem” está trabalhando e atento no momento e “quanto” de cada informação deve ser oferecida para que o usuário tenha a percepção ideal do evento de interesse.

Observa-se que a proposta apresentada em [1] se concentra em identificar o contexto, enquanto o *framework* proposto por [8] se preocupa com a questão da percepção. No entanto, oferecer percepção a uma atividade do trabalho cooperativo depende da relação existente entre percepção e contexto. Conhecer as contribuições dos demais membros de um grupo envolve não apenas o conhecimento sobre o objeto de colaboração, mas também da forma como este foi produzido. Dessa forma, é importante que os participantes da interação percebam o contexto ou circunstâncias que circundam a geração desses objetos compartilhados.

Para uma aplicação estar ciente das informações de contexto, é necessário seguir alguns passos para transformar seus dados em informações úteis. De maneira geral, a preocupação está em definir de que maneira o contexto será adquirido (onde encontrar, como captar e como interpretar as informações), gerenciado, representado e disseminado para as entidades [9,10].

Em [9], é proposto um modelo para gerenciar a criação, o armazenamento e a reutilização do conhecimento contextual. O contexto é relevante para o gerenciamento de conhecimento por oferecer mais significado às atividades, fatos, artefatos e decisões. O modelo se baseia em três aspectos: representação de informações contextuais, captura e armazenamento, e recuperação e apresentação dos contextos existentes, utilizando mecanismos de raciocínio e percepção.

O trabalho apresentado por [9] concentra-se na construção de uma memória organizacional, permitindo que, quando necessário, as pessoas envolvidas nas atividades possam recuperar informações pelos mecanismos de percepção. Entretanto, o modelo não oferece meios pelos quais essas pessoas possam ser comunicadas a respeito de mudanças ocorridas nesta base de informações.

Baseado nos conceitos e trabalhos descritos nesta seção e nos problemas existentes no DDS a respeito de comunicação e coordenação de atividades, a Seção 3 apresenta o modelo baseado em *context-awareness* para disseminação de informações em ADDS, proposto neste artigo.

nar e reutilizar os conhecimentos, outros fatores são necessários, como facilidade de contato, comunicação, compreensão mútua e compartilhamento.

Quando se trata de DDS, as equipes de trabalho podem possuir idiomas, vocabulários ou culturas regionais diferentes. Assim, cada equipe pode interpretar de formas distintas as informações que recebem, o que pode levá-las ao erro, seja pela falta de dados relevantes ou por possíveis ambigüidades. Além disso, não se pode garantir que os indivíduos obterão as informações necessárias pela comunicação direta com as outras pessoas, ainda que haja uma infra-estrutura de comunicação adequada como chats, emails ou outros métodos. Isso porque, em ambientes distribuídos, os indivíduos nem sempre se conhecem pessoalmente, o que pode ocasionar falta de confiança ou ausência da pré-disposição em ajudar os demais. Em ambos os casos, os custos acarretados variam desde o refatoramento de pequenas partes até à inviabilidade de um projeto. Dessa forma, se as informações disseminadas forem contextualizadas e possuírem uma representação única, reconhecida pelos participantes, então os indivíduos terão um entendimento comum, claro e não ambíguo das informações geradas pelas interações.

Contextualizar as atividades no DiSEN não envolve apenas saber em quais tarefas de um projeto os indivíduos estão trabalhando e como são realizadas, mas também conhecer as ferramentas e serviços que o ambiente deve oferecer.

Conforme definido em [14], o Gerenciador de Objetos do DiSEN possui um Gerenciador de Recursos. Além dos recursos materiais como equipamentos, mídias, entre outros, consideram-se recursos os Locais, Usuários e Ferramentas (Figura 2), que se constituem nas três entidades necessárias e capazes de gerar informações contextuais para o ambiente. Os usuários (pessoa ou grupo), são as entidades mais ativas do ambiente que, por meio de ferramentas e serviços, alteram o estado do ambiente como um todo e geram informações para ele.

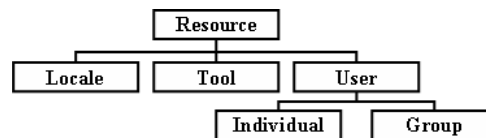


Figura 2. Especialização dos recursos no ambiente DiSEN

Em [16], o conhecimento dos indivíduos é classificado em três partes (Figura 3): **conhecimento externo** (conhecimento que o indivíduo possui, mas não utiliza para realizar determinada tarefa), **conhecimento contextual**, (conhecimento que influencia o desenvolvimento da tarefa a ser realizada), e **contexto procedural**, (conhecimento compilado para ser utilizado na solução do problema). O contexto procedural é construído com base na junção e organização de elementos ou partes do conhecimento contextual disponível, de acordo com um foco de atenção (evento ou ação a ser realizada). Cada vez que o foco muda, os elementos que formavam o contexto procedural voltam ao sub-conjunto dos conhecimentos contextuais e um novo contexto procedural é formado.

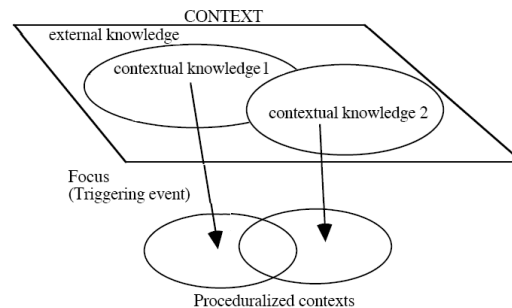


Figura 3. Tipos de contexto [16]

O conhecimento contextual de membros de equipes dispersas pode variar, devido à sua cultura e seus costumes locais. Como o contexto procedural é construído, inicialmente, a partir de um subconjunto do conhecimento contextual, para realizar uma tarefa em grupo é preciso que os envolvidos possuam um **contexto procedural compartilhado** [15], para formar seu conhecimento contextual com base nesses novos conhecimentos adquiridos. Quando os indivíduos não possuem em seu conhecimento contextual todas as informações de que necessitam, podem buscá-las a partir do conhecimento externo, adquirido pela comunicação com outros indivíduos ou em dados disponíveis na organização.

Assim, para compartilhar informações contextuais em DDS, é necessário transformá-las de forma que os participantes do trabalho as compreendam e as utilizem para realizar suas tarefas, independente de sua localização geográfica.

De maneira geral, os trabalhos encontrados na literatura [10,18] destacam quatro elementos essenciais a *context-awareness*: captura, representação, armazenamento e disseminação das informações contextuais. Seguindo essa linha, o modelo proposto para o DiSEN visa integrar esses elementos à característica de Percepção do modelo SPC, para permitir o compartilhamento de informações contextuais entre os diversos *workspaces* envolvidos no trabalho cooperativo.

Para alcançar esse objetivo, cada vez que uma informação contextual for gerada por uma entidade do Gerenciador de Objetos (Local, Usuário ou Ferramenta) em um *workspace* do ambiente, esta deverá ser capturada pelo Gerenciador de Workspace, como pode ser visto na Figura 4. Nesse gerenciador, o modelo a transforma em uma informação que pode ser compreendida por todos os membros das equipes dispersas e as compartilha com os demais *workspaces* existentes, incluindo o próprio *workspace* que gerou a informação.

A Figura 5 detalha o modelo baseado em *context-awareness* proposto. Como pode ser observado, o modelo possui quatro elementos essenciais: *Suporte à captura*, *Representação do contexto*, *Suporte ao processamento* e *Mecanismos de percepção*, além de um repositório para armazenamento das informações contextuais. Cada um desses elementos está descrito a seguir:

a. Suporte à captura: responsável por reconhecer as mudanças que ocorrem no contexto do trabalho. Essas mudanças poderão ser capturadas por agen-

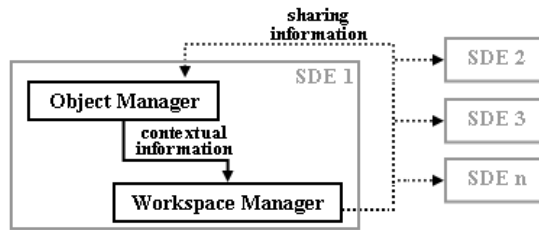


Figura 4. Compartilhando informações no DiSEN

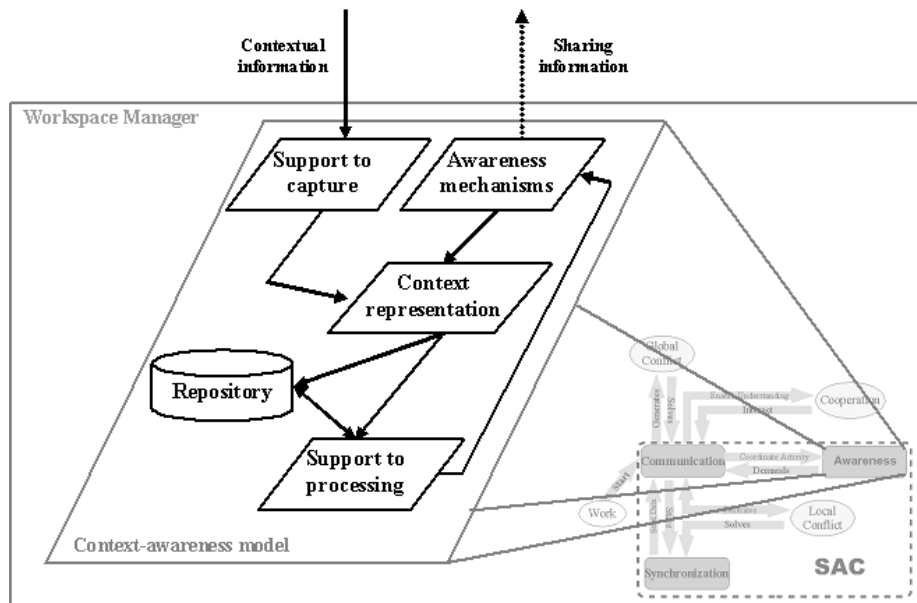


Figura 5. Modelo baseado em *context-awareness* para o DiSEN

tes humanos ou de software. Um agente humano corresponde a qualquer usuário capaz de fornecer as informações referentes ao seu contexto atual. Um agente de software pode ser qualquer entidade autônoma, capaz de reconhecer informações de contexto sem a interferência de um sistema ou de outra entidade, podendo ser pró-ativos ou despertados por evento. Quanto à origem das informações, o *Supporte à captura* possui três tipos de fontes de informação: **manuais**, quando os agentes humanos registram seu próprio contexto a partir de uma determinada interface; **locais**, quando estão armazenadas em arquivos de configuração, interfaces gráficas ou derivados de outros sensores locais (podendo ser, por exemplo, um dispositivo físico); e **repositórios**, quando estão armazenadas em algum repositório de dados comum a todos os *workspaces*, como um servidor de banco de dados central. É importante notar que ambas as fontes manuais e locais podem ser recuperadas a partir de interfaces gráficas do *workspace*. A diferença está

na voluntariedade. Quando as fontes de contexto são locais, a captura deve ser transparente, o que significa que o usuário não terá ciência de que suas ações estarão oferecendo informações contextuais. Nas fontes manuais, o contexto é solicitado ao usuário que informa seu contexto voluntariamente. As informações de fontes locais e repositórios devem ser capturadas por agentes de software, enquanto as fontes manuais sempre são alimentadas por agentes humanos.

b. Representação do contexto: responsável por receber as informações contextuais capturadas e representá-las formalmente. A importância da representação formal do contexto é destacada em diversos trabalhos encontrados na literatura [9,10,17], que apontam para a necessidade de que as diferentes entidades que participam das interações (pessoas, software, dispositivos) possuam a mesma compreensão semântica das informações de contexto. Dessa forma, o objetivo do elemento *Representação do contexto* é mapear as informações provenientes do *Suporte à captura* para o modelo de representação formal e relacioná-las com as demais informações contextuais já existentes no repositório. Existem na literatura diversas propostas de modelos de representação [9,10,17], entre eles pares valor-chave, modelos baseados em marcação, baseados em lógica, baseados em representação gráfica, entre outros, entretanto, observa-se uma convergência para modelos baseados em ontologias. Seguindo essa tendência, esse modelo deverá ser adotado para o ambiente DiSEN, devido ao seu alto nível de formalismo e possibilidade de inferência sobre as informações representadas. Além disso, o uso de ontologias favorece o reuso dos conceitos modelados, aumentando a interoperabilidade entre aplicações que poderão ser desenvolvidas no futuro, por exemplo, uma ferramenta de gerenciamento de memória organizacional.

c. Armazenamento: para o modelo proposto, as informações contextuais capturadas e representadas poderão ser persistentes ou transientes. As informações transientes são importantes para um determinado momento, mas não serão utilizadas no futuro, podendo ser disparadas para os mecanismos de percepção, sem a necessidade de armazenamento. Ao contrário, as informações persistentes são aquelas que deverão permanecer em um repositório de dados para consultas futuras, formando um histórico de informações de contexto que, posteriormente, poderá ser utilizado para a construção de uma memória organizacional. Quando necessário, esse repositório, representado pelo Armazenamento, poderá ser acessado pelo *Suporte ao processamento* para a recuperação dos registros nele contidos.

d. Suporte ao processamento: mecanismo de raciocínio capaz de deduzir contextos implícitos nas informações recebidas e corrigir possíveis inconsistências, com base nos relacionamentos existentes entre os conjuntos de informação, criados na *Representação do contexto*. As informações persistentes são adquiridas a partir do Armazenamento, enquanto as informações transientes são recebidas diretamente da *Representação do contexto*. Entretanto, todas as informações precisam ser processadas antes de serem enviadas para os *Mecanismos de percepção*, a fim de possibilitar que as informações disseminadas sejam interpretadas e possuam, assim, maior relevância.

e. Mecanismos de percepção: depois de serem representadas e processadas, as informações contextuais estarão prontas para serem disponibilizadas para os *Mecanismos de percepção*. Esses mecanismos serão responsáveis por identificar, automaticamente, quais *workspaces* estão interessados na informação, quais métodos serão utilizados para exibi-las (um chat, um email, uma ferramenta de geração de artefatos, um pop-up no *workspace* do usuário, entre outros métodos definidos em [11]) e, por fim, se encarregarão de disseminá-las para as instâncias do ambiente, independente de sua localização geográfica, por meio da infra-estrutura de comunicação disponível no ambiente DiSEN. Assim, essas informações poderão ser compartilhadas pelos membros das equipes dispersos pelos diversos *workspaces*.

O modelo apresentado na Figura 5 resume como informações contextuais podem ser obtidas, representadas e disseminadas para todos os membros de equipes que trabalham cooperativamente no ambiente DiSEN. A Seção 4 discute algumas considerações a respeito do modelo proposto e aponta as futuras direções.

4 Considerações Finais

A utilização das técnicas de CSCW e DDS, embora tenha trazido diversas vantagens em termos de produtividade e qualidade nos produtos de software, trouxe também novos desafios. Os membros de equipes geograficamente dispersas muitas vezes nem mesmo se conhecem pessoalmente, o que pode gerar uma falta de confiança ou mesmo não existir pré-disposição para troca de informações. Ainda, os diferentes idiomas ou as diferenças culturais e organizacionais decorrentes da distância local entre as empresas podem comprometer a comunicação entre as pessoas envolvidas em um projeto. Vencer as barreiras impostas pela distribuição física depende, em grande parte, de oferecer meios para que os indivíduos das equipes possam se comunicar de forma eficiente e segura.

Diante dos desafios, as principais contribuições do modelo baseado em *context-awareness* apresentado neste artigo são: (i) prover um mecanismo capaz de capturar as informações contextuais, incluindo circunstâncias físicas sociais ou históricas. Para tanto, além dos agentes humanos, está previsto também utilizar agentes de software que apresentam características que apoiam o desenvolvimento distribuído de software no que tange a autonomia e a escalabilidade; (ii) representar adequadamente essas informações contextuais por meio de uma ontologia, que se encontra em definição, possibilitando que, futuramente, essas informações representadas possam constituir uma memória organizacional; (iii) utilizar mecanismos adequados de percepção para disseminar automaticamente essas informações contextuais a todos os participantes dos diferentes projetos, independentemente de sua localização física. Com isso, as pessoas não precisam, necessariamente, consultar o repositório para estar ciente das modificações, elas serão automaticamente notificadas. Dessa forma, as informações geradas serão rápida e amplamente disseminadas e utilizadas para tomada de decisões, adequadamente, de acordo com o referido contexto.

Para dar continuidade a esta pesquisa, os próximos passos serão especificar cada um desses elementos do modelo proposto, implementar o modelo com base na infra-estrutura do ambiente DiSEN e propor mecanismos de apoio à gestão de conhecimento em ADDS.

Referências

1. Rosa, M.G.P.: Inserindo contexto em groupware. Master's Thesis. Instituto de Matemática e Núcleo de Computação Eletrônica. UFRJ (2004)
2. Herbsleb J.D., Moitra D.: Guest Editor's Introduction: Global Software Development. In: *IEEE Software*, vol. 18, no. 2, pp. 16–20, March/April (2001)
3. Prikladnicki R., Audy J.L.N.: Munddos: Um modelo de referência para desenvolvimento distribuído de software. XVIII Simpósio Brasileiro de Engenharia de Software (2004)
4. Dey A.K., Abowd G.D.: Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22. Atlanta (1999)
5. E.H.M. Huzita, et al.: Um Ambiente de Desenvolvimento Distribuído de Software – DiSEN. In: *Workshop de Desenvolvimento Distribuído de Software*. pp 31–38. João Pessoa (2007)
6. Dourish P., Bellotti V.: Awareness and coordination in shared workspaces. In: *ACM Conference on Computer-Supported Cooperative Work*. pp. 107-114. (1992)
7. Fuks H., Raposo A.B., Gerosa M.A.: Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas. XXI Jornada de Atualização em Informática. In: XXII Congresso da Sociedade Brasileira de Computação, vol. 2, cap. 3, pp. 89–128 (2002)
8. Pinheiro M.K., Lima J.V., Borges M.R.S.: Awareness em sistemas de groupware. *International Database Engineering and Applications Symposium*. Costa Rica (2001)
9. Nunes V.T., Santoro F.M., Borges M.R.S.: Um modelo para gestão de conhecimento baseado em contexto. Simpósio Brasileiro de Sistemas Colaborativos. In: XXVII Congresso da SBC, pp. 1841-1854. Rio de Janeiro (2007)
10. Santos V.V.: Gerenciamento de Contexto em Sistemas Colaborativos. PhD thesis – Monografia de Qualificação. Universidade Federal de Pernambuco, Recife (2006)
11. Pozza R.S.: Proposta de um modelo para cooperação baseado no gerenciador de workspace no ambiente DiSEN. Master's thesis, Programa de Pós-Graduação em Ciência da Computação. Universidade Estadual de Maringá (2005)
12. Fuks H., Raposo A.B., Gerosa M.A.: Do Modelo de Colaboração 3C à Engenharia de Groupware. In: *Simpósio Brasileiro de Sistemas Multimídia e Web* (2003)
13. Pascutti, M. C. D.: Uma proposta de arquitetura de um ambiente de desenvolvimento de software distribuído baseado em agentes. Mestrado em Computação. Universidade Federal do Rio Grande do Sul, Porto Alegre (2002)
14. Schiavoni F.L.: FRADE – Framework para infra-estrutura de um ambiente de desenvolvimento distribuído de software. Mestrado em Ciência da Computação, UEM. Maringá (2007)
15. Araújo R.M., Brézillon P.: Reinforcing Shared Context to Improve Collaboration. *Revue d'Intelligence Artificielle*, vol.19, pp. 537–556 (2005)
16. Brézillon P., Pomerol J.-C., Contextual Knowledge Sharing and Cooperation in Intelligent Assistant Systems. *Le Travail Humain*, vol. 62, no. 3, pp. 223–246 (1999)
17. Baldauf M., Dustdar S., Rosenberg F.: A survey on context-aware systems. *Int. J. Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp.263–277 (2007)
18. Dey, A. K.: Providing Architectural Support for Building Context-Aware Applications. Doctor of Philosophy in Computer Science. Georgia, EUA (2000)