

Um Estudo de caso com o Protótipo de Estimação de Localização baseado em Sistema Multiagente para a melhoria de segurança

Ana Régia de M. Neves¹, Letícia T. Maia¹, Célia G. Ralha², Ricardo P. Jacobi²

¹Departamento de Engenharia Elétrica – Faculdade de Tecnologia – Universidade de Brasília (UnB)

Caixa Postal 4521 – 70904-970 – Brasília – DF – Brasil

²Instituto de Ciências Exatas – Departamento de Ciências da Computação – Universidade de Brasília (UnB)

Caixa Postal 4466 – 70910-900 – Brasília – DF – Brasil

{regianeves, letmaia}@unb.br, {ghedini, rjacobi}@cic.unb.br

***Abstract.** Among the variables used to the development of context-aware applications, identification and location are known as primary context because they help the system to provide relevant information to the user. In this article, a prototype based on multiagent system for user identification and location at an indoor environment is presented. Furthermore, a simulation was performed in order to validate the developed prototype.*

***Resumo.** Dentre as variáveis utilizadas para o desenvolvimento de aplicações sensíveis ao contexto a identificação e a localização são conhecidas como contexto primário, pois auxiliam o sistema a prover informações relevantes para o usuário. Neste artigo é apresentado um protótipo para identificação e localização de usuários em um ambiente fechado baseado em sistema multiagente. Além disso, foi feita uma simulação a fim de validar o protótipo desenvolvido.*

1. Introdução

As aplicações sensíveis ao contexto utilizam informações que caracterizam a situação corrente do usuário para fornecer serviços adaptados na realização de suas tarefas [Dey 2001]. Dentre estas, a identificação e a localização são importantes para entender o contexto do usuário móvel.

Este artigo apresenta uma simulação baseada em sistema multiagente para identificar e localizar os usuários dos laboratórios do SG-11 da Universidade de Brasília. Para isso, foi utilizada a plataforma *Java Agent DEvelopment Framework (JADE)* que é um software livre distribuído sobre a licença LGPL e implementado em Java. O principal objetivo desta ferramenta é simplificar e facilitar o desenvolvimento de agentes provendo a comunicação, recursos para gerenciar o ciclo de vida e monitorar os agentes em execução [JADE 2003].

O artigo está organizado em cinco seções, desconsiderando-se a seção de introdução. A seção 2 aborda aspectos teóricos sobre a computação sensível ao contexto. A seção 3 apresenta uma breve definição de sistema multiagente. Na seção 4 são descritos os trabalhos correlatos. A seção 5 exibe a implementação da proposta e na seção 6 são apresentadas as conclusões e trabalhos futuros.

2. Computação Sensível ao Contexto

Refere-se a aplicações que são capazes de monitorar e utilizar dinamicamente informações que provêm do ambiente ou do usuário [Coutaz *et al* 2005].

De acordo com Dey [2001], contexto é toda informação que pode ser usada para caracterizar a situação de uma entidade (pessoa, lugar ou objeto) que é considerada relevante para a interação entre um usuário e uma aplicação.

Os atributos que definem um contexto podem ser divididos em dois grupos e são dependentes do domínio da aplicação [Castro e Favela 2005]: fatores humanos (hábitos, estado emocional, ambiente social e tarefas realizadas) e propriedades do ambiente físico (localização, infraestrutura e condições físicas).

O processo de desenvolvimento de uma aplicação sensível ao contexto envolve basicamente três etapas [Dey 2001]: i) o contexto deve ser capturado como um conjunto de dados de baixo nível a partir de diferentes sensores; ii) esse conjunto deve ser interpretado para construir uma informação contextual de alto nível mais significativa para aplicação, por exemplo, ele pode mapear coordenadas GPS (*Global Positioning Systems*) para um endereço de rua; e iii) deve levar a informação interpretada para a aplicação que é utilizada com outros dados para oferecer e adaptar a computação e serviços.

3. Sistema Multiagente

Um Sistema Multiagente (SMA) é um sistema computacional em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos [Lesser e Victor 1999].

A área de SMA estuda o comportamento de um grupo organizado de agentes autônomos que cooperam na resolução de problemas que estão além das capacidades de resolução de cada um individualmente [Briot e Demazeau 2001].

Segundo Wooldridge [2002], para um agente ser considerado inteligente, ele deve ser autônomo, apresentar capacidade de comunicação, de cooperação, de raciocínio, deve ter mecanismos de planejamento para executar suas ações e ser adaptável quanto a mudanças ocorridas no ambiente.

O ambiente onde os agentes estão situados é uma parte essencial do sistema [Weyns e Holvoet 2004], por isso, definir suas propriedades é uma decisão de projeto que depende dos requisitos do domínio do problema. Os agentes precisam agir e perceber esse ambiente. Essa percepção pode ser limitada pela capacidade do próprio agente e pelas propriedades do ambiente.

Essa abordagem de SMA pode reduzir o custo e a complexidade associados ao desenvolvimento de sistemas distribuídos, além de permitir uma maior flexibilidade no

processo de descoberta de soluções de problemas onde essas não podem ser previamente estabelecidas [Jennings *et al* 1998].

Conforme Bellifemine *et al* [2007] e Juchem e Bastos [2001], os principais desafios para desenvolver um sistema multiagente são: a) comunicação, definir como ela será realizada entre os agentes e que tipo de protocolo utilizar; b) interação, especificar como essa interação ocorrerá e que linguagens os agentes devem usar para interagirem entre si e combinarem seus esforços; c) coordenação, como garantir essa coordenação entre os agentes para que haja uma coerência na solução do problema ou objetivo que estão tentando alcançar.

4. Trabalhos Correlatos

Em Sachima *et al* [2004] foi desenvolvido um *middleware* para localização baseado em agentes. Foram definidos quatro tipos de agentes: i) agente de serviço (*service provider*) que atua como *proxy* e informa sobre as capacidades e restrições da localização física para fornecer o serviço; ii) agente pessoal (*service requester*) que solicita a execução de algum serviço em nome do usuário; iii) agente de localização (*location-aware middle agent*) que compara os agentes de serviço e os agentes pessoais através da localização, capacidade e solicitação para iniciar a comunicação; e iv) agente de dispositivo (*device wrapper agent*) que atua como *proxy* de um dispositivo e fornece informações derivadas de sensores para o agente de localização. Toda a comunicação entre os agentes é baseada na linguagem ACL (*Agent Communication Language*) proposta pela *The Foundation of Intelligent Physical Agents – FIPA*.

Em Nguyen *et al* [2008] foi determinada uma comunidade virtual baseada em sistema multiagente para apoiar a descoberta de serviços e localização de usuários em ambientes pervasivos. Foram definidos cinco tipos de agentes: i) agente usuário que solicita sua posição corrente; ii) agente de localização que coleta a posição do usuário por GPS, sensores ou por intensidade de sinal; iii) agente da comunidade que permite ou não a entrada de um agente usuário; iv) agente de serviços que disponibiliza os serviços referentes à comunidade; e v) agente de contexto que armazena dados personalizados sobre o usuário. Para construção do protótipo foi utilizada a plataforma JADE.

Em Neves *et al* [2010] foi desenvolvido um protótipo baseado em sistema multiagente para estimar a localização de usuários em ambientes fechados. Foram definidos três agentes reativos: i) agente cliente que solicita a sua posição no ambiente; ii) agente mapeamento que procura quais agentes sensores estão disponíveis no ambiente para encontrar a localização do usuário; e iii) agente sensor que procura e encontra o usuário no ambiente. Para a construção do protótipo foi utilizada a plataforma JADE.

5. A Arquitetura Multiagente e o Protótipo do Sistema de Identificação

Para identificar e estimar a localização dos usuários dos laboratórios do SG-11 da Universidade de Brasília foi proposta uma arquitetura baseada em sistema multiagentes.

Foram definidos três agentes: cliente, ambiente e sensor. Os agentes cliente e sensor são reativos; e o agente ambiente é reativo com regra de estados.

A Tabela 1 exibe as percepções, ações e o ambiente para cada agente definido neste trabalho.

Tabela 1. Classificação dos Agentes

Agente	Percepção	Ação	Ambiente
Cliente	<ul style="list-style-type: none"> - Receber informação de localização proveniente do agente ambiente - Receber a informação de localização de outro usuário proveniente do agente ambiente 	<ul style="list-style-type: none"> - Solicitar localização para o agente ambiente - Solicitar a localização de outro usuário para o agente ambiente 	<p style="text-align: center;">Acessível Determinístico Estático Não-episódico Discreto</p>
Ambiente	<ul style="list-style-type: none"> - Receber solicitação do agente cliente - Receber a localização do usuário proveniente do agente sensor - Armazenar posição do usuário no log 	<ul style="list-style-type: none"> - Solicitar a localização do usuário para o agente sensor - Informar ao agente cliente a localização - Informar ao agente cliente a localização de outro usuário 	
Sensor	<ul style="list-style-type: none"> - Receber a solicitação do agente ambiente 	<ul style="list-style-type: none"> - Localizar usuário - Informar ao agente ambiente a localização requisitada 	

O ambiente simulado é o prédio do departamento de engenharia elétrica da Universidade de Brasília que contém vários laboratórios, o SG11. Todos os docentes, os discentes e os funcionários desse prédio ao possuírem vínculo com a instituição possuem um número de matrícula que servirá para a identificação no ambiente em estudo. Isso é feito para controlar e restringir o acesso a esses laboratórios. Para simulação, o ambiente foi reduzido a três salas (sala 1, sala 2 e sala 3), nas quais o usuário pode se movimentar livremente [Neves *et al* 2010]. A Figura 1 ilustra o ambiente utilizado.



Figura 1. Ambiente Interno.

A arquitetura proposta é exibida na Figura 2, onde o agente cliente está dividido em três categorias: i) agente cliente p composto por docentes e podendo variar de 1 a n docentes presentes no prédio; ii) agente cliente a composto por discentes e podendo variar de 1 a x discentes presentes; e iii) agente cliente f composto por funcionários e podendo variar de 1 a z funcionários presentes.

Os agentes clientes podem solicitar a sua própria localização ou a localização de outro agente cliente ao agente ambiente e aguardam a resposta. O agente ambiente serve como interlocutor entre os agentes clientes e os agentes sensores; e inicia a busca de quais agentes sensores estão disponíveis para localizar os agentes clientes. Cada sala do ambiente contém um agente sensor. Os agentes sensores procuram e localizam o usuário no prédio SG11, depois, repassam a resposta ao agente ambiente o qual informa aos agentes clientes as localizações solicitadas.

Quando é solicitada a localização de um usuário no prédio, o agente ambiente aguarda a resposta dos agentes sensores; caso não seja encontrado, o agente ambiente verifica no log se há algum registro (data, horário e local mais recente) do usuário solicitado. Apenas usuários cadastrados poderão solicitar a localização de outro.

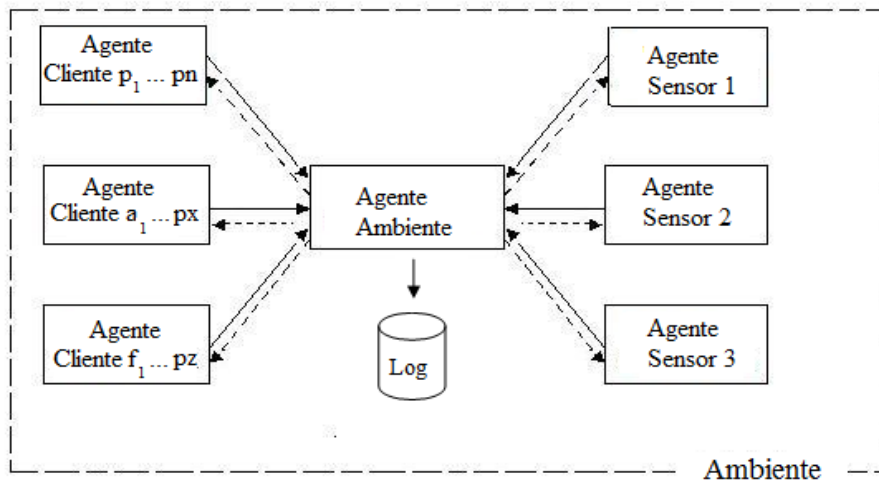


Figura 2. Arquitetura do Sistema de Identificação.

Além disso, foram implementados oito comportamentos que são responsáveis pelas principais ações dos agentes como recepção de mensagens e procura do usuário no ambiente. Esses comportamentos são estendidos da classe *behaviours* que o JADE oferece. A classe *TickerBehaviour* permite que esse comportamento seja executado em intervalos de tempo constantes. A classe *CyclicBehaviour* modela comportamentos atômicos que sempre são executados. A Tabela 2 apresenta a descrição dos comportamentos de cada agente.

Tabela 2. Comportamento dos Agentes

Agente	Comportamentos
Cliente	<p>Procura agente ambiente: solicita ao agente ambiente que informe a posição no SG11. É um comportamento que estende a classe <i>TickerBehaviour</i>.</p> <p>Aguarda retorno de localização: recebe a informação de localização enviada pelo agente ambiente. É um comportamento que estende a classe <i>CyclicBehaviour</i>.</p>
Ambiente	<p>Aguarda requisição de usuário: aguarda requisições do usuário. É um comportamento que estende a classe <i>CyclicBehaviour</i>.</p> <p>Procura agente sensor: procura os agentes sensores ativos e solicita que informe a posição do usuário. É um comportamento que estende a classe <i>TickerBehaviour</i>.</p> <p>Guarda no <i>log</i>: todos os usuários que entrarem no prédio terão um registro com hora, data e local que estiveram. É um comportamento que estende a classe <i>CyclicBehaviour</i>.</p> <p>Aguarda retorno do agente sensor: recebe a resposta do agente sensor com a posição do usuário. É um comportamento que estende a classe <i>CyclicBehaviour</i>.</p>
Sensor	<p>Aguarda requisição do agente ambiente: aguarda requisições de localização do agente ambiente e responde com a posição do usuário. É um comportamento que estende a classe <i>CyclicBehaviour</i>.</p> <p>Detecta movimento usuário: detecta a movimentação do usuário. É um comportamento que estende a classe <i>TickerBehaviour</i>.</p>

A Figura 3 mostra um cenário que envolve a interação dos agentes no ambiente, com os seguintes passos: (1) o agente cliente solicita sua localização ao entrar no ambiente ou a localização de um outro usuário; (2) o agente ambiente recebe a solicitação do agente cliente, o identifica com um ID de matrícula, procura quais agentes sensores estão disponíveis com o serviço de localização e repassa a solicitação do agente cliente com o ID de matrícula aos agentes sensores; (3) os agentes sensores recebem a mensagem do agente ambiente com o ID do usuário, procuram o usuário e informam a localização solicitada ao agente ambiente; (4) o agente ambiente guarda todas as informações de localização no log, registrando a hora, data e sala que o usuário está; (5) o agente ambiente envia a localização solicitada ao agente cliente.

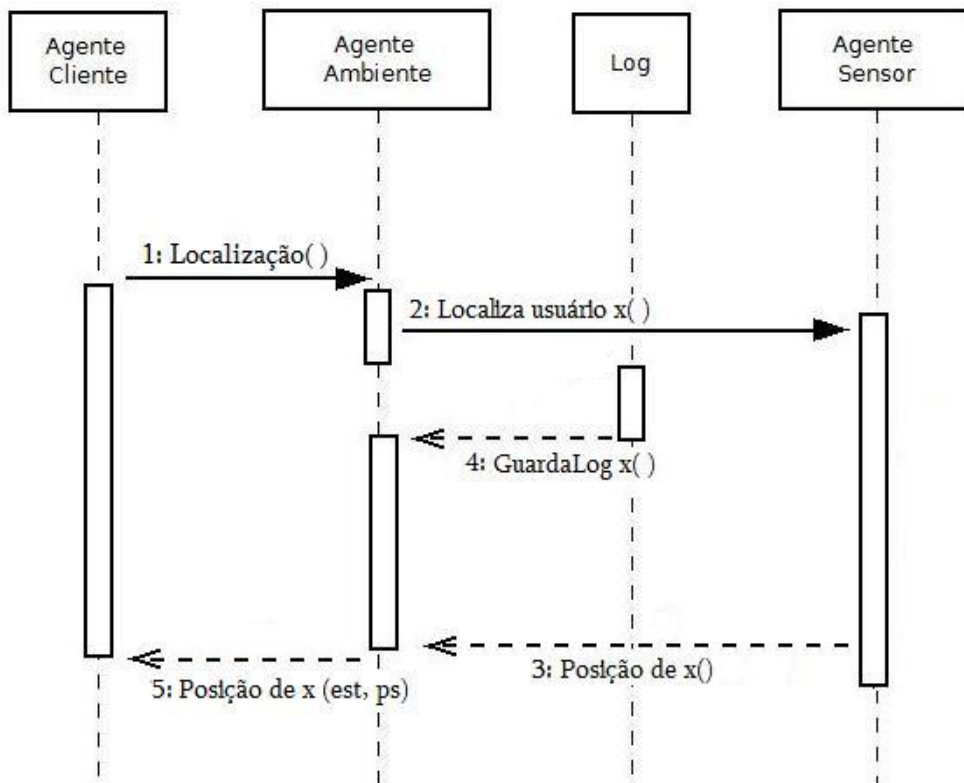
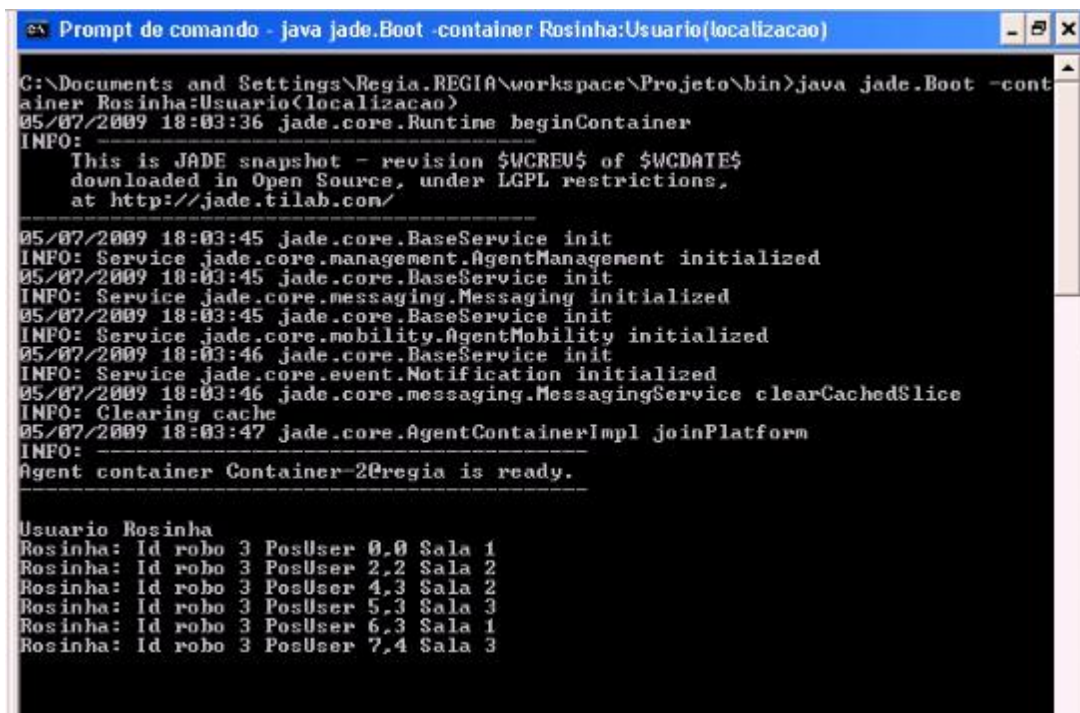


Figura 3. Diagrama de Sequência para a Localização do Usuário na simulação.

Assim, a partir do momento que o usuário for identificado com o ID, sua movimentação no ambiente será rastreada.

A Figura 4 apresenta um *screenshot* do console com um agente cliente chamado Rosinha. Esta usuária é identificada com uma ID (robo 3) a fim de ser monitorada no ambiente. As localizações (sala 1, sala 2, sala 3) da usuária são definidas em intervalos de tempos aleatórios no ambiente.



```
Prompt de comando - java jade.Boot -container Rosinha:Usuario(localizacao)
C:\Documents and Settings\Regia.REGIA\workspace\Projeto\bin>java jade.Boot -cont
ainer Rosinha:Usuario(localizacao)
05/07/2009 18:03:36 jade.core.Runtime beginContainer
INFO:
-----
This is JADE snapshot - revision $WCREU$ of $WCDATE$
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
05/07/2009 18:03:45 jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
05/07/2009 18:03:45 jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
05/07/2009 18:03:45 jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
05/07/2009 18:03:46 jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
05/07/2009 18:03:46 jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
05/07/2009 18:03:47 jade.core.AgentContainerImpl joinPlatform
INFO:
-----
Agent container Container-2@regia is ready.
-----

Usuario Rosinha
Rosinha: Id robo 3 PosUser 0,0 Sala 1
Rosinha: Id robo 3 PosUser 2,2 Sala 2
Rosinha: Id robo 3 PosUser 4,3 Sala 2
Rosinha: Id robo 3 PosUser 5,3 Sala 3
Rosinha: Id robo 3 PosUser 6,3 Sala 1
Rosinha: Id robo 3 PosUser 7,4 Sala 3
```

Figura 4. Screenshot do Console das Posições do Usuário.

6. Conclusões e Trabalhos Futuros

Como visto, estimar a localização de um usuário num ambiente é um dos aspectos mais importantes para sistemas sensíveis ao contexto, pois, o sistema poderá fornecer serviços de acordo com esta característica e conservar seguro esses serviços e as pessoas que os utiliza.

Baseado nisso, foi realizada uma simulação para identificação e localização de docentes, discentes e funcionários de três laboratórios do departamento de Engenharia Elétrica da universidade de Brasília utilizando sistema multiagente através da plataforma JADE.

Essa abordagem foi escolhida devido à abstração fornecida para a concepção de sistemas complexos, onde deve ocorrer a junção de diferentes domínios de conhecimento para a solução do problema, como, em ambientes sensíveis ao contexto. Além disso, se mostrou adequada ao desenvolvimento do protótipo proposto num ambiente estático, facilitando a definição de diversos comportamentos dos agentes.

Assim, planeja-se implementar uma arquitetura mais robusta e complexa no ambiente real, desenvolver agentes cognitivos para maior autonomia de decisão e proatividade dos agentes no sistema, além de utilizar outras técnicas de Inteligência Artificial, como, redes neurais artificiais, para detecção mais precisa da posição e uso de ontologias para contextualização de ambientes e serviços, conforme grupos de usuários e cenários distintos.

Uma das limitações da implementação de uma arquitetura multiagente no ambiente real refere-se ao acréscimo de novos agentes, pois, mesmo que aumente a funcionalidade do sistema, a coordenação dos diferentes tipos de agentes, cada um com certo tipo de autonomia, fica mais complexa.

Referências

- Abowd, G. D. e Mynatt, E. D. (2000), Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1): 29-58.
- Bellifemine, F. L.; Giovanni, C.; Greenwood, D. (2007), *Developing Multi-Agent Systems with JADE*. John Wiley & Sons, Ltd.
- Briot, J.-P. e Demazeau, Y., editors (2002). *Principes et architecture des systèmes multi-agents*. Hermes, Paris.
- Castro, L. A. e Favela, J. (2005) “Continuous Tracking of User Location in WLANs Using Recurrent Neural Networks”, In *Proc. Sixth Mexican International Conference on Computer Science. (ENC'05)*.
- Coutaz, J.; Crowley, J.; Dobson, S.; Garlan, D. (2005), Context is key, *Communications of the ACM*, Vol 48, No 3, pp. 49-53.
- Dey, A. K. (2001), *Understanding and Using Context*. Personal and Ubiquitous Computing Springer-Verlag, London, UK; Vol. 5, No 1, 201, pp. 4-7.
- Jade. JADE (2003) - *Java Agent Development Framework*. Disponível em <http://jade.tilab.com/>, Junho.
- Jennings, N. R. (2001) “An Agent-based Approach for Building Complex Software Systems”, *Communications of the ACM archive*, vol. 44, pp.35-41.
- Juchem, M. e Bastos, R. M. (2001), *Engenharia de Sistemas Multiagentes: Uma investigação sobre o estado da arte*. Technical Report Series, No 014.
- Lesser, Victor (1999) *Cooperative Multi-Agent Systems: A Personal View of the State of the Art*, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 11, No1,.
- Loke, S. (2007), *Context-Aware Pervasive Systems: architectures for a new breed of applications*. Auerbach Publications.
- Neves, A. R. M; Maia, L. T.; Ralha, C. G.; Jacobi, R. P. (2010) “Prototype for Indoor Localization Based on Multiagent System”, In *Third International Conference on Intelligent & Advanced System*.
- Nicholas R. Jennings; Sycara, K. e Wooldridge, M. (1998). A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38
- Nguyen, T; Loke, S.; Torabi T.; Lu, H. (2008) “Multiagent Place-Based Virtual Communities for Pervasive Computing”, In *Sixth Annual IEEE International Conference on Pervasive and Communications*.
- Orr, R. J e Abowd G. D. (2000) “The Smart Floor: A mechanism for natural user identification and tracking”, In *Proc. Conference on Human Factors in Computing Systems*, ACM Press, New York.
- Russell, S.; Norvig, P. (2003) “*Artificial Intelligence: A Modern Approach*”, Prentice Hall, Second Edition.

- Sashima, A.; Izumi, N.; Kurumatani, K. (2004) "Location-aware Middle Agents in Pervasive Computing", In *The 2004 International Conference on Pervasive Computing and Communications*, 2004, pp. 820-826.
- Torsun, I. S. (1995), *Foundations of Intelligent Knowledge-based systems*. London: Academic Press.
- Want, R.; Hoopper, A.; Falcão, V.; Gibbons, J. (1992) "The Active Badge Location System". *ACM Trans. On Information Systems*, 10 (1): 91-102.
- Weyns, D.; Holvoet, T. (2004) "A Formal Model for Situated Multiagent Systems", *Fundamenta Informaticae*, v.63 (2-3), pp.125-158.
- Wooldridge, M. (2002) "An Introduction to Multiagent Systems". Ed. John Wiley & Sons Ltd., 2 ed.