

Prometheus: Metodologia de Modelagem utilizada para a Simulação de Agentes da Construção Naval

Berny, Vanessa Maia; Jeannes, Fernanda Mendez;
Costa, Antônio Carlos da Rocha; Du Bois, André Rauber

Mestrado em Ciência da Computação – Universidade Católica de Pelotas
Pelotas – RS – Brazil

{vmberny, rocha, dubois}@ucpel.tche.br, fejeannes@gmail.com

Abstract. *This paper presents a case study in naval construction industry using Multi-Agent Systems, providing an overview of vessels construction project similar to the reality. Moreover, it presents three methods to model software agents and the cases use diagrams basing to Prometheus methodology.*

Keywords: *Software Agents, Artificial Intelligence.*

Resumo. *Este artigo tem o intuito de mostrar o uso de Sistemas Multiagentes utilizando a simulação dos agentes da indústria da construção naval, possibilitando uma visão geral do projeto de construção das embarcações aproximadamente comparada com a realidade. Além disso, são apresentadas três metodologias de modelagem para agentes de software, e o modelo de diagrama de casos de uso conforme a metodologia Prometheus que será utilizada na estruturação do simulador.*

Palavras-Chave: *Agentes de Software, Inteligência Artificial.*

1. Introdução

O objetivo deste trabalho é apresentar o uso da arquitetura de sistemas multiagentes (SMAs) como uma aplicação de estratégia para a indústria da construção naval, realizando simulações dos agentes e suas interações. Esta indústria é estratégica para o mercado global, pois é fornecedora da base de transporte para a hegemonia econômica e militar dos países [Ferraz et al. 2002].

Este segmento da indústria caracteriza-se por processos altamente complexos e com inúmeras interações, pois muitos fatores devem ser relevantes na construção das embarcações, como porte, tipo de embarcação, mão-de-obra qualificada, entre outras, podem transformar-se em sérios problemas.

Este problema apresenta características classificando-o de natureza distribuída, sendo assim, como proposta de amenizar este caso, serão realizadas as seguintes atividades:

1. A análise de requisitos estruturada no Trabalho Individual intitulado “Um Estudo sobre Modelagem Conceitual Baseada em Agentes” [Silva 2006];
2. A criação de um modelo conceitual representando a estrutura e o comportamento do simulador, utilizando a metodologia Prometheus [Padgham and Winikoff 2004], e a ferramenta para concepção dos diagramas Prometheus Design Tool [Prometheus 2008]; e

3. A implementação do simulador, utilizando a linguagem AgentSpeak(L) através do interpretador Jason [Bordini et al. 2007].

Sendo assim, a implementação do simulador ocasionará ao problema, uma maneira virtual que se aproxime da realidade e a possibilidade de aparição de fatores que são relevantes para o crescimento da indústria naval, como a padronização das embarcações, a rapidez na entrega, alta tecnologia utilizada juntamente com o custo-benefício agregado ao produto final e um processo de produção bem conhecido e modelado garantindo o sucesso do projeto de construção.

2. Indústria da Construção Naval

A missão básica da indústria da construção naval é suprir o mercado com novos navios. Para atender este mercado, as empresas desta indústria desenvolvem uma série de atividades em um ambiente econômico-industrial complexo, sofisticado e altamente competitivo [Cunha 2006].

A fabricação de um navio é um projeto de engenharia complexo, envolve inúmeras atividades multidisciplinares, desempenhadas em um longo período de tempo, que mobiliza quantitativos vultosos recursos de diversas naturezas e é suportada por um volume expressivo de dinheiro [Cunha 2006].

Conforme [Porter and Cho 1986], a indústria da construção naval é uma atividade econômica de natureza global, isto é, a posição competitiva de uma empresa do setor em um país é fortemente influenciada pela sua posição em outros países. Invariavelmente os armadores, os principais clientes desta indústria, utilizam-se de concorrências internacionais para suprirem suas frotas [Porter and Cho 1986] e [Stopford 1997].

A Fig. 1 representa o modelo analítico conforme [Porter 2004], com o propósito de identificar as fontes subjacentes das cinco forças competitivas que modelam a estrutura econômica da atividade da construção naval.

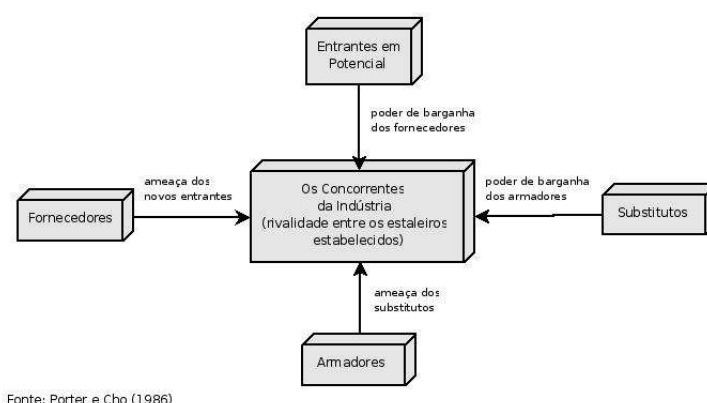


Figura 1. As Forças Competitivas do Ambiente da Indústria da Construção Naval

Os estaleiros competem entre si pelas encomendas dos armadores. Estes por sua vez tentam impor o seu poder de barganha exigindo menores preços e padrões de qualidade superiores. Na outra extremidade encontram-se os fornecedores que procuram capitalizar para si uma parcela do valor gerado pela indústria nos preços dos insumos fornecidos. Em meio a essas pressões, o ambiente competitivo da indústria de construção

naval sofre a permanente ameaça da chegada de novos concorrentes, dispostos a lutar por uma fatia do mercado, sem dizer a possibilidade de embate com produtos substitutos com melhor relação custo-benefício [Porter and Cho 1986]. De acordo com a referência citada, a rivalidade entre os concorrentes e o poder de negociação dos compradores (os armadores) são as forças predominantes da indústria.

3. Sistemas Multiagentes

Atualmente, há um crescimento bem significativo de desenvolvimento de softwares mais sofisticados para diversas áreas de conhecimento, como por exemplo, sistemas de tomada de decisão. Tais sistemas são conhecidos como agentes de software.

De acordo com [Sycara 1998], um agente é uma parte de software que age em prol do usuário ou de outro programa, obedecendo a regras de relacionamento da sociedade em que está inserido, o ambiente. Suas ações comportamentais são decididas quando melhor lhe convir. Essa idéia diz que agentes não são somente chamados ou 'invocados' exclusivamente por uma tarefa, mas também ativados por decisão própria. Este tipo de software é relacionado e derivado de conceitos de agentes, como:

1. **Inteligentes:** exibem aspectos de inteligência artificial, por exemplo, a aprendizagem e o raciocínio;
2. **Autônomos:** capazes de modificar a forma de como atingir seus objetivos;
3. **Distribuídos:** podem ser executados em diferentes máquinas físicas;
4. **Sistemas Multiagentes:** possuem agentes que não são capazes de atingir seus objetivos sozinhos e assim necessitam de meios de comunicação;
5. **Móveis:** podem ser realocados em diferentes processadores.

Há diversos tipos de agentes, dentre eles os cognitivos, que podem ser baseados em estados mentais, com capacidade de raciocínio, pois bem, capazes de construir um ou mais planos de ações que os levam a um estado pretendido. Adota-se assim, uma perspectiva de inspiração psicológica para definir suas estruturas [Zamberlam et al. 2007]. Esse tipo de agente possui características particulares como:

1. Autonomia funcional;
2. Encontram-se continuamente em funcionamento;
3. São sociáveis, ou seja, possuem capacidade de comunicação/interação;
4. O mecanismo de controle é deliberativo;
5. Possui memória;
6. As sociedades são formadas por poucos agentes.

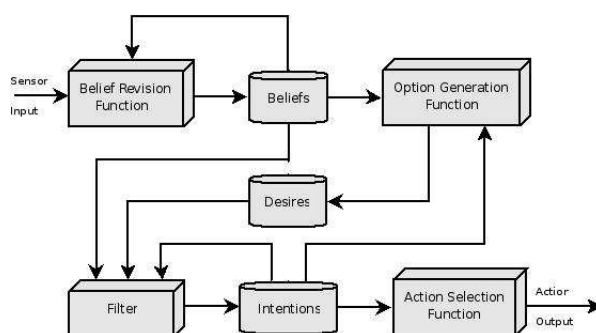
Sendo assim, os sistemas de agentes de softwares são bastante complexos e computacionalmente pesados [Hubner 2003] e [Zamberlam 2001]. Esses agentes podem ser definidos como entidades inseridas em um ambiente, aptos em perceber e agir de forma persistente nas suas metas e comunicar-se com outros agentes, assumindo comportamentos autônomos [Dalcastagne 2006] e [Russel and Norvig 1995].

As estruturas desses agentes são de entidades, cujo estado é constituído por componentes mentais, tais como crenças, desejos, capacidades, escolhas e compromissos. Dentro das arquiteturas baseadas em estados mentais encontra-se a abordagem de BDI (Belief, Desire and Intention), que tem sua origem no modelo de raciocínio prático humano. O nome atribuído ao modelo é justificado pelos estados mentais: crenças, desejos

e intenções. Essa arquitetura representa seus processos internos através desses estados mentais e define um mecanismo de controle que seleciona de maneira racional o curso das ações [Giraffa 1999].

As três atitudes mentais que compõem o modelo BDI são melhor detalhadas a seguir [Wooldridge 1999]:

1. **Crenças (Beliefs):** Representam as características do ambiente, as quais são atualizadas apropriadamente após a percepção de cada ação. Podem ser vistas como o componente informativo do sistema;
2. **Desejos (Desires):** Contêm informação sobre os objetivos a serem atingidos, bem como as prioridades e os custos associados com os vários objetivos. Podem ser pensados como a representação do estado motivacional do sistema;
3. **Intenções (Intentions):** Representam o atual plano de ação escolhido. Capturam o componente deliberativo do sistema.



Fonte: Wooldridge (1999)

Figura 2. Diagrama de uma Arquitetura BDI Genérica

O processo de raciocínio prático em um agente BDI é apresentado na Fig. 2 [Wooldridge 1999]. Conforme o autor, existem sete componentes principais em um agente BDI:

1. Conjunto de crenças (beliefs) atuais, representando a informação que o agente tem sobre seu ambiente;
2. Função de revisão de crenças (belief revision function), a qual determina um novo conjunto de crenças a partir da percepção da entrada e das crenças do agente;
3. Função de geração de opções (option generation function), a qual determina as opções disponíveis aos agentes (seus desejos), com base nas suas crenças sobre seu ambiente e nas suas intenções;
4. Conjunto de opções/desejos (desires) corrente que representa os possíveis planos de ações disponíveis ao agente;
5. Função de filtro (filter), a qual representa o processo de deliberação do agente, que determina as intenções do agente com base nas suas crenças, desejos e intenções atuais;
6. Conjunto de intenções (intentions) atuais, que representa o foco atual do agente, isto é, aqueles estados que o agente está determinado a alcançar;
7. Função de seleção de ação (action selection function), a qual determina uma ação a ser executada com base nas suas intenções atuais.

A metodologia utilizada para a estruturação da modelagem dos agentes do simulador será a Prometheus. Metodologia para análise, projeto e implementação de SMAs, baseada em agentes BDI. Esta é constituída de três fases: especificação do sistema (papéis e classes de ações dos agentes), projeto da arquitetura (funcionalidades, atributos dos agentes e interações), e projeto detalhado (planos, eventos internos e externos).

A ferramenta que será utilizada à nível de implementação é o Jason [Bordini et al. 2007]. Uma plataforma de desenvolvimento de SMAs baseada em um interpretador para uma versão estendida da linguagem AgentSpeak(L). Esta é uma linguagem de programação orientada a agentes baseada na lógica de primeira ordem, com eventos e ações, inspirada na arquitetura BDI [Rao and Georgeff 1995].

4. Um SMA para a Construção Naval

Conforme [Silva 2006], as principais classes de agentes identificadas na indústria da construção naval são:

Concorrentes da Indústria Naval; Entrantes Potenciais; Substitutos; Governo; Organizações Internacionais; e o Sistema Financeiro.

Este autor considera entre os concorrentes da indústria naval basicamente os estaleiros. Já os entrantes potenciais são divididos em:

Estaleiros; Armadores; e os Fornecedores.

Logo o trabalho proposto, é de modelar e implementar alguns destes agentes e simulá-los, obtendo-se estudos e observações do comportamento destes e suas interações, tentando cientificamente realizar uma reprodução artificial de um fenômeno real, isto é, a indústria da construção naval.

5. Metodologias de Modelagem de Agentes

Uma metodologia de desenvolvimento de SMAs é constituída por uma série de passos e procedimentos a serem seguidos durante o processo de concepção do sistema. Ela deve capturar a flexibilidade, autonomia dos agentes, com variados graus de abstração, auxiliando o projetista nas tomadas de decisão relativas à análise, projeto e implementação [Rabelo 2007].

As metodologias apresentadas a seguir correspondem a metodologias para desenvolvimento de SMAs, que utilizam linguagens de modelagem específicas para estes tipos de sistemas:

1. Metodologia TROPOS [Bresciani et al. 2004]

É uma metodologia para SMAs que fornece suporte as atividades de análise e projeto no desenvolvimento do sistema, desde a análise até a implementação do mesmo. Esta está dividida em cinco fases: fases inicial e final de requisitos, projetos arquitetural e detalhado e implementação.

Para a análise de requisitos há duas etapas: as fases inicial e final. Na inicial são definidos os stakeholders do domínio, modelados como atores sociais, com dependências baseadas em objetivos, planos e fornecimento de recursos. Já na final o modelo conceitual é estendido, inclui-se um novo ator que representa o sistema e as dependências com os outros atores do ambiente [Giunchigli et al. 2002].

As fases de projetos arquitetural e detalhado, são relacionadas à especificação do sistema. O arquitetural define a arquitetura global do sistema em termos de sub-sistemas (atores), troca de dados e fluxo de controles (dependências), também é realizado um mapeamento dos atores do sistema em um conjunto de agentes de software, cada um caracterizado com suas capacidades [Bresciani et al. 2004]. O detalhado especifica as habilidades dos agentes e suas interações. Cada agente é definido, mais especificamente em termos de entrada, saída, controle e outras informações relevantes para o sistema, já nesta fase é escolhida a plataforma de desenvolvimento.

A implementação se baseia na definição do projeto detalhado. O Tropos usa a plataforma para agentes BDI chamada Jack [Jack 2008], para a implementação dos SMAs. Os agentes em Jack são componentes autônomos que apresentam objetivos a serem alcançados ou eventos a serem tratados. Estes agentes são programados com um conjunto de planos tornando-os capazes de alcançar seus objetivos.

A modelagem realizada em Tropos é bastante confusa e rebuscada, dificultando a fase do processo de desenvolvimento. A fase de projeto detalhado é orientada especificamente à plataforma Jack [Maria 2005].

2. Metodologia GAIA [Zambonelli et al. 2003]

Esta metodologia possui uma linguagem própria para a modelagem de SMAs. O processo de desenvolvimento contém duas fases: análise e design. Esta metodologia tem início na fase de análise, visando coletar e organizar a especificação que servirá de base para a fase de design.

A fase de análise tem o objetivo de entender o sistema e decompô-lo em papéis que serão desempenhados na organização, através do modelo de papéis, e definir como os mesmos interagem, através do modelo de interação. Portanto, o modelo de papéis identifica os papéis existentes no sistema e o modelo de interação um conjunto de definições de protocolos, um para cada tipo de interação entre os papéis. Esta fase de análise inclui identificar [Zambonelli et al. 2003]: as metas das organizações presentes no sistema e o comportamento esperado dos mesmos, o ambiente, os papéis iniciais, as interações iniciais e as regras que a organização deve seguir.

Os componentes gerados na fase de análise, são utilizados como entrada para a fase de design. Esta fase pode ser logicamente decomposta em duas novas fases: A fase da elaboração da arquitetura que inclui a definição da estrutura organizacional do sistema em termos de sua topologia e regime de controle e a identificação completa dos papéis e interações; E a fase de detalhamento que compreende a definição do modelo de agentes e a definição do modelo dos serviços que os agentes devem oferecer para desempenhar seus papéis.

Para esta metodologia o SMA deve ser estático, onde o número de agentes, seus comportamentos, habilidades e as inter-relações entre eles não mudam [Rabelo 2007].

3. Metodologia PROMETHEUS [Padgham and Winikoff 2004]

Metodologia para desenvolvimento de SMAs que abrange desde a modelagem até a implementação. Esta metodologia é composta por três fases, onde os componentes produzidos são utilizados tanto na geração do esqueleto do código, como também para realização de testes.

A primeira fase, corresponde à especificação do sistema, compreende em duas atividades: determinar o ambiente do sistema e determinar os objetivos e funcionalidades do mesmo. O ambiente do sistema é definido em termos de percepções (informações do ambiente) e ações. Além disso, também são definidos dados externos. As funcionalidades do sistema são definidas através da identificação de objetivos, da definição das funcionalidades necessárias para se alcançar esses objetivos e dos cenários de casos de uso [Dam and Winikoff 2003].

A fase seguinte é o projeto arquitetural que utiliza as saídas da fase anterior para determinar quais agentes existirão no sistema e como os mesmos irão interagir. Esta fase envolve três atividades: definição dos tipos de agentes, definição da estrutura do sistema e definição das interações entre os agentes [Padgham and Winikoff 2002].

O projeto detalhado é a última fase que é responsável por definir capacidades dos agentes, eventos internos, planos e uma estrutura de dados detalhada de cada tipo de agente identificado na fase anterior.

Atualmente existem duas ferramentas que utilizam o Prometheus. O Jack Development Environment (JDE) [Jack 2008], software comercial que inclui uma ferramenta de modelagem para a construção dos diagramas, resultando na geração do código na linguagem de programação Jack. Esta ferramenta dá suporte à metodologia Prometheus pelo fato dos conceitos utilizados pelo Jack corresponderem aos componentes gerados na fase de projeto detalhado da metodologia.

A outra ferramenta é o Prometheus Design Tool (PDT). Esta permite que o usuário entre e edite o projeto utilizando os seguintes conceitos: verificar o projeto para um conjunto de possíveis inconsistências, gerar automaticamente um conjunto de diagramas de acordo com a metodologia e gerar automaticamente a descrição do projeto, o que inclui descritores para cada entidade, um dicionário para o projeto e os diagramas gerados anteriormente.

Esta metodologia facilita o aprendizado de sistemas baseados em agentes, é bem flexível quanto ao tipo de agentes e possibilita a validação e geração de código, mas poucos artigos sobre a metodologia são publicados [Drummond and Benzatti 2008].

6. Fase de Especificação do Simulador: Cenários de Casos de Uso

Conforme a metodologia Prometheus a primeira fase é a especificação do sistema proposto que se resume nas seguintes atividades [Drummond and Benzatti 2008]:

1. Identificar objetivos;
2. Desenvolver cenários de casos de uso;
3. Identificar a interface do sistema;
4. Identificar funcionalidades;
5. Identificar as informações de entrada e saída das funcionalidades;
6. Preparar esquemas das funcionalidades (nome, descrição, ações, dados, etc).

Com base nas informações citadas, o artigo abordará esta primeira etapa de desenvolvimento baseado na metodologia Prometheus apresentando o cenário de casos de uso do simulador de agentes da construção naval, mostrando a funcionalidade de cada agente com suas ações, objetivos e interações. Os componentes que fazem parte do diagrama são:

1. **Agente:** Indivíduo externo que interage com o sistema;
2. **Ação:** Atividade realizada pelo agente a curto prazo;
3. **Objetivo:** Atividade realizada pelo agente a longo prazo;
4. **Interação:** A interação/atividade que envolve dois ou mais agentes;

A Fig. 3 representa graficamente os componentes citados acima.

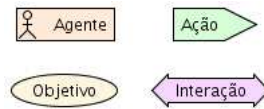


Figura 3. Componentes do Modelo de Diagrama de Casos de Uso

Para este simulador de agentes, foram escolhidos alguns dos agentes, apresentados na Seção 4, para uma simulação inicial, estes agentes já estão presentes no diagrama de casos de uso apresentado na Fig. 4.

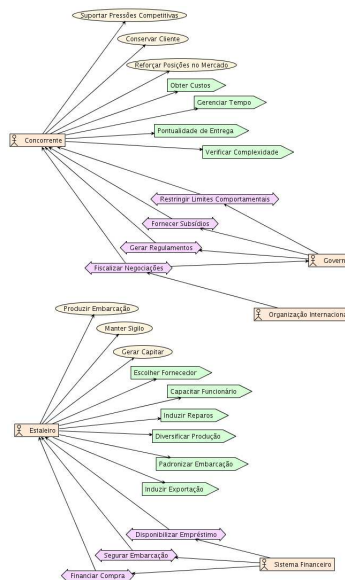


Figura 4. Diagrama de Casos de Uso do Simulador de Agentes

O propósito de apresentar o cenário de casos de uso com estes componentes, é a representação das atividades que podem ser realizadas por um ou mais agentes, facilitando o entendimento dos mesmos.

7. Conclusão e Resultados Esperados

Devido ao crescimento do desenvolvimento de SMAs, se torna necessário criar linguagens, metodologias e processos de desenvolvimento que facilitem a estruturação destes sistemas. Após analisar na literatura um conjunto variado de metodologias, foi possível perceber que muitas delas não contemplavam todo o processo de desenvolvimento ou não apresentavam de forma muito clara como a partir de uma dada modelagem se obtém o seu código correspondente, mas estas ferramentas são de grande valia para o desenvolvimento de SMAs.

O SMA proposto se comportará como um simulador de agentes e deverá ajudar a definir com precisão os aspectos relevantes de todos os agentes, o ambiente em que estão inseridos, bem como as interações entre eles realizadas, de acordo com que esta simulação esteja bem próxima as interações entre os agentes reais envolvidos na indústria da construção naval.

Referências

- Bordini, R. H., Wooldridge, M., and Hubner, J. F. (2007). *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley e Sons, 225p, London.
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). *Tropos: An Agent-Oriented Software Development Methodology*. Journal of Autonomous Agents and Multi-Agent Systems, p. 203–236.
- Cunha, M. S. (2006). *A Indústria de Construção Naval: Uma Abordagem Estratégica*. Master's thesis, Universidade de São Paulo - Escola Politécnica, São Paulo - SP.
- Dalcastagne, D. (2006). *Implementação de uma Livraria Virtual Utilizando Agentes BDI Através da Linguagem Agentspek(L)*. Universidade Regional de Blumenau - Centro de Ciências Exatas e Naturais, Blumenau - SC.
- Dam, K. H. and Winikoff, M. (2003). *Comparing Agent-Oriented Methodologies*. International Bi-Conference Workshop on Agent-Oriented Information Systems, p. 78–93, Melbourne.
- Drummond, A. and Benzatti, D. (2008). *Metodologia para o Desenvolvimento de Sistemas Baseados em Agentes Prometheus*. Disponível em: [http://www.ic.unicamp.br/~eliane/Cursos/MO409/Curso2004\(com%20MC746\)/Apresentacoes/g6_a2_agentes2.ppt](http://www.ic.unicamp.br/~eliane/Cursos/MO409/Curso2004(com%20MC746)/Apresentacoes/g6_a2_agentes2.ppt), acesso em: jul/2008.
- Ferraz, J. C., Leão, I., Santos, R. L. C., and Portela, L. M. (2002). *Estudo da Competitividade de Cadeias Integradas no Brasil: Impactos das Zonas de Livre Comércio*. Technical report, Universidade Estadual de Campinas - Instituto de Economia, Campinas - SP.
- Giraffa, L. M. M. (1999). *Uma Arquitetura de Tutor Utilizando Estados Mentais*. PhD thesis, Universidade de São Paulo - Universidade Federal do Rio Grande do Sul - Doutorado em Ciência da Computação, Porto Alegre - RS.
- Giunchigli, F., Mylopoulos, J., and Perini, A. (2002). *The Tropos Software Development Methodology: Processes, Models and Diagrams*. International Workshop on Agent-Oriented Software Engineering, p. 162–173.
- Hubner, J. F. (2003). *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Universidade de São Paulo - Escola Politécnica, São Paulo - SP.
- Jack (2008). *Jack*. Disponível em: <http://www.agent-software.com/products/jack/>, acesso em: jul/2008.
- Maria, B. A. D. (2005). *Usando a Abordagem MDA no Desenvolvimento de Sistemas Multi-Agentes*. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro - RJ.

- Padgham, L. and Winikoff, M. (2002). *Prometheus: A Pragmatic Methodology for Engineering Intelligent Agents*. Workshop on Agent-Oriented Methodologies, p. 97–108.
- Padgham, L. and Winikoff, M. (2004). *Developing Intelligent Agent Systems: A Practical Guide*. RMIT University, Melbourne, 226p, Austrália.
- Porter, M. E. (2004). *Estratégia Competitiva: Técnicas para Análise de Indústrias e da Concorrência*. Campus Elsevier, 409p, Rio de Janeiro - RJ.
- Porter, M. E. and Cho, D. S. (1986). *Competition in Global Industries*, chapter Changing Global Industry Leadership: The Case of Shipbuilding. Harvard Business School Press, p. 539-567, Massachusetts.
- Prometheus (2008). *Prometheus Design Tool (PDT) v.3.2b*. Disponível em: <http://www.cs.rmit.edu.au/agents/pdt/>, acesso em: jul/2008.
- Rabelo, R. J. (2007). *Maritime Economics*. Disponível em: <http://www.das.ufsc.br/%7Erabelo/Ensino/DAS6607/Aula5.pdf>, acesso em: jul/2008.
- Rao, A. S. and Georgeff, M. P. (1995). *BDI-Agents: From Theory to Practice*. Technical report, Proceedings of The First International Conference on Multiagent Systems, San Francisco.
- Russel, S. and Norvig, P. (1995). *Inteligência Artificial*. 2ª ed, Campus Elsevier, 1040p, Rio de Janeiro - RJ.
- Silva, R. E. S. (2006). *Um Estudo sobre Modelagem Conceitual Baseada em Agentes*. Technical report, Universidade Católica de Pelotas - Mestrado em Ciência da Computação, Pelotas - RS.
- Stopford, M. (1997). *Maritime Economics*. 2ª ed, Routledge, 592p, London.
- Sycara, K. P. (1998). *Multiagent Systems*. Association for the Advanceent of Artificial Intelligence (AAAI), p. 79–92, San Francisco.
- Wooldridge, M. (1999). *Intelligent Agents*. Multiagent Systems, The MIT Press, p. 27–78.
- Zamberlam, A. O. (2001). *Em Direção a uma Técnica para Programação Orientada a Agentes BDI*. Master's thesis, Universidade Federal do Rio Grande do Sul - Mestre em Ciência da Computação, Porto Alegre - RS.
- Zamberlam, A. O., Goulart, R. R. V., and Peres, C. C. C. (2007). *Avaliação de um Jogo Inteligente no Processo de Aprendizagem*. XXVII Congresso da Sociedade Brasileira de Computação - XIII Workshop Sobre Informática na Escola, p. 272–279, Rio de Janeiro - RJ.
- Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2003). *Developing Multiagent Systems: The Gaia Methodology*. ACM Transactions on Software Engineering and Methodology, p. 317–370.