

Projetando Sistemas Multiagentes em Organizações Empresariais¹

Murilo Juchem, Ricardo Melo Bastos
Pontifícia Universidade Católica do Rio Grande do Sul
Programa de Pós-Graduação em Ciência da Computação
Porto Alegre - RS – Brasil
e-mail: mjuchem@terra.com.br, bastos@inf.pucrs.br

Resumo

Considerando as características dos sistemas de informação tipicamente utilizados para atendimento aos processos do negócio de organizações empresariais, este trabalho apresenta uma abordagem para a modelagem de sistemas multiagentes (SMA) tendo como foco o modelo de projeto. Para a construção dos modelos são utilizadas extensões da UML. O processo de modelagem é baseado no refinamento entre os modelos, o que implica na evolução dos diagramas elaborados no modelo de análise em direção a construção do modelo de projeto. Para o atendimento aos serviços de infra-estrutura necessários para o processamento do sistema são propostos agentes especializados denominados agentes de infra-estrutura. Tal proposição está fundamentada na premissa de que os agentes devem ter elevado grau de especialização em uma sociedade de agentes. Sendo assim, aos agentes do domínio do SMA devem ser definidas atribuições restritas aos papéis que lhes foram destinados no modelo de casos de uso, cabendo ao conjunto de agentes de infra-estrutura cobrir os demais serviços necessários ao processamento do sistema.

Abstract

This work presents an approach to design multi-agent systems (MAS) for enterprise information systems. Both analysis and design models are depicting through extensions of UML. The modeling process is based on the evolution of the development models where the analysis model diagrams are refined in order to specify the design model. The domain agents attributions are defined according the roles assigned for these ones in the use case model. Considering that the agents are entities with specific roles in the multi-agent society it is proposed specialized agents called infrastructure agents to support the infrastructure services required for the system execution.

1. Introdução

Considerando as características e propriedades dos sistemas multiagentes (SMA), sua aplicação permite a definição de soluções para problemas em que os paradigmas tradicionais da engenharia de software apresentam limitações. Através da aplicação de SMA, torna-se exequível a definição de uma solução distribuída e descentralizada, modelando autonomia e colaboração entre os agentes participantes do sistema, seja no nível de ativação e controle de sua execução, seja no que se refere as suas responsabilidades e capacidades dentro da sociedade.

Do ponto de vista da produção industrial de software, em particular no atendimento as demandas dos sistemas de informação em organizações empresariais, tem-se observado

¹ Este trabalho foi parcialmente financiado pela FAPERGS – Processo 01/0569.6 e CNPq.

esforços modestos na aplicação deste paradigma. Isto se justifica pela inexistência de métodos e linguagens de modelagem consolidadas, bem como de ferramentas CASE para ambiente de produtividade em software. No entanto, é evidente o crescimento de trabalhos objetivando preencher esta lacuna [4] [5] [6] [7] [8] [9] [10] [11] [12] [13], embora muitos aspectos ainda permanecem não resolvidos, especialmente sob o ponto de vista da engenharia de software.

Um agente pode ser definido como uma entidade de software que está situado em algum ambiente no qual é capaz de realizar ações flexíveis e autônomas para alcançar seus próprios objetivos de projeto [1].

Uma solução baseada em SMA é elaborada através da criação de uma sociedade onde os agentes trabalham de forma coordenada e cooperativa, cabendo a cada um desempenhar um conjunto de atribuições para o atendimento aos requisitos do sistema. Assim sendo, uma das principais características de um SMA é a descentralização do controle, que é distribuído entre os vários agentes que compõem o sistema. Desta forma, o comportamento global do sistema deriva da interação entre os agentes que fazem parte do sistema [9].

Segundo [2], a aplicação de agentes em sistemas de informação se justifica quando observadas as seguintes características:

- o domínio envolve uma inerente distribuição de (i) dados, (ii) capacidades para resolução de problemas e (iii) responsabilidades;
- há a necessidade de manter a autonomia das subpartes envolvidas, bem como a estrutura organizacional;
- as interações são razoavelmente sofisticadas, incluindo negociação, compartilhamento de informações e coordenação;
- a solução para o problema não pode ser inteiramente descrita do princípio ao fim, devido às diversas mudanças ambientais que podem ocorrer, além da imprevisibilidade e dinamicidade dos processos de negócio e a capacidade pró-ativa dos agentes.

Um SMA apresenta dois níveis distintos em seu processo de modelagem: o nível macro, envolvendo a sociedade de agentes e o nível micro, referente ao agente [8]. No nível macro são modeladas as relações, interações e colaborações dos agentes componentes do SMA. No nível micro é especificada a arquitetura e o comportamento de cada classe de agente.

Este trabalho envolve a modelagem de SMA a nível macro, tendo como foco a elaboração do modelo de projeto. Para a construção dos modelos são utilizadas extensões da UML – *Unified Modeling Language* propostas em [3]. O processo de modelagem do SMA é baseada no refinamento entre modelos conforme proposto em [14], o que implica na evolução dos diagramas elaborados no modelo de análise em direção a construção do modelo de projeto, dando forma a uma arquitetura de sistema capaz de realizar os casos de uso previamente definidos.

Na segunda seção são realizadas considerações sobre a modelagem de SMA. Na terceira seção é examinado o modelo de casos de uso que define as funcionalidades do sistema sob a ótica da orientação a agentes. Na quarta seção são definidos os artefatos a serem produzidos para o modelo de análise envolvendo os agentes capazes de realizarem os casos de uso. Na quinta seção são discutidos os aspectos relevantes a elaboração do modelo de projeto de um SMA, tendo como referências as características dos sistemas de informação tipicamente utilizados para atendimento aos processos do negócio de organizações empresariais, sendo definido um conjunto mínimo de agentes de infra-estrutura a serem incorporados na sociedade para a execução de serviços especializados. Por fim, na sexta seção são examinados os trabalhos relacionados e na sétima seção são apresentadas conclusões e trabalhos futuros.

2. A Modelagem de SMA

Conforme proposto em [14], as principais atividades envolvidas no desenvolvimento de um sistema são: definição de requisitos, análise, projeto, implementação e teste. Para cada uma destas atividades resultam modelos. Neste trabalho, são abordados os modelos de casos de uso, análise e projeto sob a ótica de um SMA. O modelo derivado da etapa de definição de requisitos é expresso através de casos de usos. O modelo de casos de uso provê um entendimento dos requisitos dos usuários, que são a base para a construção do sistema, uma vez que exprimem todo o conjunto de funcionalidades relevantes ao domínio do problema.

No modelo de análise, estas funcionalidades são refinadas, resultando na estratégia de solução multiagente a ser utilizada. Como resultado desta etapa são definidos os agentes do domínio que irão desempenhar os serviços necessários para o atendimento aos requisitos do sistema. Além disto, é definida a estrutura organizacional do sistema, sendo estabelecidas as políticas de coordenação, negociação, cooperação e planejamento entre os agentes que compõem a sociedade.

No modelo de projeto é dada forma ao sistema e definida a sua arquitetura. Esta arquitetura implica na definição da topologia do ambiente em que os agentes do domínio irão desempenhar suas atribuições. A criação do modelo de projeto do SMA se faz através do reuso dos diagramas produzidos para o modelo de análise, integrando-se os agentes de infra-estrutura necessários para a realização de serviços especializados para o funcionamento da sociedade, tais como agentes roteadores para a comunicação entre nodos da rede física, agentes para interface com a base de dados, etc. Portanto, a decisão sobre quais agentes do domínio irão compor o SMA para o atendimento a solução multiagente definida é tomada na etapa de análise. A etapa de projeto tem como objetivo viabilizar o funcionamento do SMA em um ambiente computacional, requerendo a integração de agentes de infra-estrutura.

Atualmente, a UML vem sendo considerada como linguagem de modelagem de referência para sistemas orientados a objetos. Esta posição é resultante dos recursos de representação e especificação de sistemas oferecidos pela linguagem, além do elevado volume de literatura sobre o assunto e disponibilidade de ferramentas CASE compatíveis com a mesma. Adicionalmente, a UML é reconhecida como padrão pelo *Object Management Group* (OMG). Como consequência, existem muitos esforços no sentido de utilizar UML como uma linguagem de modelagem para diferentes classes de problemas. Tais esforços também vêm sendo realizados no sentido de adaptar a UML para modelagem de SMA.

Entretanto, as diferenças existentes entre sistemas orientados a objetos e sistemas orientados a agentes requerem importantes adaptações da UML, uma vez que não é possível especificar adequadamente todos as propriedades de um SMA. Diversos trabalhos têm examinado as diferenças entre objetos e agentes, tais como [15] [16] [17]. No sentido de convergir para o estado da arte em engenharia de software, neste trabalho os esforços já realizados na adaptação da UML para o paradigma orientado a agentes são tomados como referência para a definição de técnicas para especificação de SMA.

A exemplo de [14], propõe-se que o processo de desenvolvimento de um SMA seja incremental, ou seja, o modelo de uma etapa é o resultado do refinamento dos modelos da etapas anteriores. Sendo assim, os artefatos produzidos para a representação do modelo de análise serão entrada para a elaboração do modelo de projeto, sendo-lhes adicionado os agentes de infra-estrutura e definida a disposição física dos agentes no ambiente de processamento.

Na seqüência, serão apresentados brevemente os principais artefatos produzidos para a especificação dos modelos de casos de uso e de análise para um SMA. Posteriormente, serão

definidos os aspectos a serem atendidos na elaboração do modelo de projeto, tendo como entrada os artefatos produzidos para estes dois modelos.

3. O Modelo de Casos de Uso

Para a atividade de definição de requisitos de um SMA, a exemplo do proposto em [14], é utilizado o Diagrama de Casos de Uso [18]. Este diagrama tem como principais elementos casos de uso, atores e relacionamentos. Um caso de uso representa uma seqüência de ações e suas variantes descrevendo uma funcionalidade observável para o sistema. Cada variante representa diferentes cenários desta funcionalidade, envolvendo um fluxo de eventos padronizado e um ou mais fluxos de eventos alternativos.

Os resultados produzidos pela ocorrência de um caso de uso devem ter valor para um ator. Um ator representa uma entidade externa que interage com o sistema participando de um ou mais casos de uso. Um ator ainda pode ser visto como um coerente conjunto de papéis que varia de acordo com o caso de uso com o qual ele interage. Um modelo de casos de uso para um sistema representa as funcionalidades do sistema do ponto de vista de seus usuários, sendo esta perspectiva válida também para um SMA.

Através do modelo de casos de uso é possível identificar e compreender os requisitos de um sistema, sendo viável a definição das vantagens e adequação na aplicação de uma solução baseada em SMA. Uma vez definida a validade da utilização de SMA, faz-se necessário especificar a solução explorando o potencial deste paradigma de acordo com suas propriedades e características.

Baseado nesta premissa, o próximo passo envolve a revisão do modelo de casos de uso no sentido de definir as novas funcionalidades requeridas para o sistema ao implementar uma solução baseada em agentes. Isto envolve a redefinição do modelo pela inclusão de novos casos de uso e atores produzindo o Modelo de Casos de Uso para o SMA.

Os novos casos de uso devem descrever as ações envolvidas na implementação da solução baseada em agentes, sendo que os novos atores representam os papéis envolvidos na interação com os casos de uso. Alguns trabalhos têm examinado papéis de agentes em SMA [8] [9] [10]. Estes trabalhos consideram a possibilidade um agente ocupar mais de um papel na sociedade. Sendo assim, um ator agente pode ser visto como a agregação de um ou mais papéis, e as relações entre os agentes na sociedade definidas de acordo com estes papéis.

Para representar um modelo de casos de uso para um SMA é proposta uma extensão do Diagrama de Casos de Uso da UML utilizando estereótipos. O papel de um agente é identificado por estereótipo <<role>> e um agente ator por <<agent>>. O refinamento do modelo de casos de uso do SMA definirá os agentes como agregações de papéis.

Considere por exemplo um sistema de produção para uma empresa de manufatura. Um diagrama de casos de uso parcial para um SMA envolvendo a alocação de recursos para atendimento a uma ordem de compra é apresentado na Figura 1. Neste sistema é apresentada uma simplificação do sistema M-DRAP – *Multi-Agent Dynamic Resource Allocation Planing* proposto em [19], onde um processo do domínio é disparado por evento tal como uma ordem de produção, sendo decomposto funcionalmente em atividades empresariais. Atividades empresariais correspondem a tarefas elementares das funcionalidades de uma empresa, e que são normalmente executadas sob restrições. As atividades empresariais requerem alocação de tempo e recursos para sua execução.

A solução multiagente adotada envolve a aplicação do *FIPA-Contract-net protocol* [20] para o processo de alocação de recursos para o atendimento as atividades empresariais a serem executadas. Cada atividade empresarial requerida para execução pelo agente *processo*

do domínio pode requerer um ou mais recursos, sendo necessária, portanto, a formação de uma coalizão. Cada coalizão candidata a execução de uma atividade empresarial deve encaminhar uma proposta, cuja decisão pela contratação é tomada pelo agente *atividade empresarial*. Com o refinamento do modelo de casos de uso, considerando o conjunto de atribuições associadas a cada um dos papéis identificados no modelo, são identificados os agentes responsáveis pela sua realização. Neste exemplo, pode-se definir um ator agente *entidade funcional* como responsável por desempenhar os papéis *recurso candidato*, *recurso membro da coalizão* e *recurso coordenador da coalizão*.

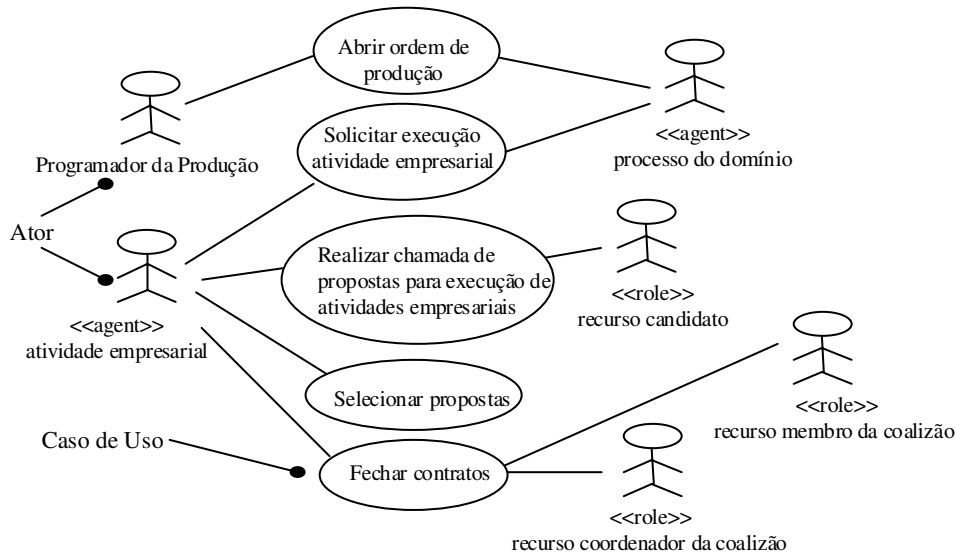


Figura 1 – Diagrama de Casos de Uso

4. O Modelo de Análise

Para a elaboração do modelo de análise, neste trabalho foi adotada a proposta apresentada em [3]. Conforme a autora, o modelo de análise pode ser especificado através do Diagrama da Sociedade, do Modelo de Funções e dos Diagramas de Interação Seqüencial. No sentido de permitir uma melhor compreensão sobre a construção do modelo de projeto apresentado na próxima seção deste artigo, na seqüência é realizada uma breve exposição dos diagramas e notações envolvidos, sendo elaborado o modelo de análise parcial referente ao diagrama de casos de uso da Figura 1.

4.1. O Diagrama da Sociedade

O Diagrama da Sociedade é inspirado no Diagrama de Classes da UML [18] representando a visão estática de um SMA. Neste diagrama são representadas as classes de agentes identificadas a partir do modelo de casos de uso do SMA, descrevendo suas propriedades, relações de hierarquia e papéis.

Embora ainda não exista consenso sobre o conceito de agente, do ponto de vista da engenharia de software um agente pode ser definido como sendo uma entidade computacional projetada para suportar as características de um ambiente computacional específico, dotada de comportamento autônomo sob o ponto de vista de capacidade de tomada de decisão, definida

para suportar um conjunto finito de atribuições em uma sociedade e com capacidade para interagir com outros agente através da troca de mensagens. Uma classe de agente deve especificar o nome do agente, quantidade máxima de instâncias possíveis na sociedade, atributos, serviços, papéis e atribuições compartilhadas por um conjunto de agentes. Na Figura 2 é representada a classe *agente atividade empresarial*.

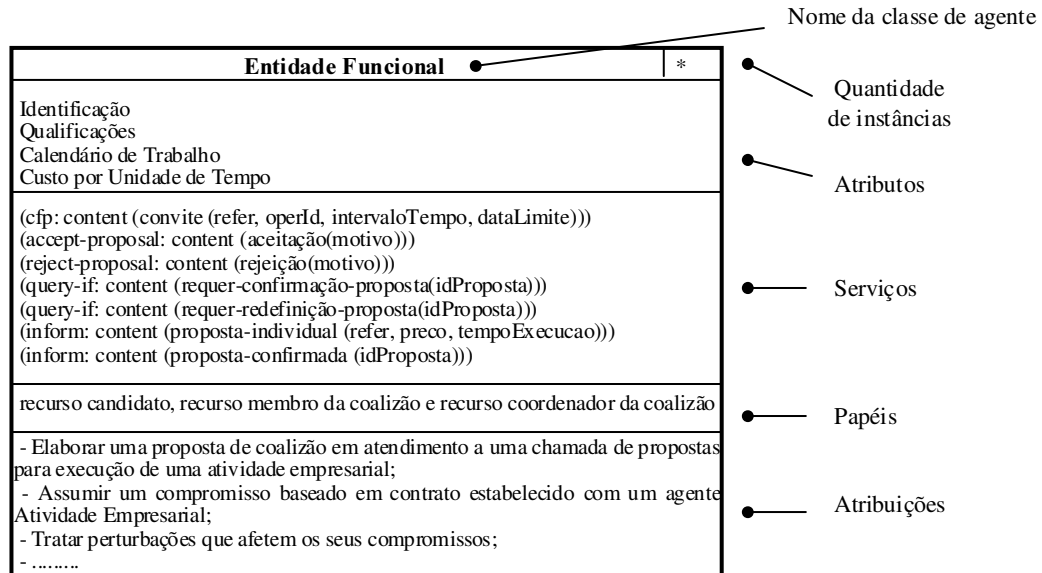


Figura 2 – Classe de Agente

A **quantidade** representa o número mínimo e máximo de instâncias de agentes da classe, podendo ser exatamente (1), um intervalo (1..3), um ou mais (1..*) ou muitos (*). Um **atributo** representa a identificação do agente e o conjunto de informações que são utilizadas/atualizadas pelo mesmo durante o processamento de suas atribuições. Os **serviços** definem a *interface* do agente, especificando o conjunto de atos de comunicação que o mesmo é capaz de interpretar e atender. Neste trabalho é aplicada uma notação simplificada da *Action Communication Language (ACL)* [20] no formato (**Communication act identification: content()**). Estes serviços devem ser consistentes com as atribuições do agente na sociedade. As **atribuições** expressam todas as funções que o agente deve ser capaz de executar dentro da sociedade visando atender aos seus objetivos.

No diagrama da sociedade, as classes de agentes são representadas por retângulos onde é definido pelo menos o nome e a quantidade máxima de instâncias. As associações entre as classes de agentes representam os canais de comunicação para troca de mensagens, definindo também as relações de hierarquia na sociedade. O nome das associações é opcional, porém deve ser definida a multiplicidade existente entre os agentes envolvidos: exatamente (1), um intervalo (1..3), um ou mais (1..*) ou muitos (*). O papel do agente na associação deve ser identificado próximo ao mesmo. Na Figura 3 é apresentado o diagrama da sociedade envolvendo os agentes identificados para a realização dos casos de uso da Figura 1.

As associações podem ser de comando (permanente ou dependente do papel) ou de comunicação. Nas associações de comando permanente, uma requisição recebida por um agente deve ser obrigatoriamente atendida, estabelecendo uma relação de autoridade entre os agentes envolvidos. Esta associação é representada por uma seta cheia. Nas associações

dependentes do papel, somente um subconjunto dos atos de comunicação (definindo a interface do papel) são válidos para atendimento pelo agente a uma requisição recebida de outro agente. Isto significa que neste tipo de associação uma requisição provinda de outro agente somente será atendida se na interface do papel constar o ato de comunicação correspondente. Esta associação é representada por uma meia seta cheia. Nas associações de comunicação, requisições não têm obrigatoriedade de serem respondidas. Esta associação é representada por uma seta vazada.

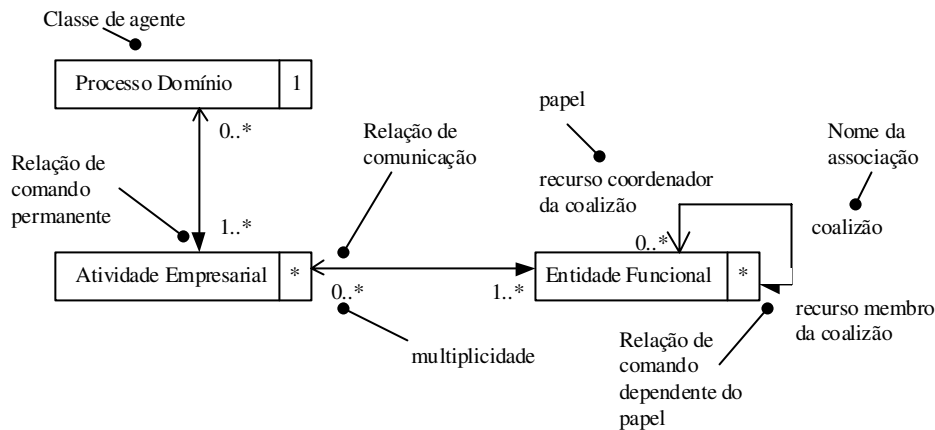


Figura 3 – Diagrama da Sociedade

4.2. O Modelo de Funções

O Modelo de Funções tem a finalidade de relacionar as atribuições inerentes a cada papel que um agente pode assumir, associando seus respectivos serviços e restrições. Cada atribuição pode ter uma ou mais restrições a serem observadas, e um ou mais serviços identificados através de atos de comunicação, definindo a *interface* do agente. Na Figura 4 é apresentada a Tabela de Papéis parcial para o agente *Entidade Funcional* envolvendo os papéis *recurso membro da coalizão* e *recurso coordenador da coalizão*.

4.3. Diagrama de Interação Seqüencial

O Diagrama de Interação Seqüencial permite descrever as interações entre os agentes participantes de um determinado caso de uso. Os agentes são representados por retângulos apresentando o nome do agente, quantidade de instância envolvidas na interação, nome da coalizão e papel do agente. As mensagens trocadas entre os agentes podem ser enviadas para um agente específico da classe (*unicast*), ao conjunto de agentes pertencentes a coalizão (*multicast*) ou para todos os agentes da classe (*broadcast*). As setas representam a direção da mensagem que é expressada por um ato de comunicação em ACL. Uma mensagem *unicast* é representada por uma linha cheia, uma mensagem *broadcast* por uma linha cheia em negrito e uma mensagem *multicast* por uma linha tracejada em negrito.

Classe de Agente: Entidade Funcional					
Papal 1: Recurso Membro da Coalizão			Papal 2: Recurso Coordenador da Coalizão		
Atribuições	Restrições	Serviços	Atribuições	Restrições	Serviços
Elaborar e apresentar uma proposta individual solicitada por um agente coordenador de coalizão.	- Manter os compromissos previamente assumidos; - Observar o prazo de submissão de proposta individual a um agente coordenador.	(cfp: content (convite (refer, operId, intervaloTempo, dataLimite)))	Elaborar proposta de coalizão em atenção a uma chamada de propostas recebida de um agente Atividade Empresarial.	Observar o prazo de submissão de proposta de coalizão.	(cfp: content (convite (refer, operId, intervaloTempo, dataLimite))) (inform: content (proposta-individual (refer, preco, tempoExecucao)))
Confirmar sua proposta individual a um agente coordenador de coalizão para fechamento de contrato.	- Manter os compromissos previamente assumidos.	(query-if: content (requer-confirmação-proposta(idProposta)))	Fechar contrato com um agente Atividade Empresarial.	Assegurar que todos os agentes membros da coalizão confirmaram suas propostas individuais.	(accept-proposal: content (aceitação(motivo)))

Figura 4 - Modelo de Funções

Na Figura 5 é apresentado um diagrama de interação seqüencial para o caso de uso *Realizar chamada de propostas para execução de atividades empresariais*. O agente *a:Atividade Empresarial* envia uma mensagem *multicast* para os agentes *Entidade Funcional* que exercem o papel de *recurso candidato* e que formam a coalizão chamada *equipe de produção*.

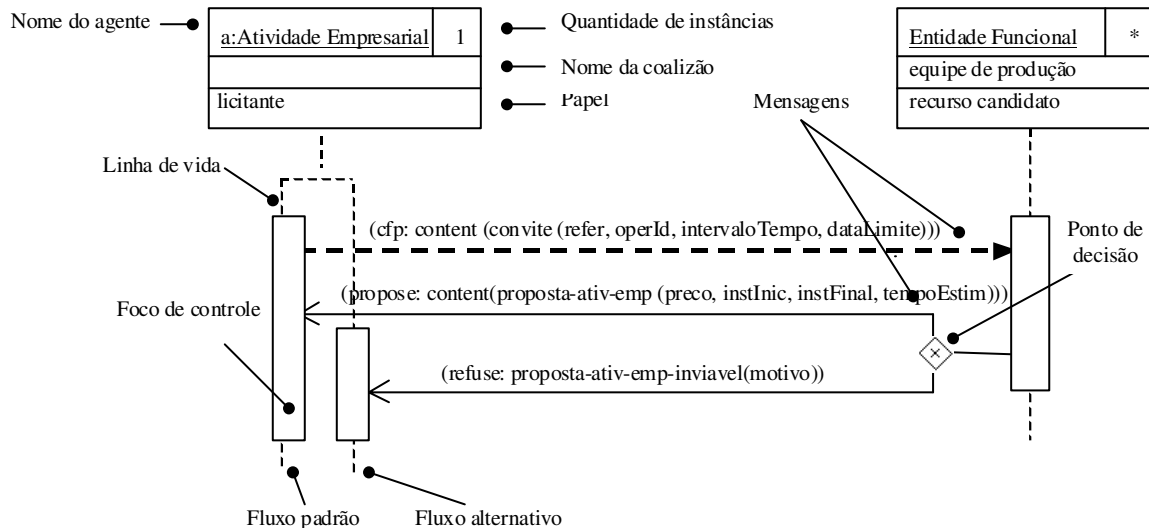


Figura 5 – Diagrama de Interação Seqüencial

Uma linha tracejada vertical representa a linha de vida de um agente e uma barra o seu foco de controle determinando sua participação na interação. O fluxo padrão define o cenário normal para o caso de uso e os fluxos alternativos suas variantes. Um fluxo alternativo ocorre como consequência de um ponto de decisão, onde a resposta do agente receptor da mensagem estabelece a necessidade de mudança de plano pelo agente requisitante.

5. O Modelo de Projeto

Conforme a abordagem proposta neste trabalho, a elaboração do modelo do projeto de um SMA implica no refinamento do modelo de análise, o que se traduz pelo atendimento das propriedades e características de um sistema orientado a agentes. Nesta seção são apresentados os requisitos a serem considerados para a elaboração de um modelo de projeto orientado a agentes, tendo como referência as características dos sistemas de informação tipicamente utilizados para atendimento aos processos do negócio de organizações empresariais. Visando o atendimento a estes requisitos, é proposta a utilização de agentes de infra-estrutura, cujas atribuições estão relacionadas a atividades especializadas de apoio aos agentes do domínio ou aos usuários do sistema. Também são examinados os refinamentos a serem realizados sobre os artefatos produzidos no modelo de análise em direção a construção do modelo de projeto.

5.1. Agentes de Infra-Estrutura

O modelo de projeto de um SMA requer o atendimento às propriedades e características deste tipo de sistema. Para tanto, existem alguns aspectos fundamentais a serem considerados para o adequado funcionamento da sociedade e satisfação de suas funcionalidades, requerendo um conjunto de serviços especializados. Estes serviços caracterizam-se por envolverem apoio aos agentes do domínio para a execução de suas atribuições, tal como acesso a informações existentes no banco de dados corporativo, ou ainda necessidades específicas dos usuários do sistema, tal como uma consulta que requeira consolidação de informações dispersas entre diversos agentes. Tais agentes, denominados agentes de infra-estrutura, devem ser incorporados a sociedade consolidando o modelo de projeto. Os agentes de infra-estrutura são representados no modelo de projeto através de estereótipos. Detalhes envolvendo a especificação destes agentes e sua arquitetura estão disponíveis em [21]. Na seqüência são examinados os requisitos básicos a serem observados na construção do modelo de projeto de um SMA e definido o perfil dos agentes de infra-estrutura necessários para atendê-los.

Ambientes Distribuídos. A descentralização do controle torna um SMA um ambiente intrinsecamente distribuído. Sendo assim, a comunicação entre os agentes torna-se a única via de coordenação para as ações que envolvam cooperação, negociação e planejamento distribuído.

Na medida em que agentes podem ingressar ou sair do sistema em tempo de execução em diferentes nodos da rede, faz-se necessária a existência de mecanismos de localização para permitir a comunicação entre os mesmos. Uma forma de prover a interoperabilidade entre os agentes de um SMA distribuídos entre os nodos de uma rede física é instanciar um **agente roteador** em cada nodo de modo a compor uma organização federada. Cada federação corresponde a um nodo físico da rede, onde apenas o agente roteador tem acesso aos canais de comunicação.

Nesta organização federada, os agentes localizados em diferentes nodos não comunicam-se diretamente uns com os outros. Cada agente comunica-se com seu roteador local que, por sua vez comunica-se com outros roteadores. Desta forma, cada grupo de agentes é ligado a um único roteador compondo uma federação, onde um agente rende parte de sua autonomia ao roteador, ou seja, deixa a cargo do roteador várias funções de comunicação. Agentes roteadores são identificados pelo estereótipo <<router>>. O roteador deve estar ligado a todos os demais agentes, sendo portanto redundante representá-lo nos diagramas da sociedade e de interação seqüencial.

Integração com o Sistema Corporativo. Considerando um SMA como parte componente do conjunto de sistemas de informação de uma organização, existe a necessidade de que alguns agentes tenham acesso a informações armazenadas no banco de dados corporativo. Partindo do pressuposto de que os agentes do domínio devem encapsular apenas capacidades coerentes com suas atribuições dentro da sociedade, faz-se necessária a criação de mecanismos de apoio que facilitem a sua interação com este banco de dados.

Para realizar esta integração é definido o **agente mediador de acesso à base de dados**, identificado pelo estereótipo «database_handler». Este agente disponibiliza uma interface contendo um conjunto mínimo de serviços para acesso à operações de consulta e atualização de dados que podem ser requisitados pelos agentes do domínio.

Este agente deve ser integrado aos diagramas da sociedade e de interação seqüencial. No diagrama da sociedade representa-se todos os canais de comunicação existentes entre este agente e os demais. No diagrama de interação seqüencial são representadas todas as solicitações de serviço enviadas ao «database_handler» e as possíveis alternativas de resposta.

Intervenção Humana no SMA. Embora a interferência humana seja minimizada em soluções utilizando SMA, existem situações cuja intervenção faz-se necessária, em particular para tomada de decisões envolvendo conflitos entre os agentes. Desta forma, surge a necessidade de tratamento para estas intervenções, sendo necessária a definição de formas de interface entre agentes e humanos.

Para atender a este aspecto é definido o **agente de interface para agentes humanos** identificado pelo estereótipo «human_interface». Este tipo de agente é responsável por prover aos usuários do sistema uma interface que lhes permita interagir com o SMA, receber notificações ou mesmo acompanhar o funcionamento do SMA. A interação dos usuários com o SMA se faz necessária, por exemplo, quando as capacidades dos agentes do domínio não forem suficientes para resolver algum problema, sendo necessária a tomada de alguma decisão externa à sociedade.

Este agente deve ser integrado aos agentes do domínio no diagrama da sociedade refletindo os diagramas de interação seqüencial cujos cenários envolvam a necessidade de intervenção humana.

Consolidação de Resultados Parciais. Em um SMA, muitas das informações parciais ou específicas decorrentes da atuação dos agentes está dispersa na sociedade. Cada um dos agentes envolvidos na resolução do problema possui um conjunto de resultados parciais de seu próprio processamento. Neste caso, para que os resultados do sistema possam ser visualizados em sua completude, é necessária a criação de condições para a consolidação das informações dispersas entre os vários agentes de acordo com as necessidades dos usuários.

O agente **extrator de relatórios**, identificado pelo estereótipo «reporter», é definido tendo como objetivo consolidar um determinado conjunto de informações recuperadas de um ou mais agentes dispersos na sociedade.

Este agente deve ser integrado aos diagramas da sociedade e de interação seqüencial, relacionando-se com os agentes do domínio que possuam informações a serem consolidadas ou que dele necessitem de algum tipo de serviço.

Mecanismos de Depuração. Uma vez implementado um SMA, faz-se necessário que sejam disponibilizados mecanismos para depuração do funcionamento da sociedade. Esses mecanismos devem fornecer informações sobre os estados locais de cada agente como forma de permitir o acompanhamento do sistema em tempo real. O conhecimento dos resultados parciais derivados das interações entre os agentes, o acompanhamento das ocorrências em tempo real, a possibilidade de verificar o histórico de execução global do sistema e de seus agentes representam informações essenciais para avaliação do funcionamento e desempenho do SMA.

O agente Depurador, identificado pelo estereótipo «debugger», tem como objetivo permitir ao desenvolvedor depurar o funcionamento de algumas situações críticas do sistema. Um agente «debugger» deve ser associado a um ou mais agentes do domínio nos diagramas de interação seqüencial cujos cenários devam ser monitorados. As relações entre os agentes do domínio e o «debugger» definidas nos diagramas de interação seqüencial também devem ser refletidas no diagrama da sociedade.

5.2. Elaboração do Modelo de Projeto

Os agentes de infra-estrutura interagem com os agentes do domínio e com outros agentes de infra-estrutura. Sendo assim, é necessário determinar a maneira pela qual estas relações são estabelecidas dentro do modelo de projeto do SMA. Esta rede de interações define a arquitetura do SMA, consolidando o modelo de projeto.

Seguindo a proposta de refinamento dos diagramas produzidos para a representação dos modelos do sistema, a integração dos agentes de infra-estrutura implica na redefinição dos diagramas produzidos no modelo de análise. Para o refinamento destes diagramas os seguintes procedimentos devem ser observados:

- a) **Adaptação dos Diagramas de Interação Seqüencial:** tendo como entrada os diagramas de interação seqüencial do modelo de análise, procede-se a inclusão dos agentes de infra-estrutura «database_handler», «human_interface» e «reporter» onde se fizerem necessários, estabelecendo-se as interações que ocorrem entre estes e os agentes do domínio. Para ilustrar este procedimento, na Figura 7 é apresentado o diagrama de interação seqüencial para o caso de uso *Fechar contrato* (Figura 1), onde aparece o agente «database_handler» *Dbhprodução* recebendo uma solicitação de inclusão no banco de dados *ProgramacaoProducao* de uma atividade empresarial programada para ser executada em um determinado intervalo de tempo.

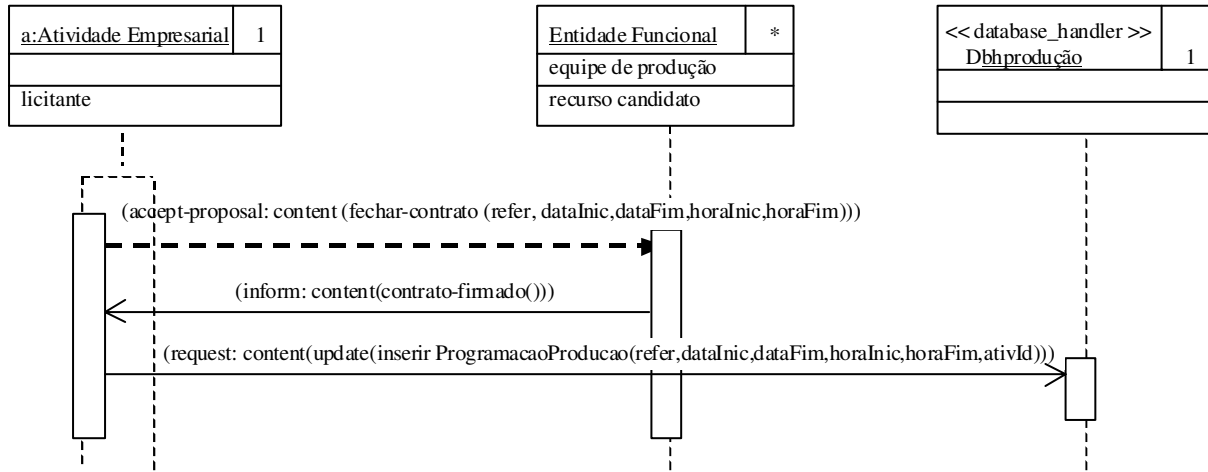


Figura 7 – Diagrama de Interação Sequencial do Modelo de Projeto

b) Atualização do Diagrama da Sociedade: o diagrama da sociedade é atualizado pela inclusão dos agentes de infra-estrutura, conforme a configuração resultante dos diagramas de interação sequencial refinados na atividade anterior. Na Figura 8 é apresentado o diagrama da sociedade do modelo de projetos parcial para o sistema M-DRAP.

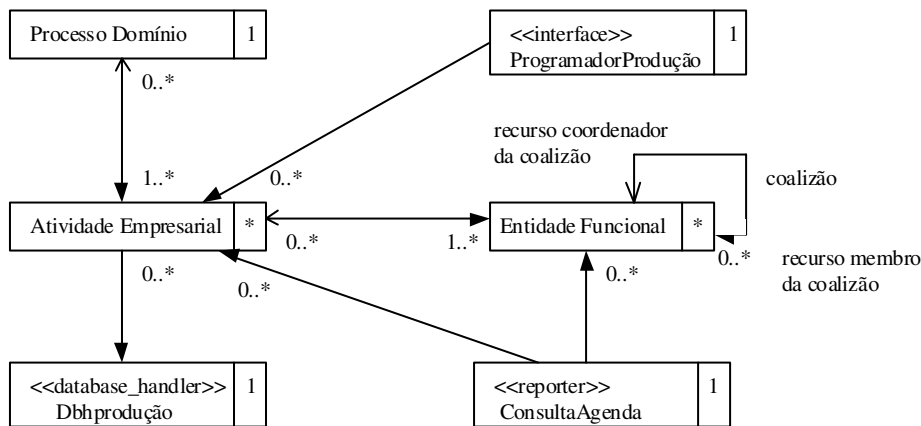


Figura 8 – Diagrama da Sociedade do Modelo de Projeto

c) Criação do Diagrama de Implantação: este diagrama é criado baseado no diagrama da sociedade pelo agrupamento dos agentes em nodos físicos configurando o ambiente de processamento do sistema. Em cada um destes nodos é adicionada a representação do agente <<router>> correspondente. O diagrama de implantação para SMA é definido com uma extensão do Diagrama de Implantação da UML, diferindo de seu formato original por permitir a representação da localização física dos agentes dentro da rede de nodos, bem como os canais de comunicação existentes para interação dos agentes. Também tem-se a representação do respectivo roteador no nodo <<router>>. Na Figura 9 é apresentado um exemplo de diagrama de implantação envolvendo dois

nodos (servidor de banco de dados e servidor de agentes) comportando os agentes do diagrama da sociedade apresentado na Figura 8. Observe a presença dos agentes <<router>> no canto superior direito de cada um dos nodos.

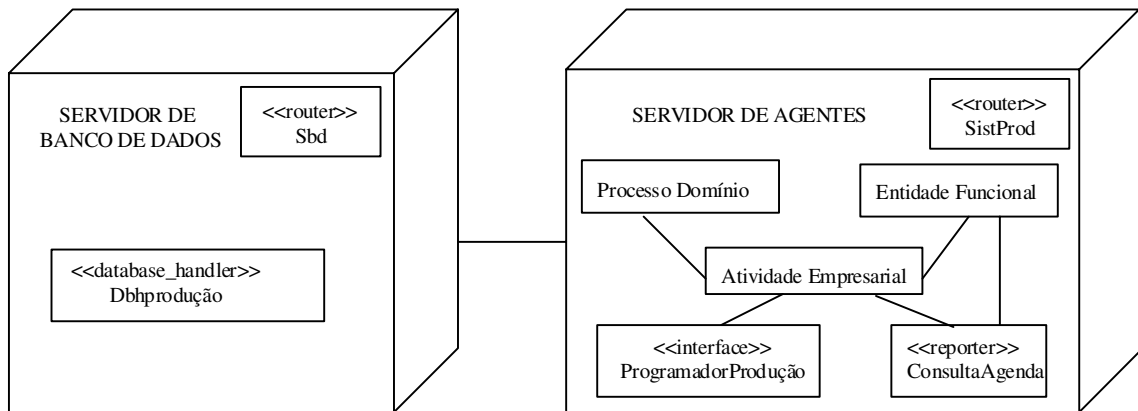


Figura 9 – Diagrama de Implantação

d) Integração dos Agentes de Depuração: os agentes <<debugger>> são incluídos nos pontos críticos do SMA, relacionando-os aos agentes a serem monitorados nos respectivos diagramas de interação seqüencial. Também devem ser atualizados o diagrama da sociedade e o diagrama de implantação. Na Figura 10 é apresentado o diagrama de interação seqüencial para o caso de uso *Realizar chamada de propostas para execução de atividades empresariais* (vide Figura 5) com a inclusão do agente <<debugger>> que permite o acompanhamento das propostas encaminhadas por um agente *Entidade Funcional* pertencente a coalizão *equipe de produção* ao agente *a:Atividade Empresarial*.

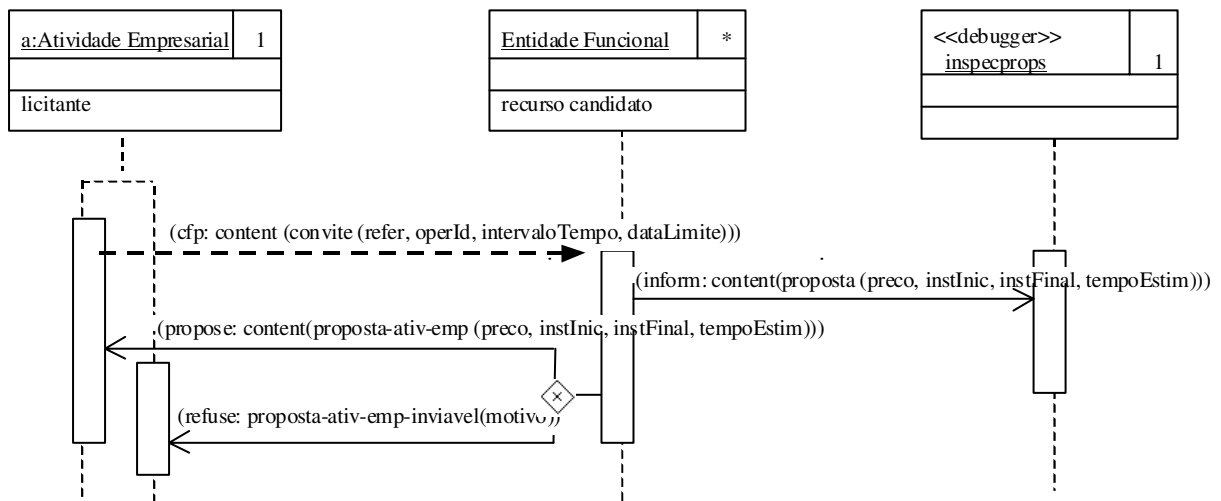


Figura 10 – Diagrama de Interação Seqüencial do Modelo de Projeto

6. Trabalhos Relacionados

Entre os trabalhos mais relevantes sobre linguagem de modelagem, em particular pela repercussão que vem tendo na comunidade científica, destaca-se a extensão da UML para SMA intitulada AUML – *Agent Unified Modeling Language* [7], [11], [12]. No presente trabalho, optou-se por utilizar extensões à UML propostas por [3] por estas englobarem características adicionais as propostas na AUML, permitindo maior riqueza na representação dos modelos. Como contribuição ao esforço de adaptar a UML para SMA é apresentada uma extensão ao diagrama de implantação para a configuração do modelo de projeto.

Em [10] é apresentada uma metodologia para o desenvolvimento de SMA intitulada MaSE. Para a construção dos modelo de análise e projeto nesta metodologia, os autores utilizam diagramas essencialmente baseados em UML com pequenas adaptações. Outro trabalho relevante é apresentado em [8], onde é definida a metodologia intitulada Gaia. Ambos os trabalhos detalham os procedimentos metodológicos a serem seguidos para a construção de um SMA, com ênfase na modelagem a nível macro do sistema. Porém, no que se refere ao modelo de projeto são tratados basicamente aspectos relativos ao modelo individual dos agentes e seus papéis e serviços, e a estrutura do sistema envolvendo o relacionamento entre os agentes componentes da sociedade.

Outros trabalhos relevantes são [4] [6] [9], que a exemplo dos trabalhos acima considerados, não examinam os aspectos ambientais e de serviços a serem especificados na modelagem do projeto de um SMA.

Na proposta apresentada em [5], os autores apresentam uma abordagem baseada na identificação de atores para o preenchimentos dos requisitos de um sistema. Aos atores são associados objetivos e atividades caracterizando suas responsabilidades (similar ao conceito de papéis adotado neste trabalho). Nesta abordagem, como resultado do processo de modelagem do sistema, também são identificados atores agentes responsáveis por atividades especializadas relacionadas a infra-estrutura do ambiente de processamento, corroborando com as proposições apresentadas no presente trabalho.

7. Conclusão e Trabalhos Futuros

O conceito de SMA teve sua origem na área de inteligência artificial, e apenas recentemente tem sido objeto de pesquisas em engenharia de software. Embora do ponto de vista da inteligência artificial já exista uma considerável evolução conceitual envolvendo o tema, do ponto de vista da engenharia de software ainda há muito esforço a ser desenvolvido em direção a consolidação de técnicas e metodologias a serem aplicadas neste tipo de sistema, sendo ainda inexpressivo o volume de trabalhos publicados. Alguns destes trabalhos, a exemplo do apresentado neste artigo, tem proposto extensões da UML para a modelagem de MAS. As abordagens metodológicas propostas na literatura também acabam por buscar subsídios na orientação a objetos, sem, no entanto, considerar plenamente as diferenças essenciais existentes entre sistemas orientados a objetos e sistemas orientados a agentes.

Este trabalho buscou contribuir para o preenchimento desta lacuna, propondo uma abordagem ao problema fundamentada nas propriedades e características inerentes aos SMA. Neste sentido, apresentou uma abordagem para a modelagem de SMA, tendo como foco o atendimento aos requisitos de projeto essenciais aos sistemas de informação tipicamente utilizados para atendimento aos processos do negócio de organizações empresariais. O processo de modelagem proposto é baseado no refinamento entre modelos de análise e projeto, sendo utilizado o modelo de casos de uso como referência na definição dos requisitos

do sistema. Os modelos são construídos através de extensões da UML capazes de refletir as propriedades e características dos SMA.

Para o atendimento aos serviços de infra-estrutura necessários para o processamento do sistema são propostos agentes especializados denominados agentes de infra-estrutura. Tal proposição está fundamentada na premissa de que os agentes devem ter elevado grau de especialização em uma sociedade de agentes. Sendo assim, aos agentes do domínio do SMA devem ser definidas atribuições restritas aos papéis que lhes foram destinados no modelo de casos de uso, cabendo ao conjunto de agentes de infra-estrutura cobrir os demais serviços necessários para o processamento do sistema.

Os resultados apresentados neste trabalho representam contribuições iniciais do projeto MASP – *Multi-Agent Systems development*. Atualmente estão sendo realizados estudos no sentido de definir uma metodologia para a modelagem a nível macro de SMA compatível com os modelos apresentados neste trabalho. Também estão sendo realizados esforços para a construção de um conjunto de agentes de infra-estrutura capazes de serem utilizados em diferentes aplicações envolvendo agentes. Paralelamente, utilizando o desenvolvimento destes agentes como estudo de caso, também vem sendo realizados esforços no sentido da consolidação de uma proposta para modelagem de agentes (nível micro), contemplando assim todo o processo de desenvolvimento de SMA.

8. Bibliografia

1. Wooldridge, M.: Agent-based Software Engineering. In: IEE Proceedings on Software Engineering, 144(1), (1997) 26-37
2. Jennings, N.R. et ali.: Using intelligent agents to manage business processes. In Proceedings of Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96), London, UK, 1996
3. Sousa, J.B.: Em direção a uma abordagem para a modelagem de um sistema multiagente. (Dissertação de Mestrado). Porto Alegre, RS, Brasil, 2001
4. Yim, H., Cho, K., Kim, J., Park, S.: Architecture-Centric Object-oriented Design Method for Multi-Agent Systems. In: Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000), Boston, Massachusetts, USA, (2000) 469-470
5. Castro, J.F.B., Kolp, M., Mylopoulos, J.: Developing Agent-Oriented Information Systems for the Enterprise. In: Proceedings of Third International Conference on Enterprise Information Systems, Stafford: Staffordshire University, 2000 1-16
6. Caire, G. et al.: Agent Oriented Analysis using MESSAGE/UML. In: Ciancarini, P., Wooldridge, M. (eds): Agent-Oriented Software Engineering. Lecture Notes in Computer Science. Springer-Verlag, Berlin, (2001)
7. Odell, J., Parunak, H.V.D. Bauer, B.: Extending UML for Agents. In: Wagner, G., Lesperance, Y., Yu, E. (eds): Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence, Austin, TX, (2000) 3-17
8. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems, vol. 3, (2000) 285-312
9. Zambonelli, F., M., Jennings, N.R., Wooldridge, M.: Organisational rules as an abstraction for the analysis and design of multi-agent systems. Int J. of Software Engineering and Knowledge Engineering 11 (3), (2001) 303-328

10. DeLoach, S.A: Multiagent Systems Engineering. *International Journal of Software and Knowledge Engineering*, vol. 11, no. 3, (2001) 231-258
11. Bauer, B., Müller, J.P., Odell, J.: Agent UML: A Formalism for Specifying Multiagent Interaction. In: Ciancarini, P., Wooldridge, M. (eds): *Agent-Oriented Software Engineering. Lecture Notes in Computer Science*. Springer-Verlag, Berlin, (2001) 91-103
12. Bauer, B.: UML Class Diagrams: Revisited in the Context of Agent-Based Systems. In: Ciancarini, P., Wooldridge, M. (eds): *Agent-Oriented Software Engineering. Lecture Notes in Computer Science*. Springer-Verlag, Berlin, (2001) 1-8
13. Bastos, R.M., Oliveira, J.P.M.: A conceptual modeling framework for multi-agent. In: Laender, A.H.F., Liddle, S.W., Storey, V.C. (eds): *Conceptual Modeling - ER 2000. Lecture Notes in Computer Science*, Vol. 1920. Springer-Verlag, Berlin, (2000) 296-308
14. Jacobson, I., Booch, G. Rumbaugh, J.: *The unified software development process*. Addison-Wesley, 1998
15. Wooldridge, M., Ciancarini, P.: Agent-Oriented Software Engineering: The State of the Art. In: Ciancarini, P., Wooldridge, M. (eds): *Agent-Oriented Software Engineering. Lecture Notes in AI*, Vol. 1957. Springer-Verlag, (2001) 207-221
16. Jennings, N. R., Wooldridge, M.: Agent-Oriented Software Engineering. In: Bradshaw, J. (ed.) *Handbook of Agent Technology*, AAAI/MIT Press, 2000.
17. Iglesias, C. A., Garijo, M., Gonzalez, J. C.: A Survey of Agent-Oriented Methodologies, In Singh, M.P., Rao, A.S. & Wooldridge, M.J. (eds.) *Intelligent Agents IV*, Springer, (1997) 313-327
18. Booch, G. Rumbaugh, J., Jacobson, I.: *The Unified Modeling Language User Guide*. Addison-Wesley, 1998
19. Bastos, R.M., Oliveira, J. P.M., Oliveira, F.M.: Decentralised Resource Allocation Planning through Negotiation, in Camariha-Matos, L.M., Afsarmanesh, H. & Marik, V. (eds.) *Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organization*. Kluwer Academic Publishers, (1998) 67-76
20. FIPA Communicative Act Library Specification. Disponível em <http://www.fipa.org/specs/fipa00037/>
21. Juchem. M.: Projeto de sistemas multiagentes em organizações empresariais. (Dissertação de Mestrado). Porto Alegre, RS, Brasil, 2001