

Metodologias para desenvolvimento de sistemas multiagentes: Visão geral e comparação

*Silvana Rossy de Brito¹, Tânia Barbosa Salles Gava²,
Orivaldo de Lira Tavares², Crediné Silva de Menezes²*

¹ FAESA - Faculdades Integradas Espírito-santenses
Colegiado de Ciência da Computação
Rua Anselmo Serrat, 199, Ilha de Monte Belo, Vitória-ES, Brasil

² UFES - Universidade Federal do Espírito Santo
Centro Tecnológico - Mestrado em Informática
Av. Fernando Ferrari, s/n, Campus Universitário de Goiabeiras, CT-VII, Vitória-ES, Brasil

srossy@faesa.br, t_salles@zaz.com.br, {tavares, credine}@inf.ufes.br

Abstract. *This paper discusses the necessity of specific methodology for the multiagent systems development, where the considered models supply more natural forms of system modeling, design and implementation. For this, it shows an overview on six methodological approaches, presenting them through some evaluation criteria.*

Resumo. *Este artigo discute a necessidade de uma metodologia que seja específica para o desenvolvimento de sistemas baseados em agentes, onde os modelos propostos forneçam formas naturais de modelagem, projeto e implementação de sistemas dessa natureza. Para isso, mostra uma visão geral sobre seis abordagens metodológicas, apresentando-as através de alguns critérios de avaliação.*

1 Introdução

Um ponto considerado fundamental para sucesso de um projeto é a seleção de uma metodologia apropriada. O termo metodologia, entretanto, apesar de ser amplamente utilizado, não possui uma definição amplamente aceita. De um modo geral, entende-se como metodologia uma série recomendada de passos e procedimentos, que compõem os métodos que devem ser seguidos durante o processo de concepção do software.

Este artigo discute algumas metodologias desenvolvidas especificamente para o desenvolvimento de sistemas baseados em agentes, buscando fazer considerações e comparações acerca das vantagens em se adotar tais metodologias. Uma metodologia projetada para a construção de sistemas multiagentes deve capturar a flexibilidade, autonomia e o poder que a abstração de agentes incorpora ao projeto e desenvolvimento de sistemas complexos. A seção 2 apresenta alguns critérios para análise e seleção de metodologias, para, em seguida, apresentar algumas abordagens metodológicas consideradas promissoras; a seção 3 contém uma comparação entre as metodologias apresentadas e, finalmente, a seção 4 apresenta as considerações finais desse trabalho.

2 Metodologias para desenvolvimento de sistemas baseados em agentes

A escolha do método ou métodos empregados como parte da metodologia não é a mesma para todos os projetos, pois são vários os fatores que determinam a escolha por

uma metodologia. Dentre esses fatores estão: experiência dos desenvolvedores, tamanho e complexidade do problema, tempo e disponibilidade de recursos, documentação disponível, facilidade de compreensão dos métodos propostos, disponibilidade de exemplos e de ferramentas que suportem a metodologia. Fundamentalmente, é importante que os métodos propostos suportem as bases do paradigma utilizado. No paradigma orientado a agentes, a metodologia deve suportar os conceitos de autonomia, reatividade, iniciativa, racionalidade. Da mesma forma, no paradigma de orientação a objetos, os métodos devem suportar abstração, encapsulamento, herança e polimorfismo.

Uma onda de interesse tem surgido em torno das metodologias que suportam o paradigma de agentes, como resultado da utilização desse paradigma em diversas áreas do conhecimento. Segundo Wooldridge & Ciancarini [13], as razões para esse interesse são várias, mas a mais importante se deve, certamente, ao conceito de um agente como um sistema autônomo, capaz de interagir com outros sistemas para satisfazer seus objetivos, bem como ser uma forma natural para se projetar software. Da mesma forma como se pode compreender um sistema como sendo composto de objetos passivos, que possuem estados e sobre os quais executam-se operações, também se pode compreender outros sistemas como sendo agentes (semi) autônomos que interagem para atingir um objetivo comum.

Muitas das abordagens para se modelar sistemas baseados em agentes utilizam técnicas orientadas a objetos, estendendo ou adaptando os modelos conhecidos de forma que a própria aplicabilidade das metodologias e das técnicas orientadas a objetos possa ser estendida. Outras abordagens propõem novos modelos, com linguagem e notação diferenciadas a para representação desses modelos.

As metodologias orientadas a agentes (OA) devem atender questões específicas do paradigma de agentes. Nesse sentido, Iglesias e Garijo [8] sugerem que se levante algumas questões fundamentais durante a definição de uma metodologia para sistemas multiagentes, tais como: o que é um agente?; quando a tecnologia de agente pode ser aplicada ou quando outras tecnologias são melhores opções?; quais são as principais diferenças entre agentes e objetos e como as metodologias orientadas a agentes podem se beneficiar das metodologias orientadas a objetos?; como as metodologias OA se beneficiam do conhecimento de “comunidade”? Uma metodologia também deve definir uma linguagem de modelagem padrão para análise e projeto, orientar o processo, possuir notação e métodos para conceituar o sistema.

Uma metodologia orientada a agentes deve, ainda suportar as propriedades dos agentes nos modelos e notação utilizada. Nesse sentido, ao avaliar uma metodologia, deve-se questionar sobre a descrição dos agentes, a modelagem das relações sociais entre os agentes, a modelagem das interações homem-máquina, a modelagem dos protocolos de interações entre agentes, a modelagem da reatividade e do processo de negociação entre os agentes, além da continuidade temporal dos agentes e de características pró-ativas. Do ponto de vista de projeto, uma metodologia deve suportar e operacionalizar os modelos de análise de modo a atender diferentes plataformas [8]. Nesse ponto, consideramos que os sistemas baseados em agentes são tipicamente distribuídos e a independência da plataforma é um critério que deve ser analisado em tempo de projeto de arquitetura.

Diante da necessidade de adotar uma metodologia adequada ao problema, faz-se

necessário o estudo de algumas das metodologias consideradas “promissoras” entre as propostas para a construção de sistemas multiagentes. A seguir, apresenta-se algumas dessas abordagens.

2.1 Análise e projeto orientado a agentes de Birgit Burmeister

Esta abordagem usa várias notações da orientação a objetos. Segundo Burmeister [5] o projeto dirigido a responsabilidade (RDD) para o projeto de sistemas orientados a objetos pode ser considerado como bastante próximo aos conceitos da tecnologia de agentes. Apesar de utilizar alguns conceitos e notações da orientação a objetos, o autor acrescenta que devido às diferenças conceituais entre agentes e objetos, os métodos orientados a objetos não são diretamente aplicáveis ao projeto e análise de sistemas orientados a agentes, necessitando, portanto, de extensões aplicáveis aos modelos.

A metodologia de Burmeister utiliza cartões CRC (Classes-Responsabilidades-Colaborações) para descrever os agentes e está baseada na construção dos seguintes modelos: modelo do agente, modelo organizacional e modelo de cooperação.

Compartilhando das idéias de Burmeister, onde as tecnologias orientadas a agentes podem ser modeladas como uma extensão da orientação a objetos, Kinny [10] e outros pesquisadores, propõem uma arquitetura para sistemas multiagentes, apresentada a seguir.

2.2 Metodologia e técnica de modelagem para sistemas multiagentes usando a arquitetura BDI

Para Kinny et al [10] as tecnologias orientadas a agentes são uma extensão natural das técnicas orientadas a objeto, onde os agentes podem ser vistos como objetos ativos. As diferenças entre os objetos e os agentes, são a estruturação do estado interno de um agente para incorporar as noções mentais como crenças, objetivos, intenções, e a caracterização de mensagens por tipos de mensagem e estruturação das mensagens em protocolos. Tais diferenças conceituais justificam que os sistemas orientados a agentes sejam analisados, projetados e implementados diferentemente.

Segundo Kinny et al. [10], esta abordagem, está baseada no modelo de Crenças-Desejos-Intenções (Belief-Desire-Intention - BDI) e estende os modelos e técnicas já exploradas e compreendidas da orientação a objetos. Para os autores, os detalhes essenciais de um projeto de sistema são capturados em três modelos distintos: Modelo de Objeto, Modelo Dinâmico e Modelo Funcional.

Nesta abordagem, os autores adotam um conjunto de modelos em dois níveis de abstração: externo e interno. Do ponto de vista externo o sistema é decomposto em agentes, modelado como objetos complexos caracterizados pelos seus objetivos e responsabilidades, serviços que executam, informações que requerem e mantêm, e as interações externas. Do ponto de vista interno, as crenças, desejos e intenções, requeridos por uma arquitetura de agente particular, devem ser modelados para cada agente. Os sistemas podem ser decompostos em papéis fundamentais. A identificação dos papéis e o relacionamento entre eles direcionam a especificação de uma hierarquia de classe de agente onde os agentes são exemplos particulares dessas classes. A análise das responsabilidades de cada classe de agente conduz a uma identificação dos serviços fornecidos e usados por um agente, e conseqüentemente suas interações externas. Questões como a duração e a criação de papéis e suas interações determinam as relações de controle entre classes de agentes. Esses detalhes são capturados em dois modelos

externos, que são realizados de forma independente da arquitetura BDI [10]: **Modelo de Agente**: descreve a relação hierárquica entre diferentes classes de agentes (abstratas e concretas) e identifica os exemplos de agentes que podem existir dentro do sistema, e a multiplicidade deles, e quando eles passam a existir; **Modelo de Interação**: descreve as responsabilidades de uma classe de agentes, os serviços que fornece, interações associadas, e controla relações entre as classes de agente. Esse modelo inclui a sintaxe e a semântica das mensagens usadas para comunicação entre agentes e entre agentes e outros componentes de sistemas, como interfaces.

Os modelos externos, nessa abordagem, são semelhantes aos modelos de papéis e de interações da metodologia GAIA [15]. Entretanto, outros modelos, chamados de modelos internos, assumem um compromisso com a arquitetura BDI.

O paradigma BDI provê uma noção forte de agência [10]: os agentes são vistos como tendo certas atitudes mentais (crenças, desejos e intenções) que representam, respectivamente, os estados: informacional, motivacional e deliberativo. Um agente pode ser completamente especificado pelos eventos que pode perceber, as ações que pode executar, suas crenças, objetivos e planos que dão origem às suas intenções. Os modelos que representam essas características são: **Modelo de Crenças** (descreve a informação sobre o estado interno do ambiente para o agente daquela classe, indicando se as crenças podem ou não mudar com o passar do tempo); **Modelo de Objetivos** (descreve as metas que um agente pode adotar, e os eventos para os quais pode responder); **Modelo de Planos** (descreve os planos que um agente pode empregar para alcançar suas metas).

2.3 MAS-CommonKADS

Baseando-se na necessidade do desenvolvimento de uma metodologia que suporte todo o ciclo de vida do desenvolvimento do software de um sistema baseado em agentes (incluindo sua gerência), esta abordagem, proposta por Iglesias et al [7], fornece uma plataforma multiagente para diferentes aplicações. A proposta é definir uma metodologia baseada na CommonKADS [1]. As características de integração de agentes são acrescentadas e um modelo para representar a comunicação entre os agentes (descrevendo os protocolos de comunicação) é incorporado à metodologia CommonKADS;

KADS [12] é o resultado de uma pesquisa realizada em um projeto ESPRIT, visando o desenvolvimento de uma metodologia ampla para a construção de Sistemas Baseados em Conhecimento (SBCs). Quando a pesquisa de KADS iniciou, estava focada principalmente na aquisição de conhecimento. CommonKADS [1] é uma versão mais recente da metodologia KADS projetada para o desenvolvimento de SBCs e prevê gerência de projeto, análise organizacional, engenharia de conhecimento e engenharia de software para esses sistemas. Em CommonKADS, o desenvolvimento de um SBC é visto como a construção de modelos de comportamento de resolução de problema em seu contexto organizacional concreto. CommonKADS provê um conjunto de modelos, que permite expressar as várias perspectivas na situação de resolução de problema. Os modelos de organização, tarefa, competência e comunicação capturam o contexto da atividade de resolução de problema de interesse para o projeto de SBC. O modelo de especialidade descreve o conhecimento e o raciocínio envolvidos na execução das tarefas. Finalmente, o modelo de projeto descreve a sua realização computacional.

A abordagem MAS-CommonKADS [7] acrescenta as características de integração de agentes e propõe um novo modelo de comunicação entre os agentes (modelo de coordenação) – CoM. O modelo CoM é usado em conjunto com CommonKADS para descrever protocolos de comunicação para sistemas multiagentes. A metodologia é definida baseando-se no refinamento da metodologia já existente. MAS- CommonKADS é uma extensão da metodologia de CommonKADS, adicionando os aspectos relevantes a sistemas multiagentes.

No conjunto de modelos do CommonKADS, Iglesias *et al* [7] observaram algumas limitações para sistemas multiagentes: O Modelo de Comunicações (CM) trata na maior parte da interação homem-máquina, sendo muito restrito para a interação computador-computador; na abordagem CommonKADS, a atribuição de tarefas deve ser executada de uma forma fixa. O CM não trata transações entre vários parceiros de uma forma natural. Para atender a essas limitações, Iglesias *et al* [7] propõem uma extensão de CommonKADS, criando um novo modelo (coordenação) para representar a interação entre agentes ou ainda para modelar a interação homem-computador.

Na abordagem para sistemas multiagentes, o modelo de agentes foi modificado para incluir características inteligentes nos agentes. O elemento central é o agente que possui 5 atributos: endereço, tipo (se humano ou não), subclasse-de, papel, posição e grupos. Outros constituintes desse modelo são: serviços (oferecidos pelos agentes), metas (objetivos dos agentes), capacidade de raciocínio (requisitos especialistas), capacidades gerais (sensores e atuadores para manipular o ambiente e linguagens de representação do conhecimento e de comunicação entre os agentes), e restrições (normas, preferências e permissões).

Além da modificação no modelo de agentes, um novo modelo foi desenvolvido já que o CM não fornece um tratamento confiável para as interações entre agentes. O Modelo de Coordenação (CoM) é usado quando os agentes são agentes de software. Assim, a metodologia deixa o CM para ser usado na representação da interação entre agentes humanos e agentes de software. Para o CoM, os agentes humanos são *agentificados* (transformados em agentes artificiais) e a comunicação é feita via ações de discurso.

O modelo de coordenação contém: a conversação (conjunto de interações representando as solicitações de serviço), a interação (a troca de mensagens), as habilidades (as bases e o conhecimento do iniciador da conversa e de outros participantes) e o protocolo (um conjunto de regras que governam a conversa definindo estados e interações permitidos).

A validação cruzada entre os modelos é feita através da identificação e descrição dos serviços (conversas) entre os agentes e da identificação dos protocolos de negociação necessários.

2.4 Metodologia GAIA

Essa metodologia é fundamentada na visão de um sistema como uma organização computacional que consiste de vários papéis interagindo e é aplicável a uma grande variedade de sistemas, tratando dos aspectos nos níveis macro (de sociedade) e micro do agente.

O primeiro artigo que apresentou a metodologia foi publicado em 1999 [14], mas foi somente em 2000 que foi nomeada como GAIA [15], apresentando poucas

mudanças da proposta de 1999. A abstração principal nos modelos de análise que adota é a abstração de papéis.

Os principais conceitos desta metodologia podem ser divididos em duas categorias: abstratos e concretos. A entidade mais abstrata na hierarquia de conceitos da metodologia é o sistema. Segundo Wooldridge et al [15], um sistema baseado em agentes pode ser considerado uma “sociedade” ou “organização artificial”.

Semelhante a abordagem de Burmeister [5], esta proposta baseia-se no conceito de papéis sendo executados dentro de uma organização. Um papel é definido através de três atributos: responsabilidades, permissões, e protocolos. A influência da modelagem de papéis, apesar de estar diretamente relacionada com a análise e especificação de requisitos, se estende à fase de projeto no momento em que um agente executa vários papéis, e assim, durante o projeto, os papéis previamente identificados serão compostos em agentes e só então a funcionalidade associada com cada papel pode ser implementada.

Em [2], [3] e [4], Brito *et al* apresentam a modelagem de três sistemas baseados em agentes, modelados através da metodologia GAIA. Outros sistemas podem ser encontrados em <http://www.inf.ufes.br/~tavares/dsba>.

A metodologia GAIA, apesar das vantagens que incorpora ao processo de desenvolvimento, apresenta algumas limitações. Dentre essas, estão [15]: limitação quanto a quantidade de agentes (os autores sugerem que não ultrapasse de 100); não trata objetivos conflitantes entre os agentes; a habilidade social e as atividades desempenhadas pelos agentes não se modificam em tempo de execução.

2.5 Modelagem via *workflow*

Nessa abordagem, o processo empresarial é visto como uma coleção de agentes autônomos que interagem entre si quando possuem interdependências [16]. Um *workflow* é modelado como um conjunto de papéis relacionados, definidos em termos de objetivos, qualificações, obrigações, permissões, protocolos, etc. Os papéis são definidos em função de suas qualificações e suas capacidades. Uma vez que um papel é nomeado a um agente, o agente herda as obrigações e permissões especificadas daquele papel. A coordenação do *workflow* é alcançada através da comunicação entre agentes.

Os autores dessa abordagem comparam os sistemas multiagentes com os sistemas baseados em *workflow*: um sistema multiagente pode ser definido como uma rede de resolvidores de problemas. Os problemas resolvidos estão além das capacidades individuais ou conhecimento de cada resolvidor, daí a necessidade de cooperarem entre si, de forma coordenada.

Os autores definem *workflow* como um conjunto de agentes autônomos resolvidores de problemas, que interagem entre si e possuem interdependências. A modelagem de um *workflow* é feita segundo papéis que se relacionam. Um papel se refere aos padrões de comportamento esperados que um agente tem que executar. Possui as seguintes características essenciais: metas, qualificações, relações, obrigações, permissões, restrições, protocolos, etc. As interações entre agentes são governadas por atos de fala fundamentados em protocolos. A coordenação do *workflow* é alcançada pela comunicação entre os agentes.

Essa proposta está dividida em três fases [16]: análise baseada em papel

(determinação dos papéis e identificação de relacionamento entre eles), projeto orientado a agentes (onde se identifica os tipos de agentes) e implementação orientada a agentes (implementação de um *workflow* orientado a agentes).

2.6 Metodologia ZEUS

Esta metodologia, proposta por Collis e Nudmu [6] é direcionada para a construção de agentes utilizando-se a ferramenta ZEUS [11]. Zeus é um ambiente para análise, projeto e construção de agentes. A metodologia é orientada a papéis e pode ser aplicada a diferentes tipos de problemas.

Na análise do domínio, a aplicação é vista como um sistema organizacional, onde os papéis trabalham para realizar suas responsabilidades, assim como na metodologia GAIA [15]. A grande vantagem dessa metodologia em relação a metodologia GAIA é o suporte da ferramenta geradora de agentes, Ferramenta Geradora de Agentes, que permite a geração de partes dos agentes em código Java. Dessa forma, evita-se mais uma etapa no processo de desenvolvimento do sistema multiagente, que é o de projeto orientado a objetos, permitindo com que a metodologia proposta seja diretamente aplicável. A ferramenta ZEUS também fornece suporte para testar e otimizar os agentes implementados.

Na identificação e modelagem dos papéis, Collis & Nudmu [6] sugerem que os papéis devem ser modulares, possuir alta coesão, serem parcimoniosos, completos, e baixo acoplamento (garantindo pouca dependência de entre papéis), assim como na metodologia GAIA [15].

A modelagem ZEUS fornece ainda algumas aplicações (exemplos) que, através da comparação, facilitam o desenvolvedor na identificação dos papéis da sua aplicação. Para a modelagem de papéis são elaborados 3 diagramas [6]: Diagrama de Papéis, Diagrama de Interação, Descrição dos Papéis.

Na fase de projeto, assim como na metodologia GAIA, os papéis são associados a agentes. A metodologia apresenta duas métricas para avaliar a criação de agentes [6]: Teste da Esfera de Responsabilidade (agentes devem ser autônomos, então cada esfera de responsabilidade deve ser suportada por um agente) e Teste do Ponto de Interação (agentes são distinguidos uns dos outros pela sua capacidade de interagir de maneira inteligente e construtiva com outros agentes - artificiais ou não).

Exemplos da utilização da modelagem ZEUS, combinada com a GAIA podem ser encontrados no endereço: <http://www.inf.ufes.br/~tavares/dsba>.

3 Comparações entre as metodologias

Os vários autores estudados concordam que os métodos de orientação a objetos não podem ser diretamente aplicados, sem modificações, principalmente porque existem diferenças conceituais importantes entre objetos e agentes e, por conseguinte nos modelos de sistemas resultantes. Porém, alguns autores defendem o uso de técnicas de orientação a objetos como um ponto de partida para os seus trabalhos.

A metodologia GAIA [15] considera o projeto de sistemas multiagentes como uma organização artificial, permitindo combinar o poder da abstração de agentes (na verdade, de papéis) com um projeto e implementação orientados a objetos (ou outro paradigma) significativamente simplificados e com alta reusabilidade. Uma característica particular da metodologia está na ênfase do uso de classes abstratas de

agente como um meio de agrupar papéis durante a análise e durante o refinamento do modelo. Desse modo, algumas decisões sobre os agentes concretos podem ser postergadas até a fase de projeto. Alterações quanto a granularidade, por exemplo, permitem que se modifique apenas a etapa de projeto, garantindo que os papéis não sejam modificados. Essa metodologia apresenta alta rastreabilidade entre os modelos de análise e projeto. A metodologia ZEUS também é orientada a papéis e apresenta poucas diferenças em relação a GAIA. A grande vantagem está no suporte de uma ferramenta para automatizar o processo de construção de agentes. Esse suporte remete o desenvolvedor aos exemplos disponíveis e implementados no ZEUS e ao conseqüente aproveitamento da modelagem desses exemplos. Entretanto, o uso de exemplos inibe a fase de análise criando dificuldades na modelagem do mundo real.

A abordagem baseada em KADS está principalmente relacionada aos aspectos de aquisição do conhecimento e objetiva aumentar o reuso do conhecimento e facilitar a manutenção dos sistemas. A idéia é separar os conhecimentos de domínio e de tarefa – problema abordado no estudo de Sistemas Baseados em Conhecimento. De maneira geral, os principais componentes de modelagem são tarefas, sub-tarefas e métodos de resolução de problemas.

Como pôde ser visto, existem similaridades entre a abstração de tarefa, utilizada pela abordagem CommonKads para sistemas multiagentes e as abordagens baseadas em papel (ZEUS, GAIA e *Workflow*). As noções de tarefa e papel podem ser facilmente mapeadas entre si, além do que a noção de responsabilidade usada nos modelos é a mesma. Uma tarefa pode ser mapeada para uma atividade realizada para alcançar as responsabilidades de um papel.

Em relação às metodologias que partem de modelos orientados a objetos, é importante compreender que noção de papel está em nível de abstração superior aos objetos. Na metodologia orientada a objetos, um subsistema pode ser visto como um conjunto de classes de objetos que colaboram para executar um conjunto de responsabilidades. Nesse sentido, a noção de papel é bem semelhante à noção de subsistema [10]. A diferença crucial é que, em um paradigma orientado a objetos, não há nenhuma construção de programação que apóie esse conceito, além do que, em tempo de análise, o poder da abstração de papéis auxilia a reduzir o *gap* semântico, tornando essa etapa mais natural para o desenvolvedor. Além disso, conforme cita Iglesias *et al* [9], as metodologias que partem dos modelos orientados a objetos não capturam a riqueza das interações que ocorrem entre os agentes, as propriedades dos agentes (estado mental, por exemplo) e sua autonomia.

A abordagem de Kinny et al [10], na arquitetura BDI, é semelhante à proposta de Burmeister [5]. A partir de uma técnica de orientação a objetos específica, isto é a OMT, Kinny [10] propõe uma metodologia que distingue entre um modelo externo e interno. O modelo externo consiste em um modelo de agente (que pode ser visto como o modelo organizacional de Burmeister) e um modelo de interação (que também pode ser visto como o modelo de cooperação de Burmeister). O modelo externo é independente da arquitetura escolhida para o agente. O modelo interno de Kinny [10] é o modelo de agentes de Burmeister, mas é específico para o agente da arquitetura BDI, pois descreve os agentes através de um modelo de plano, um modelo de objetivos e um modelo de crenças. O modelo interno da proposta de Kinny et al [10] é visto por Burmeister [5] como uma especialização de seu modelo de agente.

A tabela 1 apresenta as metodologias discutidas segundo os critérios de: ferramentas de suporte a construção dos modelos e implementação; fases do desenvolvimento; e, dependência do paradigma orientação a objetos.

Critério/ Metodologia	Burmeister	Kinny <i>et al</i>	MAS- CommonKADS	GAIA	Workflow Yu & Schmid	ZEUS
Ferramentas de suporte	Não	Não	Parcial	Não	Sim	Sim
Fases do desenvolvimento	A, P	A, P, PA	Todas	A, P	A, P, PA, I	Todas
Dependência do paradigma OO	Sim	Sim	Não	Parcial – P, I	Não	Não
Legenda: A – Análise, P – Projeto, PA – Projeto de Arquitetura, I – Implementação.						

Tabela 1: Comparação entre metodologias

4 Conclusões

Acredita-se que uma abordagem baseada em agentes coloca a maior parte do esforço de desenvolvimento de software na fase de análise e especificação de requisitos do sistema. Nesse contexto, as abordagens desenvolvidas especificamente para o uso da tecnologia de agentes reduzem o *gap* semântico entre o modelo e o mundo real, reduzindo o esforço de representar o mundo real através de uma modelagem mais natural.

As metodologias especificamente desenvolvidas para capturar as características da abstração de agentes também incorporam linguagens e notações que permitem a representação das características inteligentes do sistema. Essas características são representadas em termos de interações e responsabilidades. As interações entre os papéis (desempenhados pelos agentes) representam o conhecimento da organização e as responsabilidades específicas de cada agente podem assumir características reativas e pró-ativas, que devem ser suportadas por uma notação apropriada. Além disso, a característica de continuidade temporal é suportada na maior parte das notações utilizadas pelas metodologias orientadas a agentes (o que não acontece com metodologias orientadas a objetos).

O uso de metodologias apropriadas potencializa o reuso no desenvolvimento de sistemas multiagentes. A reutilização pode ser analisada sob diferentes níveis de granularidade: agentes, serviços, planos, ontologias.

A definição e o uso das noções de papel, responsabilidade, interação, equipe e sociedade ou organização em métodos para análise e projeto orientados a agentes, estão presentes em muitas das abordagens para sistemas multiagentes, permitindo uma modelagem natural de sistemas que podem ser vistos como agentes autônomos cooperantes.

Finalmente, observa-se que as abordagens metodológicas apresentadas não contemplam a modelagem do ambiente onde os agentes irão atuar. Essa questão pode se tornar crítica quando o ambiente (infra-estrutura), em tempo de modelagem dos agentes, ainda não existe. Um exemplo desse problema é mostrado por Brito [4], quando o ambiente dos agentes necessitou ser modelado através de outra metodologia (orientação a objetos), enquanto que o sistema multiagente foi modelado utilizando a metodologia GAIA [15].

5 Referências Bibliográficas

- [1] BREUKER, J., VAN DE VELDE, W., 1994, *CommonKADS Library for Expertise Modelling*, IOS Press.
- [2] BRITO, Silvana R., TAVARES, Orivaldo De Lira, MENEZES, Crediné S. *A Society of Intelligent Agents for an Environment of Cooperative Online Learning*. **Proceedings of the IC-AI2000**: June 26-29, 2000, Monte Carlo Resort, Las Vegas, Nevada, USA.
- [3] BRITO, Silvana R., TAVARES, Orivaldo de Lira, MENEZES, Crediné S. *An Environment of Cooperative Online Learning for Software Engineering*. **Proceedings of the International Conference on Engineering and Computer Education (ICECE2000)**. São Paulo: agosto, 2000.
- [4] BRITO, Silvana R. *Um sistema multiagente para apoio à mediação em ambientes de educação a distância orientados a projetos*. Dissertação de Mestrado (a defender). PPGI/UFES. Vitória: 2001.
- [5] BUNUEISTER, Birgit. *Models and methodologies for agent-oriented analysis and design*. In Klaus Fischer, editor, Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems. 1996. DFKI Document D-96-06.
- [6] COLLIS, Jaron, NDUMU, Divine. *The Role Modelling Guide*. Applied Research and Technology, BT Labs. Release 1.01, ago, 1999. [online] Disponível em: <http://193.113.209.147/projects/agents/zeus/library/zeus-rmg.zip> [capturado em 20/11/2000]
- [7] IGLESIAS, Carlos A., GARIJO, Mercedes, GONZÁLEZ, José C. , VELASCO, Juan R *A Methodological Proposal for Multiagent Systems*. **In:** 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW96), B. Gaines and M. Musen, editors, vol. 1, pp. 25-1/17, Banff, Canada, November 1996.
- [8] IGLESIAS, Carlos A., GARIJO, Mercedes. *Position Paper on Software Engineering for Agent Systems*. [online]. Disponível: <http://www.cs.vu.nl/~treur/SIG1.ciglesias.html>.
- [9] IGLESIAS, Carlos A., GARIJO, Mercedes, GONZÁLEZ. *A survey of agent-oriented methodologies*. In J. P. Muller, M. P. Singh, and A. S. Rao, editors, **Proceedings on .. the Fifth International Workshop on Agent, Theories, Architectures and Languages (ATAL-98)**, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1999.
- [10] KINNY, D., GEORGEFF M., RAO, A.. *A methodology and modelling technique for systems of BDI agents*. In: W. van der Velde, J. Perram (eds.): Agents Breaking Away. **Proceedings on** Workshop on Modelling Autonomous Agents in a MultiAgent World, MAAMAW-96, 7, pag. 56-71. Springer-Verlag: Berlin, Germany, 1996.
- [11] NWANA, Hyacinth, NDUMU, Divine, LEE, Lyndon COLLIS, Jaron. *ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems*. In Applied Artificial Intelligence Journal, Vol 13 (1), 1999, p129-186.
- [12] SCHREIBER, G., WIELINGA, B., BREUKER, J. *KADS: A Principled Approach to Knowledge-Based System Development*, Academic Press, 1993.
- [13] WOOLDRIDGE, M. CIANCARINI, P. *Agent-Oriented Software Engineering: The State of the Art*. **In:** P. Ciancarini and M. Wooldridge, editors, *Agent-Oriented Software Engineering*. Springer-Verlag Lecture Notes in AI Volume 1957, Jan, 2001.
- [14] WOOLDRIDGE, M., JENNINGS, N. R., KINNY, D. *A methodology for agent-oriented analysis and design*. In: O. Etzioni, J. P. Muller, and J. Bradshaw, editors: Agents '99: Proceedings on.. Autonomous Agents, Seattle, WA, 1999, Pag. 69 - 76.
- [15] WOOLDRIDGE, M., JENNINGS, N. R., KINNY, D. *The Gaia Methodology for Agent-Oriented Analysis and Design*. **In:** Journal of Autonomous Agents and Multi-Agent Systems. 3(3):285-312. 2000.
- [16] YU, Lei, SCHMID, Beat F. *A Conceptual Framework For Agent Oriented and Role Based Workflow Modeling* [online]. Disponível: http://www.knowledgemedia.net/netacademy/publications.nsf/all_print_pk/131.