

MODELAGEM DE UM SISTEMA MULTIAGENTE DE APOIO À PBL UTILIZANDO A METODOLOGIA MAS-COMMONKADS+

L. M. O. FONTES*, R. A. M. VALENTIM e F. M. MENDES NETO

Universidade Federal do Rio Grande do Norte - UFRN

Universidade Federal Rural do Semi-Árido - UFERSA

mabel.fontes@ufersa.edu.br*

Artigo submetido em fevereiro/2014 e aceito em outubro/2014

DOI: 10.15628/holos.2014.1966

RESUMO

A aprendizagem baseada em problema (*Problem-Based Learning* - PBL) é um método no qual os estudantes aprendem através da resolução de um problema que, em geral, não possui uma solução trivial e uma única solução correta. A PBL destaca o trabalho em equipe como um dos principais requisitos para o sucesso do processo de aprendizagem, ou seja, a colaboração é essencial. No entanto, a implantação de um método de ensino com base na PBL não é uma tarefa trivial. Em Ambientes Virtuais de Aprendizagem (AVAs), a complexidade de

implantação deste método é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração, nem possui todas as informações necessárias para aplicar as técnicas de aprendizagem deste método. Desta forma, este artigo apresenta o processo de modelagem de um Sistema Multiagente (SMA) de apoio à PBL. O SMA proposto foi modelado utilizando a metodologia MAS-CommonKADS+, que consiste em uma extensão da metodologia MAS-CommonKADS.

PALAVRAS-CHAVE: aprendizagem baseada em problema, MAS-CommonKADS+, sistema multiagente.

MODELING OF A MULTIAGENT SYSTEM TO SUPPORT PBL USING THE MAS-COMMONKADS+ METHODOLOGY

ABSTRACT

Problem-Based Learning (PBL) is a teaching method through which students are able to learn while solving problems that, in general, don't have trivial solutions or just one solution. PBL favors teamwork as one of the main requirements for a successful learning process. Therefore, collaboration is essential. However, applying a teaching approach like PBL is not a trivial task. The complexity of deploying such an approach on a Virtual Learning Environment (VLE) is even higher, as the

facilitator may not always be able to detect possible collaboration issues, or acquire all the information they may need in order to properly apply the method's techniques. Thus, this article presents the modeling process of a Multiagent System (MAS) to support PBL. The proposed MAS was modeled using the MAS-CommonKADS+ methodology, consisting of an extension of the MAS-CommonKADS methodology.

KEYWORDS: problem-based learning, MAS-CommonKADS+, multiagent system.

1 INTRODUÇÃO

Segundo Hmelo-Silver (2004), a aprendizagem baseada em problema (*Problem-Based Learning* - PBL) é um método no qual os estudantes aprendem através da resolução de um problema que, em geral, não possui uma solução trivial e uma única solução correta. A aprendizagem é centrada no estudante e o conhecimento é adquirido de forma autodirigida. Os estudantes trabalham em pequenos grupos colaborativos para identificar o que eles necessitam aprender para resolução do problema. O professor atua como facilitador do processo de aprendizagem ao invés de apenas transmitir conhecimentos.

A implantação de um método de ensino com base na PBL não é uma tarefa trivial. Em Ambientes Virtuais de Aprendizagem (AVAs), a complexidade de implantação deste método é ainda maior, pois o facilitador nem sempre pode detectar possíveis problemas na colaboração, nem possui todas as informações necessárias para aplicar as técnicas de aprendizagem deste método, como, por exemplo, saber quando os estudantes estão saindo do foco da discussão e tomar medidas de correção (HMELO-SILVER e BARROWS, 2006; PONTES, 2010).

O conceito de agentes de software tem se mantido como um importante tema de pesquisa no âmbito educacional. Esta abordagem tem se mostrado bastante promissora como auxílio em ambientes colaborativos de aprendizagem, devido a sua capacidade de dinamizar o processo. Eles podem ser usados, por exemplo, para auxiliar no cumprimento de uma dada teoria de aprendizagem em um AVA (PONTES, 2010). Sendo assim, a combinação de agentes e AVAs consiste em uma abordagem promissora para o aprendizado eficaz auxiliado por computador (SOLIMAN e GUETL, 2010).

Desta forma, este artigo tem como objetivo modelar um Sistema Multiagente (SMA), utilizando a metodologia MAS-CommonKADS+ (MORAIS II, 2010), para atender as principais metas relacionadas ao auxílio no cumprimento da PBL. O SMA proposto apresenta uma abordagem para aperfeiçoar a aplicação da PBL nos seguintes aspectos: detecção de estudantes passivos, detecção de conversações fora do contexto, formação de grupos e recomendação de Objetos de Aprendizagem (OAs).

A escolha da metodologia MAS-CommonKADS+ se deve ao fato dela englobar todas as características atuais de agentes racionais, empregando diagramas que oferecem uma melhor formalização para os modelos, incluindo a noção de objetos, além de uma ferramenta CASE (*Computer-Aided Software Engineering*) que possibilita a modelagem do sistema de uma forma clara e simples (MORAIS II, 2010).

Este trabalho está organizado da seguinte forma: na Seção 2, são descritos os principais conceitos sobre a PBL; a Seção 3 apresenta os conceitos sobre a metodologia MAS-CommonKADS+; a Seção 4 apresenta os trabalhos relacionados; a Seção 5 descreve a modelagem do SMA apresentada neste trabalho; e, por fim, na Seção 6, são apresentadas as considerações finais e os trabalhos futuros.

2 APRENDIZAGEM BASEADA EM PROBLEMA

A PBL é uma teoria de aprendizagem na qual os estudantes aprendem através da resolução de um problema (HMELO-SILVER, 2004). Na PBL, o facilitador tem o papel de guiar os estudantes neste processo, identificando possíveis deficiências de conhecimento e habilidades necessárias à solução do problema proposto. Assim, neste método, ao invés de termos o professor simplesmente repassando os conhecimentos e depois testando-os através de avaliações, ele faz com que os estudantes apliquem o seu conhecimento em situações novas. Os estudantes se deparam com problemas muitas vezes mal estruturados e tentam descobrir, através da investigação e pesquisa, soluções úteis.

Para o sucesso da aplicação da PBL como estratégia pedagógica, os seguintes estágios devem ser cumpridos (HMELO-SILVER, 2004; PONTES, 2010): (i) o facilitador propõe um problema mal estruturado para o grupo de estudantes, chamado cenário do problema; (ii) os estudantes analisam o problema e extraem fatos relevantes ao problema em questão, através de um *brainstorming* inicial; (iii) os estudantes têm um melhor entendimento do problema e formulam hipóteses para uma possível solução; (iv) os estudantes, auxiliados pelo facilitador, identificam deficiências de conhecimento para solução do problema; (v) os estudantes procuram por novos conhecimentos relacionados ao domínio e tentam gerar fatos sobre este novo conhecimento; (vi) ao final de cada problema, os estudantes refletem sobre os conhecimentos adquiridos. A Figura 1 ilustra o ciclo de desenvolvimento da PBL.

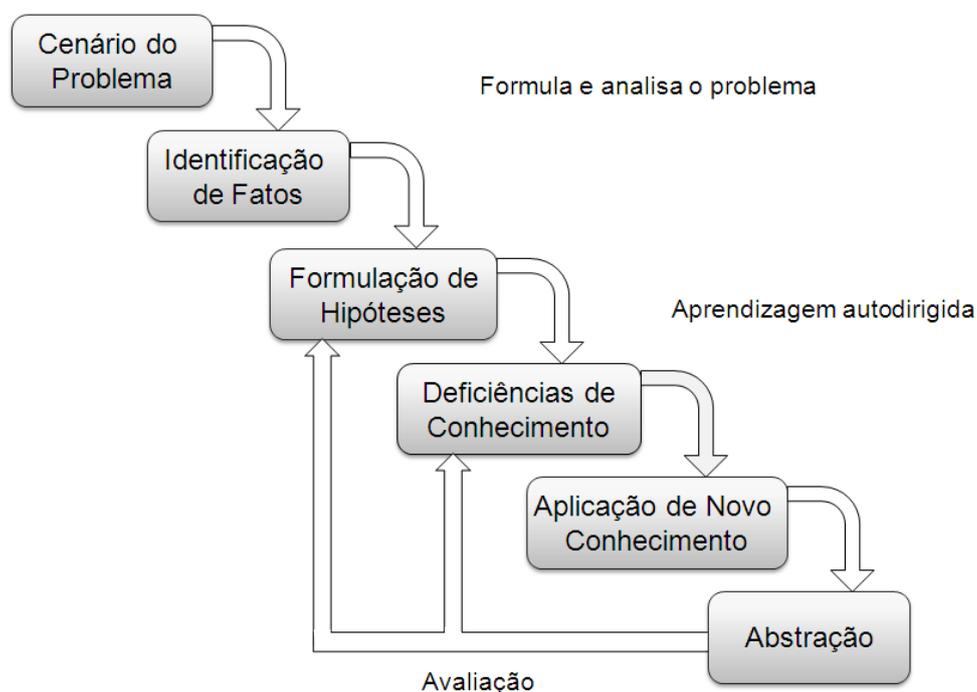


Figura 1: Ciclo da PBL.

3 METODOLOGIA MAS-COMMONKADS+

A MAS-CommonKADS+ consiste em uma extensão da metodologia MAS-CommonKADS, proposta por Morais II (2010).

A metodologia MAS-CommonKADS+ mantém muitos dos modelos já propostos na metodologia MAS-CommonKADS, porém realiza algumas modificações e adiciona novos conceitos. Ao invés de definir sete modelos, a MAS-CommonKADS+ contém nove modelos (MORAIS II, 2010; SILVA, 2012): (i) o modelo de requisitos é utilizado para descrever os requisitos do sistema; (ii) os modelos de tarefas e de conhecimento continuam sendo utilizados conforme especificado na MAS-CommonKADS; (iii) o modelo de recursos e objetos foi adicionado com o intuito de possibilitar a modelagem de objetos e recursos; (iv) o modelo de papéis tem como objetivo a identificação dos papéis do sistema e a representação de papéis que realizam as tarefas descritas no modelo de tarefas; (v) o modelo de organização descreve a estrutura organizacional de papéis do sistema, e não mais a organização de agentes, como acontecia na MAS-CommonKADS; (vi) o modelo de interação consiste na junção dos modelos de coordenação e de comunicação da MAS-CommonKADS; (vii) o modelo de agentes especifica os agentes, e por quais papéis eles são responsáveis, as percepções, os atuadores, as condições de ativação e de parada e a arquitetura do agente; (viii) o modelo de projeto descreve as características do local onde o sistema será instalado, os diagramas de implantação e informações a respeito da mobilidade dos agentes.

4 TRABALHOS RELACIONADOS

Em (MEDINA, SÁNCHEZ e CASTELLANOS, 2004), é proposto um modelo de agentes de software, baseado em ontologias¹, para recuperar informações de um conjunto de bibliotecas digitais. As ontologias são usadas para suportar duas tarefas principais: eliminar a ambiguidade de uma consulta e permitir o acesso aos repositórios. Os autores mostram ainda que os agentes propostos foram modelados utilizando a metodologia MAS-CommonKADS.

Em (SAMPAIO, FARIAS e LABIDI, 2007), são apresentadas as etapas do desenvolvimento de um software, baseado em agentes, utilizado para detectar fraudes em impostos municipais. Nas fases de análise e projeto do sistema, foram utilizadas a metodologia MAS-CommonKADS e uma ontologia. Os autores propõem um modelo de usuário genérico, obtido a partir de dados históricos de ações dos usuários e funções de inferência de comportamento. Os autores relatam ainda os resultados obtidos a partir de testes realizados no governo municipal de São Luís, Estado do Maranhão.

Em (TAVARES *et al.*, 2009), é descrito o processo de desenvolvimento de um SMA para controle de diabetes, denominado Monitor Glicêmico. Nesse trabalho, os autores apresentam uma visão geral do sistema, sua modelagem e implementação para Web e para dispositivos móveis. A modelagem do sistema Monitor Glicêmico foi realizada utilizando a metodologia MAS-CommonKADS.

Em (HASAN e ISAAC, 2011), é apresentado um *framework* para desenvolvimento de um sistema especialista baseado na Web. Esse *framework* foi proposto com base na integração da metodologia MAS-CommonKADS, a arquitetura *Model-View-Controller* (MVC) e em estratégias de otimização de aplicações Web, para lidar com a complexidade do desenvolvimento e para alcançar alta usabilidade.

¹ Neste trabalho, uma ontologia deve ser compreendida como um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes.

Como diferencial do nosso trabalho, podemos destacar que, diferentemente dos outros trabalhos discutidos nessa seção, a modelagem do SMA de apoio à PBL, proposta neste trabalho, foi realizada utilizando a metodologia MAS-CommonKADS+. Além disso, este trabalho propôs a criação detalhada do modelo de requisitos, utilizando o *framework i** (YU, 1995). Vale ressaltar ainda que o SMA proposto é voltado especificamente para aplicação da PBL, ou seja, apresenta a modelagem de um SMA para auxiliar na aplicação correta da PBL, uma teoria de aprendizagem comprovadamente eficaz (STROBEL e VAN BARNEVELD, 2009).

5 MODELAGEM DO SISTEMA MULTIAGENTE DE APOIO À PBL

A Subseção 5.1 apresenta uma breve explanação sobre os agentes que compõem o SMA proposto. Já a subseção 5.2 apresenta a modelagem do SMA proposta neste trabalho.

5.1 Sistema Multiagente

O SMA proposto é composto de sete tipos de agentes: Agente Pedagógico Animado (AgPA), Agente Monitorador de Problemas (AgMP), Agente Detector de Problemas (AgDP), Agente Monitorador de Grupos (AgMG), Agente Gerenciador de Grupos (AgGG), Agente Recomendador (AgR) e Agente DF (*Directory Facilitator*).

O AgPA (Figura 2 e Figura 3) tem o intuito de apoiar os estudantes na resolução de problemas, através da teoria de aprendizagem PBL. O AgPA consiste em um modelo humanoide tridimensional animado responsável por acompanhar os estudantes durante o processo de aplicação da PBL, além de manter os estudantes sempre motivados. Para obter sucesso nesse último caso, o AgPA expressa emoções similares às dos seres humanos.

O AgMP (Figura 4 e Figura 5) é responsável pelo monitoramento da criação de novos problemas. Este agente possui dois comportamentos: o primeiro tem o objetivo de instanciar uma ontologia (FONTES, 2010) do problema, com todas as informações do problema inseridas pelo facilitador; já o segundo, tem o objetivo de enviar uma mensagem para o AgDP, com o intuito de acionar os seus comportamentos de detecção de problemas.

O AgDP (Figura 6 e Figura 7) é responsável pela detecção de estudantes passivos e detecção de conversações fora do contexto do problema. O AgDP executa estes comportamentos uma vez por dia, durante toda a realização de um curso. Esse agente foi criado com o intuito de auxiliar o facilitador na avaliação do comportamento dos estudantes durante o processo de aplicação da PBL. Desta forma, uma vez sendo detectado um comportamento indesejado, o AgDP irá notificar o facilitador, e esse, por sua vez, poderá tomar uma providência cabível.

O AgMG (Figura 8 e Figura 9) é responsável pelo monitoramento da criação de novos perfis dos grupos. Ele é responsável por solicitar, ao AgGG, a criação automática dos grupos.

O AgGG (Figura 10 e Figura 11) é o responsável pela formação automática dos grupos e possui dois comportamentos principais, um implementado em Java (VERONESE *et al.*, 2002) e outro em Prolog (GOMES *et al.*, 2002). O comportamento do AgGG implementado em Java é responsável pela geração de candidatos que estão aptos a participar de determinado grupo. Já o comportamento em Prolog é responsável pela alocação dos estudantes aos grupos propriamente dita. No final desse processo, é gerado um arquivo que contém os *rankings* para formação de

grupos. O facilitador analisará este resultado, que é exibido através de uma interface do Moodle (MOODLE, 2011), e decidirá se acata ou não a sugestão do AgGG.

O AgR (Figura 12 e Figura 13) tem o intuito de detectar OAs adequados ao contexto do estudante, levando em consideração (i) as informações providas pelos seus respectivos perfis, (ii) os assuntos que os estudantes desconhecem e (iii) as informações dos OAs disponíveis no repositório de OAs SCORM (SCORM, 2013). O AgR utiliza um Algoritmo Genético (AG) (FONTES, 2013) para identificar os OAs a serem recomendados.

O papel de mediador da comunicação entre os agentes é realizado pelo agente DF, o qual é provido pela própria plataforma JADE (BELIFEMINE, CAIRE e GREENWOOD, 2007), conforme exigência da especificação FIPA (FIPA, 2011).

5.2 Modelagem do SMA

Nem todos os diagramas dos respectivos modelos são necessários para a compreensão da arquitetura multiagente proposta, visto que alguns deles acabam fornecendo informações que se tornam redundantes. Logo, decidiu-se representar cada modelo através de um diagrama, contribuindo assim, para a compreensão do SMA. A escolha dos modelos a serem inseridos na modelagem depende do tamanho e da complexidade do SMA sendo modelado (SILVA, 2012).

Seguindo os passos da metodologia MAS-CommonKADS+, temos, inicialmente, o modelo de requisitos. Neste trabalho, esse modelo foi criado utilizando o *framework* i*. Optou-se em utilizar o *framework* i* pelo fato de ele representar, através dos modelos *Strategic Dependency* (SD) e *Strategic Rationale* (SR), os atores e as dependências entre os mesmos, para que suas metas sejam alcançadas, recursos sejam fornecidos e tarefas sejam realizadas. Além disso, o uso do referido *framework* possibilita o entendimento das razões internas dos atores, uma vez que as mesmas são expressas explicitamente, facilitando, assim, a compreensão de cada agente que compõe o SMA.

A Figura 2 ilustra o modelo SD do AgPA. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgPA, AgDP e Estudante.

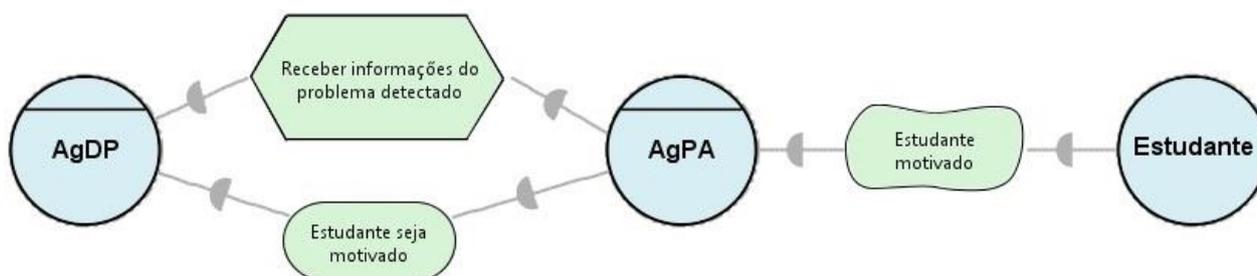


Figura 2: Modelo SD do AgPA.

A Figura 3 ilustra o modelo SR do AgPA. Esse modelo apresenta detalhadamente as estratégias internas do AgPA.

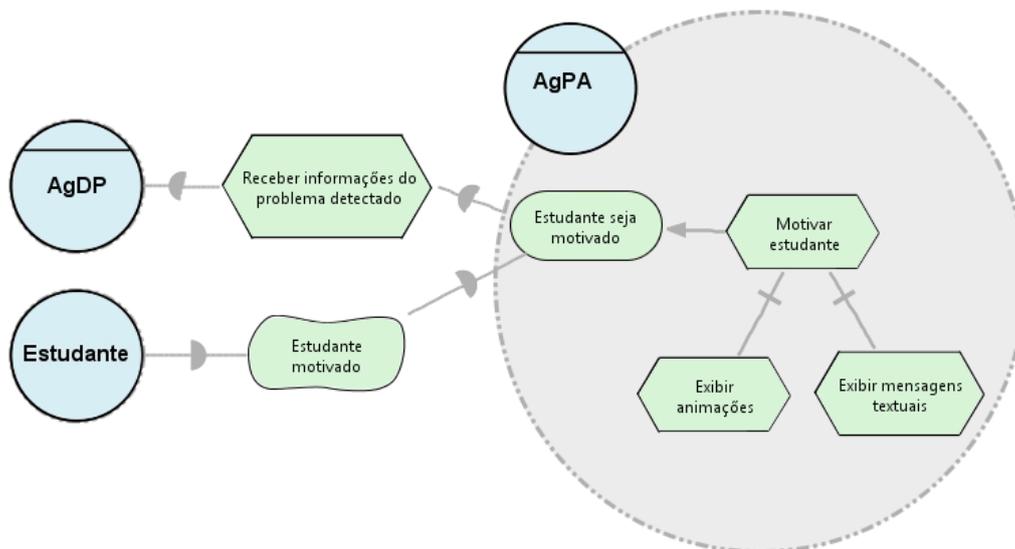


Figura 3: Modelo SR do AgPA.

A Figura 4 ilustra o modelo SD do AgMP. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgMP e Facilitador.

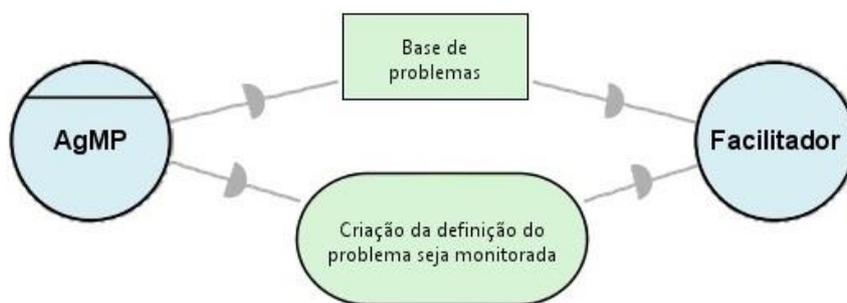


Figura 4: Modelo SD do AgMP.

A Figura 5 ilustra o modelo SR do AgMP. Esse modelo apresenta detalhadamente as estratégias internas do AgMP.

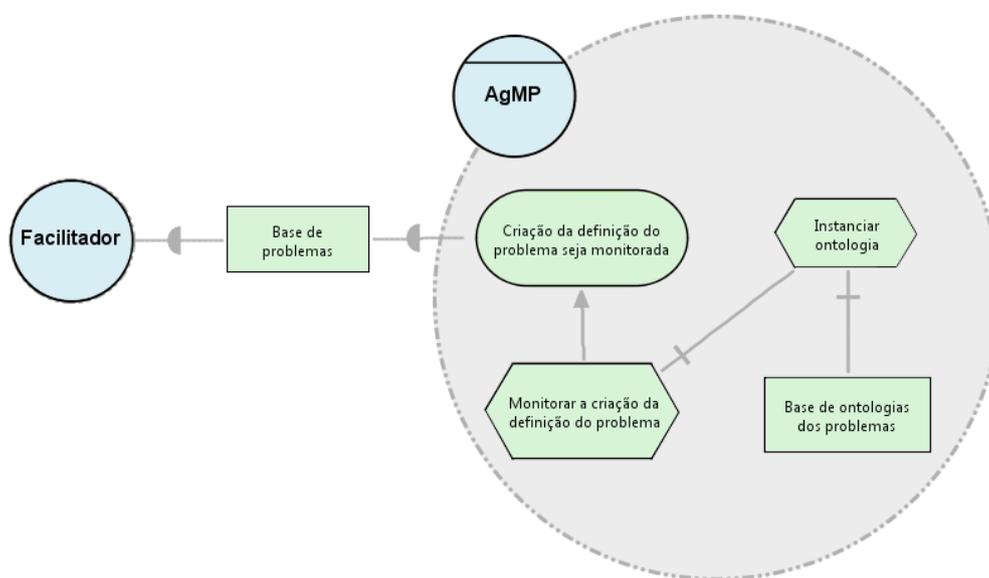


Figura 5: Modelo SR do AgMP.

A Figura 6 ilustra o modelo SD do AgDP. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgDP, AgMP, Estudante e Facilitador.

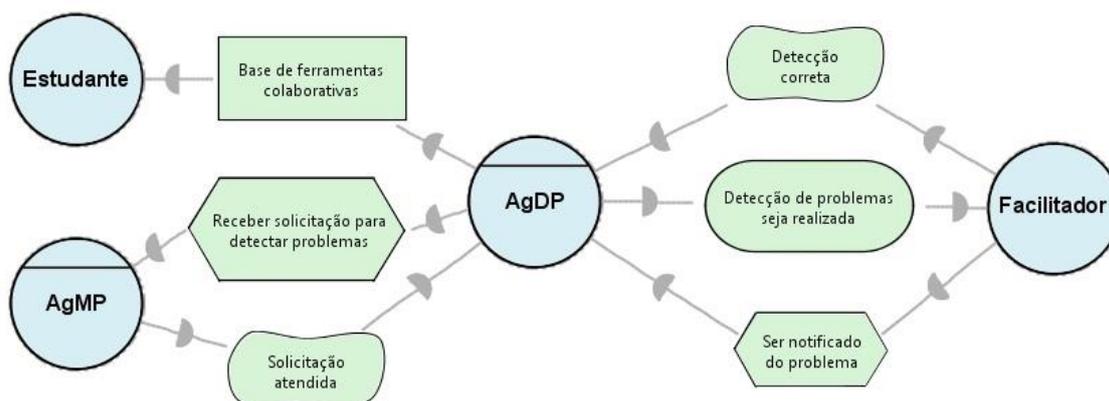


Figura 6: Modelo SD do AgDP.

A Figura 7 ilustra o modelo SR do AgDP. Esse modelo apresenta detalhadamente as estratégias internas do AgDP.

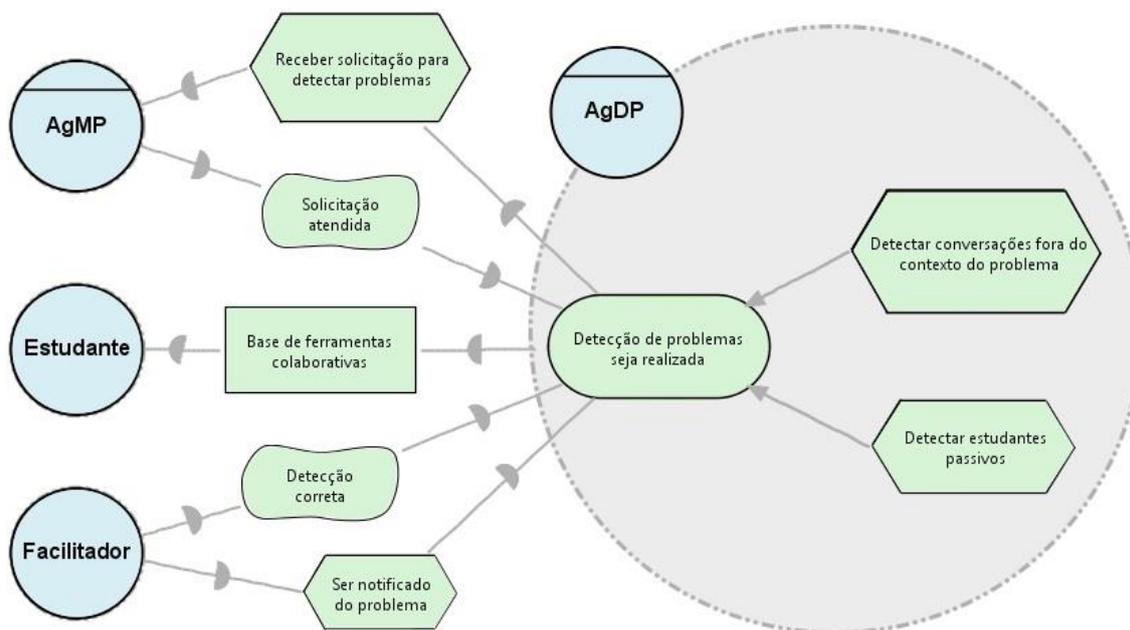


Figura 7: Modelo SR do AgDP.

A Figura 8 ilustra o modelo SD do AgMG. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgMG e Facilitador.

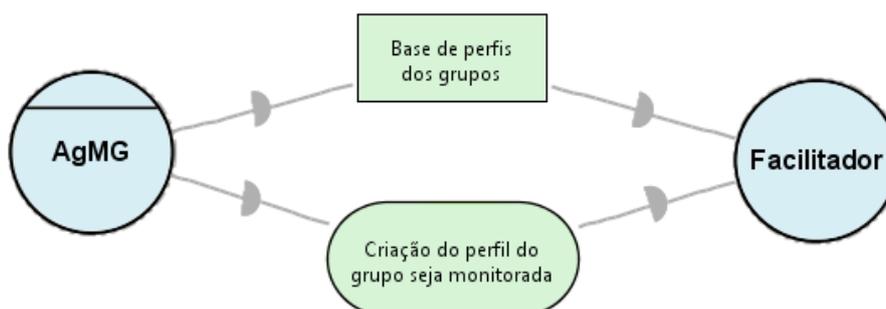


Figura 8: Modelo SD do AgMG.

A Figura 9 ilustra o modelo SR do AgMG. Esse modelo apresenta detalhadamente as estratégias internas do AgMG.

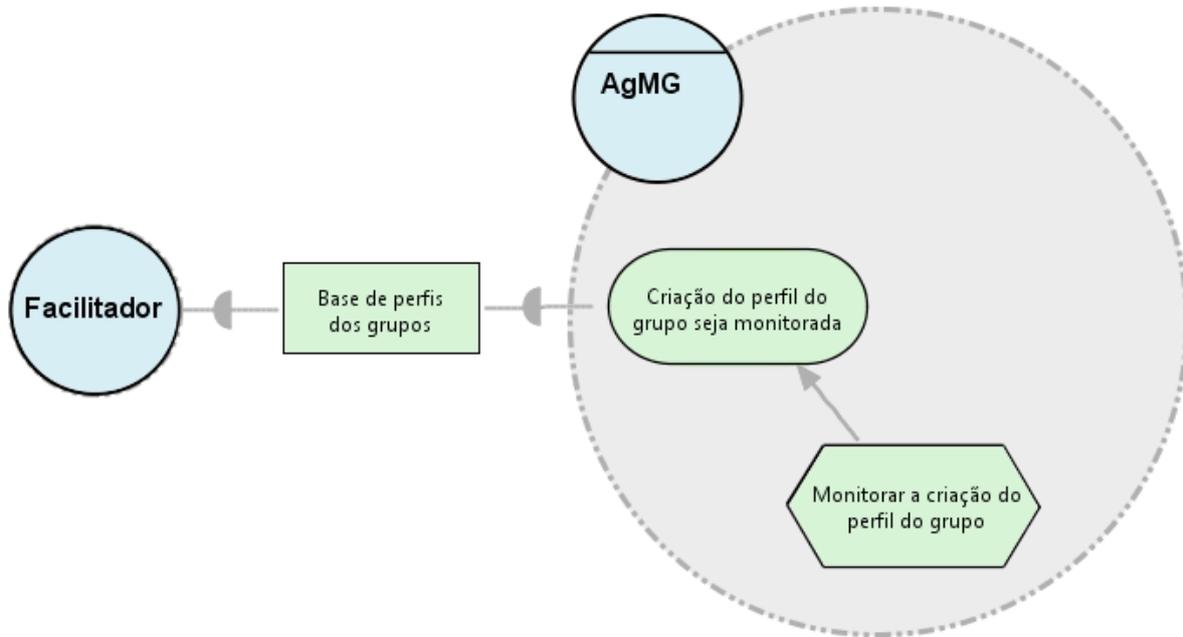


Figura 9: Modelo SR do AgMG.

A Figura 10 ilustra o modelo SD do AgGG. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgGG, AgMG, Estudante e Facilitador.

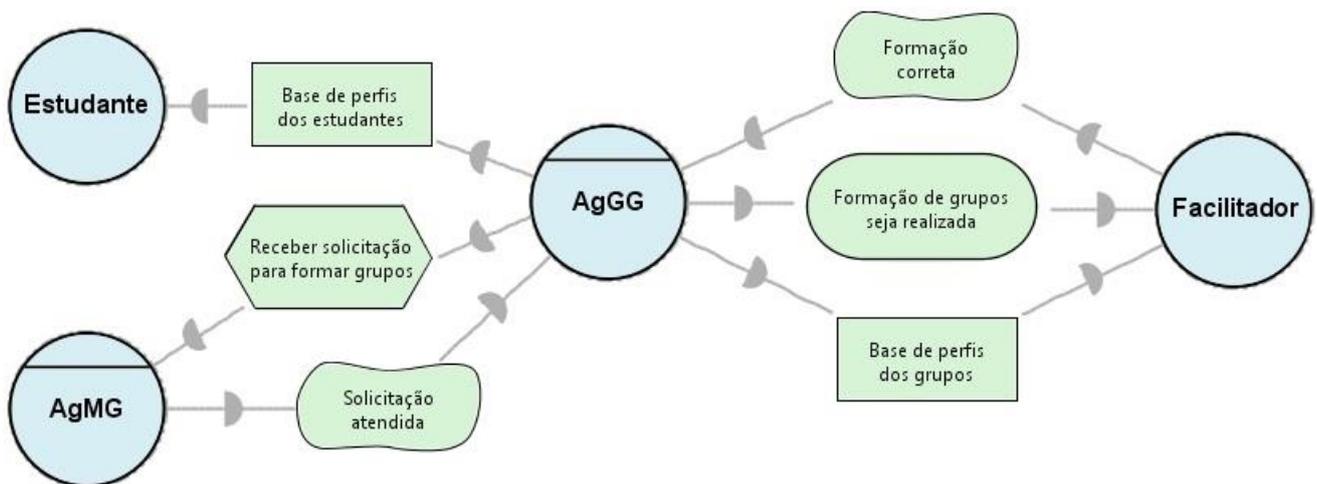


Figura 10: Modelo SD do AgGG.

A Figura 11 ilustra o modelo SR do AgGG. Esse modelo apresenta detalhadamente as estratégias internas do AgGG.

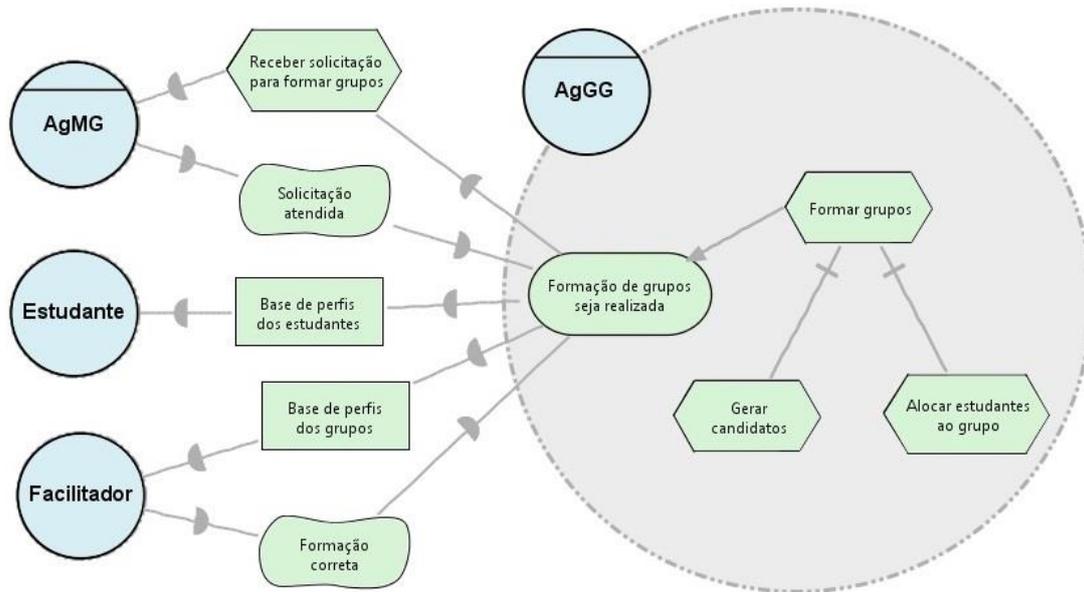


Figura 11: Modelo SR do AgGG.

A Figura 12 ilustra o modelo SD do AgR. Esse modelo exhibe os relacionamentos de dependência estratégica entre os atores AgR, Facilitador e Estudante.

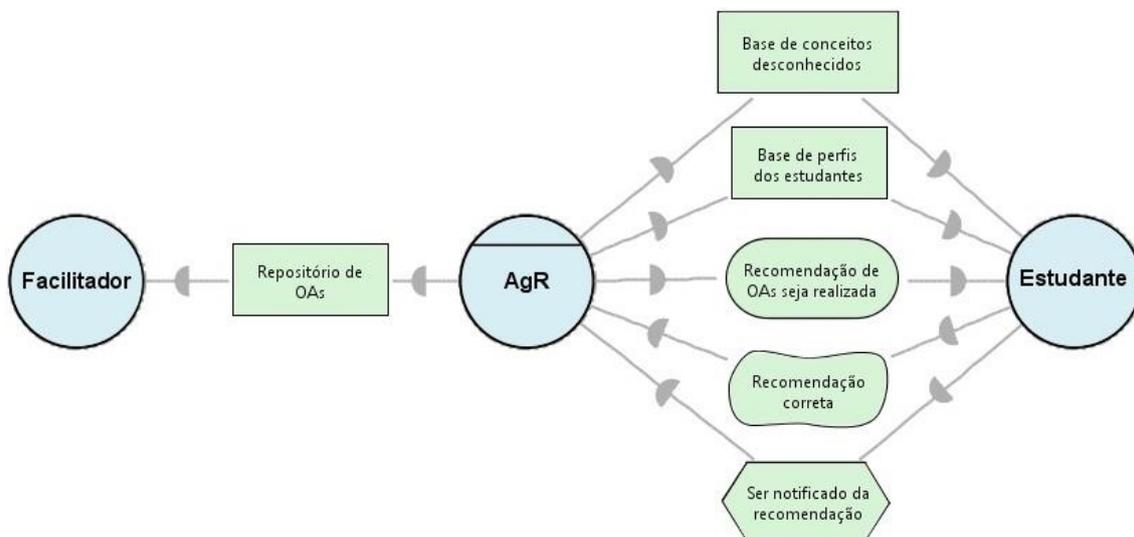


Figura 12: Modelo SD do AgR.

A Figura 13 ilustra o modelo SR do AgR. Esse modelo apresenta detalhadamente as estratégias internas do AgR.

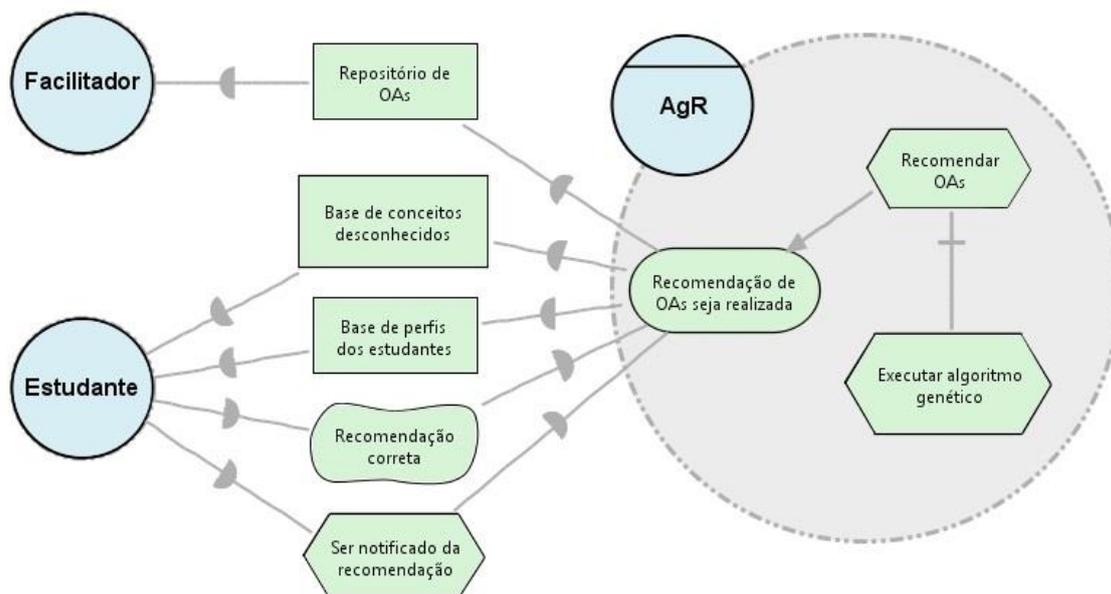


Figura 13: Modelo SR do AgR.

O DF não foi modelo com o *framework* i*, pois ele não realiza tarefas que compõem a finalidade principal do SMA. Este agente é provido pela própria plataforma JADE e sua função é mediar a comunicação entre os demais agentes.

Baseando-se nas necessidades da arquitetura proposta, o SMA deve realizar uma série de tarefas, as quais podem ser observadas no Modelo de Tarefas exibido na Figura 14.

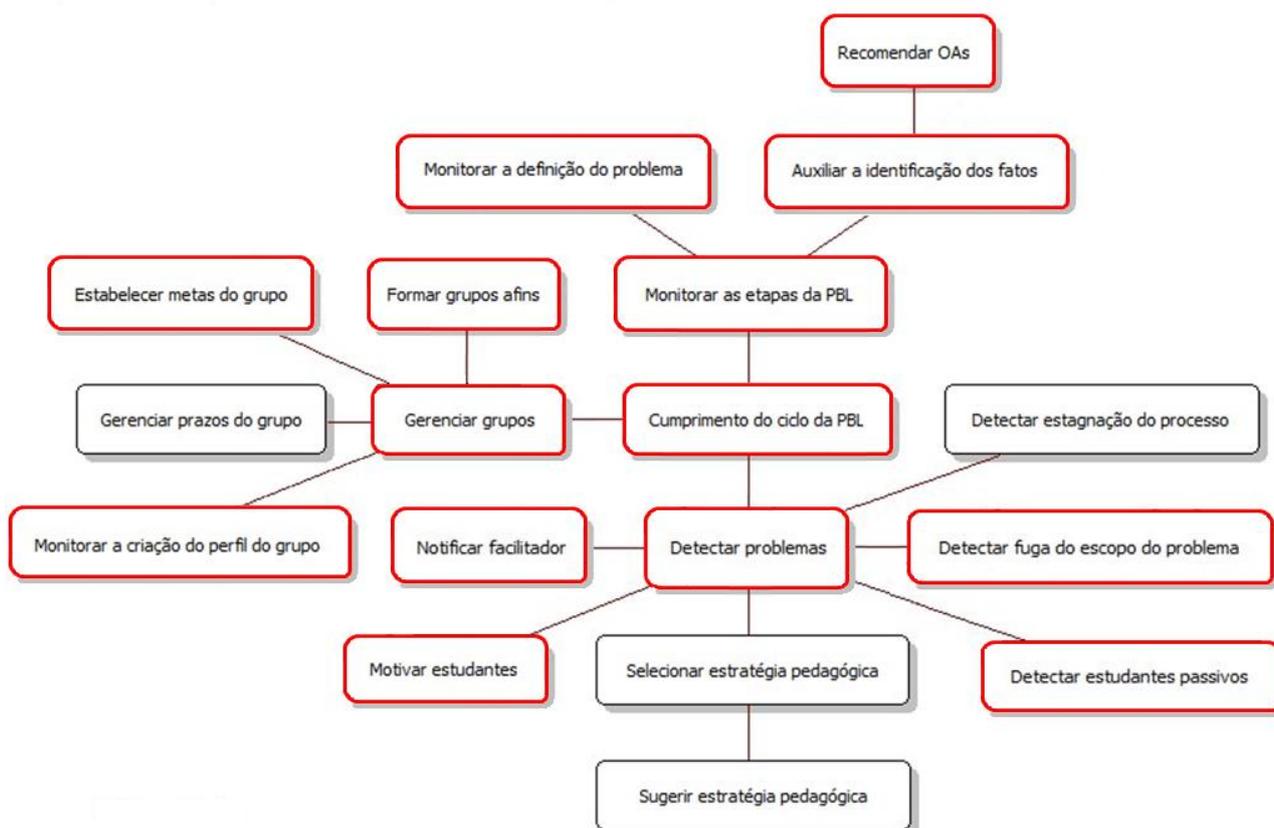


Figura 14: Modelo de Tarefas.

As tarefas destacadas em vermelho foram contempladas neste trabalho, ou seja, implementadas, enquanto as demais ficaram para trabalhos futuros. A tarefa principal desse SMA é o cumprimento do ciclo da PBL. Essa tarefa, por sua vez, foi decomposta em outras subtarefas, que têm como intuito final auxiliar no atendimento da tarefa principal.

Seguindo os passos propostos pela metodologia, deve-se realizar a captura dos objetos e recursos do sistema (MORAIS II, 2010). O diagrama utilizado para esse modelo está ilustrado na Figura 15.

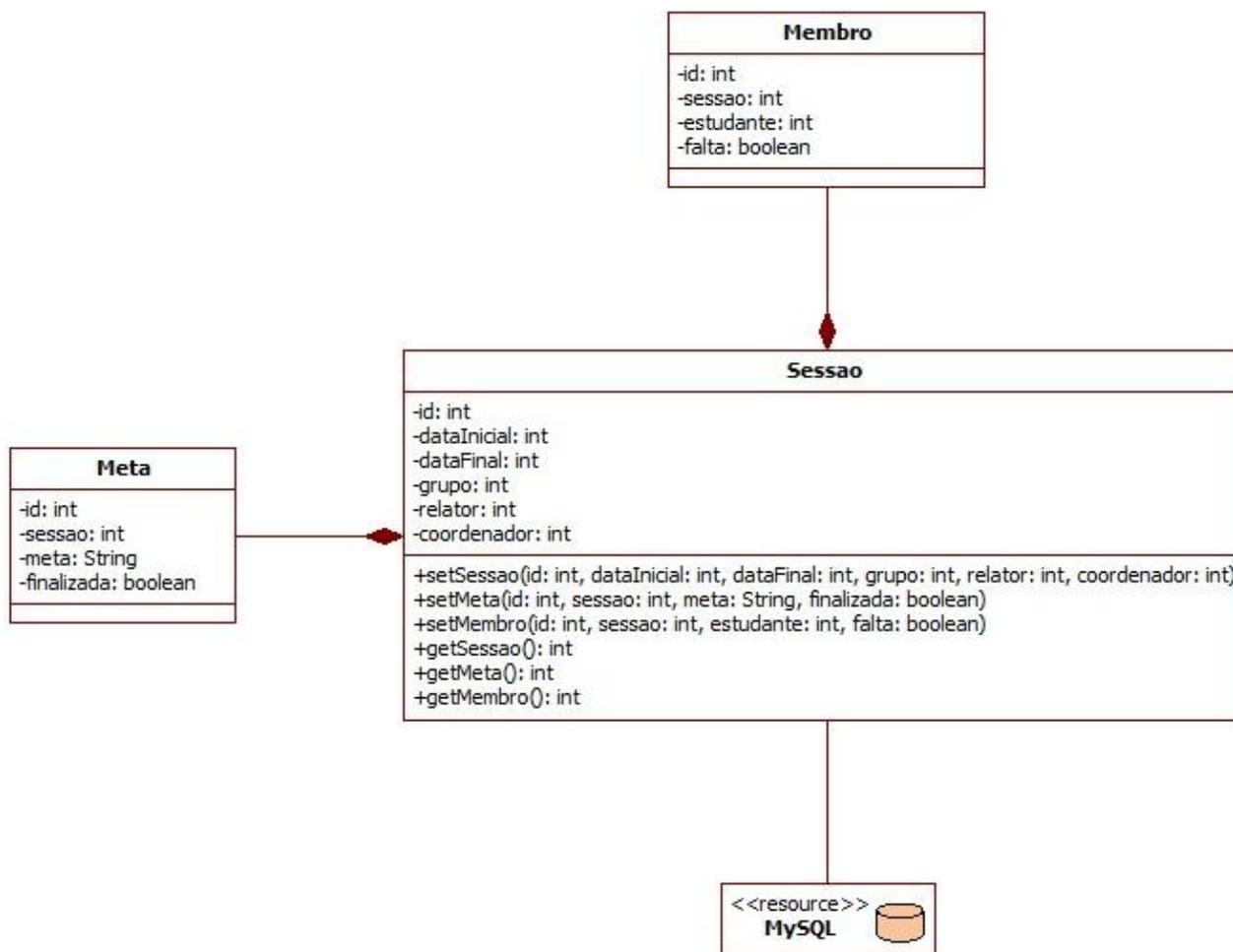


Figura 15: Modelo de Recursos e Objetos.

O próximo passo consiste em definir os papéis que cada agente irá realizar no sistema. Vale ressaltar que um único agente pode ser responsável por vários papéis em um SMA. Cada papel pode realizar várias das tarefas exibidas em um Modelo de Tarefas. O Modelo de Papéis serve para identificar quais papéis irão realizar quais tarefas. Baseando-se nas tarefas elencadas no Modelo de Tarefas da Figura 14, foi definido o Modelo de Papéis apresentado na Figura 16.

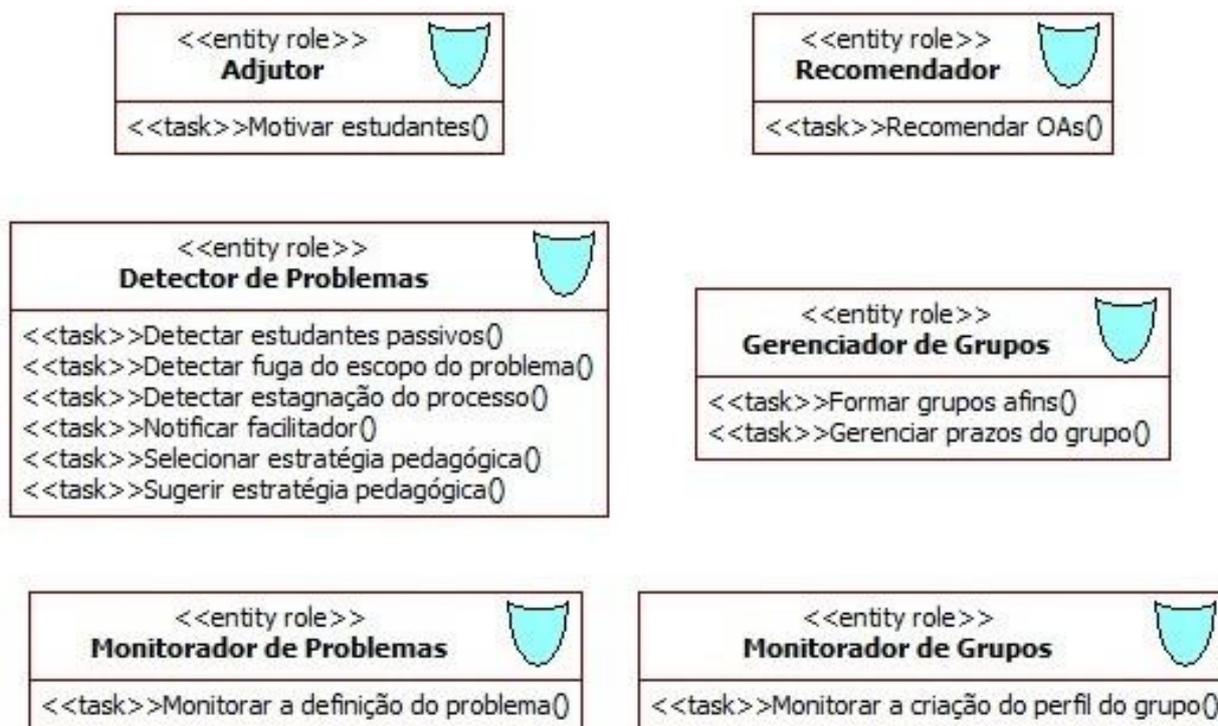


Figura 16: Modelo de Papéis.

Na Figura 16, é possível perceber que foram definidos seis tipos de papéis para atender às tarefas inerentes ao Modelo de Tarefas: adjutor, recomendador, detector de problemas, gerenciador de grupos, monitorador de problemas e monitorador de grupos. Apesar de ser possível que cada agente possa realizar mais de um papel, decidiu-se que, no SMA proposto, cada um desses papéis será realizado por um agente. Esses agentes são, respectivamente, o Agente Pedagógico Animado - AgPA, o Agente Recomendador - AgR, o Agente Detector de Problemas - AgDP, o Agente Gerenciador de Grupos - AgGG, o Agente Monitorador de Problemas - AgMP e o Agente Monitorador de Grupos - AgMG. Desta forma, a partir de agora, os papéis não serão mais citados, mas sim os agentes que realizam esses papéis. Esses agentes serão detalhados no Modelo de Agentes, em passos posteriores da metodologia. Vale ressaltar que as tarefas *detectar estagnação do processo*, *selecionar estratégia pedagógica* e *sugerir estratégia pedagógica*, do papel detector de problemas, e *gerenciar prazos do grupo*, referente ao papel gerenciador de grupos, não foram contempladas neste trabalho.

Uma vez construído o Modelo de Papéis, deve ser desenvolvido o Modelo de Organização, que serve para descrever a estrutura organizacional dos papéis no sistema, ou seja, como os papéis estão relacionados entre si. A Figura 17 mostra o Modelo de Organização correspondente aos papéis exibidos na Figura 16.

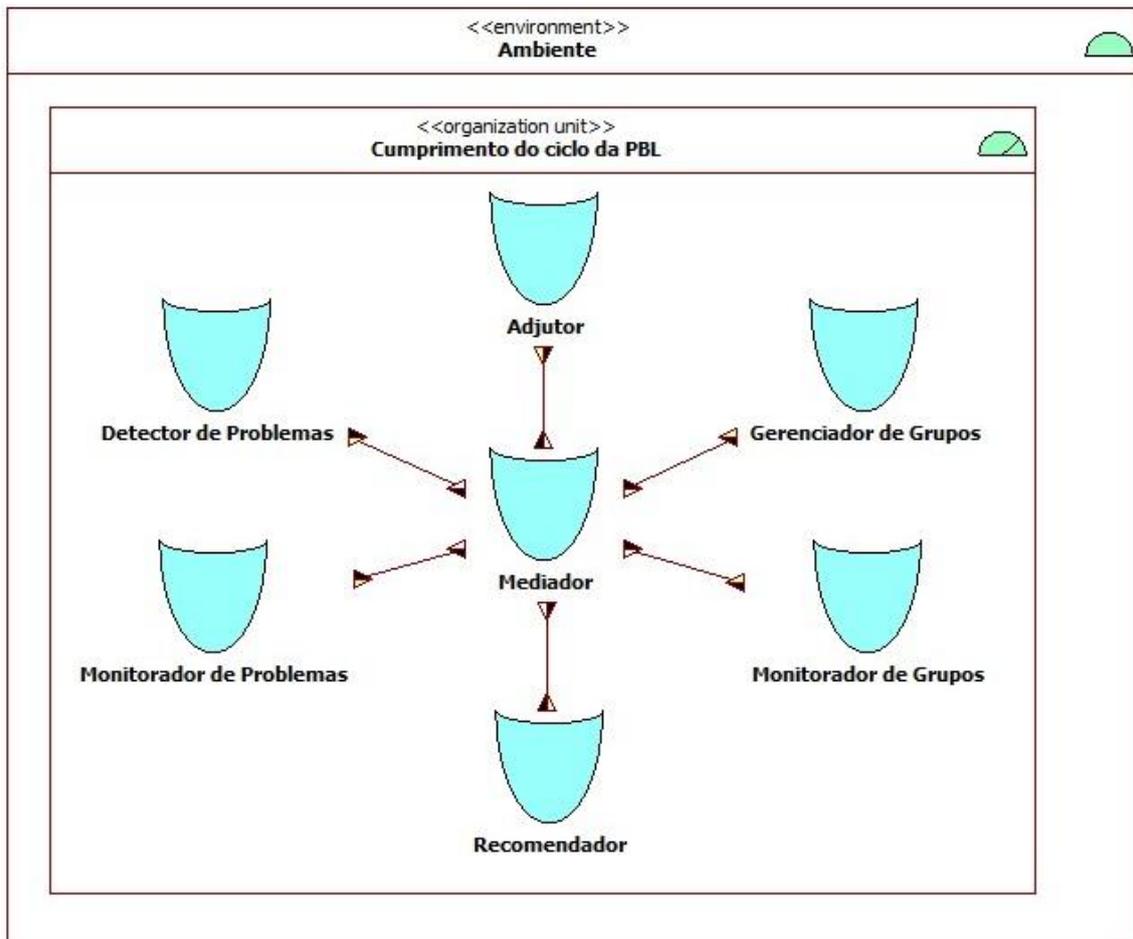


Figura 17: Modelo de Organização.

Nesse modelo é possível perceber a presença de um outro papel, o mediador. O mediador permitirá que os agentes se comuniquem sem que, inicialmente, eles se conheçam. O motivo dele não fazer parte do Modelo de Papéis é que o mesmo não realiza tarefas que compõem a finalidade principal do SMA, ou seja, não realiza uma das tarefas que compõem o Modelo de Tarefas. Mesmo assim, essa tarefa é de extrema relevância, pois é através dela que os agentes conseguem informações a respeito dos serviços de outros agentes e conseguem se comunicar uns com os outros. Ou seja, ele funciona como um intermediador, permitindo que agentes cadastrem informações e pesquisem por informações de outros agentes. Conforme já mencionado, o papel do mediador da comunicação entre os agentes é realizado pelo agente DF. O funcionamento desse agente será melhor explicado no Modelo de Agentes.

O Modelo de Organização apresenta a organização interna e externa dos papéis. A primeira apresenta os relacionamentos entre os papéis do SMA, enquanto que a segunda demonstra os relacionamentos entre as organizações em si. O SMA deste trabalho possui apenas uma unidade organizacional (*organization unit*), logo não existe uma representação externa. Como pode ser visto no Modelo de Organização da Figura 17, na representação interna da unidade “Cumprimento do ciclo da PBL”, os sete papéis estão relacionados através de um relacionamento do tipo *peer*, que é usado para papéis com a mesma autoridade e é representado por uma seta parcialmente cheia (SILVA, 2012).

Na fase inicial da modelagem, é necessário que se tenha, no mínimo, uma noção estática do SMA, o que pode ser alcançado com o Modelo de Organização. Porém, é interessante também

ter uma visão do comportamento dinâmico do SMA, o que é possível conseguir através do Modelo de Interação (SILVA, 2012). A Figura 18 mostra o modelo de interação entre os agentes AgMP, AgDP, DF e AgPA.

No modelo apresentado na Figura 18, é possível visualizar toda a interação que ocorre entre os agentes AgMP, AgDP, DF e AgPA e os recursos que são consultados por tais agentes. O diagrama mostra desde o comportamento do AgMP, no momento que o facilitador cria um novo problema, até a informação chegar no AgPA, que irá tentar motivar os estudantes envolvidos no problema detectado, e o próprio facilitador, que poderá tomar uma providência cabível. É possível perceber ainda que toda a interação ocorre com o auxílio do agente DF, através do seu serviço de páginas amarelas.

Também é possível perceber que as informações dos problemas criados pelo facilitador são instanciadas em ontologias. Outra forma possível de se armazenar essas informações seria com a utilização de um banco de dados. Porém, o uso de uma ontologia para representação desse conhecimento mostra-se como uma alternativa mais flexível, visto que esse conhecimento pode ser utilizado por outras partes da aplicação e, caso seja necessário modificar o comportamento desta, pode-se criar uma nova forma de representar a ontologia, sem modificar a codificação do sistema em si. Diante disso, está sendo utilizada uma ontologia genérica do problema, sendo instanciada uma ontologia para cada novo problema criado pelo facilitador.

Na Figura 19, é possível observar o modelo de interação entre os agentes AgMG, AgGG e DF. Nesse modelo, é possível visualizar toda a interação que ocorre entre os agentes AgMG, AgGG e DF e os recursos que são consultados por tais agentes. O diagrama mostra desde o comportamento do AgMG, no momento que o facilitador cria um novo perfil do grupo, até a informação chegar no AgGG, que irá realizar a formação do grupo propriamente dita.

Percebe-se que o AgR não participa das interações apresentadas anteriormente. De fato, o AgR não se comunica com nenhum dos agentes presentes nessa arquitetura. Quando acionado, o AgR executa suas tarefas sem precisar de nenhum serviço oferecido pelos outros agentes. No entanto, ele faz a recomendação com base, dentre outras coisas, nas informações obtidas nas bases de dados, que são alimentadas, na maioria das vezes, pelos demais agentes.

Uma vez sabendo como os diferentes papéis em um SMA estão organizados entre si, é necessário definir quais agentes serão responsáveis por cada papel. É necessário também definir a arquitetura dos agentes, seus objetivos e suas características, como entrada de dados, condições de ativação do agente e tipos de informação disponíveis. Para uma melhor visualização das informações, os *templates* textuais, com os objetivos e características dos agentes, serão descritos em tabelas em subseções específicas para cada agente. Nessas subseções, serão fornecidas informações dos modelos dos agentes.

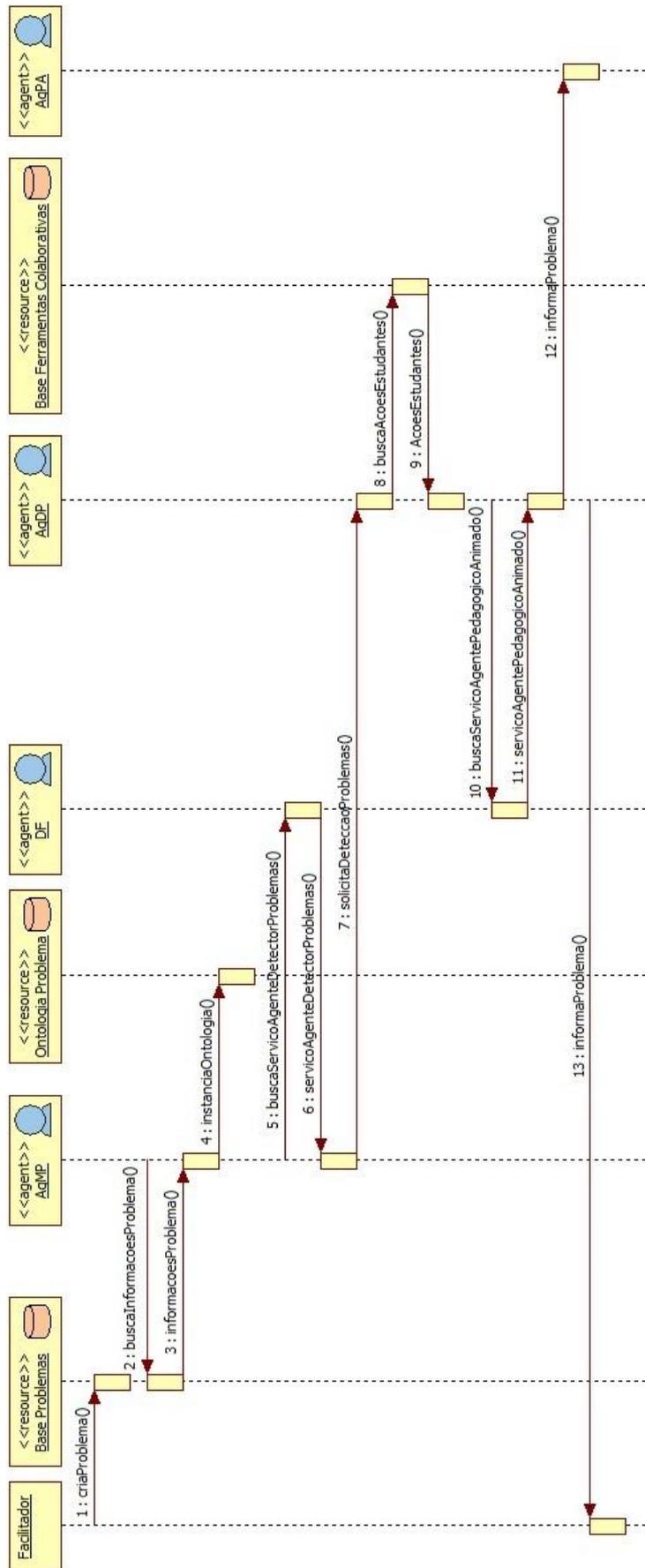


Figura 18. Modelo de interação entre os agentes AgMP, AgDP, DF e AgPA

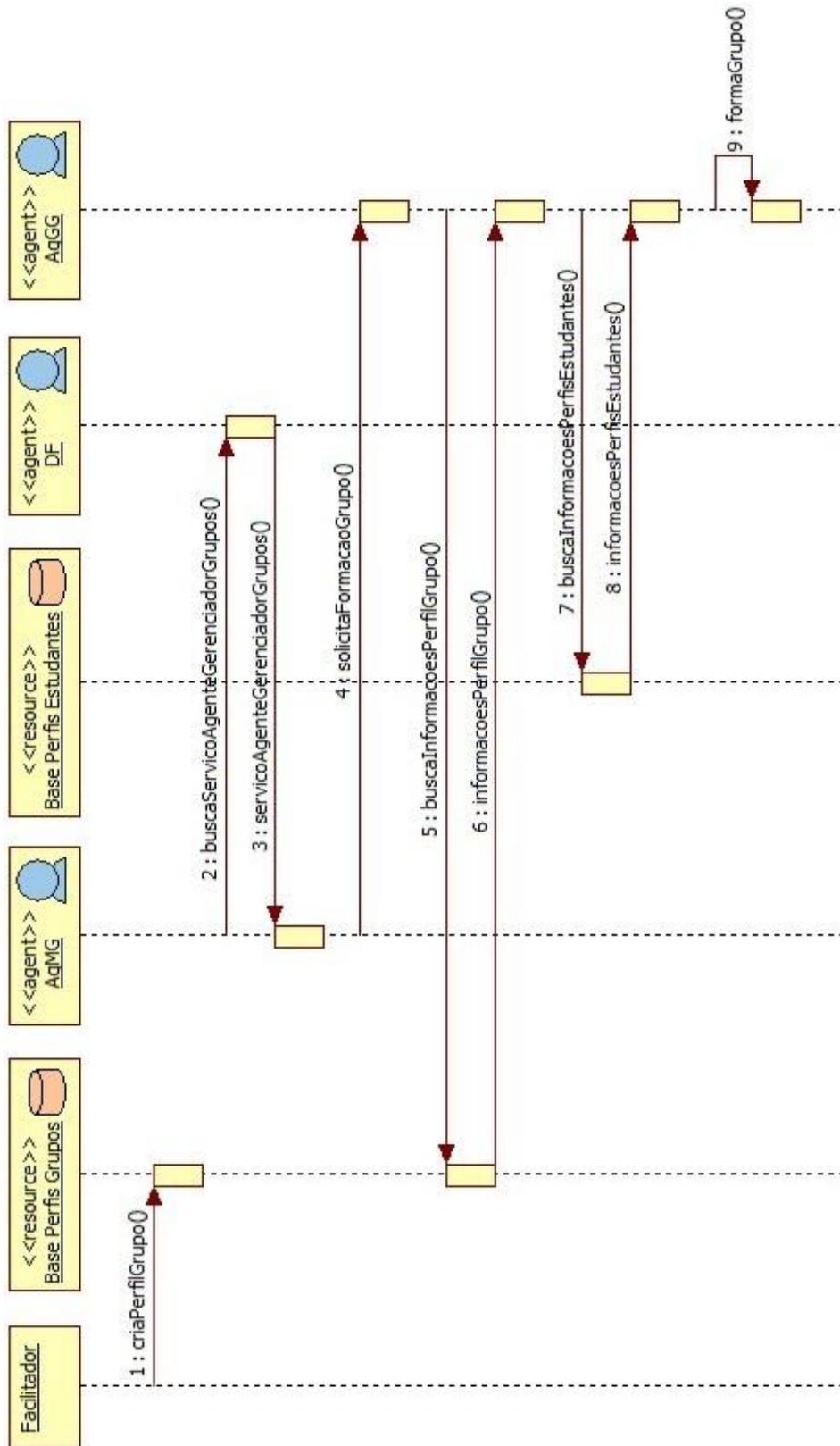


Figura 19. Modelo de interação entre os agentes AgMG, AgGG e DF

5.2.1 Modelo do Agente Pedagógico Animado – AgPA

O AgPA realiza o papel “Adjutor”. A Tabela 1 apresenta o seu *template* textual.

Tabela 1: Template textual do AgPA.

Agente: AgPA
<p>Objetivos</p> <p>Disponibilizar um serviço do tipo “<i>agente-pedagogico-animado</i>” no DF de modo que o AgDP possa encontrar esse serviço para se comunicar com ele. Uma vez em posse das informações enviadas pelo AgDP, o AgPA tenta motivar os estudantes, exibindo animações com expressões de emoções e mensagens textuais condizentes com o problema de colaboração detectado.</p>
<p>Parâmetros de Entrada</p> <p>Informações sobre os estudantes envolvidos no problema detectado enviadas pelo AgDP.</p>
<p>Parâmetros de Saída</p> <p>Animações com expressões de emoções e mensagens textuais.</p>
<p>Condição de Ativação</p> <p>Quando o agente percebe que um estudante se autenticou no ambiente.</p>
<p>Condição de Finalização</p> <p>Quando o agente percebe que o estudante se desconectou do ambiente.</p>
<p>Informação Associada</p> <p>Possui um comportamento que disponibiliza um serviço do tipo “<i>agente-pedagogico-animado</i>” no DF de modo que o AgDP possa encontrar esse serviço e se comunicar com ele. Também possui um comportamento cíclico, do tipo <i>Cyclic Behaviour</i> (BELIFEMINE, CAIRE e GREENWOOD, 2007), para ficar monitorando por requisições do AgDP. Uma vez em posse dessas informações, o AgPA tenta motivar os estudantes a participarem mais das discussões e a usarem as ferramentas disponíveis no AVA, exibindo animações com expressões de emoções e mensagens textuais.</p>
<p>Descrição</p> <p>Este agente é do tipo reativo simples (FONTES, 2013) e possui o comportamento voltado para o acompanhamento dos estudantes durante o processo de aplicação da PBL. Ele permanece ativo durante todo o período em que o estudante estiver conectado ao ambiente, motivando-o quando algum problema de colaboração for detectado.</p>

5.2.2 Modelo do Agente Monitorador de Problemas – AgMP

O AgMP realiza o papel “Monitorador de Problemas”. A Tabela 2 apresenta o seu *template* textual.

Tabela 2: Template textual do AgMP.

Agente: AgMP
<p>Objetivos</p> <p>Monitorar a criação de um novo problema, capturando o identificador do problema; instanciar uma ontologia com as informações do novo problema; e enviar uma mensagem para o AgDP, acionando os seus comportamentos de detecção de problemas.</p>
<p>Parâmetros de Entrada</p> <p>Identificador do novo problema.</p>
<p>Parâmetros de Saída</p> <p>Instância da ontologia do problema.</p>
<p>Condição de Ativação</p> <p>Quando o agente percebe que o facilitador criou um novo problema.</p>
<p>Condição de Finalização</p>

Quando o agente cria uma instância da ontologia do problema e se comunica com o AgDP, acionando os seus comportamentos de detecção de problemas.
Informação Associada Possui dois comportamentos de execução única, do tipo <i>One Shot Behaviour</i> (BELIFEMINE, CAIRE e GREENWOOD, 2007). O primeiro tem o objetivo de instanciar uma ontologia do problema, com todas as informações do problema inseridas pelo facilitador. O segundo tem o objetivo de enviar uma mensagem para o AgDP, com o intuito de acionar os seus comportamentos de detecção de problemas.
Descrição Este agente é do tipo reativo simples e possui o comportamento voltado para o monitoramento da criação de novos problemas. Ele é responsável por instanciar uma ontologia para cada novo problema criado e acionar os comportamentos de detecção de problemas do AgDP.

5.2.3 Modelo do Agente Detector de Problemas – AgDP

O AgDP realiza o papel “Detector de Problemas”. A Tabela 3 apresenta o seu *template* textual.

Tabela 3: Template textual do AgDP.

Agente: AgDP
Objetivos Disponibilizar um serviço do tipo “ <i>agente-detector-problemas</i> ” no DF de modo que o AgMP possa encontrar esse serviço para se comunicar com ele. Uma vez em posse das informações enviadas pelo AgMP, o AgDP aciona os seus dois comportamentos: detecção de estudantes passivos e detecção de conversações fora do contexto do problema. Uma vez que o AgDP tenha detectado um dos dois problemas, este irá notificar o facilitador, via e-mail, informando todas as informações inerentes aos estudantes envolvidos com o problema detectado. Em seguida, todas as informações relativas a esses estudantes, assim como as informações sobre o problema detectado, são enviadas para o AgPA.
Parâmetros de Entrada Informações sobre quais atividades os estudantes estão executando nas ferramentas colaborativas; informações sobre o conteúdo das discussões nas ferramentas colaborativas; e as ontologias dos problemas.
Parâmetros de Saída E-mail com informações sobre o problema detectado e sobre os estudantes envolvidos.
Condição de Ativação Quando o agente recebe do AgMP uma mensagem com identificador de conversação do tipo <i>deteccao-problemas</i> .
Condição de Finalização Quando o agente percebe que o ciclo da PBL foi encerrado.
Informação Associada Possui um comportamento que disponibiliza um serviço do tipo “ <i>agente-detector-problemas</i> ” no DF de modo que o AgMP possa encontrar esse serviço e se comunicar com ele. Também possui um comportamento cíclico, do tipo <i>Cyclic Behaviour</i> , para ficar monitorando por requisições do AgMP. Uma vez em posse dessas informações, o AgDP aciona os seus dois comportamentos: detecção de estudantes passivos e detecção de conversações fora do contexto do problema. Esses dois comportamentos são executados a cada determinado período de tempo e são do tipo <i>Ticker Behaviour</i> (BELIFEMINE, CAIRE e GREENWOOD, 2007). Uma vez detectado um dos problemas citados anteriormente, o AgDP notifica o facilitador, via e-mail, informando todas as informações inerentes aos estudantes envolvidos com o problema detectado. Ele também envia todas as informações relativas a esses estudantes, assim como as informações sobre o problema detectado, para o AgPA.
Descrição Este agente é do tipo reativo simples e possui o comportamento voltado para o auxílio ao facilitador durante todo o ciclo da PBL. Ele é responsável por detectar problemas de colaboração que podem vir a acontecer durante o processo de aplicação da PBL.

5.2.4 Modelo do Agente Monitorador de Grupos – AgMG

O AgMG realiza o papel “Monitorador de Grupos”. A Tabela 4 apresenta o seu *template* textual.

Tabela 4: Template textual do AgMG.

Agente: AgMG
Objetivos Monitorar a criação de um novo perfil do grupo, capturando o identificador do grupo a ser criado, e enviar uma mensagem para o AgGG, solicitando a criação automática do grupo.
Parâmetros de Entrada Identificador do grupo a ser criado.
Parâmetros de Saída Mensagem enviada para o AgGG, com a solicitação de criação automática do grupo.
Condição de Ativação Quando o agente percebe que o facilitador criou um novo perfil do grupo.
Condição de Finalização Quando o agente se comunica com o AgGG e envia para este a solicitação de criação automática do grupo.
Informação Associada Possui um comportamento de execução única, do tipo <i>One Shot Behaviour</i> , para enviar, para o AgGG, a solicitação de criação automática do grupo, passando como parâmetro o identificador do grupo a ser criado.
Descrição Este agente é do tipo reativo simples e possui o comportamento totalmente voltado para o monitoramento da criação de novos perfis dos grupos. Ele é responsável por solicitar, ao AgGG, a criação automática dos grupos.

5.2.5 Modelo do Agente Gerenciador de Grupos – AgGG

O AgGG realiza o papel “Gerenciador de Grupos”. A Tabela 5 apresenta o seu *template* textual.

Tabela 5: Template textual do AgGG.

Agente: AgGG
Objetivos Disponibilizar um serviço do tipo “ <i>agente-gerenciador-grupos</i> ” no DF de modo que o AgMG possa encontrar esse serviço para se comunicar com ele. Uma vez em posse das informações enviadas pelo AgMG, o AgGG forma, automaticamente, grupos afins com base nos perfis dos estudantes e dos grupos.
Parâmetros de Entrada Identificador do grupo a ser criado e informações sobre os perfis dos estudantes e dos grupos.
Parâmetros de Saída Arquivo com um <i>ranking</i> dos candidatos aptos a formarem o grupo.
Condição de Ativação Quando o agente recebe do AgMG uma mensagem com identificador de conversação do tipo <i>formacao-grupos</i> .
Condição de Finalização Quando o agente gera um <i>ranking</i> com os candidatos aptos a formarem o grupo.
Informação Associada Possui um comportamento que disponibiliza um serviço do tipo “ <i>agente-gerenciador-grupos</i> ” no DF de modo que o AgMG possa encontrar esse serviço e se comunicar com ele. Também possui um comportamento cíclico, do tipo

Cyclic Behaviour, para ficar monitorando por requisições do AgMG. Uma vez em posse dessas informações, um comportamento de execução única, do tipo *One Shot Behaviour*, é acionado para gerar um *ranking* com os candidatos aptos a formarem o grupo.

Descrição

Este agente é do tipo baseado em objetivo (FONTES, 2013) e possui o comportamento voltado para o auxílio ao facilitador na tarefa de formação de grupos. Ele é responsável pela alocação automatizada de estudantes a grupos de trabalho para solução de um problema.

5.2.6 *Modelo do Agente Recomendador – AgR*

O AgR realiza o papel “Recomendador”. A Tabela 6 apresenta o seu *template* textual.

Tabela 6: Template textual do AgR.

Agente: AgR
<p>Objetivos</p> <p>Detectar OAs adequados ao contexto do estudante, de acordo com (i) as informações providas pelos seus respectivos perfis, (ii) os assuntos que os estudantes desconhecem e (iii) as informações obtidas dos OAs disponíveis no repositório.</p>
<p>Parâmetros de Entrada</p> <p>Informações sobre os perfis dos estudantes, os conceitos que os estudantes desconhecem do problema proposto e informações dos OAs disponíveis no repositório de OAs SCORM.</p>
<p>Parâmetros de Saída</p> <p>Possíveis OAs adequados ao contexto do estudante.</p>
<p>Condição de Ativação</p> <p>Quando o estudante submete os conceitos desconhecidos, via interface do Moodle, durante a fase de identificação dos fatos do ciclo da PBL.</p>
<p>Condição de Finalização</p> <p>Quando o agente envia para o estudante, via e-mail, as informações do OA a ser recomendado.</p>
<p>Informação Associada</p> <p>Toma as decisões de acordo com o resultado da execução de um algoritmo genético. Essa recomendação auxilia os estudantes a sanar suas dúvidas na fase de identificação dos fatos, durante a resolução do problema.</p>
<p>Descrição</p> <p>Este agente é do tipo baseado em utilidade (FONTES, 2013) e possui o comportamento voltado para a recomendação de OAs. Ele é responsável por detectar e realizar a recomendação de OAs.</p>

5.2.7 *Modelo do Agente DF*

O Agente DF realiza o papel “Mediador”. A Tabela 7 apresenta o seu *template* textual.

Tabela 7: Template textual do Agente DF.

Agente: DF
<p>Objetivos</p> <p>Possibilitar que os agentes possam cadastrar e procurar por serviços oferecidos por outros agentes (serviço de páginas amarelas).</p>
<p>Parâmetros de Entrada</p> <p>Registros de serviços por parte dos agentes.</p>
<p>Parâmetros de Saída</p> <p>Lista de serviços e os respectivos agentes responsáveis pelo mesmo.</p>

Condição de Ativação
Quando a plataforma JADE é inicializada.
Condição de Finalização
Caso a plataforma JADE seja finalizada.
Informação Associada
É um agente inicializado pela própria plataforma JADE como um dos componentes integrantes do padrão FIPA.
Descrição
Este agente é do tipo reativo simples e possui o comportamento voltado para a mediação entre os outros agentes. Sua função principal é fornecer uma arquitetura do tipo “quadro-negro”, onde agentes escrevem informações, procuram por informações escritas por outros agentes e conseguem, através do serviço provido pelo DF, se comunicar com o agente que escreveu aquela informação.

Após definir os *templates* textuais, é necessário definir o modelo de agentes. Devido ao tamanho deste artigo, optamos por apresentar apenas a modelagem do AgR, conforme ilustrado na Figura 20.

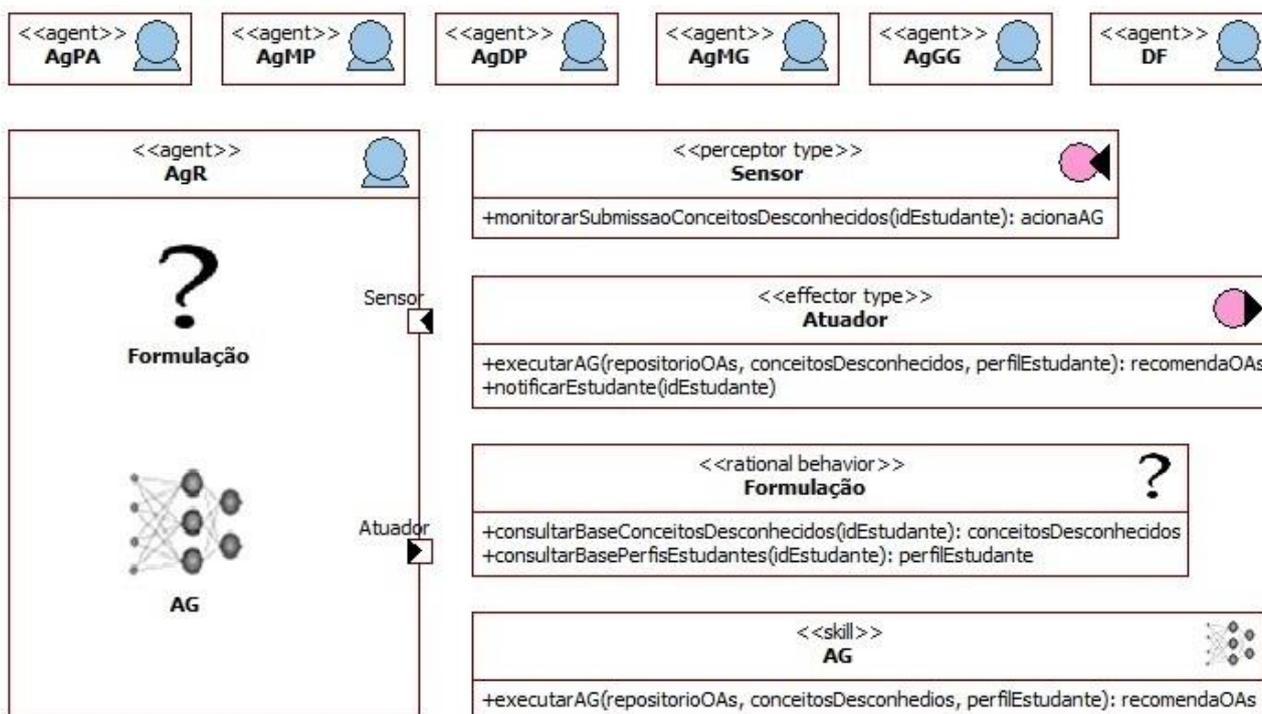


Figura 20. Modelo de Agentes

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A PBL tem apresentado bons resultados com relação ao processo de ensino e aprendizagem. Contudo, foi visto que sua implantação é complexa, pois necessita de uma série de requisitos para seu sucesso. A sua implantação em AVAs é uma tarefa ainda mais complicada, pois o facilitador não está presente fisicamente nas sessões de interação e nem sempre possui informações sobre como o processo está sendo realizado ou se há algum problema a ser solucionado.

O uso de agentes de software tem se destacado como auxílio neste problema, facilitando o gerenciamento do ambiente e, ainda mais importante, o cumprimento de uma teoria de aprendizagem eficaz para o sucesso do processo de ensino e aprendizagem no ambiente.

Como contribuição deste trabalho podemos destacar a modelagem de um SMA, utilizando a metodologia MAS-CommonKADS+, para atender as principais metas relacionadas ao auxílio no cumprimento da PBL. Vale destacar ainda que este trabalho propôs a criação detalhada do modelo de requisitos, utilizando o *framework i**, algo não visto na metodologia MAS-CommonKADS+. O SMA apresentado neste trabalho facilita a tarefa de ensino e aprendizagem quando a PBL é utilizada como teoria de aprendizagem. Ele também oferece mais subsídio para que o facilitador possa acompanhar o processo de aplicação da PBL, possibilitando uma intervenção mais eficaz quando problemas de colaboração são detectados.

Como trabalhos futuros, pretende-se abordar as tarefas que não foram contempladas neste trabalho, como, por exemplo, gerenciar prazos do grupo, selecionar e sugerir estratégias pedagógicas e detectar estagnação do processo, que é um problema que ocorre quando os estudantes não possuem conhecimentos suficientes para continuar o processo, ou seja, o grupo fica “perdido”.

7 REFERÊNCIAS BIBLIOGRÁFICAS

1. BELIFEMINE, F.; CAIRE, G.; GREENWOOD, D. **Developing multi-agent systems with JADE**. Liverpool, Inglaterra: John Wiley & Sons, Ltd, 2007.
2. FIPA. Welcome to the Foundation for Intelligent Physical Agents. **Site Oficial do Padrão FIPA**, 2011. Disponível em: <<http://www.fipa.org/>>. Acesso em: 08 jan. 2013.
3. FONTES, L. M. O. **Uma Arquitetura Multiagente de Apoio à Aprendizagem Baseada em Problema**. 2013. 153 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual do Rio Grande do Norte, Universidade Federal Rural do Semi-Árido, Mossoró, 2013.
4. FONTES, L. M. O. **Um Sistema Multiagente de Apoio à Aprendizagem Baseada em Problema**. 2010. 78 f. Monografia (Graduação em Ciência da Computação) - Universidade Federal Rural do Semi-Árido, Mossoró, 2010.
5. GOMES, A. K.; BERNADINI, F. C.; MONARD, M. C.; BATISTA, G. E. A. P. A. **Uma Sintaxe Padrão Prolog para Classificadores Simbólicos**. São Carlos: n. 154, 2002.
6. HASAN, S. S.; ISAAC, R. K. An integrated approach of MAS-CommonKADS, Model-View-Controller and web application optimization strategies for web-based expert system development. **International Journal Expert Systems with Applications**, Elsevier, v. 38, n. 1, p. 417-428, 2011.
7. HMELO-SILVER, C. E.; BARROWS, H. S. Goals and strategies of a problem-based learning facilitator. **The Interdisciplinary Journal of Problem-based Learning**, v. 1, n. 1, p. 21-39, 2006. Disponível em: <<http://docs.lib.purdue.edu/ijpbl/vol1/iss1/4/>>. Acesso em: 8 abr. 2010.
8. HMELO-SILVER, C. E. Problem-Based Learning: What and How Do Students Learn? **Educational Psychology Review**, v. 16, n. 3, 2004.
9. MEDINA, M. A.; SÁNCHEZ, A.; CASTELLANOS, N. Ontological Agents Model based on MAS-CommonKADS methodology. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS,

- COMMUNICATIONS AND COMPUTERS - CONIELECOMP, 14. p. 260–263, 2004.
10. MOODLE. About Moodle. **Site oficial do Moodle**, 2011. Disponível em: <http://docs.moodle.org/21/en/About_Moodle>. Acesso em: 02 dez. 2011.
 11. MORAIS II, M. J. D. O. **MAS-CommonKADS+: Uma Extensão à Metodologia Mas-CommonKADS para Suporte ao Processo Detalhado de Sistemas Multiagentes Racionais**. Dissertação de Mestrado. Universidade Estadual do Ceará - UECE. Fortaleza, CE. 2010.
 12. PONTES, A. A. A. **Uma Arquitetura de Agentes para Suporte à Colaboração na Aprendizagem Baseada em Problemas em Ambientes Virtuais de Aprendizagem**. 2010. 114 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual do Rio Grande do Norte, Universidade Federal Rural do Semi-Árido, Mossoró, 2010.
 13. SAMPAIO, C. H. C.; FARIAS, O. S.; LABIDI, S. Using Agents for Detection of Frauds in Municipal Taxes. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS - ISDA, 7. p. 100–108, 2007.
 14. SCORM. ADL SCORM. **Site Oficial do SCORM**, 2013. Disponível em <<http://www.adlnet.gov/capabilities/scorm>>. Acesso em: 08 jan. 2013.
 15. SILVA, L. C. N. **MobiLE - Um Ambiente Multiagente de Aprendizagem Móvel para Apoiar a Remomendação Ubíqua de Objetos de Aprendizagem**. 2012. 108 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual do Rio Grande do Norte, Universidade Federal Rural do Semi-Árido, Mossoró, 2012.
 16. SOLIMAN, M.; GUETL, C. Intelligent Pedagogical Agents in Immersive Virtual Learning Environments: A Review. In: PROCEEDINGS OF THE INTERNATIONAL CONVENTION - MIPRO. Opatija, Croatia, p. 827-832, 2010.
 17. STROBEL, J.; VAN BARNEVELD, A. When is PBL More Effective? A Meta-synthesis of Meta-analyses Comparing PBL to Conventional Classrooms. **Interdisciplinary Journal of Problem-based Learning**, v. 3, n. 4, 2009.
 18. TAVARES, D. G.; EICHLER, J.; PEREIRA, L. F.; SILVA, T. S.; CUNHA, A. L. F.; MARTINS, R. F. A.; CYSNEIROS, L. M.; WERNECK, V. M. B. Processo de Desenvolvimento do Sistema Multi-Agentes Monitor Glicêmico. In: CADERNOS DO IME – SÉRIE INFORMÁTICA, v. 28, p. 33–40, 2009.
 19. VERONESE, G.; CORREA, A.; WERNER, C.; JEZINI NETTO, F. **ARES: Uma Ferramenta de Engenharia Reversa Java-UML**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE – SBES, 16. Gramado: p. 347-352, 2002.
 20. YU, E. S. K. Modelling Strategic Relationships for Process Reengineering. 1995. 131 f. Tese (Doutorado em Filosofia) - Universidade de Toronto, Toronto, Canadá, 1995.