TÍTULO DO PROJETO

ENGENHARIA DE SOFTWARE BASEADA EM AGENTES INTELIGENTES PARA SISTEMAS DE COMPUTAÇÃO PERVASIVA

TÍTULO DO PLANO

SUPORTE FERRAMENTAL NO DESENVOLVIMENTO DE SOFTWARE BASEADO EM AGENTES PARA SISTEMAS DE COMPUTAÇÃO PERVASIVA

ORIENTADOR

Carla Taciana Lima Lourenço Silva Schuenemann

GRUPO DE PESQUISA CADASTRADO NO CNPq

Grupo de Pesquisa em Computação Aplicada – *Applied*

2009

1. Introdução

A cada dia os computadores e o processamento de informação se tornam mais simples e cada vez mais presentes no nosso dia-a-dia. O visionário Mark Weiser (1991) acreditava em um mundo de dispositivos completamente conectados com redes sem fio baratas onde a informação fosse acessível de qualquer lugar. Um mundo no qual computadores se tornassem invisíveis e indistinguíveis na vida diária: "a qualquer hora, em qualquer lugar e sempre" (Weiser, 1991).

A computação pervasiva é um avanço na computação centrada no homem, onde a tecnologia não é mais uma barreira, mas trabalha para nós, se adaptando às nossas necessidades e preferências e permanecendo nos "bastidores" até que sejam necessários. Aplicações dotadas de capacidades adaptativas e de sensibilidade a contexto, tais como Edifícios Inteligentes, Ambientes monitorados em tempo-real, Sistemas de cuidados com a saúde e Gestão de desastres, podem se beneficiar da computação pervasiva. Estas aplicações executam em ambientes instrumentados com sensores onde os dispositivos se comunicam através de uma rede wireless. Hoje, vários elementos da computação pervasiva estão começando a aparecer e a serem úteis por si só, já que houve um aumento na quantidade de dispositivos e objetos que possuem um endereço de rede (têm um único identificador) e estão conectados (geralmente sem fio) (Ley, 2007). Para desenvolver sistemas pervasivos, precisamos tanto destes dispositivos de hardware, como de aplicações de software que permitam que o sistema apresente características de consciência de contexto e auto-adaptação. A recente engenharia de software orientada a agentes usa a abstração de agentes para melhorar sistemas computacionais com o mesmo propósito de diminuir o envolvimento humano na computação. De fato, agentes de software apresentam propriedades que são adequadas para desenvolver aplicações para sistemas pervasivos, tais como as propriedades inicialmente identificadas por Michael Wooldridge (2002):

- Autonomia - Agentes operam sem intervenção direta de humanos ou outros, e têm algum tipo de controle sobre suas ações e estado interno;
- Habilidade social - Agentes interagem com outros agentes (e possivelmente humanos) através de alguma linguagem de comunicação de agentes;
- Reatividade. Agentes percebem seu ambiente e respondem de forma temporizada à mudanças que ocorrem nele;
- Proatividade. Agentes não agem simplesmente em resposta ao seu ambiente; eles estão aptos a exibir comportamento dirigido por metas pela tomada de iniciativa.

Para desenvolver tais sistemas, metodologias e ferramentas apropriadas têm de ser desenvolvidas. Tropos (Castro et al., 2002; Giorgini et al., 2005) propõe uma metodologia de desenvolvimento de software orientado a agentes para ambientes dinâmicos e distribuídos. Para aumentar desenvolver sistemas multi-agentes flexíveis, fáceis de reusar e manter, vários padrões de projeto orientados a agentes foram propostos (Hayden et al., 1999; Kolp et al., 2005). Padrões descrevem problemas recorrentes e soluções no pojeto de software e seu uso contribui para reduzir custos do desenvolvimento e aumentar a qualidade do produto desenvolvido (Gamma et al., 1995). Tropos inclui um conjunto de padrões de projeto, chamados de padrões sociais (Kolp et al., 2005) – um conjunto de padrões de projeto capturando aspectos sociais e intencionais que são recorrentes no projeto de sistemas multi-agentes e cooperativos. Os padrões propostos são inspirados pelos padrões federados apresentados em (Hayden et al, 1999). Uma categoria particular de padrões sociais são os padrões de mediação. Estes padrões apresentam agentes intermediários que ajudam outros agentes a cumprir acordos sobre troca de serviços. Eles incluem padrões tais como *monitor*, *broker*, *mediator*, *wrapper*, *embassy* e *matchmaker* (Kolp et al., 2005).

Embora seja possível reusar soluções de projeto pelo uso de padrões de mediação, a maneira atual de se instanciar estes padrões no desenvolvimento de sistemas multi-agentes, torna o núcleo da aplicação altamente acoplado com a implementação dos padrões, reduzindo as oportunidades de reuso tanto dos módulos da aplicação como da implementação dos padrões (Noda, N., Kishi, 2001). Esta observação foi feita no contexto de uma implementação de padrões de mediação usando JADE (Bellifemine et al., 2003), uma plataforma de programação de agentes baseada em Java.

Para tratar desta questão, os padrões foram descritos usando uma técnica de descrição de padrões de projeto orientado a agentes, chamada de *Agent Pattern Specifications* (APS) (Silva et al., 2007), que leva em consideração a separação de interesses relacionados ao padrão no nível de projeto de sistemas multi-agentes. Um interesse é alguma parte do problema que nós queremos tratar como uma unidade conceitual única (Dijkstra, 1976). Além disso, algumas diretrizes de mapeamento foram definidas para

guiar a implementação dos padrões descritos com a técnica APS através da integração de da plataforma JADE com o ambiente de implementação orientado a aspectos AspectJ (Kiczales et al., 2001), que é uma extensão orientada a aspectos da linguagem de programação Java (Sun, 2008). Esta implementação foi avaliada em termos de um conjunto de métricas para avaliar atributos de software bem conhecidos, tais como separação de interesses, acoplamento, coesão e tamanho (Sant'Anna, 2008) e mostrou que soluções orientadas a aspectos para padrões de mediação melhoram a separação de interesses relacionados a padrões. Estes resultados foram apresentados em (Silva et al., 2009) e destacam a importância de descrever os padrões usados uma abordagem como a técnica APS, que promove uma avançada separação de interesses relacionados aos padrões no projeto e, conseqüentemente, na implementação de sistemas multi-agentes.

Esta pesquisa inclui a descrição de todos os padrões sociais, incluindo a outra categoria de padrões sociais, chamada de padrões em pares, usando a técnica APS, de forma a prover um catálogo de soluções reusáveis para o projeto de sistemas multi-agentes. Para facilitar a descrição e uso destes padrões, deverá ser criada uma ferramenta de modelagem de sistemas multi-agentes baseada nas técnicas de modelagem de projeto detalhado (Silva et al., 2007a) adotada pelo Tropos. Esta ferramenta deverá ser capaz de gerar código fonte nas linguagens JADEX (Braubach et al., 2004) e AspectJ a partir dos modelos construídos com a linguagem de modelagem adotada. JADEX é um framework de software para a criação de agentes orientados a metas, isto é, agentes proativos que seguem o modelo de arquitetura interna Belief-Desire-Intention (BDI) (Rao et al., 1995). De fato, para apoiar a fase de projeto detalhado do Tropos, a integração entre JADEX e AspectJ ainda precisa ser feita, bem como o mapeamento entre a linguagem de modelagem de projeto detalhado e estas plataformas de implementação. O resultado deste projeto de pesquisa será uma metodologia de desenvolvimento de software inteligente para sistemas pervasivos, chamada Tropos-pervasivo, que será apoiada por ferramentas CASE (*Computer Aided Software Engineering*). O desenvolvimento de software baseado em agentes seguindo a metodologia Tropos- pervasivo permitirá atender a demanda por sistemas pervasivos, que requerem softwares cada vez mais flexíveis, auto-adaptáveis e distribuídos.

2. Fundamentação Teórica

2.1. Computação Pervasiva

Em 1991, Mark Weiser escreveu sobre computação pervasiva: "As tecnologias mais profundas são aquelas que desaparecem. Eles tecem-se no tecido da vida quotidiana até que se tornam indistinguíveis dela. Este desaparecimento é uma conseqüência básica não da tecnologia, mas da psicologia humana. Sempre que as pessoas aprendem algo suficientemente bem, elas deixam de estar cientes daquilo. Quando você olha para o semáforo da rua, por exemplo, você absorve a informação do semáforo sem executar conscientemente a ação de ler de o semáforo"(Weiser, 1991).

A computação pervasiva engloba a maioria das áreas da Tecnologia da Informação e alcançá-la dependerá de vários fatores que vêm juntos (Ley, 2007):

- Miniaturização (processadores menores e com menor poder computacional; tecnologias de sensores e dispositivos sem fio);
- Conectividade pervasiva;
- Interoperabilidade (padrões para redes e dispositivos; identificação; descoberta de rede e dispositivo; redes auto-configuráveis e sem emendas; etc.);
- Interfaces inteligentes melhoradas (interfaces naturais; agentes inteligentes; tecnologias de display; etc.);
- Sistemas Inteligentes (incluindo redes de sensores; consciência de contexto; localização; redes semânticas; manipulação de dados; busca; etc.);
- Segurança e confiabilidade (sistemas seguros e confiáveis; e características de privacidade).

Os elementos chave que os nós/objetos/dispositivos em um ambiente de computação pervasiva requerem são (Ley, 2007):

- Identificação: Para fazer com que objetos e dispositivos se tornem parte amplamente útil de uma rede de compartilhamento de informação inteligente, é vital que cada um deles tenha uma identidade única;

- Localização: A habilidade de objetos e dispositivos possuírem informação de localização adiciona outro nível importante de inteligência, permite a descoberta de pessoas, objetos e recursos e a possibilidade de localização baseada em ferramentas e serviços;
- Sentidos: ter uma identidade e informação de localização possibilita uma variedade de aplicações e usos, mas adicionar uma capacidade de sentir pode dar "olhos e ouvidos" aos sistemas, criando redes que podem coletar um leque de dados e até responder a eventos;
- Conectividade: conectividade sem fio é a chave para possibilidade a computação pervasiva.

Uma plataforma de computação pervasiva, chama For-All (Brito, 2008), será desenvolvida no Grupo de Pesquisa em Computação Aplicada (*Applied*) do Campus IV da UFPB e os softwares-baseados em agentes projetados com as técnicas adotadas pela metodologia Tropos serão construídos para executar nesta plataforma.

2.2. O paradigma da orientação a agentes

Embora muitas diferentes perspectivas de orientação a agentes tenham sido descritas e discutidas, ainda não há uma definição amplamente aceita do que exatamente determina um agente. Neste projeto, adotamenos a seguinte definição de agentes (Wooldridge, 2002):

> "*Um agente é um sistema computacional encapsulado que está situado em algum ambiente, e que é capaz de agir de forma flexível e autônoma neste ambiente a fim de alcançar seus objetivos de projeto*"

Baseando-nos uma pesquisa bibliográfica, estabeleceu-se as seguintes propriedades de agentes com sendo aquelas que precisam ser consideradas para construir um software orientado a agentes (Silva et al., 2003; Silva et al., 2004):

- *Autonomia*: ver definição na seção 1;
- *Proatividade*: ver definição na seção;
- *Reactividade*: ver definição na seção 1;
- *Habilidade social:* ver definição na seção 1;
- *Organisação*: Uma organização consiste de um grupo de agentes cujo comportamento e relacionamentos são regulados por um conjunto de normas sociais com o objetivo de criar uma unidade capaz de atingir uma meta comum;
- *Interação*: habilidade de se comunica com o ambiente bem como com outros agentes. A interação de agente pode ser classificada como: Comunicação, que é a habilidade de trocar mensagens com outros agentes; Cooperação, que é a habilidade de interagir com outros agentes para atingir um propósito comum; e Competição, que é a habilidade de interagir com outros agentes onde o sucesso de um agente implica na falha de outros agentes (é o oposto de cooperação);
- *Coordenação*: habilidade de executar alguma atividade em um ambiente compartilhado com outros agentes, determinando metas que eles compartilham e tarefas comuns, evitando conflitos desnecessários e centralizando recursos;
- *Negociação*: habilidade de interagir com outros agentes a fim de alcançar um acordo sobre algum problema. Envolve a troca de informação, o relaxamento de metas iniciais, concessões mútuas, mentiras ou ameaças;

2.3. O Framework Tropos

O aumento no interesse em agentes de software e sistemas multi-agentes tem levado ao desenvolvimento de metodologias baseadas nos conceitos de agentes. Em particular, este projeto envolve a consolidação do framework Tropos (Castro et al., 2002; Giorgini et al., 2005). Tropos apóia cinco fases do desenvolvimento de software:

- Requisitos iniciais: a análise de requisitos iniciais foca nas intenções dos *stakeholders*. Preocupa-se com o entendimento de um problema pelo estudo de um ambiente organizacional;
- Requisitos finais: O sistema a ser desenvolvido é descrito dentro do seu ambiente de execução, juntamente com funções e qualidade relevantes (ex. desempenho, segurança, disponibilidade). A análise de requisitos finais resulta em uma especificação de requisites que descreve todos os requisitos funcionais e não-funcionais do futuro sistema;

- Projeto arquitetural: A arquitetura global do sistema é definida em termos de sub-sistemas, interconectados através de dados, controle e dependências. Um estilo arquitetural específico é selecionado dentre um catálogo usando com critério as qualidades desejadas do sistema;
- Projeto detalhado: A fase de projeto detalhado introduz detalhes em cada componente arquitetural do sistema. Consiste na definição de como os componentes presentes no modelo arquitetural vão cumprir suas responsabilidades de acordo com os padrões de projeto;
- Implementação: É realizado um mapeamento entre a especificação do projeto detalhado e os elementos da plataforma de implementação. Esse mapeamento gera um esqueleto para a implementação.

2.4. Padrões Sociais

Os padrões sociais são classificados em duas categorias (Kolp et al., 2005): Padrões em Pares e Padrões de Mediação. Os padrões em pares descrevem interações diretas entre agentes que negociam e incluem os padrões *Booking*, *Subscription*, *Call-For-Proposals* e *Bidding*. Os padrões de mediação, conforme explicado na seção 1, incluem os padrões *Monitor*, *Broker*, *Mediator*, *Embassy*, *Matchmaker* e *Wrapper*. Os padrões sociais estão descritos a seguir:

- O padrão **Booking** envolve um cliente e um número de provedores de serviço. O cliente emite um pedido para reservar alguns recursos de um fornecedor de serviços. O fornecedor pode aceitar o pedido, negá-lo, ou propor-se a colocar o cliente em uma lista de espera, até que o recurso solicitado fique disponível quando algum outro cliente cancelar uma reserva;
- O padrão **Subscription** envolve um agente de páginas amarelas e um número de provedores de serviço. Os provedores anunciam seus serviços se registrando nas páginas amarelas. Um provedor que não queira mais ser anunciado pode solicitar a sua remoção das páginas amarelas;
- O padrão **Call-For-Proposals** envolve um iniciador e vários participantes. O iniciador lança uma chamada de propostas para um serviço a todos os participantes e, então, aceita as propostas que ofereçam o serviço a um custo específico. O iniciador seleciona um participante para fornecer o serviço;
- O padrão **Bidding** envolve um iniciador e vários participantes. O iniciador organiza e lidera o processo de leilão e recebe propostas. A cada interação, o iniciador publica o lance atual; Ele pode aceitar a oferta, aumentar o lance ou cancelar o processo;
- O padrão **Monitor**, assinantes se inscrevem para receber, de um agente monitor, notificações de mudanças de estado em alguns indivíduos de seu interesse. O monitor aceita inscrições, solicita notificações dos indivíduos de interesse, recebe tais notificações de eventos e alerta os assinantes para eventos relevantes. O indivíduo provê notificações de mudanças de estado como solicitado. O assinante se inscreve para notificação de mudanças de estado em indivíduos distribuídos, recebe notificações com a informação do estado atual, e atualiza a informação do seu estado local;
- No padrão **Broker**, o agente *broker* é um árbitro que media o acesso a serviços de um agente (provedor) para satisfazer a solicitação de um cliente. O agente *broker* localiza provedores que correspondem à requisição de um cliente para o serviço. Ele solicita e recebe o serviço dos provedores e então repassa o serviço para o cliente;
- No padrão **Matchmaker**, um agente *matchmaker* localiza um provedor que corresponde a uma requisição do cliente para um serviço e, então, dá ao cliente o acesso direto ao provedor escolhido. É o oposto do *broker*, que lida diretamente com todas as interações entre o cliente e o provedor. A negociação para o service e o real fornecimento do serviço são duas fases diferentes;
- No padrão **Mediator**, um agente mediador media interações entre agentes diferentes. Um iniciador aborda o mediador em vez de pedir diretamente a outro colega, o executor. O mediador tem modelo de conhecimento dos colegas e coordena a cooperação entre eles. Inversamente, cada executor tem um modelo de conhecimento do mediador. Enquanto um *broker* simplesmente media interações entre provedores e cliente, um mediador a mediador encapsula interações e mantêm modelos do comportamento dos iniciadores e executores ao longo do tempo;

- O padrão **Embassy**, uma embaixada faz o roteamento de um serviço solicitado por um agente estrangeiro para um agente local e manipula a resposta de volta. Se o acesso ao agente local é concedido, o agente estrangeiro pode submeter mensagens à embaixada para tradução. O conteúdo é traduzido de acordo com uma ontologia padrão. Mensagens traduzidas são repassadas para agentes locais alvo. Os resultados da busca são traduzidos e passados de volta para o agente estrangeiro;
- O padrão **Wrapper** incorpora um sistema legado a um sistema multi-agentes. O agente *wrapper* faz interface com os clientes do sistema legado, atuando como um tradutor entre eles. Isto garante que os protocolos de comunicação são respeitados e o sistema legado permanece desacoplado do resto do sistema de agentes.

2.5. A técnica Agent Pattern Specifications

A técnica APS (Agent Pattern Specifications) foi proposta em (Silva et al., 2007) para descrever padrões sociais suando abstrações e mecanismos oferecidos pela orientação a aspectos (Kiczales et al., 1997) para apoiar a separação avançada de interesses em sistemas multi-agentes. Esta técnica inclui um *template* usado para especificar os padrões, bem como diagramas de especificação da estrutura do sistema e da comunicação e dos planos dos agentes. O *template* usado na técnica APS é um sub-conjunto do *template* proposto por Gamma et al. (1995). Um exemplo complete deste *template* pode ser visto na Tabela 1.

Tabela 1. O template para descrição de padrões sociais

Elemento	**Descrição**
Nome	Padrão Monitor
Intenção	Este padrão monitor indivíduos de interesse para mudanças de estado e notifica agentes sobre esta mudança.
Aplicabilidade	O padrão monitor é adequado se um número indefinido de agentes é dependente de um indivíduo de interesse ou se é desejável que o indivíduo de interesse seja desacoplado de agentes dependentes que serão notificados de suas mudanças.
Exemplo de Motivação	Manter a consistência em um sistema distribuído é um desafio. Forte acoplamento preserva consistência, mas compromete manutenção e reutilização. Neste modelo, os agentes que devem permanecer atualizados com relação a um indivíduo, notificam o monitor do seu interesse no indivíduo através de uma inscrição para notificação de mudanças particulares neste indivíduo. O monitor então solicita notificação de mudanças de estado do indivíduo. Quando o indivíduo muda, ele notifica o monitor sobre a informação do seu novo estado. O monitor então notifica os agentes assinantes para quem a mudança de estado é relevante. Este acordo mantém consistência com desacoplamento entre o monitor, os assinantes e os indivíduos de interesse.
Problema	Um agente é dependente de outro agente, mas tem de estar desacoplado do agente dependido e, mesmo assim, notificado de suas mudanças.
Solução	Inscrição, solicitação de notificação, a notificação em si e revogação de inscrição são todos requerido por este padrão.
Participantes	Este padrão envolve pelo menos um monitor, um número de agentes assinantes e pelo menos um indivíduo ou evento de interesse. O papel do monitor é aceitar inscrições, solicitar notificações de indivíduos de interesse, receber tais notificações de eventos e para alertar assinantes sobre eventos relevantes. O papel do indivíduo é fornecer notificações de mudança de estado conforme solicitado. O papel do assinante é se inscrever para notificações de mudanças de estado em indivíduos distribuídos (dados ou objetos), receber notificações com a informação do estado atual, e atualizar a sua informação de estado local. Um indivíduo pode ter muitos assinantes, cada um dos quais se podem manifestar interesse em diferentes mudanças de estado. Os assinantes são notificados quando o evento ou mudança de estado de interesse ocorre. A

	notificação inclui todas as informações de estado necessárias para atualizar o assinante e, assim, manter a consistência com a informação do estado.

Uma especificação estrutural de um padrão de agentes (*Structural Agent Pattern Specification* - SAPS) consiste de uma estrutura de papéis de padrão, onde um papel especifica propriedades que um elemento do modelo de projeto deve possuir se ele faz parte de uma solução do padrão. A Figura 1 mostra um SAPS que especifica soluções pata o padrão social Monitor.

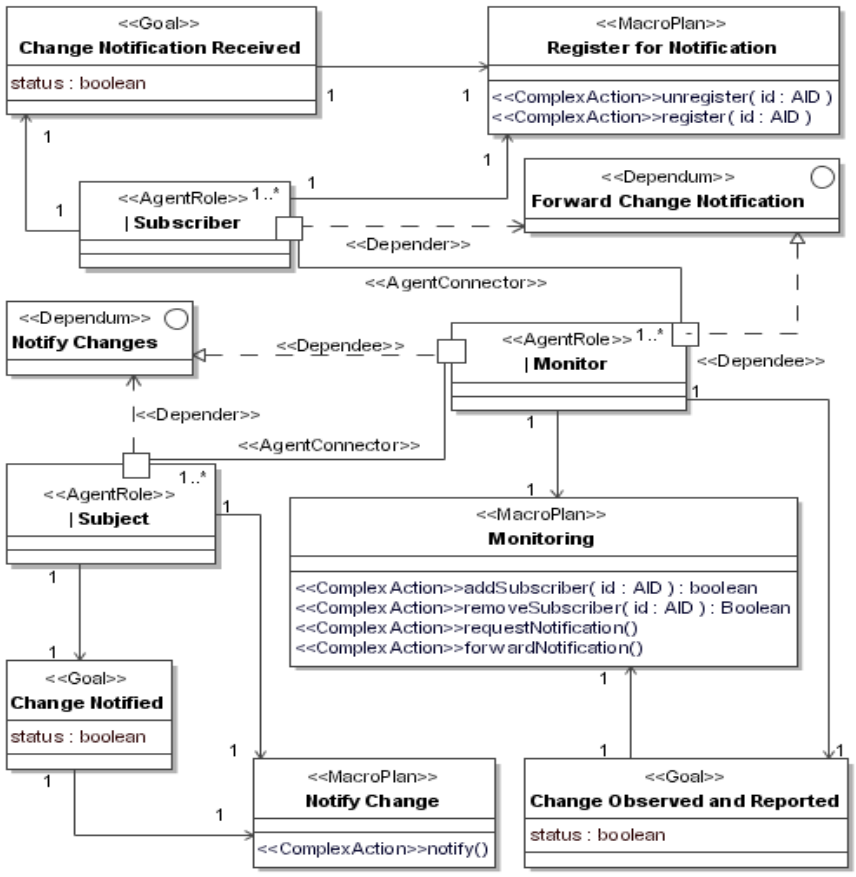

Figure 1. Especificação estrutural do padrão Monitor

O diagrama de plano é usado para descrever em que pontos os agentes serão afetados pelos aspectos que encapsulam os interesses do padrão (cada aspecto encapsula o interesse de um participante do padrão). Esta dimensão da descrição de padrões é chamada de especificação de plano de um padrão de agentes (*Plan Agent Pattern Specification* - PAPS). Na Figura 2 encontra-se o PAPS que define como compor o participante *Subscriber*, do padrão Monitor, com os agentes da aplicação.

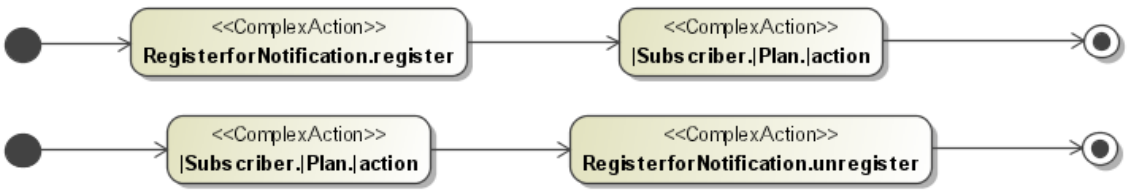

Figure 2. Um fragmento da especificação de plano do padrão Monitor

O diagrama de comunicação é outro diagrama usado para descrever os padrões de mediação. Esta dimensão da descrição de padrões é chamada de especificação de comunicação de um padrão de agentes (*Communication Agent Pattern Specification* - CAPS) e descreve um padrão de comunicações

entre os participantes do padrão. Este diagrama é apresentado na Figura 3, onde encontramos um CAPS para o padrão Monitor.

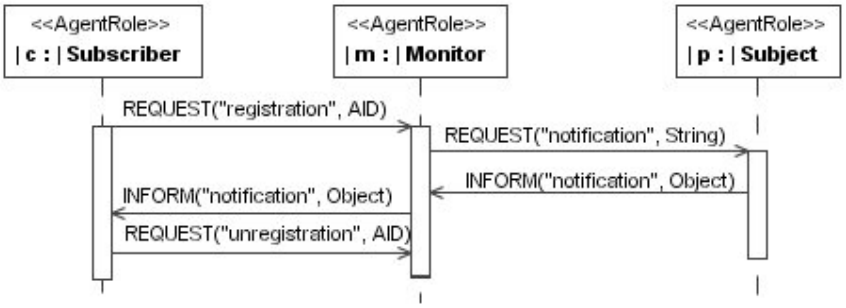

Figure 3 – Especificação de comunicação do padrão Monitor

O ultimo passo na fase de projeto detalhado de sistemas multi-agentes é selecionar e aplicar os padrões sociais para refinar o projeto arquitetural. Um dos desafios chave é escolher o padrão apropriado a ser aplicado ao projeto do sistema. Pode-se analisar o template (Tabela 1) que descreve várias características do padrão e, só então, decidir que padrão melhor se encaixa na situação em particular.

Para permitir a implementação dos padrões sociais separadamente dos interesses da aplicação, nós precisamos usar uma integração dos ambientes de implementação JADE e AspectJ. Esta integração não foi difícil, pois ambos são baseados em Java, o que facilitou a combinação dos construtores de cada linguagem no mesmo programa. Algumas diretrizes foram propostas em (Silva et al., 2007) para mapear a notação usada no diagrama estrutural da técnica APS (Figura 1) para os construtores de JADE e AspectJ. De modo geral, os agentes da aplicação são implementados como agentes da plataforma JADE e os participantes do padrão são implementados como aspectos no ambiente AspectJ, que por usa vez modificam o comportamento e a estrutura dos agentes da aplicação de acordo com o comportamento do padrão social.

3. Metodologia

Este projeto iniciará com o estudo da base teórica envolvida, incluindo o estudo da metodologia Tropos, da linguagem de modelagem de projeto orientado a agentes e da técnica APS de descrição de padrões sociais. Durante o estudo do estado da arte, haverá reuniões freqüentes do grupo de pesquisa *Applied*, no qual os membros participantes do projeto apresentarão seminários expondo os trabalhos científicos que são a fundamentação teórica do presente projeto. Após deste estudo, será feita a catalogação de todos os padrões sociais descritos com a técnica APS. Esta catalogação servirá para fazer com que os participantes do projeto amadureçam o uso da linguagem de modelagem de projeto detalhado usada para definir a técnica APS. Em seguida, será iniciado o processo de adaptação da metodologia Tropos para o desenvolvimento de software baseado em agentes para sistemas pervasivos, o que resultará na criação do Tropos-pervasivo.

Após a criação do Tropos-pervasivo, serão desenvolvidas ferramentas para modelagem e implementação de sistemas multi-agentes que dêem suporte a todas as fases da metodologia Tropos-pervasivo. Estas ferramentas serão construídas com base do ambiente ECLIPSE (2009), que permite construir ambientes de modelagem de software a partir de metamodelos, como também facilita a geração de código, a partir dos modelos de software especificados nestes ambientes, para linguagens estendidas do Java, tal como o JADE, o JADEX e o AspectJ. Além disso, construir o ambiente de modelagem de projeto detalhado usando ECLIPSE permitirá uma futura integração com o ambiente IstarTool (Siqueira et al., 2008), que é baseado no ECLIPSE e permite a especificação de modelos de requisitos para o Tropos.

Tão logo o suporte ferramental da metodologia Tropos-pervasivo tiver sido concluído, poderá ser iniciado o desenvolvimento de software baseado em agentes para um sistema pervasivo. A plataforma de computação pervasiva alvo deste projeto é a plataforma For-All (Brito et al., 2008), que estará sendo construída em paralelo a este projeto em um outro projeto de pesquisa liderado pelo Prof. Alisson Brito, do grupo de pesquisa *Applied*. Após o desenvolvimento do software, este será

implantando na plataforma For-All, que já deverá ter sido concluída, e em seguida serão realizados alguns testes e demonstrações do sistema multi-agentes pervasivo em funcionamento.

Para a realização das etapas citadas acima, algumas tecnologias e ambientes de software serão utilizados, todos gratuitos e disponíveis para *download*:

- Ambiente de desenvolvimento ECLIPSE;
- Plataforma de implementação JADEX;
- Ambiente de programação AspectJ;
- Ferramenta de Modelagem de Requisitos IstarTool.

4. Referências

Brito, A. V., Rharon Guedes, R., Cardoso, D., Silva, P., Leite, J., "Projeto For-All – Computação para todos, em todos os lugares". CONNEPI. Fortaleza, Ceará, 2008.

Bellifemine, F., Caire, G., Poggi, A., Rimassa, G., "JADE: A software framework for developing multi-agent applications. Lessons learned", *Information and Software Technology*, Elsevier, 2008.

Braubach, L., Pokahr, A. and Lamersdorf, W., "Jadex: A Short Overview", *Main Conference Net.ObjectDays*, AgentExpo, 2004.

Castro, J. Kolp, M. and Mylopoulos, J., "Towards Requirements-Driven Information System Engineering: The Tropos Project". *Information Systems*, Elsevier, Amsterdam, The Netherlands, v.27, n.6, p.365 - 389, 2002.

Dijkstra, E. "*A Discipline of Programming"*, Prentice-Hall, 1976.

Eclipse. The Eclipse Wiki, último acesso em Abril de 2009: http://wiki.eclipse.org

Gamma, E., Helm, R., Johnson, R. and Vlissides, J., "*Design Patterns: Elements of Reusable Object-Oriented Software"*, Addison-Wesley, 1995.

Giorgini, P., Kolp, M., Mylopoulos, J. and Castro, J., "Tropos: A Requirements-Driven Methodology for Agent-Oriented Software", Book Chapter in Agent-Oriented Methodologies. ed.: Idea Group, pp. 20-45, 2005.

Hayden, S., Carrick, C. and Yang, Q., "Architectural design patterns for multiagent coordination", *3rd International Conference on Autonomous Agents*, Seattle, USA, 1999.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J. and Irwin, J., "Aspect-Oriented Programming", *11th European Conference Object-Oriented Programming*, Springer-Verlag, Finland, 1997.

Kolp, M., Do, T., Faulkner, S. and Hoang, H., "Introspecting Agent Oriented Design Patterns", In Chang, S. K (ed.): *Advances in Software Eng. and Knowledge Engineering*, Vol. 3: Recent Advances, World Scientific, pp. 105-134, 2005.

Ley, D., "Ubiquitous Computing", Emerging technologies for learning, 2nd Vol, Chapter 6. British Educational Communications and Technology Agency (Becta), 2007.

Noda, N. and Kishi, T. "Implementing Design Patterns Using Advanced Separation of Concerns", Proceedings of OOPSLA 2001, Workshop on Advanced Separation of Concerns in Object-Oriented Systems, Tampa Bay, FL, 2001.

Rao, A.S. and Georgeff, M.P., "BDI agents: from theory to practice", Technical Note 56, Australian Artificial Intelligence Institute, 1995

Sant'Anna, C., Lobato, C., Kulesza, U., Chavez, C., Garcia, A., Lucena, C., "On the Quantitative Assessment of Modular Multi-Agent Architectures", *International Journal of Agent-Oriented Software Engineering, Special Issue on Multiagent Systems and Software Architecture*., Vol. 2, pp. 34–61, 2008.

Silva, C., Pinto, R., Castro, J., Tedesco, P., "Requirements for Multi-Agent Systems", Proceedings of the WER03 - VI Workshop em Engenharia de Requisitos, 2003, Piracicaba, Brasil.

Silva, C., Tedesco, P. C, Castro, J., Pinto, R., "Comparing Agent-Oriented Methodologies Using a NFR Approach", SELMAS 2004 - Software Engineering for Large-Scale Multi-Agent Systems, 2004.

Silva, C., Castro, J., Tedesco, P., Araújo, J., Moreira, A., Mylopoulos, J., "Improving Multi-Agent Architectural Design", In Choren, R., Garcia, A., Giese, H., Leung, H., Lucena, C., Romanovsky, A. (eds.): *Software Engineering for Multi-Agent Systems V: Research Issues and Practical Applications*. LNCS, Vol. 4408, Springer-Verlag, pp. 165–184, 2007.

Silva, C., Araújo, J., Moreira, A., Castro, J., "Designing Social Patterns using Advanced Separation of Concerns", In Krogstie, J., et al. (eds.): *19th Intl. Conf. on Advanced Information Systems Engineering* (CAiSE'07). LNCS, Vol. 4495. Springer-Verlag, pp. 309–323, 2007.

Silva, C., Castro, J., Araújo, J., Moreira, A., Tedesco, P., Mylopoulos, J., "Advanced Separation of Concerns in Agent Oriented Design Patterns", International Journal of Agent-Oriented Software Engineering (IJAOSE), 2009.

Siqueira, B., "IStar Tool - Uma proposta de ferramenta para modelagem de i*", Dissertação de Mestrado. Centro de Informática. Universidade Federal de Pernambuco, 2008.

Sun, "*Java Reference Documentation",* Available in http://java.sun.com/reference/docs/index.html, Last access in 03/2008.

Weiser, M., "The Computer for the 21st Century". In: Scientific American 265, Nr. 3, pp. 94-101, 1991.

Wooldridge, M., "Introduction to Multiagent Systems", John Wiley and Sons, New York, 2002.

PLANO DE TRABALHO

SUPORTE FERRAMENTAL NO DESENVOLVIMENTO DE SOFTWARE BASEADO EM AGENTES PARA SISTEMAS DE COMPUTAÇÃO PERVASIVA

1. Objetivos específicos, relevância, vinculação ao projeto e resultados esperados

Objetivos específicos:

Este plano de trabalho tem como objetivo principal criar ferramentas de suporte para uma metodologia de desenvolvimento de software baseado em agentes inteligente para sistemas pervasivos, chamada de Tropos-pervasivo. Para desenvolver estas ferramentas será preciso alcançar os seguintes objetivos específicos:

- Adaptar as técnicas e métodos adotados na metodologia Tropos para o desenvolvimento de software baseado em agentes para sistemas pervasivos;
- Desenvolver uma ferramenta CASE baseadas no ECLIPSE para modelagem de software multi-agentes e geração de código em JADEX e AspectJ;
- Integrar a ferramenta CASE de modelagem com a ferramenta IstarTool;
- Desenvolver software baseado em agentes com a metodologia Tropos-pervasivo;
- Implantar o software baseado em agentes desenvolvido com o Tropos-pervasivo na plataforma de computação pervasiva For-All;
- Realizar testes e demonstrações.

Relevância e Vinculação ao projeto:

O desenvolvimento de sistemas inteligentes pervasivos tem tido maior foco ultimamente devido a miniaturização dos dispositivos de hardware, ao avanço das redes sem fio e da presença cada vez mais constante e mais transparente da computação no nosso dia-a-dia. O desenvolvimento de sistemas pervasivos inteligentes permitirá que a computação esteja espalhada em objetos interconectados do ambiente ao nosso redor e atuando de forma inteligente e invisível para tornar as nossas atividades diárias mais fáceis e produtivas. Aplicações desenvolvidas com esta tecnologia melhorarão principalmente à qualidade de vida de idosos e deficientes, como também beneficiarão hospitais e escolas. No entanto, para desenvolver estes sistemas, precisa-se de metodologia apropriadas e apoiadas por ferramentas CASE. O desenvolvimento de ferramentas CASE para a metodologia Tropos-pervasivo é o objetivo principal deste plano de trabalho. A metodologia Tropos-pervasivo será uma adaptação do framework Tropos, que tem sido alvo dos esforços do grupo LER (Laboratório de Engenharia de Requisitos), desde 2001, juntamente com pesquisadores da Itália e Canadá. O LER faz parte do Centro de Informática (CIn), na Universidade Federal de Pernambuco (UFPE). O grupo LER é formado por pesquisadores nas áreas de engenharia de requisitos, rastreamento de requisitos, modelagem de software e sistemas multi-agentes.

Nos últimos 8 anos, a proponente deste plano de trabalho tem feito parte do grupo LER e desde Abril de 2009 também faz parte do grupo de pesquisa *Applied*, do Centro de Ciências Aplicadas e Educação (CCAE), da Universidade Federal da Paraíba (UFPB). O grupo *Applied* é formado por pesquisadores nas áreas de sistemas distribuídos, engenharia de software, informática para educação e computação móvel, inteligente e pervasiva.

A realização deste plano de trabalho também visa estreitar os laços entre os grupos de pesquisa LER e *Applied*, de forma a promover uma sinergia entre as áreas de atuação dos dois grupos. Em particular, este plano de trabalho irá criar ferramentas CASE de apoio a uma metodologia de desenvolvimento orientado a agentes, chamada Tropos-pervasivo, para uma plataforma de computação pervasiva, chamada For-All.

Resultados Esperados:

A realização deste plano de trabalho pretende trazer vários benefícios para o grupo de pesquisa *Applied* no que diz respeito a: (i) intensificação da interação entre os membros do grupo; (ii) preparação de estudantes para a pesquisa científica; (iii) capacitação de estudantes em tecnologias, métodos, técnicas e ferramentas de desenvolvimento promissoras; (iv) consolidar a abordagem Tropos como metodologia de desenvolvimento de software para sistemas pervasivos; (v) desenvolvimento/evolução do suporte ferramental para esta metodologia; (vi) criação de software inteligente para a plataforma de computação pervasiva For-All; (vii) interação entre grupos de

pesquisa de diferentes centros e universidades; e (viii) escrita de artigos científicos reportando os avanços alcançados e submissão destes aos principais eventos científicos da área.

2. Cronograma de Atividades

Esta seção apresenta as metas semestrais deste plano e o cronograma de realização das atividades para alcançar as metas. No primeiro semestre de realização deste plano, a meta principal é adaptar o Tropos para a criação do Tropos-pervasivo. Para alcançar esta meta, é preciso realizar as seguintes atividades.

1ª. Atividade: Levantamento bibliográfico;
2ª. Atividade: Estudo das técnicas e métodos adotados na metodologia Tropos e definição do Tropos-pervasivo;
3ª. Atividade: Estudo da tecnologia ECLIPSE;
4ª. Atividade: Construção da ferramenta CASE para o Tropos-pervasivo;

No segundo semestre de realização deste plano, a meta principal é o desenvolvimento e implantação do software baseado em agentes. Para alcançar esta meta, é preciso realizar as seguintes atividades.

5ª. Atividade: Integração das ferramentas CASE para o Tropos-pervasivo com o IstarTool;
6ª. Atividade: Desenvolvimento de software baseado em agentes com a metodologia Tropos-pervasivo;
7ª. Atividade: Implantação do software baseado em agentes na plataforma de computação pervasiva For-All;
8ª. Atividade: Redigir e submeter artigos científicos para eventos relevantes na área.

A seguir detalhamos o período de realização das atividades do plano de trabalho. A Tabela 1 mostra todas as atividades deste plano, bem como os meses necessários para realizá-las.

Tabela 1: Cronograma de Realização das Atividades

Atividades	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
1ª												
2ª												
3ª												
4ª												
5ª												
6ª												
7ª												
8ª												

Viabilidade de Execução

Nesta seção, são apresentados detalhes referentes à instituição de ensino alvo do projeto de pesquisa. Em particular, dentro da Universidade Federal da Paraíba, o plano de trabalho será desenvolvido no Centro de Ciências Aplicadas e Educação (CCAE), situado no campus IV da referida instituição de ensino superior. A realização do plano de trabalho contará com o apoio dos pesquisadores do Grupo de Pesquisa em Computação Aplicada (*Applied*). Além disso, para apoiar as atividades envolvidas neste plano de trabalho, o CCAE dispõe de um Laboratório de Informática no qual podemos encontrar computadores, *softwares*, impressoras e materiais de consumo necessários

para execução do plano de trabalho. Além disso, o CCAE conta com uma equipe de técnicos responsáveis pela operação e manutenção do Laboratório de Informática.