

Um Processo para a Engenharia de Domínio e de Aplicações Multiagente: As Fases de Projeto de Domínio e de Aplicações

Adriana Leite², Rosario Girardi¹

¹DEINF/GESEC - Universidade Federal do Maranhão (UFMA)
Av. dos Portugueses, s/n, Campus Bacanga, CEP 65080-040 – São Luís – MA – Brasil

²DAI - Instituto Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA)
Av. Getúlio Vargas, nº 04 - Monte Castelo - São Luís-MA - CEP 65025-001
adri071c@gmail.com, rgirardi@deinf.ufma.br

Resumo. *A Engenharia de Domínio é um processo para o desenvolvimento de uma família de aplicações reusáveis em um domínio particular de problema e a Engenharia de Aplicações para a construção de uma aplicação específica da família a partir do reuso de artefatos de software previamente produzidos no processo da Engenharia de Domínio. MADAE-Pro é um processo dirigido por ontologias para a Engenharia de Domínio e de Aplicações Multiagente, o qual promove a construção e o reuso de famílias de produtos de software multiagente. Este artigo introduz uma visão geral do MADAE-Pro, enfatizando a descrição das fases de projeto de domínio e projeto de aplicações e exemplificando como as arquiteturas genéricas produzidas na primeira fase são reusadas pela segunda na construção de uma aplicação multiagente particular.*

Abstract. *Domain Engineering is a process for the development of a reusable application family in a particular domain problem, and Application Engineering, the one for the construction of a specific application based on the reuse of software artifacts in the application family previously produced in the Domain Engineering process. MADAE-Pro is an ontology-driven process for multi-agent domain and application engineering which promotes the construction and reuse of agent-oriented applications families. This article introduces an overview of MADAE-Pro emphasizing the description of its domain design and application design phases and showing how generic architectures produced from the first phase are reused in the last one for construction of a particular multi-agent application.*

1. Introdução

Um processo de desenvolvimento de software é um modelo que especifica o ciclo de vida do software, descrevendo as fases pelas quais transita o produto de software ao longo de sua concepção e desenvolvimento e a metodologia que estabelece as técnicas a serem aplicadas em cada uma das fases de acordo a um determinado paradigma de desenvolvimento [Leite e Girardi 2009].

A Engenharia de Domínio é um processo para o desenvolvimento de uma família de aplicações reusáveis em um domínio particular de problema e a Engenharia de Aplicações para a construção de uma aplicação específica da família a partir do reuso

de artefatos de software previamente produzidos no processo da Engenharia de Domínio. MADAE-Pro (“Multi-agent Domain and Application Engineering Process”) é um processo composto por dois subprocessos complementares para o desenvolvimento e o reúso de famílias de produtos de software multiagente. No MADAE-Pro, a metodologia MADEM (“Multi-agent Domain Engineering Methodology”) é usada para guiar o processo da Engenharia de Domínio Multiagente e a metodologia MAAEM (“Multi-agent Application Engineering Methodology”) [Leite, Girardi e Cavalcante, 2008a] para aquele da Engenharia de Aplicações Multiagente. Os produtos do processo MADAE-Pro são representados como fatos da base de conhecimento da ONTORMAS (“ONTOlogy driven tool for the Reuse of Multi-Agent Systems”) [Leite, Girardi e Cavalcante, 2008b]. A ONTORMAS é uma ferramenta baseada no conhecimento para suportar e parcialmente automatizar as tarefas do MADAE-Pro.

Este artigo está organizado da seguinte forma. A seção 2 descreve o processo de desenvolvimento de software MADAE-Pro. A seção 2.1 introduz o seu ciclo de vida e uma descrição geral do suporte que as metodologias MADEM e MAAEM provêm em cada uma de suas fases. A seção 2.2 detalha as tarefas particulares das fases de Projeto de Domínio e Projeto da Aplicação, juntamente com as diretrizes providas pelas metodologias. A seção 3 referencia trabalhos relacionados discutindo suas similaridades e diferenças com o MADAE-Pro. Finalmente, a seção 4, conclui o artigo com algumas considerações finais sobre o processo e trabalhos em andamento.

2. O Processo MADAE-Pro

O MADAE-Pro é um processo baseado no conhecimento. Suas fases, tarefas e produtos são conceitualizados como classes da base de conhecimento da ONTORMAS e tanto as famílias de produto de software quanto as aplicações específicas são representadas como suas instâncias.

O MADAE-Pro adota a metodologia MADEM para guiar as tarefas referentes às fases da Engenharia de Domínio Multiagente e a metodologia MAAEM para as tarefas das fases da Engenharia de Aplicações Multiagente. As metodologias MADEM e MAAEM são baseadas, respectivamente, nas técnicas para a Engenharia de Domínio e de Aplicações [Czarnecki e Eisenecker 2000] e para o desenvolvimento de sistemas multiagente [Bresciani et al. 2004] [Cossentino et al. 2004]. Para representar os conceitos, tarefas e produtos do MADAE-Pro, uma linguagem de modelagem gráfica foi integrada à ferramenta ONTORMAS. Para a especificação do domínio de problema a ser resolvido, o MADAE-Pro orienta a realização da modelagem de objetivos, papéis, interações de entidades de uma organização e prototipação da interface com usuário. As entidades possuem conhecimento e o usam para exibir comportamento autônomo. Uma organização é composta de entidades com objetivos geral e específicos que estabelecem o que a organização pretende alcançar. A execução dos objetivos específicos permite alcançar o objetivo geral da organização. Por exemplo, um sistema de informação pode ter o objetivo “satisfazer as necessidades de informação de uma organização” e os objetivos específicos de “satisfazer as necessidades de informação dinâmicas ou em longo prazo”. Os objetivos específicos são alcançados através da execução de responsabilidades que as entidades têm para a execução de papéis com certo grau de autonomia. Os papéis têm destrezas sobre uma ou um conjunto de técnicas que suportam a execução de responsabilidades. Pré-condições e pós-condições podem

necessitar serem satisfeitas para ou após a execução de uma atividade. Conhecimento pode ser requerido e produzido na realização de uma responsabilidade. Por exemplo, uma entidade pode atuar como o papel “recuperador” com a responsabilidade de executar atividades para satisfazer as necessidades de informações dinâmicas de uma organização. Outra entidade pode atuar como o papel de “filtrador” com a responsabilidade de executar atividades para satisfazer as necessidades de informações em longo prazo das organizações. Destrezas podem ser, por exemplo, as regras da organização que as entidades conhecem para acessar e estruturar suas fontes de informação. Algumas vezes, as entidades têm que se comunicar com outras entidades internas ou externas para cooperar na execução de uma responsabilidade. Por exemplo, uma entidade atuando como o papel de “filtrador” pode necessitar interagir com um usuário (entidade externa) para observar seu comportamento e requisitar a inferência do seu perfil de interesses de informação.

2.1. O ciclo de vida do MADAE-Pro

O ciclo de vida do MADAE-Pro é iterativo, incremental e dirigido por objetivos. O desenvolvimento é realizado a partir de sucessivos incrementos, com o intuito de reduzir a sua complexidade. O processo é composto por seis fases de desenvolvimento. As três primeiras, Análise de Domínio [Leite e Girardi 2009], Projeto de Domínio e Implementação de Domínio, referem-se à Engenharia de Domínio Multiagente, onde se constrói artefatos genéricos destinados ao reúso. As três últimas, Análise da Aplicação [Leite e Girardi 2009], Projeto da Aplicação e Implementação da Aplicação, referem-se à Engenharia de Aplicação Multiagente, onde se constrói uma aplicação específica a partir do reúso dos artefatos desenvolvidos na Engenharia de Domínio Multiagente. A Figura 1 ilustra o ciclo de vida do processo MADAE-Pro. Ele inicia-se a partir da decisão de desenvolvimento de uma nova família de aplicações ou de uma aplicação específica (novo objetivo geral), conforme ilustrado no losango da figura. O ciclo reinicia-se com a decisão de incrementar aquelas através da introdução de um novo objetivo específico (manutenção evolutiva) ou de alterar um objetivo específico preexistente (manutenção evolutiva e/ou corretiva). Iterações também podem ocorrer entre as fases, com o objetivo de aprimorar os produtos da modelagem.

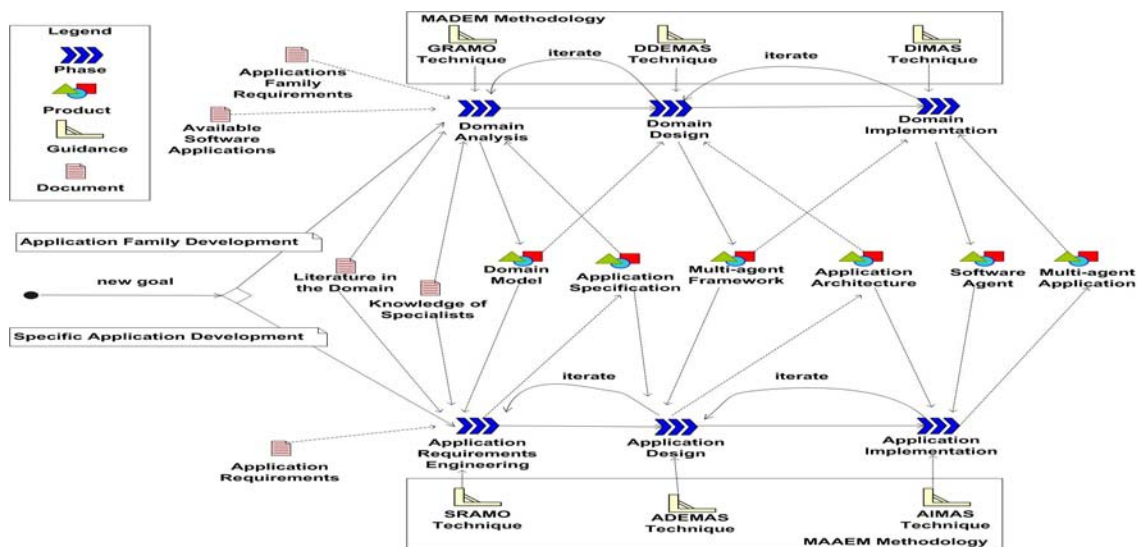


Figura 1. O ciclo de vida do MADAE-Pro

Na fase de Análise de Domínio, a metodologia MADEM guia a construção de um Modelo de Domínio, no qual são especificados os requisitos atuais e futuros de uma família de aplicações, considerando o conhecimento do domínio e as experiências de desenvolvimento extraídas a partir de especialistas e de aplicações já desenvolvidas no domínio a que pertence a família de aplicações. Nessa fase são realizadas as seguintes tarefas: Modelagem de Conceitos, onde é realizada uma “tormenta de idéias” sobre os conceitos do domínio e seus relacionamentos; Modelagem de Objetivos, na qual são identificados e representados os objetivos de uma família de aplicações, as entidades externas com as quais coopera e as responsabilidades necessárias para alcançá-los; Modelagem de Papéis, onde as responsabilidades identificadas na tarefa de Modelagem de Objetivos são associadas aos papéis que serão encarregados delas; Modelagem de Variabilidades, onde são distinguidas as características comuns das variáveis em uma família de aplicações multiagente; Modelagem de Interações entre Papéis, onde é identificado como as entidades externas e internas cooperam para alcançar um objetivo específico e a Prototipação da Interface com o Usuário, cujo objetivo é identificar as interações dos usuários com o sistema e construir um protótipo da interface. O produto desta fase é um Modelo de Domínio.

A fase de Projeto de Domínio aborda o projeto arquitetural e detalhado de modelos de frameworks multiagente provendo uma solução aos requisitos de uma família de aplicações multiagente especificadas em um Modelo de Domínio. Esta fase possui duas subfases: o Projeto Arquitetural, que estabelece um modelo arquitetural da sociedade multiagente incluindo seus mecanismos de coordenação e cooperação; e o Projeto do Agente, que define o projeto interno de cada agente, modelando sua estrutura e comportamento. Um Modelo de Framework da Sociedade Multiagente é construído como produto desta fase. Ele é composto por um Modelo do Conhecimento da Sociedade Multiagente, um Modelo da Sociedade Multiagente, um Modelo dos Mecanismos de Cooperação e Coordenação, um Modelo de Interações entre Agentes, um conjunto de Modelos do Conhecimento do Agente e um conjunto de Modelos das Ações do Agente.

A fase de Implementação do Domínio aborda o mapeamento dos conceitos de projeto para conceitos de uma determinada plataforma de implementação de agentes. Nessa fase, os agentes, suas responsabilidades e as interações modeladas na fase de projeto são mapeadas para agentes, comportamentos e atos de comunicação envolvidos no framework JADE [Bellifemine et al. 2003], o qual tem sido adotado como plataforma de implementação. Um Modelo de Implementação da Sociedade Multiagente é construído como produto desta fase da MADEM, composto de um Modelo de Comportamentos e de um Modelo de Atos de Comunicação. A modelagem de variabilidades é realizada paralelamente em todas as fases da metodologia MADEM para determinar as partes comuns e variáveis da família de aplicações. Isto é feito identificando-se “pontos de variação” e suas correspondentes “variantes”. Um ponto de variação é um conceito sujeito a variação. Uma variante representa as variações alternativas ou opcionais desse conceito.

A MAAEM é uma metodologia para a análise, o projeto e a implementação de aplicações multiagente através da reutilização de artefatos de software anteriormente produzidos na Engenharia de Domínio Multiagente. Sendo baseada no reuso composicional, ela envolve a seleção, adaptação e composição de artefatos como

Modelos de Domínios, Frameworks Multiagente e Agentes de Software para a construção de uma aplicação multiagente particular. Na fase de Engenharia dos Requisitos da Aplicação, pretende-se identificar e especificar os requisitos de uma determinada aplicação, partindo de modelos de domínio, elaborados na fase correspondente da Engenharia de Domínio Multiagente. Desse modo, a tarefa central do desenvolvedor é reusar um conjunto de requisitos da família em um domínio. Tais requisitos específicos da aplicação, quando provenientes da família, são levantados a partir da seleção dentre os requisitos comuns ou variáveis no domínio. O produto desta fase é a especificação da aplicação, composto por um Modelo de Conceitos, um Modelo de Objetivos, um conjunto de Modelos de Papéis e de Modelos de Interações entre Papéis, um para cada objetivo específico definido no Modelo de Objetivos e um Protótipo da Interface Usuário.

Na fase de projeto da aplicação, os desenvolvedores reusam soluções de projeto de uma família de aplicações e as adaptam para os requisitos específicos da aplicação em desenvolvimento. Um conjunto de modelos compondo a arquitetura da aplicação multiagente são produzidos seguindo um conjunto de tarefas de modelagem definidas na fase de projeto da aplicação da metodologia MAAEM. Esta fase consiste de duas tarefas: o projeto arquitetural, o qual define a arquitetura da sociedade multiagente e o projeto do agente, o qual define a estrutura interna de cada agente da sociedade. O projeto arquitetural consiste de cinco subtarefas: a modelagem do conhecimento da sociedade multiagente, a modelagem da sociedade multiagente, a modelagem de interações entre agentes e a modelagem dos mecanismos de cooperação e coordenação. O projeto do agente consiste de duas tarefas: a modelagem do conhecimento do agente e a modelagem das ações do agente.

Na fase de implementação da aplicação, as ações, interações e o conhecimento dos agentes são identificados e especificados em uma linguagem/plataforma de desenvolvimento de agentes. Um Modelo de Comportamentos e um Modelo de Atos de Comunicação são gerados nessa fase. Em todas as fases da metodologia MAAEM, o reúso é realizado através da identificação de pontos de variação nos produtos da metodologia MADEM e selecionando-se as variantes apropriadas.

2.2. As Tarefas das Fases de Projeto de Domínio e de Aplicações

Esta seção descreve as tarefas das fases de Projeto de Domínio e de Projeto da Aplicação do MADAE-Pro, mostrando como os artefatos de software do framework ONTOSERS [Mariano et al. 2008] são produzidos e reusados no desenvolvimento da arquitetura do sistema de recomendação de informações InfoTrib [Mariano et al. 2008]. ONTOSERS é uma família de aplicações para a recomendação de informações, que usa as abordagens de filtragem da informação baseada no conteúdo, colaborativa e híbrida. A família ONTOSERS é composta de um modelo de domínio (produto da fase de análise de domínio), um modelo de framework multiagente (produto da fase de projeto de domínio) e de um modelo de implementação da sociedade multiagente com os agentes de software executáveis correspondentes (produtos da fase de implementação do domínio). InfoTrib é um sistema de recomendação de instrumentos jurídico normativos no domínio do direito tributário que, baseado no perfil de um usuário, onde é especificado o seu interesse nas diversas espécies de tributos, o sistema provê

A tarefa de modelagem dos mecanismos de cooperação e coordenação produz um Modelo dos Mecanismos de Cooperação e Coodenação. Esse modelo especifica como os agentes da sociedade vão cooperar, organizar-se e interagir para realizar uma determinada tarefa. O padrão de cooperação determina como os agentes vão interagir para realizar suas responsabilidades. Exemplos de padrões de cooperação são o quadro-negro e o federativo [Girardi, 2004]. O padrão de coordenação especifica como os agentes se organizam para cooperar. Eles podem estar em um mesmo nível hierárquico como é o caso do padrão de mercado, onde os agentes são classificados em consumidores e vendedores ou em níveis de hierarquia diferentes como, por exemplo, o mecanismo de mestre-escravo, onde os agentes são classificados em gerentes (mestres) e em trabalhadores (escravos). Os agentes trabalhadores são coordenados por um gerente que distribui as tarefas entre estes e espera o resultado. A ONTORMAS possui um conjunto predefinido de padrões arquiteturais que são instâncias da classe “Architectural Pattern” que possuem uma descrição dos contextos em que são aplicáveis. Para a família ONTOSERS foi escolhido o padrão arquitetural em duas camadas. Uma das vantagens desse modelo é o grau de independência entre o conteúdo de cada camada. Assim cada camada precisa saber apenas: os agentes externos, com os quais irá se relacionar; as interfaces com as camadas imediatamente superior e inferior e os respectivos agentes que compõem a camada. Na modelagem de ONTOSERS, foi definida a camada “User processing”, a qual controla a interação com o usuário através do agente “User interface” e a camada “Information processing”, a qual possui o agente “Filter” que monitora a base de informações do ONTOSERS.

O Modelo de Conhecimento do Agente representa o conhecimento particular de cada agente da sociedade. Ele é construído com base no Modelo de Interações entre Papéis, o qual provê o conhecimento compartilhado entre os papéis durante as suas interações e no Modelo do Conhecimento da Sociedade Multiagente, do qual se obtém os conhecimentos específicos de cada agente. Parte do Modelo de Conhecimento do agente “Interface Usuário” do ONTOSERS é representada na Figura 5. Um ponto de variação foi colocado sobre o conhecimento “Perfil do Usuário”, que tem como variante opcional o conhecimento “User navigational behavior”, para os casos onde a aplicação utiliza a aquisição implícita do perfil do usuário.



Figura 5. Parte do Modelo do Conhecimento do agente “User Interface”

A última tarefa da fase de Projeto de Domínio é a modelagem das ações do agente que produz o Modelo das Ações do Agente. O Modelo de Ações do Agente representa as ações que o agente terá no seu ambiente de acordo com as responsabilidades de que tiver sido encarregado. Nesse modelo são representadas as percepções que o agente tem do ambiente: um evento ou as possíveis mensagens que ele poderá receber de outros agentes ou de entidades externas que irão disparar as suas ações. Existem duas formas de mapeamento dessas percepções para ações. A primeira é quando uma percepção é mapeada diretamente para uma ação. Nesse caso, o agente é

considerado tipo reativo. A outra forma é quando ocorre um planejamento e na busca da ação adequada de acordo com a percepção do agente para atingir um objetivo. Nesse caso, o agente é considerado do tipo deliberativo e usa um motor de inferência no seu processo de raciocínio. Para realizar uma ação o agente pode necessitar de uma destreza (uma destreza é uma habilidade que o agente deverá possuir para a execução de suas responsabilidades) e também atender a uma determinada pré-condição. Ao término de uma ação, uma pós-condição também pode ser necessária. Aqui os pontos de variação dos agentes em relação a responsabilidades variantes são os mesmos definidos no Modelo da Sociedade Multiagente.

As Tarefas da Fase de Projeto de Aplicações.

Na fase de Projeto da Aplicação, o desenvolvedor irá reutilizar algumas soluções de projeto relacionadas a uma família de aplicações e adaptá-la aos requisitos específicos da aplicação em desenvolvimento. Nesta fase, o reuso é suportado pela ferramenta ONTORMAS. A seleção dos artefatos de software é suportada pela recuperação semântica, onde o usuário entra com uma consulta especificando as características do produto que ele(a) pretende reusar e obtém como resultado os artefatos reusáveis do repositório, os quais satisfazem sua consulta. Depois da seleção do artefato que mais se aproxima das necessidades do usuário, este verifica se o artefato pode ser reusado integralmente ou se é necessário realizar adaptações e/ou integrá-lo com outros artefatos. A primeira tarefa desta fase é a modelagem do conhecimento da sociedade multiagente que produz o Modelo do Conhecimento da Sociedade Multiagente. Nesse modelo o conhecimento compartilhado pelos agentes da sociedade são representados em uma rede semântica que dará origem à ontologia de comunicação da sociedade multiagente. O Modelo do Conhecimento da Sociedade Multiagente do sistema InfoTrib foi desenvolvido a partir do Modelo do Conhecimento da Sociedade Multiagente de ONTOSERS (Figura 2), apenas especializando o conhecimento para o domínio tributário. A seleção de modelos da fase de projeto de domínio é bastante facilitada, pelos relacionamentos semânticos existentes entre estes e os modelos da fase de análise de domínio. Um exemplo disso são os agentes da fase de projeto que realizam um ou mais papéis da fase de análise (especificado através do relacionamento “plays”). Assim, para reusar um Modelo da Sociedade Multiagente basta realizar uma busca na ONTORMAS por agentes associados aos relacionamentos “plays” dos papéis previamente reusados na fase de análise. Na figura 6 é exemplificada uma busca na ONTORMAS por Modelos da Sociedade Multiagente, tendo como parâmetros os papéis “User modeler” e “Input interface”, obtendo-se como resultado o Modelo da Sociedade Multiagente de ONTOSERS.

The screenshot shows a query window titled "Enter a Path" with the following SPARQL query:

```
( (:instance "Multi-agent Society Model" ?model)
  (concepts ?model ?agent)
  (plays ?agent ?role1)
  (name ?role1 "User modeler")
  (plays ?agent ?role2)
  (name ?role2 "Input interface") )
```

Below the query is a "Results" table with the following data:

I...	agent	model	role1
1	User Interface. Multi-agent Society Model of ONTOSERS-DD		User modeler

Figura 6. Seleção de um Modelo da Sociedade Multiagente na ONTORMAS

Após encontrar o modelo que se deseja reusar, o próximo passo é a seleção da parte variável da família que esteja de acordo com os requisitos da aplicação. Isso é feito escolhendo-se as variantes do modelo genérico que deverão fazer parte do modelo da aplicação. Depois da seleção das variantes, pode ser necessário ainda realizar alguma adaptação no modelo. Por exemplo, novos conceitos específicos da aplicação em desenvolvimento que não estavam previamente definidos na família de aplicações multiagente podem ser adicionados. No caso do sistema InfoTrib, entre as variantes presentes no Modelo da Sociedade Multiagente de ONTOSERS (Figura 4), foi escolhida a variante do “Alternative Group 2” (“Explicit Profile Acquisition”), uma vez que um dos requisitos da aplicação, definidos no Modelo de Objetivos da fase de Análise dos Requisitos da Aplicação, é a modelagem explícita do perfil do usuário tributário, realizada através de formulários Web. Parte do Modelo da Sociedade Multiagente de InfoTrib desenvolvido a partir do reuso do modelo da sociedade multiagente de ONTOSERS (Figura 3) está representado na Figura 7, mostrando o agente “User Interface” e suas responsabilidades.

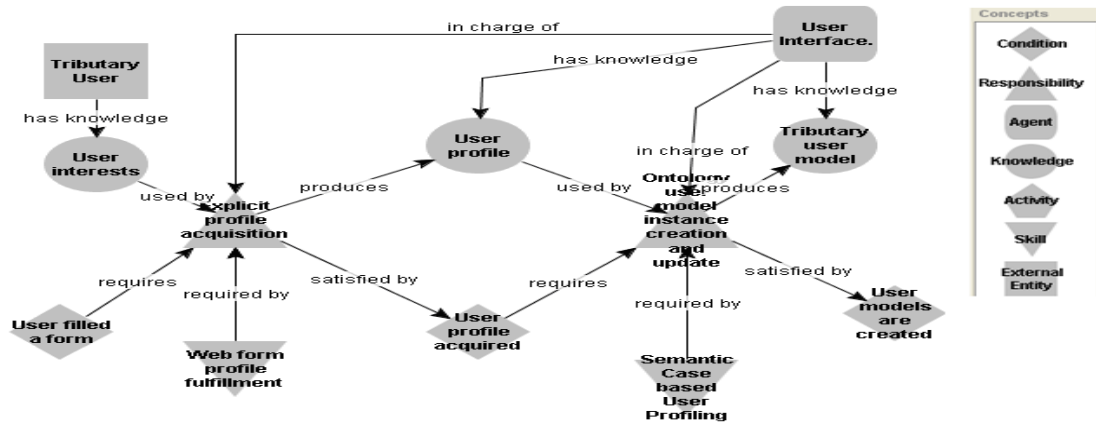


Figura 7. Parte do Modelo da Sociedade Multiagente InfoTrib, enfatizando o Agente de Interface do Usuário

Para a construção do Modelo de Interações entre Agentes são utilizados como insumos as interações presentes no Modelo de Interações entre Papéis, selecionadas de acordo com o mapeamento de papéis para agentes. Os agentes do Modelo da Sociedade Multiagente podem ser utilizados como parâmetros para selecionar um Modelo de Interações entre Agentes na base de conhecimento da ONTORMAS. As interações serão reusadas quando o agente emissor e receptor da mensagem fizer parte do Modelo da Sociedade Multiagente.

A Modelagem dos Mecanismos de Cooperação e Coordenação é realizada através da seleção de padrões arquiteturais no repositório ONTORMAS. Para tanto, a descrição do objetivo da aplicação é comparada com o problema descrito em cada padrão e com seu contexto de aplicação, no caso, o dos sistemas multiagente. Por exemplo, na escolha de um padrão arquitetural em camadas, uma ou mais camadas podem ser reusadas se fazendo as adaptações necessárias à aplicação em desenvolvimento. O Modelo dos Mecanismos de Cooperação e Coordenação do InfoTrib foi desenvolvido a partir do reuso do modelo correspondente da família ONTORSERS. Este modelo apresenta algumas diferenças, em relação ao apresentado no ONTOSERS, uma delas é a camada “User Interface” que recebe apenas os interesses do usuário informados expressamente. No Modelo de Conhecimento do Agente, o

conhecimento de cada agente da sociedade é definido. Os agentes reusados no Modelo da Sociedade Multiagente são utilizados como parâmetros da consulta na ONTORMAS, onde são selecionados Modelos de Conhecimento na base de conhecimento da ONTORMAS. O Modelo do Conhecimento do Agente de InfoTrib, ilustrado na Figura 8, foi construído a partir do Modelo do Conhecimento do Agente de ONTOSERS (Figura 5), a variante opcional "User navigational behavior" não foi selecionada por que não se aplica aos requisitos de InfoTrib (modelagem explícita). Além disso, foram adicionados alguns conceitos novos, específicos do direito tributário como "Tributary user model".

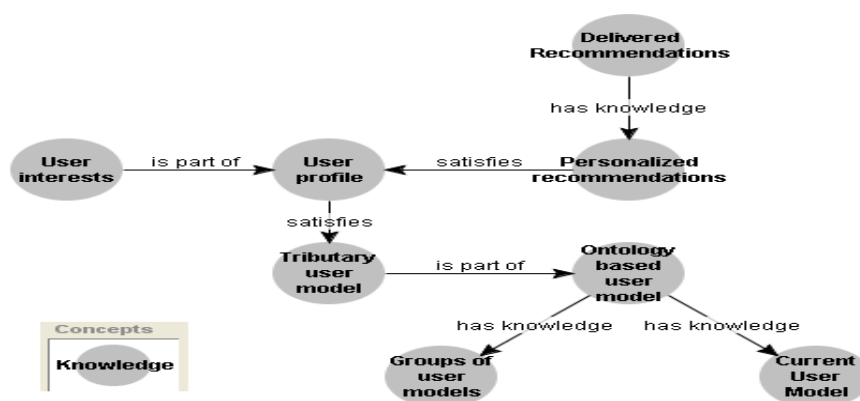


Figura 8. Modelo do Conhecimento do Agente "User Interface"

O Modelo das Ações do Agente representa as ações que o agente realiza de acordo com as suas responsabilidades e as percepções do ambiente. O agente pode ser do tipo reativo ou deliberativo. No agente reativo as ações são mapeadas diretamente a partir das percepções, já no caso do agente deliberativo a partir das percepções, ele realiza planejamento para decidir que ações tomar através de um mecanismo de inferência. No Modelo das ações do Agente, o planejamento é realizado a partir das percepções que são o conhecimento das entidades externas, do conjunto de ações, do estado atual do mundo (pré-condição), do estado desejado (pós-condição) para determinar uma seqüência de ações que leve do estado atual do mundo para o estado desejado. O Modelo de Ações do sistema InfoTrib foi construído a partir do correspondente modelo da família ONTOSERS. As adaptações foram feitas a partir da seleção da responsabilidade variante referente a aquisição explícita do perfil de usuário e dos conceitos que estavam diretamente ligados a ela, como conhecimento e pré e pós-condições. Também foi adicionado conhecimento referente ao domínio do direito tributário.

3. Trabalhos Relacionados

Algumas abordagens para o desenvolvimento de agentes, como GAIA [Zambonelli et al. 2003], PASSI [Cossentino et al. 2004] e TROPOS [Bresciani et al. 2004], já tem sido desenvolvidas para melhorar a produtividade do processo de desenvolvimento de software multiagente, a reusabilidade dos produtos gerados e a efetividade do gerenciamento do projeto de software. Algumas abordagens baseadas na Engenharia de Domínio também já foram propostas para o desenvolvimento orientado a objetos, como o processo CBD-Arch-DE (Component-Based Development – Architecture – Domain Engineering) [Blois et al.. 2004].

GAIA é uma metodologia para as fases de análise e projeto do desenvolvimento de sistemas multiagente. Tropos é uma metodologia orientada a agentes que suporta o desenvolvimento completo de sistemas multiagente. Ela é baseada no framework de modelagem organizacional i* [Bresciani et al. 2004]. PASSI (a Process for Agent Societies Specification and Implementation) é um processo para o desenvolvimento de sistemas multiagente que integra conceitos da Engenharia de Software orientada a objetos e da Inteligência Artificial. Ela permite o desenvolvimento de sistemas multiagente para propósitos específicos como agentes móveis e usa a linguagem AUML para construir seus modelos. Algumas características da GAIA, PASSI, TROPOS e MADAE-Pro são descritas a seguir. Todas as abordagens adotam o ciclo de vida iterativo, onde o produto de software passa por diversos refinamentos durante o seu processo de desenvolvimento. Com exceção da metodologia GAIA, todas as outras abordagens utilizam também o ciclo de vida incremental, onde para facilitar o entendimento do produto de software, o seu desenvolvimento é dividido em partes que vão sendo adicionadas até que o software esteja completo. Das fases de desenvolvimento de um sistema multiagente, todas as abordagens suportam a análise e o projeto, enquanto que a PASSI, TROPOS e o MADAE-Pro também suportam a fase de implementação. Que tenhamos conhecimento, somente MADAE-Pro permite o desenvolvimento de família de aplicações multiagente. Para as ferramentas de suporte ao desenvolvimento, a metodologia TROPOS é apoiada pela ferramenta TAOM-Tool [Bresciani et al., 2004], um plug-in para o ambiente de desenvolvimento Eclipse, que permite a modelagem dos sistemas utilizando o framework i*. O MADAE-Pro é suportado pela ferramenta ONTORMAS que permite a modelagem e o armazenamento de aplicações individuais e de famílias de aplicações multiagente como instância de ontologias. Atualmente a GAIA não possui ferramenta de apoio ao seu desenvolvimento. Para as atividades de reúso, GAIA e TROPOS permitem o reúso informal de modelos e código. PASSI permite o reúso de código fonte associado aos seus diagramas de classe e diagramas de atividades. O processo MADAE-Pro permite reusar tanto modelos quanto código fonte dos produtos de software.

O processo CBD-Arch-DE utiliza uma abordagem composicional para guiar as atividades arquiteturais da Engenharia de Domínio em um desenvolvimento orientado a objetos. Os artefatos produzidos no processo são diagramas de seqüência e de classes. A criação desses modelos é parcialmente suportada pelo ambiente Odyssey [Werner et al., 2000]. As tarefas do seu ciclo de vida são realizadas iterativamente. Esse processo é interessante, pois produz um conjunto de componentes reutilizáveis, no entanto, não fornece diretrizes de como esses componentes deverão ser reusados na construção de uma aplicação individual e nem possui uma fase para guiar a implementação desses componentes.

Duas principais características distinguem o MADAE-Pro de outras abordagens. Primeiro, ele provê suporte para reúso no desenvolvimento de sistemas multiagente, através da integração de conceitos da Engenharia de Domínio e da Engenharia de Aplicações. Segundo, ele é um processo baseado no conhecimento, onde os modelos de agentes e frameworks são representados como instâncias da base de conhecimento da ONTORMAS. Os conceitos da ONTORMAS estão semanticamente relacionados permitindo buscas mais efetivas e a realização de inferências, facilitando com isso o entendimento e o reúso de modelos durante o desenvolvimento de uma aplicação no domínio.

4. Considerações Finais

Este trabalho apresentou MADAE-Pro, um processo para a Engenharia de Domínio e de Aplicações Multiagente, enfatizando a descrição da sua fase de projeto de domínio e projeto da aplicação e ilustrando como uma arquitetura genérica é reusada na construção da arquitetura de uma aplicação particular. Foram utilizados exemplos extraídos da família de aplicações para recomendação de informação ONTOSERS e da aplicação para recomendação de informação tributária InfoTrib. Os artefatos de software produzidos pelo MADAE-Pro são representados como instancias da ontologia ONTORMAS, a qual serve como repositório para os artefatos reusáveis e também como uma ferramenta de suporte ao desenvolvimento das aplicações.

MADAE-Pro tem sido avaliado com estudos de casos abordando tanto o desenvolvimento de famílias de aplicações, como ONTOSERS [Mariano et al. 2008], quanto aplicações específicas, como InfoTrib [Mariano et al. 2008] e InfoNorma [Drumond et al. 2006]. A representação do MADAE-Pro na linguagem SPEM tem melhorado a clareza do processo durante o processo de desenvolvimento. Com a base de conhecimento provida pela ONTORMAS, um sistema especialista está sendo desenvolvido, para a automação de várias tarefas do MADAE-Pro, permitindo assim, um desenvolvimento mais rápido das aplicações multiagentes e a geração automática de parte do código fonte

O processo MADAE-Pro atualmente suporta somente o reuso composicional, baseado na seleção, adaptação e integração de artefatos. Uma abordagem gerativa para o reuso tem sido explorada com a especificação da metodologia GENMADEM e da ferramenta ONTOGENMADEM [Jansen e Girardi 2006] que provê suporte para criação de Linguagens Específicas de Domínio para serem usadas na geração de uma família de aplicações no domínio. Futuramente iremos estender a ONTORMAS para suportar a ONTOGENMADEM permitindo o reuso gerativo na Engenharia de Aplicações Multiagente. As técnicas de desenvolvimento para verificação e controle de qualidade dos artefatos de software produzidos, teste, o gerenciamento do ciclo de vida e sua integração ao MADAE-Pro está sendo objeto de pesquisa.

Referências

- Bellifemine, F., Caire, G., Poggi, A., Rimassa, G. (2003) "JADE A White Paper". Exp v. 3 n. 3, Sept., <http://jade.tilab.com/>.
- Blois, a. P. ; Becker, k ; Werner, C. M. L. (2004). "Um Processo de Engenharia de Domínio com foco no Projeto Arquitetural Baseado em Componentes". In: Quarto Workshop de Desenvolvimento Baseado em Componentes, João Pessoa. UFPB, 2004. v. 1. p. 15-20.
- Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A. (2004) "TROPOS: An Agent-Oriented Software Development Methodology", In: Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers Volume 8, Issue 3, pp. 203-236.
- Cossentino, M., Sabatucci, L., Sorace, S. and Chella, A., (2004) "Patterns reuse in the PASSI methodology". In: Proceedings of the Fourth International Workshop Engineering Societies in the Agents World (ESAW'03), Imperial College London, UK, pp. 29-31.

- Czarnecki, K., Eisenecker, U. W., (2000) "Generative Programming: Methods, Tools, and Applications". ACM Press/Addison-Wesley Publishing Co., New York, NY.
- Drumond, L., Girardi, R., Lindoso A., L. Marinho.(2006) "A Semantic Web Based Recommender System for the Legal Domain". Proc. of the European Conference on Artificial Intelligence (ECAI 2006) Workshop on Recommender Systems, pp. 81-83.
- FIPA. ACL Message Structure Specification. Disponível em:<<http://www.fipa.org/specs/fipa00061/SC00061G.html>>. Acessado em: 30 Maio 2009.
- Leite, A. Girardi, R. (2009) "A Process for Multi-Agent Domain and Application Engineering: the Domain Analysis and Application Requirements Engineering Phases", In: 11th International Conference on Enterprise Information Systems (ICEIS 2009), Proceedings of the 11th International Conference on Enterprise Information Systems, Ed. INSTIIC. Milan, Italy, pp. 156-161.
- Leite, A. Girardi, R. Cavalcante, U. (2008a) "MAAEM: A Multi-agent Application Engineering Methodology". In Proceedings of the Twentieth International Conference on Software Engineering and Knowledge Engineering (SEKE 2008), Ed. Knowledge Systems Institute Graduate School, Redwood City, USA, pp. 735-740.
- Leite, A. Girardi, R. Cavalcante, U. (2008b) "An ontology for Multi-Agent Domain and Application Engineering Information", In: IEEE International Conference (IRI 2008), pp. 98-103.
- Girardi, R. (2004). "Engenharia de Software baseada em Agentes", Anais do IV Congresso Brasileiro de Ciência da Computação (CBCOMP 2004), Ed. UNIVALI, Itajaí, Santa Catarina, Brasil, pp. 913-937.
- Jansen, M., Girardi, R. (2006) "GENMADEM: A Methodology for Generative Multi-agent Domain Engineering". In: The 9th International Conference on Software Reuse, 2006, Torino. Proceedings of the 9th International Conference on Software Reuse, Lecture Notes in Computer Science (LNCS), v. 4039, Berlin: Springer-Verlag, p. 399-402.
- Mariano, R., Girardi, R., Leite, A., Drumond, L. Maranhão, D. (2008) "A Case Study on Domain Analysis of Semantic Web Multi-agent Recommender Systems". In: Proceedings 3th International Conference on Software and Data Technologies, Porto. Portugal, p. 160-167.
- Zambonelli, F., Jennings, N., Wooldridge, M. (2003) "Developing multiagent systems: The Gaia methodology". ACM Transactions on Software Engineering and Methodology, pp. 417-470.
- Werner, C. M. L., Braga, R. M. M., MATTOSO, M., et al., (2000) "Infra-estruturaOdyssey: estágio atual", XIV Simpósio Brasileiro de Engenharia de Software, João Pessoa, outubro, pp.366-369.