UMA ANÁLISE SOBRE OS PROCESSOS DE DESENVOLVIMENTO PARA SISTEMAS MULTIAGENTES

Iderli Souza Filho [1]

Iderli Pereira de Souza Filho [2]

Gilleanes Thorwald Araujo Guedes [3]

Resumo:

Sistemas multiagentes (SMAs) constituem uma área da Inteligência Artificial Distribuída, que se difere da Inteligência artificial (IA) tradicional pois tenta dividir um problema em diversos subproblemas mais simples, buscando assim chegar a uma solução mais rapidamente. O conceito de agente de software segundo o artigo Introdução a Sistemas Multiagente (Silveira, 2006) diz que um agente de software é uma entidade que funciona de forma contínua e autônoma em um ambiente particular. Muitas vezes um agente de software atua no ambiente com outros agentes, sendo capaz de intervir no seu próprio ambiente, sem requerer intervenção humana. O desenvolvimento de sistemas multiagentes produziu novos desafios tanto para a Engenharia de Software como para a Inteligência Artificial, o que implicou na necessidade de criar processos de desenvolvimento de software exclusivos para este tipo de software. Podemos definir a importância de processos de software tendo como base o livro Engenharia de software (PRESSMAN, 2006), nele é dito que processos de software são importantes pois propiciam estabilidade, controle e organização. Ou seja, os processos de software evitam que o desenvolvimento ocorra de forma caótica, fazendo com que os produtos de trabalho sejam apropriados para a equipe de projeto, tendo uma garantia maior do produto ser entregue. Considerando a necessidade de processos de desenvolvimento de software particulares para sistemas multiagentes, pretendemos neste artigo demonstrar o estado da arte destes tipos de processo atualmente existentes com o objetivo de identificar as vantagens e desvantagens de cada um e identificar as suas possíveis lacunas. Como resultado da pesquisa realizada chegamos a um total de 14 processos analisados, Gaia, Gaia2Jade, RoadMap, PROMETHEUS, Mase, O-MaSE, ROMAS, ADELFE, Message, ASPECS, PASSIM, PASSI, Agile Passi e INGENIAS. Para este trabalho foi descrito 4 processos, dos 14 analisados, ASPECS (Massimo Cossentino Et al., 2014), ADELFE (Bonjean Et al, 2014), O-MaSE (Scott Deloach Et al, 2014) e PASSI (Massimo Cossentino Et al, 2014). Estes processos foram escolhidos para representar uma diferença entre os focos dos processos, visto que os descritos diferem em seus objetivos e especificações no ciclo de vida.

Palavras-chave: Engenharia de software, MAS, AOSE, ES, Processos, Sistemas multiagente, Agente,

Software

Modalidade de Participação: Iniciação Científica

UMA ANÁLISE SOBRE OS PROCESSOS DE DESENVOLVIMENTO PARA SISTEMAS MULTIAGENTES

[1] Aluno de graduação. iderlisouzaf@gmail.com. Autor principal

[2] Aluno de graduação. juninho270199iderli@gmail.com. Apresentador

[3] Docente. gilleanesguedes@unipampa.edu.br. Orientador

UMA ANÁLISE SOBRE OS PROCESSOS DE DESENVOLVIMENTO PARA SISTEMAS MULTIAGENTES

1 INTRODUÇÃO

Sistemas multiagentes (SMAs) constituem uma área da Inteligência Artificial Distribuída, que se difere da Inteligência artificial (IA) tradicional pois tenta dividir um problema em diversos subproblemas mais simples, buscando assim chegar a uma solução mais rapidamente. O conceito de agente de software segundo o artigo Introdução a Sistemas Multiagente (Silveira, 2006) diz que um agente de software é uma entidade que funciona de forma contínua e autônoma em um ambiente particular. Muitas vezes um agente de software atua no ambiente com outros agentes, sendo capaz de intervir no seu próprio ambiente, sem requerer intervenção humana.

O desenvolvimento de sistemas multiagentes produziu novos desafios tanto para a Engenharia de Software como para a Inteligência Artificial, o que implicou na necessidade de criar processos de desenvolvimento de software exclusivos para este tipo de software. Podemos definir a importância de processos de software tendo como base o livro Engenharia de software (PRESSMAN, 2006), nele é dito que processos de software são importantes pois propiciam estabilidade, controle e organização. Ou seja, os processos de software evitam que o desenvolvimento ocorra de forma caótica, fazendo com que os produtos de trabalho sejam apropriados para a equipe de projeto, tendo uma garantia maior do produto ser gerado.

Considerando a necessidade de processos de desenvolvimento de software particulares para sistemas multiagentes, pretendemos neste artigo demonstrar o estado da arte deste tipo de processos atualmente existentes com o objetivo de identificar as vantagens e desvantagens de cada um e identificar as suas possíveis lacunas.

2 METODOLOGIA

Para a coleta de dados foram utilizadas pesquisas em sites de buscas como ScienceDirect, Scopus, IEEE Xplore library e ACM Digital library. As buscas utilizadas foram: “multiagent system process”, “development process for multiagent systems”, “agile multiagent system process”, “incremental multiagent system process” e “multiagent system development process”. Outra fonte de pesquisa foi o livro “Handbook on Agent-Oriented design Process” de Massimo Cossentino et al.

3 RESULTADOS e DISCUSSÃO

Como resultado da pesquisa realizada chegamos a um total de 14 processos analisados, Gaia, Gaia2Jade, RoadMap, PROMETHEUS, Mase, O-MaSE, ROMAS, ADELFE, Message, ASPECS, PASSIM, PASSI, Agile Passi e INGENIAS. Na análise realizada foram levantados pontos relevantes destes processos, através de um estudo sobre os fundamentos de cada processo e de seu ciclo de vida. Um dos pontos observados é que na maioria dos casos o foco principal é na área de engenharia de requisitos, etapa da engenharia de software voltada a descobrir os requisitos e funcionalidades do software. Outra observação feita é que a maioria dos processos se baseia somente na linguagem de modelagem UML ou em extensões da mesma para definir suas etapas.

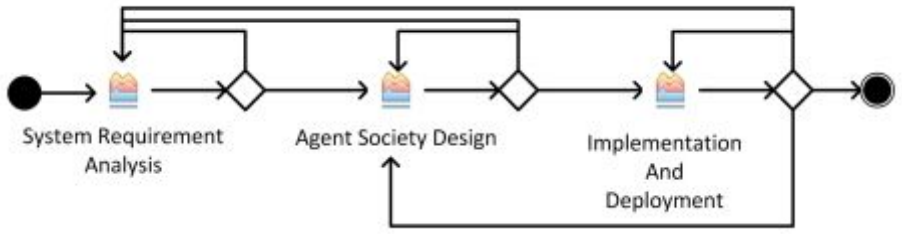
Para exemplificar o funcionamento destes processos nos próximos tópicos será apresentado o funcionamento e o ciclo de vida de alguns processos voltados a MAS.

3.1 O processo ASPECS

O processo ASPECS (Massimo Cossentino Et al., 2014) é utilizado para desenvolvimento de sistemas multiagente que combinem arquitetura organizacional holônica, segundo o artigo Arquiteturas organizacionais holônicas (Mathews, 1996), podemos definir uma arquitetura holônica como uma arquitetura cujo a conquista da ordem do sistema surge como uma questão de coordenação de entidades semi-autônomas, em vez de controle centralizado sobre as proprias operações, e abordagens organizacionais para facilitar o desenvolvimento de softwares mais complexos. O resultado é um conjunto de abstrações orientadas para a organização que foram integradas em um processo metodológico completo.

O ciclo de vida do processo ASPECS define três etapas principais: análise de requisitos, projeto da sociedade de agente e implementação, como pode ser observado na figura 1. Na etapa de análise de requisitos trabalha-se com três tipos de requisitos, requisitos de domínio, requisitos organizacionais e requisitos de sistema. Para analisar cada um dos tipos de requisitos é necessário utilizar artefatos específicos, como casos de uso ou modelo de papéis.

Figura 1- Ciclo de vida ASPECS



Fonte: Massimo Cossentino et al, 2014.

Na segunda etapa é realizado o projeto da sociedade de agente. Para isso são definidos algumas técnicas que devem ser cumpridas ao realizar essa etapa do processo. As técnicas definidas para o projeto são:

- Refinar do ponto de vista social ontologias, papéis e interações;
- Mapear a estrutura organizacional hierárquica em uma holarquia;
- Instanciar organizações em formas de grupos (identificando funções para holons e agentes);
- Mesclar os elementos para obter o conjunto completo de holônicos envolvidos na solução.

Finalmente, na etapa de implementação, o software já analisado e projetado deve ser codificado.

3.2 ADELFE.

ADELFE (Bonjean Et al, 2014) é uma sigla em francês que significa “Atelier de Developpement de Logiciels a Fonctionnalité Emergente” que traduzindo significa Workshop de Desenvolvimento de Software com Funcionalidade Emergente. A Metodologia ADELFE é dedicada a aplicações caracterizadas pela necessidade de adaptação do sistema em relação ao ambiente. O principal objetivo desta metodologia é ajudar a orientar os projetistas durante o desenvolvimento de um Sistema multiagente adaptativo (AMAS). É recomendado utilizar como linguagem de modelagem para ADELFE a UML.

O ciclo de vida da metodologia ADELFE consiste em 5 etapas que são chamadas de Definições de trabalho, como pode ser observado na figura 2. A primeira etapa representa uma descrição das especificações entre clientes, usuários e projetistas sobre o que deve ser o sistema juntamente com suas limitações e restrições, esta etapa é chamada de Requisitos preliminares. Após adquirir os requisitos preliminares a próxima etapa é chamada de requisitos finais, onde os requisitos são transformados em modelos de casos de uso, e também suas prioridades são organizadas e gerenciadas. A terceira etapa, a etapa de Análise, serve para definir uma visão compreensiva do sistema e sua estrutura, também tem como função identificar se a teoria AMAS é necessária. Na etapa de projeto deve ser feito um detalhamento da arquitetura do sistema. E na última etapa, fase de implementação, deve ser feita a programação do sistema.

Figura 2- Ciclo de vida ADELFE



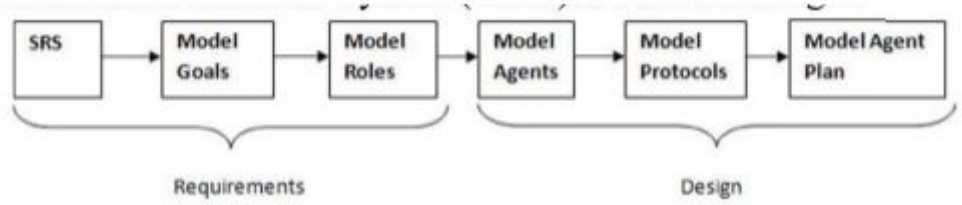
Fonte: Massimo Cossentino, 2014.

3.3 O-MaSE

O-MaSE (Scott Deloach Et al, 2014) nasceu com o objetivo de servir de melhoria para a metodologia MASE. Segundo DeLoach (2001) muitas das formas tradicionais de desenvolver software, não encaixam no paradigma multiagente. Com isso, foi observado a necessidade de criar um ciclo de vida para desenvolver sistemas multiagente. O trabalho da O-MaSE é orientar o desenvolvedor a obter um conjunto inicial de requisitos, analisar, projetar e implementar um sistema multiagente funcional.

A metodologia O-MaSE é aberta para adaptações, porém existem 12 papéis que são reconhecidos pela metodologia, eles são: engenheiro de requisitos; modelador de metas; modelador de domínio; modelador de organização; modelador de funções; modelador de classes de agentes; modelador de protocolos; modelador de políticas; modelador de planos; modelador de capacidades; modelador de ações e programador. Cada papel tem seus respectivos trabalhos e é responsável por produzir produtos. Os produtos de trabalho a serem gerados vão variar de acordo com às necessidades do projeto, porém os produtos de trabalho mais comuns podem ser observados na figura 3.

Figura 3- Principais produtos de trabalho da metodologia O-MASE



Fonte: Scott Deloach et al, 2014.

3.4 PASSI

PASSI (Massimo Cossentino Et al, 2014) significa Process for Agent Societies Specification and Implementation (Processo para Especificação e Implementação de Sociedades de Agentes), é uma metodologia para desenvolvimento SMAs. A metodologia PASSI utiliza a notação UML e diversos modelos e fases para a execução do processo.

O ciclo de vida do PASSI inclui cinco fases organizadas em um modelo de processo iterativo/incremental (Figura 4).

Figura 4- Principais produtos de trabalho da metodologia O-MASE



Fonte: Massimo Cossentino, 2014.

A primeira fase do ciclo de vida PASSI é a especificação dos requisitos do sistema, esta fase abrange todas as fases relacionadas a elicitação de requisitos, análise de requisitos e identificação de agentes e suas funções. A próxima fase é a de sociedade de agente, onde são abordados os aspectos de sociedade de agentes, como: ontologia, comunicações, descrição de funções e protocolos de interação. Para a terceira etapa são implementados os agentes. Isto significa que é dado uma visão da arquitetura do sistema em termos de classes e métodos para descrever a estrutura e o comportamento de agentes. Na etapa de código deve-se desenvolver o código fonte do sistema. E na etapa de implementação, finalizando o ciclo de vida, deve-se especificar restrições para o agente, para migrações e mobilidade.

4 CONSIDERAÇÕES FINAIS

Na engenharia de software convencional existem variados processos de desenvolvimento, desde processos pragmáticos como Cascata, até processos evolucionários. Com esta variedade de processos existentes, é papel do engenheiro de software reconhecer o que melhor se adequa ao seu caso de desenvolvimento.

Este trabalho realizou uma análise dos processos voltados a sistemas multiagentes para que fique mais claro e prático fazer a decisão de qual processo aplicar em cada caso de desenvolvimento. Com a pesquisa realizada foi encontrado um total de 14 processos, para cada processo foi especificado qual o seu foco e quais são as atividades em seu ciclo de vida.

Com isso podemos concluir que as atividades feitas neste trabalho podem contribuir com o desenvolvimento de sistemas multiagentes, auxiliando os engenheiros de software a escolherem e entenderem os processos para cada caso.

REFERÊNCIAS

BONJEAN, Noelie et al. ADELFE 2.0. In: **Handbook on agent-oriented design processes.** Springer, Berlin, Heidelberg, 2014, p. 19-63.

COSSENTINO, Massimo et al. **Handbook on agent-oriented design processes.** Berlin Heidelberg, Springer, 2014

COSSENTINO, Massimo et al. The ASPECS process. In: **Handbook on Agent-Oriented Design Processes**. Springer, Berlin, Heidelberg, 2014. p. 65-114.

DELOACH, Scott A.; GARCIA-OJEDA, Juan C. The o-mase methodology. In: **Handbook on Agent-Oriented Design Processes**. Springer, Berlin, Heidelberg, 2014. p. 253-285.

GARCIA-OJEDA, Juan C. et al. O-MaSE: a customizable approach to developing multiagent development processes. In: **International Workshop on Agent-Oriented Software Engineering.** Springer, Berlin, Heidelberg, 2007. p. 1-15.

MATHEWS, John, Holonic organisational architectures. **Human Systems Management**, v. 15, n. 1, p 27-54, 1996.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Makron books, 1995.

SILVEIRA, Ricardo Azambuja. Introdução a Sistemas Multiagente. **Universidade Federal de Santa Catarina (UFSC)**, 2006.