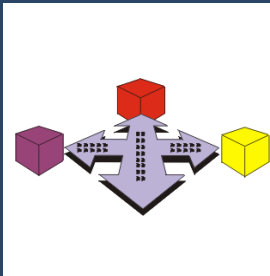




Jéssica Nascimento



Plataformas de desenvolvimento

Visão Geral

Dois modelos de funcionamento de um agente:

- Reativo – estímulo resposta
- BDI (Belief-Desire-Intention)

Visão Geral

Para auxiliar o desenvolvimento de SMAs sistemas multi-agentes, existe um conjunto de plataformas que em geral são associados a alguma linguagem de programação para fornecer um ambiente de criação e execução de operações maiores. Dentre esse conjunto de plataformas, as mais utilizadas são:

JACK

JAM

JADE

JADEX

JASON

Desenvolvida por:

- AOS (Agent Oriented Software)

Linguagem:

- JACK Agent Language baseada em JAVA

Principais características:

- Suporta a criação de classes, definições e comandos orientados a agentes;
- Fornece alto desempenho;
- Facilmente entendido;
- Fornece suporte a agentes BDI;
- Possui uma IDE (Integrated Development Environment) que permite a criação das unidades funcionais e gera o código automaticamente;
- Possui um editor de planos gráficos e uma ferramenta de designer.



JACK

As unidades funcionais:

- *Agent* – define o comportamento de um agente;
- *Capability* – representa os aspectos funcionais de um agente;
- *BeliefSet* – representa as crenças de um agente, usando um modelo relacional genérico;
- *View* – permite que consultas de propósito geral sejam feitas sobre o modelo de dados;
- *Event* – Descreve uma ocorrência, para a qual o agente deve tomar uma ação como resposta;
- *Plan* – São as instruções que o agente deve seguir para atingir seus objetivos e tratar os eventos designados a ele.

Pontos negativos:

- A ferramenta IDE da JACK é paga;
- Não dá suporte a agentes reativos simples;
- Não foi encontrado algo em relação a papel de agente ou organização.

JACK

JAM

Desenvolvida por:

- Hubber

Linguagem:

- Java

Principais características:

- Enfoque para o modelo BDI;
- Cada agente é composto por 5 componentes primários:
 - *World Model* – representa as crenças do agente;
 - *Plan Library* – biblioteca de planos que o agente pode usar para atingir seus objetivos;
 - *Interpreter* – interpretador, considerado o cérebro dos agentes;

JAM

- *Intention Structure – representação dos objetivos e das atividades correntes do agente;*
- *Observer – observador, um plano leve que é executado entre os passos do plano, tem o propósito de realizar funcionalidades que não estão determinadas no curso normal do plano.*

Alguns pontos negativos:

- Não possui uma IDE;
- Não suporta o conceito de agente reativo simples;
- Nenhuma referencia a papel ou organização.

Desenvolvida por:

- É software livre e é distribuído pela Telecom Itália, em um código fonte aberto sobre os termos da licença LGPL (Lesses General Public License Version 2).

Linguagem:

- JAVA

Principais características:

- Simplifica o desenvolvimento de sistemas multi-agentes fazendo uso da especificação FIPA (Foundations of Intelligent Physical Agents);
- Possui ferramentas gráficas e plugins para IDEs;
- Suporta debugging;
- Open source;
- Fornece suporte a sistemas distribuídos;



JADE

- Utiliza os eventos Java para realizar a comunicação entre os agentes;
- Registra automaticamente o agente com o AMS;
- Os agentes recebem um GUID (Globally Unique Identifier) da plataforma – serviço de nomenclatura;
- Gratuita;

Requisitos:

- Java SDK (Software Development Kit) 1.4 ou superior.

Alguns pontos negativos:

- Não dá suporte a agentes proativos (baseados em objetivos e utilidade) por não possuir o componente objetivo em sua estrutura;
- Não foi encontrado referência em relação a papéis de agente;

Observações:

- Plataforma já disseminada no mercado, atendendo

Observação:

- Plataforma já disseminada no mercado, atendendo a vários setores como o de telecomunicações e aplicações de internet.

JADEX

Desenvolvida por:

- Pokahr, Braubach e Lamersdorf.

Linguagem:

- JAVA e XML

Principais características:

- Agentes racionais escritos em XML (Extensible Markup Language);
- Segue o modelo BDI;
- Os principais componentes:
 - *Capability* – permitem que crenças, planos e objetivos sejam colocados em um módulo de agente;
 - *Beliefs* – representam o conhecimento do agente sobre o seu ambiente e sobre si mesmo;
 - *Goals* – são os desejos concretos e momentâneos de um agente que orientam suas ações;
 - *Plans* – representam a forma como o agente atuará em seu ambiente;

JADEX

- *Events – capacidade de reagir a diferentes tipos de eventos. JADEX suporta dois tipos de eventos: eventos internos (no próprio agente) e eventos mensagem (comunicação entre agentes).*

Alguns pontos negativos:

- Plataforma ainda em fase de maturidade, novas versões acabam não sendo compatíveis com as anteriores;
- O uso de XML aumenta a complexidade do desenvolvimento.

Observação:

- No JADEX, agentes têm crenças, que são armazenadas em uma base de crenças, objetivos representam motivações concretas, e para atingir seus objetivos, o agente executa planos, os quais são roteiros procedurais codificados em Java.

Desenvolvida por:

- Bordini, Wooldridge e Hübner.

Linguagem:

- É baseada em um interpretador para uma versão estendida da linguagem AgentSpeak(L).

Principais características:

- Permite depuração do agente;
- Possui IDE;
- Open source;
- A linguagem AgentSpeak(L) tem como especificações:
 - *Beliefs Base* – conjunto de crenças base (fórmulas atômicas);

JASON

- *Goal – é um estado do sistema em que o agente deseja chegar, podendo ser de dois tipos: Achievement Goal: o agente deseja atingir o estado de mundo onde a fórmula atômica associada é verdadeira e o Test Goal: o agente deseja testar se a fórmula atômica associada é uma de suas crenças.*
- *Plan Library – é uma biblioteca de planos do agente, na qual cada plano determina um conjunto de ações que deve ser executado para atingir o objetivo. É constituído por: Head: formado por um evento ativador (triggering event), o qual define quais eventos podem iniciar a execução de um plano. E pelo Body: que inclui ações básicas que o agente pode executar a fim de alterar o ambiente.*

Alguns pontos negativos:

- Não oferece o recurso de troca de papéis entre agentes.

CONCLUSÃO



Dos frameworks propostos, Jack não é uma plataforma livre e é mais voltada para aplicações da indústria; JADE, Jason e JADEX possuem uma IDE gratuita com suporte a depuração de código. JADE não precisa redefinir estruturas de XML como no JADEX e também direcionada para agentes BDI, como JADEX e Jason, possibilitando sua extensão para este tipo de agente.



REFERÊNCIAS

- Arquitetura BDI: Uma visão geral – Claudia F. Bratficher Dário.
- Plataforma Didática para Desenvolvimento de sistemas multiagente - Rafael da Silva Mendonça.
- JADE – A FIPA-compliant agent framework - Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa