Sistemas Multiagentes

O que seria um agente?

> Um agente é alguma coisa que pode ser vista como percebendo seu ambiente através de sensores e que age neste ambiente através de seus atuadores (RUSSEL e NORVIG, 1995).

> Um agente inteligente é um software que assiste a pessoas e age em seus lugares. Agentes inteligentes permitem que pessoas deleguem trabalhos que poderiam ser feitos por agentes de software. Agentes podem, assim como assistentes, automatizar tarefas repetitivas, lembrar coisas que você esqueceu, sumariar inteligentemente dados complexos, aprender por você e até fazer recomendações para você (GILBERT, 1997).

> Um agente autônomo é um sistema situado dentro e uma parte de um ambiente que sente esse ambiente e age sobre ele, através do tempo, realizando sua própria agenda e assim afetando o que ele sentirá no futuro (FRANKLIN e GRAESSER, 1996).

Essas são algumas definições, entretanto, os próprios autores reconhecem que essa definição é muito ampla. Há uma enorme quantidade de entidades que se encaixam nessas definições. Para definir subtipos de agente, então, os autores enumeraram propriedades para diferencia-los, sugerindo uma taxonomia de agentes inspirada na taxonomia dos seres vivos.

De forma geral, agentes de software podem ser considerados programas aos quais são delegadas tarefas, entendendo-se delegação como uma transferência de poder de decisão. Assim, é possível imaginar agentes de software decidindo qual a melhor compra a ser feita, quais os produtos interessantes para oferecer a um cliente, quais os momentos adequados para venda ou compra de ações, etc.

O que seria os Sistemas Multiagentes?

- ✓ Podem ser projetados com um ou mais Agentes;
- ✓ Podem possuir uma estrutura homogênea, onde possui Agentes com arquiteturas iguais;
- ✓ Podem possuir estrutura heterogêneas, com Agentes de diferentes arquiteturas

Arquitetura do SMA e IAD

Uma arquitetura SMA permite esquematizar propriedades e estrutura de interação entre agentes, garantindo funcionalidade do sistema.

Segundo GIRARDI (2004), “A construção de software de alta qualidade não é tarefa simples, principalmente devido à complexidade natural do software, caracterizada pelas interações necessária entre seus componentes”.

O paradigma de desenvolvimento baseado em Agentes aborda a complexidade do software de três modos:

- ➢ Decomposição: divisão de problemas de um sistema complexo em problemas menores, para fácil gerenciamento;
- ➢ Abstração: utilização de foco nos itens relevantes do sistema;
- ➢ Interações flexíveis: identificação e gerenciamento dos relacionamentos entre os componentes de resolução do problema (JENNINGS, 2000).

Na decomposição de softwares baseados em Agentes, cada agente autônomo interage de maneira flexível, para que o nível de INTERDEPÊNDENCIA entre eles seja menor, então os agentes sabem quando devem atuar e quando devem atualizar o seu estado. Assim, a complexidade do software passa a ser distribuída e descentralizada.

Na abstração, basta fazer um mapeamento com o mundo real para os sistemas computacionais, onde os agentes focam em aspectos pertinentes e descartam aspectos irrelevantes

Nas interações flexíveis, é proporcionado a flexibilidade necessário para lidar com novas condições e requisitos à medida que surgem, permitindo que o sistema com um todo se adapte sem precisar altera-lo por completo, devido a autonomia do agente.

A IAD (Inteligência Artificial Distribuída), Segundo Bond e Gasser (1988) está focada na resolução de 5 tipos de problemas.

1. Descrever, decompor e alocar tarefas para um conjunto de agentes.
2. Originar a interação e comunicação entre os agentes
3. Coordenar, controlar e assegura o comportamento coerente
4. Governar os conflitos e incertezas
5. Definir que linguagens e ambientes de programação devem ser utilizados para implementação dos agentes

Arquiteturas de SMA e COMPLEXIDADE

- Arquitetura simples: composta por um único e simples agente;
- Arquitetura moderada: quando é composta por agentes que realizam as mesmas tarefas, mas possuem diferentes usuários e podem residir em máquinas diferentes (Fig. 1);
- Arquitetura complexa: quando é composta por diferentes tipos de agentes (Fig. 2), cada um com certa autonomia, podendo cooperar e estar em diferentes plataformas (KNAPIK; JOHNSON, 1998).

Arquiteturas e a cooperação entre Agentes

Partilha de tarefas -> Agentes auxiliam outros Agentes em uma determinada tarefa.

Partilha de resultados -> Agentes disponibilizam suas informações para a comunidade.

De acordo com a cooperação, a arquitetura de um SMA pode ser classificada em três grupos:

- Arquitetura Quadro-Negro: É como uma caixa muito grande que possui informações dentro, com níveis diferentes. Um Agente deposita uma

mensagem de pergunta para essa caixa, o outro Agente pega a mensagem da caixa, processa e depois deposita uma mensagem de resposta. Basicamente, um Agente solicitando recurso de outro por meio de um intermediário.

- Arquitetura de troca de mensagens: Comunicação direta entre Agentes de forma assíncrona. Agentes precisam saber todos os nomes e endereços dos outros Agentes do sistema. Mensagens precisam de protocolos.
- Arquitetura Federativa: Quando o número de Agentes no sistema é muito grande. Mensagens de broadcast demoram demais para ser processadas, então existe um tipo de hierarquia de grupos (federações), onde existem Agentes intermediadores para encaminhar as mensagens de um grupo remetente para um grupo destinatário.

Arquiteturas de Agentes e sua coordenação

Existem 2 mecanismos de coordenação:

- ✓ Mecanismo mestre-escravo: Agentes que são gerentes, Agentes que são trabalhadores;
- ✓ Mecanismo de mercado: todos possuem um mesmo nível e conhecem as tarefas que os outros podem desempenhar. Isso diminui a quantidade de mensagens trocadas.

Comunicação entre Agentes

ACL (Agent Communication Language)

→Vocabulário: Dicionário com todos os conceitos utilizados no domínio dos Agentes. Conhecido também como ontologia.

→Linguagem Interna: Linguagem de programação baseada em lógica de primeira ordem, para codificação de dados simples, regras, restrições e expressões. KIF (Knowledge Interchange Format);

→Linguagem Externa: Encapsula estruturas KIF, para comunicação mais eficientes. KQML (Knowledge Query and Manipulation Language).

O Padrão FIPA

É um tipo de protocolo, um padrão para o desenvolvimento de agentes. Esse padrão está presente nas plataformas Multiagentes JADE e FIPA-OS. Segundo Vázquez (2002), o padrão FIPA pode ser considerado o mais completo e reconhecido internacionalmente, tornando-se uma referência quando o assunto é desenvolvimento de agentes.

FIPA-OS

Podemos entender essa plataforma como um facilitador, que busca reduzir barreiras na adoção do padrão FIPA, através do suplemente de documentos e especificações técnicas com o código fonte aberto gerenciável (OS - OPEN SORCE); Foi desenvolvida completamente em JAVA.

- Diferentes tipos de Agents Shells para a implementação facilitada de agentes que possam se comunicar através da plataforma.
- Suporte de multiplicas camadas para a comunicação de agentes;
- Configuração dinâmica da plataforma para suporte dos componentes pemutáveis;
  Interfaces abstratas;
  Ferramentas de diagnóstico e visualização

JADE

Plataforma utilizada para facilitar as tarefas do desenvolvimento de agentes de acordo com as especificações do padrão FIPA.

- ✓ Ambiente de agentes compatíveis a FIPA;
- ✓ Plataforma distribuída de Agentes;
- ✓ Ferramentas de Debug;
- ✓ Suporte a execução concorrente de agentes;
- ✓ Transporte de mensagens;
- ✓ Protocolo de bibliotecas FIPA;
- ✓ Automação de logs;
- ✓ Serviços de nome e integração;

Glossário de siglas

SMA -> Sistemas multiagentes

IAD -> Inteligência Artificial Distribuída

FIPA -> Foundation for Intelligent Physical Agents

FIPA-OS -> |||||||| || |||||||||| ||||||||| |||||||||||| Open-Source

JADE -> Java Agent Development Framework

Referências

https://www.gsigma.ufsc.br/~popov/aulas/das6607/jade.pdf

https://www.devmedia.com.br/introducao-aos-sistemas-multiagentes/28973

https://www.scielo.br/j/raeel/a/KHKj9xPC6GKS3BdtzbZtNGh/