

## 6 Comunicação entre Personagens

A comunicação permite que os agentes em um ambiente multiagente troquem informações que servirão de base para coordenar suas ações e realizar cooperação.

Linguagens para comunicação entre agentes prometem ter o mesmo papel que as linguagens naturais exercem para as pessoas (Labrou *et al.*, 1999). Neste capítulo são expostos conceitos sobre a comunicação entre agentes e sobre propostas de linguagens de comunicação que estão sendo desenvolvidas atualmente.

### 6.1. Comunicação entre agentes

Comunicação entre agentes é a troca intencional de informações causada pela produção e percepção de sinais por parte dos agentes (Siebra, 2000).

Para que haja alguma forma de interação entre sistemas baseados em agentes é preciso que haja um consenso, pelo menos em relação as interfaces, sobre os seguintes níveis de interação (Finin e Weber, 1993):

- Transporte: como os agentes enviam e recebem mensagens.
- Linguagem: qual o significado de mensagens individuais.
- Política: como os agentes estruturam conversações.
- Arquitetura: como conectar sistemas em concordância com protocolos existentes.

A comunicação entre agentes está relacionada ao nível “Linguagem”. Em (Fullbright e Stephens, 1994), é relatado que para dois agentes comunicarem alguma forma de conhecimento é necessário percorrer um triangulo significativo (*meaning triangle*) (ver **Figura 29**).

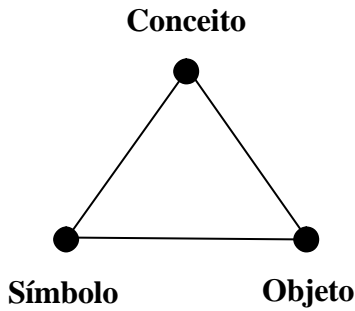


Figura 29. *Meaning Triangle* .

Na **Figura 29**, o Objeto é uma entidade que existe no ambiente, por exemplo, uma maçã. O símbolo é algo que está associado a um objeto, a palavra “maçã” refere-se ao objeto maçã. O conceito é o significado do símbolo, indicando a conexão semântica entre o símbolo e o objeto. Para um agente A comunicar um conceito para outro agente B é necessário que seja escolhido um ou mais símbolos que pertencem a este conceito. Esta escolha é feita através de uma função de inferência. Uma vez escolhido os símbolos, estes são transmitidos para o agente B que vai interpretá-los através de uma função de inferência própria dele. Utilizando esta função de inferência, o agente B vai formar seu próprio conceito a respeito dos símbolos recebidos. Um diagrama do processo de comunicação é ilustrado na **Figura 30**.

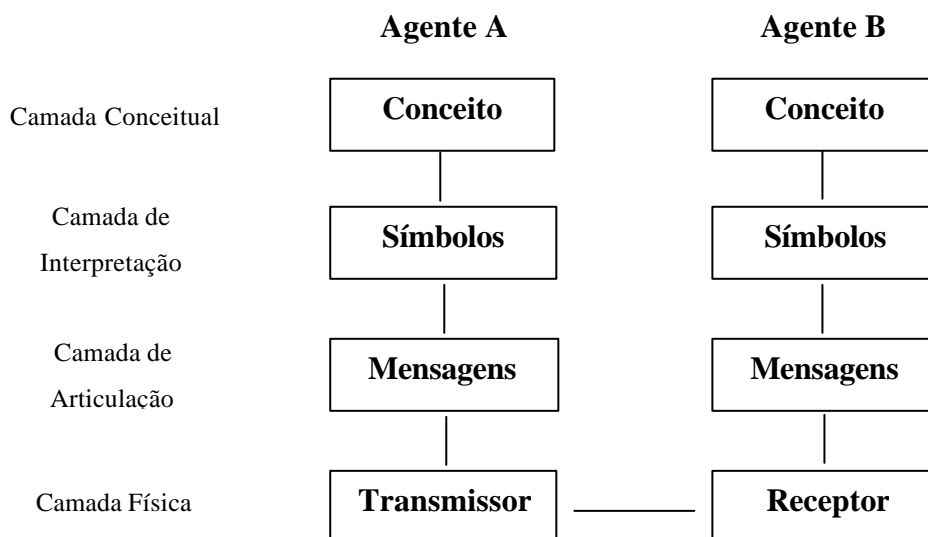


Figura 30. Processo de Comunicação (Fullbright e Stephens, 1994).

As camadas de “interpretação” e “articulação” são camadas onde podem ocorrer o surgimento de erros, como a interpretação errada de uma mensagem

recebida, se o agente não tiver mecanismos bem elaborados nessas camadas (Fullbright e Stephens, 1994).

A comunicação poderia ser muito mais eficiente se a transferência de conhecimento fosse direta, ou seja, não fosse preciso passar por todas estas camadas intermediárias. E é justamente essa abordagem que é adotada por desenvolvedores de jogos eletrônicos. O fato de haver um processo de interpretação de mensagens em cada agente é muito custoso, por isso, as mensagens são enviadas diretas sem interpretação de conteúdo, como se houvesse uma transferência de conhecimento da mente de um agente para outro. Isto pode ser justificado pelo fato de normalmente haver agentes similares, ou seja, que utilizam a mesma arquitetura. Por exemplo, em um jogo de combate, onde há grupos de soldados representados por agentes reativos, normalmente a comunicação, estabelecida entre eles, se refere a assuntos relativos a: posições que são seguras; posições que são perigosas; ou sobre a posição de algum inimigo; ou a posição de alguém que está em perigo; entre outras. O que indica a utilização de perguntas e respostas objetivas, com conteúdo simples, não havendo a necessidade de haver um processo de interpretação, pois só haveria uma sobrecarga de processamento. Está próximo o tempo em que os jogos irão imitar com detalhes todos os comportamentos humanos, por enquanto, ainda prevalece a idéia de “parecer real”, e a comunicação está inserida nesta idéia.

A troca ou passagem de mensagens entre agentes pode ser feita de três formas distintas:

- **ponto-a-ponto:** nesta abordagem, as mensagens são enviadas para um endereço específico, o receptor, que deve ser conhecido pelo emissor. Como vantagens desta abordagem, têm-se: um agente sempre sabe para onde uma mensagem está sendo emitida; e que os controles de segurança são facilmente introduzidos, já que um agente pode assegurar que a mensagem nunca será enviada para agentes desconhecidos;
- **broadcast:** a passagem de mensagem *broadcast* é baseada na emissão de uma mensagem não para um endereço específico, mas para todos os agentes da sociedade. Nesta abordagem, um agente particular pode ser substituído por outro agente com conduta equivalente e o procedimento do sistema como um todo permanecerá

inalterado. A passagem de mensagem *broadcast* não é segura, já que qualquer agente pode examinar os conteúdos de qualquer mensagem;

- **multicast**: uma maneira que está sendo utilizada para evitar os problemas do *broadcast* é a estruturação do espaço de agentes em grupos. Assim, cada agente é membro de um grupo menor e se esse agente transmite uma mensagem, esta será emitida para todos os membros de um ou mais (dependendo do sistema) dos grupos de agentes.

## 6.2. Linguagens de Comunicação para Agentes

As Linguagens de Comunicação para Agentes (LCA) têm o papel de fornecer mecanismos para troca de informação e conhecimento entre agentes. A comunicação teve um papel fundamental para desenvolvimento da sociedade humana, não é mera coincidência que o surgimento das primeiras linguagens, coincidiu com início de complexas e duradouras comunidades (Labrou *et al.*, 1999). Não seria uma utopia considerar que este mesmo progresso poderia ser experimentado por agentes computacionais, através da troca de informações utilizando linguagens de comunicação.

Em (Labrou *et al.*, 1999) é relatado que antes das LCAs já havia mecanismos para troca de informações entre aplicações. E que entre eles estão o *remote procedure call (RPC)*, *remote method invocation (RMI)* e CORBA. Mas existem distinções entre as LCAs e estes mecanismos. Distinções relacionadas com os objetos em discurso e com a complexidade semântica. São dadas duas razões indicando o porquê das LCAs estarem em um nível acima de CORBA:

- As LCAs trabalham com proposições, regras e ações em vez de simples objetos sem associação semântica alguma.
- Uma mensagem de uma LCA descreve um desejo em uma linguagem declarativa, em vez de um procedimento ou método.

Em (Finin *et al.*, 1994) são especificadas algumas características importantes na determinação de uma linguagem de comunicação para agentes:

- **forma:** deve ser declarativa, sintaticamente simples, legível, concisa e sua sintaxe deve ser extensível, pois poderá ser integrada em uma grande variedade de sistemas;
- **conteúdo:** deve fornecer um conjunto definido de ações de comunicação (primitivas);
- **semântica:** a descrição da semântica de uma linguagem normalmente é feita através da linguagem natural, a linguagem deve ser baseada numa teoria, não deve ser ambígua e deve considerar tempo e local;
- **implementação:** a implementação deve ser eficiente, tanto para velocidade como para a utilização da largura da banda. Ela deve prover uma boa adaptação com a tecnologia de *software* existente. A interface deve ser fácil de utilizar, detalhes das camadas de rede que existem abaixo das primitivas de ações de comunicação devem ser escondidas do usuário. A linguagem deve ser apropriada para implementação parcial, para que os agentes inteligentes possam manusear somente um subconjunto de primitivas de ações de comunicação;
- **rede:** deve adaptar-se às tecnologias modernas de rede, suportando todas conexões básicas – ponto-a-ponto, *multicast* e *broadcast*. As conexões síncrona e assíncrona devem ser suportadas. A linguagem deve conter um conjunto de primitivas bastante ricas que possam servir como um substrato sobre quais linguagens de alto-nível e protocolos possam ser desenvolvidos. Além disso, estes protocolos de alto-nível devem ser independentes dos mecanismos de transporte (ex. TCP/IP, *e-mail*, http, etc.) utilizados;
- **ambiente:** o ambiente em que agentes inteligentes devem trabalhar deve ser altamente distribuído, heterogêneo e extremamente dinâmico. Ele deve suportar interoperabilidade com outras linguagens e protocolos;
- **confiabilidade:** devem ser oferecidos recursos para trocas de mensagens seguras entre dois agentes. A linguagem deve suportar

mecanismos razoáveis para identificação e sinalização de erros e advertências.

Algumas especificações de LCAs foram propostas com o objetivo de viabilizar a comunicação entre agentes desenvolvidos em projetos diferentes e possibilitar a descrição de regras que determinam o fluxo de comunicação entre eles. Entre as propostas de LCAs existentes, destacam-se duas: *Knowledge Query and Manipulation Language* (KQML) e *Agent Communication Language* (ACL). Neste trabalho somente será explicada a linguagem KQML, que é a base inspiradora para a comunicação utilizada na implementação do trabalho. A linguagem FIPA ACL, estabelecida pela *Foundation for Intelligent Physical Agent* (FIPA), é mais recente que a KQML, mas possui similaridades e tem o mesmo objetivo.

Como dito anteriormente, em jogos eletrônicos, a comunicação entre agentes normalmente é feita de forma direta através de mensagens simples e objetivas para não haver consumo de processamento durante o processo de entendimento do conteúdo da mesma. Mas já existem trabalhos que fazem uso dessas linguagens, por exemplo, no jogo de estratégia “Guararapes”, desenvolvido em ambiente acadêmico na Universidade Federal de Pernambuco (Siebra, 2000), onde há a utilização da linguagem KQML na comunicação entre os agentes integrantes do jogo.

### **6.2.1. KQML**

KQML é uma linguagem de alto nível projetada para suportar interações entre agentes inteligentes em tempo de execução. Ela está sendo desenvolvida como parte do Programa de Compartilhamento de Informações (ARPA - *Knowledge Sharing Effort*) e implementada independentemente por vários grupos de pesquisa. Foi utilizada com sucesso para implementar uma variedade de sistemas de informações usando diferentes arquiteturas de software.

É uma linguagem que focaliza principalmente o formato das mensagens e os protocolos para tratamento destas. As mensagens KQML são chamadas “performativas” e estão baseadas na Teoria de Atos de Fala, onde cada mensagem é enviada com o objetivo de gerar uma determinada ação (Finin e Weber, 1993).

A Teoria dos Atos de Fala vê a linguagem humana como ações, por exemplo, ações do tipo: solicitações, sugestões, comprometer-se, e respostas. Um Ato de Fala contém três aspectos importantes (Ljungberg e Holm, 1995):

- **Locução:** a maneira de falar;
- **Ilocução:** o significado da intenção por trás da maneira de falar;
- **Perlocução:** a ação que resulta da locução.

Quando uma pessoa *P* diz “Por favor, feche a porta”, este ato consiste de um som gerado por *P* e a intenção de *P* nesta mensagem indica um comando. Mas, às vezes, a intenção de uma mensagem não é facilmente identificada. Por exemplo, quando *P* diz “Eu estou com frio”, a intenção desta mensagem pode ser vista como uma afirmação, ou um pedido por um agasalho, ou até uma exigência para aumentar a temperatura do ambiente. Desta forma, é preciso ter certeza, durante uma conversa entre agentes, qual é a real intenção das mensagens sendo trocadas.

Na Teoria dos Atos de Fala há um termo chamado “performativa” que identifica qual é a intenção por trás das mensagens, ou seja, é identificada a força ilocucionária da maneira de falar do falante. Desta forma, pode-se dizer que a mensagem tem uma premissa fundamental, que consiste em ter dois componentes:

- um conteúdo semântico;
- e a intenção do falante representando a força ilocucionária.

Em (Russel e Norvig, 2003) são informadas as principais intenções de um Ato de Fala:

- informar;
- questionar;
- responder;
- pedir ou dar um comando;
- prometer ou ofertar;
- confirmar;
- compartilhar.

A KQML além de ser utilizada como uma linguagem que facilita a interação entre agentes, também pode ser utilizada no compartilhamento de conhecimento entre dois ou mais sistemas inteligentes para encontrar soluções na resolução de problemas. Isto demonstra que aqueles que estão em conformidade com a

linguagem KQML podem responder a estas mensagens de maneira adequada e independentemente da estrutura do seu emissor.

Conceitualmente, são identificadas três camadas na linguagem KQML (Labrou *et al.*, 1999) (ver **Figura 31**):

- **conteúdo:** possui o significado da mensagem e pode ter no seu conteúdo qualquer representação de linguagem;
- **comunicação:** codifica um conjunto de características para aceitar parâmetros de baixo nível, tais como a identidade do emissor e receptor da mensagem e também um identificador único associado com a mensagem;
- **mensagem:** determina o tipo de interação que os agentes pretendem ter. É considerado o núcleo da KQML. A função primária desta camada é identificar o protocolo usado para transmitir a mensagem e a performativa que o transmissor anexa ao conteúdo.

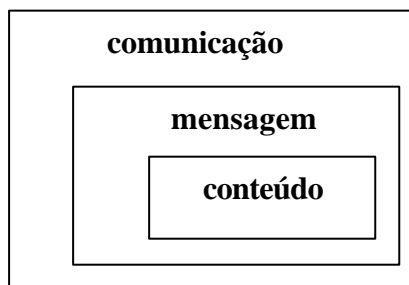


Figura 31. As camadas de uma mensagem KQML.

Uma mensagem KQML é expressa como uma *string* de caracteres ASCII usando a sintaxe definida por Finin (1993). Esta sintaxe está constituída de palavras chaves que devem começar por dois pontos “:”, e que precedem um valor de um parâmetro correspondente. Uma mensagem especificada com a KQML geralmente tem o formato da mensagem a seguir (ver **Figura 32**).



```
( perfName
  :sender    A
  :receiver  B
  :content   X
  :language  L
  :ontology  N
  :reply-with W
  :in-reply-to P
)
```

Figura 32. Típico formato de uma mensagem KQML.

Esta é uma mensagem (ver **Figura 32**) de um agente A para outro agente B, em resposta a uma mensagem previamente recebida, identificada por P. Qualquer mensagem enviada como resposta a esta mensagem deve incluir no campo “*in-reply-to*” o valor W. O conteúdo X tem a sintaxe especificada segunda a linguagem L (Por exemplo: Prolog, Lisp, ...) e os termos utilizados pertencem a ontologia N. A ontologia refere-se a um vocabulário de palavras e seus significados, que os agentes devem utilizar para se entenderem durante a conversa. O significado da mensagem é determinado pela combinação da performativa *perfName* e o conteúdo X. O conteúdo destas mensagens detalha o que vai ser perguntado, dito, inserido, entre outros.

Em (Finin e Weber, 1993) está estabelecido uma divisão de algumas performativas em categorias básicas, veja a **Tabela 6**.

<b>Categoria de Performativas</b>	<b>Nomes</b>
Consultas básicas	<i>evaluate, ask-if, ask-in, ask-one, ask-all</i>
Múltiplas respostas (consultas)	<i>stream-in, stream-all</i>
Resposta	<i>reply, sorry</i>
Informações genéricas	<i>tell, achieve, cancel, untell, unachieved</i>
Geradoras	<i>standby, ready, next, rest, discard, generator</i>
Definição de capacidade e notificação	<i>advertise, subscribe, monitor, import, export</i>
Rede	<i>register, unregister, forward, broadcast, route</i>

Tabela 6. Performativas (Finin e Weber, 1993).

Os agentes que estão de acordo com a KQML não precisam reconhecer todas as mensagens, pois um pequeno subconjunto pode ser suficiente num determinado sistema multiagente. Dependendo da necessidade, podem-se escolher somente algumas performativas para serem utilizadas na comunicação.

Dois agentes que queiram se comunicar usando KQML podem exigir os serviços de um facilitador KQML (*facilitator* ou *matchmaker*), que funciona como um ponto de encontro centralizado, ou de forma mais técnica, como um servidor de nomes. A comunicação com o facilitador é feita através de mensagens KQML padrões. Os agentes podem se registrar com o facilitador como um provedor de serviços ou de informações, usando performativas *advertise*. A existência do facilitador evita que um agente precise enviar uma mensagem a todos os outros agentes quando precisar realizar uma colaboração. Por outro lado, o facilitador representa um elemento centralizador do sistema, portanto trazendo problemas de escalabilidade e tolerância à falhas.

### 6.3. Conversa

A função dos agentes, durante uma comunicação, é decidir quando e qual ato de fala deve ser utilizado. Como as interações entre os agentes são caracterizadas pelo não determinismo, ou seja, durante a comunicação pode haver diversas alternativas a serem seguidas e os agentes deverão manter várias opções condicionais de atos de fala (Siebra, 2000).

Segundo Smith e Cohen (1995), alguns pesquisadores, como Finni (Finni, 1994) e Flores (Terry e Flores, 1988), notaram que diálogos seguem um modelo de Máquina de Estados Finitos (MEF). Este é um modelo simples e direto, sendo apropriado para especificar todos os estados possíveis que uma conversa pode ter. O modelo de conversação de Flores e Winograd (Terry e Flores, 1988; Smith e Cohen, 1995, p.9) está ilustrado através de um exemplo no diagrama da **Figura 33** demonstrando a estrutura do fluxo de uma conversa utilizando uma MEF.

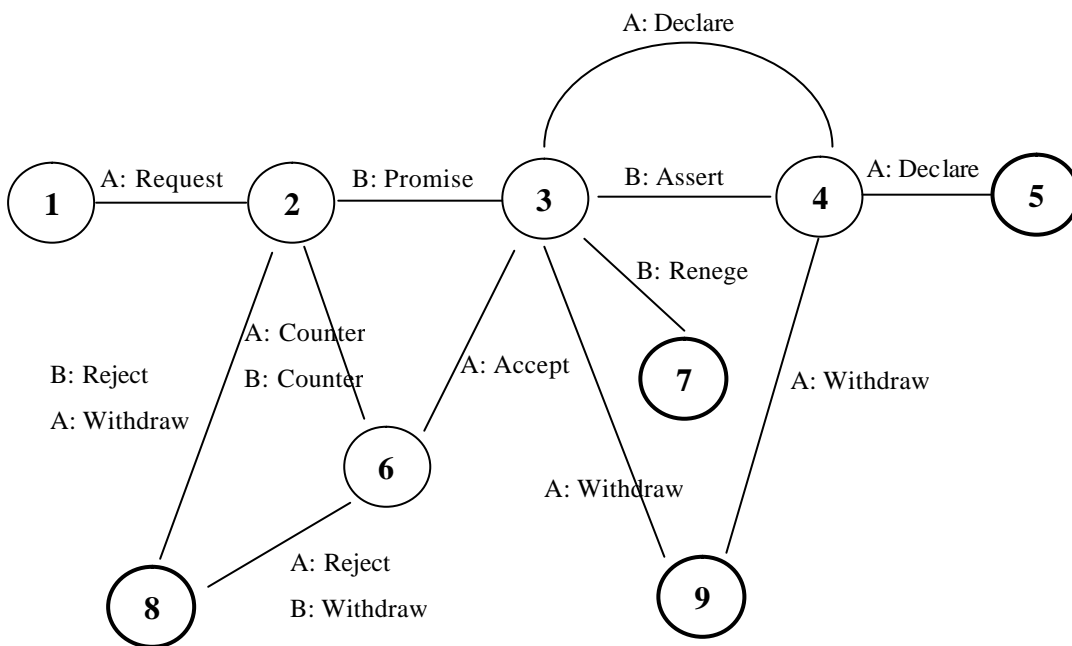


Figura 33. Estrutura de uma conversa utilizando uma MEF (Smith e Cohen, 1995, p.9).

Esta estrutura esquemática é composta por nós (círculos) representando estados e arcos (linhas) representando os atos de fala que são os responsáveis pelas transições. Os nós com círculos em negrito indicam os nós terminais, ou seja, uma vez dentro destes estados a conversa está terminada.

Além das MEF, existem outras propostas para estruturar uma conversação, entre elas destacam-se: o trabalho de Purvis *et al.*(2000) que utiliza Redes de Petri; e o trabalho de Parunak (1996) que utilizou Grafos de *Dooley*.

A utilização de MEF para modelagem de conversas entre agentes em jogos computadorizados apresenta uma solução viável e com custo baixo, pois todo o trabalho de especificação do diálogo fica na fase de projeto do jogo. Nesta alternativa podem ser especificadas as várias intenções que uma mensagem possa vir a ter, por exemplo no caso dito anteriormente sobre a mensagem “Eu estou com frio” que poderia ter várias intenções. As MEF não são novidade no ambiente de desenvolvimento de jogos e têm sido utilizadas para diversos propósitos, entre o mais comum está o de retratar o comportamento de entidades.