
Comunicação

Linguagens de Comunicação de
Agentes

Introdução

- Comunicação tem sido reconhecida como um tópico de importância central na ciência da computação.
 - Muitos formalismos têm sido desenvolvidos para representar as propriedades de comunicação em sistemas concorrentes (Hoare, 1978; Milner 1980).
-

Introdução

- O problema característico em comunicação, na pesquisa de sistemas concorrentes, é aquele de sincronização de múltiplos processos.
 - Dois processos necessitam se sincronizar, quando existe a possibilidade que eles possam interferir na execução deles, em um modo destrutivo.
-

Introdução

- Exemplo clássico: “atualização perdida”
(lost update)
 - Neste cenário, temos dois processos:
P1 e P2.
 - Ambos têm acesso a alguma variável “v”
compartilhada.
-

Introdução

- O processo P1 inicia atualizando valor de “v”, primeiro lendo o valor e então modificando-o (por exemplo, incrementando-o).
 - P1 salva o novo valor atualizado em “v”.
 - Mas, entre P1 ler e outra vez salvar o valor de “v”, P2 atualiza “v”, salvando algum valor nele.
-

Introdução

- Quando P1 salva seu valor modificado de “v”, a atualização realizada por P2 é assim perdida, o que certamente não foi o que se pretendeu.
 - Esse problema de “atualização perdida” é uma questão real no projeto de programas que se comunicam através de estruturas de dados compartilhadas.
-

Introdução

- Então, a comunicação entre os processos P1 e P2 deve ser tratada em “baixo nível”.

 - E como **comunicação** é tratada na comunidade de **agentes** ?
-

Introdução

- Para entender a resposta, vejamos como comunicação é tratada na comunidade de programação orientada a objeto, isto é, comunicação como invocação de método.
 - Em Java, sejam dois objetos O1 e O2. Seja O1 ter um método público m1. O objeto O2 pode se comunicar com O1 por invocar o método m1.
-

Introdução

- Em Java, O2 executa uma instrução parecida a algo como: `o1.m1(arg)` , onde `arg` é o argumento que O2 deseja comunicar a O1.
 - Porém, considere: qual objeto toma a decisão sobre a execução do método `m1` ?
-

Introdução

- Neste cenário, O1 não tem nenhum controle sobre a execução de m1. A decisão sobre executar m1 é inteiramente de O2.
 - Considere um cenário similar, mas agora, **orientado a agente**.
 - Sejam dois agentes A_i e A_j , onde A_i tem a capacidade de realizar uma ação a , a qual corresponde “**fracamente**” a um método.
-

Introdução

- Mas, não existe nenhum conceito no mundo orientado a agente, de um agente A_j “invocar um método” sobre o agente A_i .
 - Isto é porque, A_i é um **agente autônomo**: ele **tem controle sobre seu estado e seu comportamento**.
-

Introdução

- A_i não pode ser tomado para dar algo por certo, ou seja, que o agente A_i executará uma ação a , exatamente porque um outro agente A_j deseja A_i para alguma ação a .
 - Realizar a ação a pode não ser o melhor para o agente A_i .
-

Introdução

- O controle com respeito a **decisão sobre executar uma ação** é, assim, **muito diferente em agentes**, do que em objetos.
-

Introdução

- Em geral, **agentes podem nem mesmo forçar outros agentes** a realizar alguma ação.
 - Ou nem mesmo, por exemplo, a escrever dados sobre o estado interno de outros agentes.
-

Introdução

- **Isto não significa, que eles não possam se comunicar.**
 - **O que agentes devem fazer é realizar ações comunicativas, numa tentativa de influenciar outros agentes apropriadamente.**
-

Introdução

- Por exemplo: Suponha que eu diga a você: “**está chovendo em Londres**”, em um modo bem sincero.
 - Sob circunstâncias normais, uma tal **ação de comunicação**, é uma tentativa minha, para modificar sua crença.
-

Introdução

- É claro, simplesmente proferindo a sentença “**está chovendo em Londres**”, não é normalmente o bastante, para ocasionar essa situação de crença (pelas razões explicadas antes).
-

Introdução

- Eu não tenho controle sobre suas próprias crenças (desejos, intenções).
 - Você pode acreditar que eu sou, notoriamente, não confiável sobre o assunto do tempo atmosférico. Ou mesmo que eu sou, patologicamente, mentiroso.
-

Introdução

- Mas, em realizar a ação de comunicação de proferir “**está chovendo em Londres**”, eu estou **tentando mudar seu estado interno de crença.**
-

Introdução

- Além disso, **eu estou realizando esta para algum propósito** – presumivelmente, **porque eu pretendo que você acredite que está chovendo ...**
-

Ações de Fala

- **A Teoria das Ações de Fala (Austin, 1962) trata comunicação como ação.**

- **Característica de ação:**

“mudam o estado do mundo em um modo análogo a ações físicas”.

Ações de Fala

- É assumido, sob hipótese, que **ações de fala são realizadas por agentes**, exatamente como outras ações, no contexto de suas intenções.
-

Ações de Fala

- John Searle (1969) estendeu o trabalho de John Austin (1962).
 - No fim dos anos 60 e início dos anos 70, um número de pesquisadores em IA começaram a construir sistemas que poderiam “planejar” como, autonomamente, alcançar metas (Allen et al., 1990).
-

Ações de Fala

- Claramente, se um tal sistema de IA, requer interagir com humanos, ou outros agentes autônomos, então tais “planos” deveriam incluir **ações de fala**.
-

Ações de Fala em IA

- Isto introduziu a questão de como as propriedades de ações de fala poderiam ser representadas, de modo que sistemas de planejamento (planning systems) poderiam raciocinar sobre essas.
-

Linguagens de Comunicação de Agentes

- Teorias de Ações de Fala têm diretamente influenciado um número de linguagens que foram desenvolvidas especificamente para comunicação entre agentes.
-

Linguagens de Comunicação de Agentes

- Início dos anos 90, o projeto KSE (Knowledge Sharing Effort) (DARPA, 1990),
 - Com o intuito de desenvolver protocolos para troca de **conhecimento representado** entre sistemas de informação autônomos, gerou duas importantes linguagens (Finin et al., 1993):
-

Linguagens de Representação de Conhecimento

- KQML (Knowledge Query and Manipulation Language)
 - KIF (Knowledge Interchange Format)
-

Linguagens de Representação de Conhecimento

- KQML define um formato de envelope para mensagens, usando o que um agente pode explicitamente afirmar.
 - KQML não está interessada no conteúdo da mensagem.
 - Pode portar outras linguagens em seus formatos de envelopes.
-

Linguagens de Representação de Conhecimento

- Por exemplo, XML dentro do envelope KQML.
 - KIF é uma linguagem explicitamente voltada para permitir representação de conhecimento sobre algum domínio de discurso em particular.
-

Linguagens de Representação de Conhecimento

- KIF foi testada primeiramente para formar o conteúdo de KQML.
 - KIF (Genesereth and Fikes, 1992)
 - KIF não foi criada para ser uma linguagem na qual mensagens pudessem ser expressas.
-

Linguagens de Representação de Conhecimento

- KIF é baseada na Lógica de Primeira Ordem (Lógica dos Predicados)
 - KIF pode expressar:
 - propriedades de “coisas” em um domínio; (Michel é vegetariano.)
 - relações entre “coisas” em um domínio; (Michel e Janine são casados.)
-

Linguagens de Representação de Conhecimento

- Para expressar coisas, KIF possui um parato lógico fixado: `and`, `or`, `not`, `forall`, `exists`.
 - KIF possui um vocabulário básico de “objetos”: números, caracteres, strings.
 - KIF possui algumas funções e relações padrões: “+”, “less than”
-

Linguagens de Representação de Conhecimento

- KIF pode manipular uma lista de “objetos”.
 - Usando este aparato básico pode definir novos “objetos” e funções e relações entre esses.
-

Linguagens de Representação de Conhecimento

- Expressões KIF:

```
(=(temperature m1)(scalar 83  
celsius))
```

= é a **relação de igualdade**: uma relação entre dois objetos no domínio; = é provida como padrão em KIF.

Linguagens de Representação de Conhecimento

temperature é uma função de um único argumento **m1**;

scalar é um função que toma dois argumentos.

Estas funções devem ser definidas em KIF.

Linguagens de Representação de Conhecimento

- Como definições podem ser usadas para introduzir novos conceitos num domínio, em termos de conceitos existentes:

```
(defrelation bachelor (?x) :=  
  (and (man ?x)  
        (not (married ?x) ) ) )
```



Linguagens de Representação de Conhecimento

- **?x** é uma variável;
 - Existem duas **relações**: `man`, `married`, as quais tomam um único argumento;
 - **:=** significa “**é, por definição,**”
-

Linguagens de Representação de Conhecimento

- Como relacionamentos entre indivíduos no domínio podem ser representados:

```
(defrelation person (?x) :=>  
  (mammal ?x))
```

person **e** mammal **são relações.**

Linguagens de Comunicação de Agentes

- KQML é uma linguagem baseada em mensagem para comunicação de agentes.
 - KQML define um formato comum para mensagens.
-

Linguagens de Comunicação de Agentes

- Cada mensagem KQML tem uma **performativa** (a qual pode ser pensada como uma classe de mensagens) e um número de parâmetros (pares de atributos/valores), os quais podem ser pensados como instâncias de variáveis.
-

Linguagens de Comunicação de Agentes

- Exemplo de uma mensagem KQML:

```
(ask-one
  :content (PRICE_IBM ?price)
  :receiver stock-server
  :language LPROLOG
  :ontology NYSE-TICKS
)
```

- O sender está perguntando o preço de ações IBM.
-

Linguagens de Comunicação de Agentes

- **ask-one** é uma performativa, a qual um agente usará para fazer uma pergunta a um outro agente.
 - Os outros componentes desta mensagem representam seus atributos.
 - **:content** especifica o conteúdo da mensagem.
-

Linguagens de Comunicação de Agentes

- **:receiver**
o recipiente pretendido para a mensagem.
 - **:language** LPROLOG
a linguagem usada no conteúdo da mensagem.
 - **:ontology** NYSE-TICKS
representa a terminologia usada na mensagem.
-

Linguagens de Comunicação de Agentes

- Outros parâmetros para mensagens KQML:

:force se o sender da mensagem recusa o conteúdo da mesma.

:reply-with se o sender espera uma resposta, e se assim, um identificador para o reply.

Linguagens de Comunicação de Agentes

- Outros parâmetros para mensagens KQML:

:in-reply-to refere-se a :reply-with

:sender é o sender da mensagem

Linguagens de Comunicação de Agentes

- Diversas versões de KQML foram propostas.
 - Ver tabela de performativas KQML, dada em Finin et al. (1993), com 41 performativas.
 - Na tabela:
 - S denota o **:sender**,
 - R denota o **:receiver**,
 - C denota o **conteúdo da mensagem**.
-

Linguagens de Comunicação de Agentes

- Agentes usando KQML para se comunicar, podem ser implementados em diferentes linguagens de programação e paradigmas.
 - Em particular, qualquer informação que agentes têm, pode ser representada internamente em diferentes modos.
-

Linguagens de Comunicação de Agentes

- Nenhum agente pode assumir que um outro agente usará a mesma representação interna.
 - Para propósitos de comunicação, faz sentido para agentes tratarem outros agentes como eles tendo alguma representação interna de conhecimento.
-

Linguagens de Comunicação de Agentes

- Assim, pode-se dizer que:

“agentes ***atribuem conhecimento*** a outros agents”.

- Este conhecimento atribuído é conhecido como a ***base de conhecimento virtual***.
-

Linguagens de Comunicação de Agentes

- FIPA (Foundation for Intelligent Physical Agents) desenvolve padrões para sistemas de agentes.
 - ACL (FIPA, 1999),
Agent Communicating Language
20 performativas
 - Plataforma: JADE
(Java Agent Development Environment)
-

Ontologias para Comunicação de Agentes

- Definição
 - Exemplos
 - Arquitetura do Servidor Ontolíngua
-

Linguagens de Coordenação

- Comunicação de agentes observando uma estrutura de dados compartilhada e fontes de conhecimento (agentes) que interagem através dessa estrutura (proporcionando soluções parciais) para resolverem um problema.
 - Linda (Gelernter, 1985; Carriero and Gelernter, 1989) e o **Espaço de Tuplas** e operações (out, in, rd) definidas sobre esse espaço.
-

Linguagens de Comunicação de Agentes
