



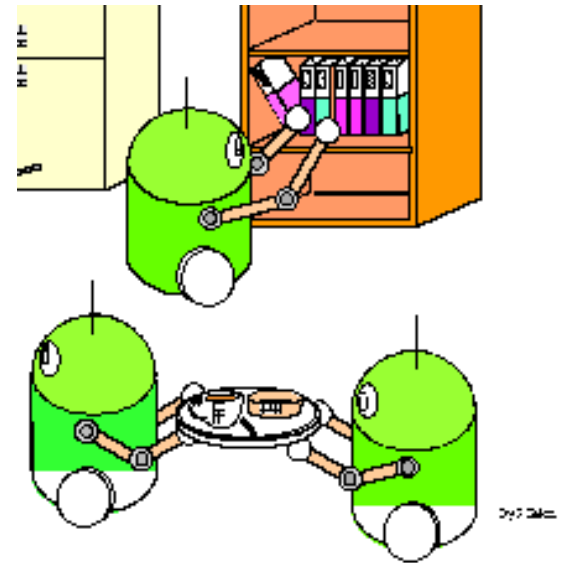
Universidade Federal de Santa Catarina
Disciplina de Inteligência Artificial
Campus Araranguá

Sistemas Multiagentes

Prof. Eliane Pozzebon
epozzebon@gmail.com

Programa

- Introdução
- Agentes Inteligentes
- Sistemas Multiagentes
- Aplicações



Introdução

- Inteligência Artificial Distribuída
 - Ramo da IA que está relacionada a **solução cooperativa** de problemas dentro de um certo ambiente por intermédio de agentes distribuídos.
 - A IAD se sub-divide em dois grandes grupos:
 - Solução Distribuída de Problemas (SDP)
 - Foco: o problema
 - Sistemas Multiagentes (SMA)
 - Foco: o Agente

Introdução

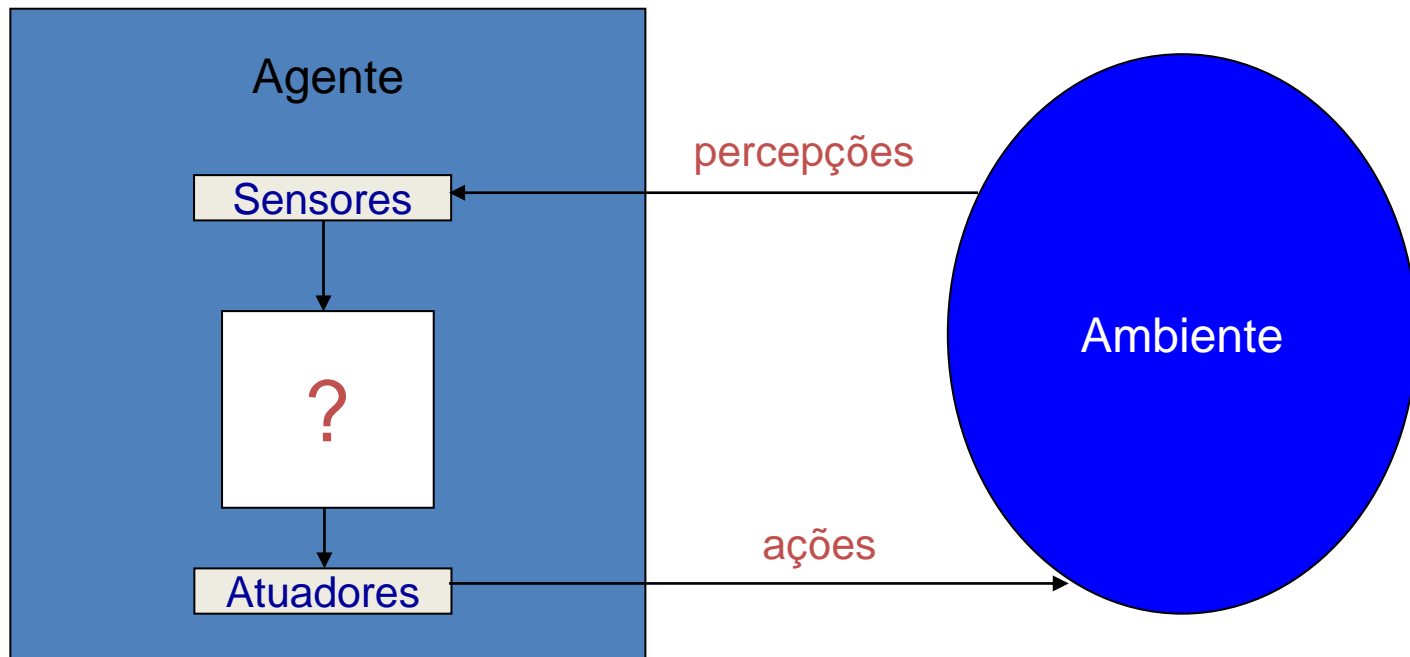
- Definição de agente:
 - *Um agente é um sistema computacional, posicionado em algum ambiente, que é capaz de agir com autonomia flexível visando atingir os objetivos para o qual foi projetado.*
 - Exemplo:
 - Agente humano, agente animal, agente robótico, agente em software, ...

Agentes Inteligentes

- Um agente pode ser definido como uma **entidade** capaz de **perceber** o ambiente por meio de **sensores** e de **agir** sobre esse ambiente por intermédio de **atuadores**.
- O **comportamento** do agente é descrito pela **função agente** que **mapeia** dados da percepção para ações.

Agentes Inteligentes

- Visão geral de um agente:



Agentes Inteligentes

- Um agente é um nó processador que deve possuir:
 - Um certo grau de autonomia para raciocinar e tomar decisões por sua própria vontade.
 - Capacidade para interagir com outros agentes, sistemas ou humanos.
 - Um certo grau de independência para resolver um problema, isto é, o agente tem conhecimento sobre como resolver pelo menos parte do problema

Agentes Inteligentes

- Para atingir seus objetivos, um agente deve possuir as seguintes capacidades:
 - **Receptividade**
 - Deve poder perceber o seu ambiente e responder a tempo e adequadamente as mudanças que ocorram nele.
 - **Pró-atividade**
 - Não devem apenas agir em resposta ao seu ambiente, mas devem agir oportunisticamente por iniciativa própria de acordo com seus objetivos.
 - **Sociabilidade**
 - Devem poder agir, quando apropriado, com outras entidades do ambiente de forma a finalizar seus problemas e ajudá-las nas suas atividades.

Agentes Inteligentes

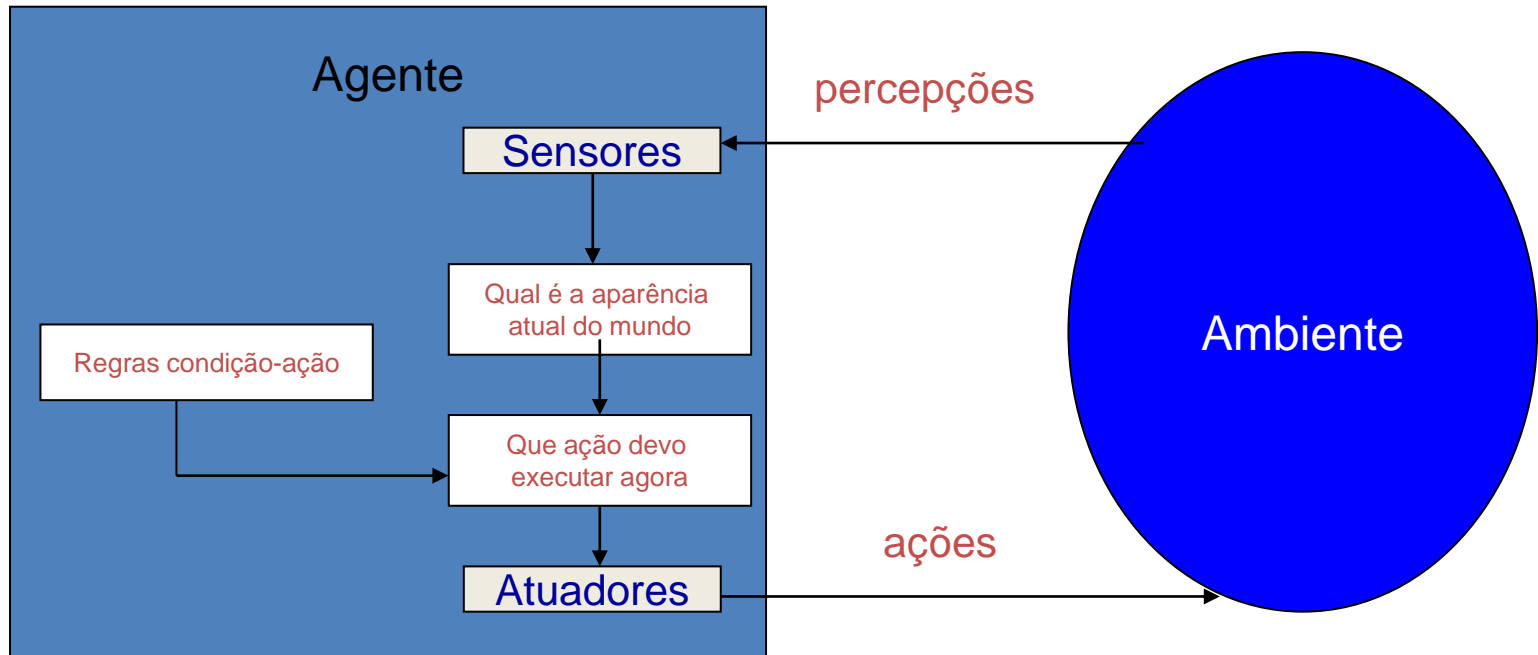
- Tipos de agentes:
 - Software ou hardware
 - Estacionários ou móveis
 - Persistentes ou temporários
 - Reativos ou cognitivos

Agentes Inteligentes

- Agentes Reativos
 - Escolhem suas ações baseados unicamente nas percepções que têm do ambiente.
 - Características:
 - O objetivo é “sobreviver” e não é explicitamente representado.
 - Ação baseada em estímulo-resposta.
 - Não raciocinam sobre o mundo, não têm memória e raramente interagem com outros agentes, e por isso agem reativamente.
 - Plano de ações pré-concebido quando do seu projeto.
 - Não se planejam para futuras ações.
 - Possuem baixa complexidade e normalmente estão em grande quantidade em um sistema.
 - Comunidade pode ter um comportamento global inteligente. (Ex.: colônia de formigas)

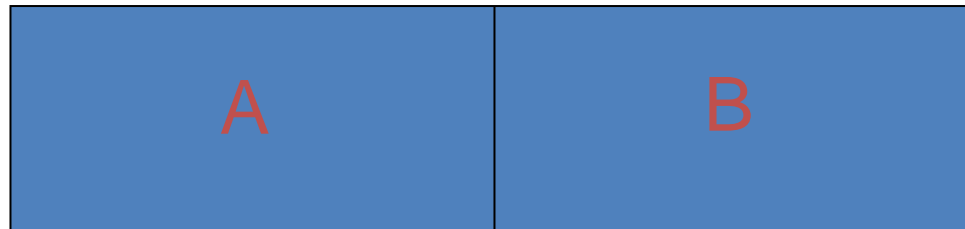
Agentes Inteligentes

- Agentes Reativos



Agentes Inteligentes

- Agentes Reativos
 - Exemplo: **Aspirador de pó**
 - Condição-ação:
 - Se *Estado=Sujo* então retorna Aspirar
 - Senão, se *Local=A* então retorna Direita
 - Senão, se *Local=B* então retorna Esquerda



Agentes Inteligentes

- Agente Cognitivo
 - Os agentes possuem um “estado mental” e funcionam racionalmente, isto é, raciocinam para construir um plano de ações que leva a um objetivo pretendido.
 - Características:
 - O objetivo é “cooperar” e é explicitamente representado.
 - Ação baseada em intenções.
 - Raciocinam sobre o mundo, têm memória e interagem com outros agentes e ambiente, e por isso agem inteligentemente.
 - Plano de ações construído / adaptado dinamicamente.
 - Se planejam para futuras ações.
 - Possuem média-alta complexidade e normalmente estão em pequena quantidade em um sistema.
 - Agentes pode ter um comportamento individual inteligente.
 - Requerem sofisticados mecanismos de coordenação e protocolos de alto nível de suporte à interação.

Agentes Inteligentes

- Agente Cognitivo
 - É um agente que age de forma “correta” no ambiente.
 - Fatores que definem a racionalidade de um agente:
 - Medida de desempenho que define o critério de sucesso.
 - Conhecimento anterior que o agente tem do ambiente.
 - As ações que o agente pode executar.
 - A seqüência de percepções do agente até o momento.
 - Um agente racional deve ser autônomo.

Agentes Inteligentes

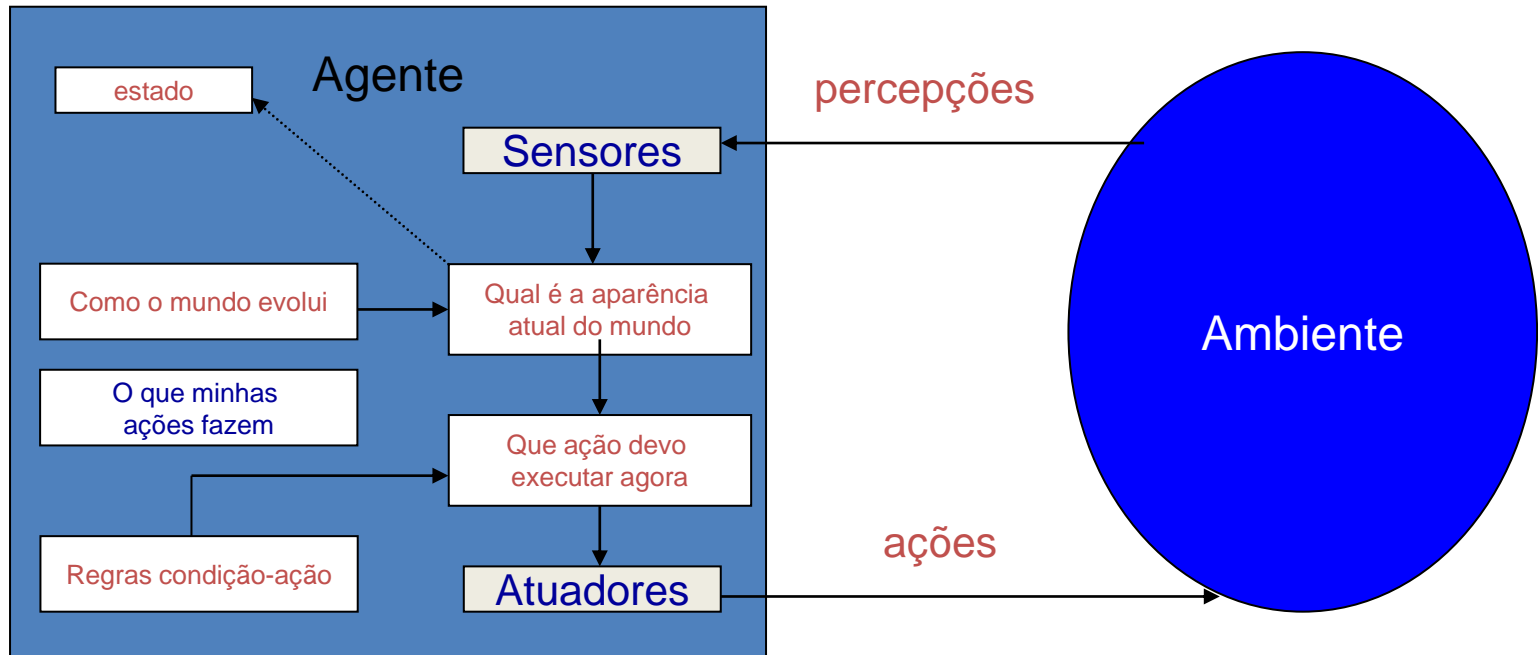
- Um ambiente no qual um agente atua com o objetivo de executar uma dada tarefa pode ser classificado em:
 - Completamente observável ou parcialmente observável.
 - Determinístico ou estocástico.
 - Episódico ou seqüencial.
 - Estático ou dinâmico.
 - Discreto ou contínuo.
 - Agente único ou multiagente.

Agentes Inteligentes

- Um agente pode ser classificado em quatro tipos dependendo do modo como seleciona a ação a ser executada.
 - Agente reativo
 - Agente reativo baseado em modelos
 - Agente baseado em objetivos
 - Agente baseado na utilidade

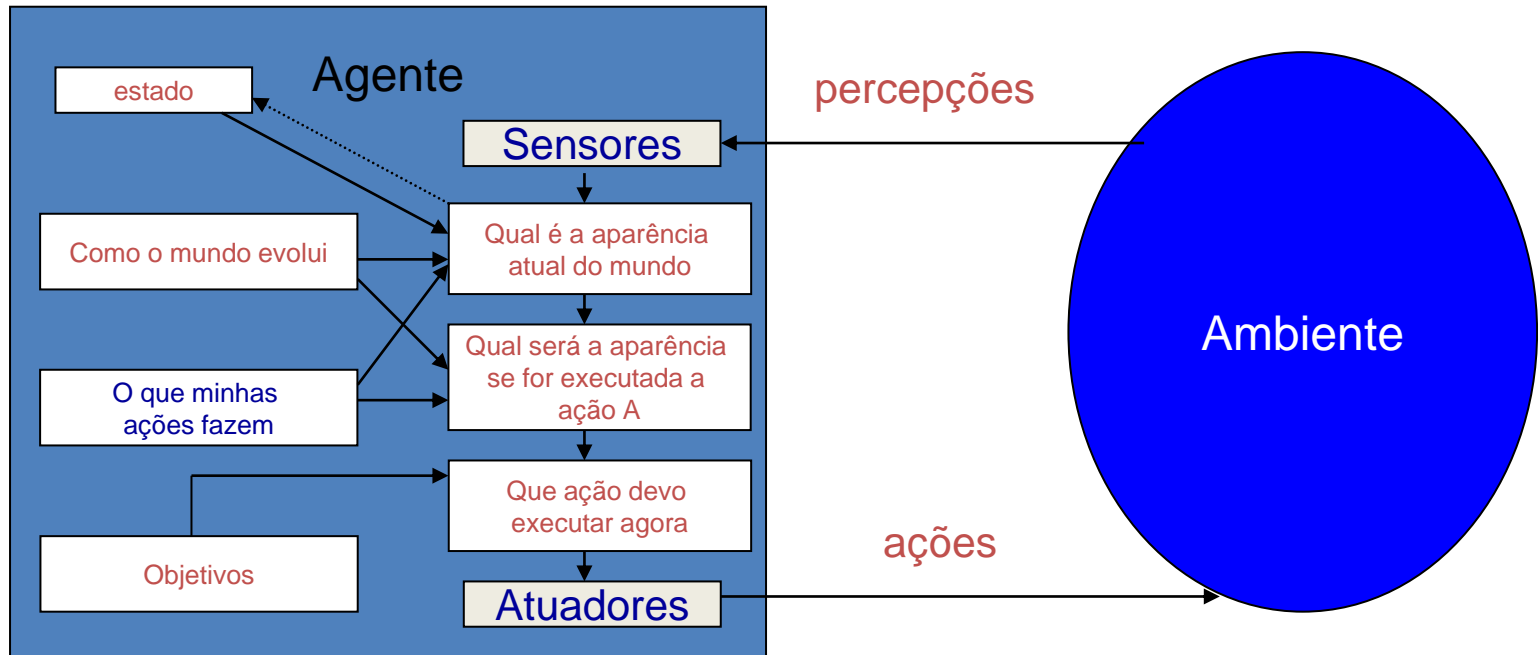
Agentes Inteligentes

- Agente Reativo Baseado em Modelos



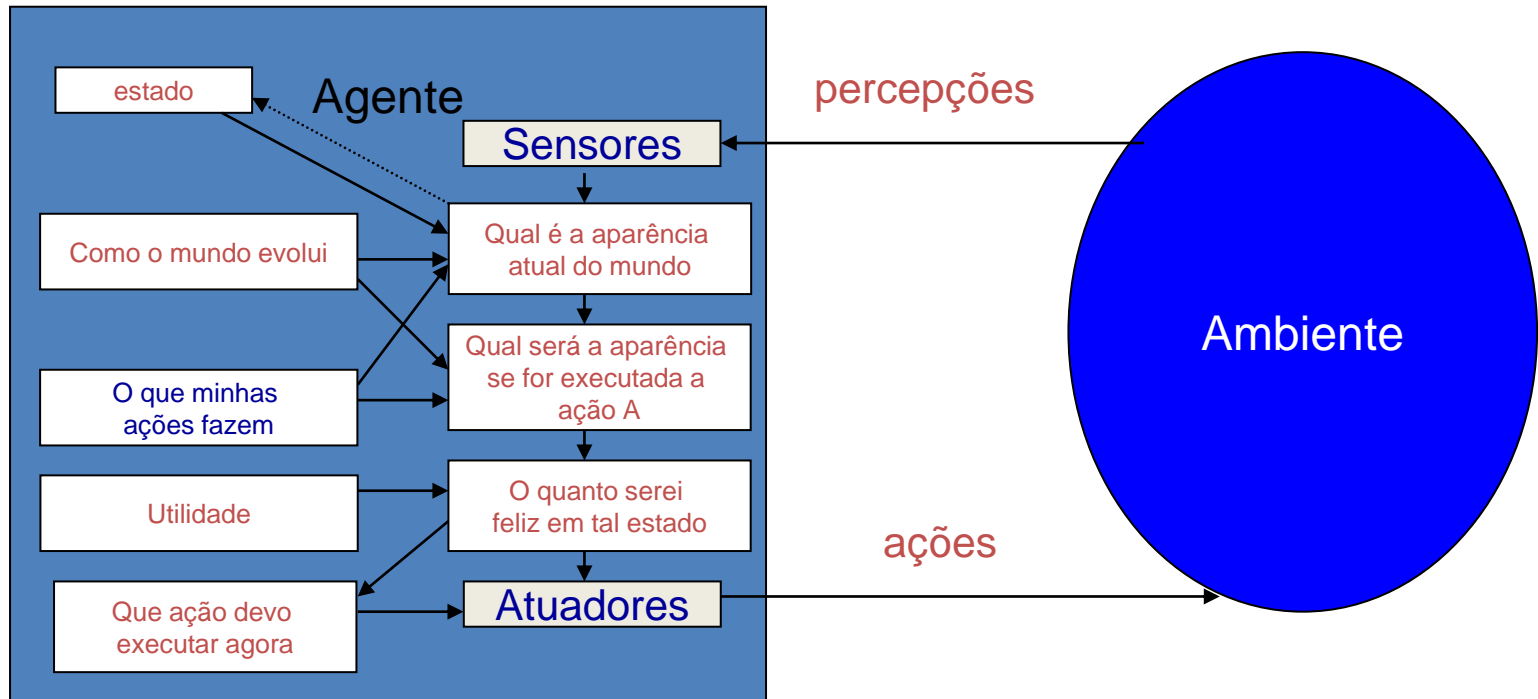
Agentes Inteligentes

- Agente Baseado em Objetivos



Agentes Inteligentes

- Agente Baseado na Utilidade

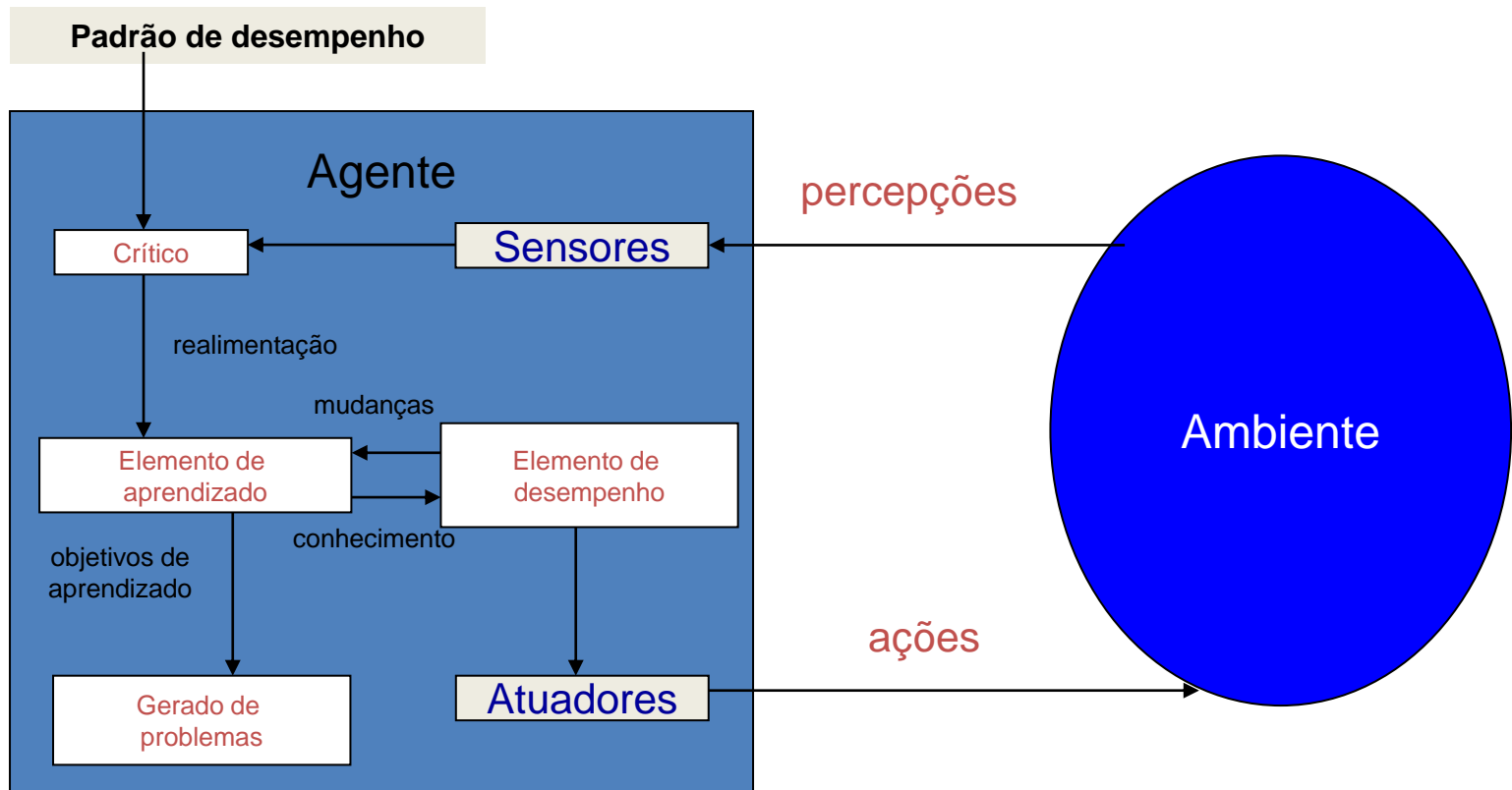


Agentes Inteligentes

- Agente com aprendizagem
 - O aprendizado permite que ao agente operar em ambientes inicialmente desconhecidos e se tornar mais competente a medida que interage com o ambiente.
 - Um agente de aprendizado pode ser dividido em quatro componentes conceituais:
 - Elemento de aprendizado
 - Elemento de desempenho
 - Crítico
 - Gerador de problemas

Agentes Inteligentes

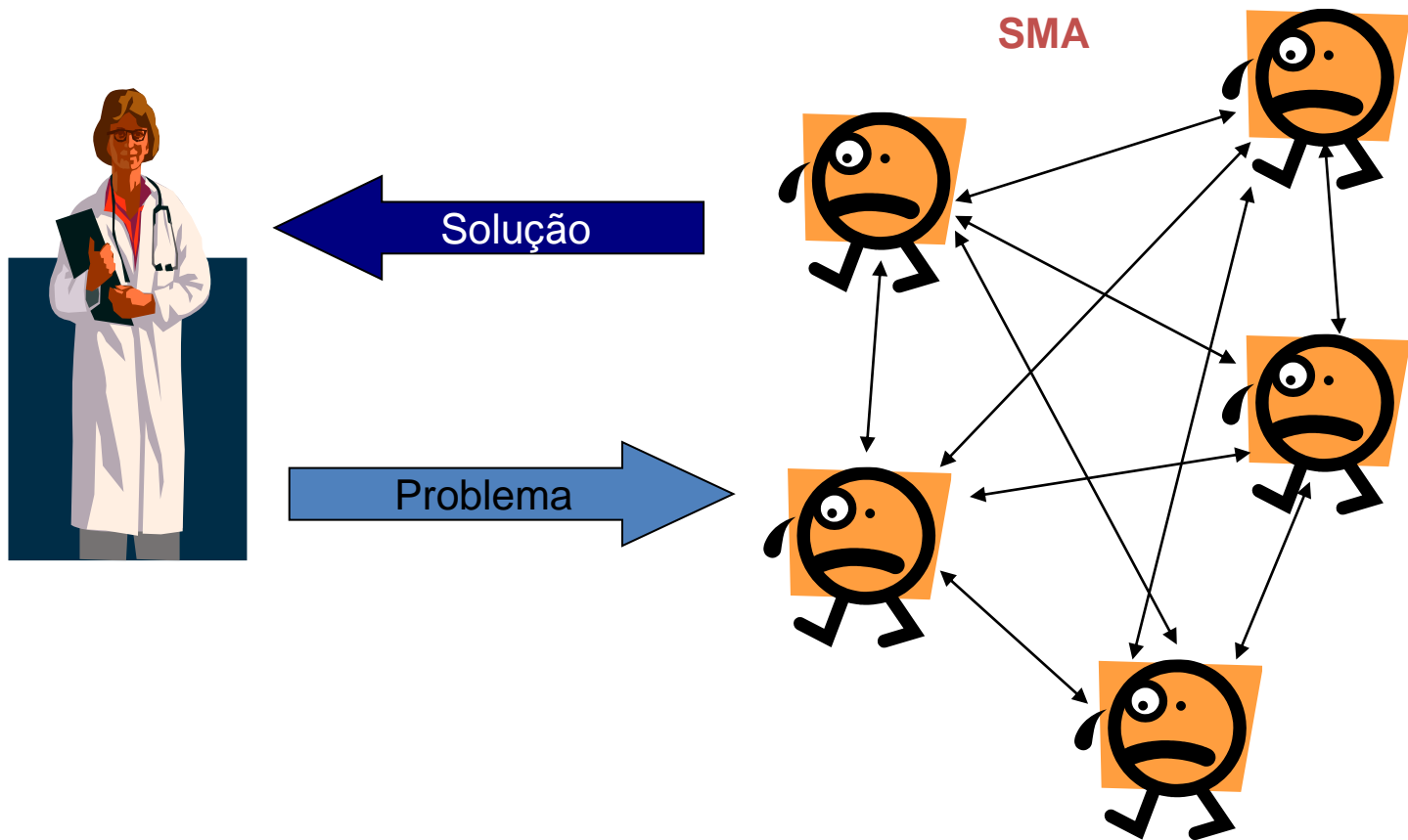
- Agente com aprendizagem



Sistemas Multiagente

- Definição:
 - Um **Sistema Multiagente** consiste de uma aplicação distribuída **composta** por um conjunto de **agentes**, que cooperam entre si para a solução de um problema complexo que está além das suas capacidades individuais.
 - Um agente é uma entidade lógica ou física à qual é atribuída uma certa missão que ela é capaz de cumprir de maneira autônoma e em coordenação com outros agentes.

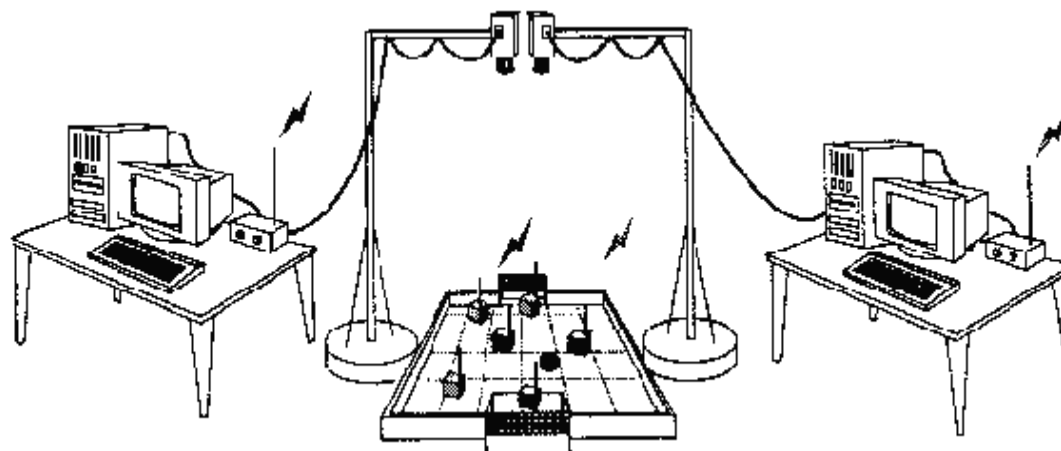
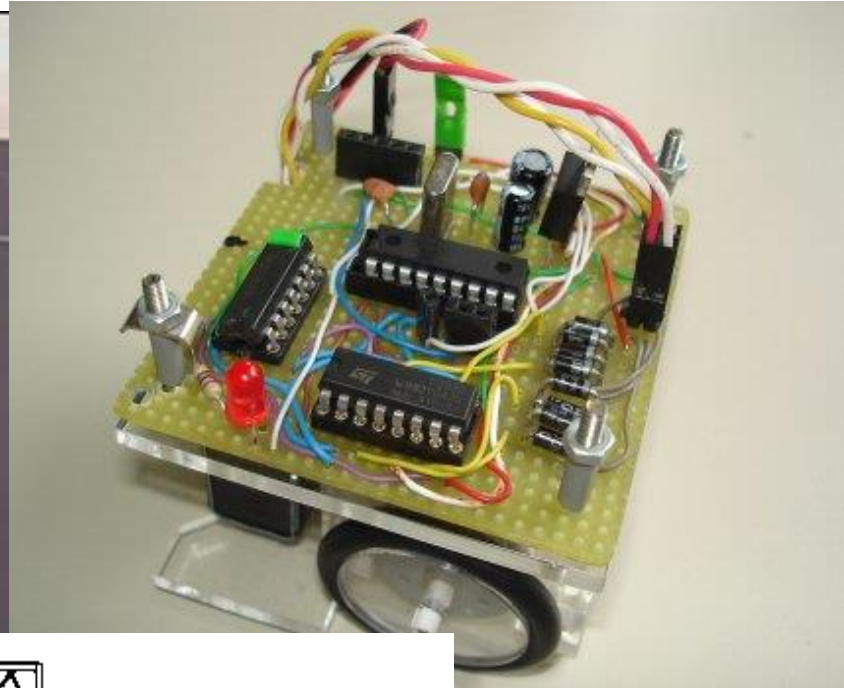
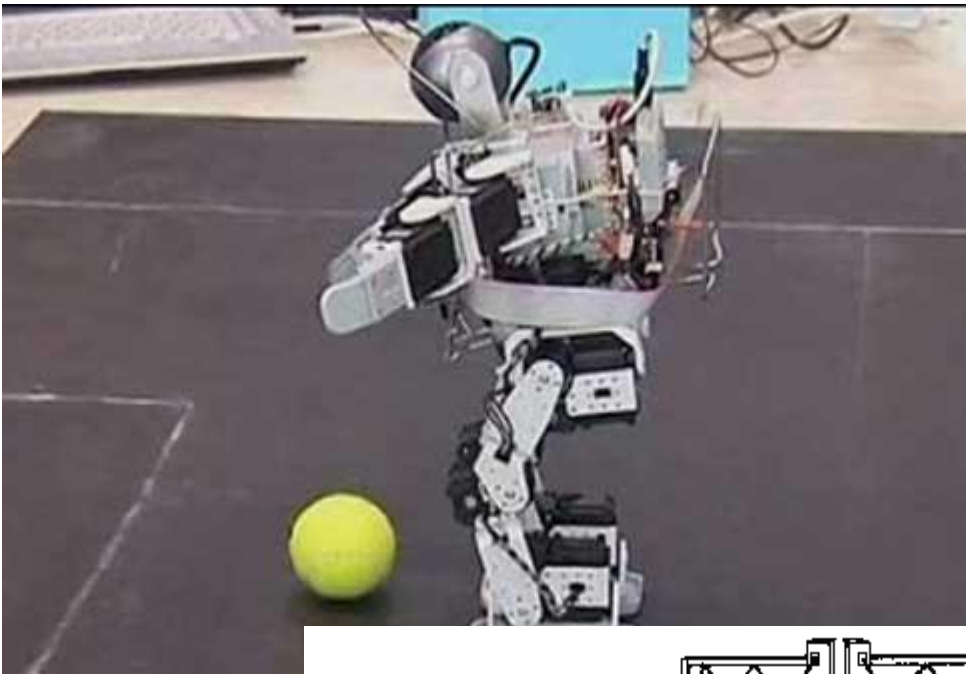
Sistemas Multiagente



Sistemas Multiagente

- Quando utilizar multiagente:
 - Quando o domínio de aplicação / problema apresenta as seguintes características:
 - É intrinsicamente distribuído.
 - Requer uma junção de diferentes domínios de conhecimento para a solução do problema.
 - Requer a aplicação de diferentes resolvedores de problemas, integrados em um mesmo ambiente.
 - Inclui diferentes níveis de autonomia e descentralização de resultados e decisões.
 - É dinâmico.
 - É extremamente conflitante, em função das muitas, normais e dinâmicas restrições usualmente existentes, o que requer variados níveis de cooperação e negociação a fim de que o processamento não seja interrompido.

Exemplo de sistemas

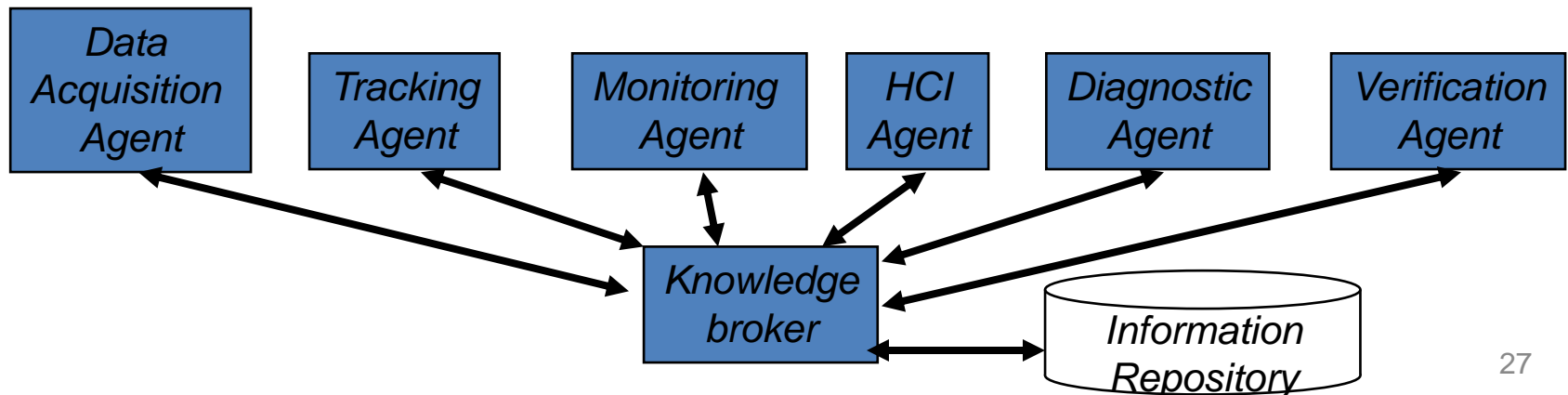


Exemplo de sistemas

- **Time de Futebol - Sistema Multiagentes Cognitivo Cooperativo**
 - Cada jogador possui um conhecimento individual e limitado;
 - Cada jogador não pode resolver o problema sozinho;
 - Cada jogador pode ter características diferentes dos demais;
 - Cada jogador age de forma autônoma e assíncrona;
 - Existe um objetivo global que é de conhecimento de todos os indivíduos;
 - Este objetivo global está **acima** dos objetivos individuais de cada agente;
 - Não existe um controle global;
 - A junção das capacidades individuais resolve o problema.

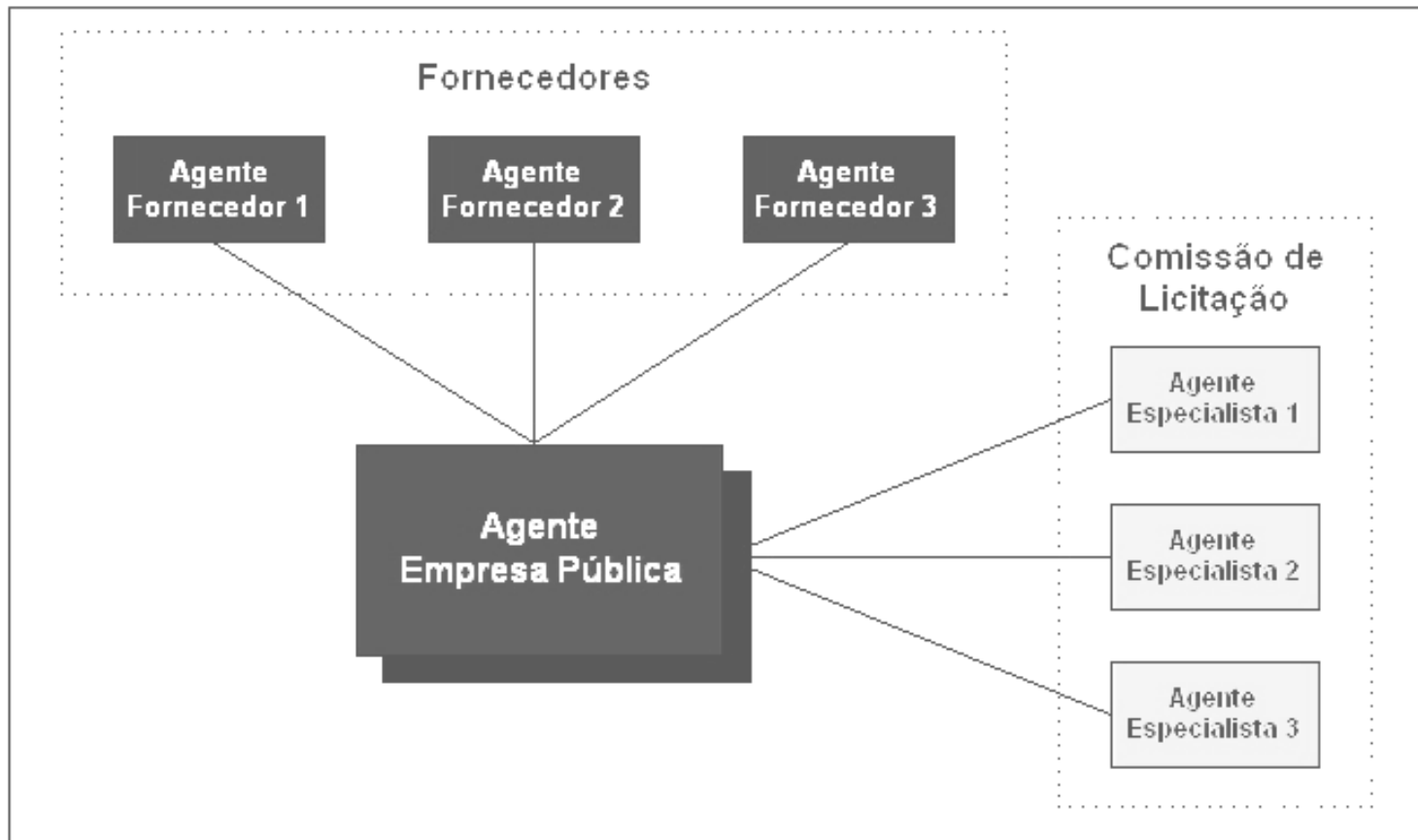
Exemplo de sistema

- **Sistema de Fornecimento de Água para Caldeiras - Resolução Distribuída de Problemas**
 - O importante é atingir o objetivo global;
 - As tarefas são pré-definidas
 - A ênfase é dada na modelagem de cada agente
 - Usualmente não existe grande comunicação entre os agentes;
 - Existe um controle global

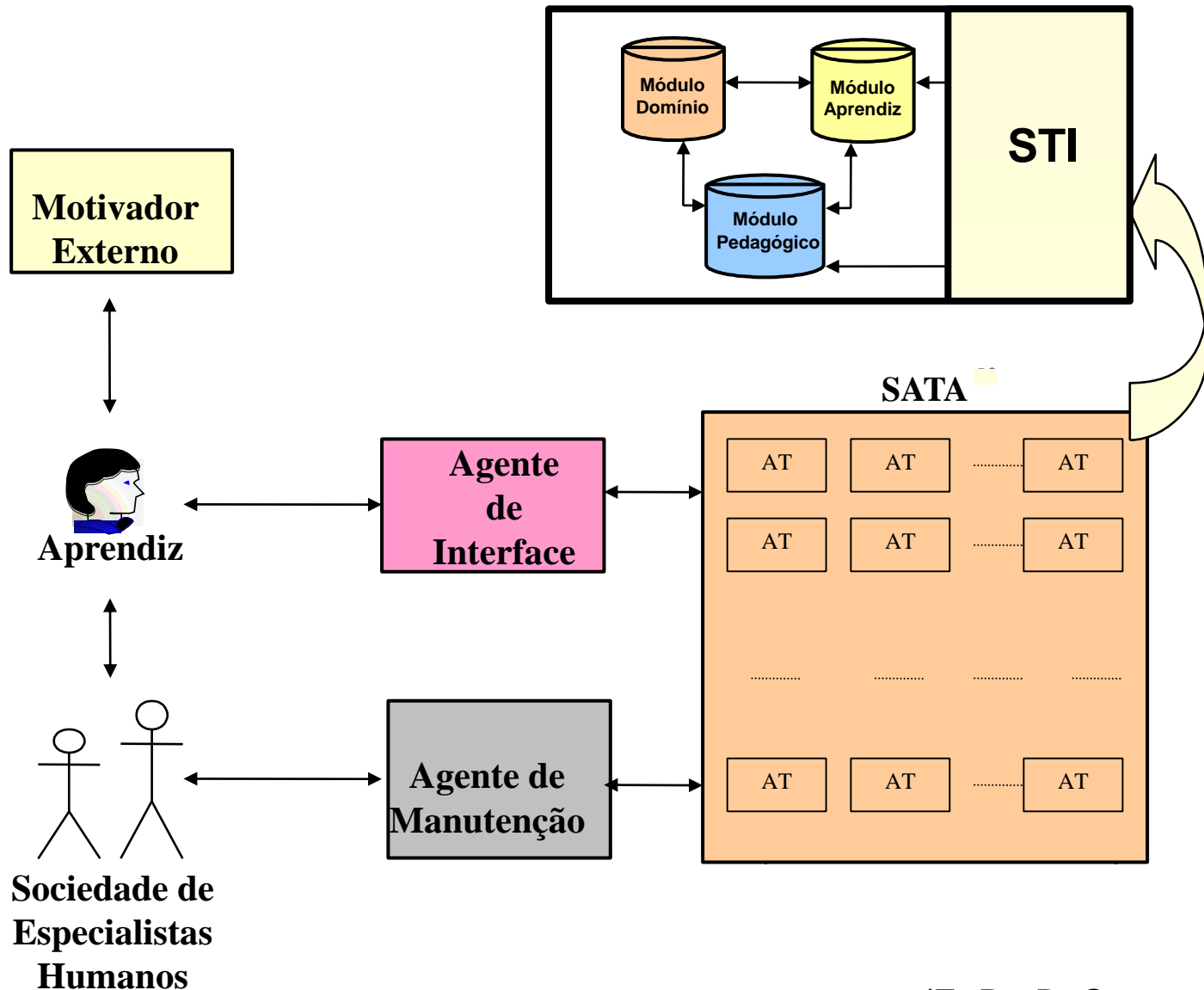


Exemplos de sistemas

- Um ambiente interativo multiagente para licitação pública



Exemplo de Sistemas



Exemplo de sistemas

- **Formigueiro - Sistema Multiagentes Reativo**

- O mais importante é o problema;
- O problema é resolvido por nós individuais que interagem entre si;
- Existe um grande número de nós;
- Os nós são normalmente idênticos e possuem conhecimento limitado;
- Cada nós não tem consciência do problema geral;
- Os nós ‘cooperam’ entre si;
- A solução ‘surge’ através das interações entre os nós.

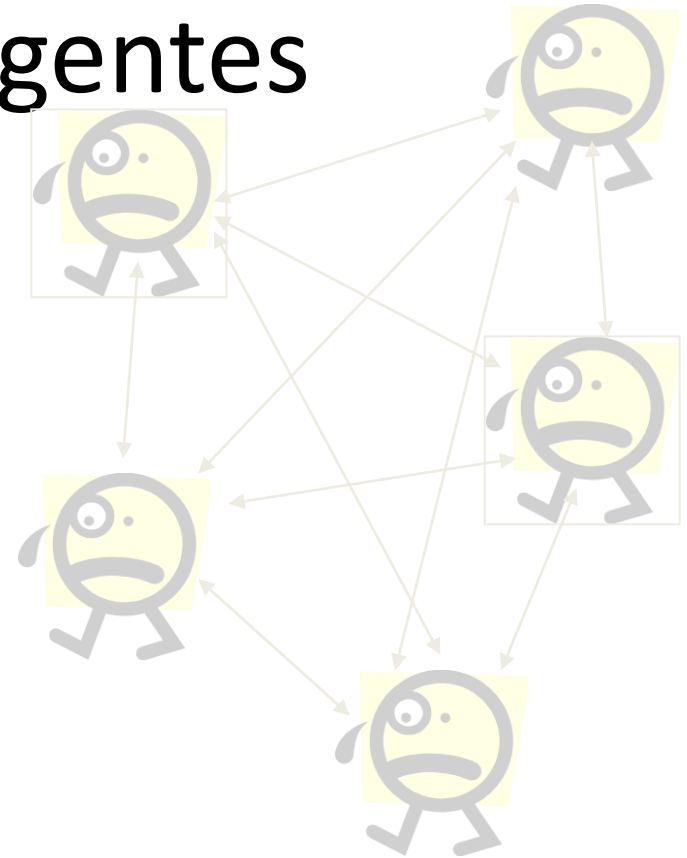
Exemplo de sistemas



- **Negociação trabalhista - Sistema Multiagentes Cognitivo Não-Cooperativo**
 - Os objetivos de cada parte são usualmente contrapostos;
 - A informação de cada parte é incompleta;
 - Existe um objetivo global desejado, mas que não é mais importante que os objetivos individuais;
 - Cada parte procura convencer seu oponente para que ele ceda (ocorre um processo de negociação);
 - Não dá para ter certeza sobre o que a outra parte vai fazer;
 - Não existe um controle centralizado do processo;

.. Continuação.

Sistemas Multiagentes



Programa

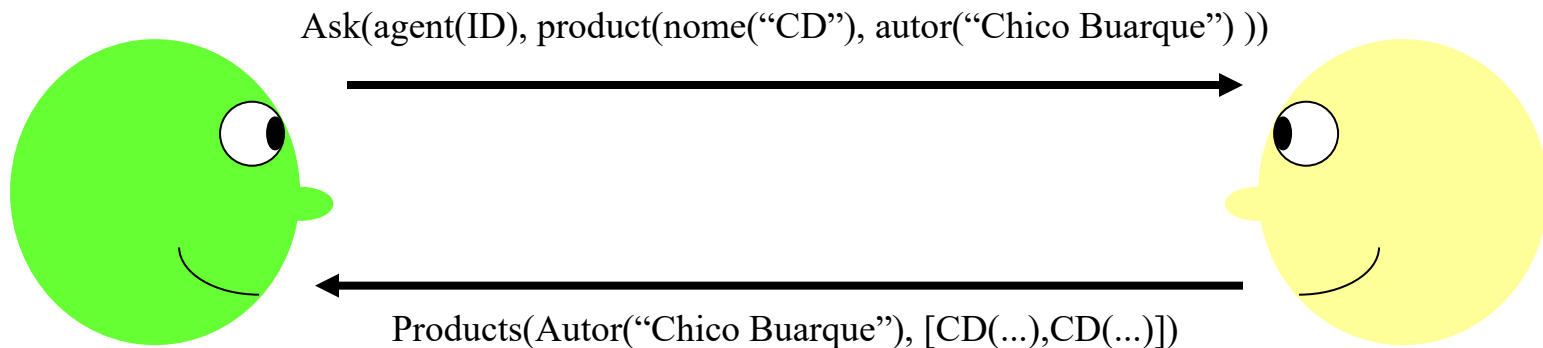
- Introdução: problemas de SMA;
- Processo de desenvolvimento de agentes;
- Organizações;
- Alocação de tarefas ;
- Interação entre os agentes;
- Resolução Cooperativa Distribuída de Problemas;
- Resolução dos conflitos;
- Cooperação entre agentes;
- Negociação entre Agentes;
- Classes de Coordenação de Negociação (RdC);
- Um exemplo.

Problemas dos SMA

- Como descrever e alocar as entidades de um problema entre os agentes do sistema?
- Como habilitar os agentes para se comunicar e interagir?
- Como garantir que os agentes vão agir de forma coerente, sem provocar efeitos indesejáveis?
- Como habilitar os agentes para representar e raciocinar sobre as ações de outros agentes para conseguir se coordenar com eles?
- Como gerenciar a limitação de recursos de cada agente?
- Como construir um SMA de forma efetiva? Como definir plataformas tecnológicas e metodologias de desenvolvimento?

Processo de Desenvolvimento de um SMA

- Similar a divisão-e-conquista;
- Busca-se definir:
 - Quem são os agentes
 - Protocolo de comunicação entre os agentes
 - Regras de funcionamento de cada agente (Relações entre: crenças, percepções e ações)

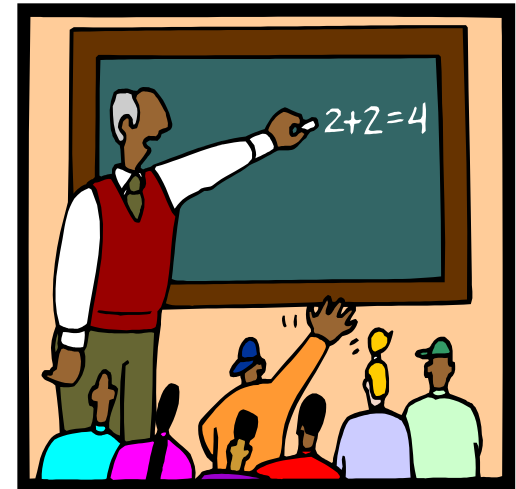


Algumas características básicas de SMA

- A comunidade é desenhada para **cooperar** em um ambiente aberto, onde cada agente tem **autonomia** e pode, eventualmente, participar na resolução de um dado problema.
- Um agente pode solucionar sozinho uma determinada classe do problema.
- Competem entre si por recursos, tendo assim que saber **lidar com conflitos** e **coordenar suas atividades** para aumentar a eficiência na solução do problema.
- Não necessitam utilizar a mesma linguagem, o que, por outro lado implica na **necessidade de traduções e mapeamentos** para as suas representações individuais.

Organizações de SMA

- As organizações oferecem *frameworks* para a interação de agentes por meio de:
 - Papéis
 - Comportamentos esperados
 - Relações de autoridade

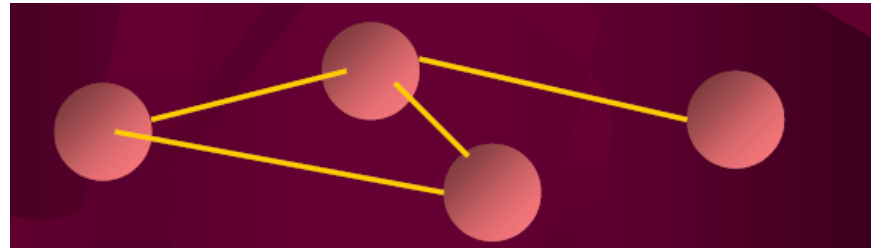


Visões Organizacionais

- Em geral:
 - Vistas em termos de estrutura (padrões de informação) e relações de controle.
- Teoria da Organização:
 - Um conjunto de agentes com compromissos mútuos, objetivos e crenças globais
- A organização de um sistema multiagente pode ser classificada como:
 - democrática
 - federada
 - hierárquica

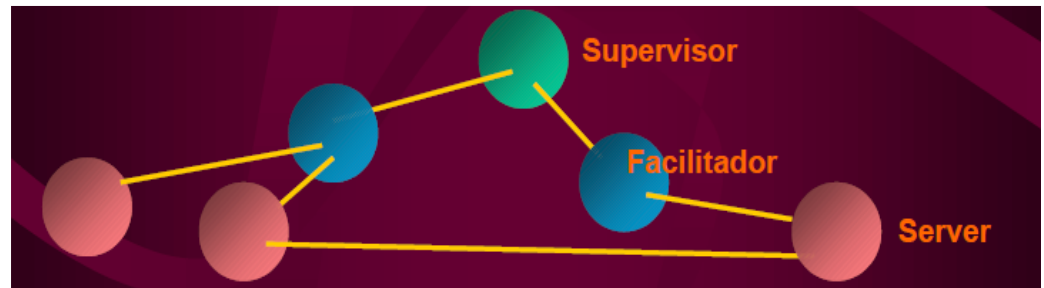
Organização Democrática

- Também chamada de organização “anárquica”.
- Agentes “sem organização”: *Flat*
- Normalmente os agentes tem graus similares de independência e autonomia, porém **sem** hierarquia alguma.
- Podem ser tanto bastante homogêneos como bastante heterogêneos.
- Ex.: sistema multiagente representando empresas independentes.



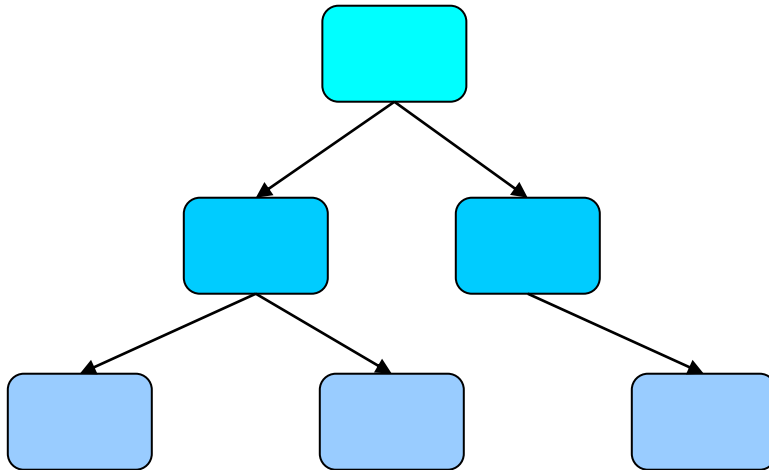
Organização Federada

- Agentes possuem algum tipo de hierarquia.
- Presença de um Facilitador, agente-intermediário entre o “cliente” (agente(s) supervisor(es) e o “servidor” (agentes executores).
- Grupos de agentes da “Federação” tem graus similares de independência e autonomia.
- Normalmente são bastante bastante heterogêneos.
- Procedimentos de coordenação de média complexidade dado que a própria existência do facilitador tende a garantir mecanismos mais rápidos de convergência e/ou distribuição.
- Ex.: sistema multiagente representando um sistema de supervisão de subestações elétricas.



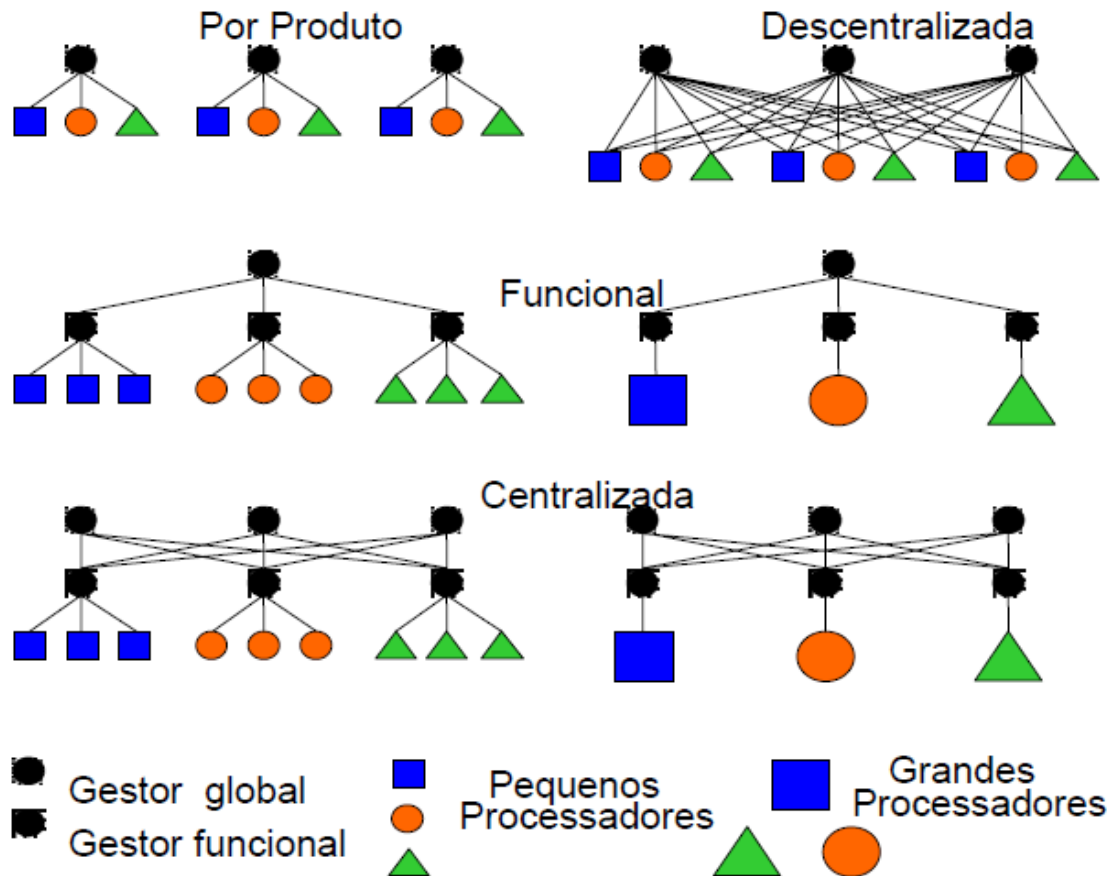
Organização Hierárquica

- Um único agente detém toda a autoridade
- A comunicação ocorre verticalmente



Em termos de aplicações industriais e comerciais reais, as organizações hierárquicas são as mais utilizadas.

Organização Hierárquica



Organização Hierárquica

No primeiro tipo, **por produto**, cada produto (“objetivo”) específico tem processadores (“agentes”) pré-estabelecidos.

No segundo, **descentralizada**, os vários processadores são partilháveis por qualquer produto, com o gestor global contactando diretamente os processadores para uma posterior seleção dos mais adequados.

No terceiro e quarto, **funcional**, o gestor global delega a função a um gestor funcional, que então se encarrega de contactar e seleccionar os processadores (“agentes”) mais adequados, sejam eles de reduzida ou de grande capacidade.

No quinto e sexto, **centralizada**, existe o mesmo comportamento que o funcional. Porém, aqui o gestor global entra em comunicação com o gestor funcional para que este se encarregue de contactar e seleccionar os Processadores (“agentes”) mais adequados, sejam eles de reduzida ou de grande capacidade. Após isto ter sido feito, o gestor funcional deve comunicar ao gestor global acerca dessa seleção.

Organização Hierárquica

- **Avaliação**

- **Critérios de Avaliação de Desempenho:**

- Custos de produção
 - Custos de coordenação
 - Custos de vulnerabilidade

TAREFA

- Descreva o problema
- Quais os agentes? Papéis/funções
- Como é a interação/ comunicação (Protocolo)
- Como será a organização
 - Democrática/federada ou hierárquica

Organização Hierárquica

- **Métrica Custos de Produção**

Os **Custos de Produção** referem-se aos **custos para se realizar uma tarefa**, considerados aqui como um índice de espera para uma dada tarefa ter seu processamento iniciado.

Em termos práticos, a este índice podem ser associados custos, que por sua vez podem ser expressos em termos de estoque intermediário, atraso no pedido dos clientes, entre outros.

- **Métrica Custos de Coordenação**

Os Custos de Coordenação referem-se aos custos de tomadas de decisão e comunicação necessários para coordenar as tarefas, considerados proporcionais ao número de mensagens necessárias para atribuir uma tarefa a um dado “processador”.

- **Métrica Custos de Vulnerabilidade**

Os Custos de Vulnerabilidade referem-se aos custos causados com interrupções de execução provocadas por falhas inesperadas nos agentes (“processadores”).

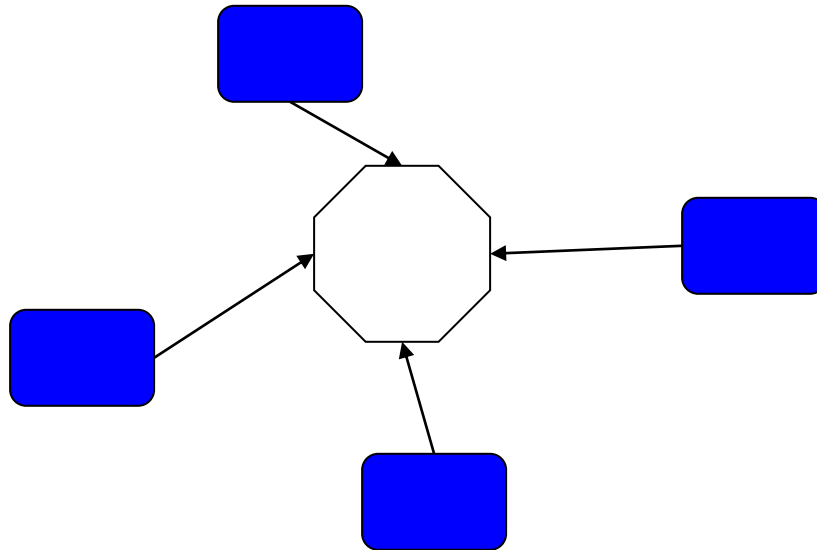
Organização Hierárquica

- **Avaliação**

Tipo de Hierarquia	Custos de Produção	Custos de Coordenação	Custos de Vulnerab.
Por Produto	Ruim	Bom	Razoável
Descentralizada	Razoável	Ruim	Bom
Centralizada (pouca capacidade)	Razoável	Razoável	Razoável
Funcional (pouca capacidade)	Razoável	Razoável	Razoável
Centralizada (grande capacidade)	Bom	Razoável	Ruim
Funcional (grande capacidade)	Bom	Razoável	Ruim

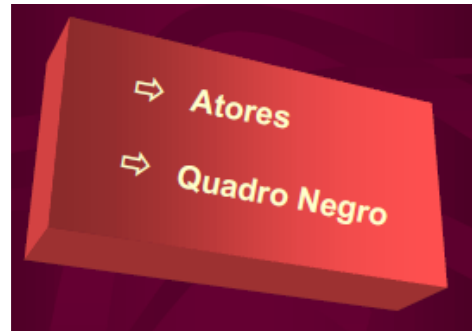
Comunidade de especialistas

- O agente é um *expert* na sua atividade.
- Controladas por “regras de ordem”
- A organização é plana.
 - Exemplo: arquiteturas blackboard



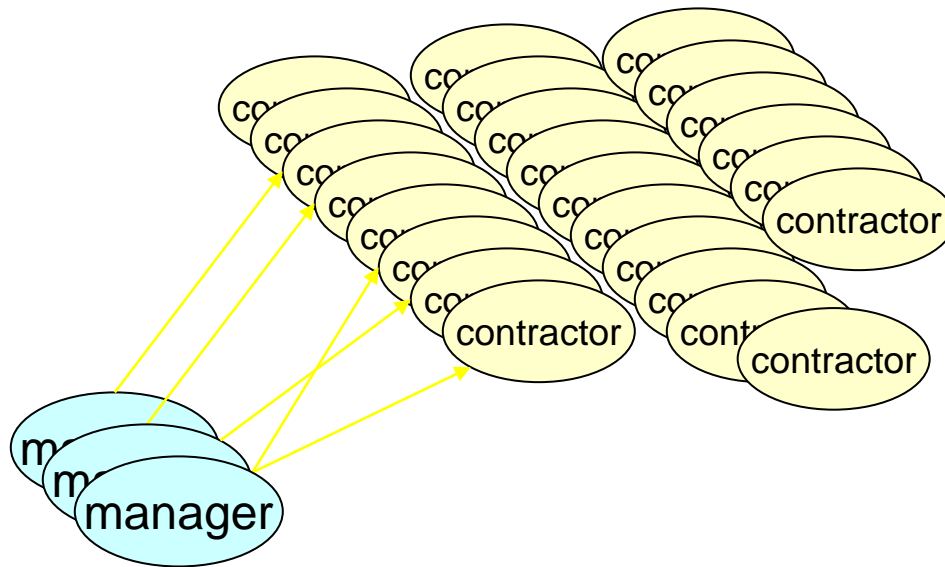
Quadro Negro (Blackboard)

- Apesar dos vários tipos de implementações existentes, os sistemas baseados em **Quadro Negro** [Hayes-Roth, 85] são utilizados em várias aplicações mais numa perspectiva de controle de execução e construção de novas soluções com progressivas correções.



Market-based

- Os agentes competem por recursos através de leilões e contratos



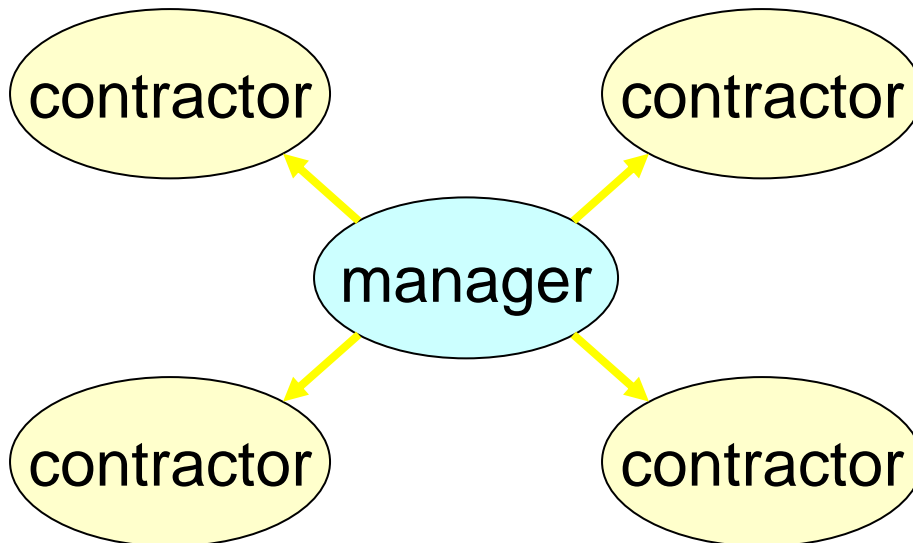
Alocação de tarefas

- Como atribuir responsabilidades e recursos para melhorar a eficiência e a coerência das soluções?
- Exemplos
 - Método Dinâmico
 - Método Planejado

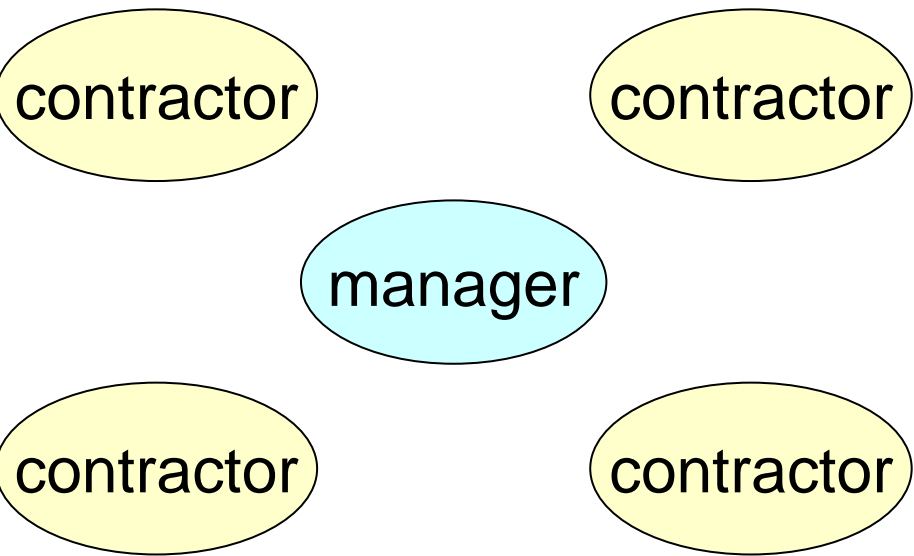
Alocação de Tarefas - Dinâmica

- Alocação dinâmica de tarefas
 - Exemplo - **contract net protocol** (CNP)
 - Gerente
 - Contratador

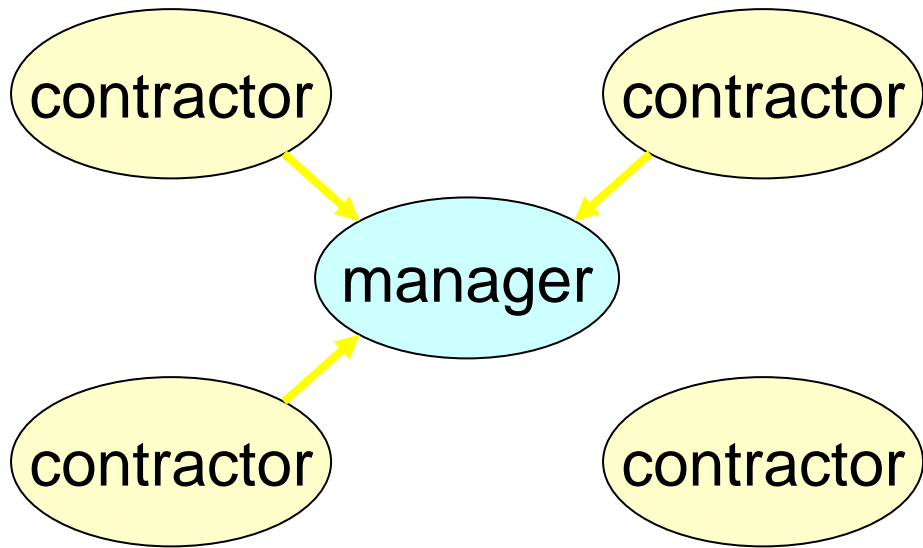
Step 1 – manager send request for bids



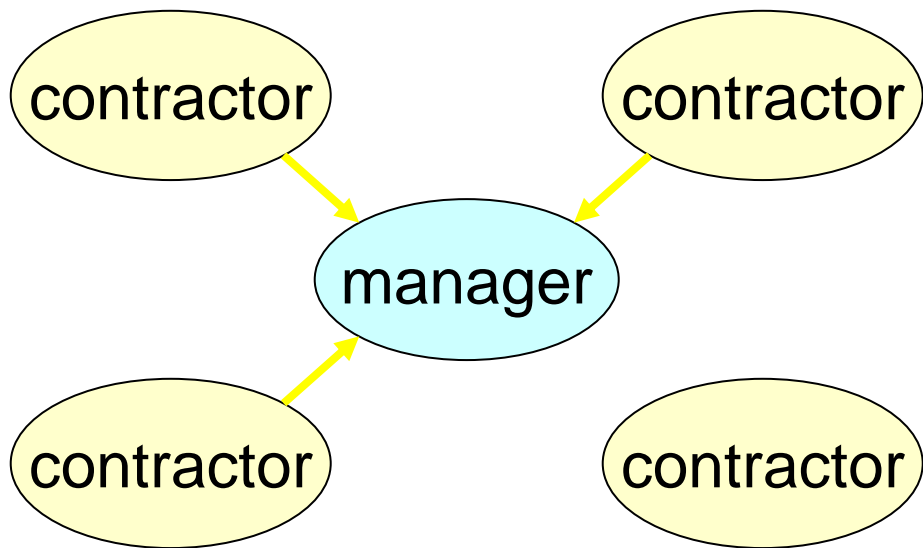
Step 2. – contractors deliberate



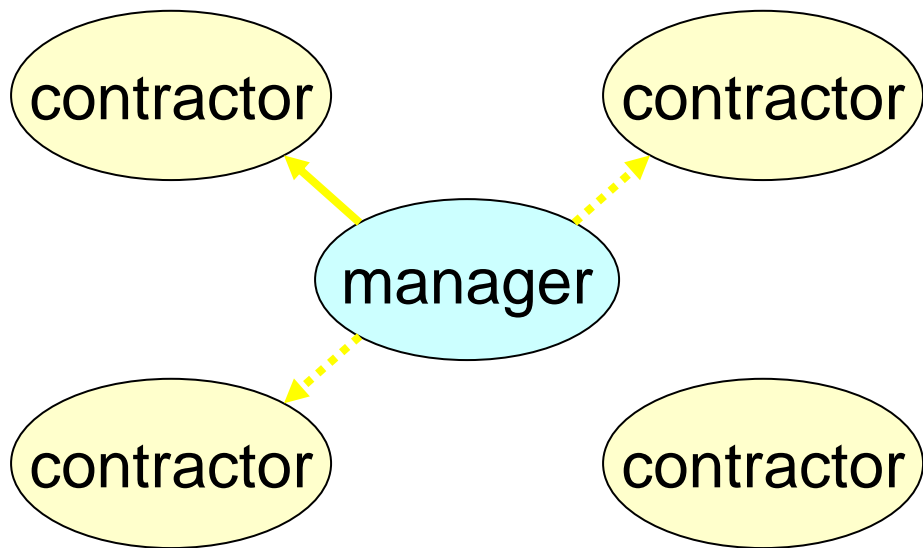
Step 3 – contractors respond with bid



Step 4 – manager selects winner

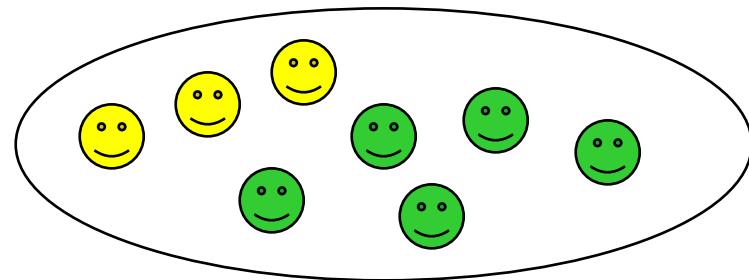


Step 5 – manager notifies winner (& losers)



Alocação de Tarefas - Planejada

- O planejamento multiagente deve considerar:
 - Restrições que as ações de outros agentes impõem sobre a ação de cada agente.
 - Restrições que os compromissos assumidos por um agente impõem sobre suas ações.
 - Evolução imprevisível do mundo, causada pela ação de outros agentes.



- Problema
- Quantos e quais agentes (tipos)
- Qual o papel de cada agente.
- Como é o protocolo de comunicação/interação
(Quem pode se comunicar com quem?)
- Arquitetura dos agentes
 - Democrática/Federada/Hierarquica.

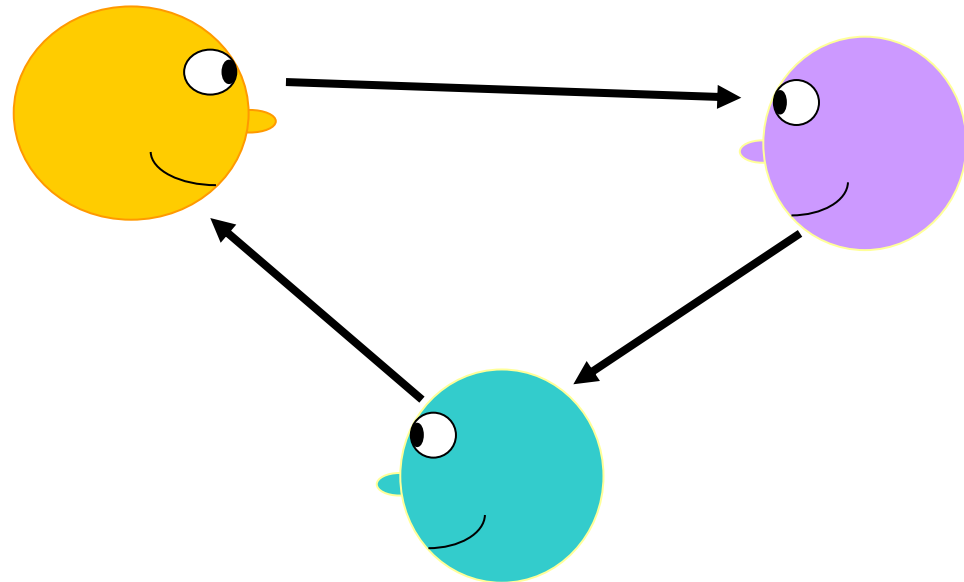
Conflitos

- A detecção e correção de disparidades e inconsistências é difícil.
- A principal abordagem para a correção de conflitos tem sido a negociação:
 - Assume agentes auto-interessados, com racionalidade limitada e informação incompleta.
 - Agentes trocam propostas e contra-propostas.



Métodos de Resolução de Conflitos

- Abstração e Generalização
- Obtenção de Objetivos Comuns
- Verificação de Pressupostos
- Mediação
- Relaxamento de Restrições
- Argumentação
- Convenções
- Padronização
- Votação
- **Negociação**
- ...



Resolução de Conflitos

- Conflito é a situação gerada dentro de um agente quando, visando resolver seu problema, ele se defronta com a ausência ou insuficiência de:
 - *a Objetivos Compatíveis e/ou*
 - *a Recursos Suficientes e/ou*
 - *a Habilidades Suficientes*

*Para isso, ele busca **cooperação**, na forma de interação.*

*Notar que para tal tarefa de raciocínio, é fundamental a qualidade da **representação do conhecimento** (por parte do agente) dele mesmo e dos outros agentes*

Interação entre Agentes

- É a capacidade de comunicar-se com outros agentes, usuários e sistemas visando atingir seus objetivos.
- Durante o planejamento da interação deve-se lembrar que a informação pode ser incompleta, imprecisa e ou previsiva, e a qualidade dela varia de acordo com o tipo de agente.
- A interação pode ser dividida em 4 camadas de complexidade:
 - Comunicação
 - Coordenação
 - Cooperação
 - Colaboração

Interação entre Agentes

- **Agente com agente:** agentes do mesmo SMA que se comunicam através de um protocolo geralmente proprietário, tornando mais leve a comunicação realizada; pois são transmitidas somente informações relevantes para o sistema.
- **Agente com usuário:** utilizado quando o agente precisa da informação de um usuário para completar uma tarefa. Normalmente realizado através de uma interface gráfica.
- **Agente com SMA:** agentes que se comunicam com outros agentes acessam serviços “exportáveis” pelo SMA utilizando protocolos padrão para haver compatibilidade e interoperabilidade entre os sistemas.

Interação entre Agentes

- **SMA com sistemas:** um agente se comunica com um sistema de informação no estilo cliente-servidor. O agente acessa serviços disponibilizados pela API (*Application Program Interface*) do sistema para colher informações necessárias a ele.
- **Agente com equipamentos industriais:** onde agentes trocam informações com máquinas para que estas realizem tarefas ou para colher informações, por exemplo do estado desta, usando protocolo de chão de fábrica .

Cooperação entre agentes

- Existem dois tipos de cooperação:
 - partilha de informação
 - partilha de tarefas.
- A partilha de informação se dá quando um dado agente dispõe ou produz informações parciais que julga serem úteis a partilhar e as envia para os outros agentes.
- A partilha de tarefa é efetuada quando um dado agente, ao decompor uma dada tarefa, detecta subtarefas que não pode ou não quer realizar, sendo necessário procurar outros agentes que possam auxiliá-lo.

Cooperação entre agentes

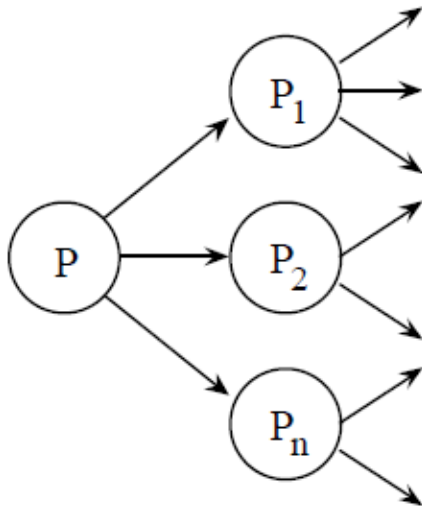
- Os agentes buscam cooperação, na forma de interação quando ocorrem conflitos.
 - Conflito é a situação gerada dentro de um agente quando, visando resolver seu problema, ele se defronta com a ausência ou insuficiência de recursos ou habilidades.
- Nesta camada as informações trocadas com outros agentes podem ser diferentes e tem influência no resultado final.
- A camada mais refinada é a **colaboração**. Uma colaboração é feita quando um agente tem capacidade de detectar possíveis objetivos comuns, e poder planejar sua agenda com os outros de forma a atingir o objetivo da melhor forma possível, aproveitando ao máximo a partilha de informações.
- A camada cooperação é uma camada encontrada apenas em sistemas baseados em agentes, onde a cooperação reflete uma estratégia de ação decidida pelo agente, permitindo a negociação.

Resolução Cooperativa Distribuída de Problemas

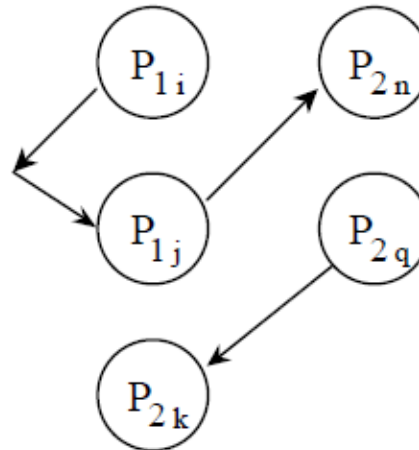
- Dividir para conquistar: Um problema é dividido em subproblemas e cada um é executado separadamente por um “processador” (um *agente*), cada um destes comunicando ou cooperando entre si quando necessário, com a idéia básica de que a soma dos resultados locais corresponderá à solução do problema geral.
- Fases de uma Resolução Cooperativa Distribuída de Problemas
 1. Um dado problema P é subdividido em subproblemas P_i , P_{ij} e assim sucessivamente;
 2. Subproblemas são distribuídos para um grupo de agentes A_{ik} ;
 3. Agentes A_{ik} processa-os.
 4. Soluções locais de cada agente são sintetizadas para fins de geração de uma solução global.
- A resolução por completo de um problema pode requerer a passagem por estas quatro fases em vários ciclos.

Fases de uma Resolução Cooperativa Distribuída de Problemas

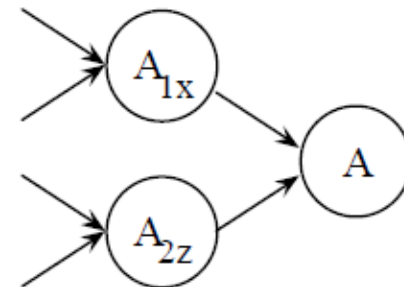
Decomposição (1)
e Distribuição (2)
do Problema



Solução dos
Subproblemas (3)



Síntese dos
Resultados (4)



Fases de uma Resolução Cooperativa Distribuída de Problemas

1) Decomposição do Problema

Esta fase consiste em subdividir o problema inicial em uma hierarquia de subproblemas, normalmente representada por uma árvore E-OU.

Um nó -E- é resolvível se todos os seus filhos também o são, enquanto que um -OU- é resolvível se pelo menos um dos nós-filhos o é.

Há vários métodos de decomposição.

2) Distribuição dos Subproblemas

Esta fase é basicamente responsável por identificar quais as tarefas que cada nó processará (“problema da conexão”).

Um conhecimento acerca das capacidades e estado de cada nó candidato é necessário para a decisão. De forma subjacente, considera-se que o principal objetivo desta fase é o de balancear a carga computacional e reduzir o tempo total de computação.

Há vários critérios para alocação de tarefas (evitar gargalos, K fitting, urgência, carga da rede, etc.) e de decisão do melhor agente.

Fases de uma Resolução Cooperativa Distribuída de Problemas

3) Solução dos Subproblemas

Esta fase tem a ver com a solução de cada subproblema, nível por nível, do último para o primeiro (*bottom-up*).

- Um objetivo importante desta fase é o de reduzir o tempo “parado” (de não processamento), um fator essencial no desempenho geral da rede.
- Visando atingir este objetivo, há que se determinar o tempo de execução de cada tarefa (“problema do tempo”).
- Normalmente o número de subproblemas a serem resolvidos é muito maior do que o número de nós disponíveis na rede. Ou seja, um nó pode ter mais do que uma tarefa à espera em sua “fila de execução”.
- A partir do momento que os resultados dos níveis inferiores funcionam como dados de entrada para os níveis superiores (e assim, como habilitadores para suas consequentes execuções), acaba por haver a necessidade de um estabelecimento de uma apropriada ordem de execução (através de algum critério) à medida que os nós vão resolvendo os subproblemas.

4) Síntese dos Resultados

Esta última fase consiste em coletar e interpretar os resultados provenientes dos nós inferiores, à medida que estes os vão produzindo.

Cooperação em Sistemas Multiagente

Cooperação ... Mas afinal, será mesmo que os agentes devem Cooperar ?

Duas situações básicas onde há a necessidade de cooperação:

- A primeira é quando um agente precisa da ajuda de outro(s) para a solução do seu problema;
- A segunda é quando outro(s) agente(s) precisa(m) da ajuda deste.

Por outras palavras, uma cooperação permite que “novos” recursos, capacidades e oportunidades sejam criadas / aumentadas, contrariamente ao que aconteceria se caso não houvesse.

Cooperação em Sistemas Multiagente

Por outras palavras, a Cooperação é importante ...

- Porque as tarefas e as consequentes ações dos agentes executores são interdependentes;
- Porque a solução de subproblemas requer que se respeitem restrições globais;
- Porque é preciso juntar “forças” (com complementaridade, *holisticamente*) em termos de competências, recursos e informação de vários agentes, visando a solução de um problema.

Cooperação em Sistemas Multiagente



Observar que a Cooperação
pode ser apenas um
comportamento “esperto” de
um agente egoísta ou
destrutivo !

Algumas propriedades dos Agentes :

- **(semi-)autônomo:** algumas das suas decisões e ações dependem de outros agentes;
- **cooperante:** atua de uma forma cooperante (e não “egoísta” ou “destrutiva”) com os demais agentes na solução de um problema;
- **benévolo:** está sempre interessado em cooperar;
- **responsável:** uma vez assumido um compromisso, o agente fará todos os esforços para o cumprir;
- **racional:** atingirá uma solução (correta, porém não necessariamente a mais correta);
- **honesto:** não “mente” em uma interação;
- **deliberativo:** raciocina e age consoante ao seu estado interno (contrário aos puramente “reativos”), através de uma manipulação simbólica do modelo do mundo.
- **auto-supervisor:** monitora a sua própria existência e estado.
- ...

Propriedades ? E daí ?!

- Relação das propriedades do agente com o comportamento que o sistema multiagente desejado precisa exhibir.
- A suas considerações geram um impacto direto nas características de toda a infraestrutura de automação que será necessária (redes, protocolos de baixo e alto níveis, rigidez dos mecanismos de privacidade e segurança de informações, etc.), pois esta deverá estar preparada para “emulá-las” / suportá-las.

A Negociação

- A **Negociação** é um de uma série de métodos através dos quais disparidades entre agentes são acordadas quando se verificam incertezas ou conflitos.
- A noção do uso de **negociação como uma meta-forma** para resolução distribuída de problemas foi introduzida em [Davis, 83], sendo o enquadramento proposto baseado em um outro seu trabalho importante, no qual propõe um protocolo de suporte, chamado **Rede de Contrato (Contract net)** [Davis, 80], basicamente usado em problemas de alocação de tarefas.

A Negociação

As definições de gerais sobre **Negociação** apresentadas na literatura são de certa forma vagas, refletindo quase que o conceito intuitivo dos autores quando não aplicadas a um domínio ou aplicação específicos.

[Davis, 83] “uma discussão na qual partes interessadas trocam informações e chegam a um acordo”.

[Sycara, 88] “o processo de negociação envolve a identificação de potenciais interações feitas através ou de comunicação ou de raciocínio (baseado nos estados atuais e nas intenções dos demais agentes do sistema) e modificação das intenções dos agentes a fim de se evitar interações não desejáveis ou de se criarem situações de cooperação”.

[Lesser, 91] “o processo de melhorar a chegada a um acordo (reduzindo-se inconsistência e incerteza) sobre pontos de vistas comuns ou planos, através de uma troca estruturada de informação relevante”.

Em suma, a negociação é um paradigma de suporte à resolução de conflitos via troca de informação.

Classes de Coordenação de Negociação

1- Negociação em controle de tráfego aéreo

O objetivo é permitir que cada avião (um nó da rede) possa construir um plano de voo que mantenha uma adequada separação dos demais aviões em voo e que satisfaça outras restrições / objetivos, tais como chegar ao destino com o menor consumo de combustível possível, por exemplo.

2- Modelagem cognitiva da negociação

Consiste em suportar que os agentes, em uma troca de mensagens sobre uma certa proposta de compromisso, preponderantemente por iniciativa própria, possam: persuadir com argumentos, expressar concordâncias ou discordâncias em relação à proposta, requerer mais informações a respeito, relaxar certas restrições, etc..

Classes de Coordenação de Negociação

3- Rede de contrato (RdC) (Contrat-Net)

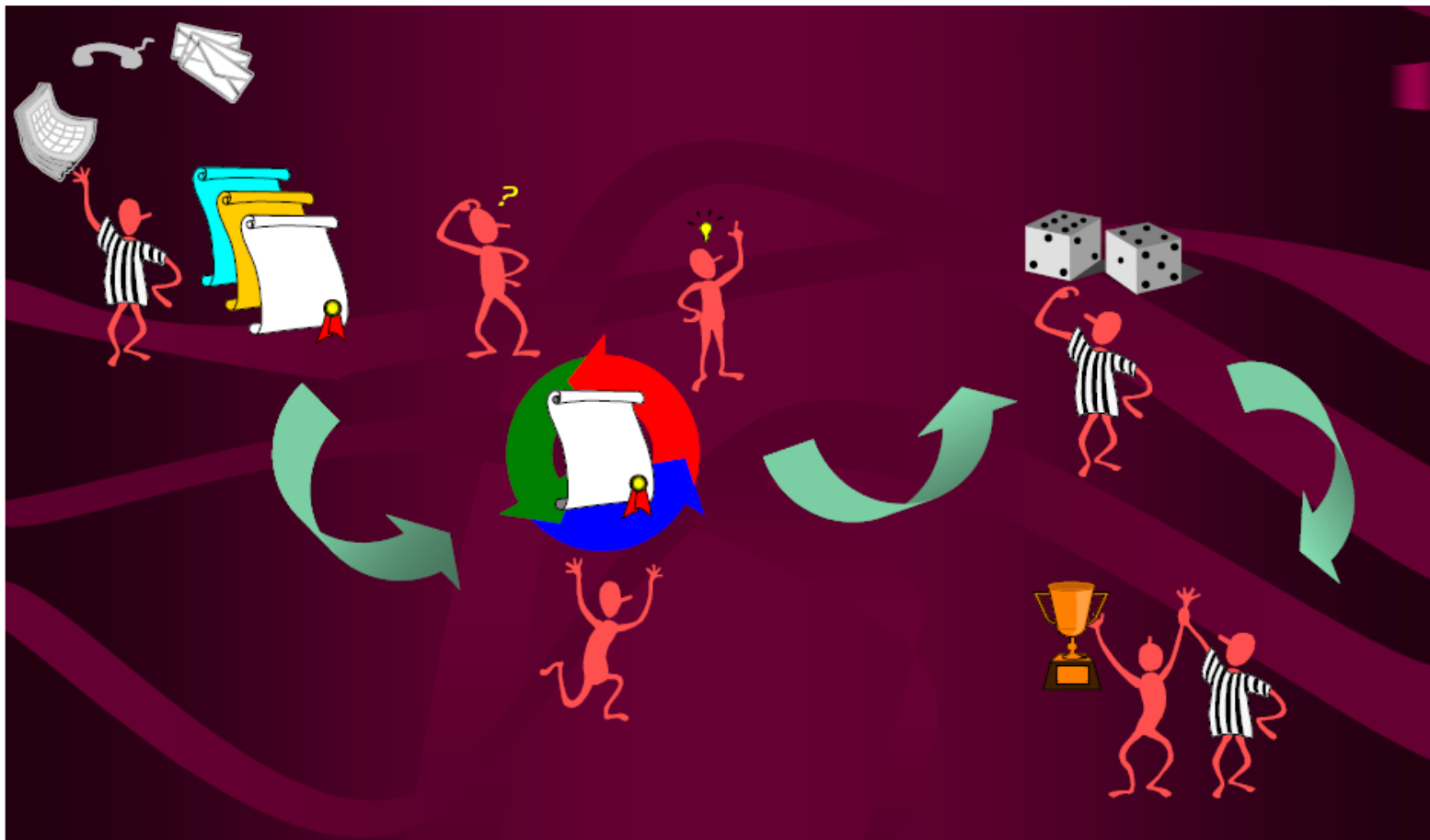
- Em uma rede de contrato, um nó (agente) pode desempenhar dois papéis: contratante e proponente/contratado. No primeiro caso, o nó é responsável por anunciar, monitorar e sintetizar os resultados de uma tarefa; no segundo, pela sua execução propriamente dita.
- Os nós coordenam suas atividades por intermédio de contratos por forma a atingir certos objetivos.
- Em cada estágio, o contratante decompõe os seus contratos em subcontratos, que deverão ser realizados por contratados.
- Este processo envolve um protocolo de “oferta” (*bid*), em uma filosofia de transferência de informação ida-e-volta (suportada por um protocolo de mensagens), a fim de estabelecer a natureza do contrato e determinar quais serão os contratados para os subcontratos.

Classes de Coordenação de Negociação

Rede de contrato (RdC) (Cont.)

- Os subcontratos são anunciados a nós (“ociosos” ou não), que recebem e avaliam a tarefa (associada ao subcontrato).
- Os nós que tenham recursos, conhecimento e informação apropriadas respondem ao contratante com uma oferta, indicando sua adequação à tarefa.
- O contratante recebe as potenciais várias ofertas, analisa-as e concede a tarefa ao nó cuja oferta seja a mais adequada.
- Ao fim de um processo cria-se um relação de controle contratante-contratado, distribuída ao longo da rede, com o contratante fornecendo informações sobre a tarefa e o contratado reportando o estado do andamento desta (e eventualmente o resultado da tarefa).

Rede de Contrato



Classes de Coordenação de Negociação

4- Negociação multi-estágio

Extensão da RdC.

Permite que os nós se comuniquem iterativamente durante o processo de oferta e concessão de uma tarefa.

Os nós tentam escolher ações locais para afetar e ligar *recursos de comunicação* e, posteriormente, tentam iterativamente trocar essas informações (apenas as relevantes) com outros nós.

A cada iteração cada nó avalia como suas escolhas locais bem como as feitas pelos outros nós afetam o re-estabelecimento dos *circuitos de comunicação*.

Através desta troca iterativa de informação, os nós convergem para escolhas compatíveis entre si (isto é, que atendam às restrições de todos) ou, se for o caso, reconhecem que o problema é super constrangido.

Áreas de Aplicação da Negociação

- Engenharia concorrente
- Telecomunicações
- Distribuição de energia elétrica
- Gestão de recursos em construção civil
- Robótica móvel
- Negociação automática em empresas virtuais
- Economia de mercado
- Compra-venda automática
- Escalonamento

Exemplo: Jogos dos 8 Números

H	E	A
B	F	D
C	G	

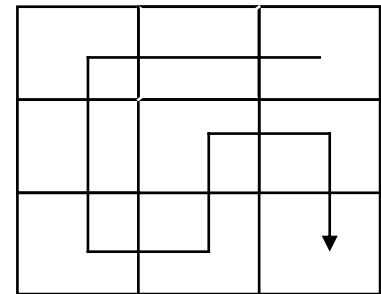
- Objetivo: Colocar as letras em ordem alfabética no menor tempo possível
- Este problema é np-completo

Jogos dos 8 Números

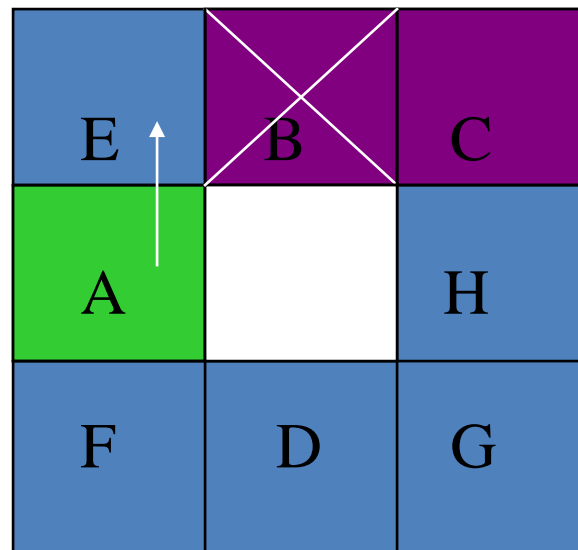
- Considerando cada agente como uma peça, temos que:
 - Cada agente tem como objetivo ir para a sua posição;
 - Cada agente sabe a sua posição atual e pode estar ou não ativo;
 - Apenas um agente está ativo de cada vez (A, em seguida B, etc.)
 - Cada agente pode:
 - ‘Atacar’ um outro agente;
 - Fugir quando for atacado;
 - A fuga só pode ocorrer quando o agente não estiver bloqueado;
 - Percorrer o caminho padrão, mesmo estando em sua posição.

Jogos dos 8 Números

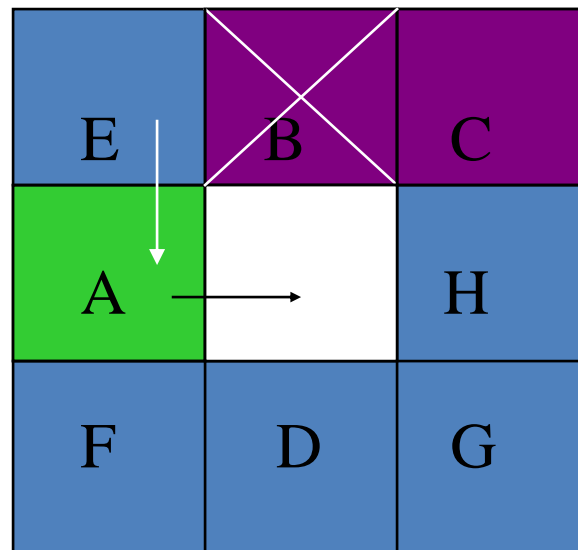
- O protocolo de comunicação é:
 - Ataque: ataque (atacante(...), posição_atacado(...))
 - Fuga: Fugir (posição_fuga(...))
 - Bloqueado: Block (atacante(...))
 - PercorrerCaminhoPadrão();



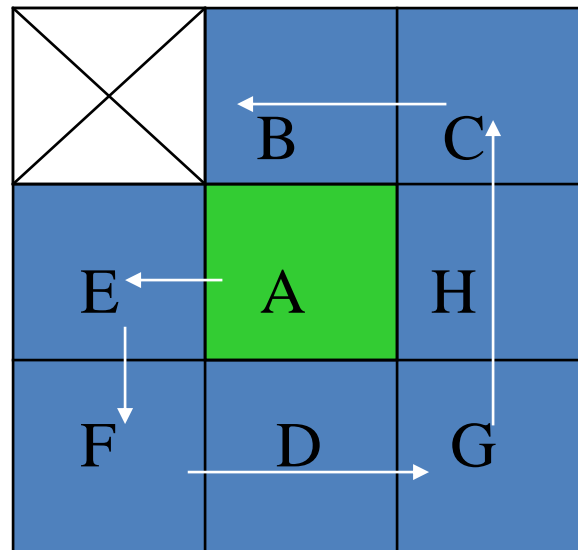
Jogos dos 8 Números



Jogos dos 8 Números

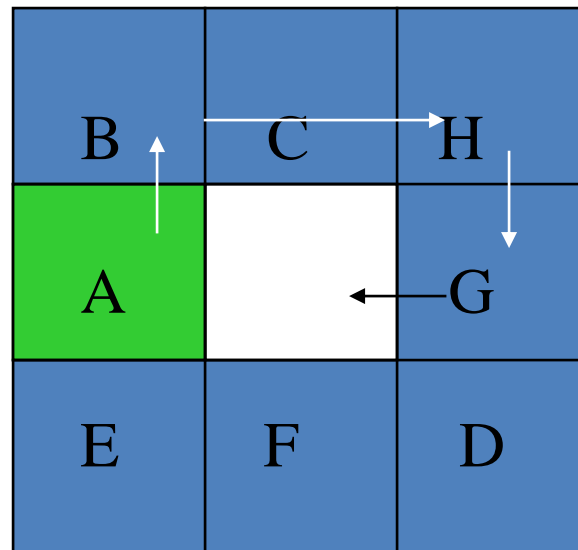


Jogos dos 8 Números



 Ativo

Jogos dos 8 Números



 Ativo

Jogos dos 8 Números

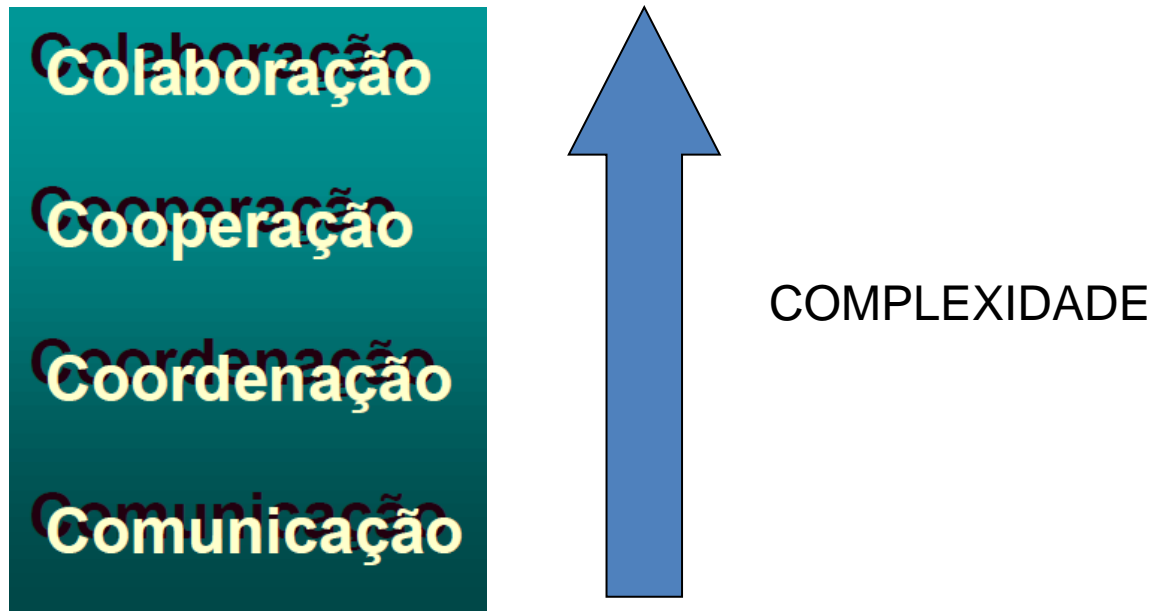
A	B	C
	G	H
E	F	D

	Bloqueado
	Ativo

Tarefa:



Camadas de Complexidade de Interação

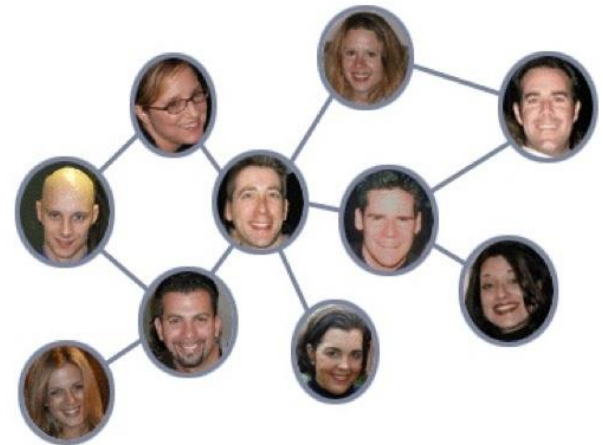


Camada Comunicação

É a camada básica de qualquer software que precisa interagir.

- O que se faz necessário aqui é uma linguagem de comunicação, com uma precisa sintaxe, e que todos têm que conhecer / igual para todos os envolvidos.
- Semântica é implícita, devendo ser acordada entre as partes envolvidas.

A semântica (do grego σημαντικός, derivado de sema, sinal) refere-se ao estudo do significado, em todos os sentidos do termo.



Camada Coordenação

- É a camada que define as regras de interação considerando as agendas / workflow dos agentes de forma a se evitar comportamentos indesejados.
- A Semântica é mais explícita, conhecida por todos.



Camada Cooperação

- É uma camada típica em sistemas baseados em agentes, não encontrada em outros tipos de sistemas.
- A cooperação reflete uma estratégia de ação, pensada / decidida pelo agente, o que pode levá-los até a negociar.
- Conforme os objetivos do agente e do conhecimento geral que tem (dele e do ambiente), as informações a serem trocadas com outros agentes podem ser diferentes e, conseqüentemente, tem influência no resultado final.



Camada Colaboração

É a camada mais refinada em sistemas baseados em agentes.

- Uma colaboração é feita quando um agente tem capacidade de detectar possíveis objetivos comuns e, assim, pode planejar sua agenda em conjunto / de acordo com os outros de forma a melhor atingir o objetivo, tirando o máximo proveito da partilha de informações.
- A diferença essencial com relação à cooperação é que esta pode ser vista mais como uma interação de “curto prazo”. Já na colaboração, a interação tende a ser bem mais longa, com potenciais várias mudanças nos estados internos dos agentes e nas regras de coordenação.
- Esta camada requer ainda maiores estudos, especialmente na formalização dos modelos de interação.

Protocolos de Baixo Nível

Camada Transporte:

- TCP/IP (*sockets, http, SQL, ...*)
- CORBA , DCOM
- ...

Encapsula os aspectos de baixo nível da rede, servindo de transporte da mensagem relativa à aplicação propriamente dita.

Protocolos de Alto Nível

Camada Aplicação (ACL: Agent Communication Language)

Contract-net

KQML

KIF

Protocolos proprietários

...

Contract-net

Primeiro protocolo inter-agente.

- Foi desenvolvido para suportar uma rede de contratos (*contract-net*), permitindo uma troca de mensagens associadas com um anúncio duma tarefa, recebimento desta, proposta de execução, recebimento de propostas, contratação e negociação.
- As mensagens do protocolo foram totalmente adequadas ao problema de sensoreamento distribuído em vigilância de tráfego aéreo e marítimo.
- Mensagens representadas na forma de objectos [Davis & Smith, 83].

Exemplo:	
Para	todos
De	nó-25
Tipo Mensagem	anúncio
Número Contrato	43-6
Critério Elegibilidade	"ter FFTBOX"
Descrição Tarefa	[[tipo_tarefa, transf_Forrier], [número_pontos,1024], [nome_nó,nó-25], [posição,[lat 64N long 10w]]]
Tipo Resposta Esperada	tempo_completação
Tempo de Expiração	[29 1645Z NOV 1980]

KQML

O “KQML” (*Knowledge Query and Manipulation Language*) [Finin94] pode ser considerado o **protocolo mais representativo** para suporte de comunicação inter-agente e há uma tendência que se torne um standard.

- O KQML **não foi originariamente concebido apenas para comunicação inter-agente**, mas para quaisquer sistemas autônomos que envolvam processos distribuídos / trocas de informação.
- Foi criado no âmbito do projeto “DARPA Knowledge Sharing Initiative”, com o objetivo de desenvolver técnicas e metodologias para construção de bases de conhecimento partilháveis, reutilizáveis e de larga dimensão.
- O KQML foi desenvolvido para poder ser aplicável a qualquer domínio de aplicação. É fornecido um conjunto de 34 mensagens genéricas - chamadas de “performativas” (*performatives*) - aplicáveis, teoricamente, a qualquer situação inter-agente. Dado a esta generalidade, o tratamento propriamente dito de cada performativa é da responsabilidade de cada aplicação em particular.

KQML

- cada performativa pode ser vista como uma “meta-mensagem”, pois não apenas encapsula o conteúdo duma mensagem propriamente dita, como também a caracteriza. Esta caracterização pode ser feita através de oito atributos, pertencentes ao próprio vocabulário KQML, instanciados de acordo com o tipo de mensagem.
- o **conteúdo propriamente dito é “encapsulado”** como sendo um atributo da performativa. Diferentemente da generalidade de outros protocolos, os atributos “sender” e “receiver” não são exatamente os mesmos que os típicos “from” e “to”. Correspondem ao receptor e emissor atual.
- Dadas as suas potencialidades, basicamente residentes na sua generalização e independência de domínio, muitos sistemas, conchas (shells), etc., começam a fornecer interfaces para KQML.

KQML

- Exemplo: Performativa *Tell*

tell		
:content	<i>(= (torque motor1) (sim-time 5) (scalar 12 kgf))</i>	comunica ao agente <A> que dispõe do "content" na sua base de conhecimento conteúdo da mensagem, no domínio " <i>motors</i> " linguagem de expressão de "content"
:language	<i>KIF</i>	ontologia a que se refere o "content"
:ontology	<i>motors</i>	identifica a qual mensagem a que esta se refere
:in-reply-to	<i>q1</i>	especifica o identificador que deve ser usado numa resposta
:reply-with	<i>nil</i>	
:force	<i>permanent</i>	indica que nunca irá desmentir o "content"
:sender	<i>< B ></i>	indica o (último) emissor da performativa
:receiver	<i>< A ></i>	indica o receptor (corrente) da performativa

KIF

- O “KIF” (*Knowledge Interchange Format*) não é, na verdade, um protocolo, mas sim uma linguagem “padrão” para especificação do conteúdo das mensagens propriamente ditas.
- Assim como o KQML, o KIF não foi desenvolvido única e exclusivamente para interações multiagentes, mas para quaisquer sistemas inteligentes que envolvam processos distribuídos / trocas de informação.

Ambientes de Desenvolvimento de Sistemas Multiagente

Contudo, ainda é incipiente o estudo de como sistematizar de forma metodológica o trabalho do projetista de um sistema multiagente.

As ferramentas atualmente disponíveis já partem do princípio que o projeto do sistema está feito, e então o passo é simplesmente modelá-lo (normalmente orientado por objeto)..

Vários desses Kits não são usáveis, na prática, pelas empresas. Algumas razões:

- são simples demais;
- são complicados demais;
- assumem pressupostos irrealísticos;
- infraestrutura pesada.

Ambientes de Desenvolvimento de Sistemas Multiagente

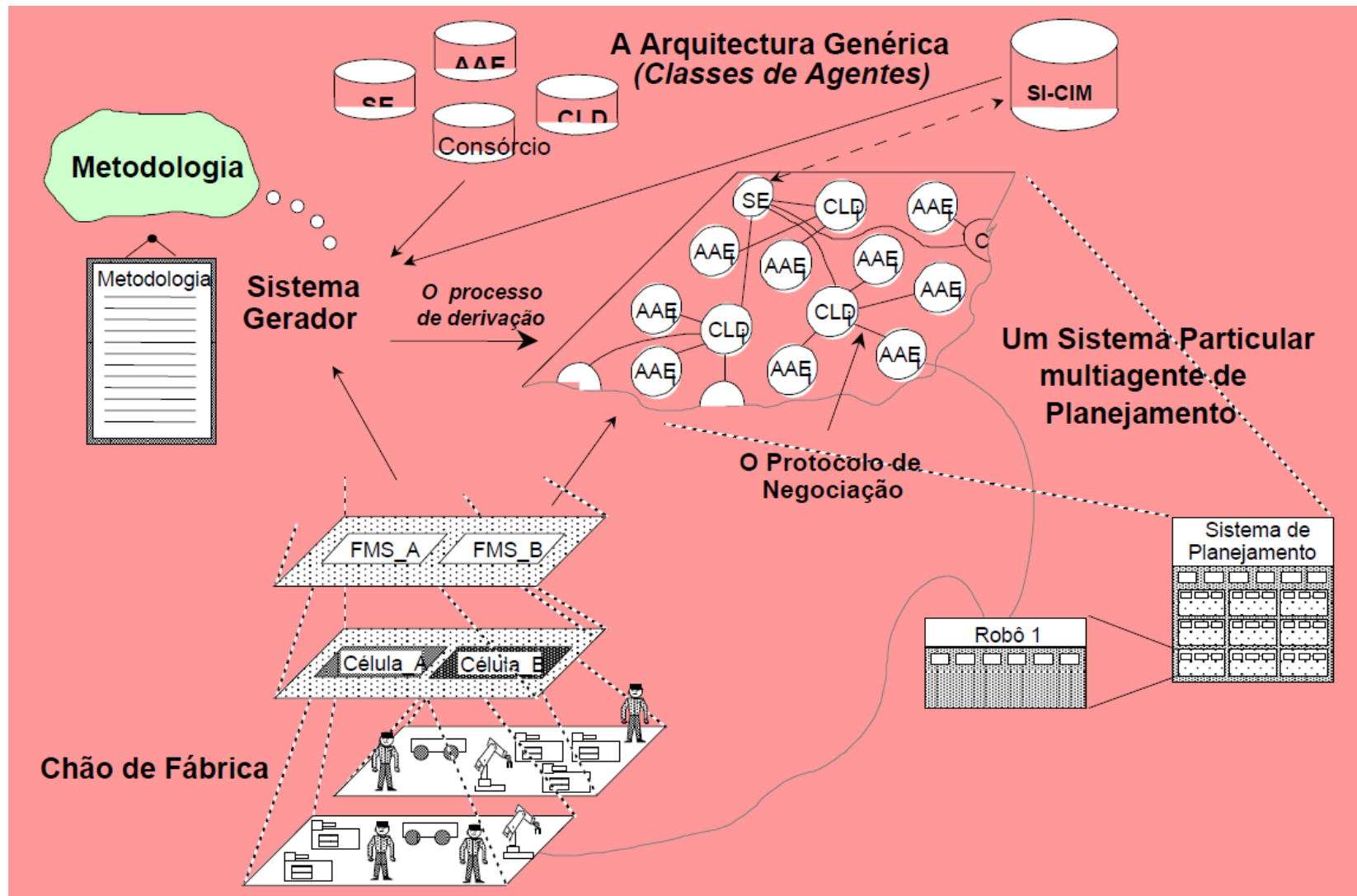
- *AgentBuilder*
- *ABE*
- *AgentTalk*
- *Aglets*
- *Concordia*
- *Java Intelligent Agent Library*
- *KAFKA*
- *LiveAgent*
- *Microsoft Agents*
- *Odyssey*
- *Voyager*
- *Agent Building Shell*
- *Agent TCL*
- *Cable*
- *Cybele*
- *dMARS*
- *Gypsy*
- *Infospiders*
- *JADE*
- *JAFMAS*
- *JATLite*
- *Kasbah*
- *LALO*
- *MASSYVE*
- *Mole*
- *Process Link*
- *Sodabot*
- *Swarm*
- *IMAJ*
- *Zeus*
- ...

Modelagem de um Sistema Multiagente

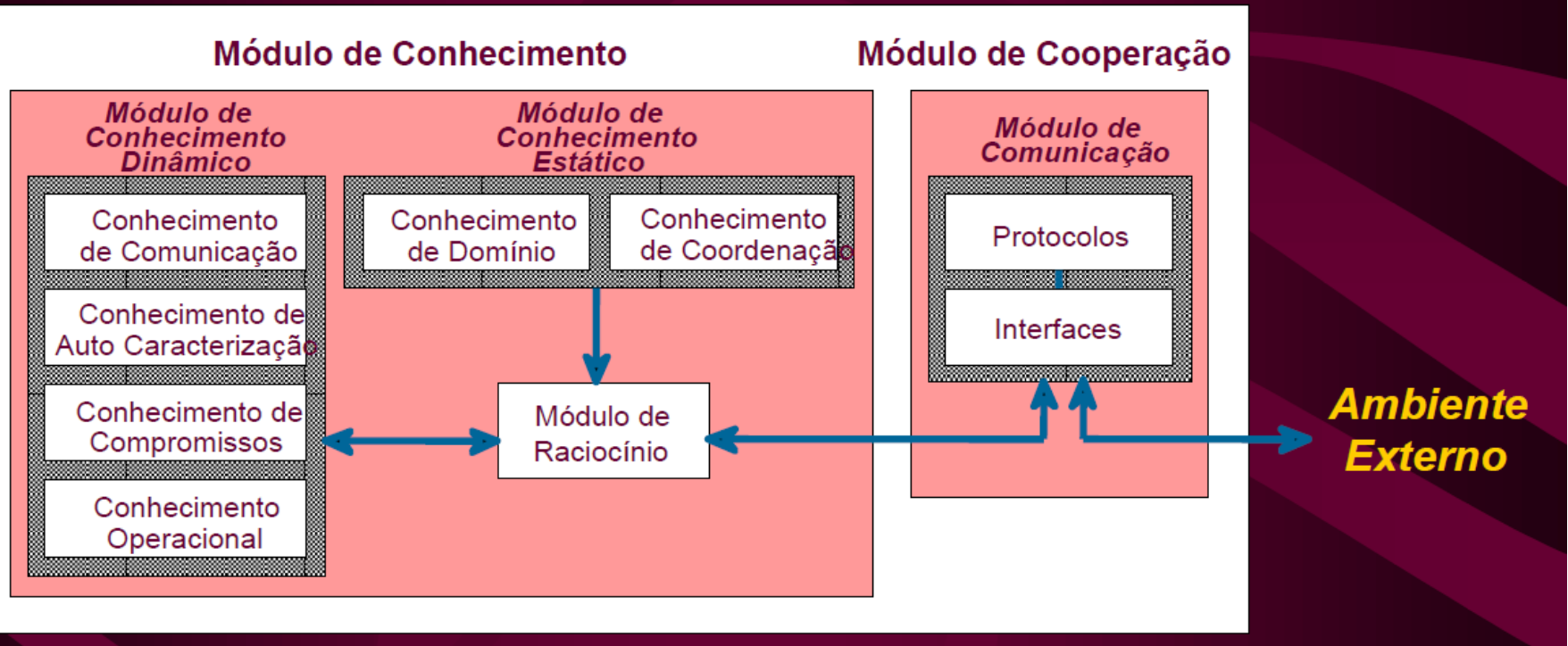
O projeto de um sistema multiagente deve considerar vários aspectos:

- Análise do **Tipo ou Classe do problema** a ser tratado;
- Definição dos **tipos de comportamentos** que os agentes deverão exibir e, por conseguinte, dos tipos de agentes necessários;
- **Níveis de autonomia** existentes e necessários dos agentes, e grau de interação com usuários, outros sistemas e outros agentes, e por conseguinte, dos **protocolos de comunicação** (e ontologias) a serem utilizados;
- Definição da **“topologia”** / organização do sistema multiagente;
- Paradigma de **representação do conhecimento** e modelagem de dados;
- **Modelos** de coordenação, cooperação e de resolução de conflitos;
- **Plataforma de implementação** (linguagem, hardware e sistema operacional).

Arquitetura de Referência: Ilustração-exemplo



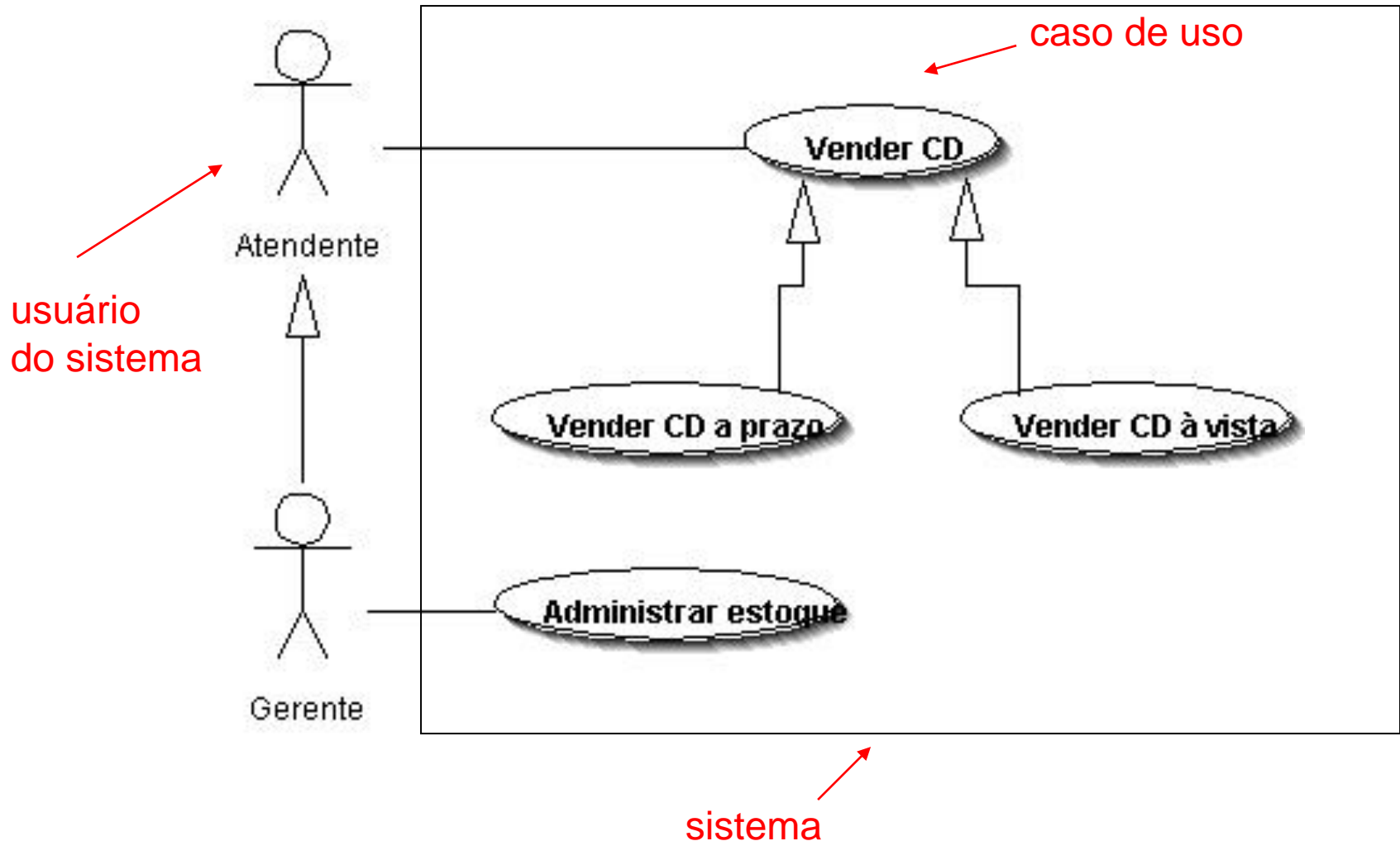
Arquitetura de um Agente



AUML: Agent UML

- Modificações propostas na UML padrão:
 - Suporte para expressar linhas de interação concorrentes (ex: broadcast) visando permitir a modelagem de protocolos de agentes
 - Uma noção de *papel* que estende a que é fornecida na UML, permitindo a um agente desempenhar vários papéis
- Grupo Agent UML: <http://www.auml.org>

Diagrama de use case de UML



Exemplos de Diagramas de Casos de

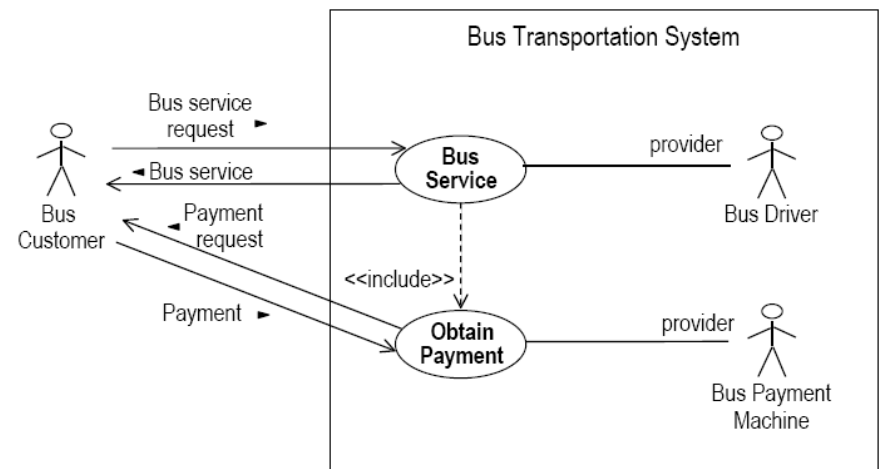
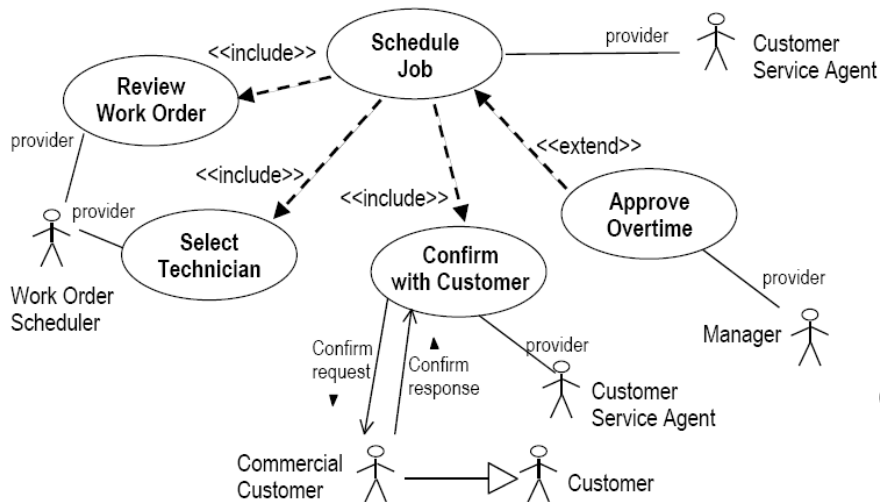
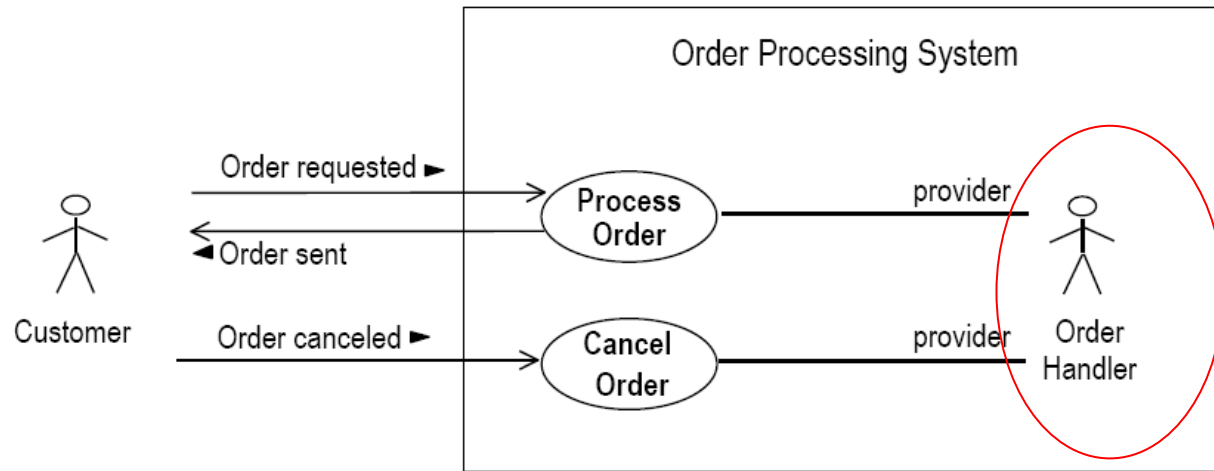


Diagrama de Classes de Agente

- Descreve os tipos de agentes do sistema e seus relacionamentos estáticos

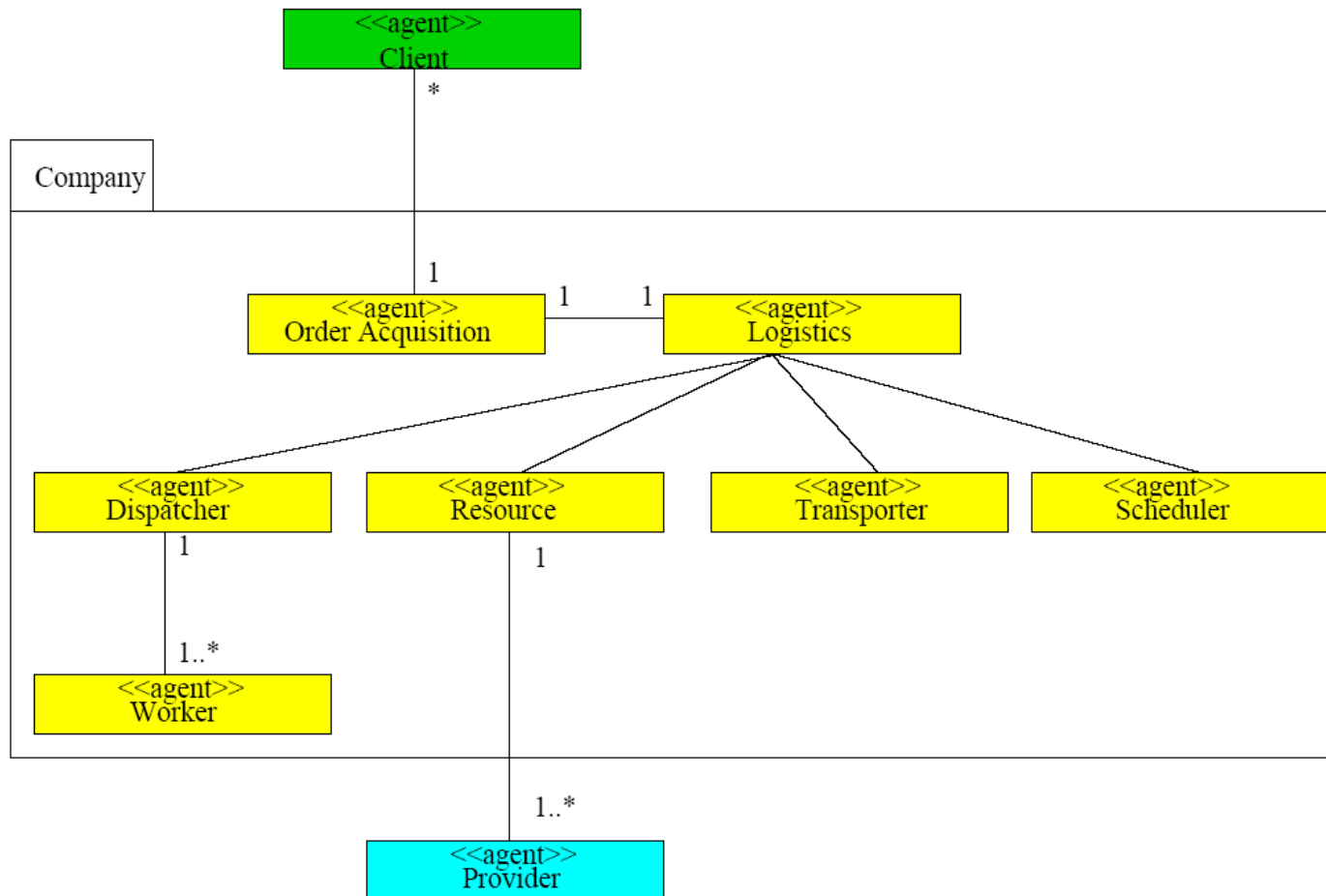


Diagrama de classe em UML

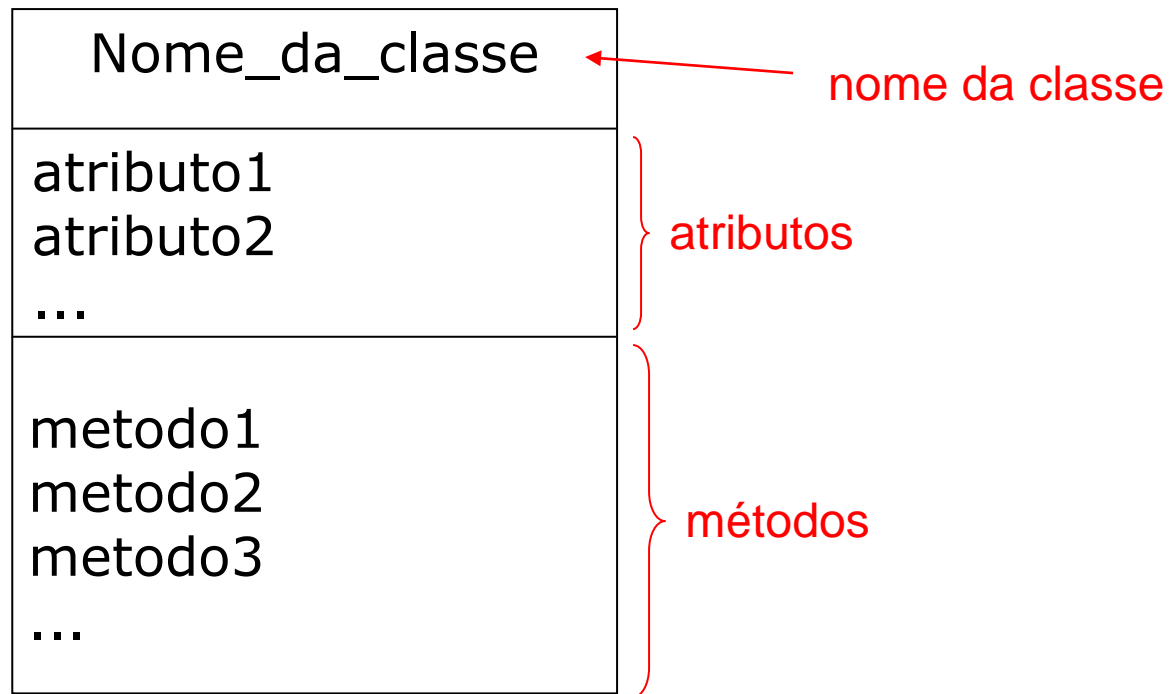


Diagrama de classes de AUML

agente

communication act

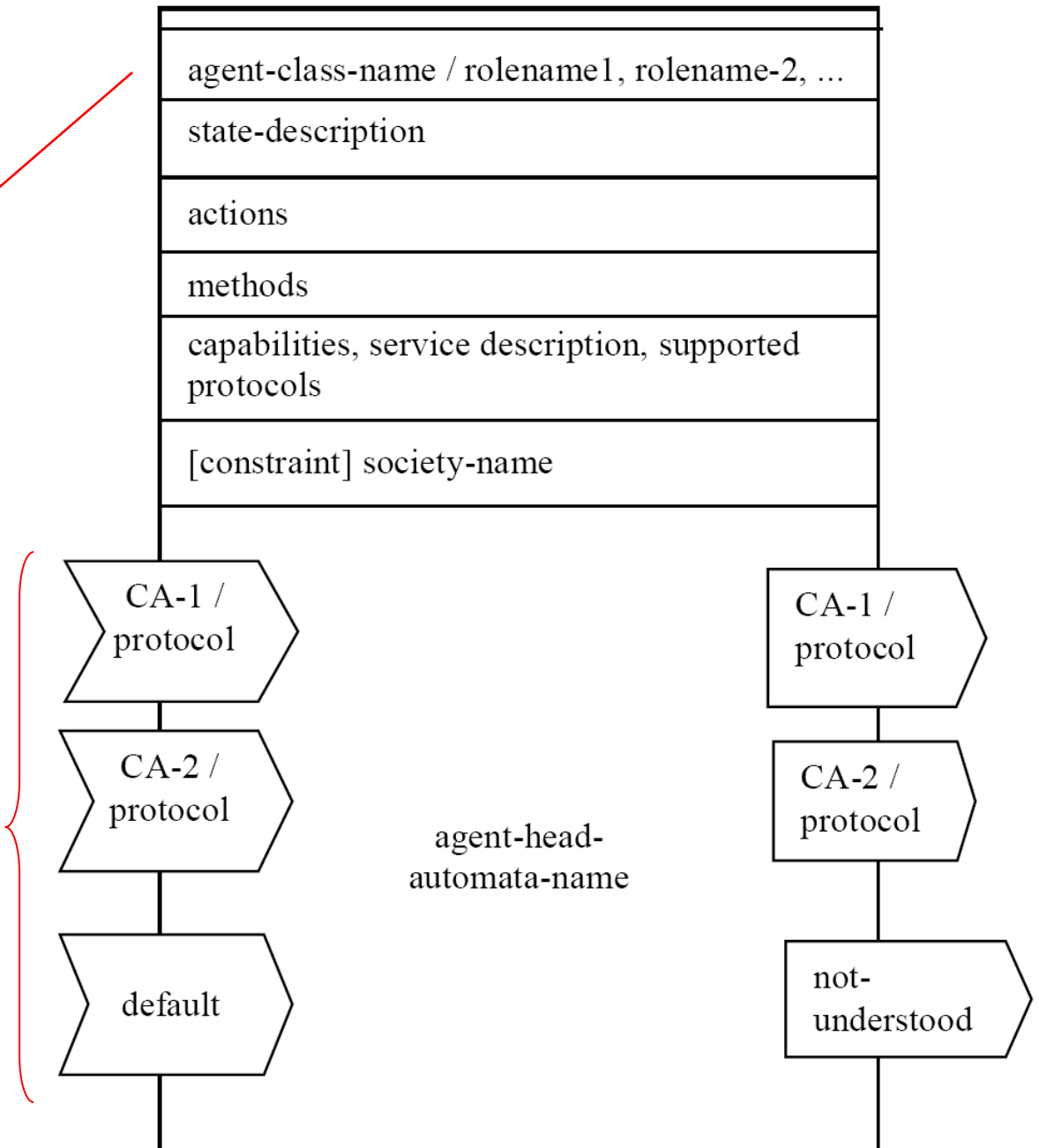


Diagrama de classes de AUML

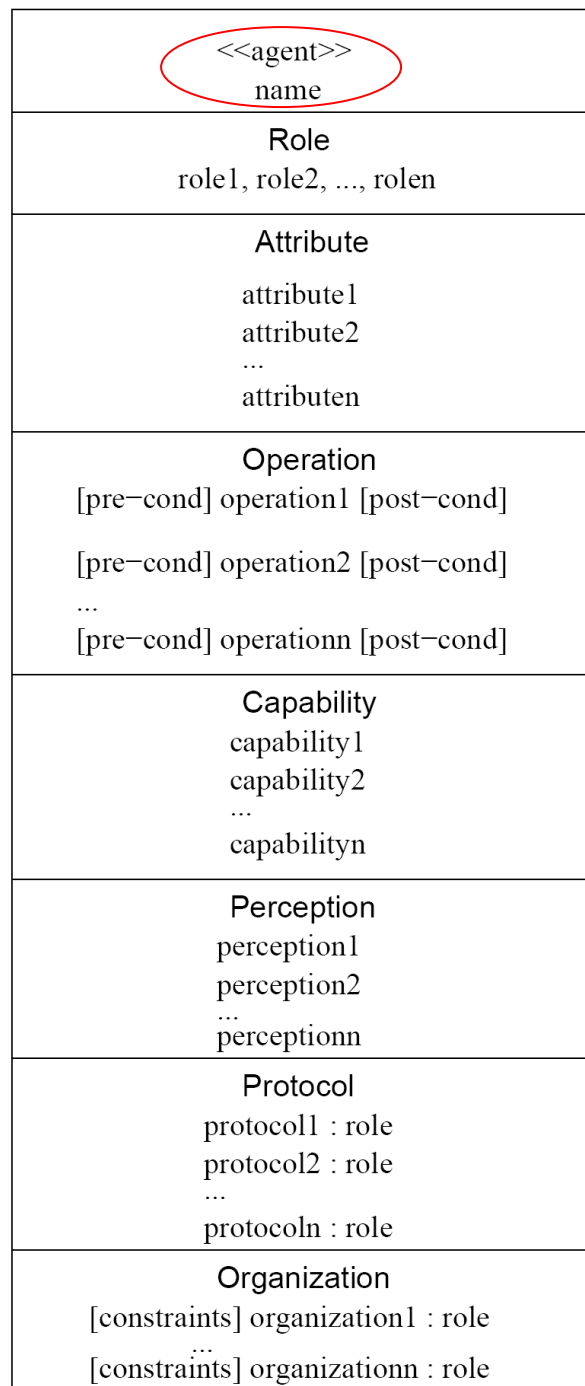
visão de agente

visão de organização

visão de ambiente

visão de interação

visão de organização



visão da organização

service1 [conditions]

service2 [conditions]

servicen [conditions]

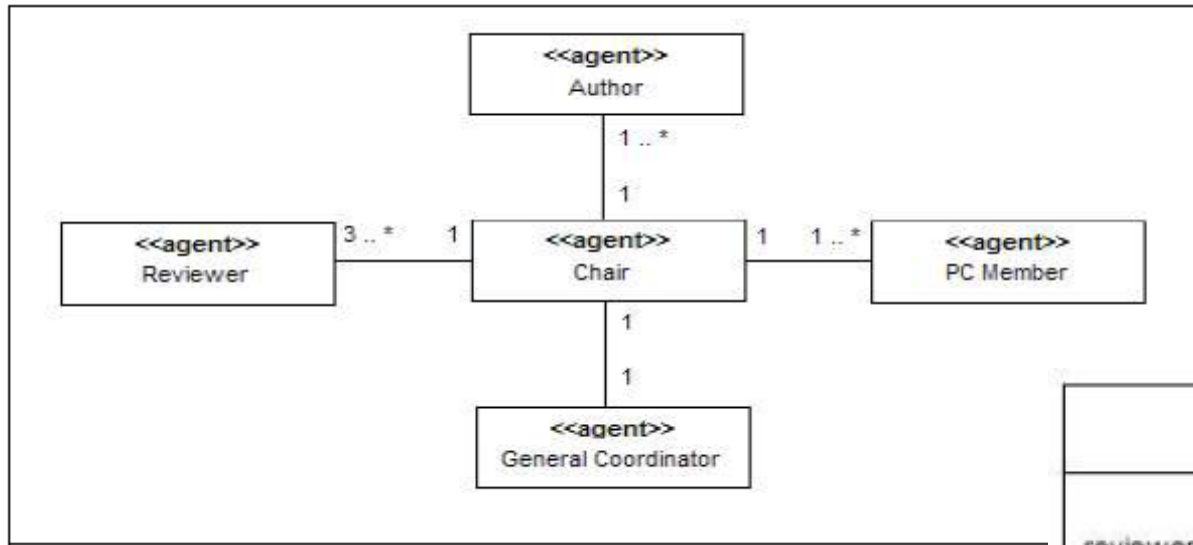
Representação de
belief, desire e
intentionation como
atributos

bel1: Believe

variable = "b"
value = 10

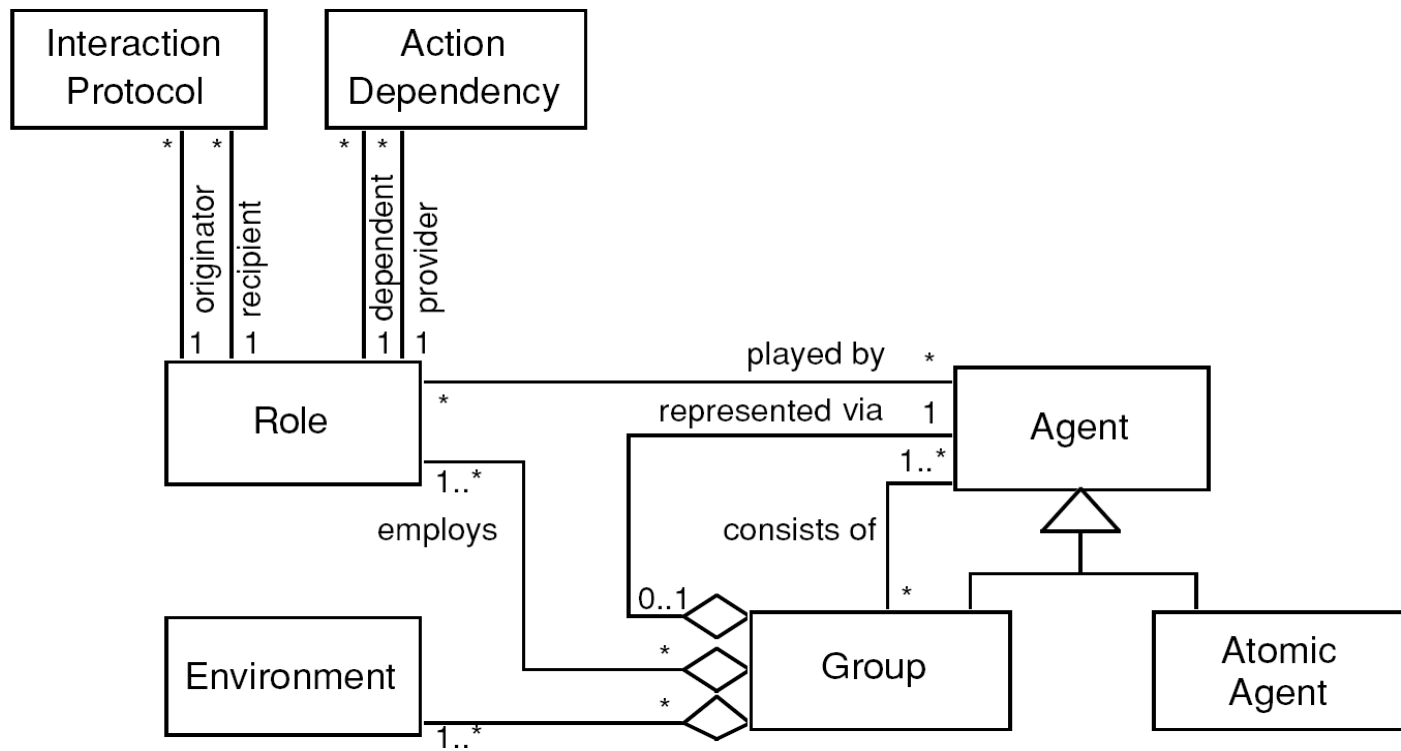
visão de agente

Exemplo de Diagrama de Classe de Agente e Detalhamento da Classe Agente



<<agent>> Reviewer
Role reviewer
Statechart reviewer_behaviour
Attribute name: string institution: string areas_of_interest: string[]
Operation [Basic proposal received] boolean hasFreeTime(Date deadline)
Protocol Reviewing papers: reviewer

Representando Organizações



Organizações, papéis e agentes

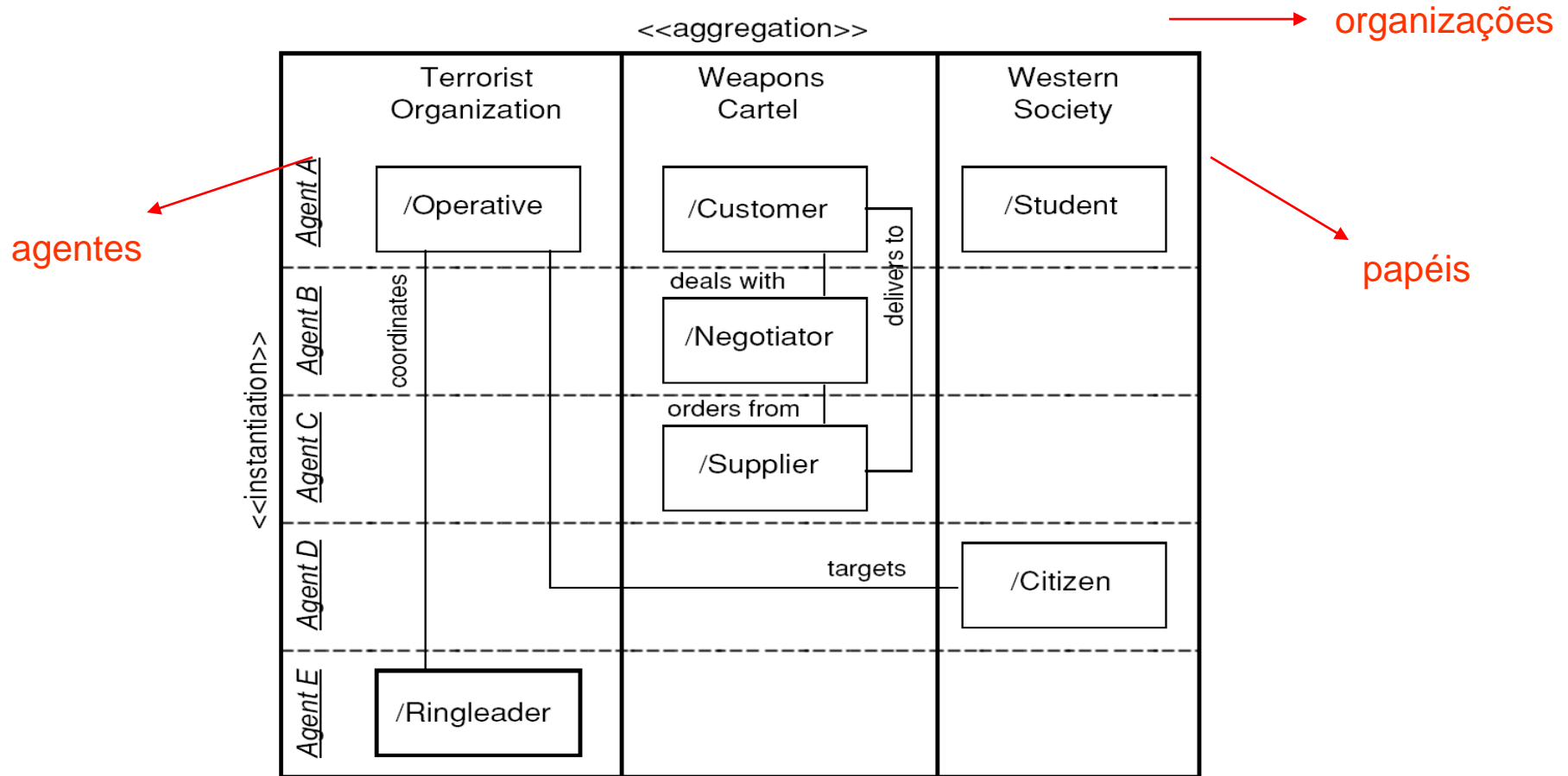
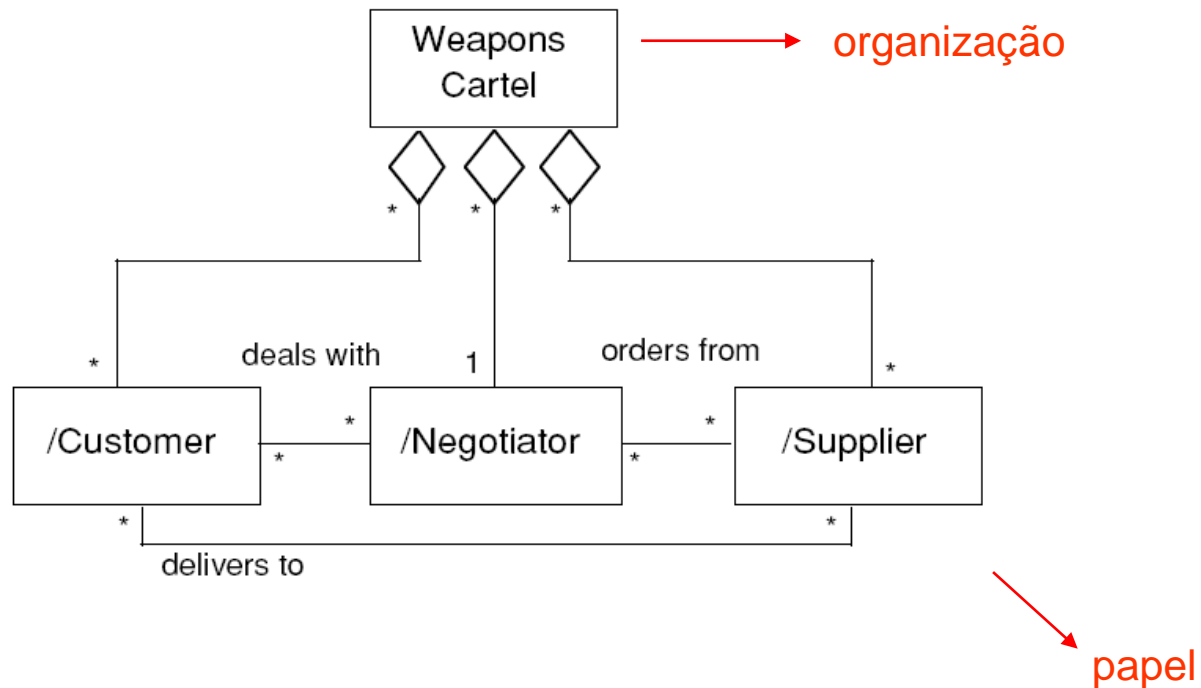


Diagrama de classe



Organizações UML 2.0 v. extendida

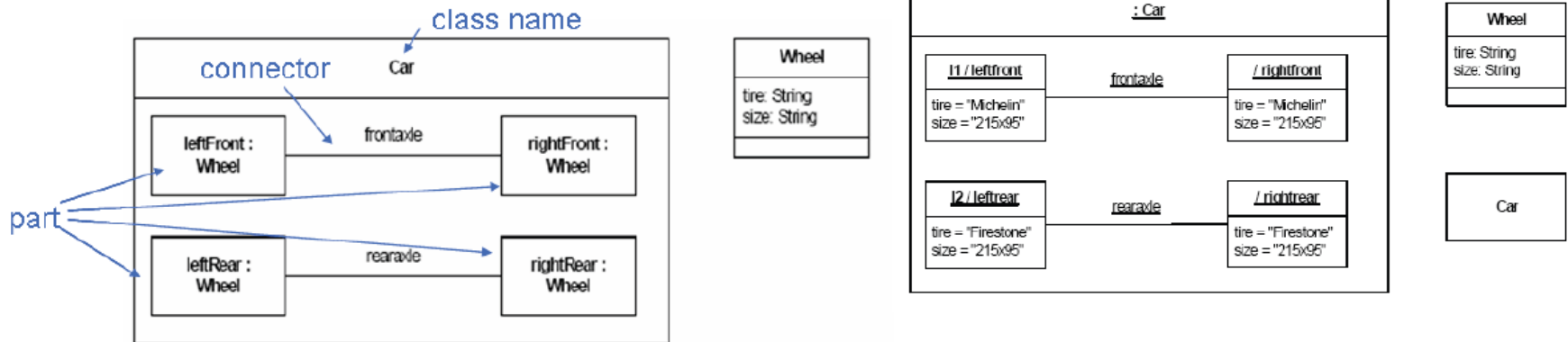


Figure 3 Composite Structure Diagram and its Instantiation

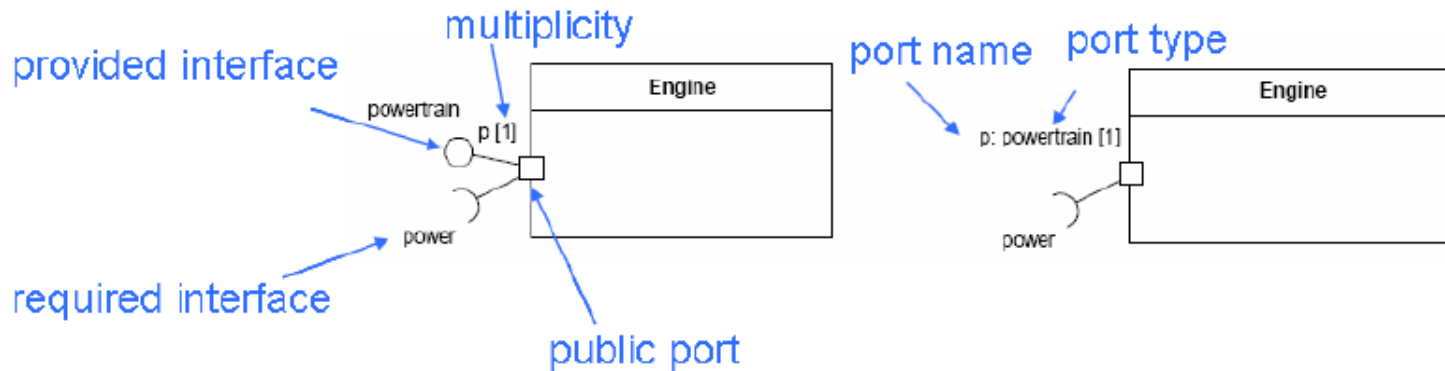


Figure 4 Composite Structure Diagram – Ports

Bibliografia

- Gatti, M., von Staa, A., Lucena, C.; ***AUML-BP: A Basic Agent Oriented Software Development Process Model Using AUML***; Monografias em Ciência da Computação, Departamento de Informática, PUC-Rio, No. 21/07, 25 pg., 2007.
- B. Bauer and J. Odell, **UML 2.0 and agents: how to build agent-based systems with the new UML standard**, Engineering Applications of Artificial Intelligence, Volume 18, Issue 2, Agent-oriented Software Development, March 2005, Pages 141-157.
- Michael Winikoff. **Towards Making Agent UML Practical: A Textual Notation and a Tool**. First international workshop on Integration of Software Engineering and Agent Technology (ISEAT 2005). September 2005, Melbourne, Australia.