

# **Aplicação de Organização de Sistemas Multiagentes em Futebol de Robôs**

Jomi Fred Hübner (FURB)  
Jaime Simão Sichman (USP)

Escola Regional de Informática, Lages, maio de 2003

# Roteiro

- Sistemas Multiagentes
- Organização de SMA
- Futebol de Robôs
- Organização de times de futebol

# **Sistemas Multiagentes**

# Motivações para SMA: Novas Fontes de Inspiração

- Fontes de “inspiração” para a Computação
  - ★ Filosofia: Orientação a Objetos
  - ★ Psicologia: Inteligência Artificial
  - ★ Lógica: Sistemas de raciocínio
  - ★ Biologia: Redes Neurais
- Sociologia e Etologia: ?

# Motivações para SMA: Coletividade

- IA × SMA
  - ★ IA: “fazer **uma** entidade artificial apresentar propriedades inteligentes”. (psicologia + engenharia)
  - ★ Abordagens
    - \* Simbolista (mente)
    - \* Conexionista (cérebro)
- “Inteligência” como processo **emergente**
  - ★ Formigueiro
  - ★ Cérebro
  - ★ Cidade

**O todo é mais que a soma das partes**
- O que tem no “todo” que não tem nas partes?

# Motivações para SMA: Desenvolvimento de Sistemas

Ciclo “clássico” de desenvolvimento de sistemas (distribuído):

- Identificação de requisitos (problema)
- Análise (do problema)
- Projeto (de solução para o problema)
- Implementação
- Teste

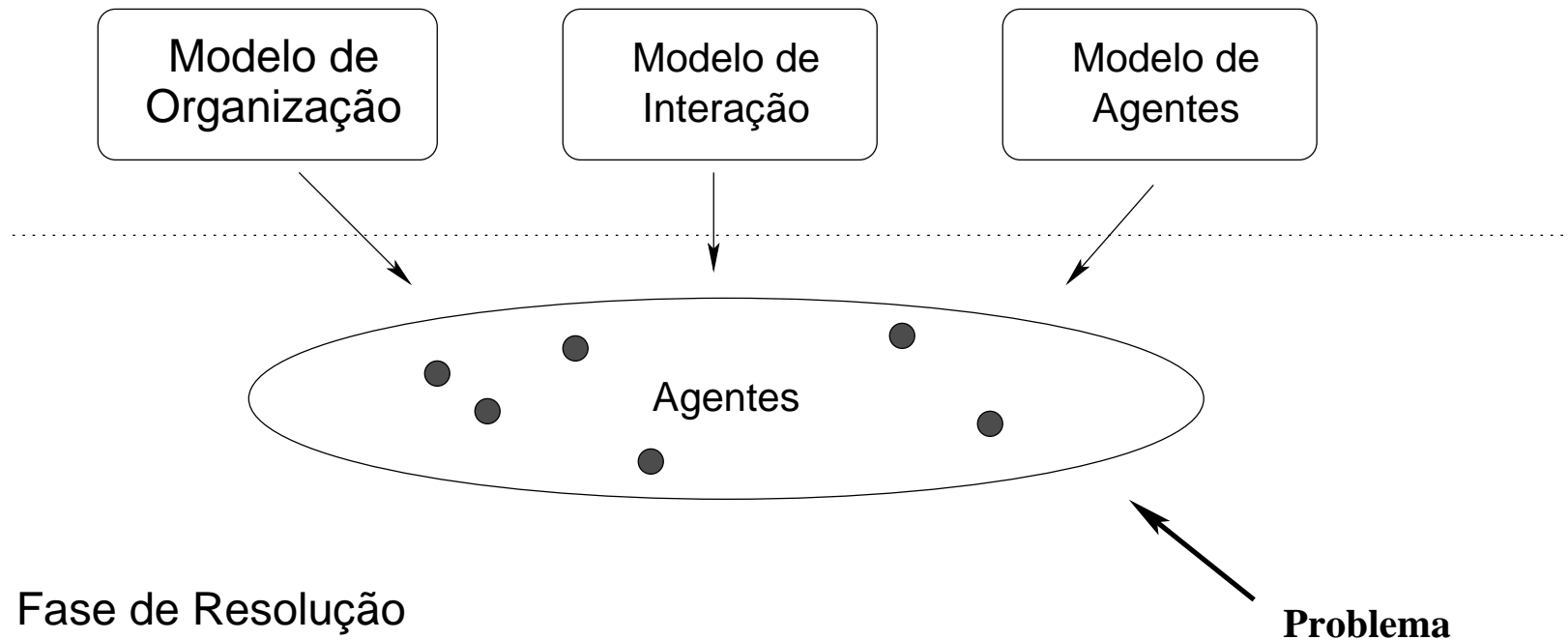
Se o problema muda, a solução tem que mudar! (forte acoplamento entre os módulos)

Propostas de solução:

- objetos, componentes .... **agentes**

# Ciclo (**ideal**) proposto pela área de SMA

Fase de Concepção



# Características dos SMA

- os agentes são concebidos independentemente de um problema particular
- a interação entre os agentes não é projetada anteriormente, busca-se definir protocolos que possam ser utilizados em situações genéricas
- a decomposição de tarefas para solucionar um dado problema pode ser feita pelos próprios agentes
- não existe um controle centralizado da resolução do problema



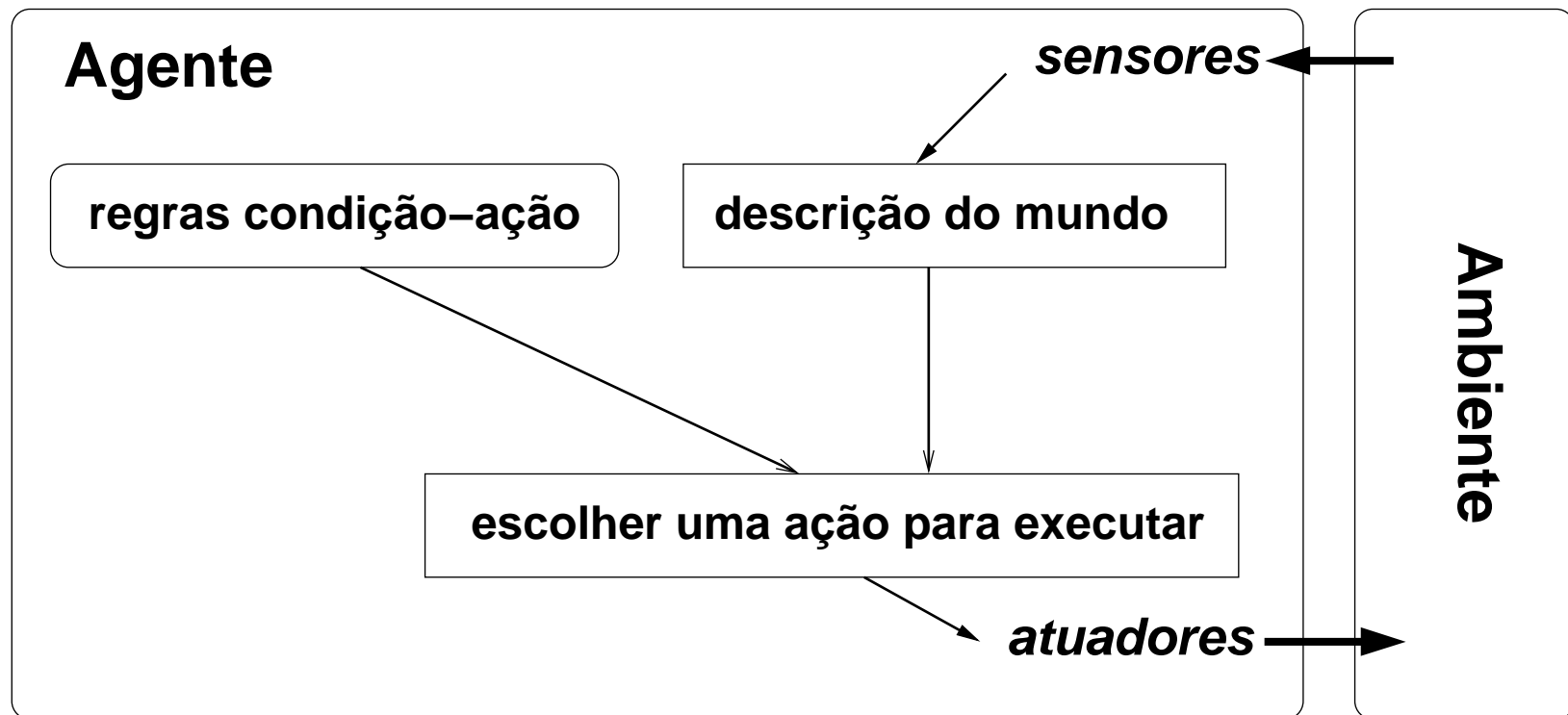
# Vantagens dos SMA

- Viabilizam sistemas **adaptativos** e **evolutivos**
- É uma **metáfora natural** para a modelagem de sistemas complexos e distribuídos
- Toma proveito de ambientes **heterogêneos** e **distribuídos**
- Permite conceber **sistemas abertos**

# Agentes

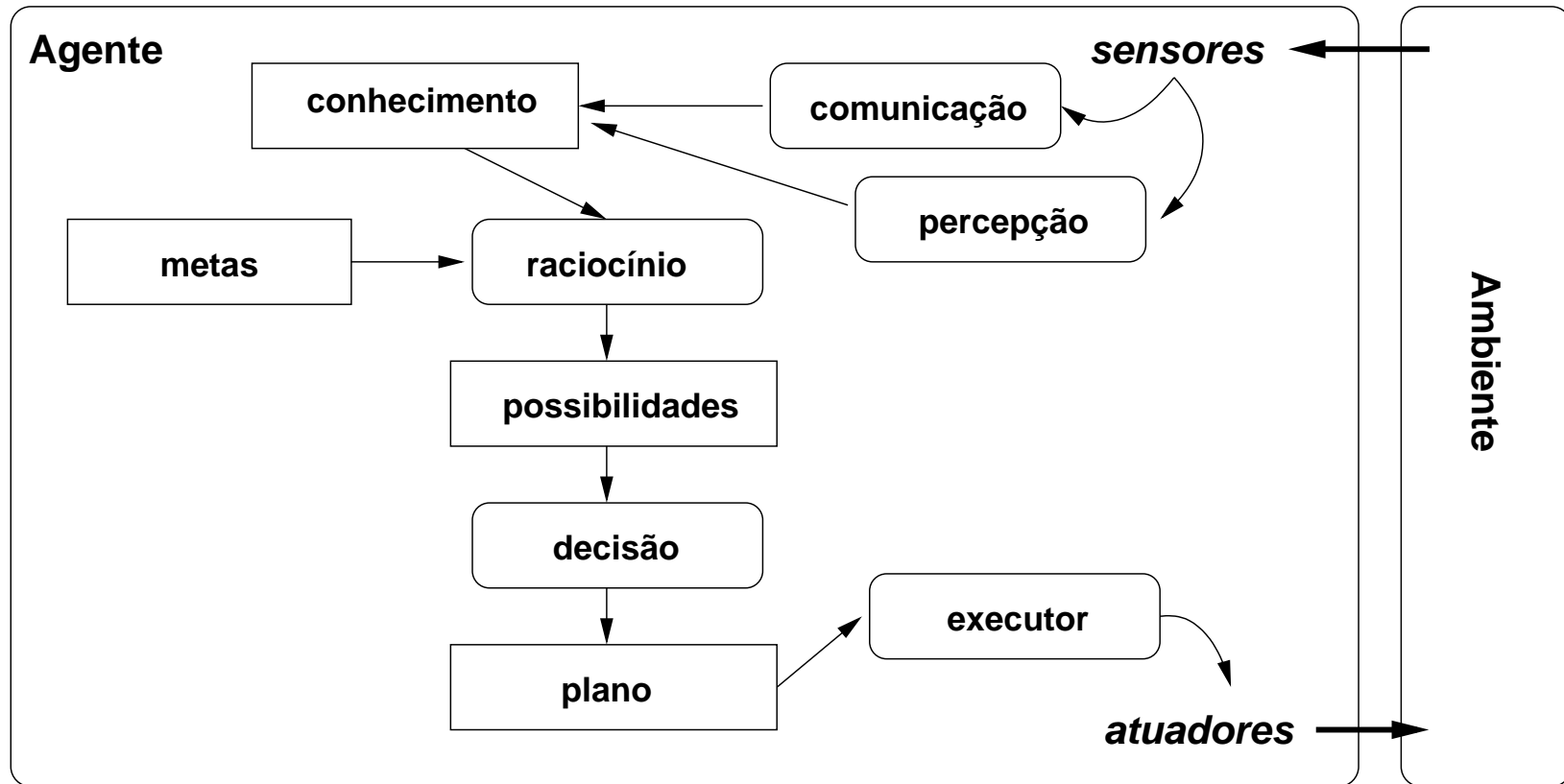
- Teorias (agentes reativos × cognitivos)
- Arquiteturas
- Linguagens
  - ★ extensões de outras linguagens (Java, TCL)
  - ★ novas linguagens (Agent-0, AgentSpeak)

# Exemplo de arquitetura **reativa**



(Russel, 1995)

# Exemplo de arquitetura **cognitiva**



(Demazeau, 1990)

# Interação

- Teorias (atos de fala)
- Arquiteturas (KQML, FIPA ACL)
- Linguagens (FIPA-OS, JADE, SACI)

# Exemplo de comunicação com **KQML**

KQML (Knowledge Query and Manipulation Language) é uma especificação de linguagem de comunicação entre agentes (Finin, 1997).

- Qualquer linguagem pode ser usada para escrever o conteúdo da mensagem
- A informação necessária para a interpretação da mensagem está na própria mensagem
- Os agentes podem ignorar o mecanismos de transporte (TCP/IP, RMI, IIOP, ...)
- O formato é simples, fácil de ler a verificar

## Exemplo de mensagem KQML:

*(ask-one*

:language      *SQL*  
:ontology      *SOStore*

:sender          *jomi*  
:receiver        *ricardo*  
:reply-with      *id1*

:content          *“select price from  
stocktable where  
ent = Conectiva”*

} nível de mensagem  
  
}  
  
} nível de comunicação  
  
}  
  
} nível de conteúdo

```
(tell
  :language      prolog
  :ontology      SOStore
  :receiver      jomi
  :sender        ricardo
  :in-reply-to   id1
  :reply-with    id45
  :content       "[price(10.0)]")
```

Embora a KQML tenha um conjunto pré-definido de performativas e palavras-chave, este conjunto não é nem mínimo nem fechado. Contudo, os agentes que usam uma das performativas reservadas devem usá-las da forma padrão.

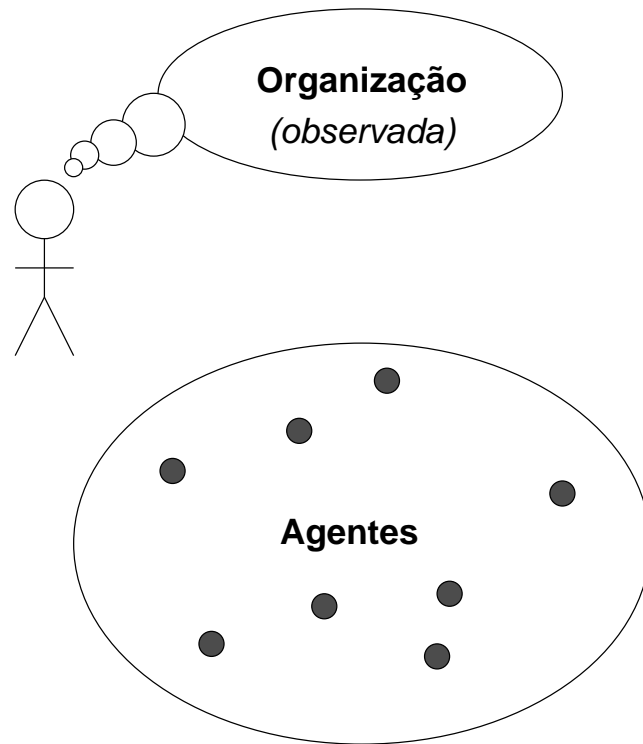


# Organização em SMA

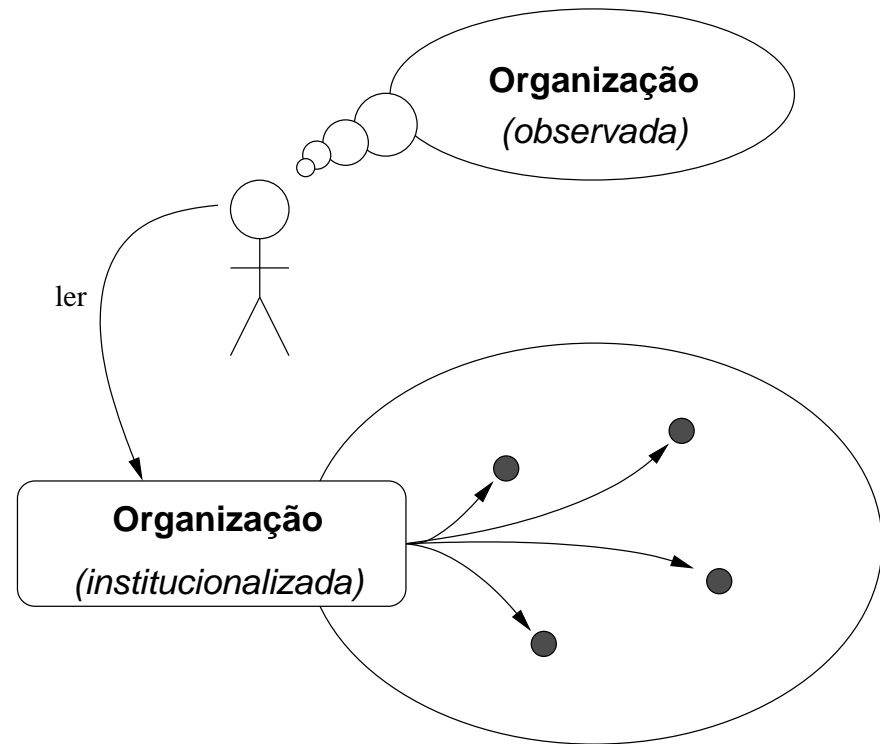
# Noção **intuitiva** de organização

- Exemplos:
  - ★ Uma mesa de trabalho
  - ★ Um formigueiro
  - ★ Um célula
  - ★ Um time de futebol
- Perguntas:
  - ★ O que é exatamente organização?
  - ★ Quais os tipos de organização?
  - ★ Por que organizar?

# Duas formas de ver organização



(a) Visão centrada nos agentes



(b) Visão centrada na organização

(Lemaître e Excellence, 1998)

# Por que organizar?

- Se os agentes são autônomos (autonomia de funcionamento, objetivos, etc.), como o sistema vai fazer atingir um objetivo global?
- A autonomia precisa ser “limitada”.
- Exemplo “todos nós somos autônomos, mas quando assumimos o **papel** de aluno, já não podemos mais fazer certas coisas e podemos fazer outras.”.
- Ou seja, na sociedade humana, a noção de papel é muito utilizada para representar direitos e obrigações que, de certa forma, controlam nossa autonomia.

# O que é Organização

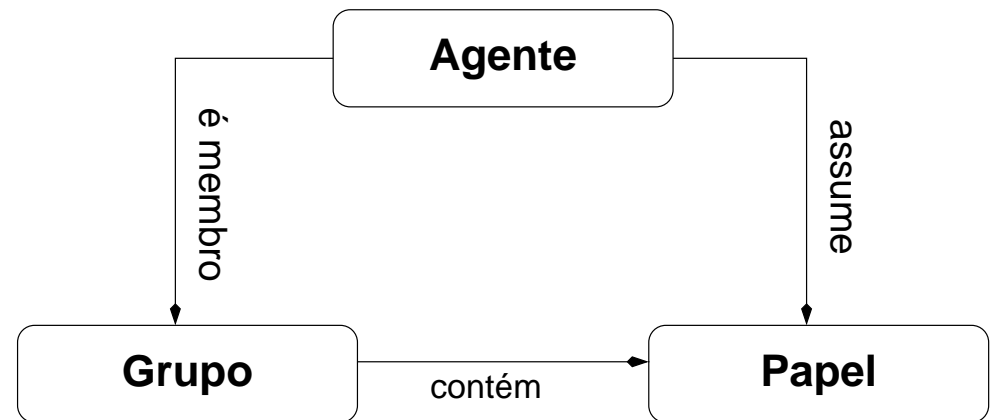
**A organização de um SMA é um conjunto de restrições ao comportamento dos agentes a fim de conduzi-los a uma finalidade comum.**

- Estas restrições podem estar explícitas ou não.
- No caso onde se deseja explicitar a organização (observada ou institucionalizada), como descrevê-la?
- Que noções utilizar (grupos, papéis, tarefas, missões, autoridade, etc.)?
- Qual o significado destas noções?

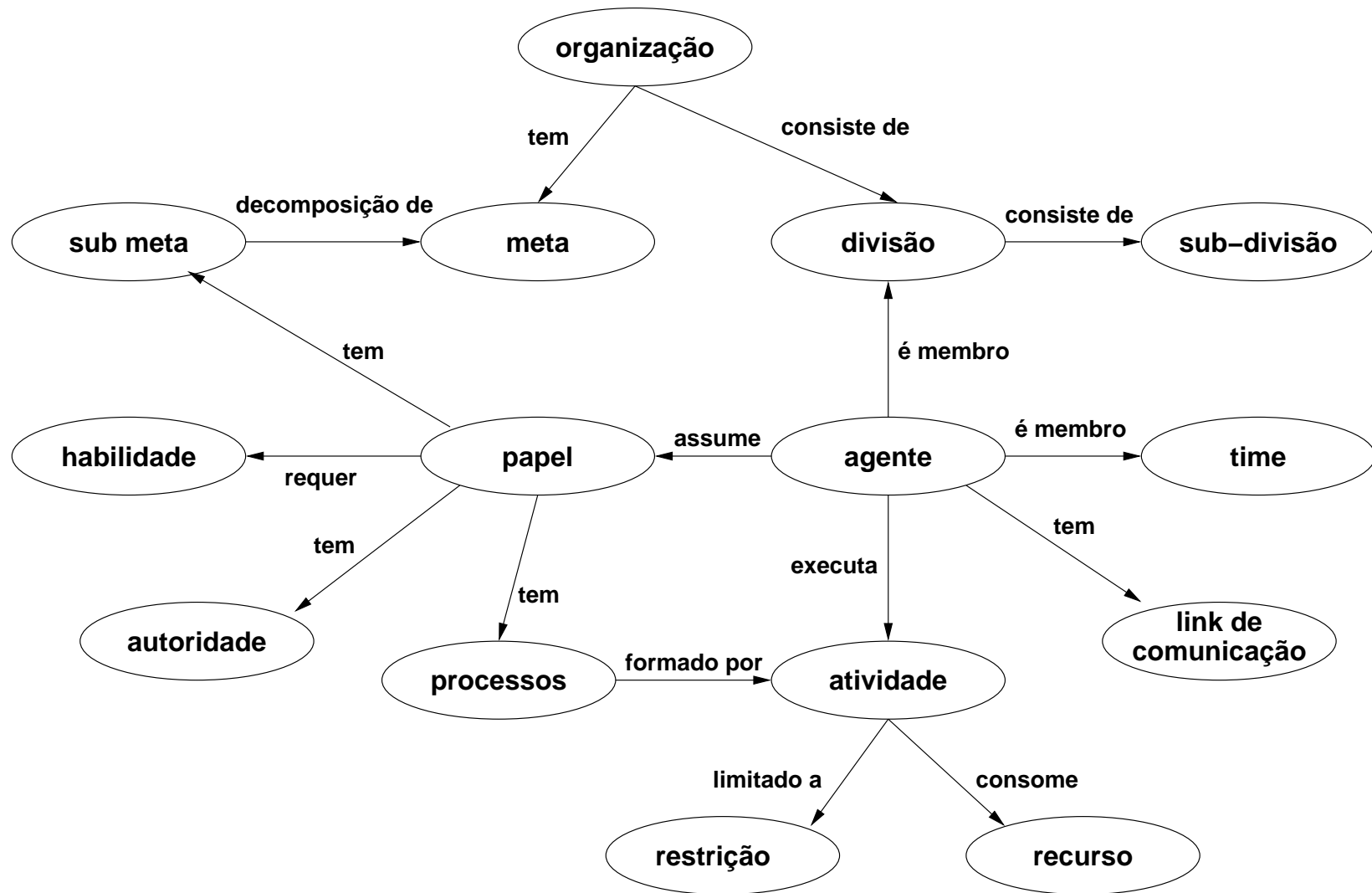
# O modelo AALAADIN

- Um organização é um conjunto de grupos e agentes com papéis nestes grupos.
- Um grupo tem um conjunto de papéis necessários para seu funcionamento.
- Papel é um conjunto de **funções** que os agentes assumem ao entrar em um grupo.

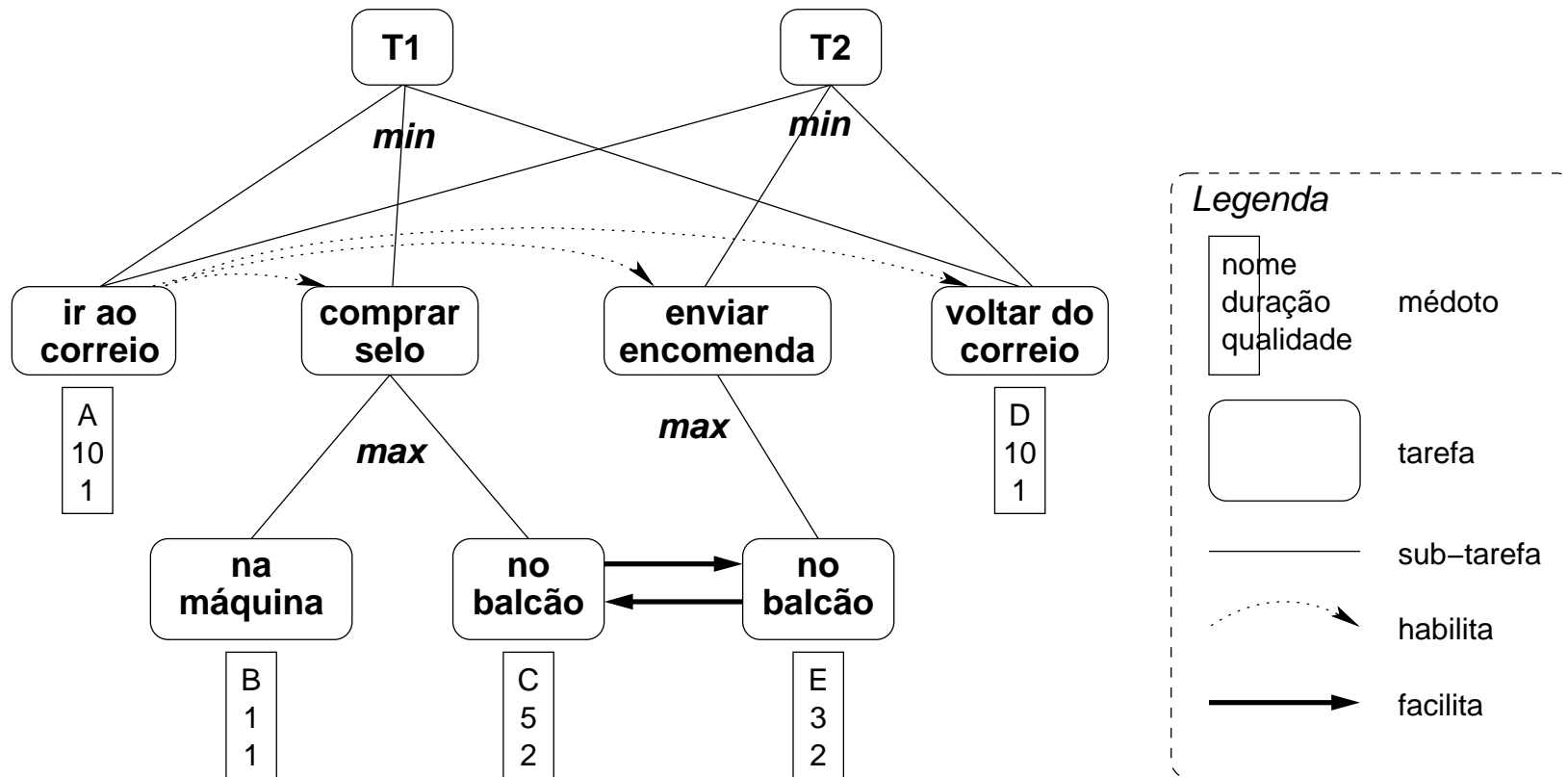
- Não importa a arquitetura dos agentes.
- A organização é **instanciada** pelos agentes.



# O modelo TOVE



# O modelo TÆMS





# O modelo $\mathcal{MOISE}^+$

O  $\mathcal{MOISE}^+$  é um modelo organizacional, baseado no MOISE (Hannoun, 2000), voltado para a reorganização

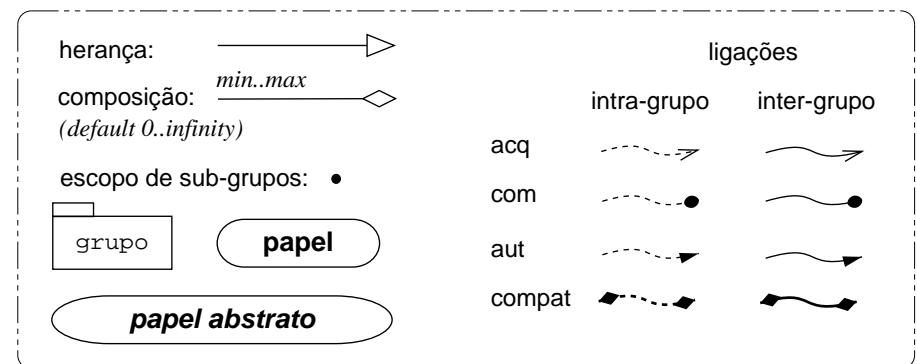
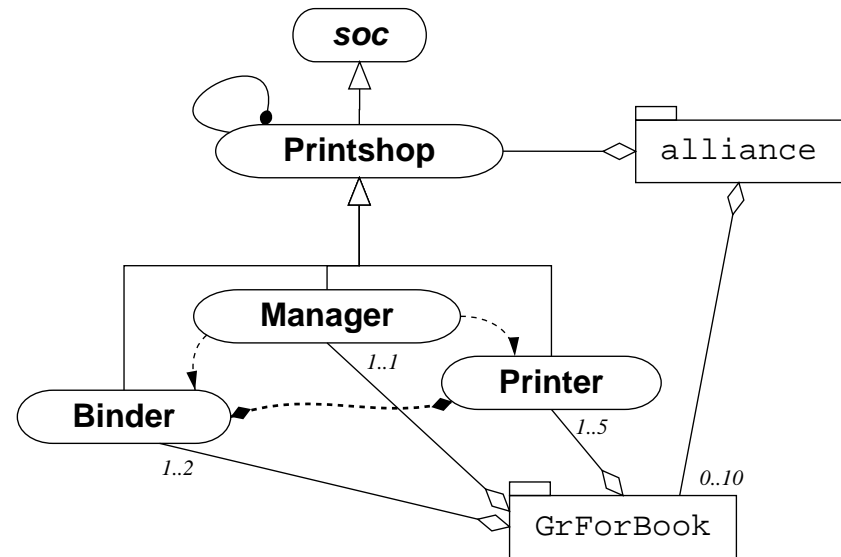
A organização é descrita/especificada em três dimensões:

- i)* **estrutural**: papéis, grupos, relações, ...
- ii)* **funcional**: tarefas globais, missões, ...
- iii)* **deôntica**: obrigações e permissões dos agentes

# Dimensão estrutural

A estrutura define o conjunto de comportamentos **possíveis** pela organização.

- **papel** tem um caráter coletivo
- **ligações** e compatibilidades: herança e escopo
- hierarquia de papéis
- **grupos** e sub-grupos
- cardinalidade



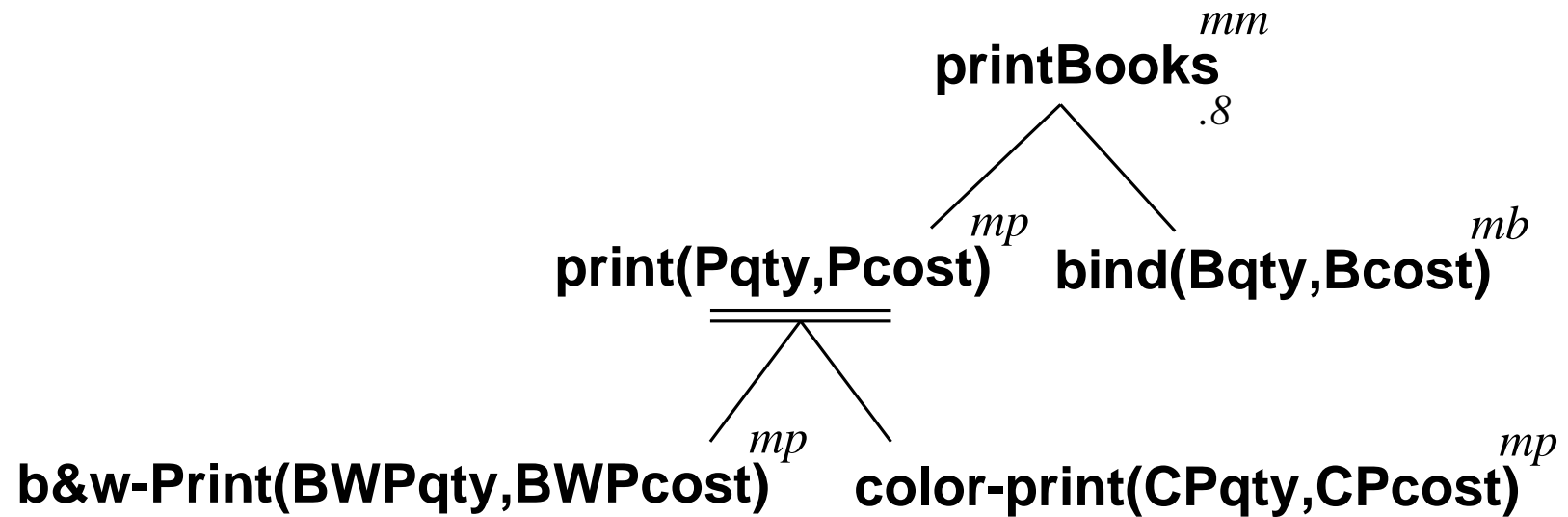
# Dimensão funcional

A dimensão funcional define o conjunto de **esquemas** que um SMA utiliza para alcançar suas metas.

Esquemas sociais = (planos + **missões**)

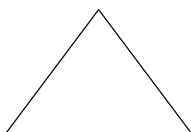
- a ligação dos agentes com os planos é feita por meio de missões,
- pode-se desenvolver os esquemas sem referir-se à estrutura (papéis)
- pode-se determinar preferências entre missões
- as missões também possuem cardinalidade

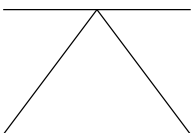
# Esquema social

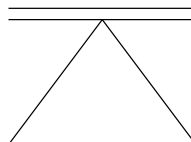


## Legenda

*missões*  
meta  
*taxa de sucesso*

  
sequência

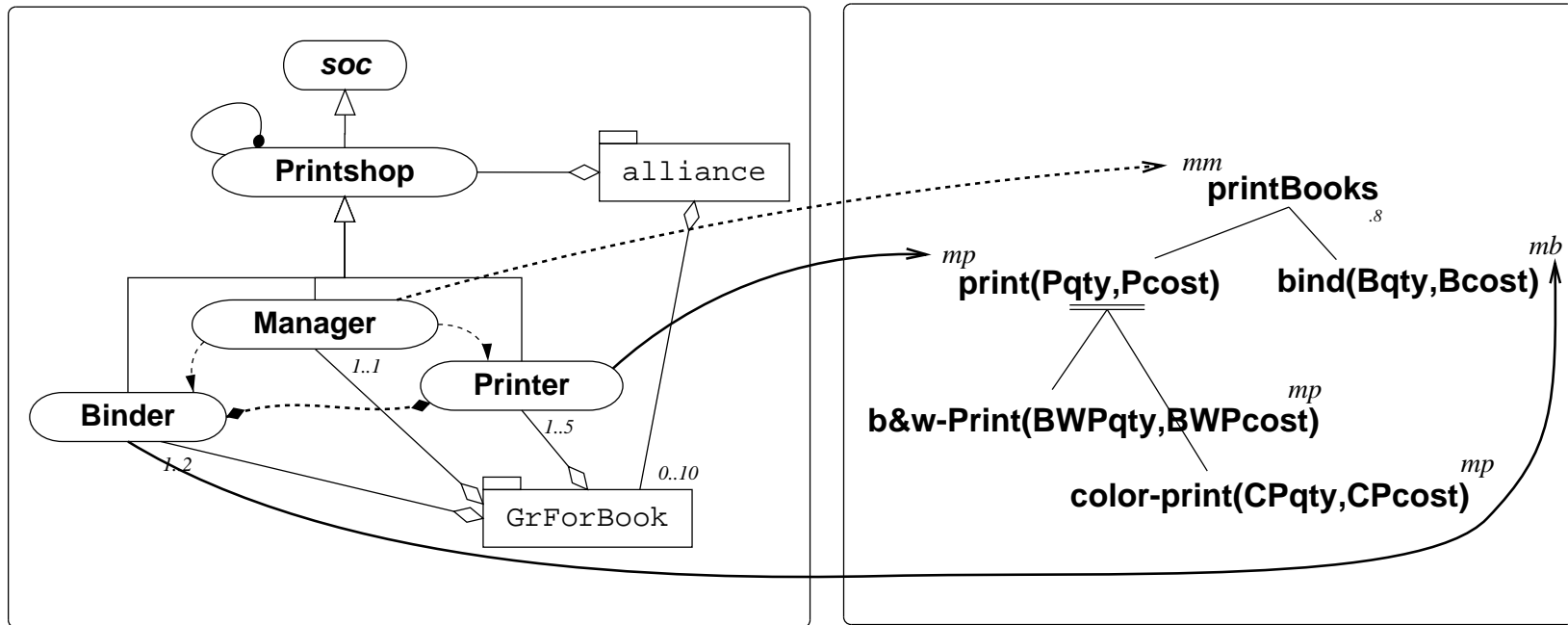
  
escolha

  
paralelismo

# Dimensão deôntica

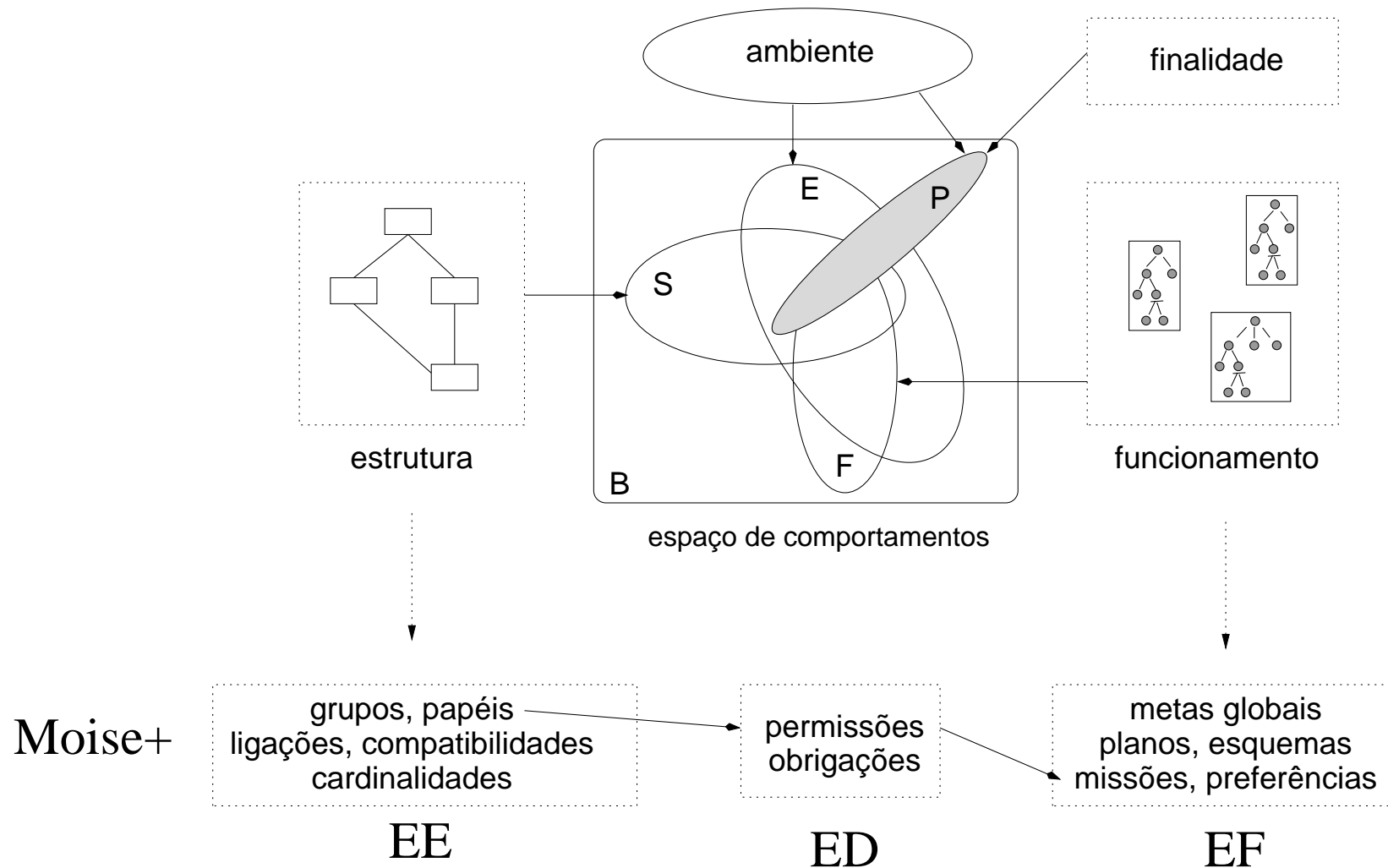
Esta relação, entre estrutura e funcionamento, é estabelecida no nível individual: papel → missão

- papel tem um caráter normativo
- tipo: permissão ou obrigação
- restrições temporais
- são herdadas

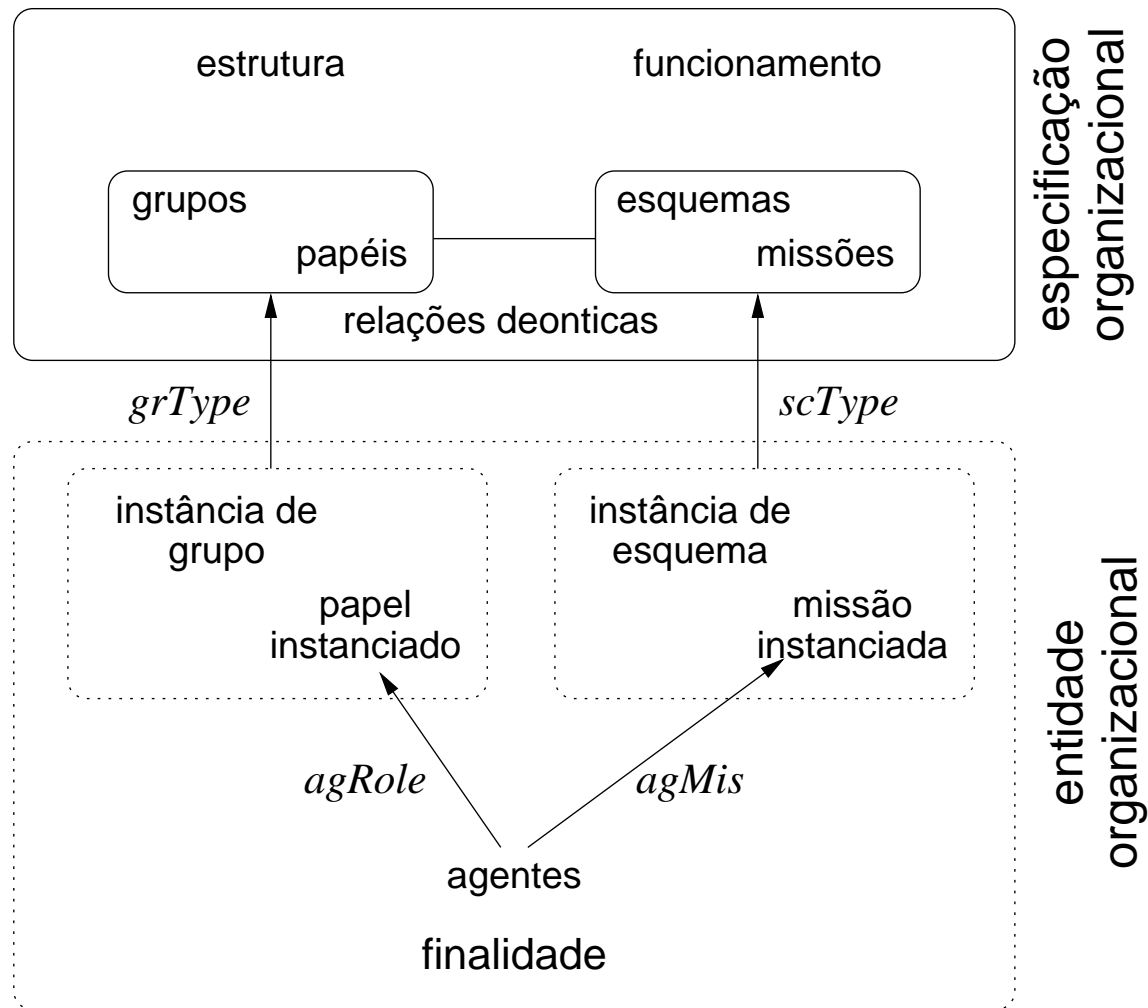


papel	relação deôntica	missão	restrições de tempo
<i>Manager</i>	<i>per</i>	<i>mm</i>	neste mês
<i>Printer</i>	<i>obl</i>	<i>mp</i>	<i>Ptc</i>
<i>Binder</i>	<i>obl</i>	<i>mb</i>	<i>Btc</i>

# Visão geral do $\mathcal{MOISE}^+$

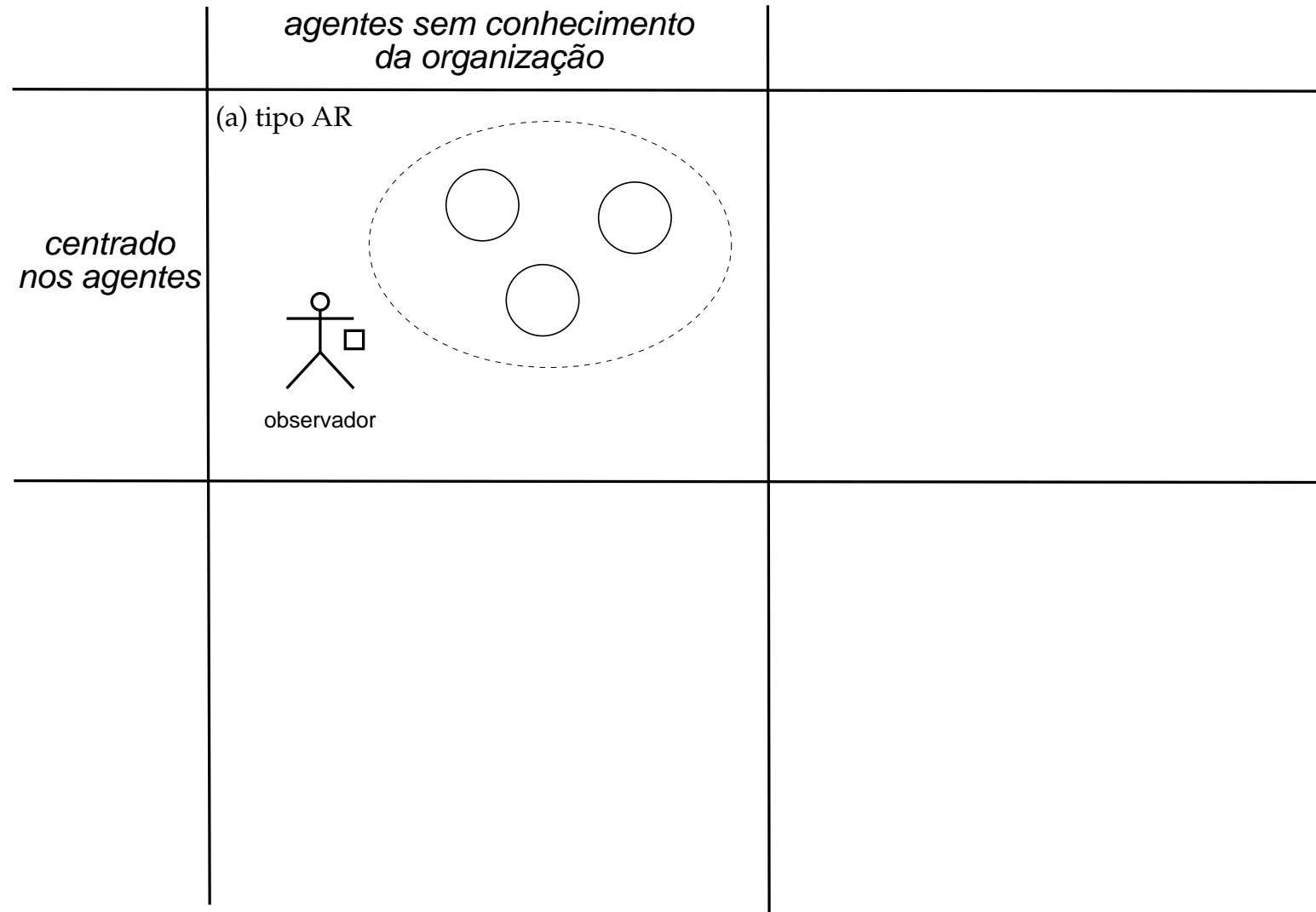


# Entidade Organizacional

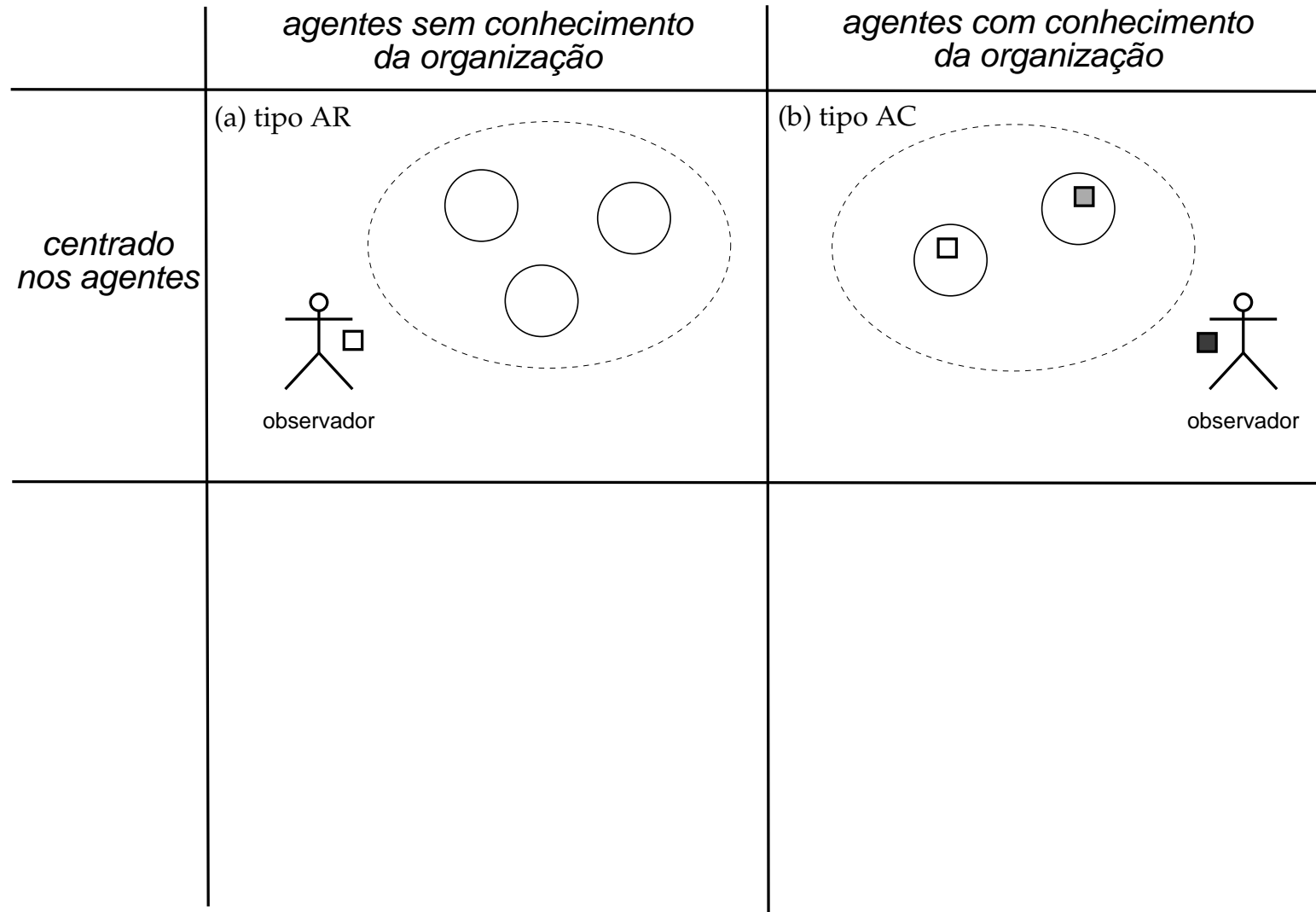




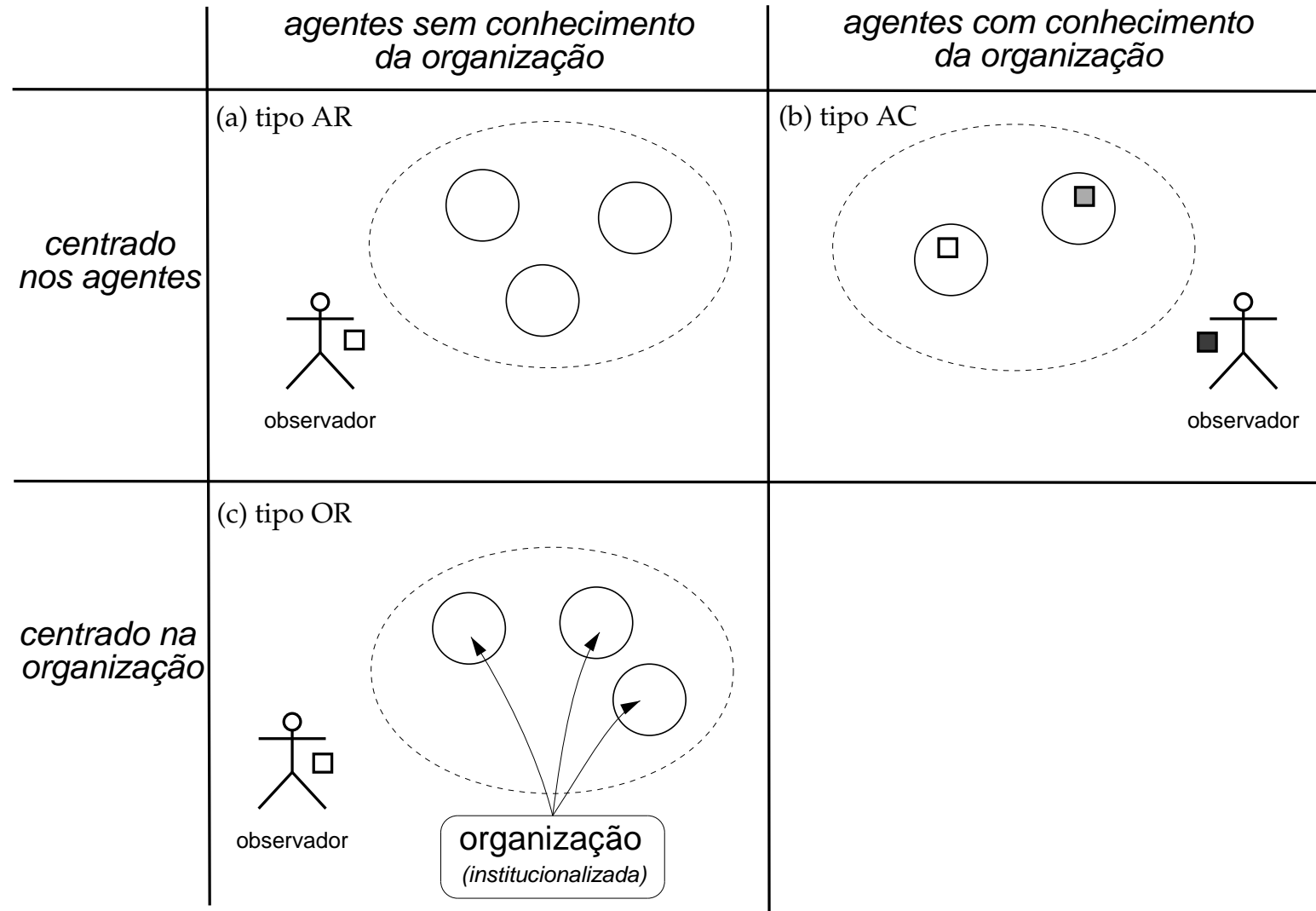
# Tipos de organização



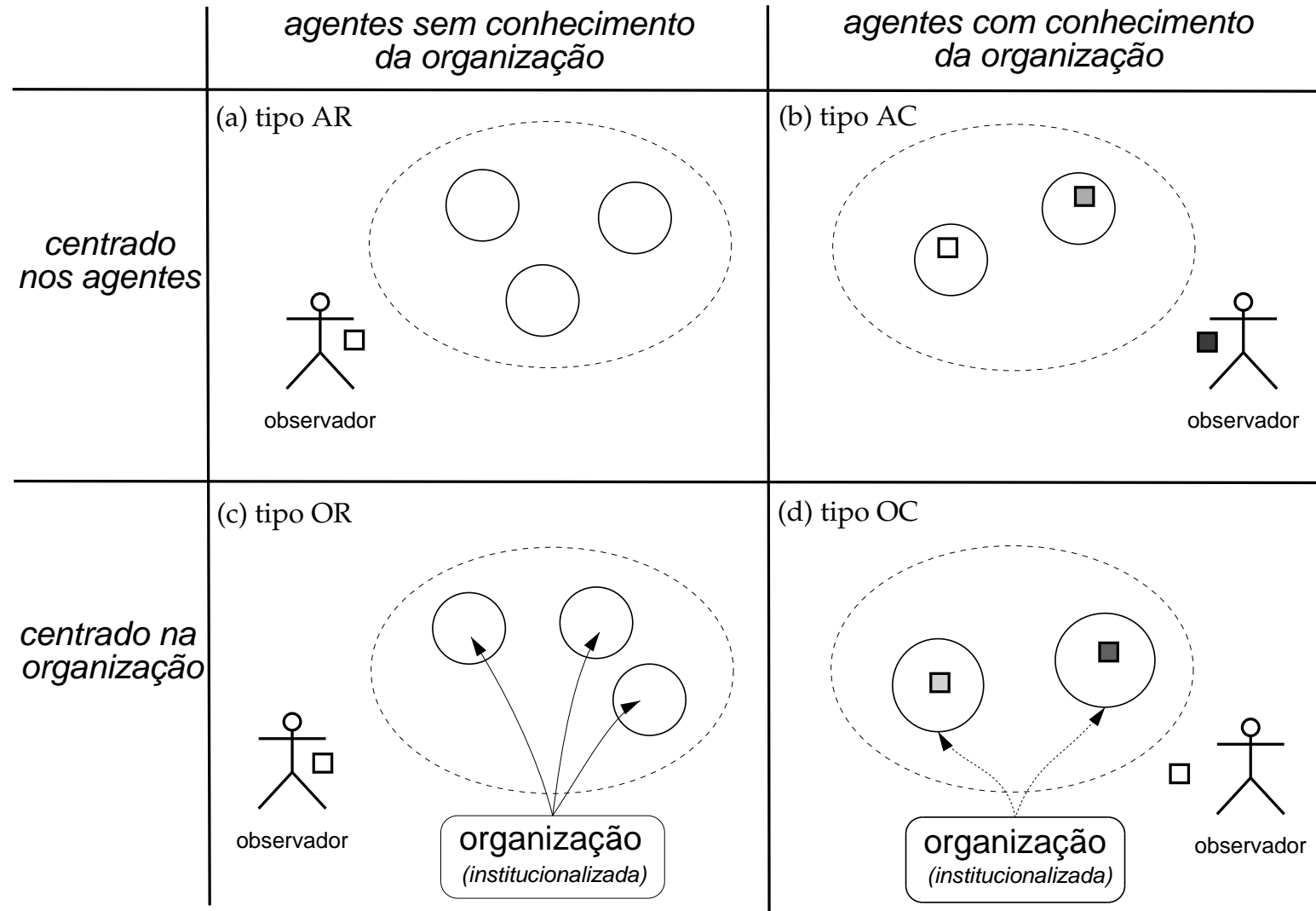
# Tipos de organização



# Tipos de organização



# Tipos de organização



# Futebol do Robôs

# Por que times?

- Pelas mesmas razões que motivaram os **SMA**
  - ★ alta complexidade e
  - ★ limitações temporais, espaciais e funcionais.
- Exemplos:
  - ★ Exploração planetária / subaquática
  - ★ Combate a incêndios em florestas
  - ★ Busca e resgate
  - ★ Remoção de minas terrestres
  - ★ Limpeza de grandes áreas
  - ★ Jogar bola (!)

**Um time é um SMA**

# Vantagens dos times

- **Tempo** de execução: um time cobre uma grande área num tempo menor.
- **Custo**: um único robô exigiria maior capacidade de processamento, maior autonomia, maior robustez para executar uma tarefa complexa.
- **Redundância**: caso um robô falhe ou seja destruído, o restante do time pode continuar a tarefa.

# Dificuldades em times

- Como comunicar adequadamente (como + quanto)?
- Como decompor uma tarefa e alocar as sub?
- Como garantir que o time agirá coerentemente?
- Como os robôs reconhecerão e resolverão conflitos?
- Como dotar um time com capacidade de adaptação?

Mudar seu comportamento em resposta às mudanças dinâmicas do ambiente, às mudanças de metas, às mudanças de capacidades e composição do time, etc.



# Características dos times

- **Granularidade**: tamanho = número de robôs
- **Heterogeneidade**: diversidade sw/hw entre robôs
- **Comunicação**
  - ★ explícita: informações são intencionalmente trocadas
  - ★ implícita: informações são adquiridas por observação das ações dos outros membros ou por rastros não intencionalmente deixados
- Estrutura de **controle**
  - ★ centralizado: robôs se comunicam com um computador central, que distribui atribuições, metas, informações (robôs semi-autônomos, dependentes do computador central para decidir suas ações)

- ★ distribuído: robôs tomam suas próprias decisões e agem de modo independente.

- **Cooperação**

- ★ não-ativa: robôs não compartilham explicitamente uma meta comum (possuem sub-metas individuais)
- ★ acidental (não intencional) é não-ativa. Ex: dois robôs demolindo uma parede
- ★ ativa: quando pelo menos alguns dos membros se reconhecem e trabalham conjuntamente para atingir meta comum explícita, caracterizando intencionalidade na cooperação (pode exigir sensores mais complexos). Ex: passe em futebol.

# Exemplo de configurações de times

- Aplicações que **não** impõem **restrições drásticas no tempo** de execução e que requerem numerosas **repetições**: limpeza (praias, grandes estacionamentos), coleta (missões espaciais), busca e resgate
  - ★ time: homogêneo, alta granularidade, controle descentralizado, cooperação não-ativa, sem comunicação explícita
  - ★ robôs: paradigma reativo (poucos e simples comportamentos), concorrentes e independentes

- Aplicações que com restrições drásticas na **eficiência** de sua execução
  - ★ time: heterogêneo, baixa granularidade, controle distribuído, cooperação ativa, comunicação explícita
  - ★ robôs: diferentes capacidades
  - ★ pontos importantes de projeto: mapeamento apropriado das sub aos robôs (em função das suas capacidades);

# RoboCup

“By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.”

- Um novo desafio para a IA!

# Categoria small size



# Categoria middle size

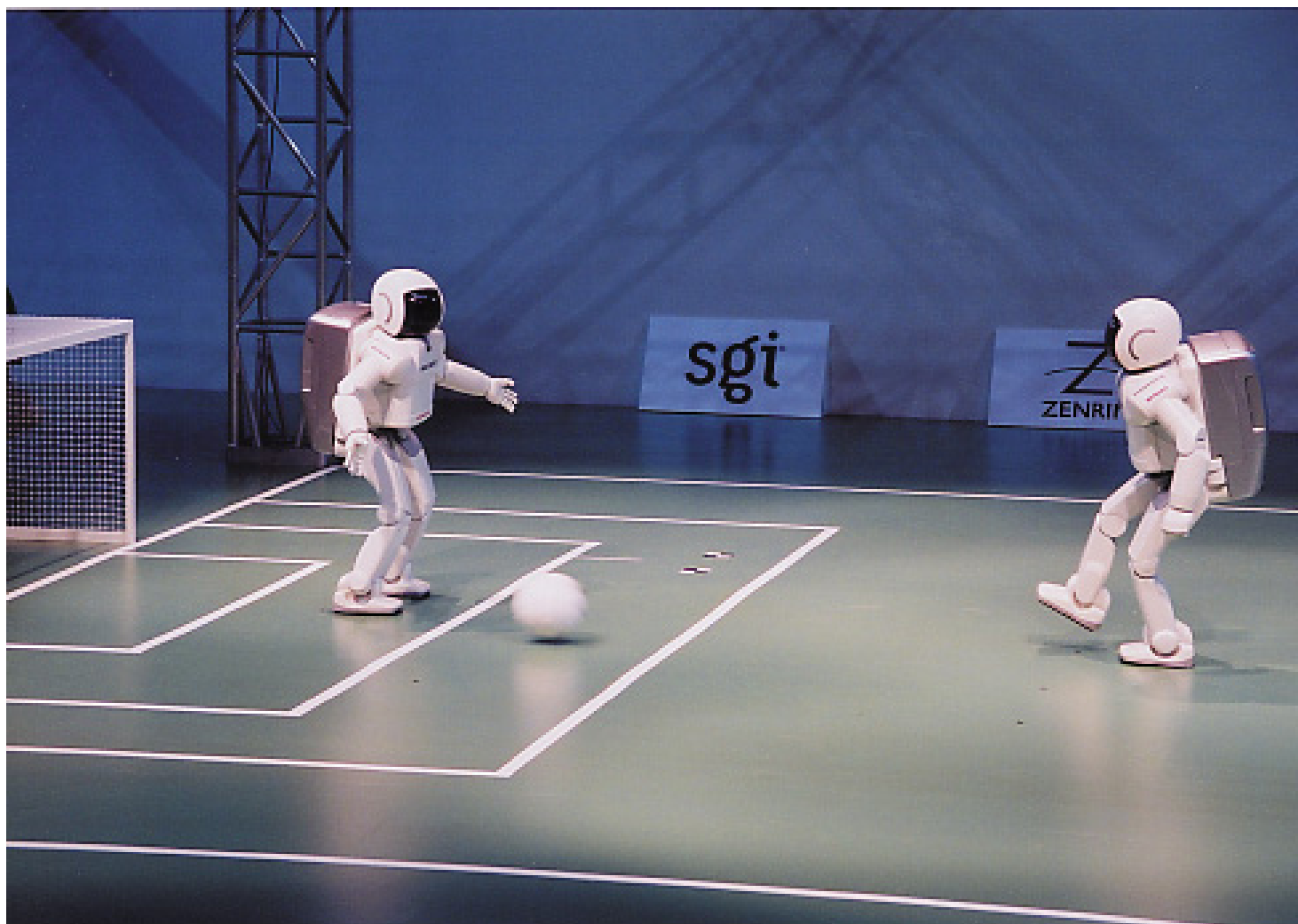


# Categoria “quatro pernas”





# Categoria Humanóides



# Categoria Simulador

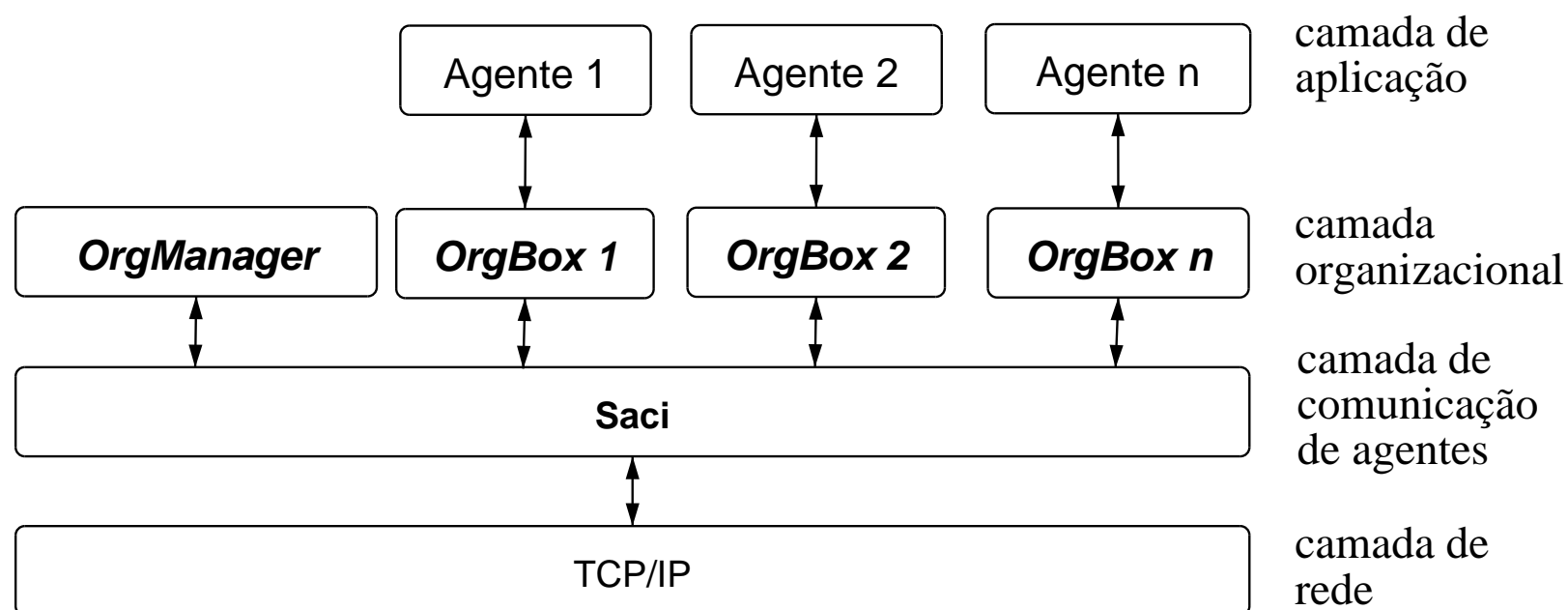


# **Organização em Times de Futebol do Robôs**

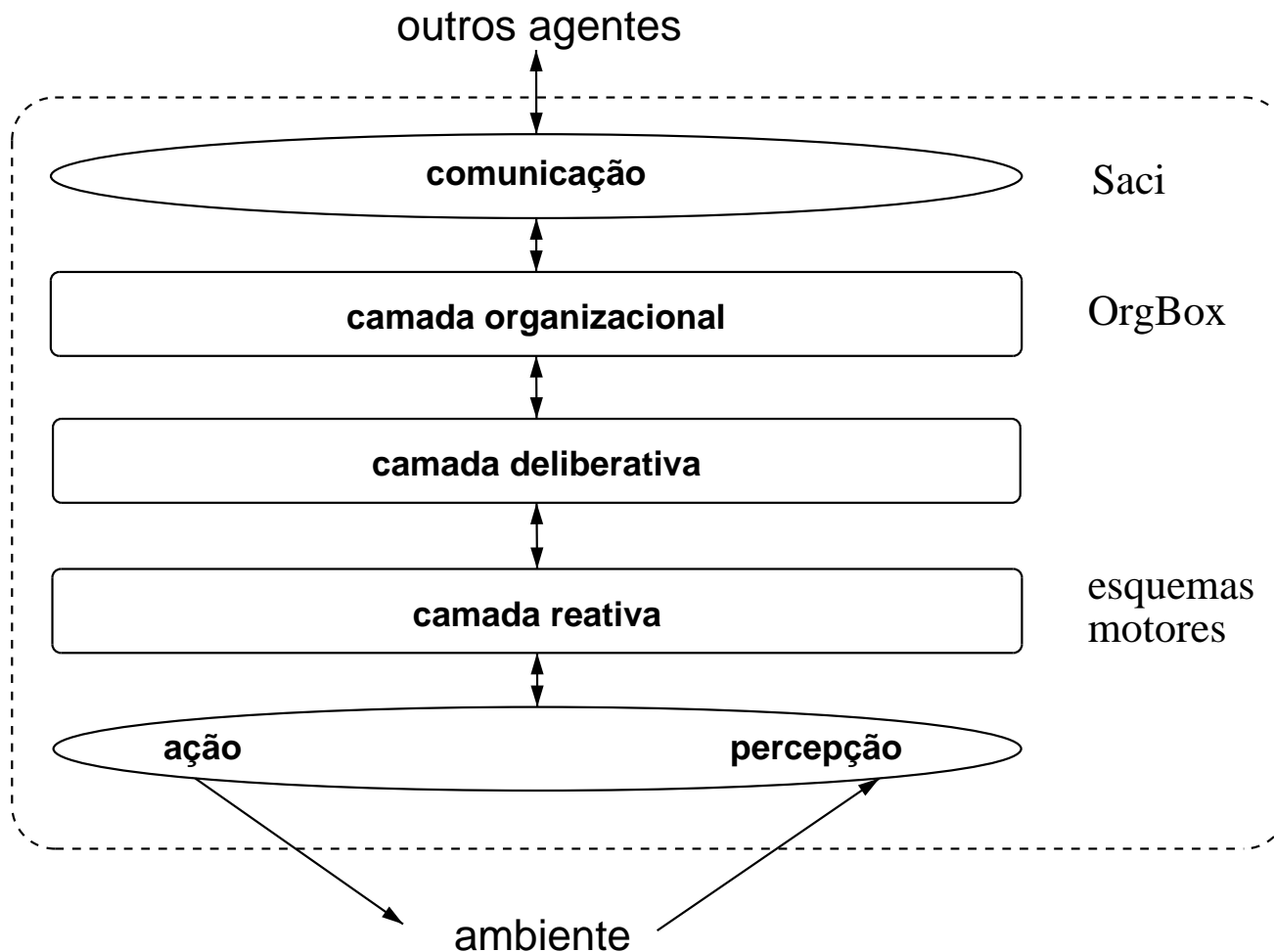
# Objetivo

- Especificar a organização de um time de futebol com o  $\mathcal{MOISE}^+$
- Fazer um time que segue uma especificação organizacional
- Apenas mudando a organização o time muda
- **Adaptação organizacional**

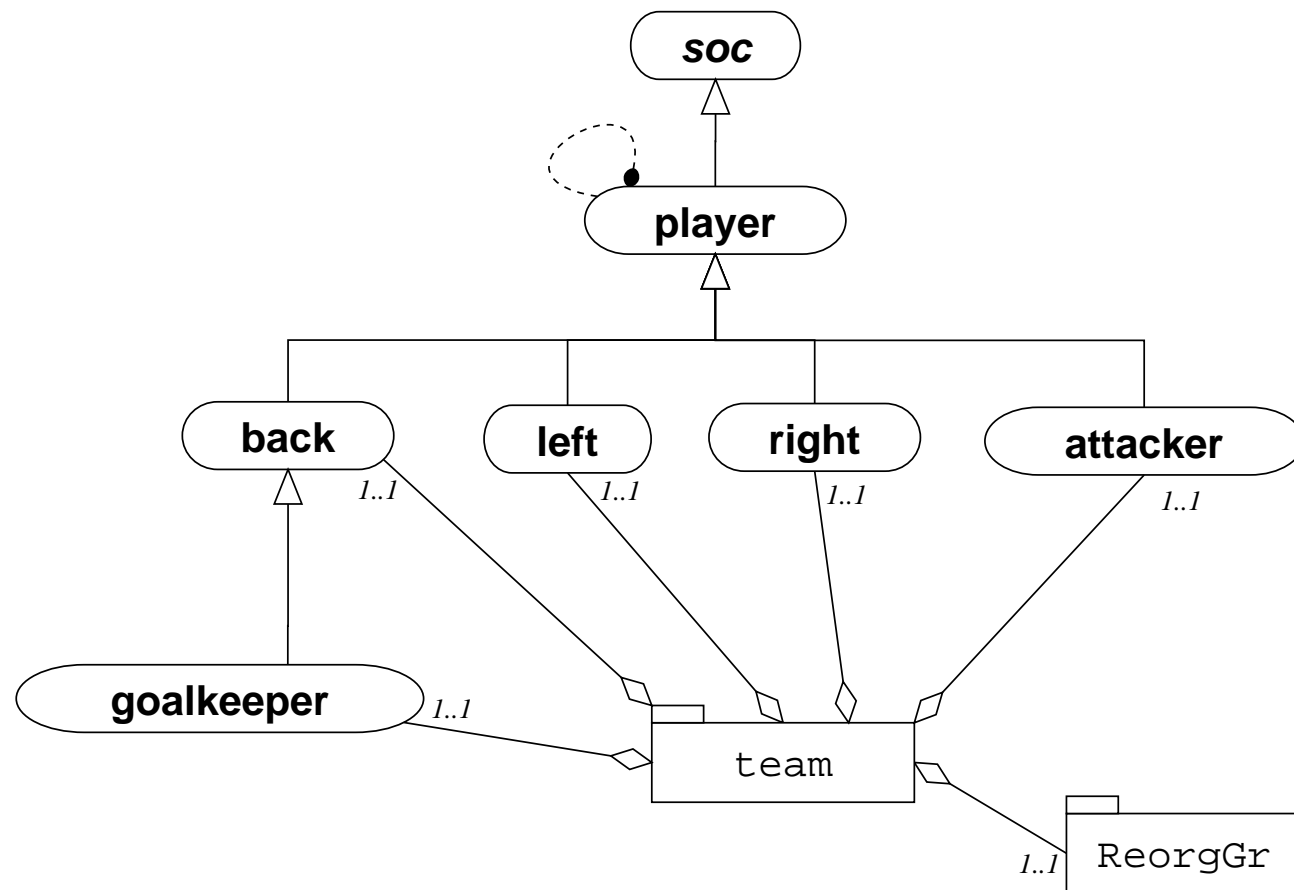
# Visão geral do funcionamento do time

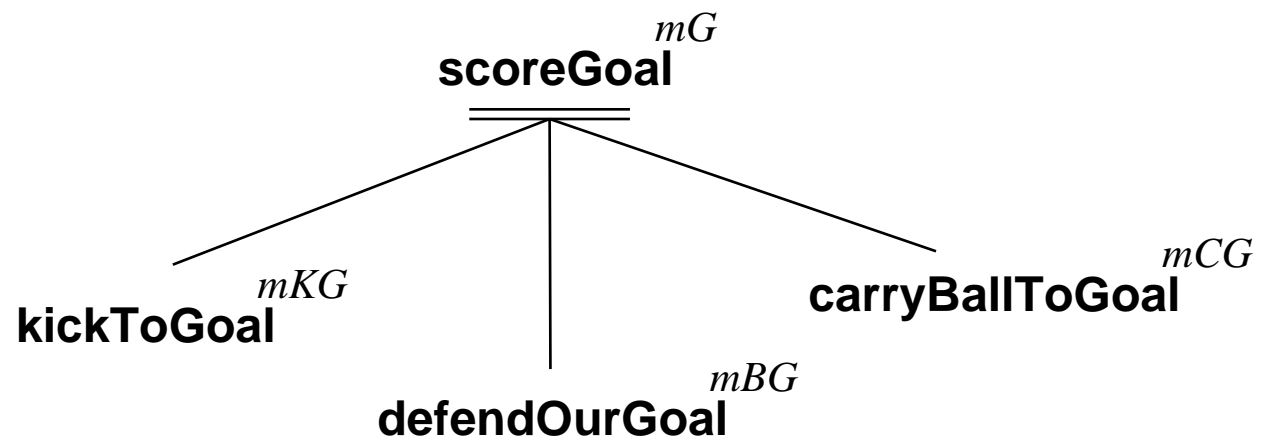


# Arquitetura dos Agentes

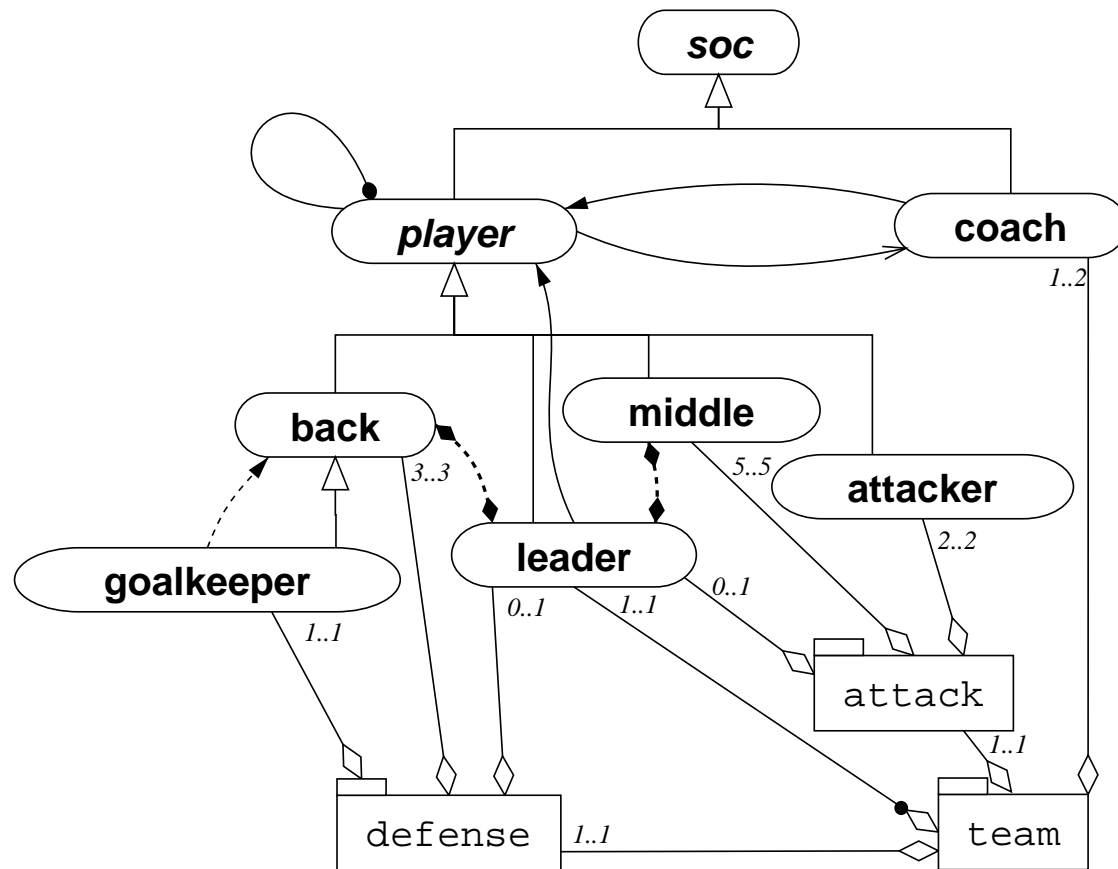


# Exemplo de organização









## Organizational Entity (structure 3-5-2)

Marcos ----- goalkeeper

Lucio ----- back  
Edmilson ----- back

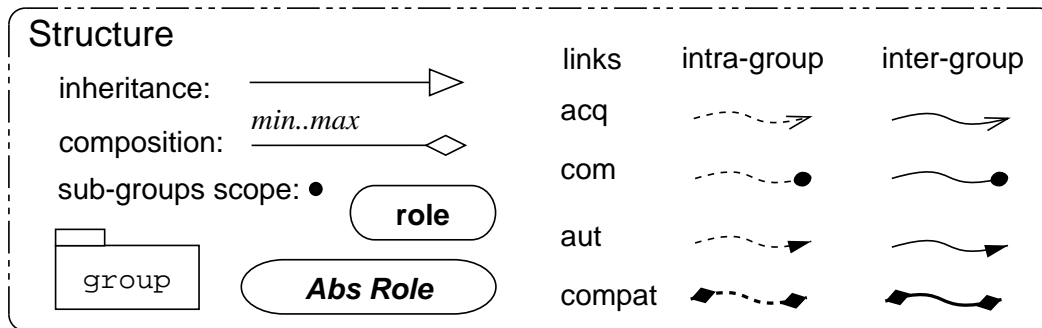
Roque Jr. ----- leader

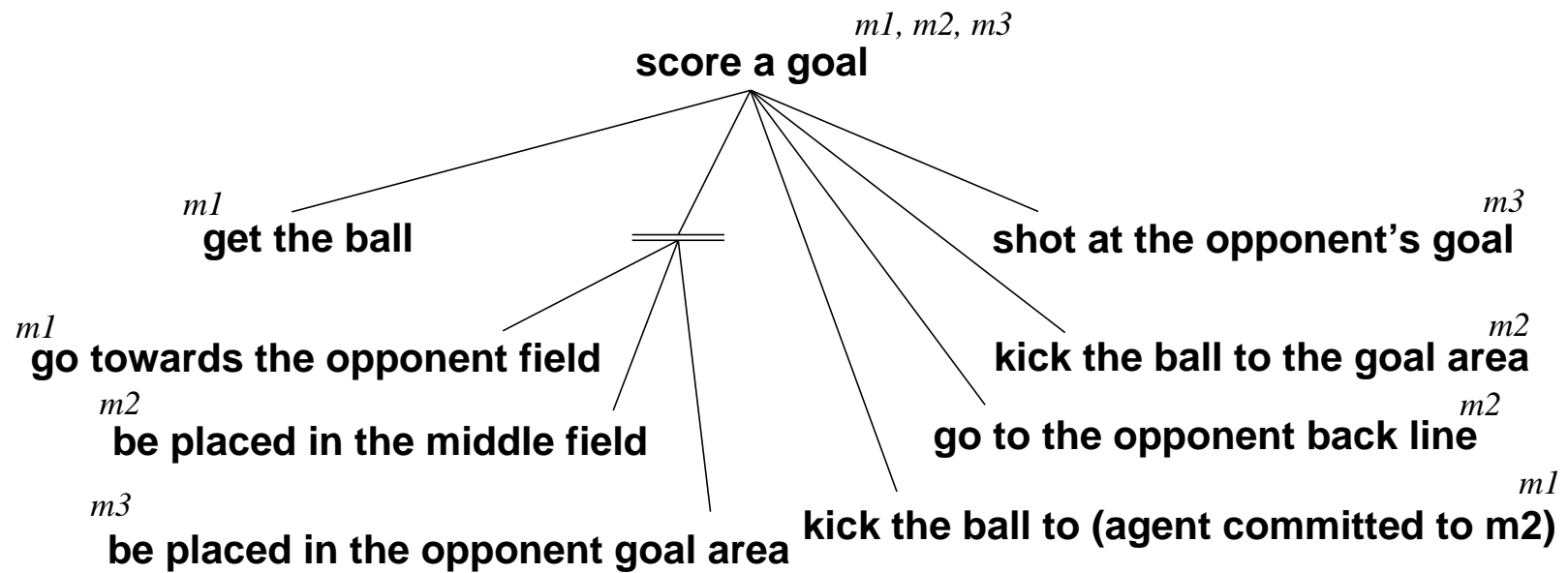
Cafu ----- middle  
Gilberto Silva ----- middle

Juninho ----- middle  
Ronaldinho ----- middle  
Roberto Carlos ----- middle

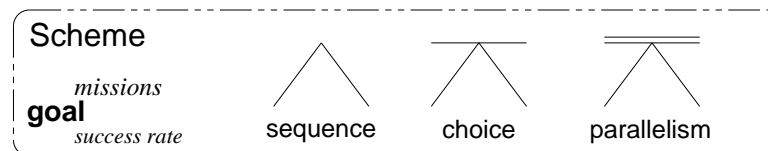
Ronaldo ----- attacker  
Rivaldo ----- attacker

Key

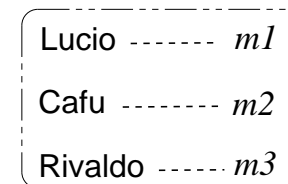




### Key



### Organizational Entity



# Conclusões

- As técnicas de SMA oferecem boas soluções para os problemas de futebol de robôs
- Um time que  **siga**  uma organização tem a vantagem de se adaptar em um nível maior de abstração (a organização)