

---

# Especificação Organizacional de Sistemas Multiagentes e o modelo *Moise*<sup>+</sup>

Jomi Fred Hübner (FURB/SC)  
jomi@inf.furb.br

Brazil Agent School - Setembro 2005

---

## Nota

Estes slides são uma versão atualizada dos slides utilizados nos seguintes cursos/tutoriais:

- Jomi F. Hübner e Jaime S. Schiman. JAIA 2003.
- Olivier Boissier e Jaime S. Schiman. EASSS 2005.

## Roteiro

- **Modelos organizacionais**
- Modelo  $\mathcal{MOISE}^+$
- Programação orientada a organizações

## Noção **intuitiva** de organização

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

## Por que organizar um SMA?

- Se os agentes são autônomos (autonomia de funcionamento, objetivos, etc.), como o sistema vai 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”.
- Na sociedade humana, a noção de papel é muito utilizada para representar direitos e obrigações que, de certa forma, controlam nossa autonomia.

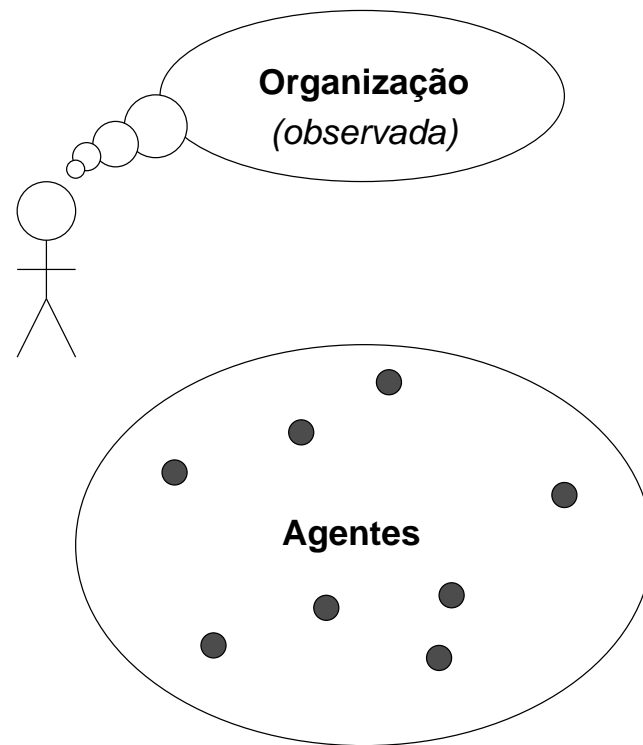
## Motivações para organizar um SMA na perspectiva de um agente (local)

- Facilitar o processo de decisão (diminuir seu espaço de busca)
  - ★ “já que eu **tenho que fazer** o trabalho da disciplina, melhor fazê-lo logo”
  - ★ Noções organizacionais envolvidas no raciocínio: ser aluno, brigades de aluno.
- Facilitar o raciocínio e a interação com os demais
  - ★ “se ele é um **vendedor** de casas, então posso lhe perguntar as ofertas atuais”.
  - ★ Noções organizacionais envolvidas no raciocínio: ser vendedor, obrigações dos vendedores.

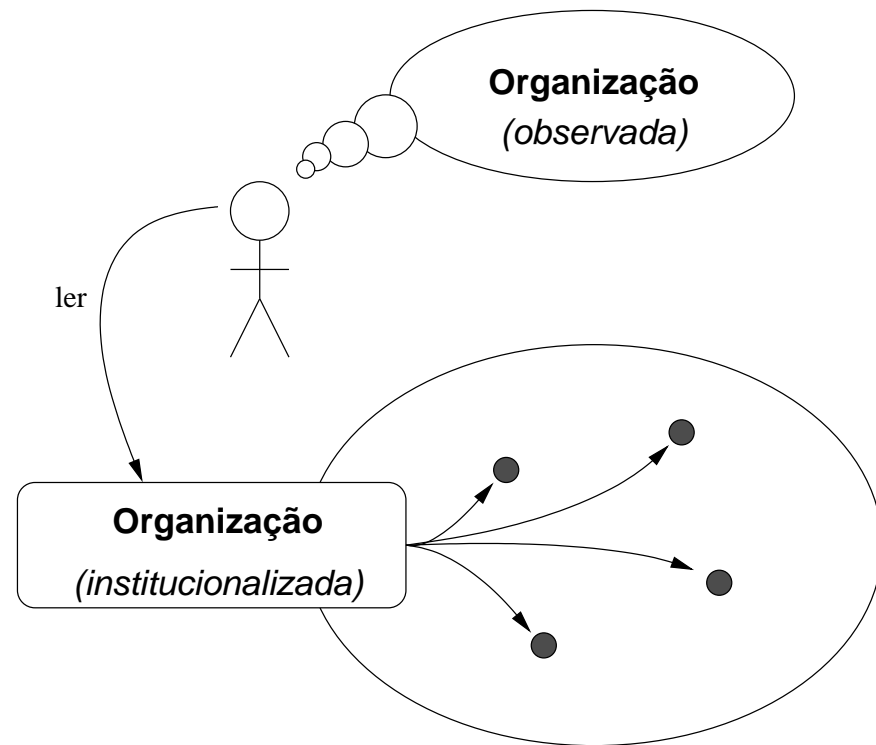
## Motivações para organizar um SMA na perspectiva do sistema (global)

- Facilitar um comportamento global coeso, voltado a uma finalidade.
- Principalmente no contexto de **sistemas abertos**
  - ★ Novos agentes podem entrar no sistema.
  - ★ Esses novos agentes podem ter diferentes arquiteturas, formas de programação, objetivos, ...

# Duas formas de ver organização [Lemaître and Excelente, 1998]



(a) Visão centrada nos agentes



(b) Visão centrada na organização



## Algumas definições

- “Organizations are structured, patterned systems of activity, knowledge, culture, memory, history, and capabilities that are distinct from any single agent” [[Gasser, 2001](#)]

Organização como um fenômeno supra-individual.

- “A decision and communication schema which is applied to a set of actors that together fulfill a set of tasks in order to satisfy goals while guarantying a global coherent state” [[Malone, 1999](#)]

Organização como um padrão de comportamento para atingir um objetivo global.

# O que é Organização

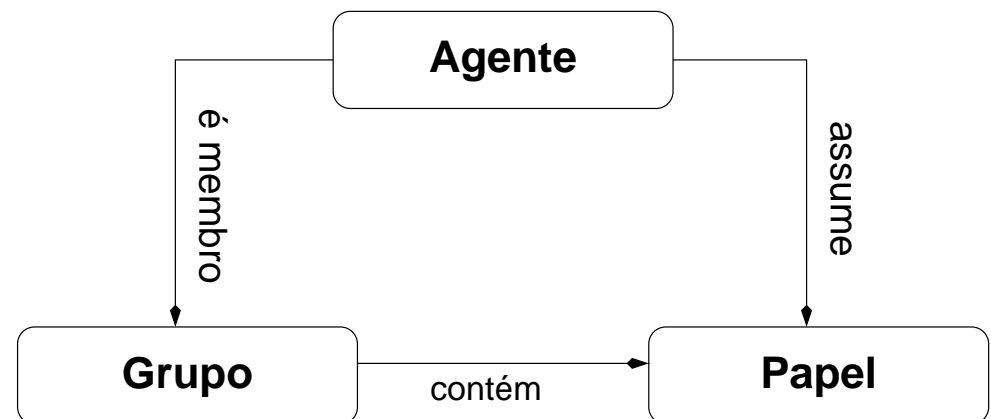
**A organização de um SMA é um conjunto de restrições ao comportamento dos agentes a fim de conduzi-los a uma finalidade comum.** [Dignum and Dignum, 2001]

- Estas restrições podem estar explícitas ou não e os agentes podem ser ou não conscientes delas.
- 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?

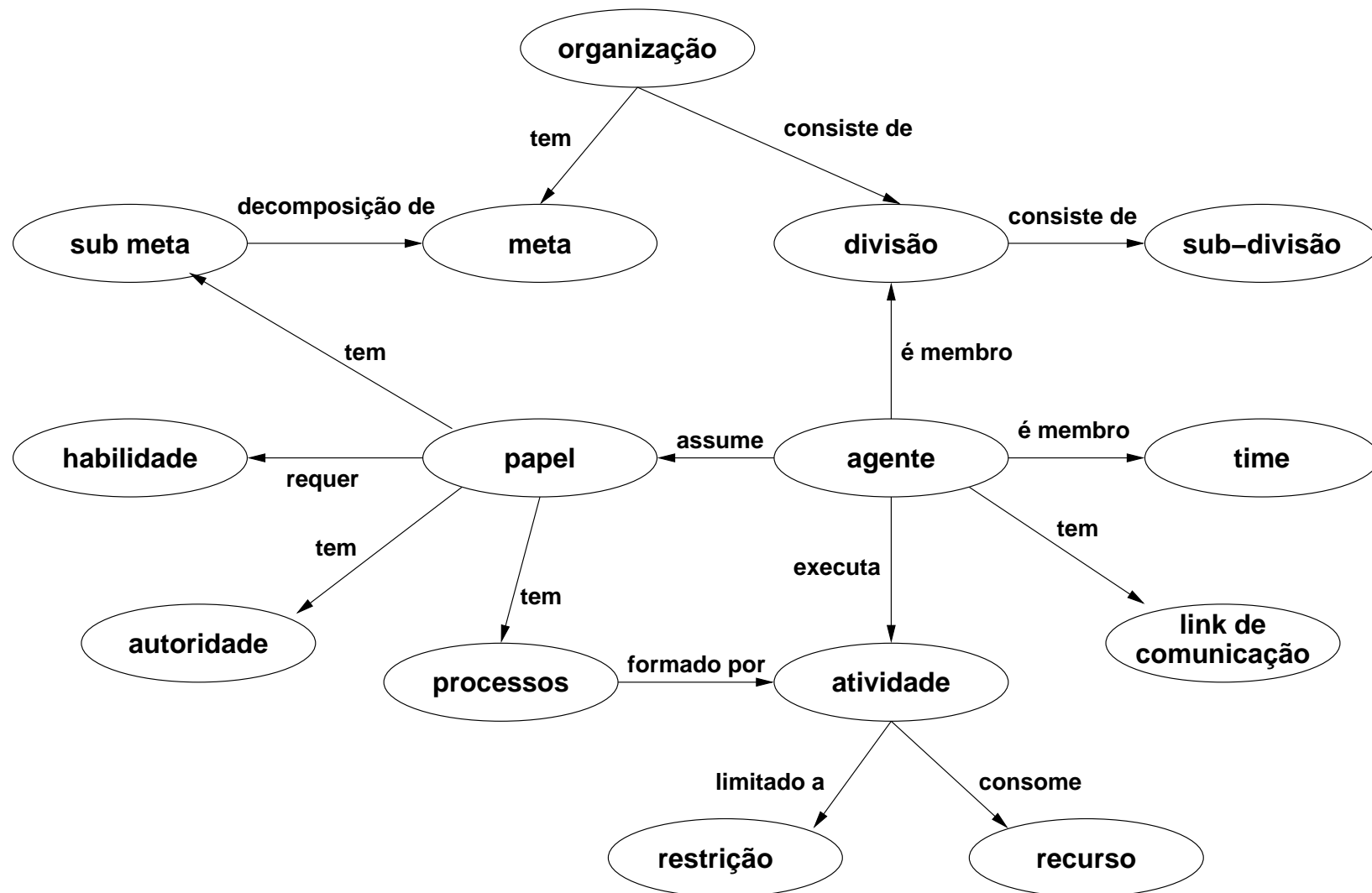
⇒ **Modelos Organizacionais**

## O modelo AGR (Agent Group Role) [Ferber and Gutknecht, 1998]

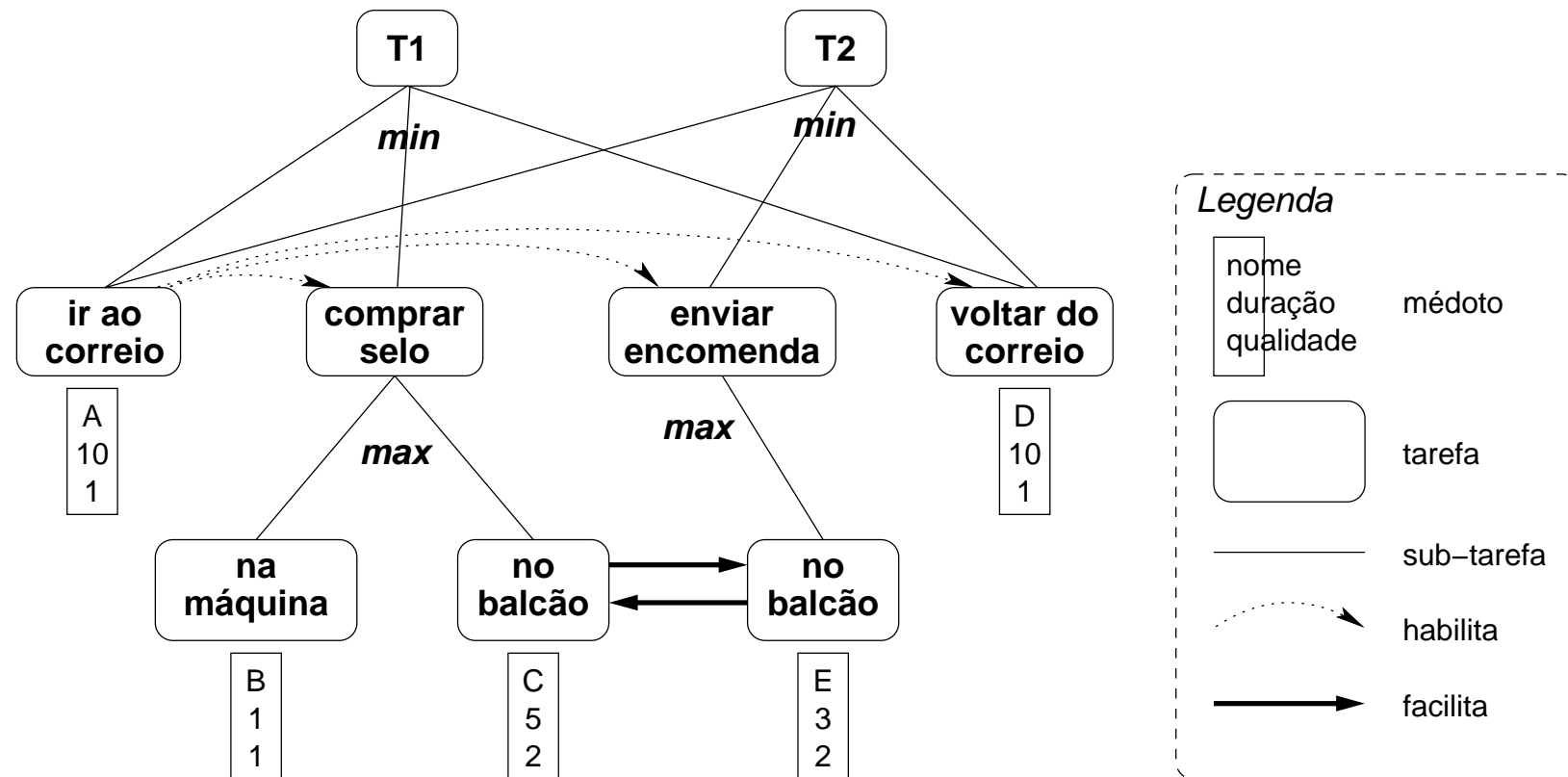
- Uma organização é um conjunto de grupos e agentes com papéis nestes grupos.
- Um grupo tem um conjunto de papéis necessários para seu funcionamento.
- Papel é um conjunto de **funções** que os agentes assumem ao entrar em um grupo.
- Não importa a arquitetura dos agentes.
- Implementado no MADKIT.
- A organização é **instanciada** pelos agentes.



# O modelo TOVE [Fox et al., 1998]

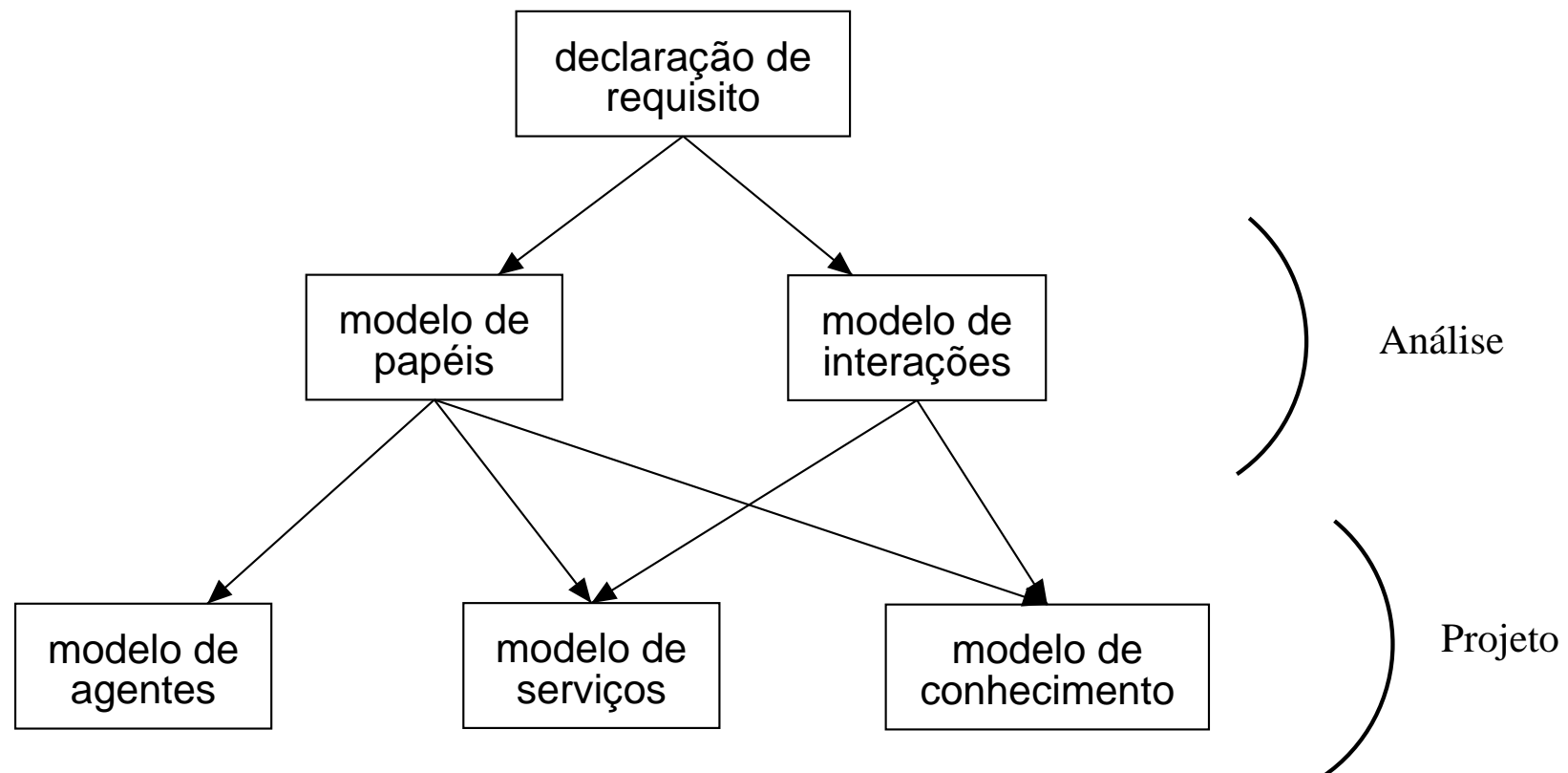


# O modelo TÆMS [Decker, 1998]



# Organização nas metodologias de desenvolvimento de SMA

**GAIA** [Wooldridge et al., 1999]: A organização do sistema é definida por meio dos modelos de papéis e de interação.



Cada papel organizacional é formado por

- responsabilidades (funções que o agente deve realizar para o sistema, definidas por meio de atividades e protocolos),
- permissões para utilização de recursos,
- atividades (ações que o agente pode realizar sozinho) e
- protocolos (como o agente deve interagir com os demais).

**MESSAGE** [Garijo et al., 2001], a organização é composta por

- o propósito da organização (um conjunto de metas que devem ser satisfeitas pelas tarefas);
- estrutura de *workflow* (relação entre as tarefas, suas dependências e dos agentes responsáveis pelas tarefas);
- estrutura organizacional (relação dos papéis e agentes que os assumem);
- entidade de controle (define como se dá a resolução de conflitos);
- recursos da organização; e
- relações organizacionais (utilizadas para formar hierarquias entre papéis, por exemplo).



**MESSAGE** [Garijo et al., 2001], a organização é composta por

- o propósito da organização (um conjunto de metas que devem ser satisfeitas pelas tarefas);
- estrutura de *workflow* (relação entre as tarefas, suas dependências e dos agentes responsáveis pelas tarefas);
- estrutura organizacional (relação dos papéis e agentes que os assumem);
- entidade de controle (define como se dá a resolução de conflitos);
- recursos da organização; e
- relações organizacionais (utilizadas para formar hierarquias entre papéis, por exemplo).

Nestas metodologias, a organização não faz parte do SMA, não serve aos agentes mas ao projetista.

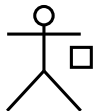
# Síntese: **Tipos** de organização [Hübner, 2003]

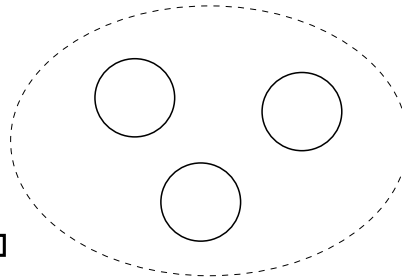
agentes **sem** conhecimento  
organizacional

agentes **com** conhecimento  
organizacional

(a) tipo AR

centrado nos  
agentes

  
observador



# Síntese: **Tipos** de organização [Hübner, 2003]

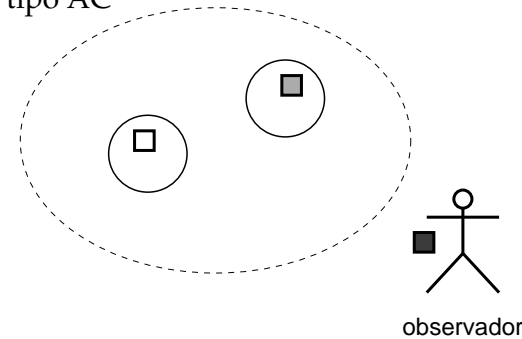
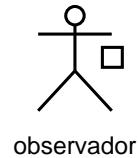
agentes **sem** conhecimento  
organizacional

agentes **com** conhecimento  
organizacional

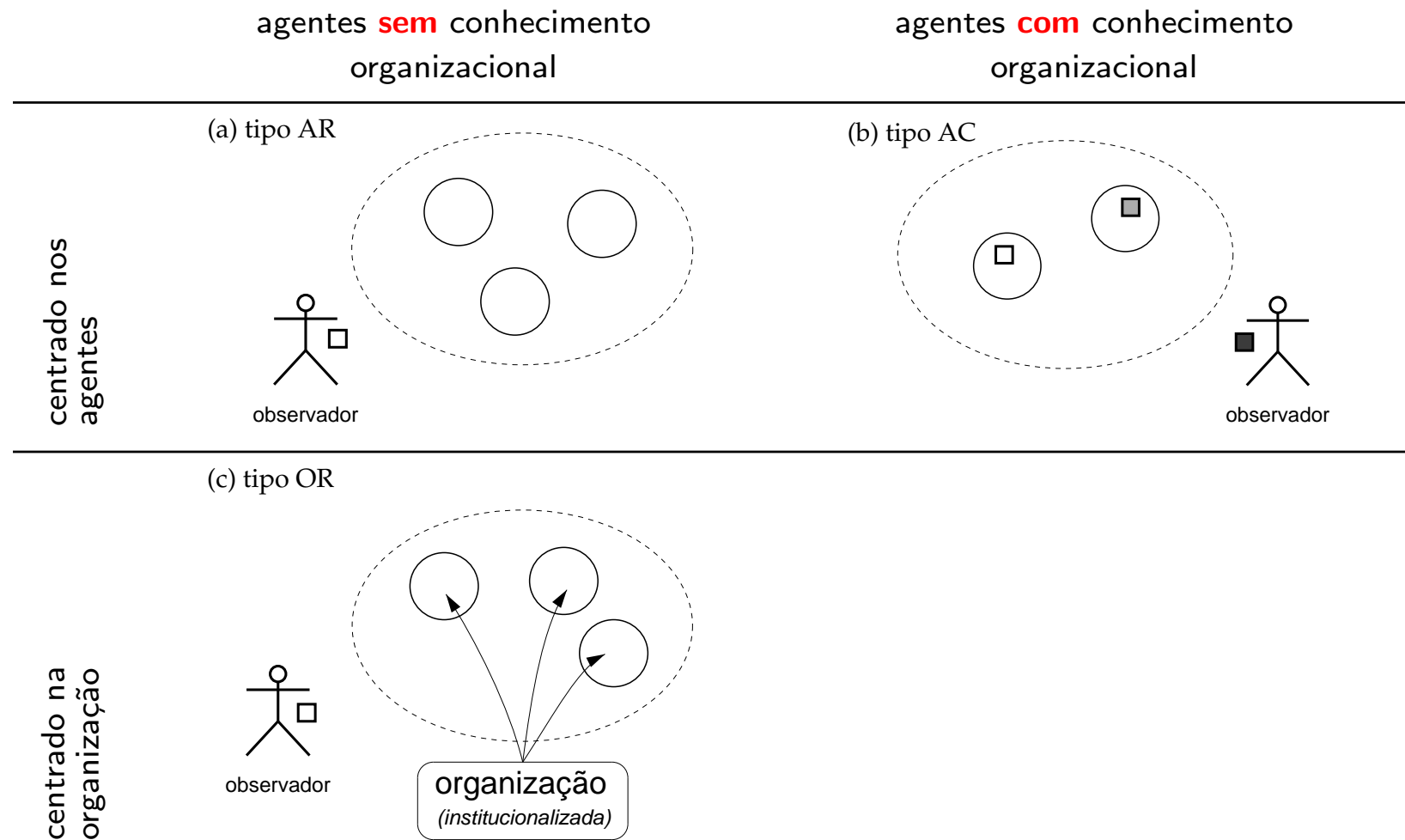
(a) tipo AR

(b) tipo AC

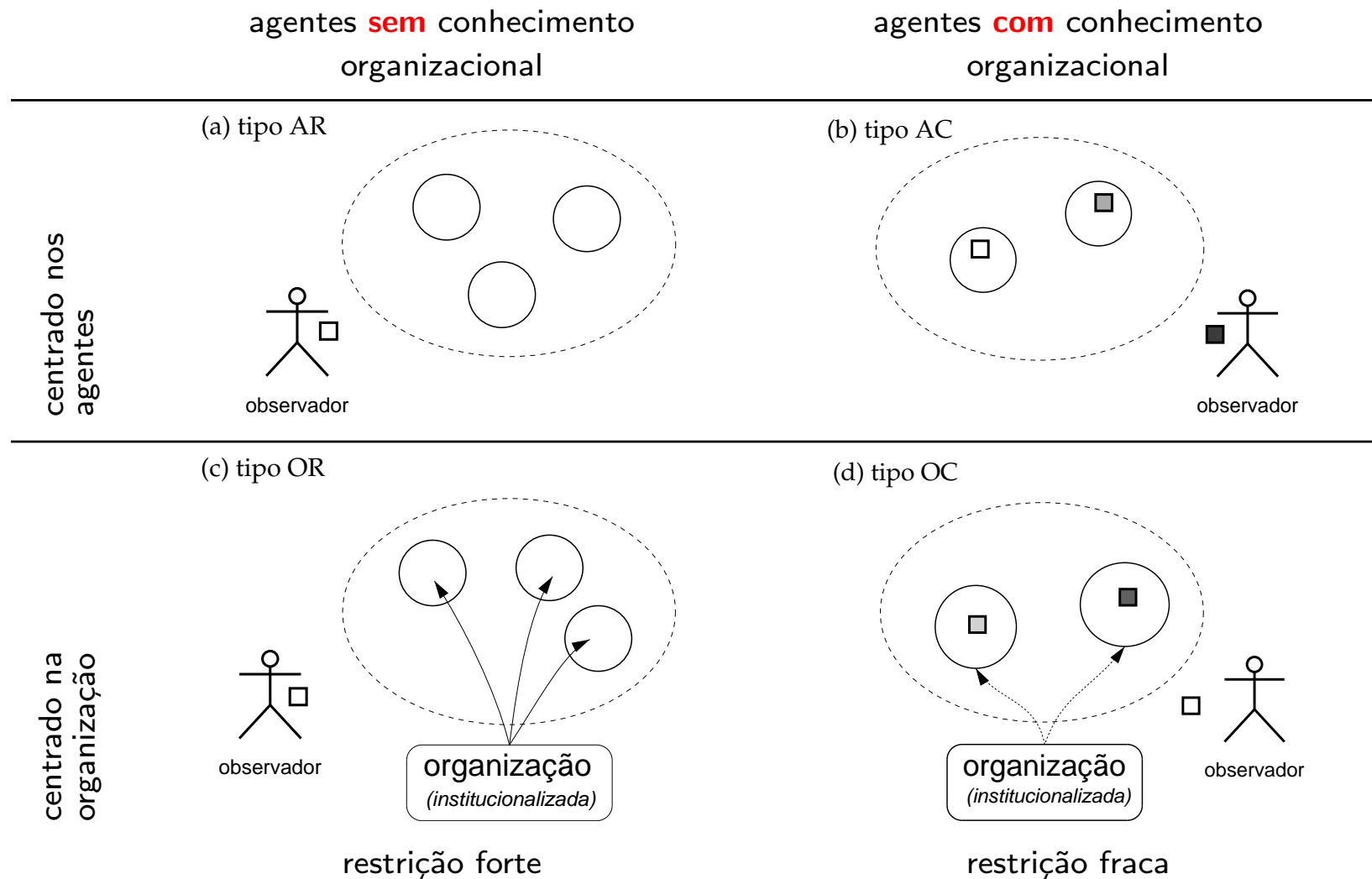
centrado nos  
agentes



# Síntese: **Tipos** de organização [Hübner, 2003]



# Síntese: **Tipos** de organização [Hübner, 2003]



## Resumo

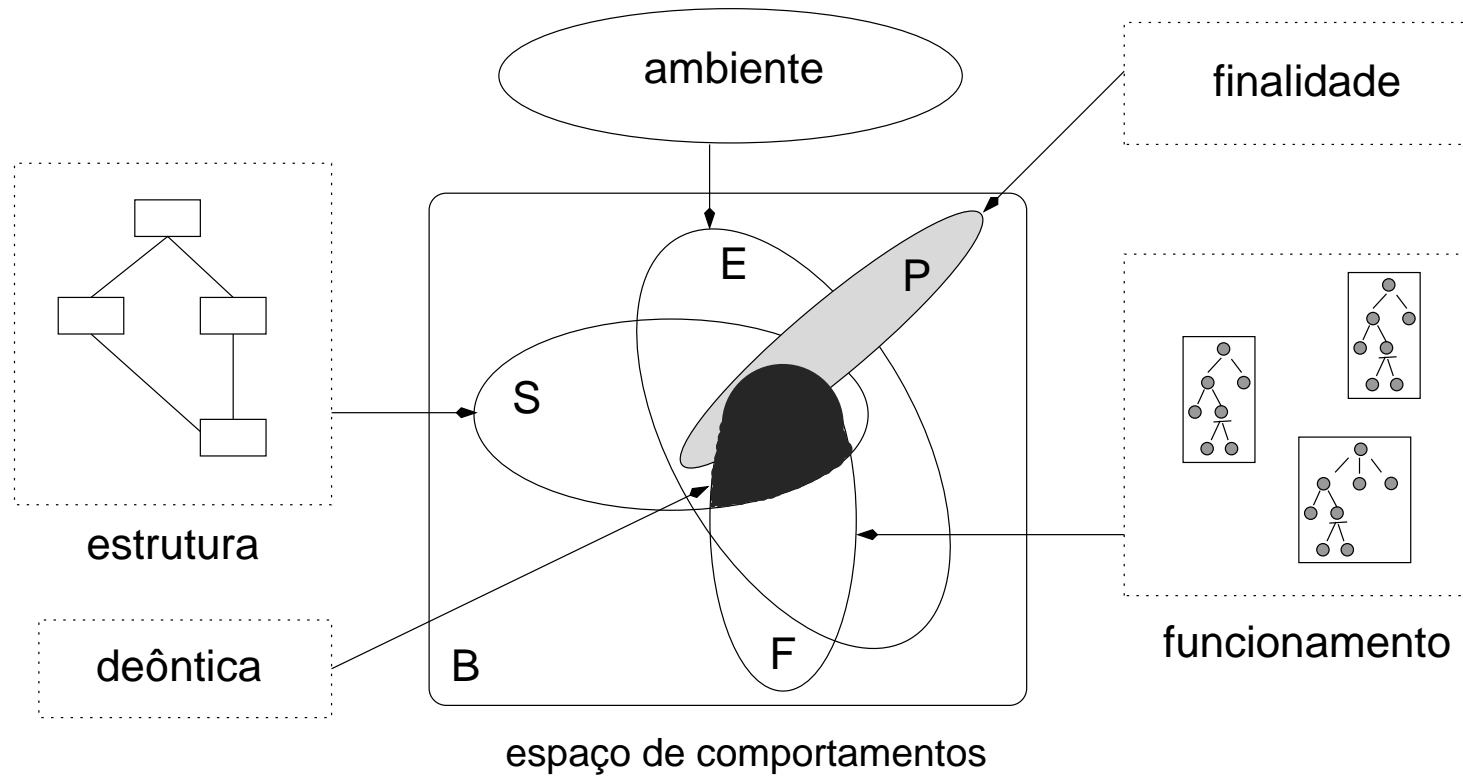
**A organização de um SMA é um conjunto de restrições ao comportamento dos agentes a fim de conduzi-los a uma finalidade comum.**

- Organização observada
- Organização institucionalizada, descrita a partir de modelos
  - ★ Funcionais (TÆMS)
  - ★ Estruturais (AGR)
  - ★ Mistos (TOVE)
- Os agentes podem ou não ter capacidade de raciocinar sobre sua organização.

## Roteiro

- Modelos organizacionais
- **Modelo**  $\mathcal{M}$ oise<sup>+</sup>
- Programação orientada a organizações

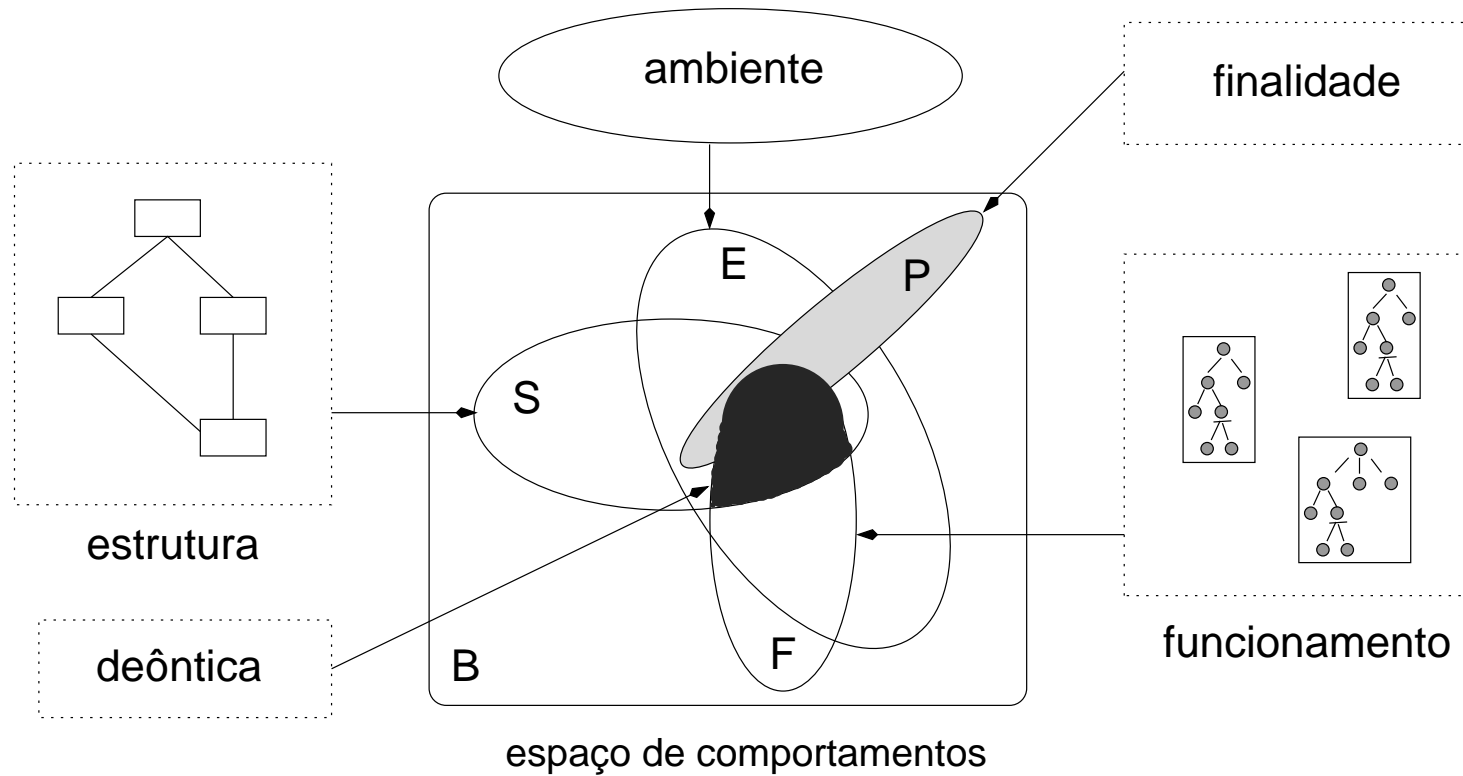
# Visão geral



- **Estrutura:** o que os agentes **podem** fazer

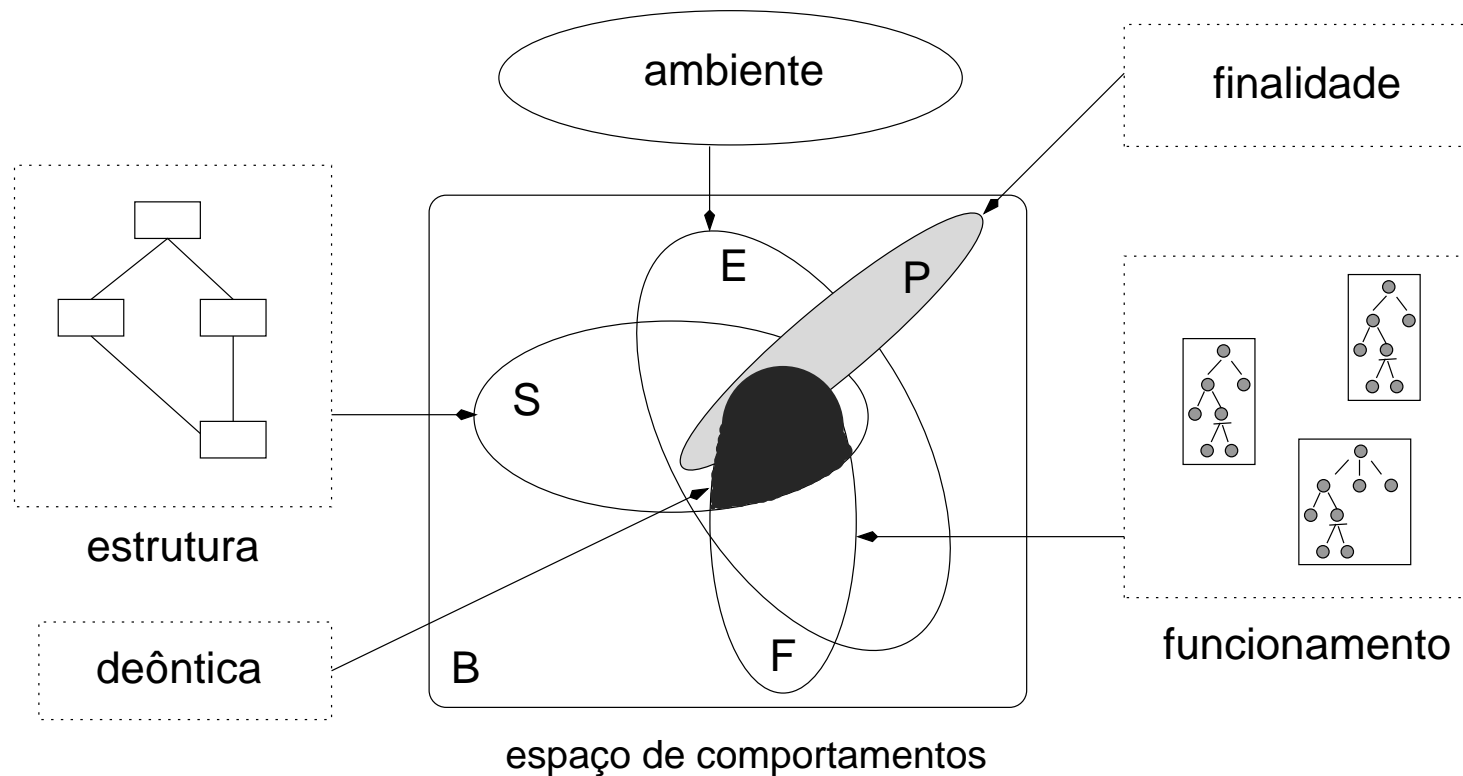


# Visão geral



- **Estrutura:** o que os agentes **podem** fazer
- **Funcionamento:** **como** os agentes podem fazer

# Visão geral



- **Estrutura:** o que os agentes **podem** fazer
- **Funcionamento:** **como** os agentes podem fazer
- **Deôntica:** o que os agentes **devem** fazer

## Especificação Estrutural

- **Papel**: conjunto de restrições comportamentais que **um agente** aceita quanto entra em um grupo
  - ★ em relação a outros agentes (exemplo: autoridade) e

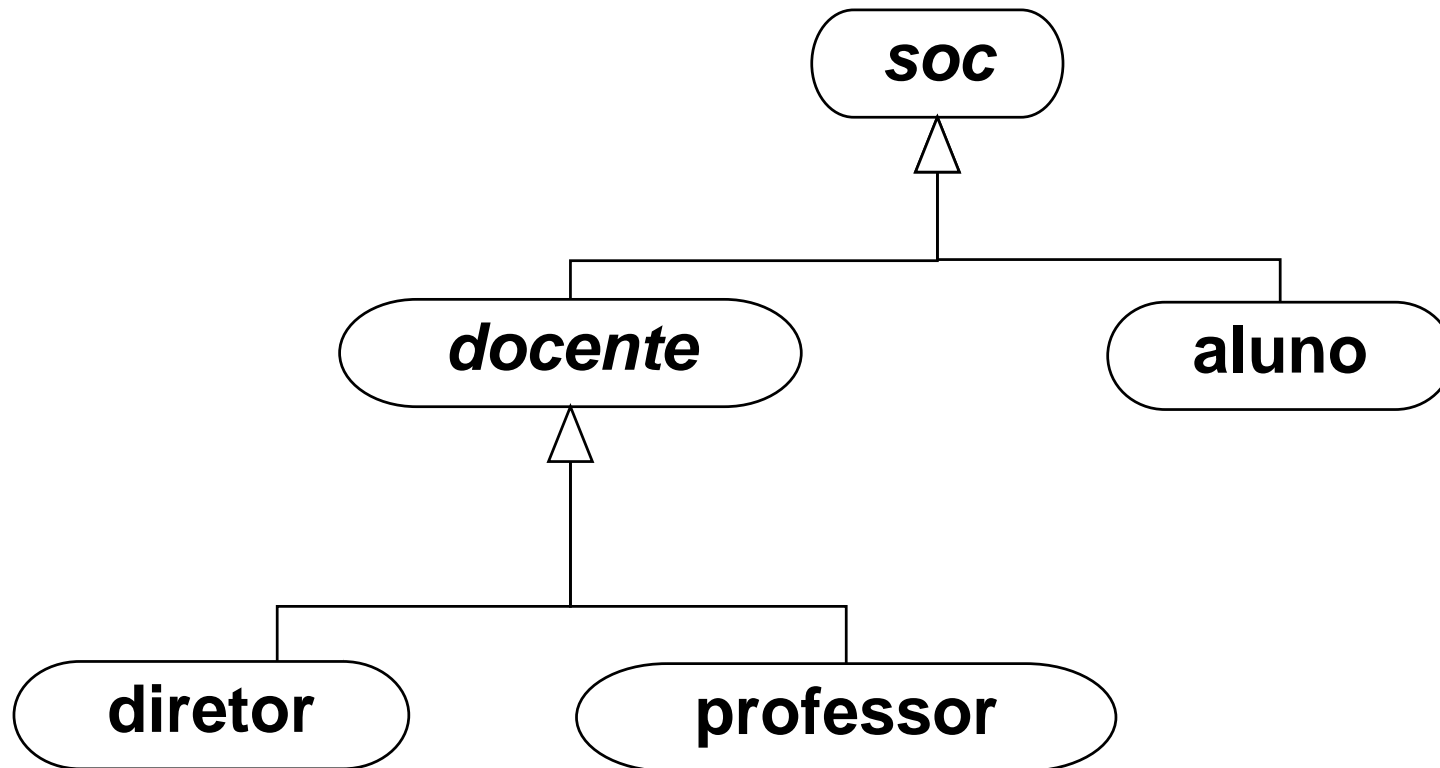
## Especificação Estrutural

- **Papel**: conjunto de restrições comportamentais que **um agente** aceita quanto entra em um grupo
  - ★ em relação a outros agentes (exemplo: autoridade) e
  - ★ em relação a tarefas comuns (objetivos globais)

## Especificação Estrutural

- **Papel:** conjunto de restrições comportamentais que **um agente** aceita quanto entra em um grupo
  - ★ em relação a outros agentes (exemplo: autoridade) e
  - ★ em relação a tarefas comuns (objetivos globais)
- A especificação estrutural é feita em três níveis
  - ★ individual: definição dos papéis
  - ★ social: ligação entre papéis
  - ★ coletiva: agrupamento de papéis

## Relação de **herança** entre papéis



Notação para “o papel  $\rho$  tem uma especialização  $\rho'$ ”:

$$\rho \sqsubseteq \rho'$$

Notação para “o papel  $\rho$  tem uma especialização  $\rho'$ ”:

$$\rho \sqsubset \rho'$$

$$\rho_{docente} \sqsubset \rho_{diretor}$$



Notação para “o papel  $\rho$  tem uma especialização  $\rho'$ ”:

$$\rho \sqsubset \rho'$$

$$\rho_{docente} \sqsubset \rho_{diretor}$$

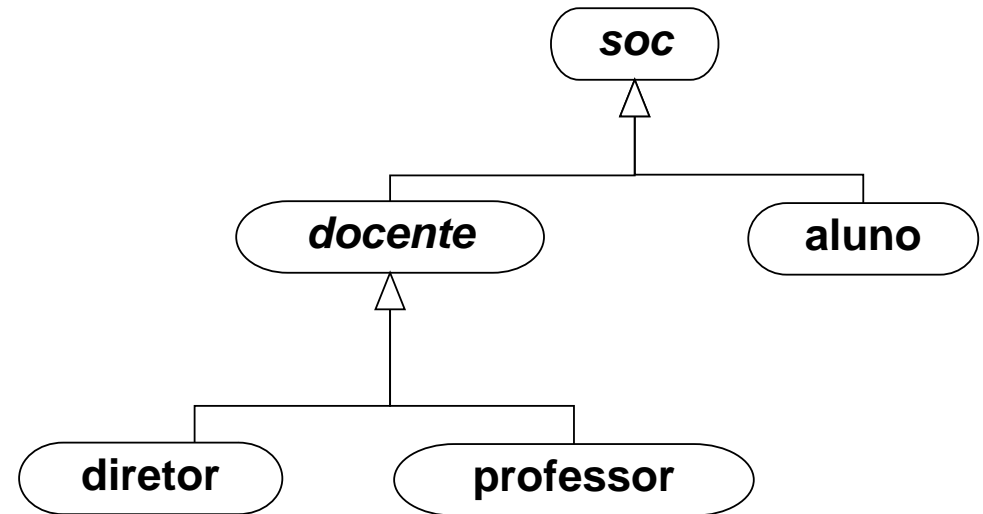
Propriedades:

$$\rho \sqsubset \rho' \wedge \rho' \sqsubset \rho \Rightarrow \rho = \rho'$$

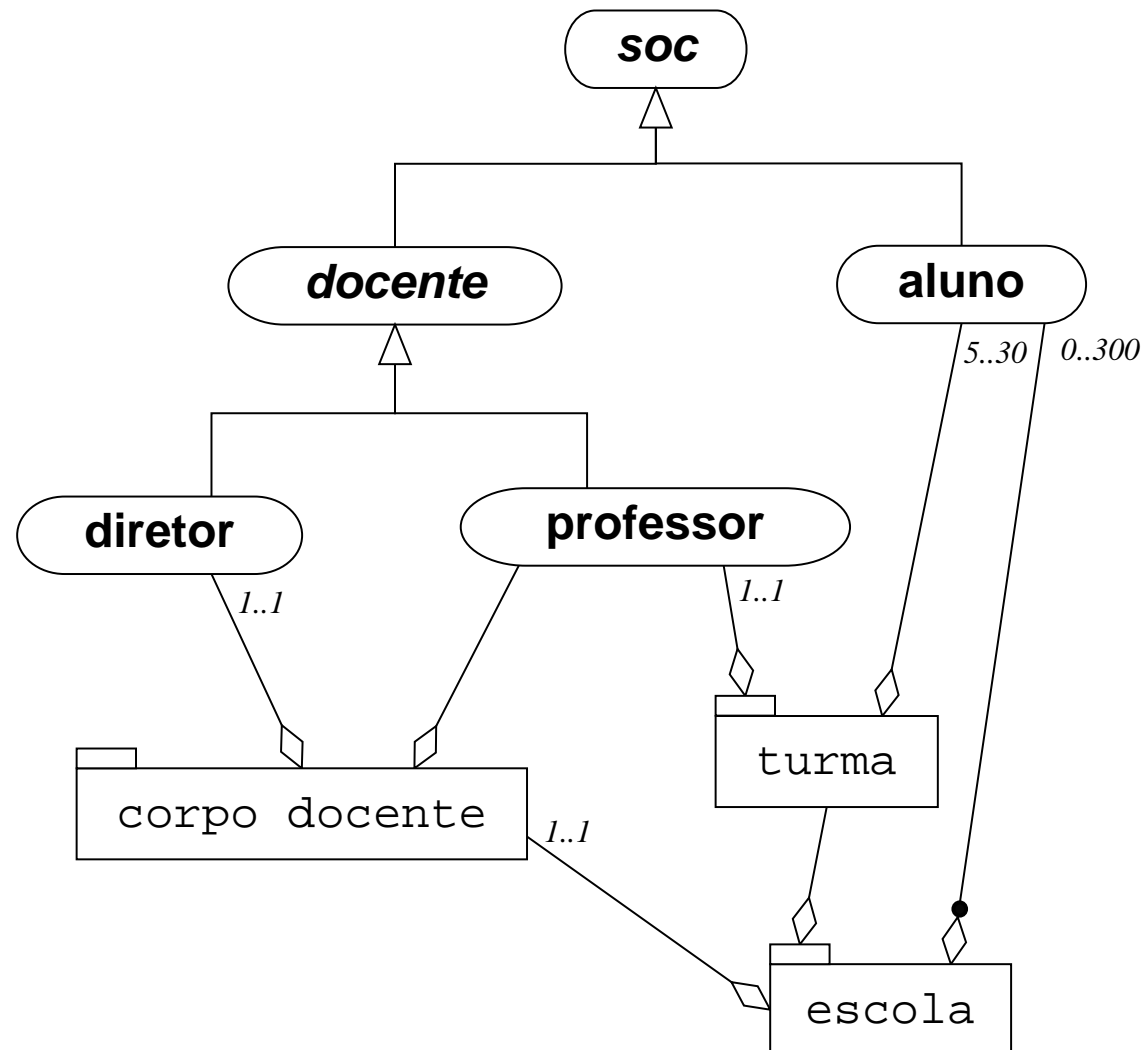
$$\rho \sqsubset \rho' \wedge \rho' \sqsubset \rho'' \Rightarrow \rho \sqsubset \rho''$$

$$\forall \rho \bullet \rho_{soc} \sqsubset \rho$$

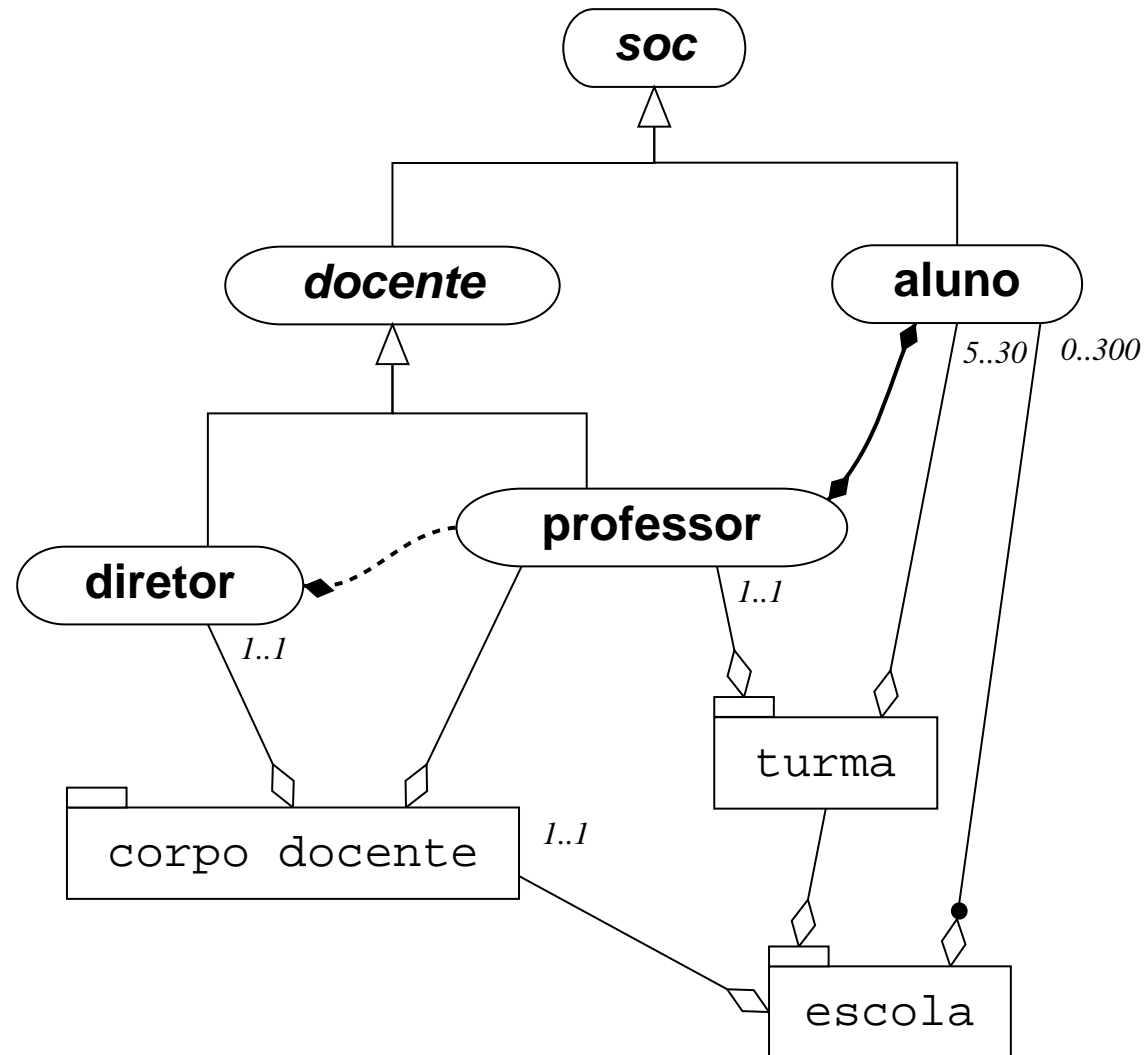
$$\nexists \rho \bullet \rho \sqsubset \rho_{soc}$$



## Nível coletivo: Grupos



## Nível social: **Compatibilidades** entre papéis



Notação para “agente com o papel  $\rho$  também pode assumir o papel  $\rho'$ ”:

$$\rho \bowtie \rho'$$

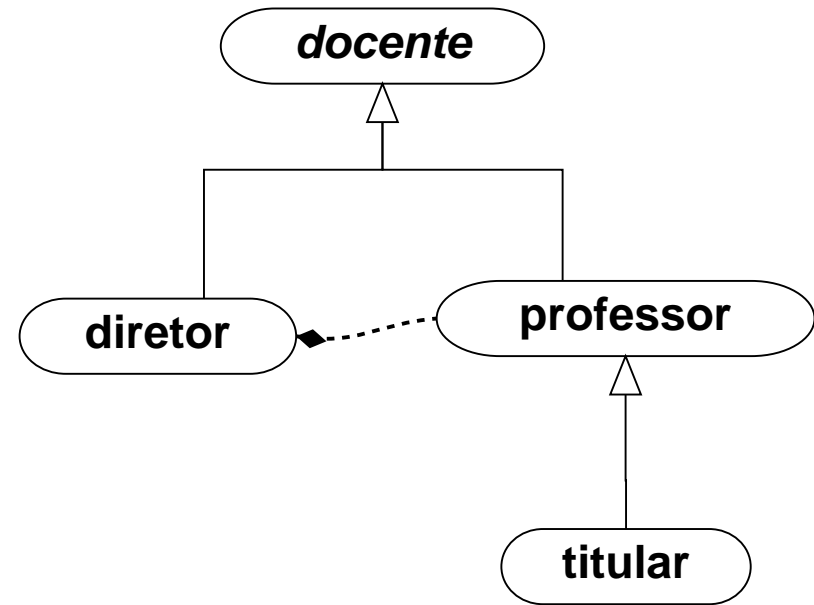
Notação para “agente com o papel  $\rho$  também pode assumir o papel  $\rho'$ ”:

$$\rho \bowtie \rho'$$

$$\rho_{professor} \bowtie \rho_{diretor}$$

Notação para “agente com o papel  $\rho$  também pode assumir o papel  $\rho'$ ”:

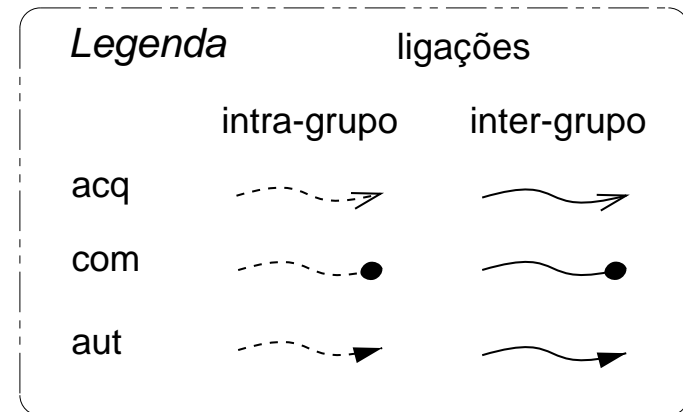
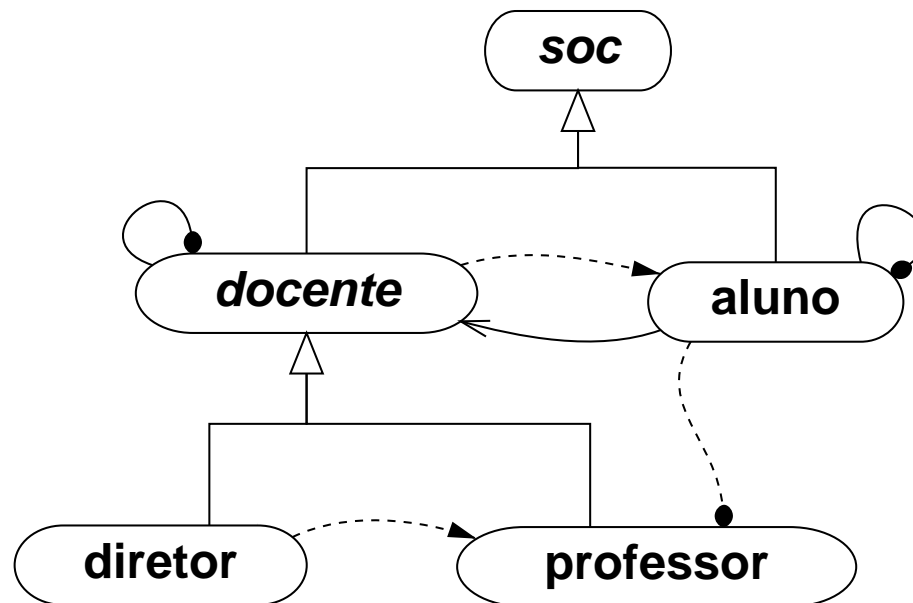
$$\begin{aligned}\rho &\bowtie \rho' \\ \rho_{professor} &\bowtie \rho_{diretor}\end{aligned}$$



Propriedades:

$$\begin{aligned}\rho &\bowtie \rho \\ \rho &\bowtie \rho' \wedge \rho' \bowtie \rho'' \Rightarrow \rho \bowtie \rho'' \\ \rho_a \bowtie \rho_b \wedge \rho_a \not\sqsubseteq \rho_b \wedge \rho_a \sqsubseteq \rho' &\Rightarrow \rho' \bowtie \rho_b \\ \rho \sqsubseteq \rho' &\Rightarrow \rho' \bowtie \rho\end{aligned}$$

## Nível social: **Ligações** entre papéis



Notação para “o papel  $\rho_s$  tem uma ligação do tipo  $t$  com o papel  $\rho_d$ ”:

$$link(\rho_s, \rho_d, t)$$



Notação para “o papel  $\rho_s$  tem uma ligação do tipo  $t$  com o papel  $\rho_d$ ”:

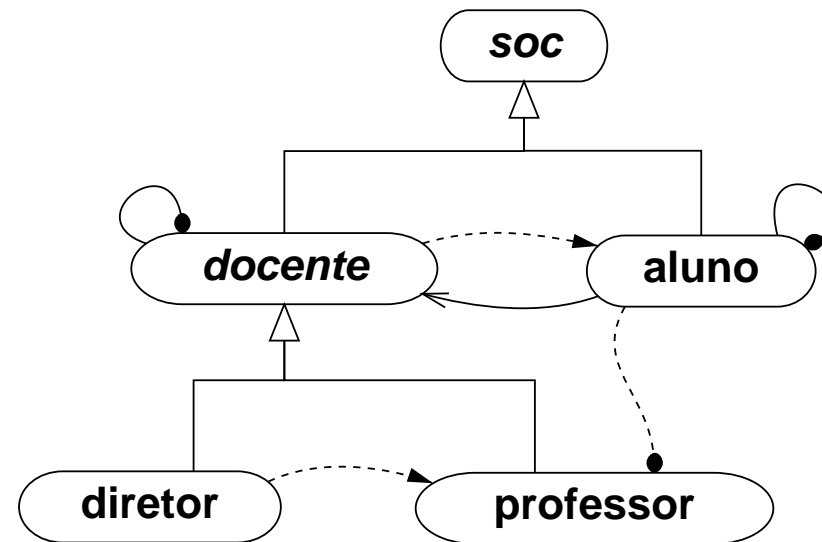
$$link(\rho_s, \rho_d, t)$$

$$link(\rho_{docente}, \rho_{aluno}, aut)$$

Notação para “o papel  $\rho_s$  tem uma ligação do tipo  $t$  com o papel  $\rho_d$ ”:

$$link(\rho_s, \rho_d, t)$$

$$link(\rho_{docente}, \rho_{aluno}, aut)$$



Propriedades:

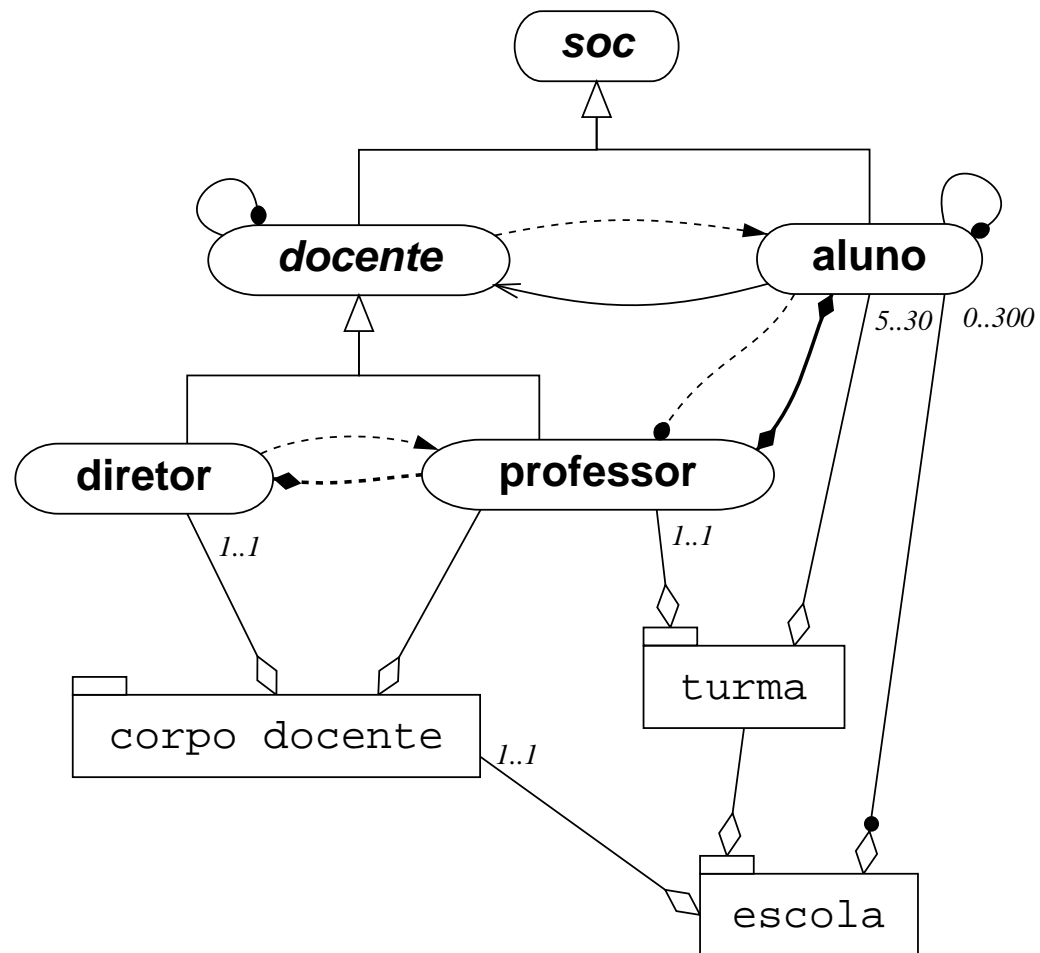
$$link(\rho_s, \rho_d, t) \wedge \rho_s \sqsubset \rho'_s \Rightarrow link(\rho'_s, \rho_d, t)$$

$$link(\rho_s, \rho_d, t) \wedge \rho_d \sqsubset \rho'_d \Rightarrow link(\rho_s, \rho'_d, t)$$

$$link(\rho_s, \rho_d, aut) \Rightarrow link(\rho_s, \rho_d, com)$$

$$link(\rho_s, \rho_d, com) \Rightarrow link(\rho_s, \rho_d, acq)$$

# Exemplo I



## Legenda

herança:

composição: *min..max*

escopo de sub-grupos:

grupo

papel

*papel abstrato*

ligações

intra-grupo

inter-grupo

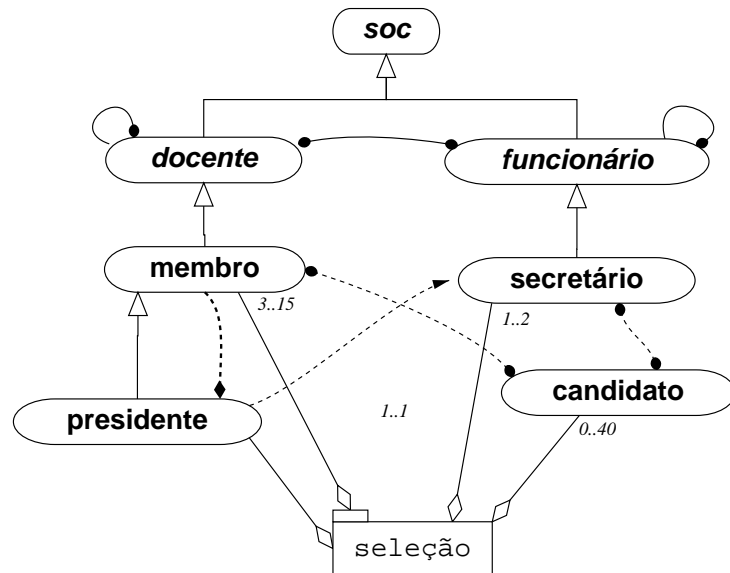
acq

com

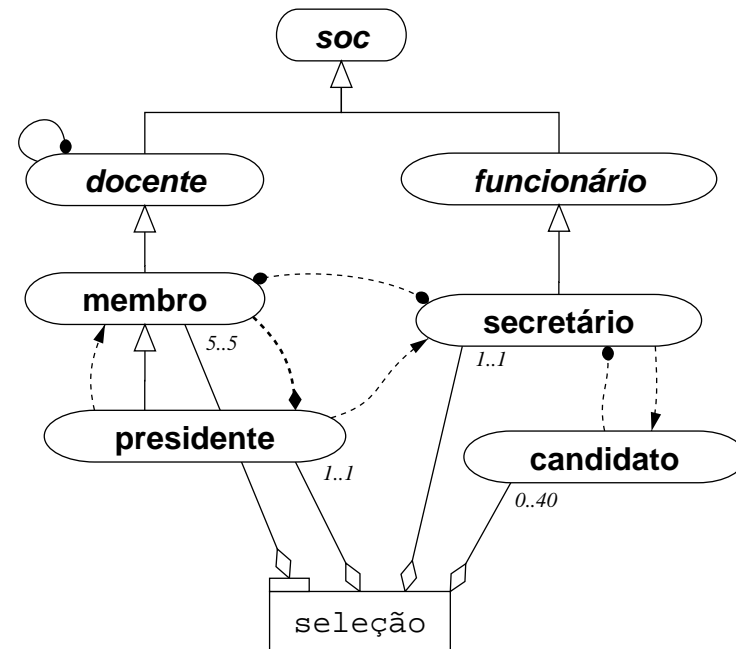
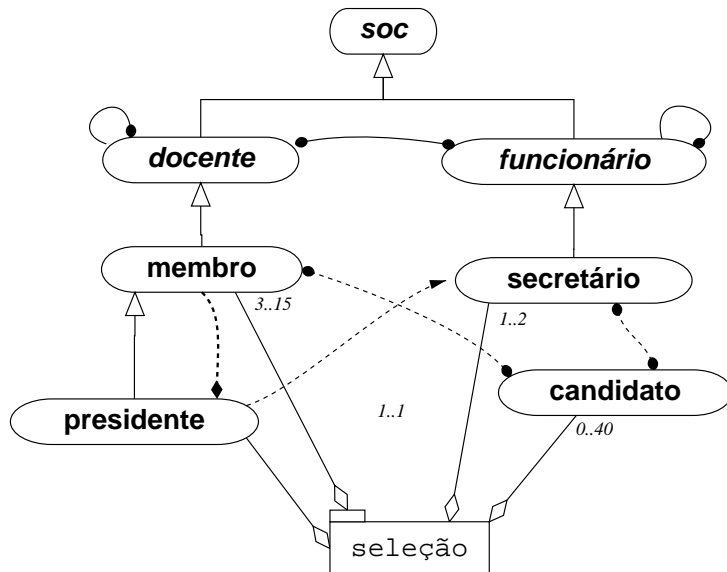
aut

compat

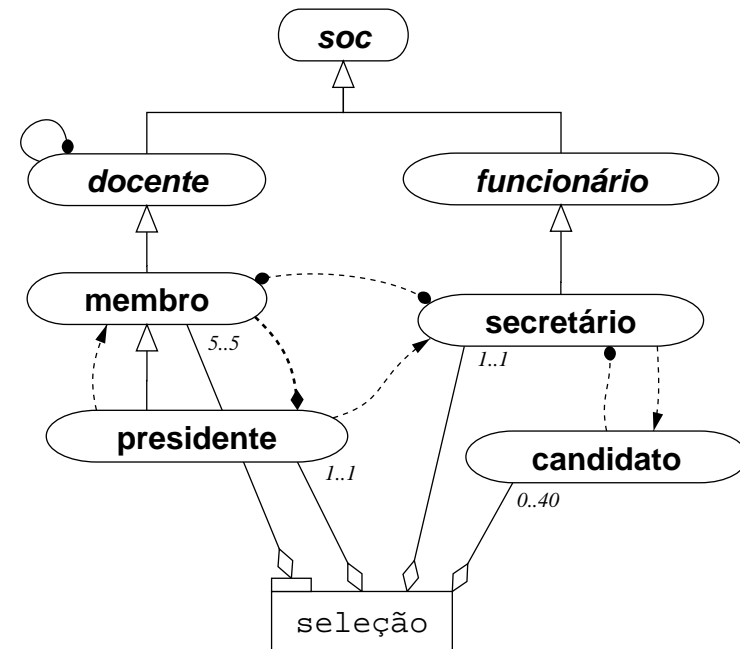
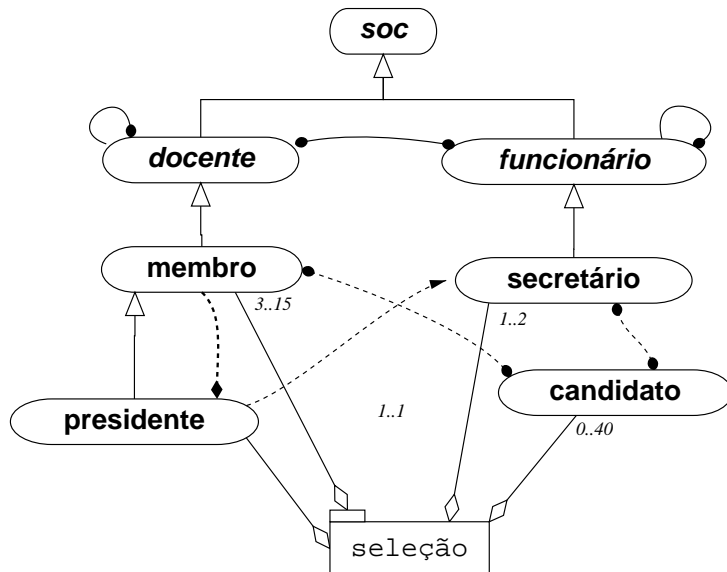
## Exemplo II



## Exemplo II

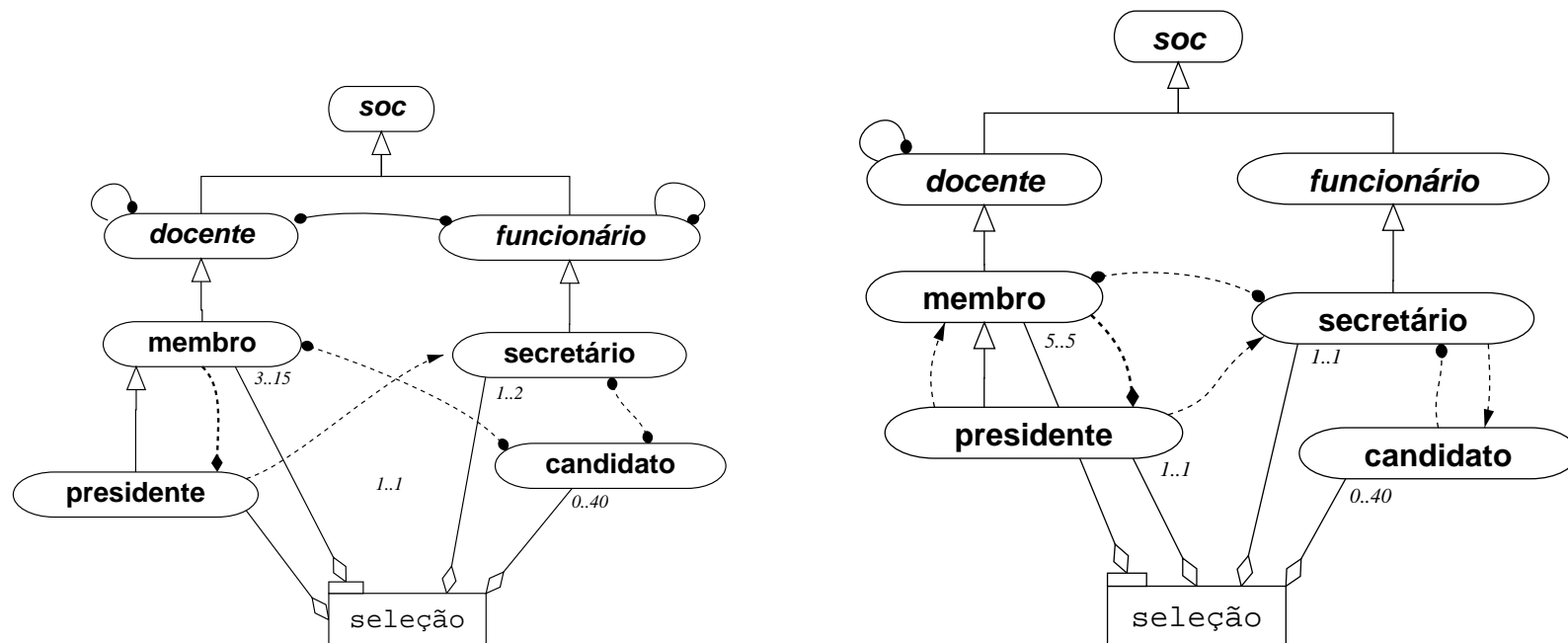


## Exemplo II



- **Estrutura:** o que os agentes podem fazer.

## Exemplo II



- **Estrutura: o que os agentes podem fazer.**
  - ★ Quanto maior o conjunto de possibilidades, maior a autonomia, maior a flexibilidade de adaptação, maior o esforço de raciocínio do agente.

## Especificação Funcional

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

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

- Os planos determinam a **coordenação** na realização das metas.
- As missões ligam os agentes aos planos.



# Metas globais

Um meta global representa um estado do mundo que é desejado pelo SMA.

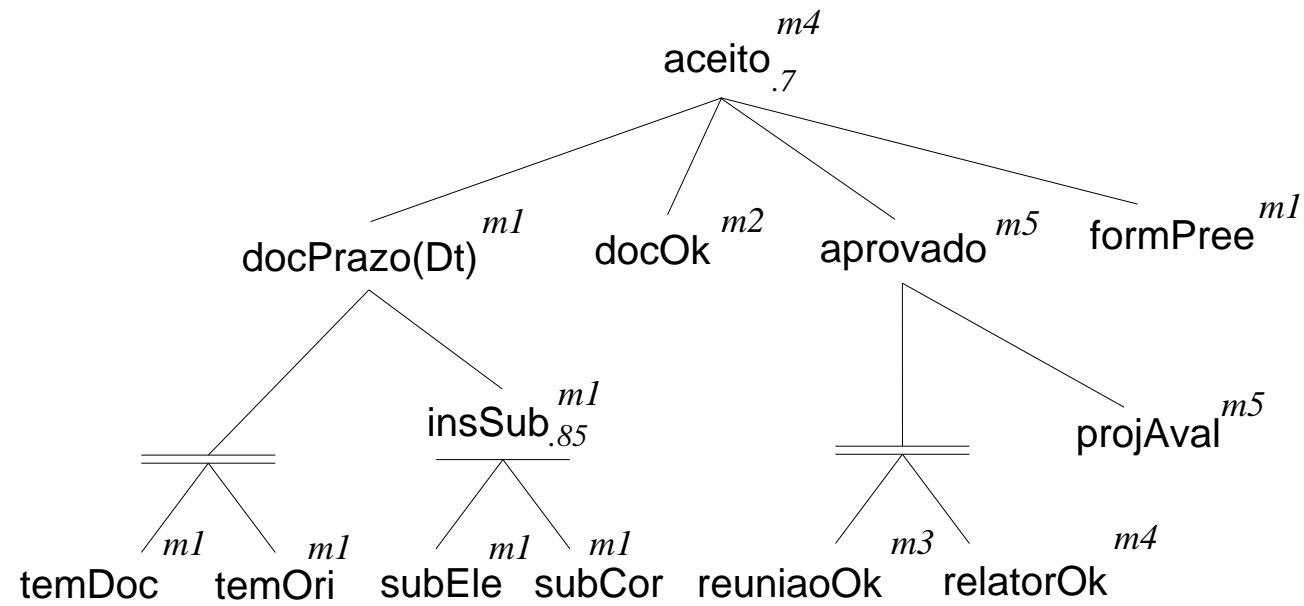
A cada meta é associado uma combinação de três valores que indicam

1. seu **nível de de satisfatibilidade**: indica se a meta já foi alcançada ou não , ou ainda se ela é impossível de ser alcançada;
2. seu **nível de alocação**: indica se já existe ou não algum agente comprometido a satisfazer a meta;
3. seu **nível de ativação**: indica se as pré-condições necessárias para que a meta seja satisfeita estão presentes. Por exemplo, a meta “entregar a documentação” não é permitida até que a documentação esteja toda preparada.

# Missões - atribuições dos agentes

meta	descrição	missão
aceito	o/a candidato/a é aceito no programa de pós-graduação	$m_4$
docPrazo	a documentação é recebida no prazo	$m_1$
docOk	a documentação está correta	$m_2$
aprovado	o/a candidato é aprovado pela comissão	$m_5$
temDoc	o/a candidato/a tem toda a documentação necessária	$m_1$
temOri	o/a candidato/a tem um/a orientador/a	$m_1$
insSub	a inscrição está submetida	$m_1$
subEle	submissão eletrônica	$m_1$
subCor	submissão por correio	$m_1$
reuniaoOk	uma reunião está marcada	$m_3$
relatorOk	um relator está indicado	$m_4$
projAval	o projeto do candidato é avaliado	$m_5$
formPreen	o formulário de matrícula preenchido é recebido	$m_1$

# Planos



## Legenda

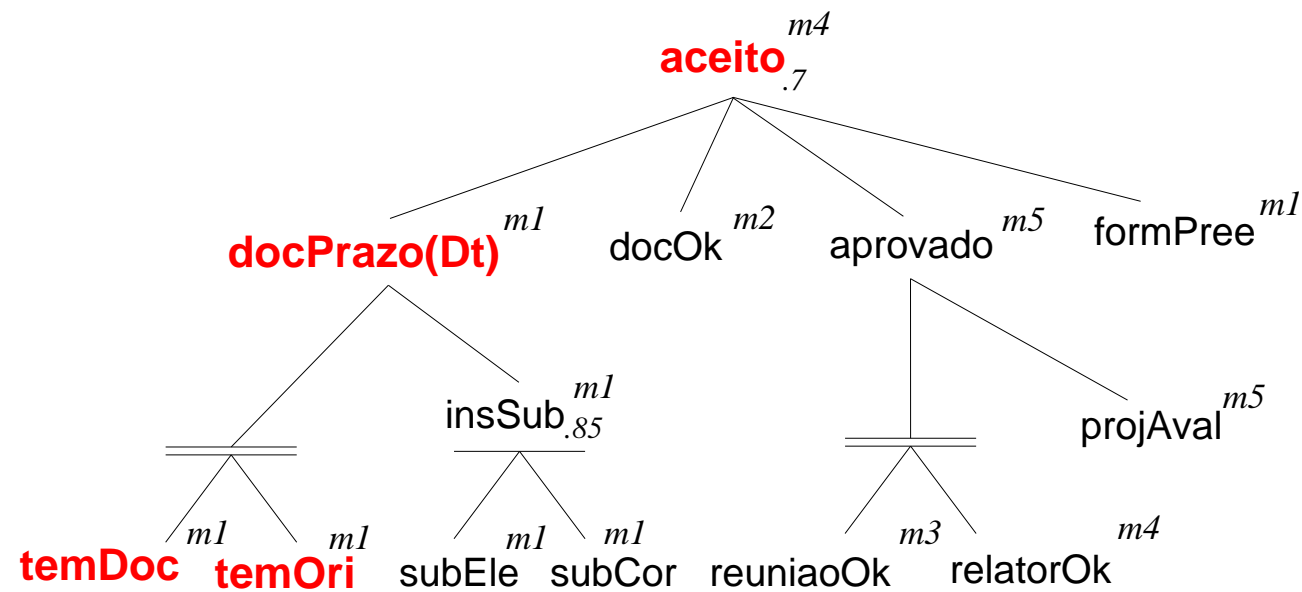
*missões*  
**meta**  
*taxa de sucesso*

seqüência

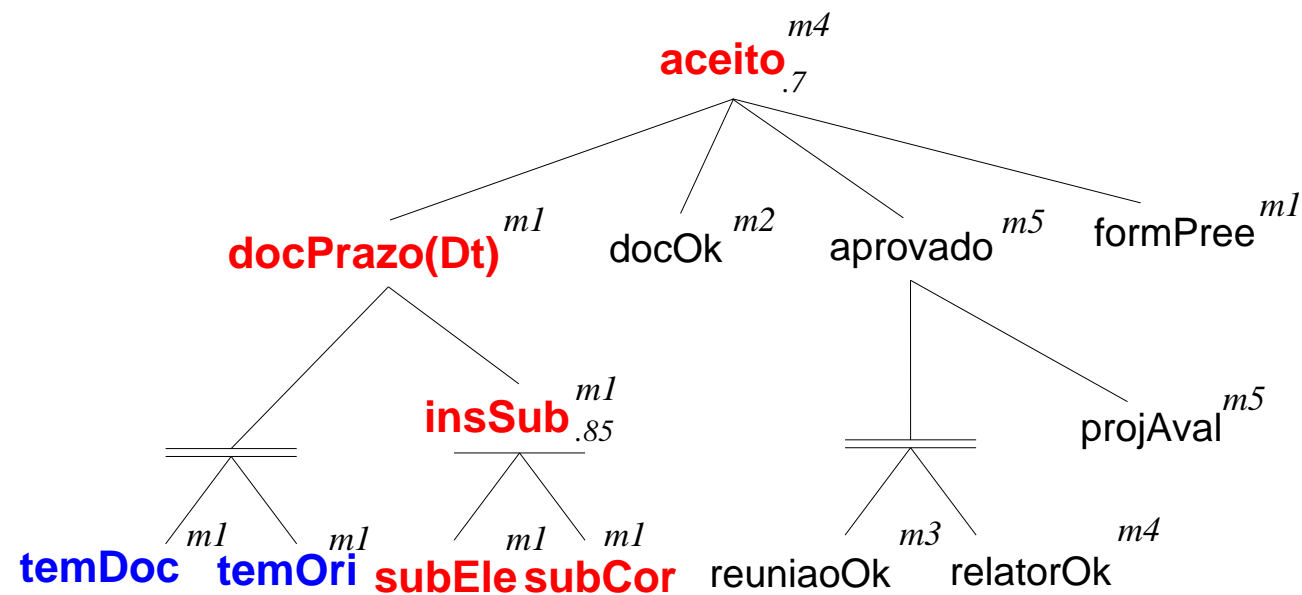
escolha

paralelismo

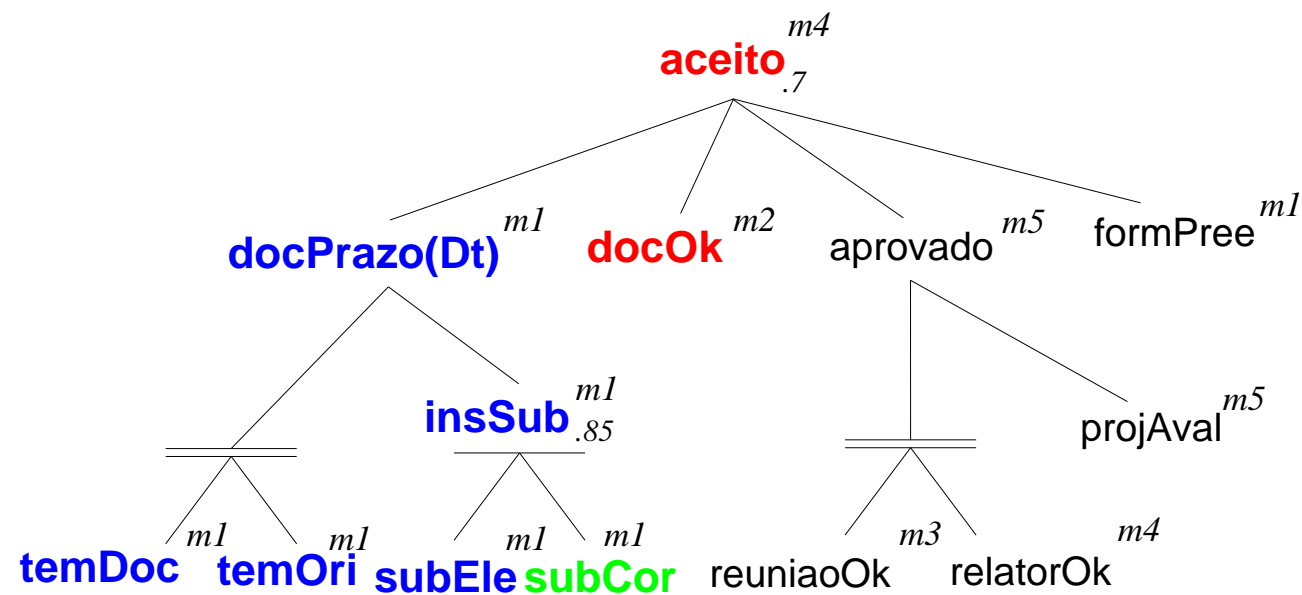
## Planos - coordenação entre os agentes



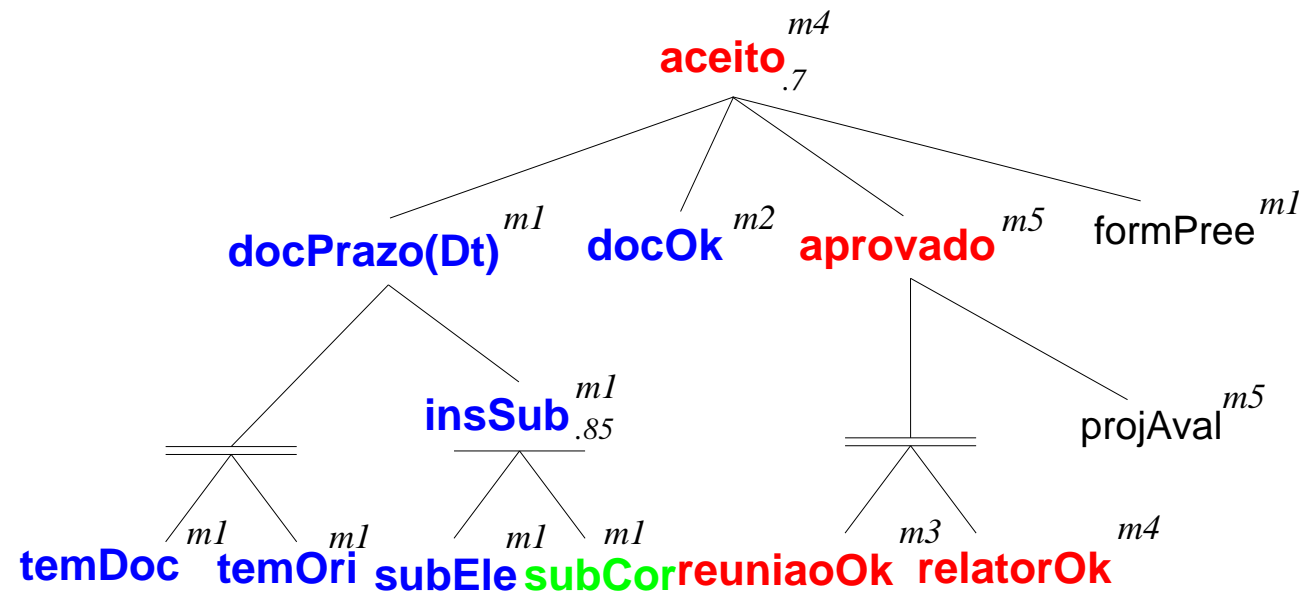
## Planos - coordenação entre os agentes



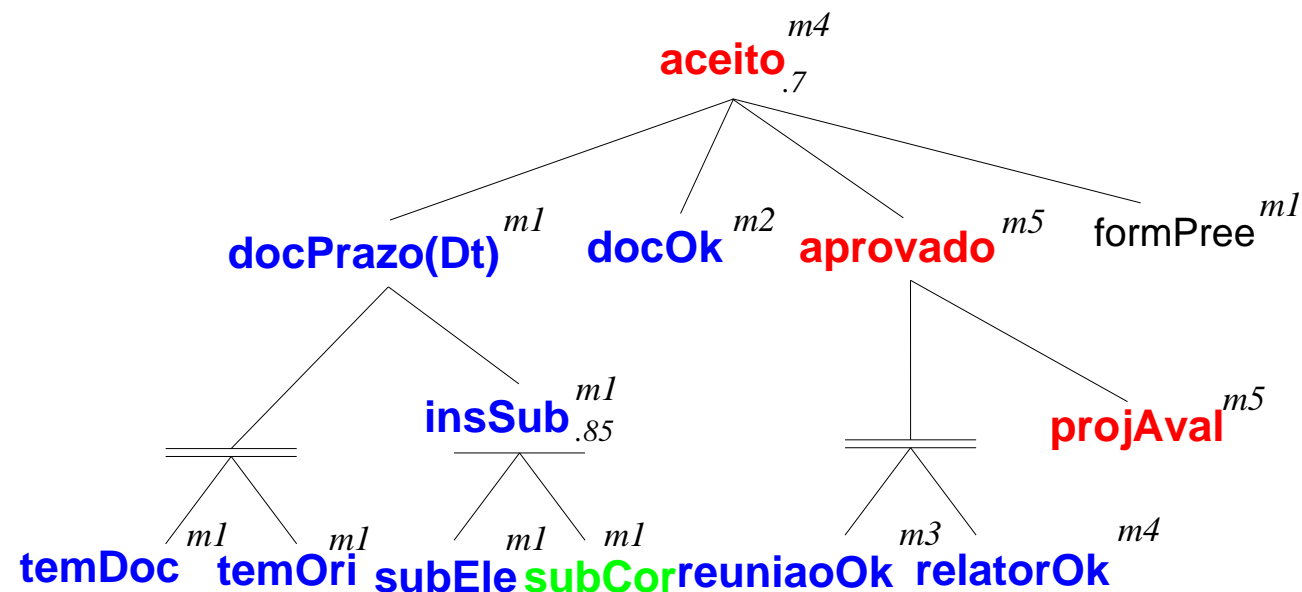
## Planos - coordenação entre os agentes



## Planos - coordenação entre os agentes

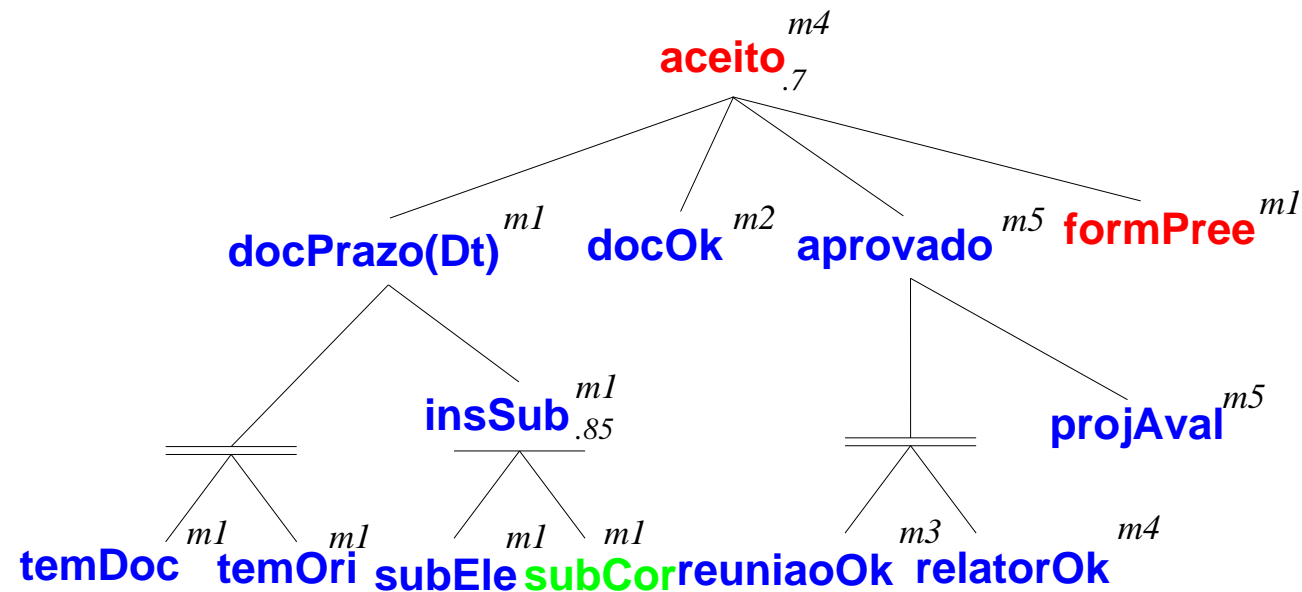


## Planos - coordenação entre os agentes

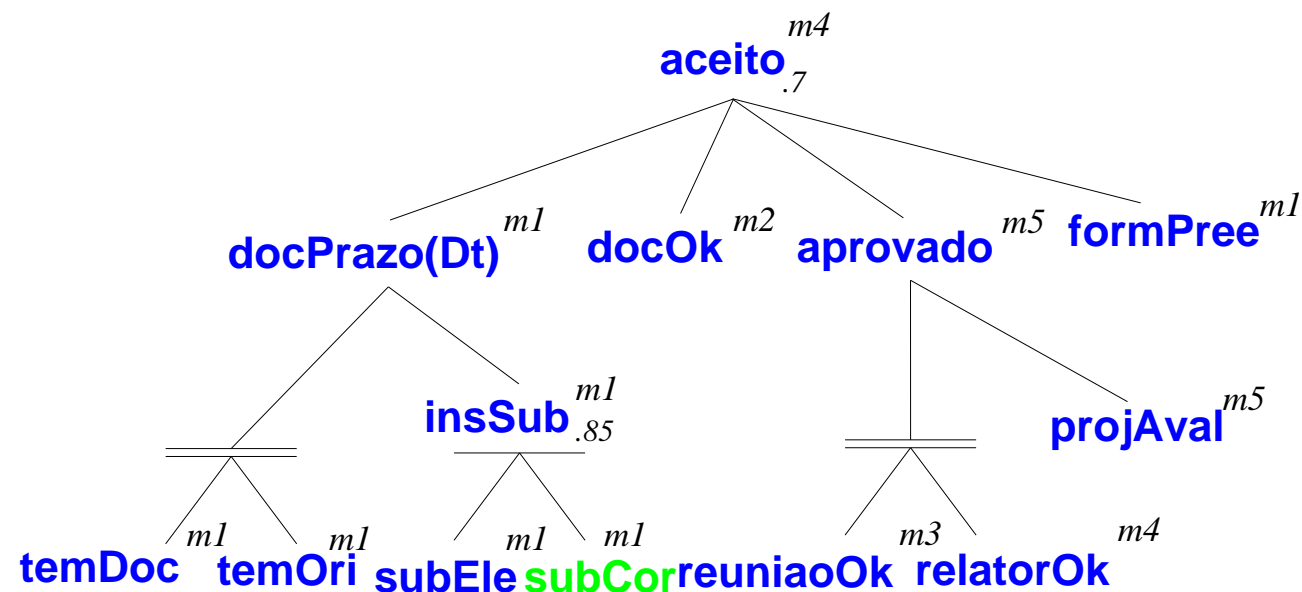




## Planos - **coordenação** entre os agentes



## Planos - **coordenação** entre os agentes



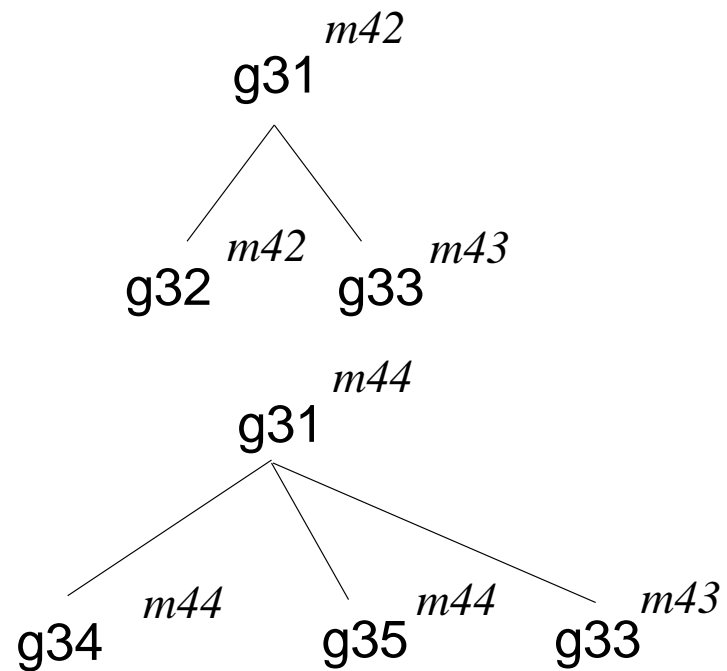
## Esquema social

Um esquema é formado por

- missões
- planos
- cardinalidade das missões (um agente deve assumir a missão de avaliar o projeto do candidato)

Um conjunto de esquemas mais uma relação de preferência entre as missões formam a **especificação funcional**.

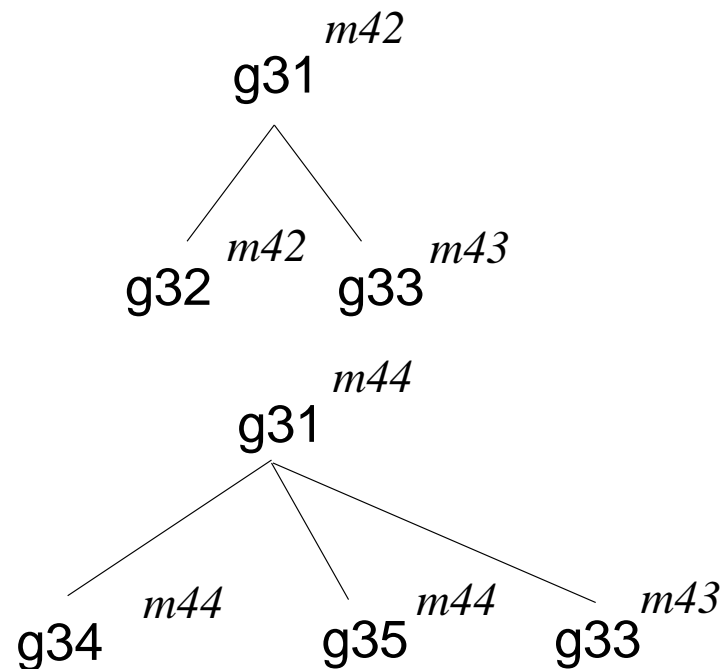
## Exemplo



meta	descrição
$g_{31}$	realizar uma prova
$g_{32}$	preparar a prova
$g_{33}$	responder a prova
$g_{34}$	encontrar uma prova já aplicada
$g_{35}$	alterar o texto da prova

$$m_{42} \prec m_{44}$$

## Exemplo



meta	descrição
$g_{31}$	realizar uma prova
$g_{32}$	preparar a prova
$g_{33}$	responder a prova
$g_{34}$	encontrar uma prova já aplicada
$g_{35}$	alterar o texto da prova

$$m_{42} \prec m_{44}$$

A especificação funcional determina **como** os agentes **podem** alcançar o objetivo do sistema.

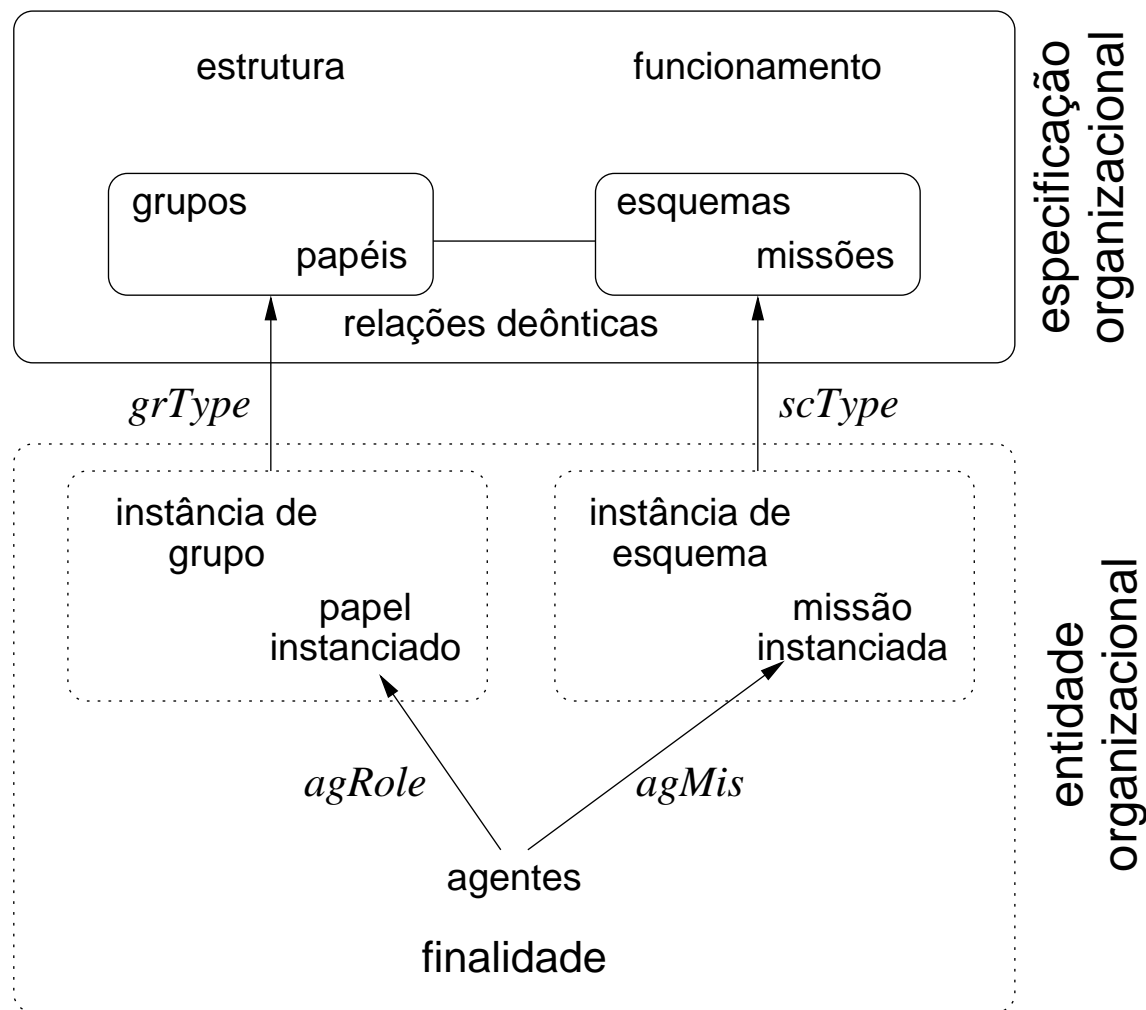
# Especificação Deontica

A relação entre estrutura e funcionamento é estabelecida no nível individual: papel  $\rightarrow$  missão

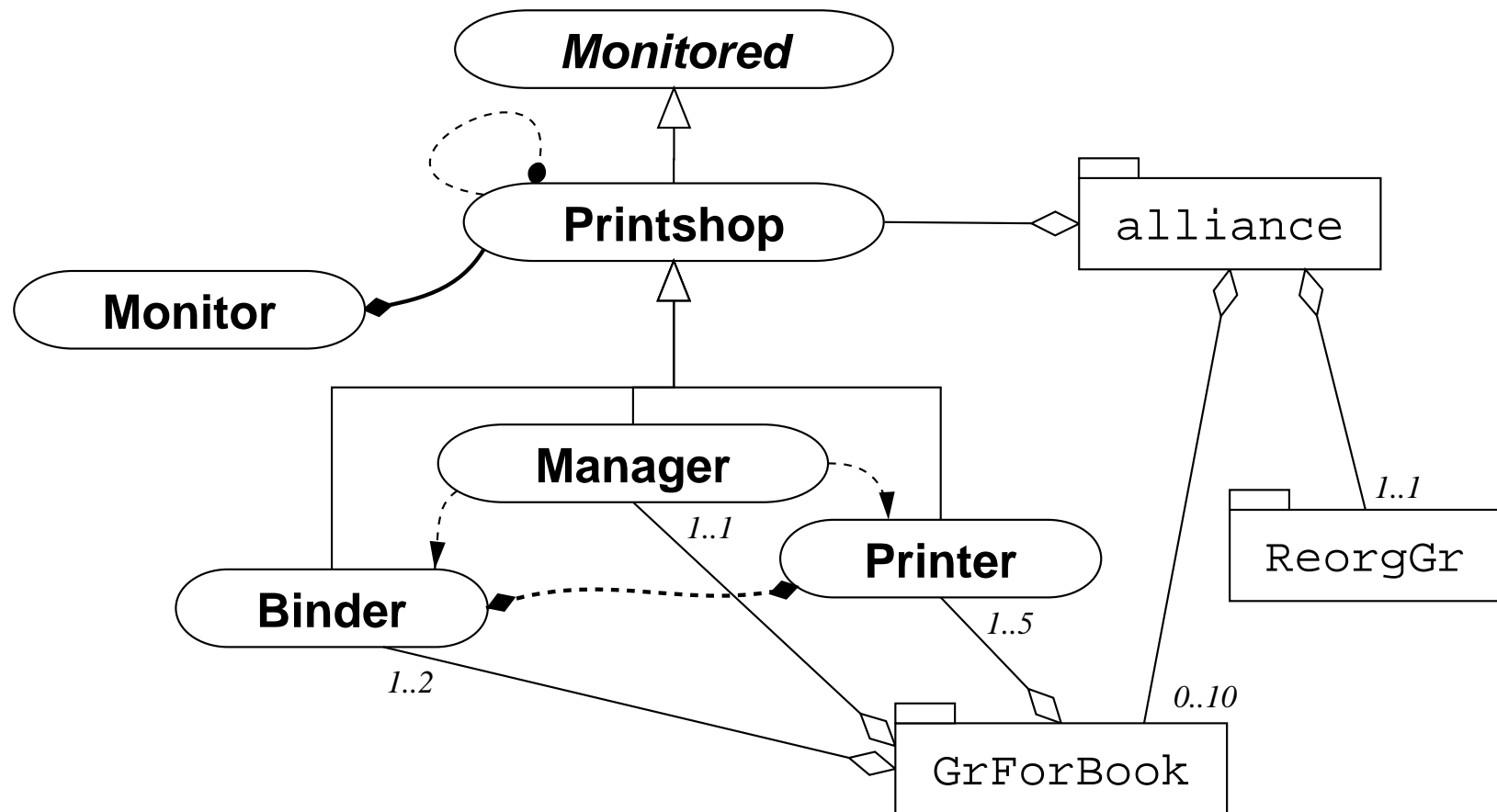
- tipo: permissão ou obrigação
- restrições temporais

Papel	Relação deontica	Missão	Restrições temporais
presidente	<i>per</i>	<i>schingresso.m4</i>	[ <i>feb/02 feb/28</i> ]
secretario	<i>per</i>	<i>schingresso.m2</i>	<i>Any</i>
secretario	<i>per</i>	<i>schingresso.m3</i>	<i>Any</i>
membro	<i>per</i>	<i>schingresso.m5</i>	<i>Any</i>
candidato	<i>per</i>	<i>schingresso.m1</i>	<i>Any</i>
aluno	<i>obl</i>	<i>schprova1.m43</i>	<i>Any</i>
professor	<i>obl</i>	<i>schprova1.m42</i>	<i>periodic 3</i>

# Entidade Organizacional

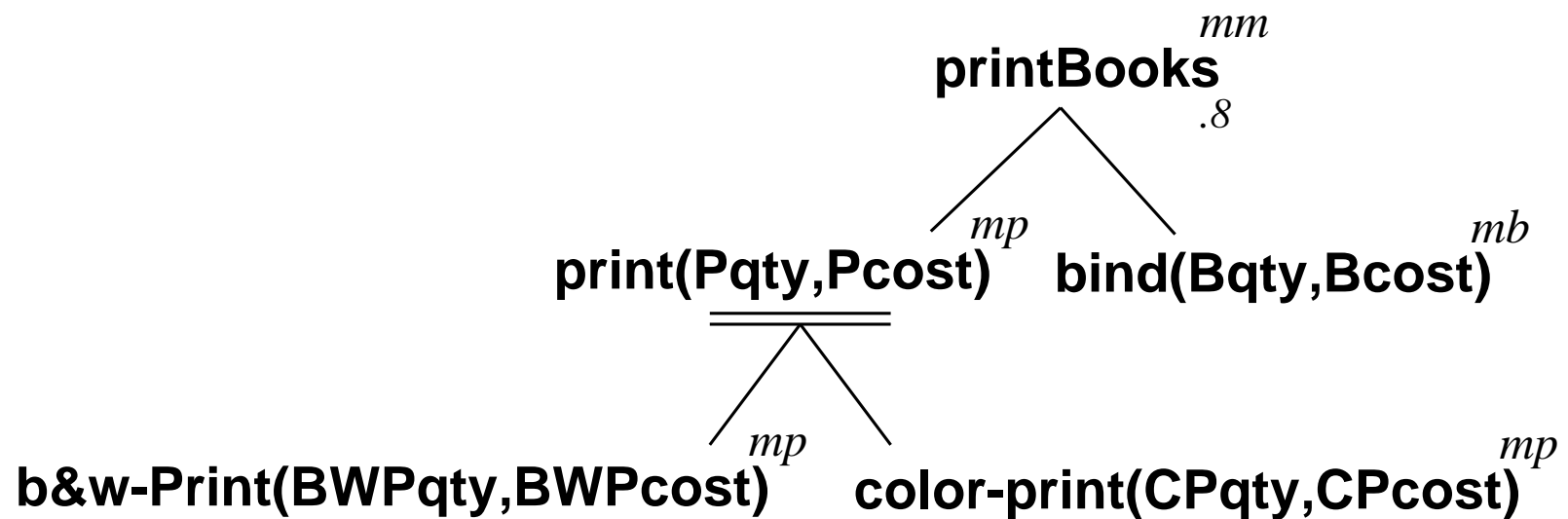


## Exemplo B2B: estrutura

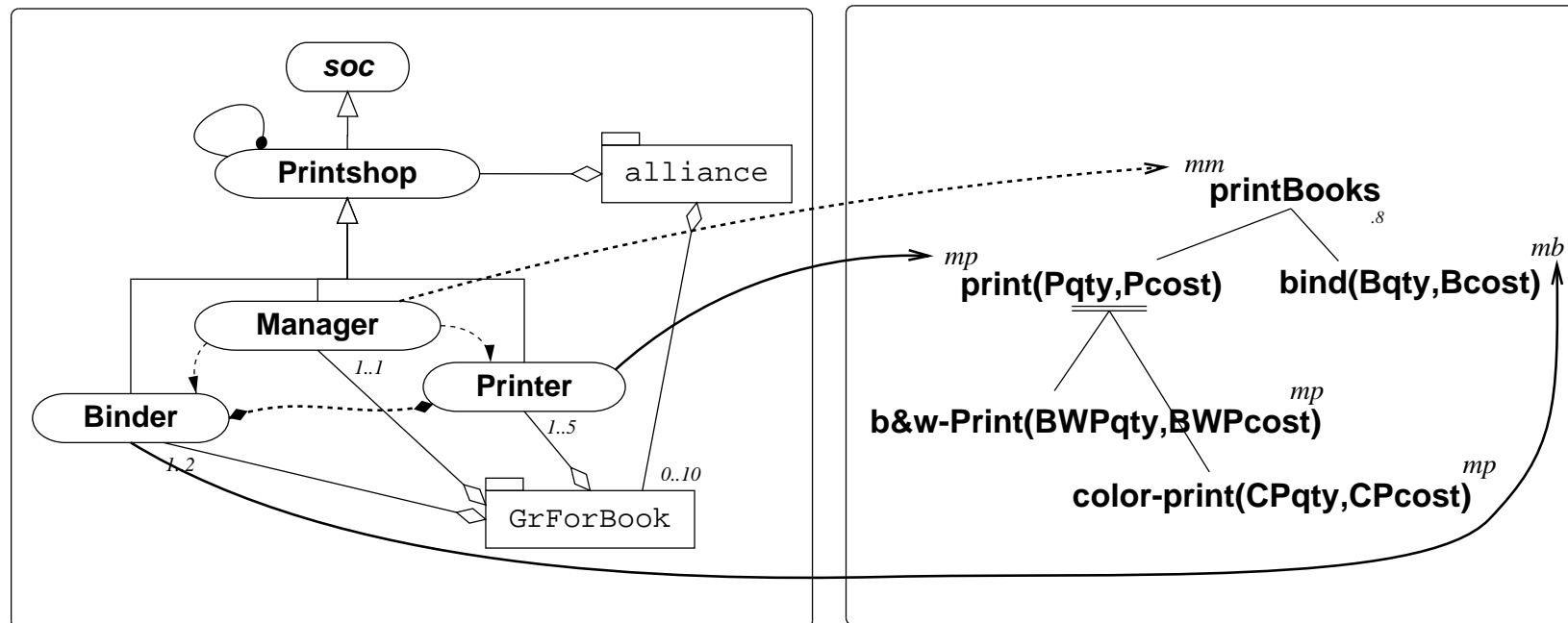




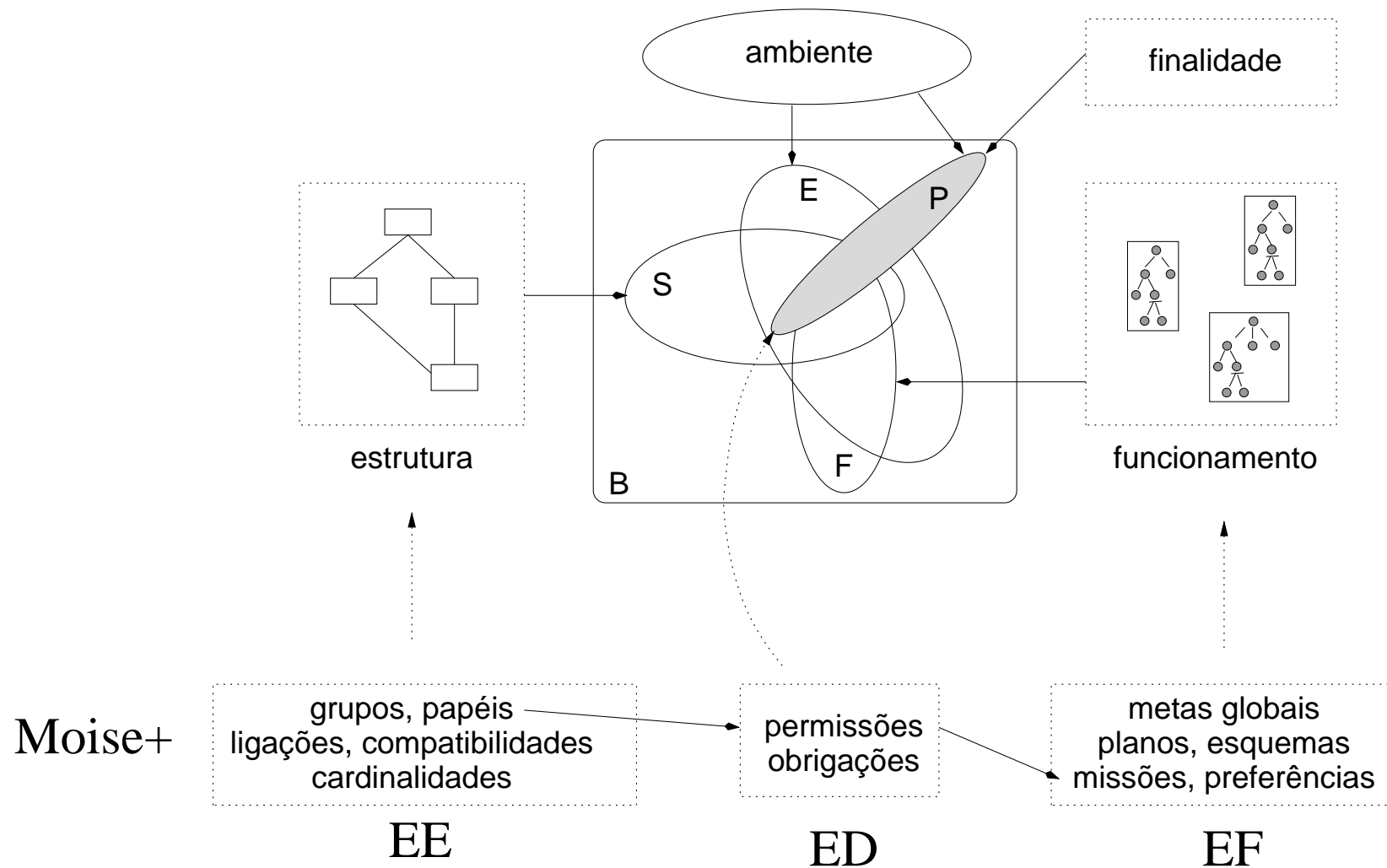
## Exemplo **B2B**: funcionamento



## Exemplo B2B: obrigações



# Visão geral do $\mathcal{MOISE}^+$



## Resumo

O modelo  $\mathcal{MOISE}^+$  apresenta uma concepção de como a organização contribui para a finalidade do SMA: restringindo os comportamentos dos agentes através de uma estrutura de ligações entre papéis e um conjunto de planos globais.

- i)* Não se pretende especificar os agentes e nem estabelecer qualquer requisito para eles.
- ii)* Permite descrever uma organização em três dimensões: estrutural, funcional e deôntica. Em cada dimensão é possível definir um escopo individual (como papéis e missões) e um escopo coletivo (como grupos e esquemas).

- iii)* A noção de papel do modelo  $\mathcal{MOISE}^+$  permite conceber um papel com as seguintes características:
  - a) coletivo
  - b) normativo
- iv)* Permite estabelecer restrições sobre a dinâmica de formação da entidade através da noção de “bem formado” dos grupos e missões.
- v)* Torna explícita a finalidade do sistema.
- vi)* Permite a especificação dos aspectos organizacionais (papéis, planos, ...) de forma **independente**.

## Roteiro

- Modelos organizacionais
- Modelo  $\mathcal{MOISE}^+$
- **Programação orientada a organizações**

## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.

## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?



## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?
  - ★ Com que linguagem se descreve a organização do SMA?

## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?
  - ★ Com que linguagem se descreve a organização do SMA?
  - ★ Onde as informações organizacionais ficam armazenadas (em um único lugar ou descentralizado)?

## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?
  - ★ Com que linguagem se descreve a organização do SMA?
  - ★ Onde as informações organizacionais ficam armazenadas (em um único lugar ou descentralizado)?
  - ★ Como o modelo interfere no comportamento dos agentes?

## Motivação para uma arquitetura organizacional

- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?
  - ★ Com que linguagem se descreve a organização do SMA?
  - ★ Onde as informações organizacionais ficam armazenadas (em um único lugar ou descentralizado)?
  - ★ Como o modelo interfere no comportamento dos agentes?
  - ★ Como se dá a coordenação dos agentes envolvidos na execução de um esquema?

## Motivação para uma arquitetura organizacional

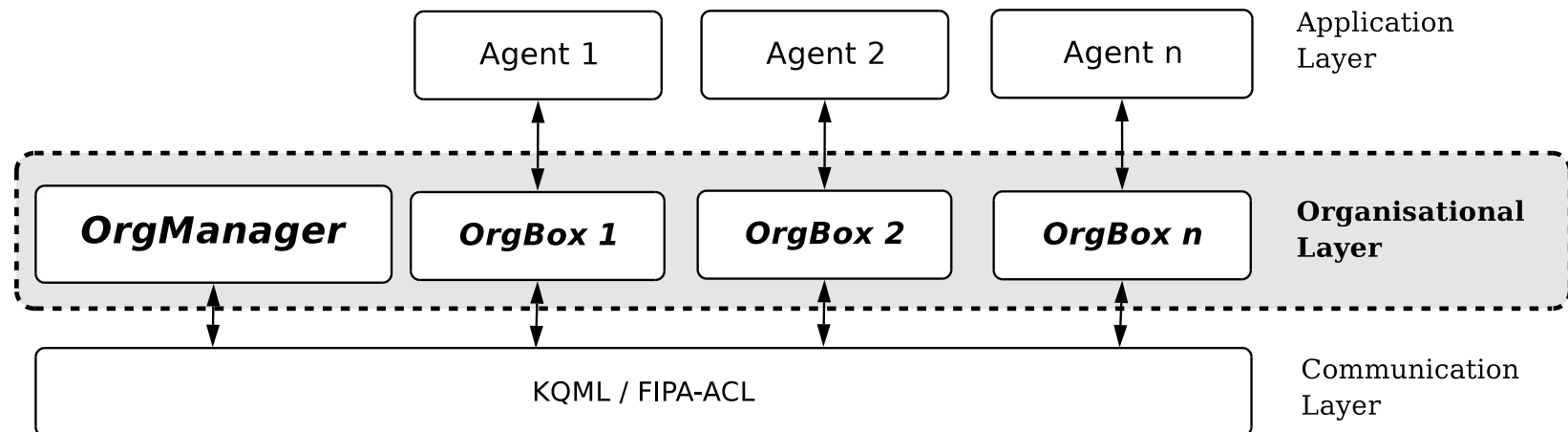
- Enquanto o modelo determina **o que** é uma organização, a arquitetura determina **como** um SMA organizado funciona.
  - ★ Como funciona uma sociedade que segue uma organização?
  - ★ Com que linguagem se descreve a organização do SMA?
  - ★ Onde as informações organizacionais ficam armazenadas (em um único lugar ou descentralizado)?
  - ★ Como o modelo interfere no comportamento dos agentes?
  - ★ Como se dá a coordenação dos agentes envolvidos na execução de um esquema?

# Abordagens para uma arquitetura organizacional

- Centrada nos agentes [[Castelfranchi et al., 1999](#), [Ossowski, 1999](#)]
  - ★ Os agentes são programados de tal forma que “naturalmente” seguem a organização corrente.
  - ★ Os agentes tem recursos de deliberação que lhes permitem raciocinar sobre a organização corrente.
  - ★ Abordagem inadequada para sistemas abertos.
- Centrada no sistema (**instituições**)
  - ★ O ambiente/infraestrutura do sistema verifica se os agentes estão seguindo a organização corrente.
  - ★ Exemplos:
    - \* AMELI [[Esteva et al., 2004](#)] (baseado no ISLANDER)
    - \* MADKIT [[Gutknecht and Ferber, 2000](#)] (baseado no AGR)
    - \* KARMA [[Pynadath and Tambe, 2003](#)] (baseado no STEAM)

## $\mathcal{S}$ -MOISE<sup>+</sup> [Hübner et al., 2005]

- **Uma** possível arquitetura para o modelo  $\mathcal{M}$ OISE<sup>+</sup>.
- Voltado para sistemas abertos e reorganizáveis.
- Abordagem centrada na organização (baseada em camadas).



## Camada **Organizacional**

- Oferece os serviços de manutenção do estado da entidade organizacional e o controle do cumprimento das regras estabelecidas pela organização.



## Camada **Organizacional**

- Oferece os serviços de manutenção do estado da entidade organizacional e o controle do cumprimento das regras estabelecidas pela organização.
- ★ O **OrgManager** é um agente cuja função é manter o **estado** da entidade consistente, por exemplo, não deixando um agente assumir dois papéis incompatíveis.  
Toda mudança na EnO (entrada de um agente, criação de um grupo, adoção de um papel, ...) deve passar por este agente.

## Camada Organizacional

- Oferece os serviços de manutenção do estado da entidade organizacional e o controle do cumprimento das regras estabelecidas pela organização.
  - ★ O **OrgManager** é um agente cuja função é manter o **estado** da entidade consistente, por exemplo, não deixando um agente assumir dois papéis incompatíveis.

Toda mudança na EnO (entrada de um agente, criação de um grupo, adoção de um papel, ...) deve passar por este agente.
  - ★ O **OrgBox** é uma **interface** que os agentes utilizam para acessar a organização e os demais agentes.

Sempre que um agente deseja realizar uma ação sobre a entidade (se comprometer com uma missão, por exemplo) ou enviar uma mensagem, ele deve solicitar este serviço ao seu OrgBox.

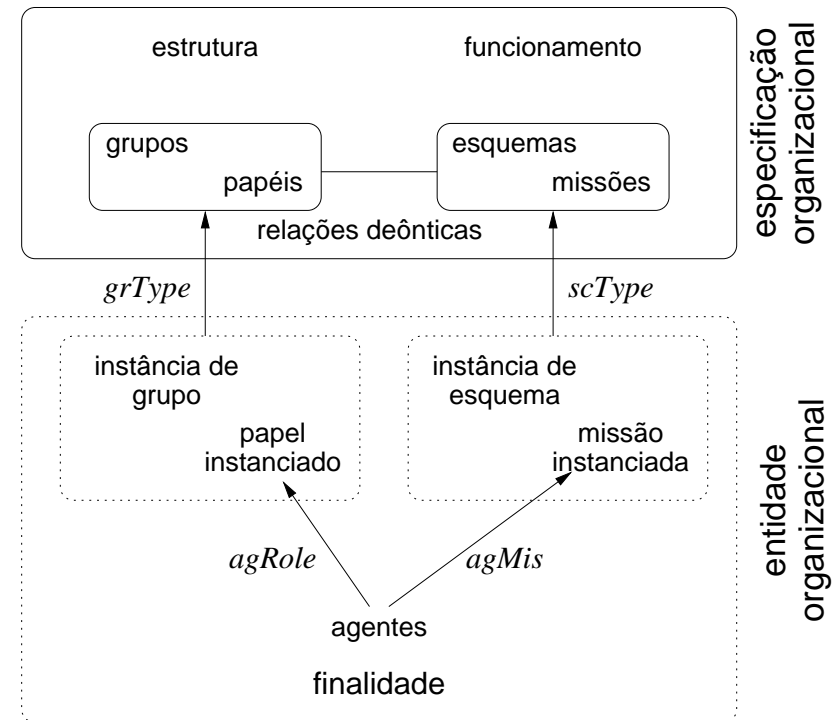
## Eventos organizacionais

- Criação da entidade organizacional.
- Criação ou remoção de um grupo.
- Criação ou finalização de esquemas.
- Alteração no estado de uma meta global.
- Entrada e saída de agentes.
- Adoção e abandono de papel.
- Comprometimento e descomprometimento com missões.

# Criação da entidade organizacional

## Parâmetros

- Finalidade da entidade
- Especificação organizacional (EE, EF, ED)



## Criação de sub-grupo

- Argumentos
  - ★ identificação do novo grupo (exemplo: turmaAlemão),
  - ★ especificação de grupo (exemplo: turma),
  - ★ super-grupo (exemplo: USP, da especificação escola).
- Condições
  - ★ não existir grupo com identificação turmaAlemão,
  - ★ turma ser sub-grupo de escola, e
  - ★ a cardinalidade de turma esta correta.

## Adoção de papéis

- Argumentos
  - ★ identificação do agente (exemplo: Gustavo),
  - ★ identificação do papel (exemplo: aluno), e
  - ★ identificação do grupo (exemplo: turmaAlemão).
- Condições
  - ★ Gustavo pertencer ao sistema,
  - ★ aluno pertencer aos papéis de turma,
  - ★ cardinalidade de aluno estar satisfeita para a turmaAlemão,
  - ★ os papéis atuais de Gustavo são compatíveis com aluno.

## Criação de esquemas

- Argumentos
  - ★ identificação do novo esquema (exemplo: prova1),
  - ★ especificação do esquema (exemplo: prova), e
  - ★ conjunto de grupos responsáveis pelo esquema (exemplo turmaAlemão).
- Condições
  - ★ os grupos existirem na entidade.

# Comprometimento com missões

- Argumentos
  - ★ identificação do agente (exemplo: Gustavo),
  - ★ identificação da missão (exemplo: m42 - preparar prova), e
  - ★ identificação do esquema (exemplo: prova1).
- Condições
  - ★ a cardinalidade da missão m42 no esquema prova1 não é violada,
  - ★ o esquema ainda não terminou, e
  - ★ os papéis de Gustavo nos grupos responsáveis por prova1 **permitem** o compromisso com a missão m42.



## Mudança no **estado** das metas

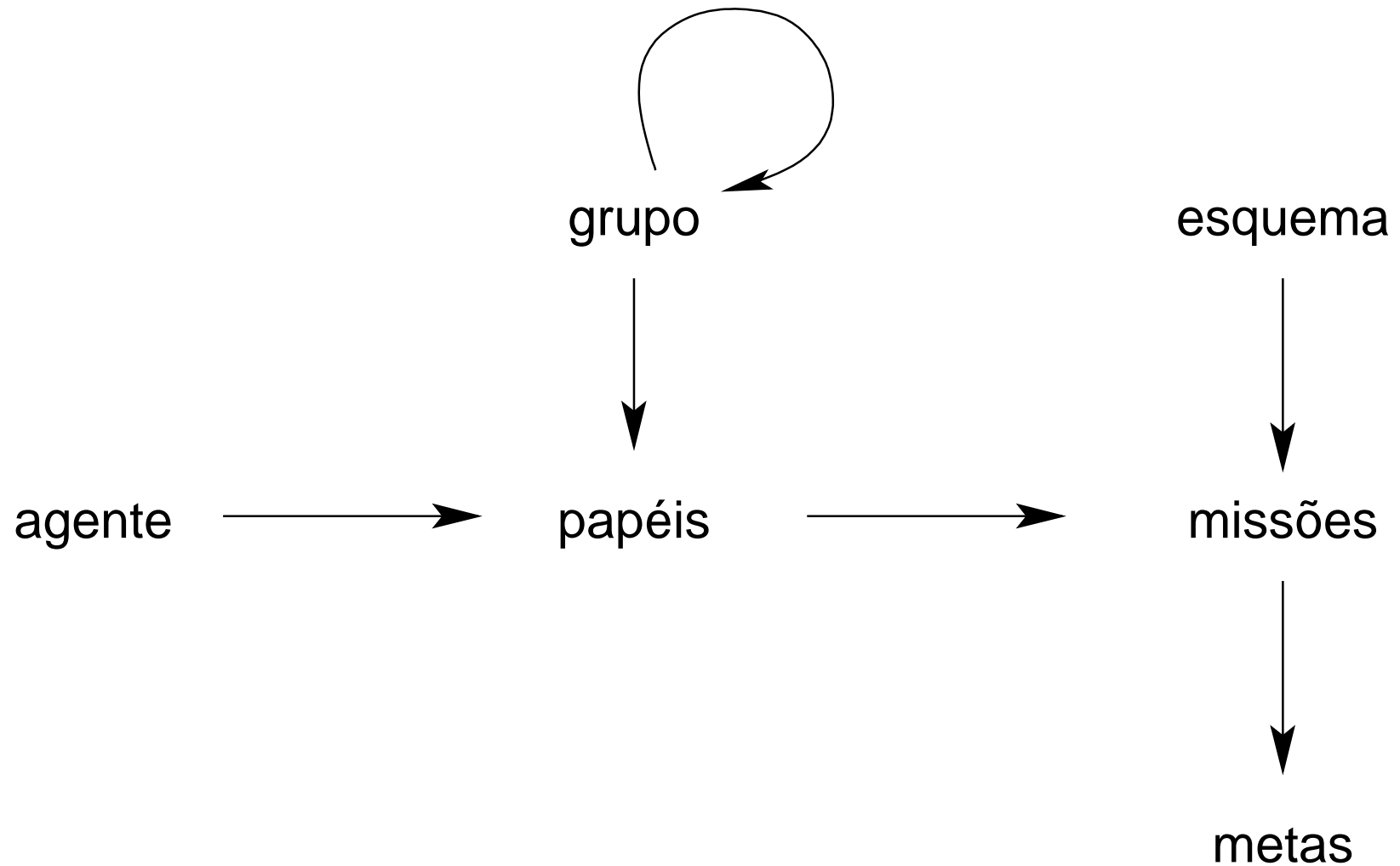
Dos valores de uma meta, somente o nível de **satisfação** é alterado diretamente por eventos organizacionais.

- Argumentos
  - ★ identificação da meta (exemplo: prepararProva), e
  - ★ identificação do esquema (exemplo: prova1).
- Condições
  - ★ a meta é permitida,
  - ★ tem agentes comprometidos com ela, e
  - ★ é possível.

# Algoritmo para determinar o nível de **ativação** de uma meta

```
1 function isPermitted(scheme sch, goal g)
2   if g is the sch root then
3     |   return true;
4   else
5     |   g is in a plan that match " $g_0 = \dots g \dots$ ";
6     |   if g is in a plan that match " $g_0 = \dots g_i , g \dots$ " then
7       |     if gi is already satisfied then
8         |       |   return true;
9         |     else
10        |       |   return false;
11      |   else
12      |     |   return isPermitted(sch, g0);
```

## Dependências para a **remoção** de um grupo (visão geral)



# OrgBox

- Serviços para os agentes (camada de aplicação)
  - ★ Comunicação: o envio e recebimento de mensagens KQML para/de outros agentes.  
Verificação das **ligações de comunicação**.

# OrgBox

- Serviços para os agentes (camada de aplicação)
  - ★ Comunicação: o envio e recebimento de mensagens KQML para/de outros agentes.  
Verificação das **ligações de comunicação**.
  - ★ Geração de eventos organizacionais: o agente pode entrar no sistema, assumir um papel, criar um grupo, ...

# OrgBox

- Serviços para os agentes (camada de aplicação)
  - ★ Comunicação: o envio e recebimento de mensagens KQML para/de outros agentes.  
Verificação das **ligações de comunicação**.
  - ★ Geração de eventos organizacionais: o agente pode entrar no sistema, assumir um papel, criar um grupo, ...
  - ★ Informações de obrigações: o OrgBox mantém o agente informado de quais missões ele é obrigado a se comprometer.

# OrgBox

- Serviços para os agentes (camada de aplicação)
  - ★ Comunicação: o envio e recebimento de mensagens KQML para/de outros agentes.  
Verificação das **ligações de comunicação**.
  - ★ Geração de eventos organizacionais: o agente pode entrar no sistema, assumir um papel, criar um grupo, ...
  - ★ Informações de obrigações: o OrgBox mantém o agente informado de quais missões ele é obrigado a se comprometer.
  - ★ Informações de metas possíveis: o OrgBox mantém o agente de quais são as metas globais que podem ser buscadas

## Algoritmo para obter as **obrigações** de um agente

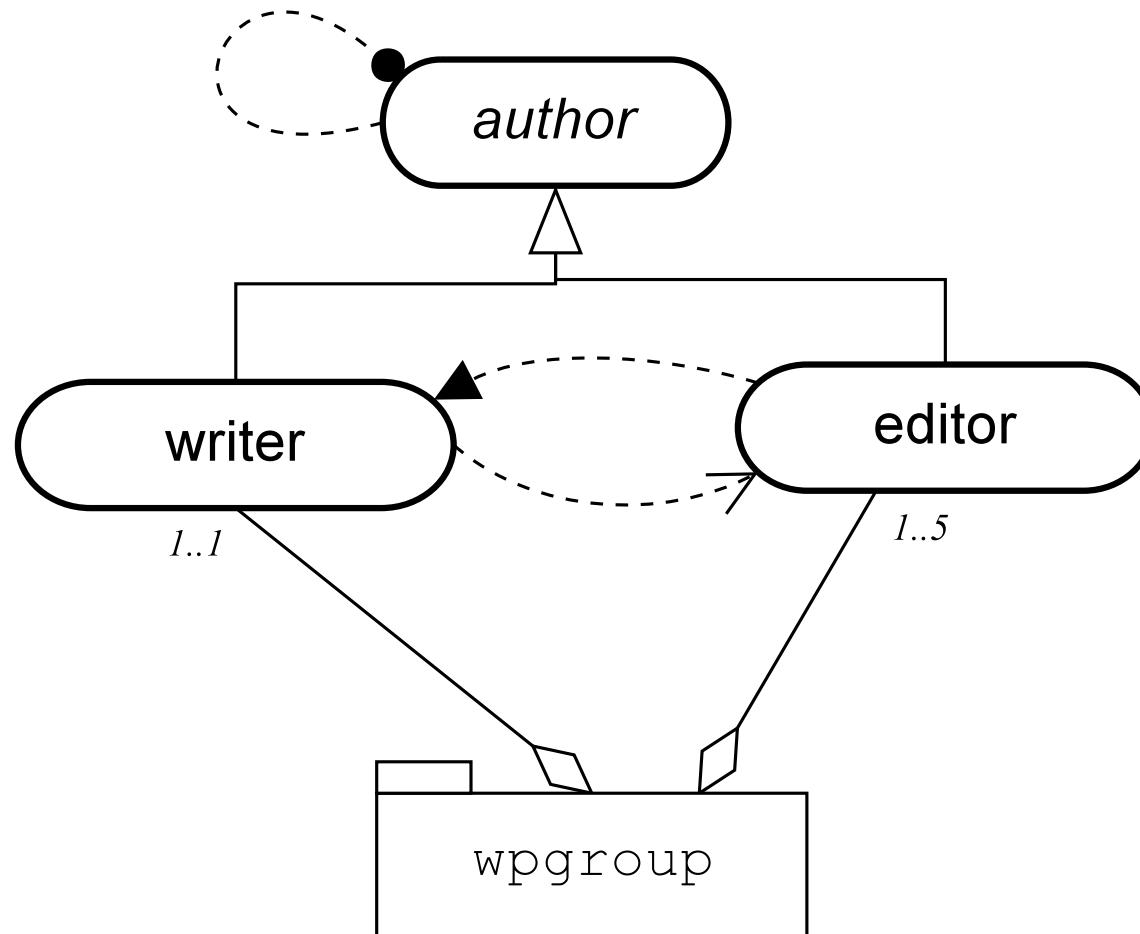
```
1 function getObligatedMissions(agent  $\alpha$ )
2  $all \leftarrow empty\ list$  // list of obligated missions
3 forall role  $\rho$  the agent  $\alpha$  plays do
4    $gr \leftarrow$  the group where  $\rho$  is being played;
5   forall scheme  $si$  that  $gr$  is responsible to do
6     if  $si$  is not finished then
7       forall mission  $m$  in the scheme  $si$  do
8         if  $obligated(\rho, m)$  is in the deontic specification
9           then
10             $all \leftarrow append(all, m);$ 
11 return  $all;$ 
```

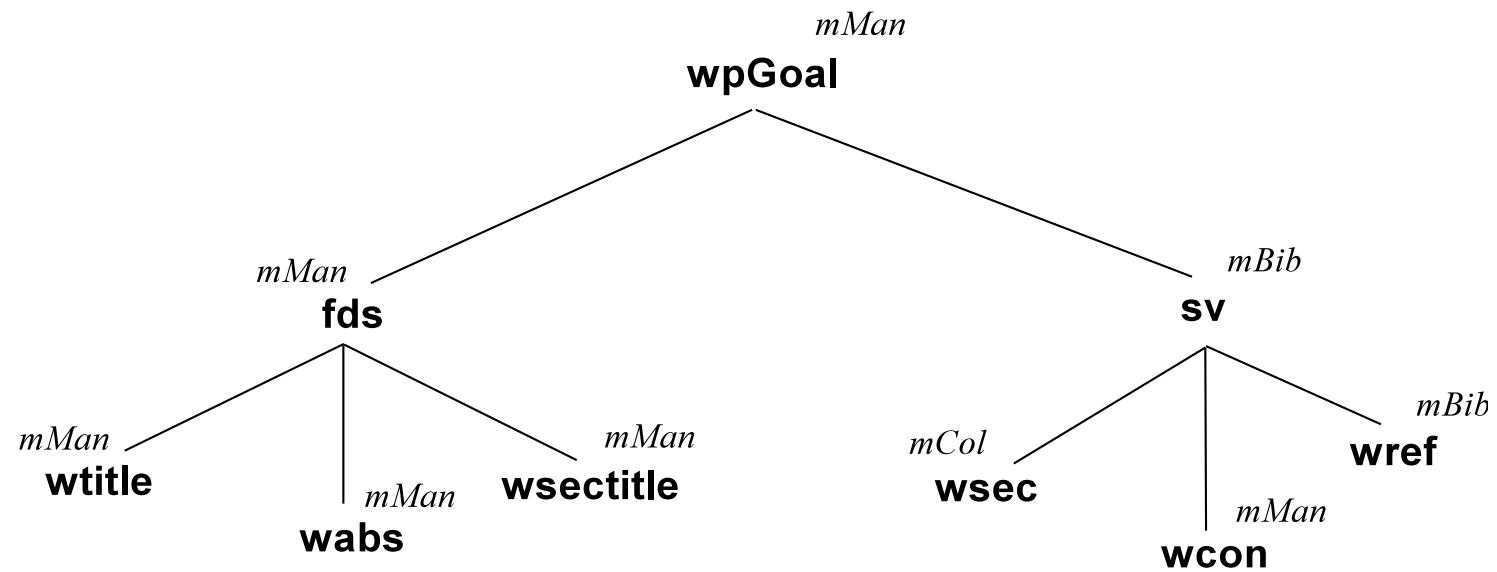


## Algoritmo para obter as metas **possíveis** para um agente

```
function getPossibleGoals(agent  $\alpha$ )  
   $all \leftarrow \langle \rangle$   
  for all missão  $m$  que  $\alpha$  está comprometido do  
     $sch \leftarrow$  o tipo do esquema onde  $m$  foi assumido  
    for all meta  $g$  da missão  $m$  do  
      if  $\neg isSatisfied(g) \wedge isPossible(g) \wedge isPermitted(g)$  then  
        for all  $g_s$  que é super-meta no  $sch$  do  
          if  $\neg isSatisfied(g_s) \wedge \neg isImpossible(g_s)$  then  
             $all \leftarrow append(all, \langle g \rangle)$   
          end if  
        end for  
      end if  
    end for  
  end for  
  ordena  $all$  de acordo com as preferências entre as missões  
  return  $all$ 
```

## Exemplo: escrever um artigo (**OS**)





role	deontic relation	mission	cardinality
editor	<i>per</i>	<i>mMan</i>	1..1
writer	<i>obl</i>	<i>mCol</i>	1..5
writer	<i>obl</i>	<i>mBib</i>	1..1

## Exemplo: escrever um artigo (**OE**)

agent	role	in group	mission
Jaime	editor	wpgroup	<i>mMan</i>
Jomi	writer	wpgroup	<i>mCol</i>
Gustavo	writer	wpgroup	<i>mCol</i>
Gustavo	writer	wpgroup	<i>mBib</i>

## Um **Arquitetura** muito simples para os agentes

```
1 while true do
2    $g \leftarrow \text{choseGoal}()$ ;
3    $p \leftarrow \text{makePlan}(g)$ ;
4   execute( $p$ );
```

```
1 function choseGoal() : Goal;
2 if there is an organisational goal permitted to be achieved then
3   | returns it;
4 if I have no role then
5   | adopts a role;
6   | returns choseGoal();
7 if try to commit to an obligated mission then
8   | returns choseGoal();
9 if try to commit to a permitted mission then
10  | returns choseGoal();
11 if try to uncommit to finished schemes then
12  | for all mission  $m$  I am committed to do
13  |   | if the scheme of  $m$  is already finished then
14  |   |   | uncommit( $m$ );
15 returns no goal;
```

# Programação com $\mathcal{S}$ -MOISE<sup>+</sup>

```
public class JomiAg extends BaseOrgAgent {

    public static void main(String[] args) {
        JomiAg a = new JomiAg();
        if (a.enterSoc("jomi", "writePaperSoc")) {
            a.run();
        } }

    boolean adoptRole() {
        String grTeamId = getOrgBox().getRootGroupInstance( "wpgroup" );
        if (grTeamId != null) {
            getOrgBox().adoptRole("writer", grTeamId);
            return true;
        } else {
            return false;
        } }
}
```

```
protected void plan() {  
    currentPlan = null;  
    if (currentGoal != null) {  
        currentPlan = new ArrayList();  
        currentPlan.add("print("+currentGoal+")");  
    }  
}
```

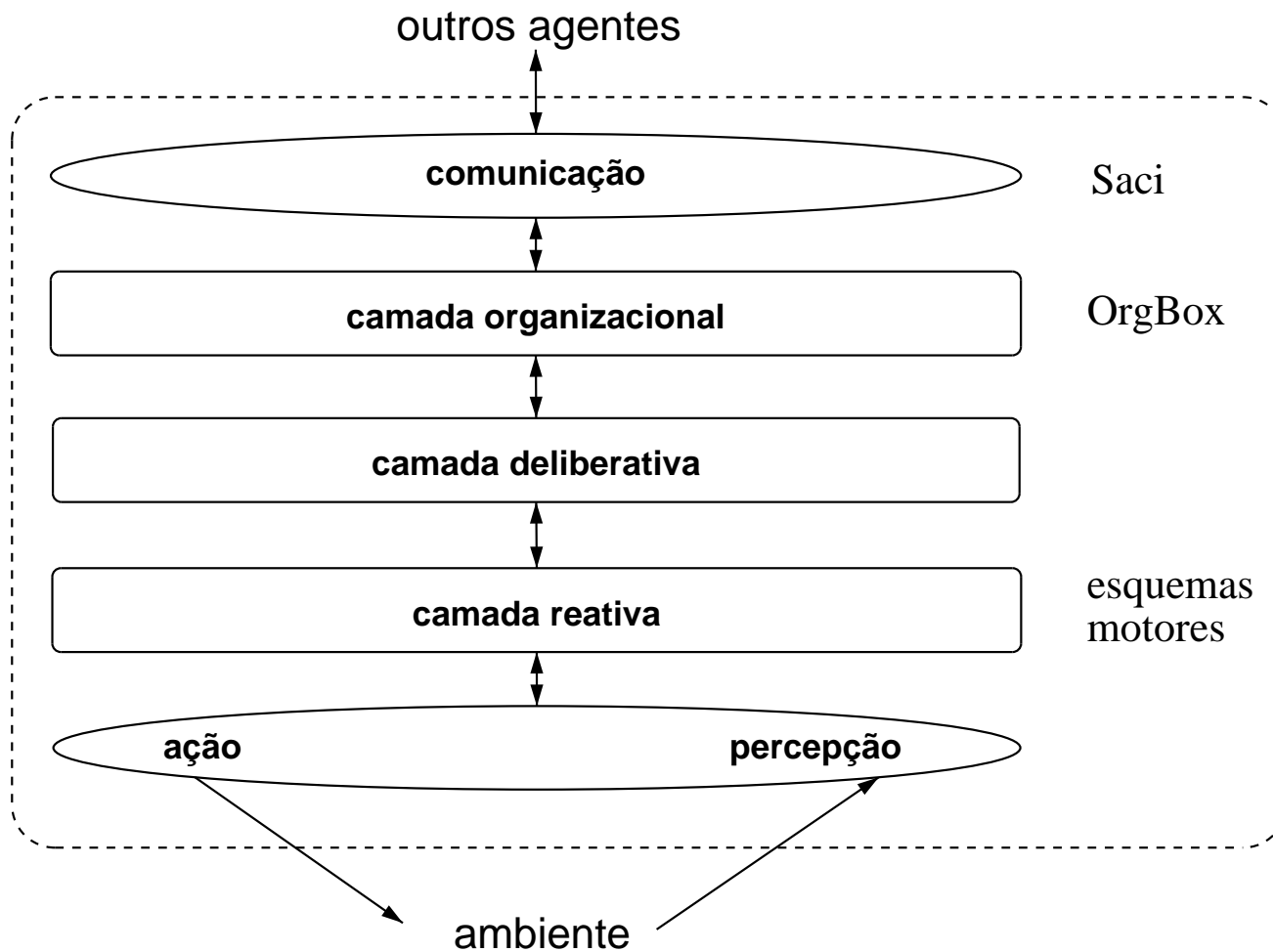
```
protected void executeAct(String action) {  
    if (action.startsWith("print")) {  
        print(action);  
    }  
}  
}
```



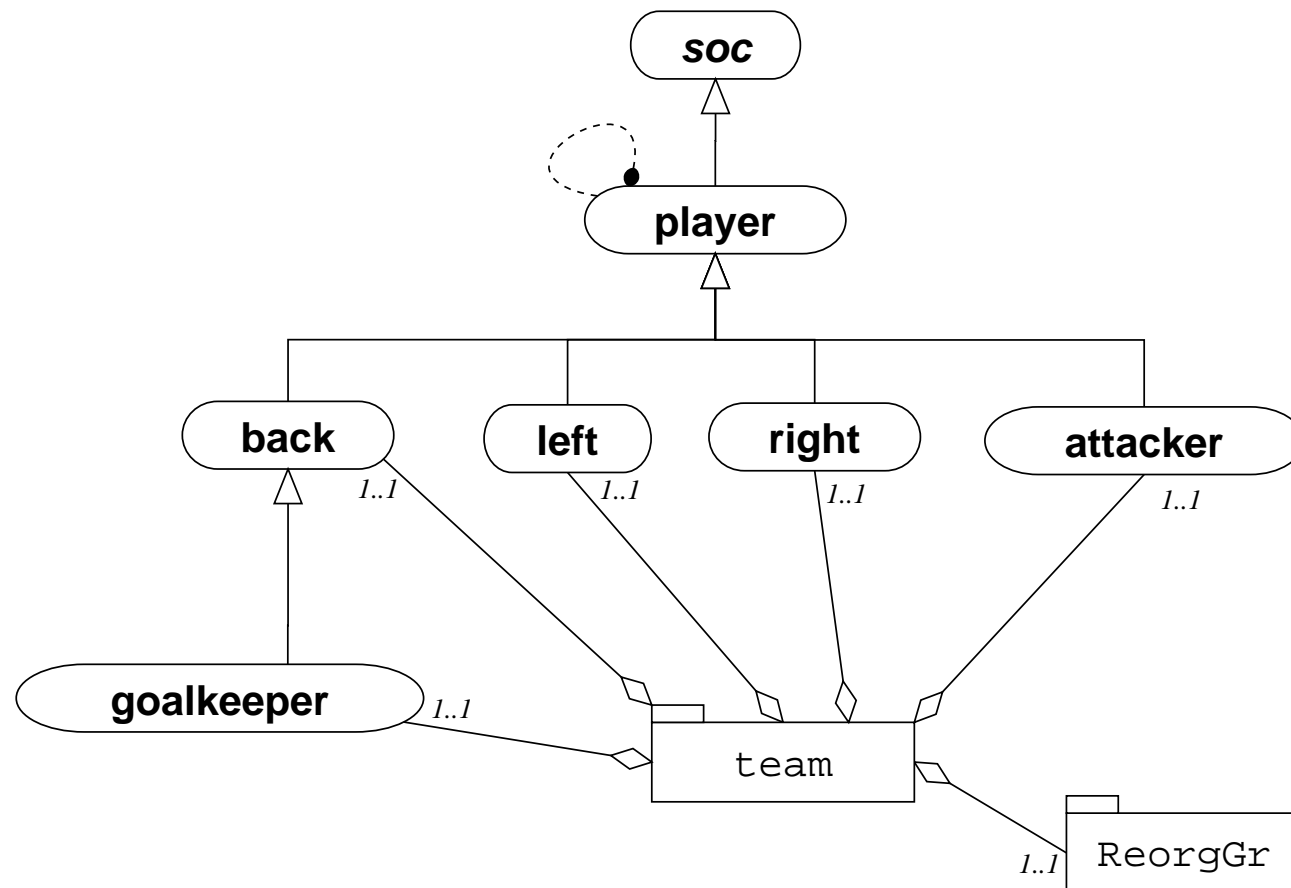
## Exemplo de aplicação desta arquitetura: JOJTEAM

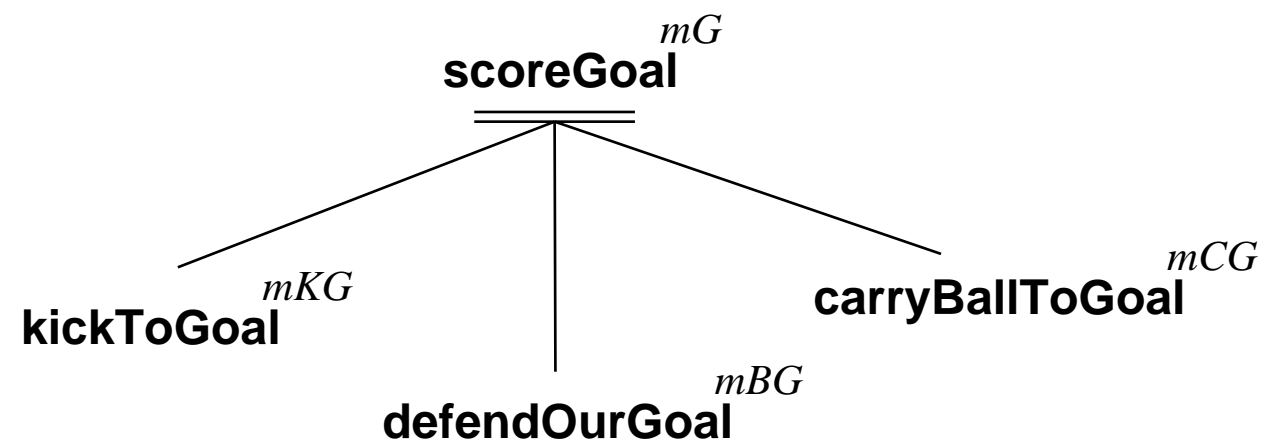
- Especificar a organização de um time de futebol com o  $\text{MOISE}^+$
- Fazer um time que segue uma especificação organizacional
- Apenas mudando a organização o time muda
- **Adaptação organizacional**

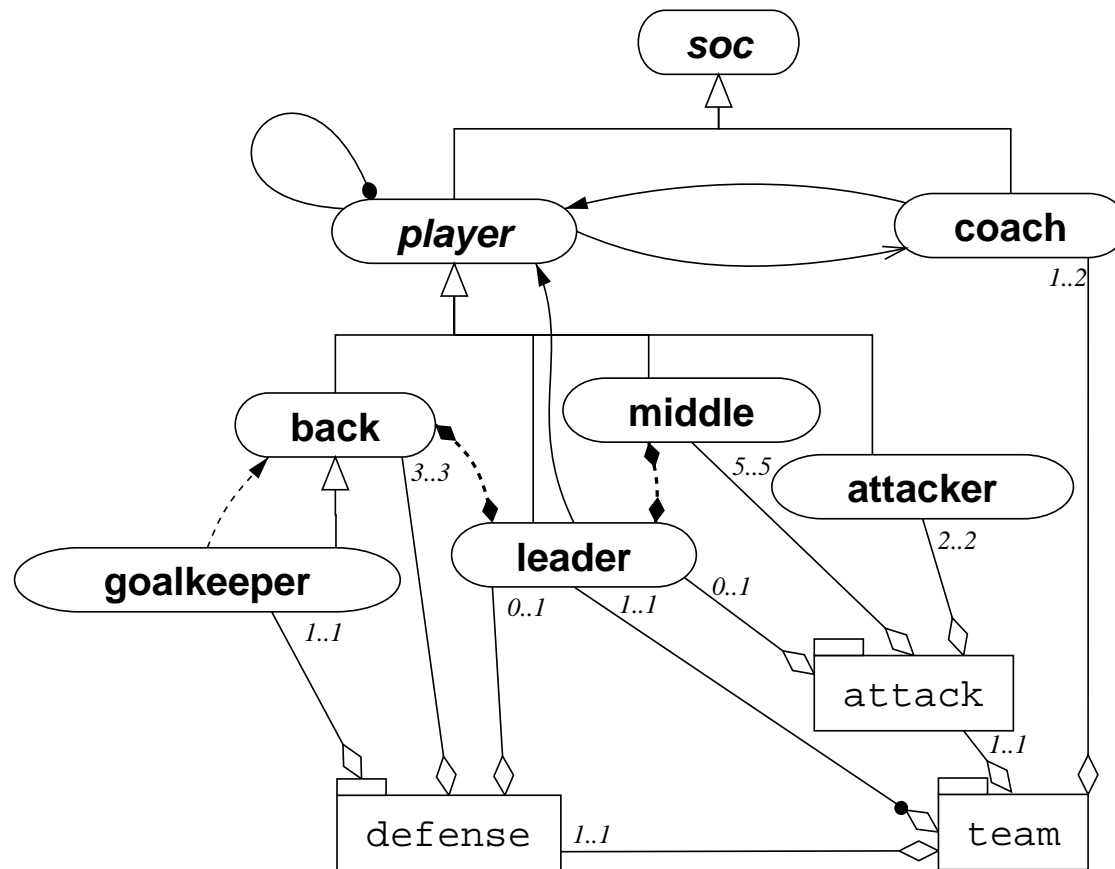
# Arquitetura dos Agentes



## Exemplo de organização







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

Marcos ----- goalkeeper

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

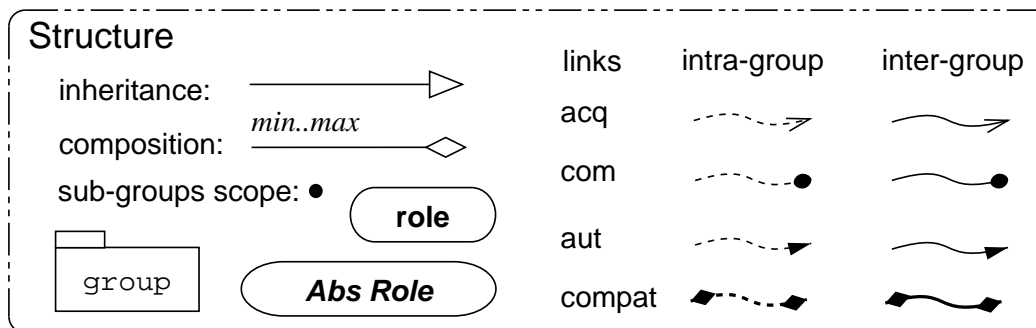
Roque Jr. ----- leader  
Cafu ----- leader

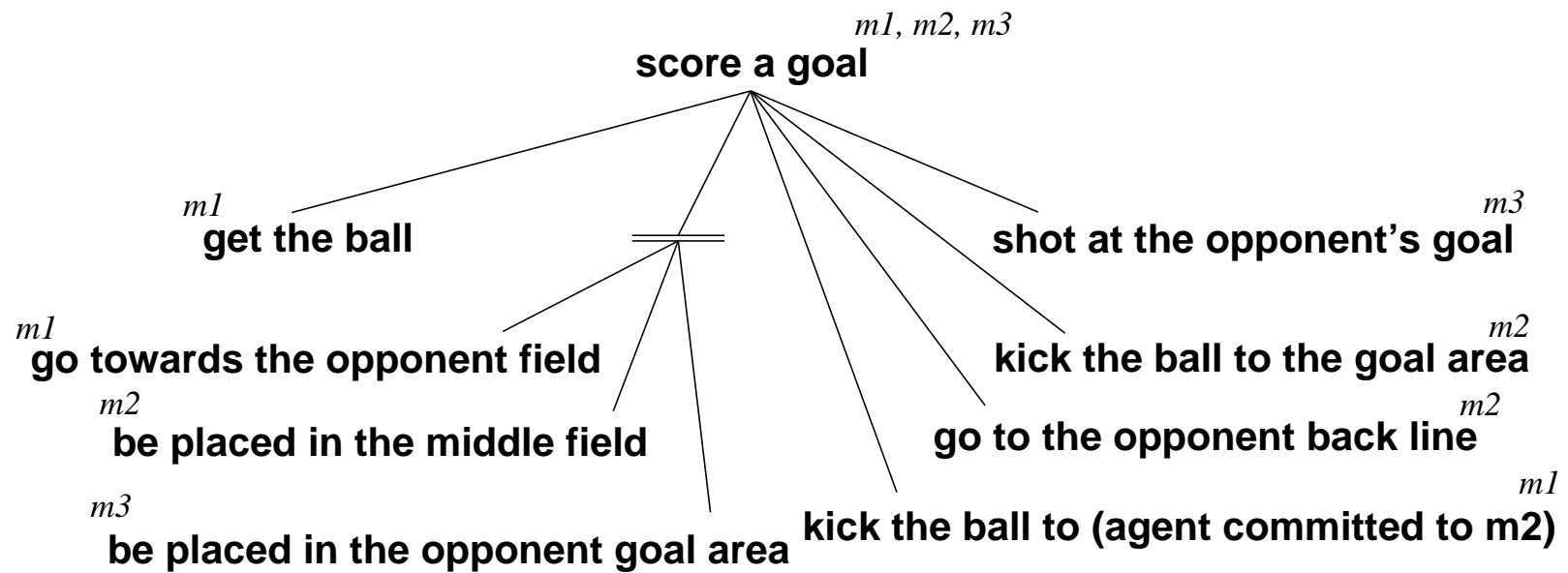
Gilberto Silva ----- middle  
Juninho ----- middle

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

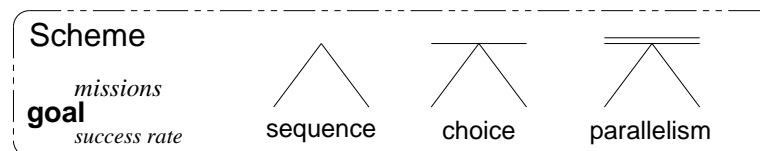
Ronaldo ----- attacker  
Rivaldo ----- attacker

Key





### Key



### Organizational Entity



# Resumo

- $\mathcal{S}$ - $\text{MOISE}^+$ : uma arquitetura para SMA com organização **tipo OC** (centrado na organização com agentes que raciocinam sobre sua organização).

Disponível em

★ <http://www.lti.pcs.usp.br/moise>

★ <http://www.lti.pcs.usp.br/saci>

- Proposta independente da arquitetura dos agentes (camada de aplicação).

# Resumo

- $\mathcal{S}$ -**MOISE**<sup>+</sup>: uma arquitetura para SMA com organização **tipo OC** (centrado na organização com agentes que raciocinam sobre sua organização).

Disponível em

★ <http://www.lti.pcs.usp.br/moise>

★ <http://www.lti.pcs.usp.br/saci>

- Proposta independente da arquitetura dos agentes (camada de aplicação).
- Limitações (com solução em uma **arquitetura** de agente organizacional)
  - ★ Não garante as ligações de autoridade.
  - ★ Não tem tratamento de exceções (um agente deixar uma missão sem terminar as metas)



## Considerações finais

- O que é uma organização para SMA
- Para que serve uma organização
- Quais as formas de conceber uma organização
- Como programar um SMA com organização
- $\mathcal{MOISE}^+$ 
  - ★ organização independente dos agentes
  - ★ três dimensões
  - ★ suporte para reorganização

## Referências

- [Castelfranchi et al., 1999] Castelfranchi, C., Dignum, F., Jonker, C. M., and Treur, J. (1999). Deliberate normative agents: Principles and architecture. In *Proceedings of The Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*.
- [Decker, 1998] Decker, K. S. (1998). Task environment centered simulation. In Prietula, M. J., Carley, K. M., and Gasser, L., editors, *Simulating Organizations: Computational Models of Institutions and Groups*, chapter 6, pages 105–128. AAAI Press / MIT Press, Menlo Park.
- [Dignum and Dignum, 2001] Dignum, V. and Dignum, F. (2001). Modelling agent societies: Co-ordination frameworks and institutions. In Brazdil, P. and Jorge, A., editors, *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)*, LNAI 2258, pages 191–204, Berlin. Springer.
- [Esteva et al., 2004] Esteva, M., Rodríguez-Aguilar, J. A., Rosell, B., and L., J. (2004). AMELI: An agent-based middleware for electronic institutions. In Jennings, N. R., Sierra, C., Sonenberg, L., and Tambe, M., editors, *Proceedings*

*of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2004)*, pages 236–243, New York. ACM.

[Ferber and Gutknecht, 1998] Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agents systems. In Demazeau, Y., editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135. IEEE Press.

[Fox et al., 1998] Fox, M. S., Barbuceanu, M., Gruninger, M., and Lon, J. (1998). An organizational ontology for enterprise modeling. In Prietula, M. J., Carley, K. M., and Gasser, L., editors, *Simulating Organizations: Computational Models of Institutions and Groups*, chapter 7, pages 131–152. AAAI Press / MIT Press, Menlo Park.

[Garijo et al., 2001] Garijo, F., Gómez-Sanz, J. J., Pavón, J., and Massonet, P. (2001). Multi-agent system organization: An engineering perspective. In *Pre-Proceeding of the 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*.

[Gasser, 2001] Gasser, L. (2001). Organizations in multi-agent systems. In *Pre-Proceeding of the 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*, Annecy.

[Gutknecht and Ferber, 2000] Gutknecht, O. and Ferber, J. (2000). The MadKit

agent platform architecture. In *Agents Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55.

[Hübner, 2003] Hübner, J. F. (2003). *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Universidade de São Paulo, Escola Politécnica.

<http://www.inf.furb.br/~jomi/pubs/2003/Hubner-tese.pdf>.

[Hübner and Sichman, 2003] Hübner, J. F. and Sichman, J. S. (2003). Organização de sistemas multiagentes. In Vieira, R., Osório, F., and Rezende, S., editors, *III Jornada de Mini-Cursos de Inteligência Artificial (JAIA'03)*, volume 8, pages 247–296. SBC, Campinas.

<http://www.inf.furb.br/~jomi/pubs/2003/Hubner-jaia2003.pdf>.

[Hübner et al., 2005] Hübner, J. F., Sichman, J. S., and Boissier, O. (2005). S-MOISE+: A middleware for developing organised multi-agent systems. In Boissier, O., Dignum, V., Matson, E., and Sichman, J. S., editors, *Proceedings of the International Workshop on Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS (OOP'2005)*.

<http://www.inf.furb.br/~jomi/pubs/2005/Hubner-ooop2005.pdf>.

[Lemaître and Excelente, 1998] Lemaître, C. and Excelente, C. B. (1998). Multi-agent organization approach. In Garijo, F. J. and Lemaître, C., editors, *Proceedings of II Iberoamerican Workshop on DAI and MAS*.

- [Malone, 1999] Malone, T. W. (1999). Tools for inventing organizations: Toward a handbook of organizational process. *Management Science*, 45(3):425–443.
- [Ossowski, 1999] Ossowski, S. (1999). *Co-ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents*, volume 1535 of *LNAI*. Springer.
- [Pynadath and Tambe, 2003] Pynadath, D. V. and Tambe, M. (2003). An automated teamwork infrastructure for heterogeneous software agents and humans. *Autonomous Agents and Multi-Agent Systems*, 7(1–2):71–100.
- [Wooldridge et al., 1999] Wooldridge, M., Jennings, N. R., and david Kinny (1999). A methodology for agent-oriented analysis and design. In *Proceedings of the Third International Conference on Autonomous Agentes (Agent's 99)*. ACM.