

# Modelagem de Agentes com Framework i\*

---

Percy Machado  
Universidade do Estado do Rio de Janeiro

# O que é $i^*$ ?

É um framework de modelagem que propõe uma abordagem orientada a agentes

Focado nas interdependências dos agentes (metas, tarefas, recursos e etc)

O nome  $i^*$  (i-estrela) origina-se do conceito de “intenção distribuída”



# Histórico

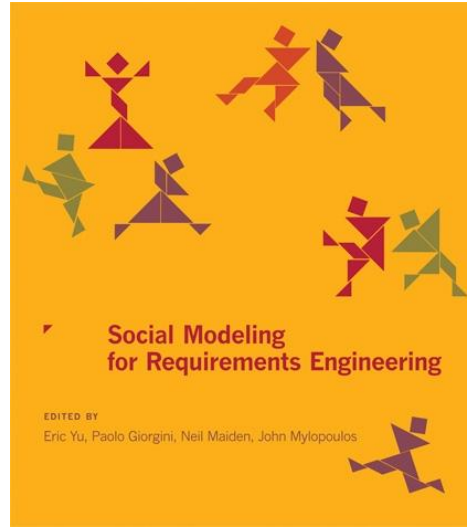
Proposto por Eric Yu em 1995 - Tese de Doutorado da University of Toronto

Em 2008, framework i\* se tornou parte do URN (*User Requirements Notation*)

Padrão internacional reconhecido pela ITU (*International Telecommunication Union*)



Livro: Social Modeling for Requirements Engineering



# Motivação

## **Facilitar o entendimento sobre o sistema em análise**

Representar atores e seus relacionamentos através de uma visão gráfica (intenções, dependências, responsabilidades e vulnerabilidades)

- ❑ Melhorar a compreensão sobre os relacionamentos da organização
- ❑ Entender os fatores envolvidos nos processos de decisões
- ❑ Descrever possíveis alternativas para o processo

# Principais Características

Ênfase em aspectos sociais dos atores (intenções, dependências, responsabilidades e vulnerabilidades)

Busca compreender o processo a fundo por meio de uma visão intencional e estratégica

Permite modelar processos que envolvem atores humanos e computacionais

Responde QUEM e POR QUE, mas não *o que*



# Áreas de Aplicação

Indicado para domínios onde as várias partes (atores) possuem os interesses distintos, que podem ser conflitantes ou benéficos uns aos outros

## **Requirements Engineering**

Levantamento de requisitos

## **Business Process Reengineering**

Identificar motivações e justificativas

## **Organizational Impacts Analysis**

Avaliar impacto de mudanças (oportunidades, custos, liderança)

## **Software Process Modelling**

Capturar motivações e racional das atividades do processo de desenvolvimento software

# Modelos

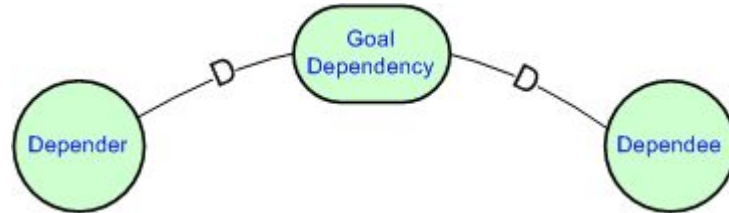
- ★ **Dependência Estratégica** (SD – Strategic Dependency)
  - ★ **Razão Estratégica** (SR – Strategic Rationale)
- 
- ❑ Utilizados para capturar as intenções e responsabilidades dos atores
  - ❑ Auxilia a definir arquitetura do projeto
  - ❑ Podem ser usados como parte da documentação do software

# SD - Strategic Dependency

Rede de dependências entre os atores

*Depender* (ator dependente), depende que outro ator

*Dependee* (ator de quem se depende), responsável por satisfazer a dependência

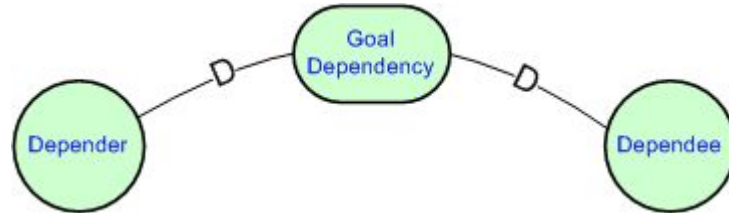




# SD - Tipos de Dependência

## Goal dependency

*Depender* depende do *Dependee* para atingir um determinado estado (meta)



# SD - Tipos de Dependência

## Goal dependency

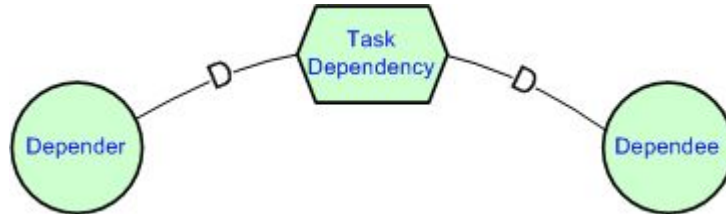
Exemplo



# SD - Tipos de Dependência

## Task dependency

*Depender* depende do *Dependee* para realizar uma tarefa



# SD - Tipos de Dependência

## Task dependency

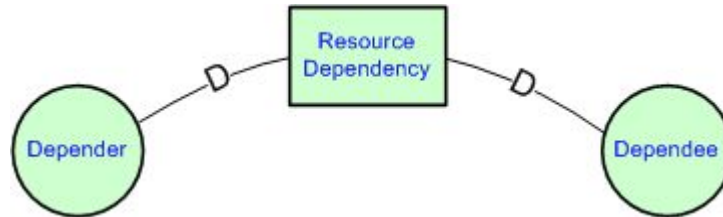
Exemplo



# SD - Tipos de Dependência

## Resource dependency

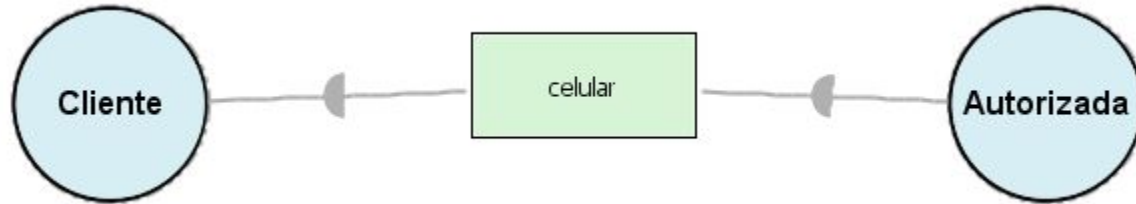
*Depender* depende do *Dependee* para a disponibilidade de um recurso (físico ou lógico)



# SD - Tipos de Dependência

## Resource dependency

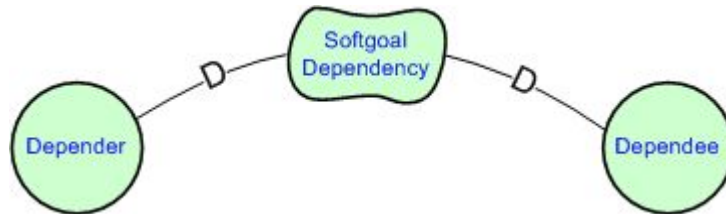
Exemplo



# SD - Tipos de Dependência

## Softgoal dependency

*Depender* depende do *Dependee* para executar alguma tarefa que atende a uma meta flexível (requisito de qualidade ou não funcional)



# SD - Tipos de Dependência

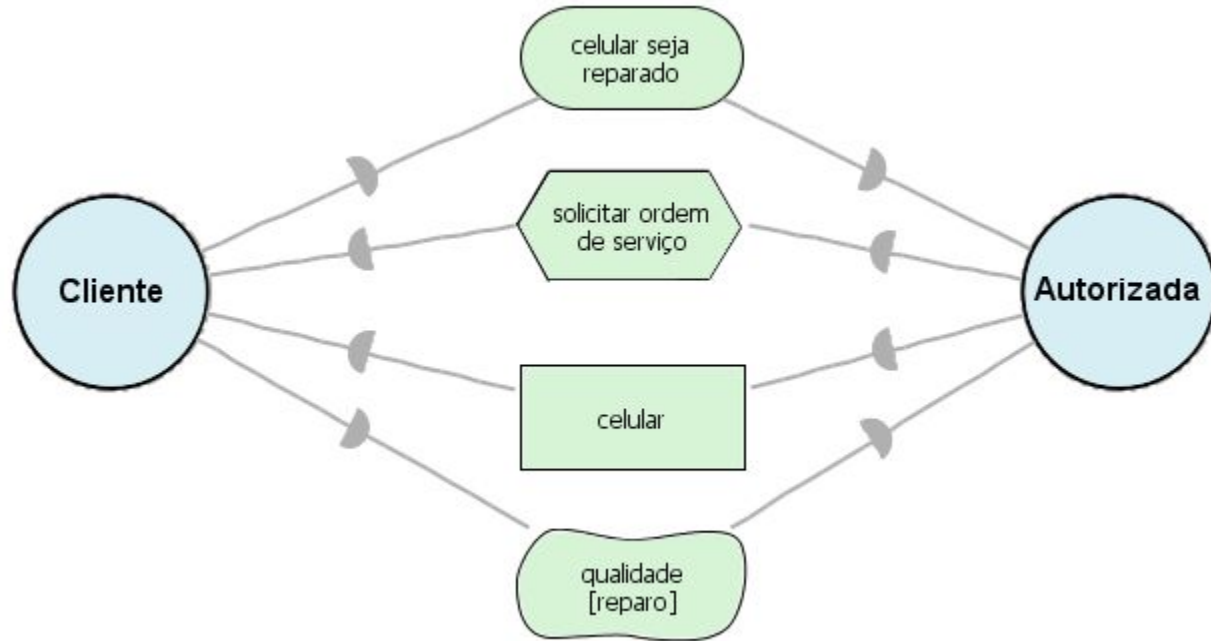
## Softgoal dependency

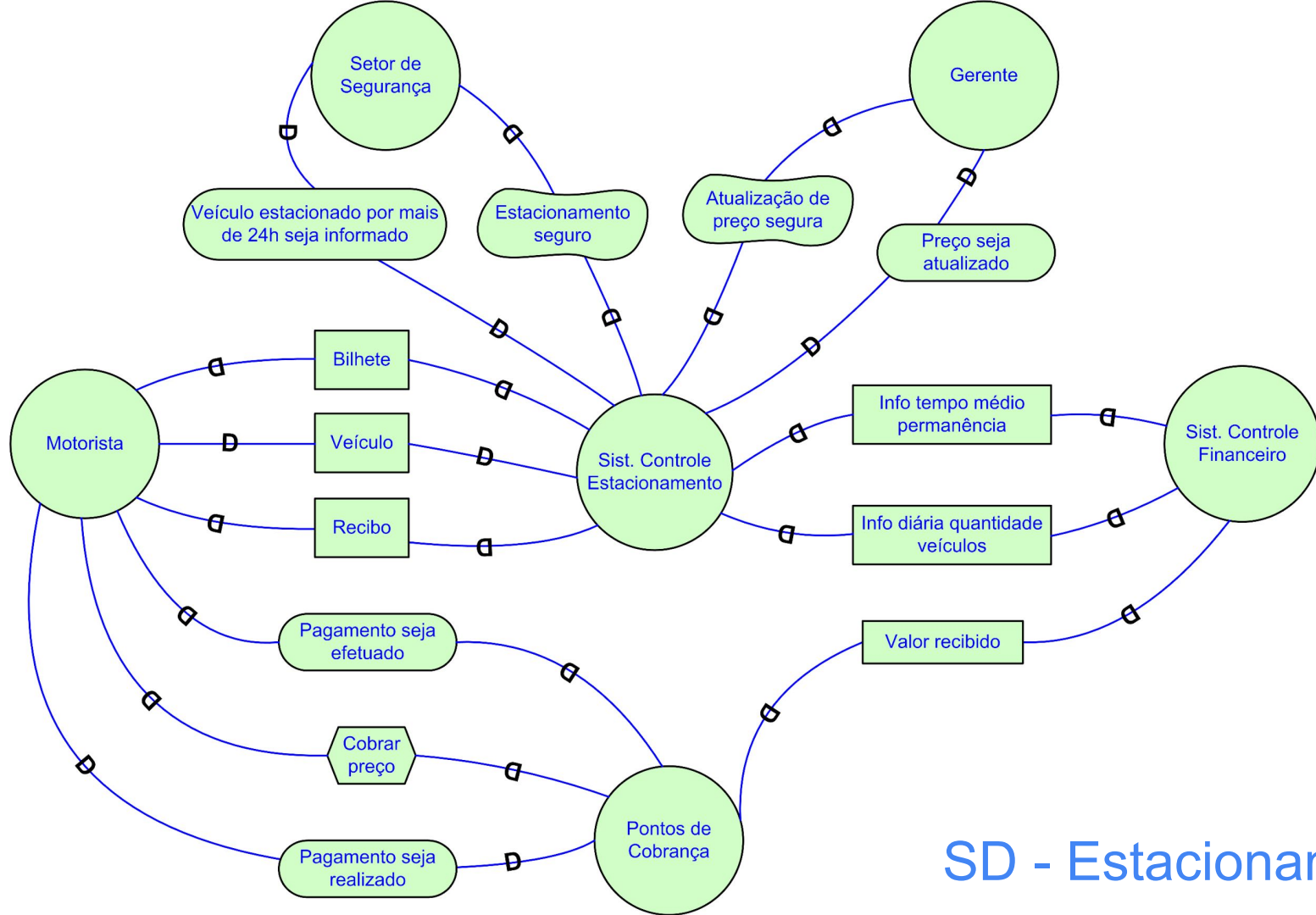
Exemplo





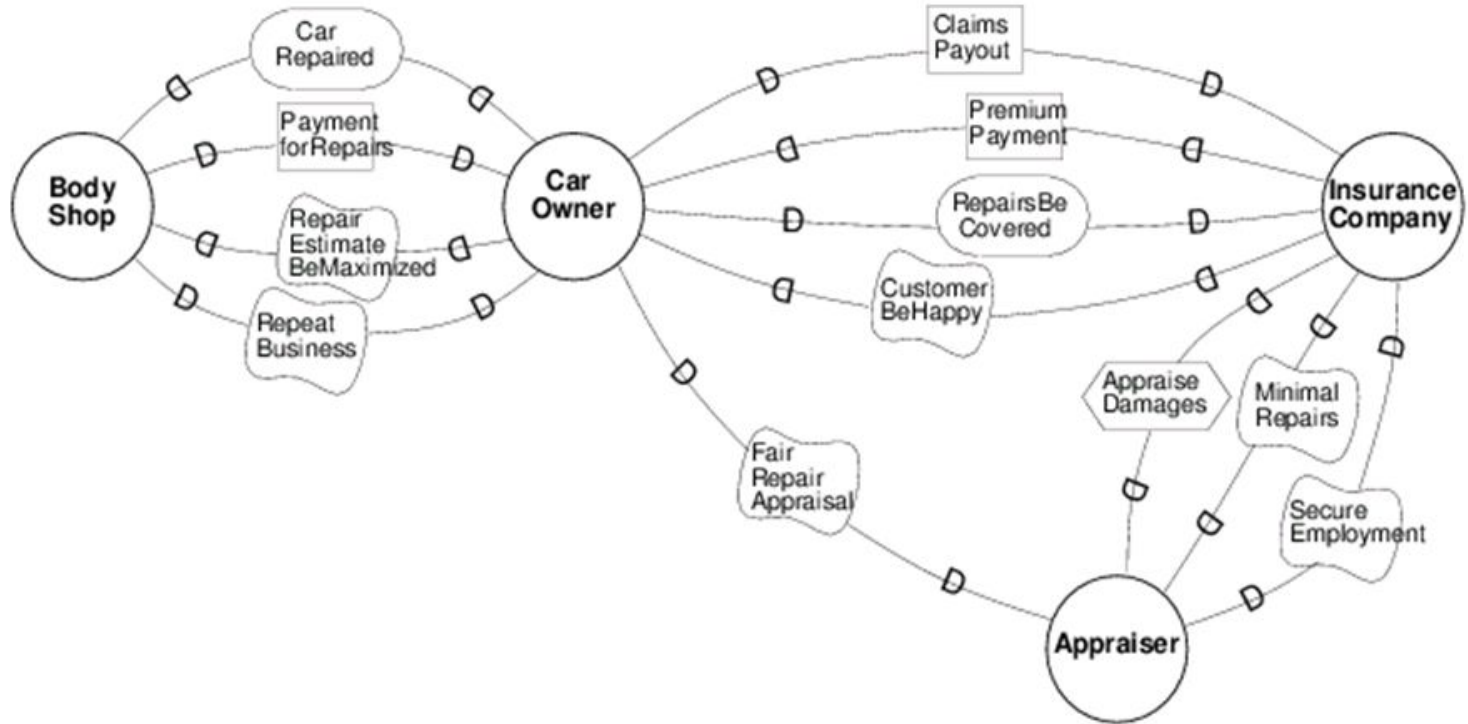
# SD - Reparo de Celular





SD - Estacionamento

# SD - Automobile Insurance



# SD - Strategic Dependency

## Conclusão

Modelagem de alto nível de um processo

Captura apenas o importante para os atores e oculta os detalhes

## Análises

- ★ Identificar e mitigar vulnerabilidades da cadeia de dependências
- ★ Explorar oportunidades disponíveis para cada ator

# SR - Strategic Rationale

Compreensão aprofundada sobre o raciocínio estratégico dos atores

Descrição intencional dos processos e as razões por trás deles

Representa explicitamente QUEM, POR QUE e alternativas para o processo

Os principais tipos de relacionamentos são means-ends e task-decomposition

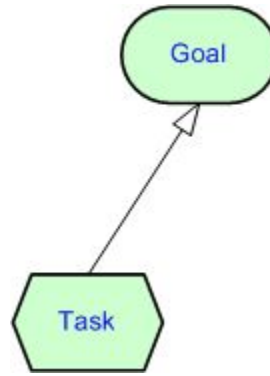


# SR - Strategic Rationale

## Means-ends

Relação entre uma extremidade e um meio para alcançá-la

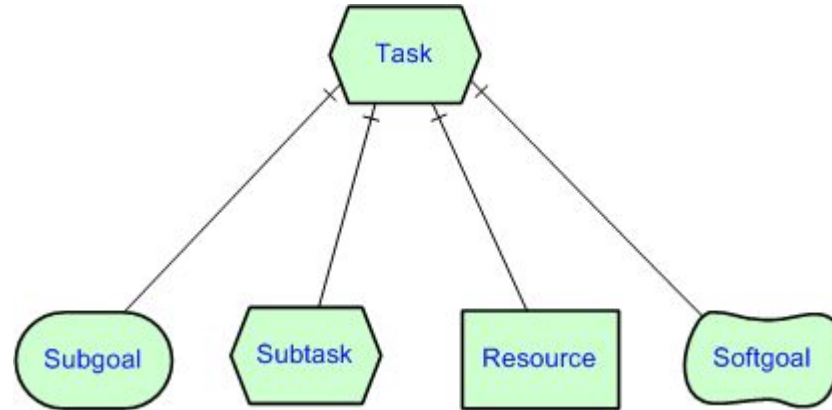
Usado apenas para ligar uma tarefa a uma meta



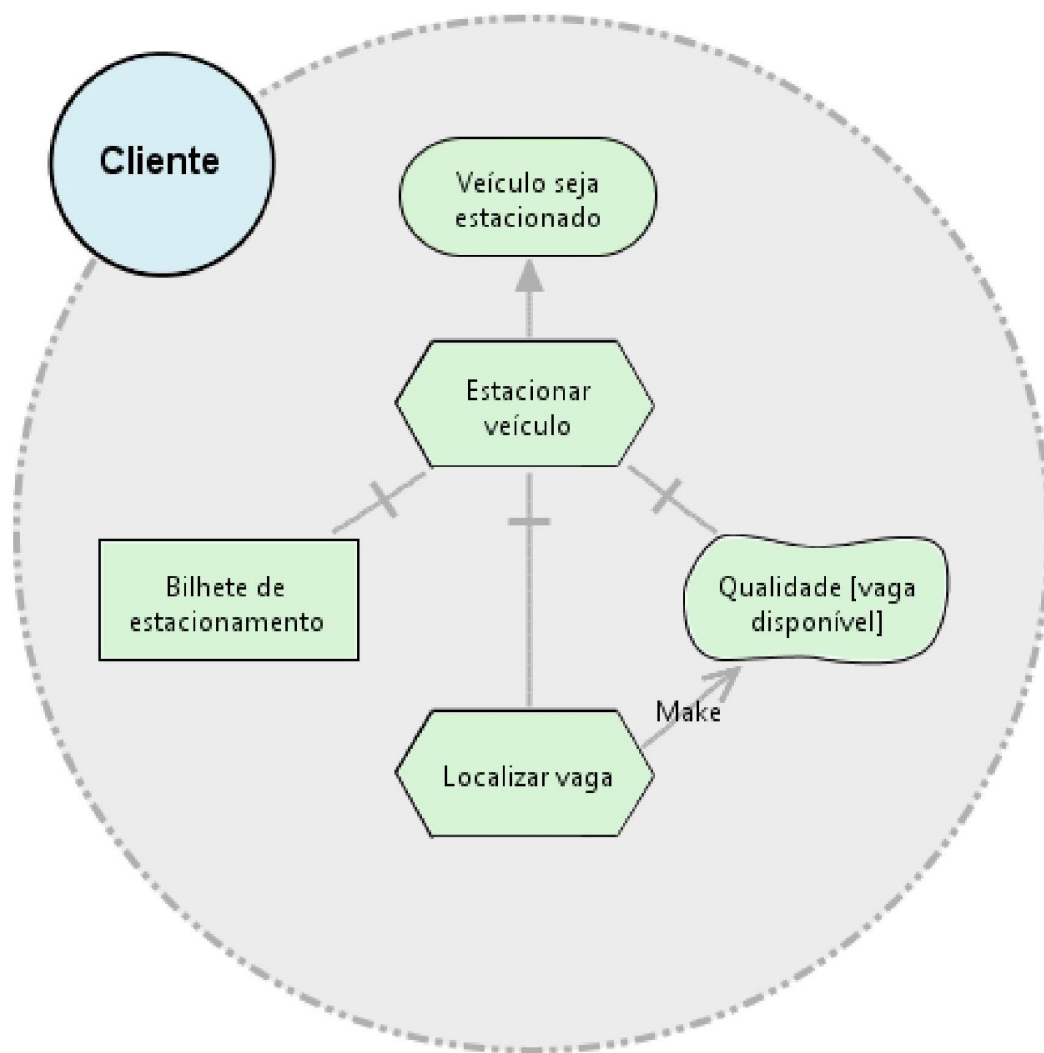
# SR - Strategic Rationale

## Task-decomposition

Tarefa é dividida em subelementos (metas, tarefas ou recursos)



## SR - Cliente



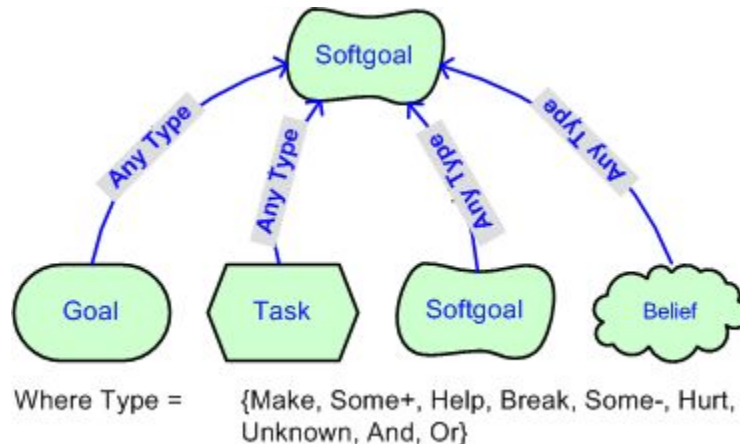


# SR - Strategic Rationale

## Contribution

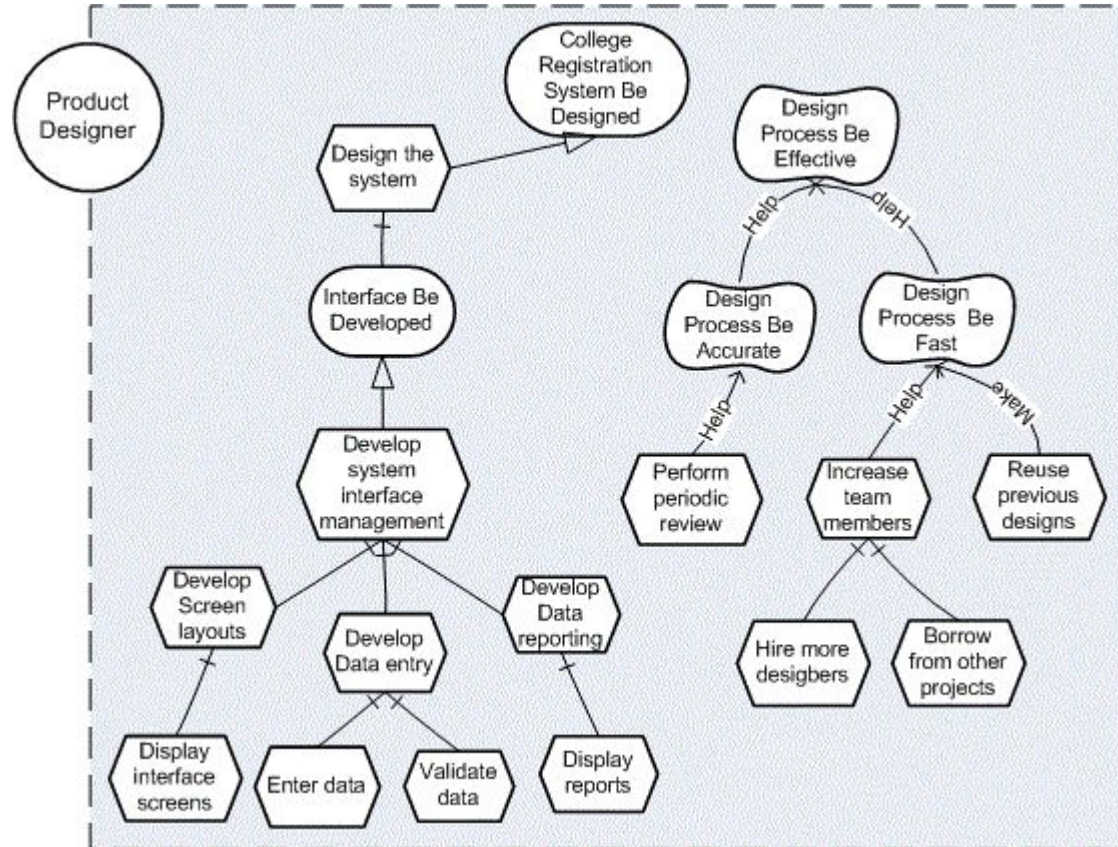
Elemento (meta, tarefa, crença) contribui para que a softgoal seja satisfeita

Tipos: *Make, Some+, Help, Unknown, Hurt, Some-, Break, Or, And*

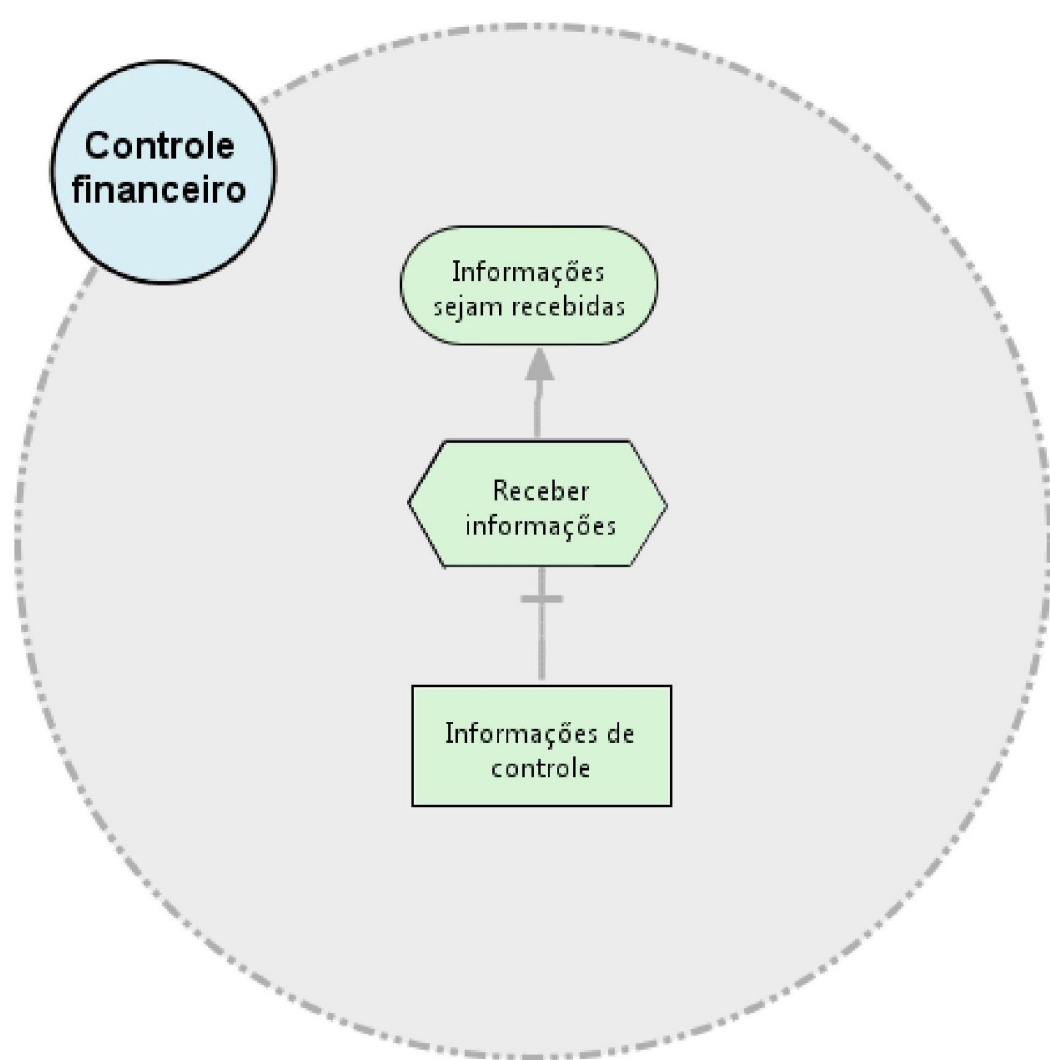


# SR - Strategic Rationale

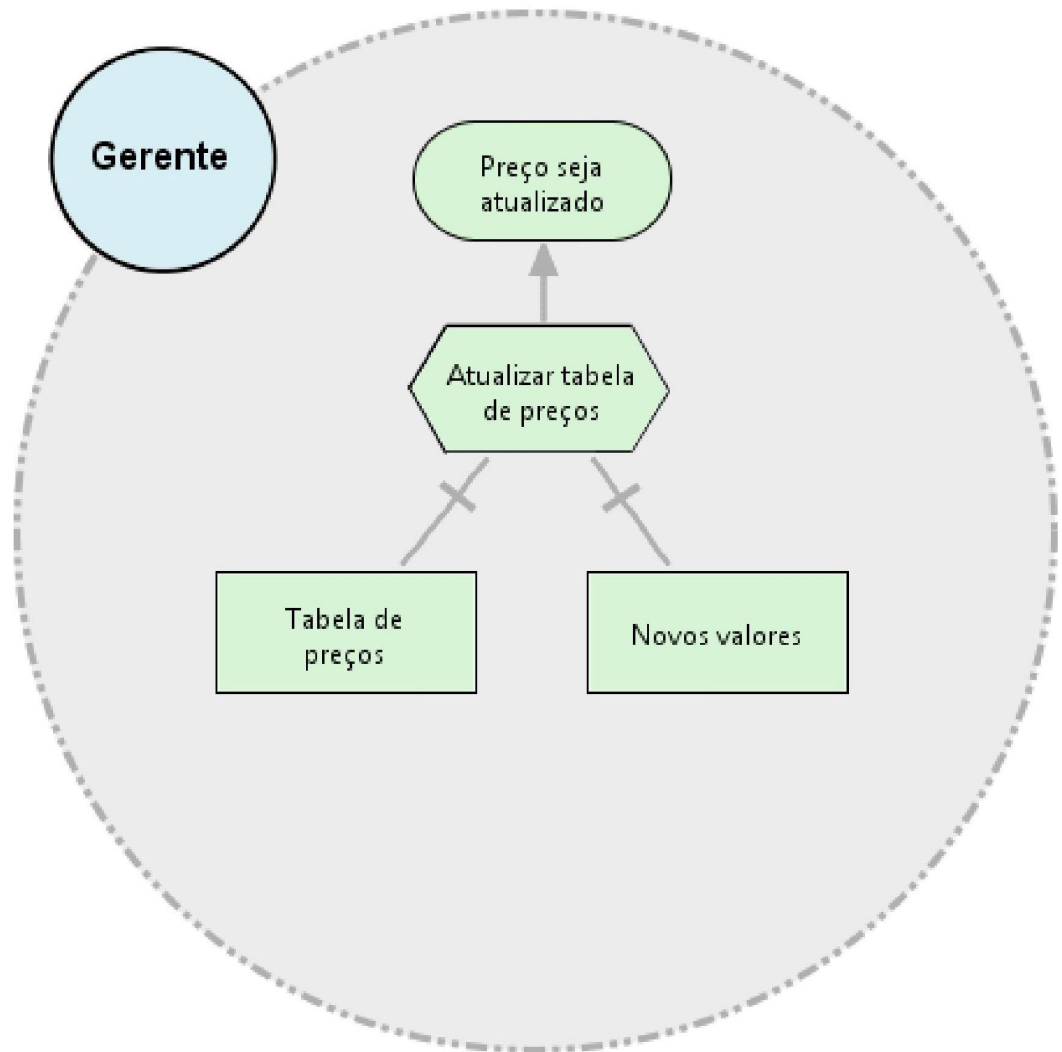
## Contribution Exemplo



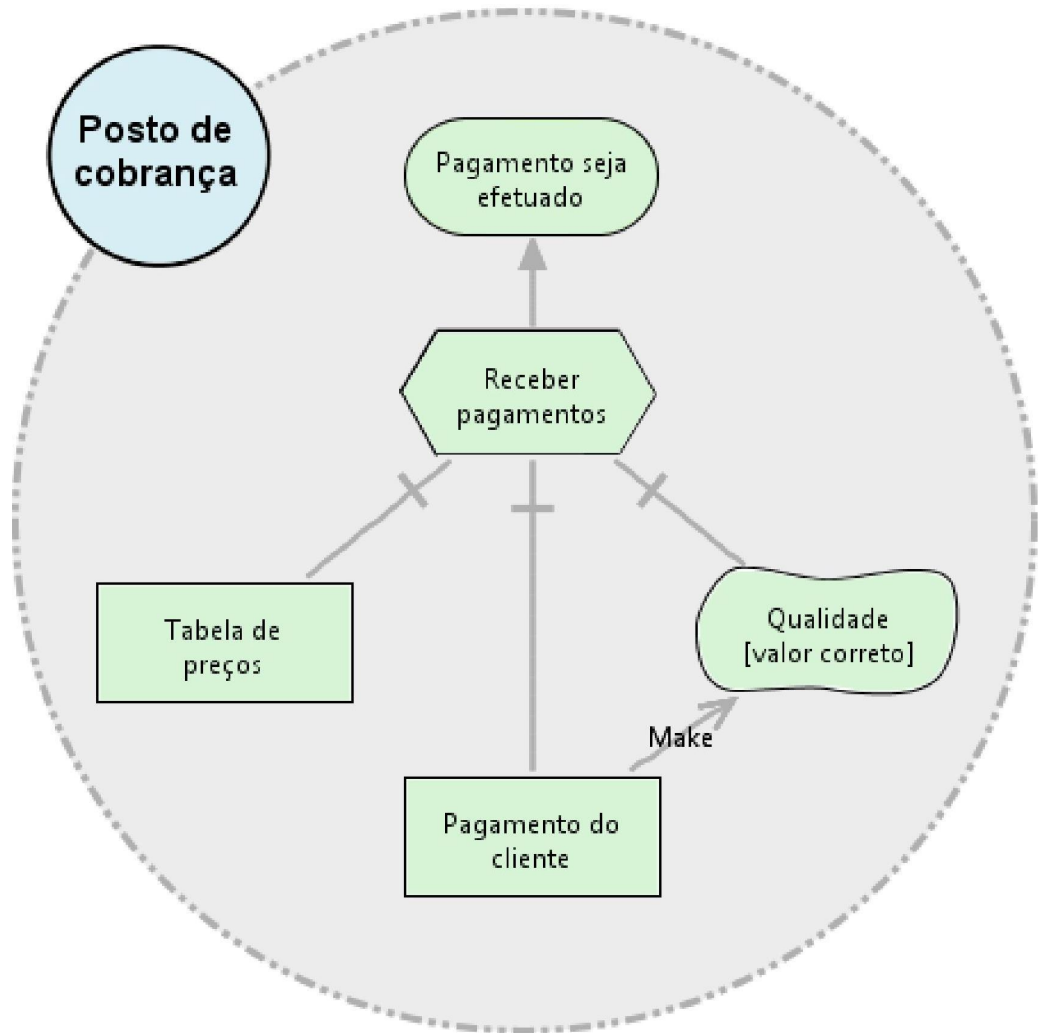
# SR - Controle Financeiro



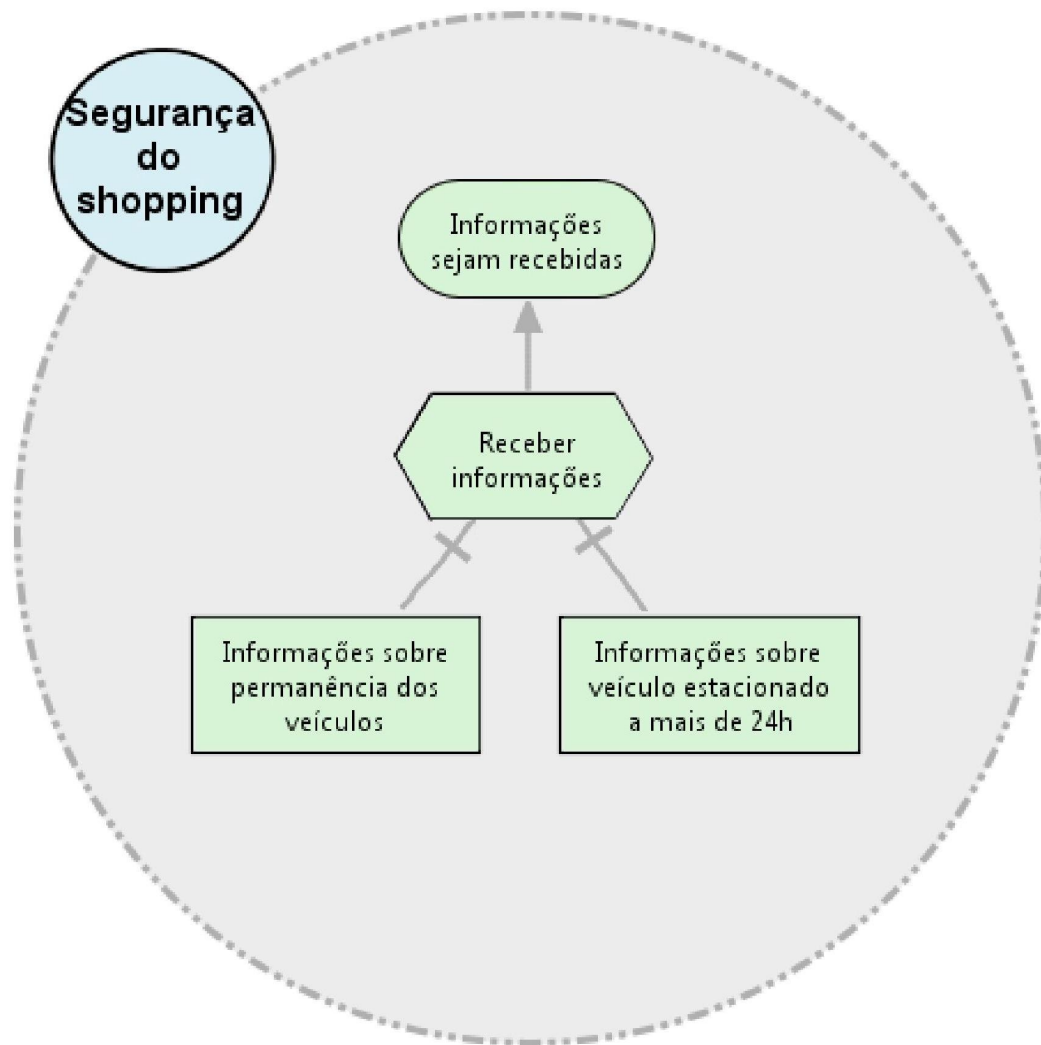
## SR - Gerente

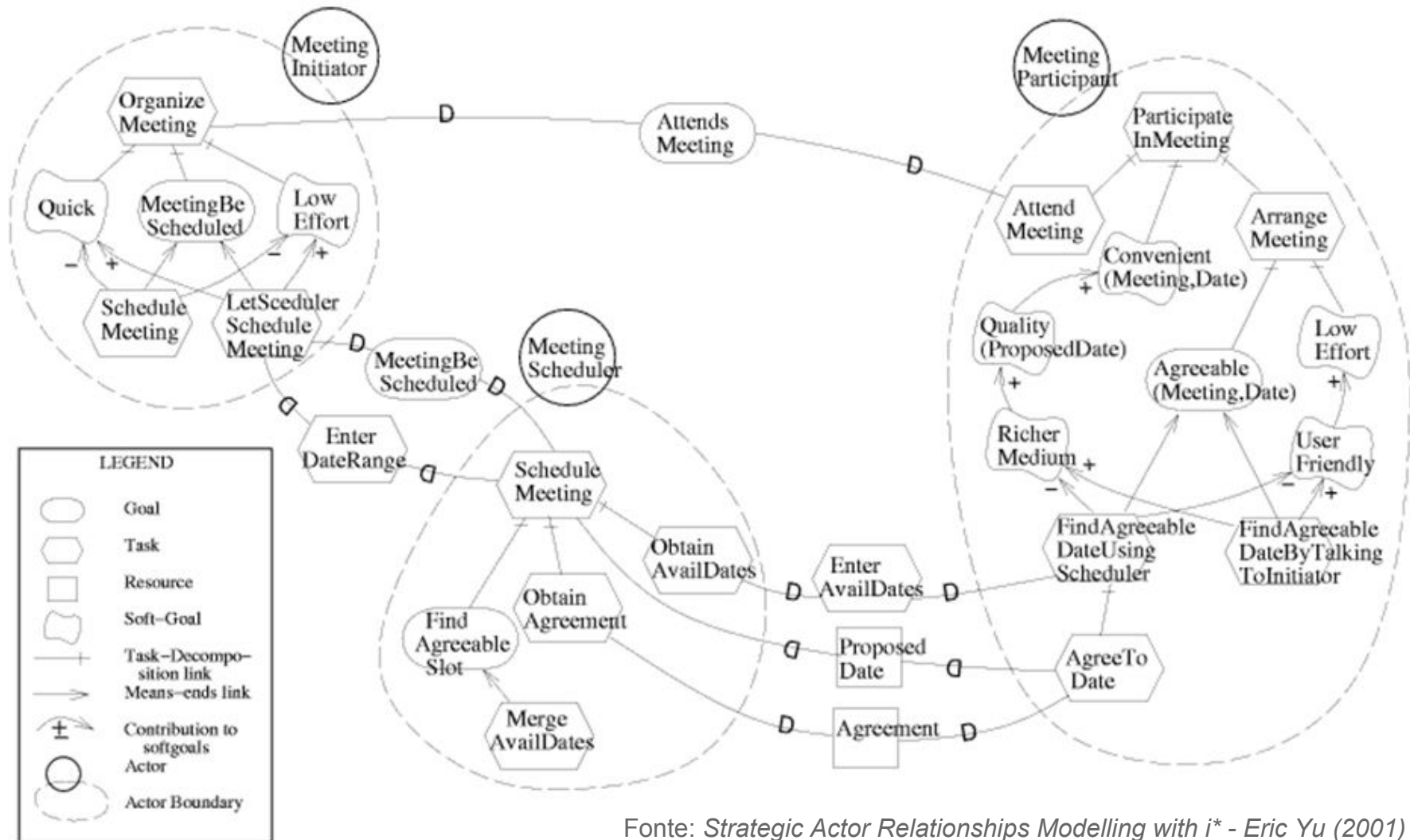


## SR - Posto de Cobrança



# SR - Segurança do Shopping





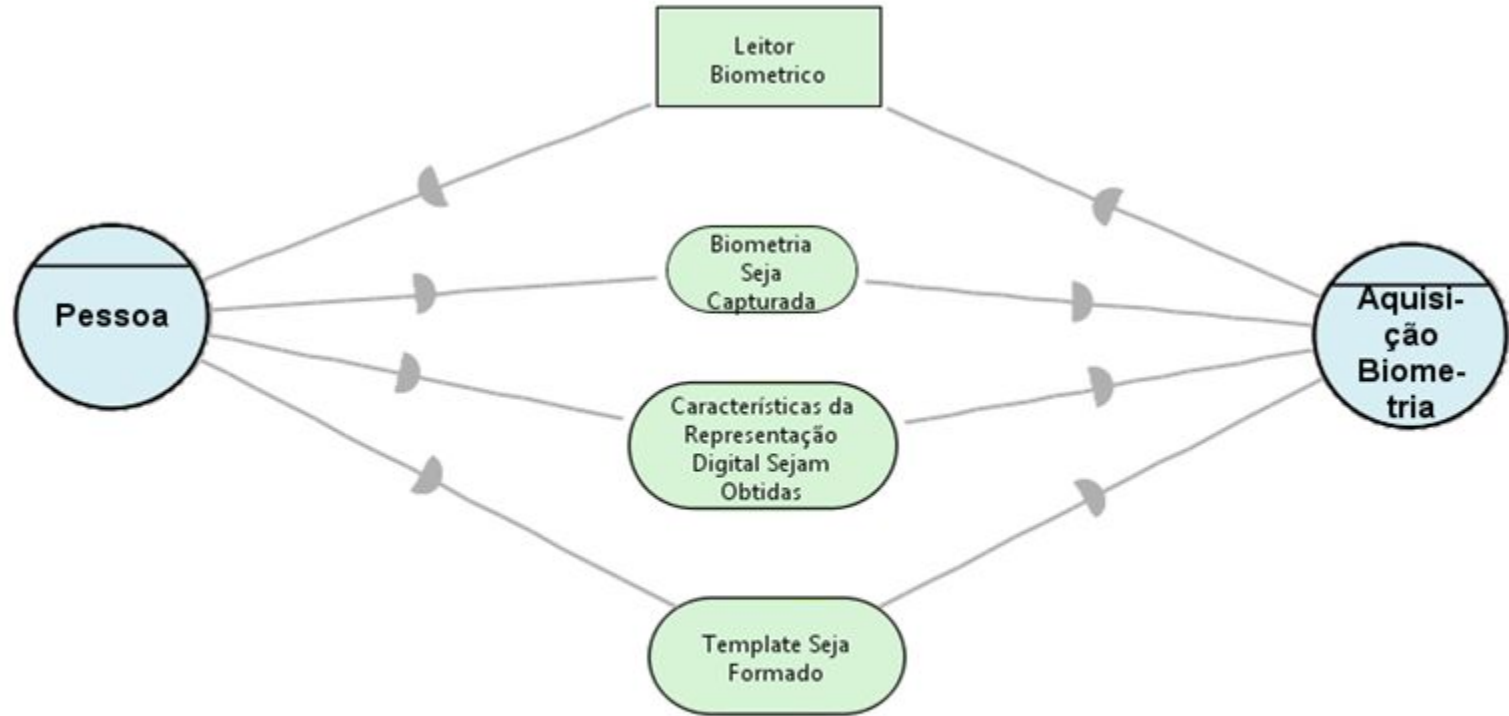






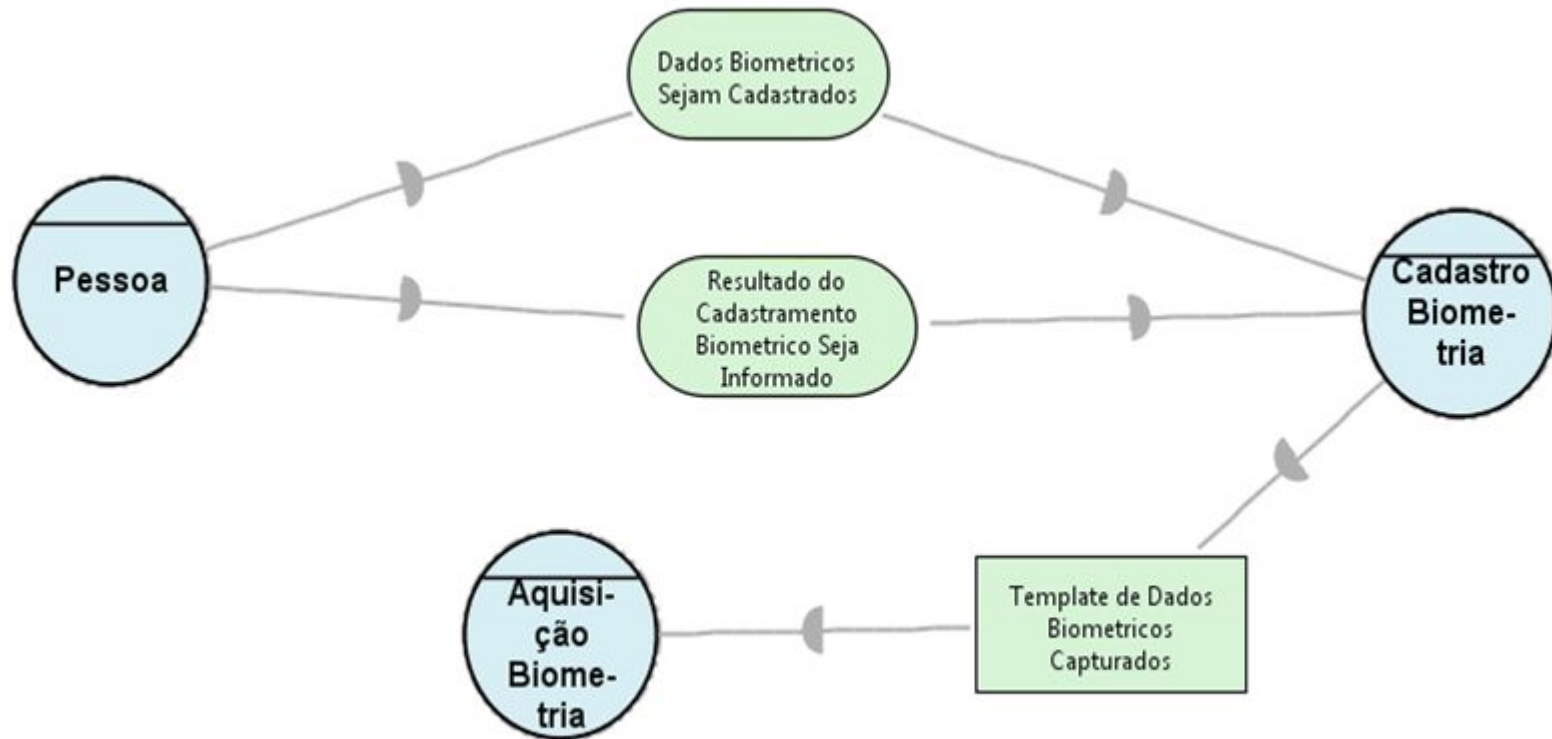
# Estudo de Caso

## Aquisição de Dados Biométricos - SD



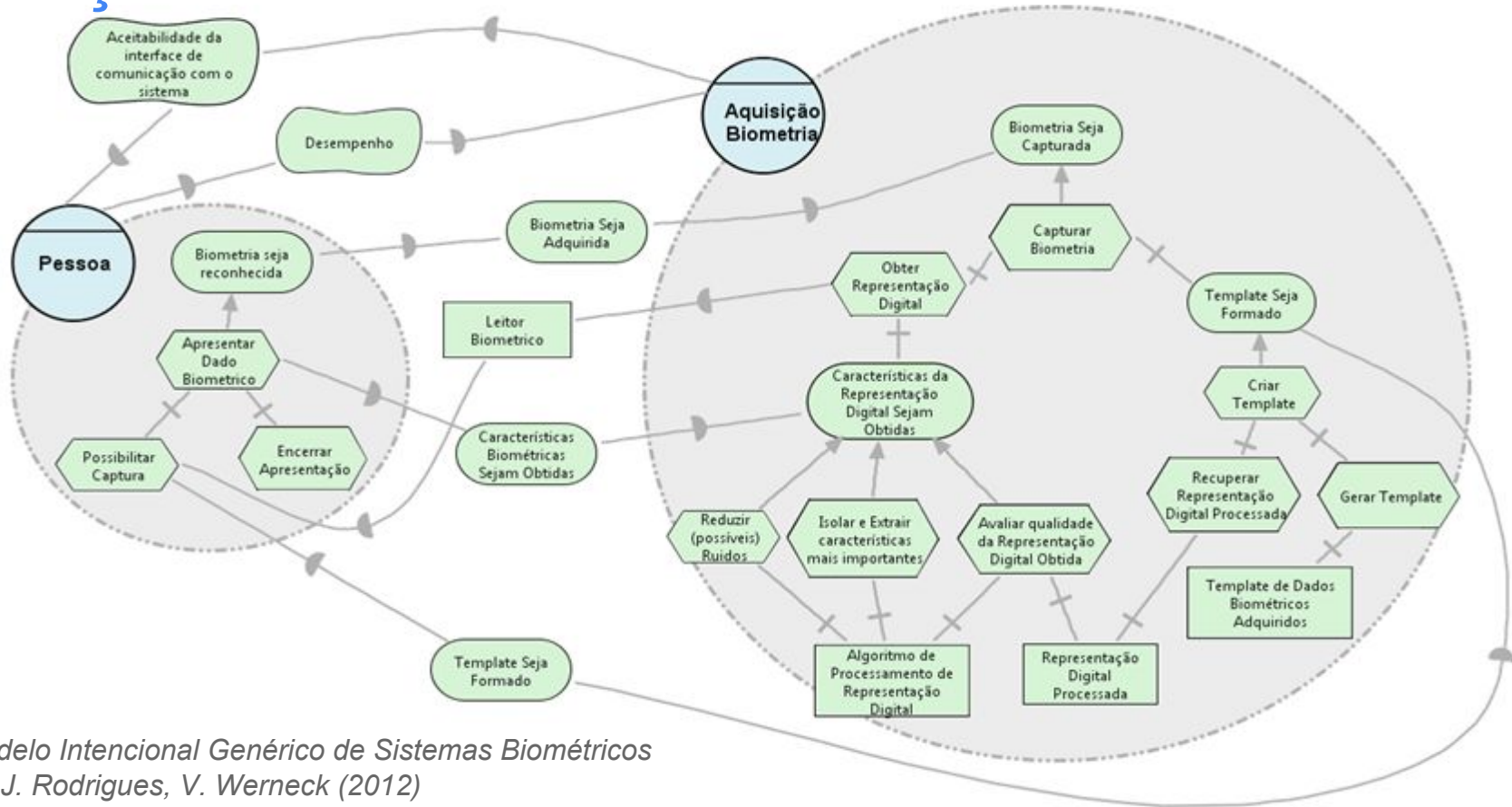
# Estudo de Caso

## Cadastramento de Dados Biométricos - SD



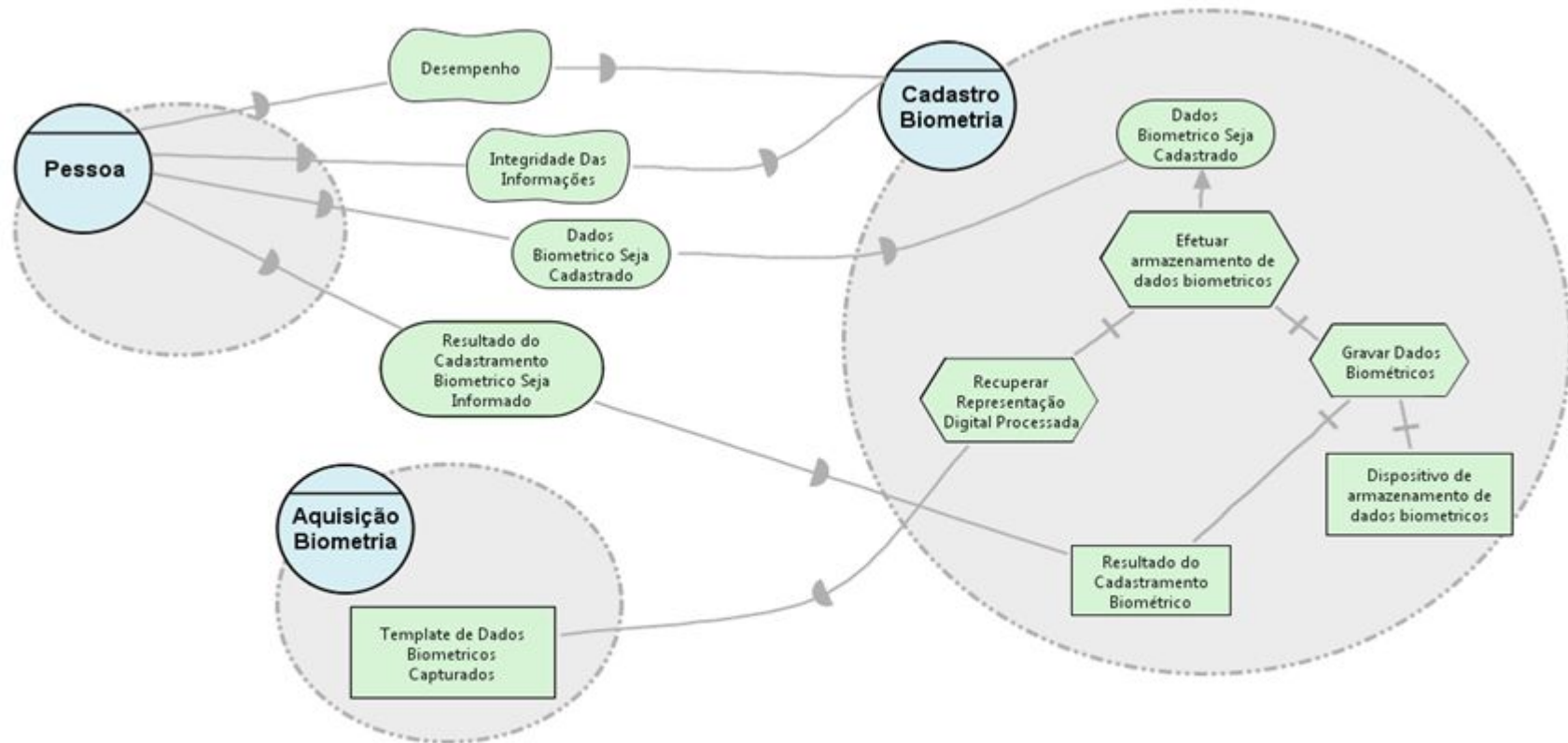
# Estudo de Caso

## Aquisição de Dados Biométricos - SR



# Estudo de Caso

## Cadastramento de Dados Biométricos - SR



# SR - Strategic Rationale

## Análises

- ★ Capacidade  
Ator possui algum meio para cumprir a meta?
- ★ *Workability*  
Processo que vai funcionar?
- ★ Viabilidade  
Será efetivo?
- ★ Credibilidade  
Evidências para confirmar que vai, ou não, funcionar

# Exercício

## Booktrading

Crie os diagramas SD e SR para um sistema de comércio eletrônico, com as seguintes características:

- ☐ Vendedores tentam vender pelo maior preço possível
- ☐ Compradores tentam comprar pelo menor preço possível
- ☐ Orientado a agentes



# Ferramentas

OME

<http://www.cs.toronto.edu/km/ome>

Open OME

<https://se.cs.toronto.edu/trac/ome>

# Referências

University of Toronto

<http://www.cs.toronto.edu/km/istar>

i\* Wiki

<http://istar.rwth-aachen.de/tiki-index.php>



# Obrigado



Percy Machado  
Universidade do Estado do Rio de Janeiro