

## 2 Técnicas, Métodos e Conceitos

Na primeira seção este capítulo faz uma recuperação das técnicas, métodos e conceitos já divulgados pela literatura e na segunda seção, apresenta técnicas, métodos e conceitos elaborados durante a pesquisa para o Método ERi\*c.

### 2.1 Técnicas, Métodos e Conceitos Utilizados

Esta seção apresenta os seguintes assuntos: “O Léxico Ampliado da Linguagem”, “A Técnica de Cenários” e “O Framework de Modelagem i\*”.

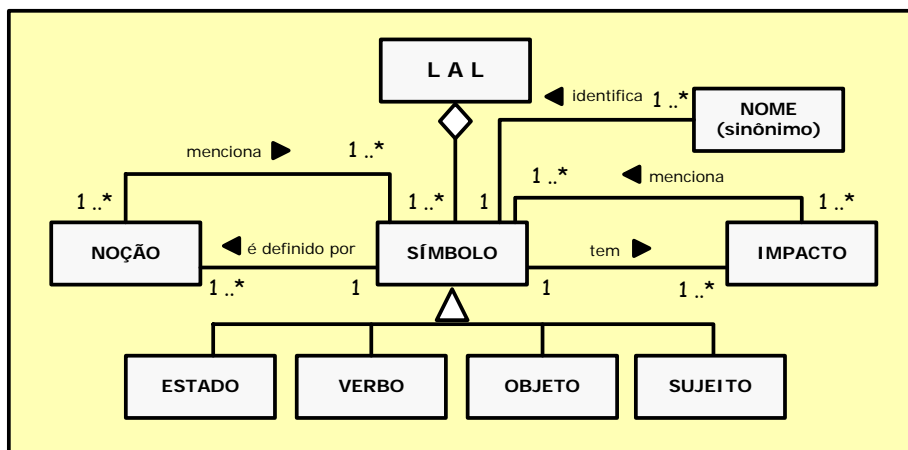
#### 2.1.1 O Léxico Ampliado da Linguagem

O Léxico Ampliado da Linguagem (LAL) [Leite 93] é a principal abordagem utilizada para a elicitación dos atores e metas da primeira etapa do método proposto. O LAL, ou Léxico, se baseia na premissa de que em uma organização (meio social) existe uma linguagem estendida da linguagem natural que é utilizada pelos atores da organização. Essa linguagem, com vocabulário peculiar (composto de símbolos), é denominada linguagem da aplicação [Leite 93].

O Léxico tem como idéia central um argumento muito simples: “entender a linguagem do domínio sem se preocupar com o entendimento do problema envolvido”. O objetivo do LAL é representar palavras e sentenças, chamadas de símbolos, próprias ou peculiares ao Universo de Informação (UdI). UdI se refere ao contexto de uma aplicação e é composto por todas as fontes de informação e pessoas relacionadas ao software. “UdI é o contexto geral no qual o software será desenvolvido e operado”.

A Figura 2.1 [Oliveira 06a] mostra os componentes do LAL através de um diagrama de classes. O LAL é formado por símbolos, também chamados de entradas, e cada um deles é identificado por um nome. Um símbolo pode possuir mais de um nome (caso de sinônimos) e é representado por duas descrições. A primeira, denominada noção, é a denotação do símbolo, ou seja, seu significado, equivalente à descrição encontrada em dicionários. A segunda, chamada de impacto ou resposta comportamental, é a conotação do símbolo, isto é, informação extra sobre os efeitos comportamentais do símbolo no contexto do UdI. O diagrama mostra que ambos, noção

e impacto, fazem menção a outros símbolos. Os símbolos no LAL são classificados em quatro categorias diferentes: objeto, sujeito, estado e verbo. O objetivo do LAL é capturar o vocabulário intrínseco à aplicação [Franco 92].



**Figura 2.1** – Os elementos do LAL representados em um diagrama de classes

Com a finalidade de evitar representações prolongadas que podem poluir a representação dos símbolos, duas leis básicas precisam ser seguidas pelo engenheiro de requisitos na elaboração do LAL. A primeira, denominada “**princípio da circularidade**”, recomenda maximizar o uso dos outros símbolos do léxico quando se descreve a noção e o impacto de um novo símbolo. A segunda, denominada “**princípio do vocabulário mínimo**”, recomenda minimizar o uso de símbolos externos ao UdI.

Dicionários, de modo geral, só representam a denotação dos termos. O LAL, através dos impactos, representa também os relacionamentos comportamentais que envolvem os símbolos e, justamente por isso, faz uma representação muito mais rica e detalhada do UdI.

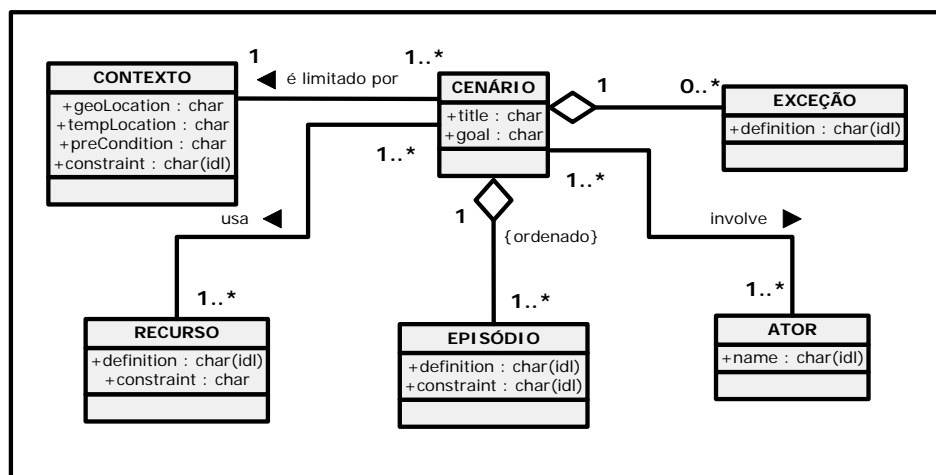
O processo de construção do LAL consiste em seis passos, listados a seguir [Kaplan 00]:

- ◆ Identificar as fontes de informações do UdI;
- ◆ Identificar a lista de símbolos relevantes no UdI;
- ◆ Classificar os símbolos como: sujeito, objeto, verbo, e estado;
- ◆ Descrever os símbolos através da noção e do impacto garantindo os princípios do LAL;
- ◆ Verificar o LAL através de inspeção;
- ◆ Validar o LAL com os atores do UdI.

Para identificar a qual classe o símbolo pertence são sugeridas as seguintes heurísticas: se o símbolo pratica uma ação, ele é classificado como sujeito; se sofre a ação, é classificado como objeto; se o símbolo é uma situação em um dado momento, é classificado como estado; se representa uma ação, ele é classificado como verbo.

### 2.1.2 A Técnica de Cenários

A Técnica de Cenários é uma técnica bem conhecida na área da engenharia de requisitos. Um cenário é uma descrição estruturada de uma situação [Carroll 00] [Leite 00] que ocorre no mundo real. Os “cenários” possuem várias características: eles são concretos e têm objetivo; necessitam de recursos e envolvem um ou mais atores; acontecem em local e tempo definidos; podem ter restrições que qualificam o cenário ou fazem alguma imposição. Eles são individualmente independentes, inter-relacionados e podem ter alternativas de curso representadas por exceções. A Figura 2.2 [Oliveira 06a] mostra o relacionamento entre os elementos trabalhados pela Técnica de Cenários. Cada cenário é identificado por um nome ou título, e pode ter exceções que devem ser consideradas. Um cenário satisfaz um objetivo e um contexto especifica sua fronteira. Cenários são formados por episódios, que definem requisitos funcionais.



**Figura 2.2** – Os relacionamentos entre os elementos do cenário representados em um diagrama de classes

Uma característica importante é que a Técnica de Cenários [Leite 00] usa a linguagem do UdI. Desse modo os cenários ficam com a especificação muito coesa, pois têm ancora nos símbolos do LAL. Essa característica traz benefícios para a

representação dos cenários, os quais são demonstrados e discutidos por Oliveira et al. [Oliveira 06a], que apresenta um exemplo do uso da técnica.

### 2.1.3 O Framework de Modelagem i\*

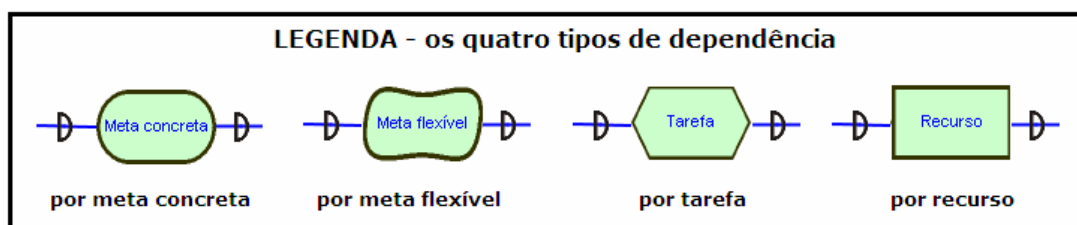
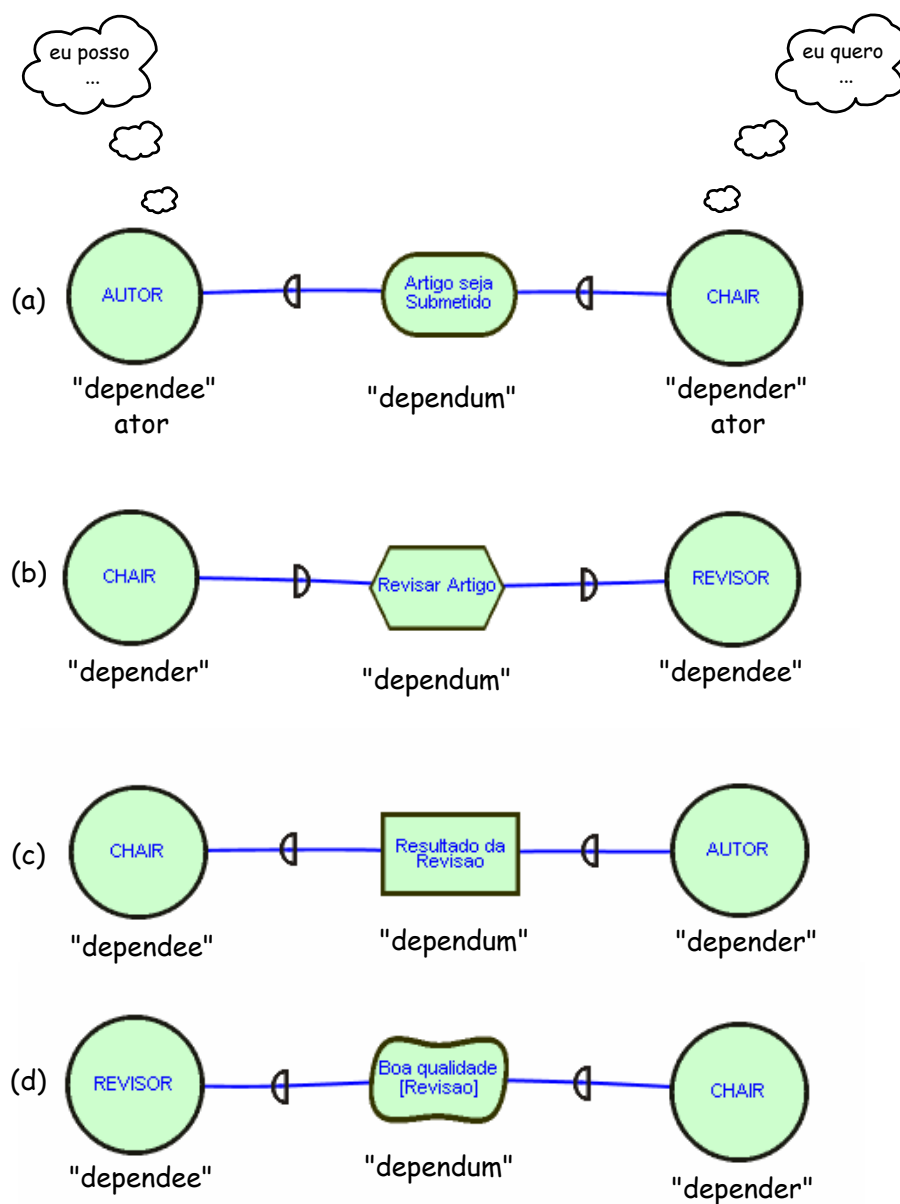
O Framework de Modelagem i\* (i-estrela) [Yu 95] modela contextos organizacionais com base nos relacionamentos de dependência entre os atores participantes. A idéia central do i\* é representar através de modelos os atores e as dependências que eles têm uns com os outros para que metas próprias sejam atingidas. O conceito mais relevante no i\* é o conceito de meta: “*A goal is a condition or state of affairs in the world that an actor would like to achieve*” [Yu 95] e [Yu 01a]. Dessa forma, atores dependem uns dos outros para que metas sejam alcançadas, recursos sejam fornecidos, tarefas sejam realizadas e metas flexíveis sejam “razoavelmente satisfeitas”. Este trabalho aplica o mesmo conceito usado por Chung [Chung 93] de que uma meta flexível (requisito não funcional – RNF) raramente tem um critério definido, o qual indica se ela foi satisfeita ou não.

O i\* lida com dois modelos diferentes e complementares: o modelo SD – modelo de dependências estratégicas e o modelo SR – modelo de razões estratégicas. No modelo SD são expressos exclusivamente relacionamentos de dependência estratégica entre os atores através de uma rede de nós e arestas, onde cada nó representa um ator e cada aresta mapeia uma dependência entre os atores. No modelo SR, que detalha as razões estratégicas internas aos atores, o comportamento de cada ator é modelado através de relações meios-fim (“means-ends”), decomposições de tarefas (“task decompositions”) e contribuições de meta flexível (“softgoal contributions”).

#### Modelo SD:

No Modelo SD cada dependência representa um relacionamento de cooperação entre dois atores, onde um ator chamado de *dependor* depende de um outro chamado de *dependee*. O elo da dependência, chamado de *dependum*, é o objeto da dependência que pode ser: uma meta concreta, uma meta flexível, uma tarefa ou um recurso, e esse é sempre uma entidade física ou informacional. As relações de dependência mostram as vulnerabilidades do *dependor*, pois o *dependee* pode falhar no cumprimento do acordo e, por isso, cada dependência tem um grau de importância (“critical”, “committed” ou

“open”) para refletir sua relevância. A Figura 2.3, retirada de [Oliveira 06a], mostra os quatro tipos de dependências estratégicas possíveis.



**Figura 2.3** – Ilustração da dependência entre atores utilizada em modelos SD

A Figura 2.3 parte (a) faz uma ilustração do significado de uma dependência entre dois atores: um ator “quer” alguma coisa que um outro ator “pode” suprir. Na parte (a) foi representada uma dependência por meta: o CHAIR da conferência (*dependor*) depende do AUTOR (*dependee*) para que “**Artigo seja Submetido**”. A Figura 2.3 parte (b) faz a representação de uma dependência por tarefa. O ator CHAIR (*dependor*) depende do ator REVISOR (*dependee*) executar a tarefa “**Revisar Artigo**”. A Figura 2.3 parte (c) faz a representação de uma dependência por recurso. O AUTOR (*dependor*) depende do CHAIR (*dependee*) fornecer o “**Resultado da Revisão**”. Por último, a Figura 2.3 parte (d) faz a representação de uma dependência por meta flexível. O ator CHAIR (*dependor*) depende do ator REVISOR (*dependee*) executar a revisão com boa qualidade.

Observe na Figura 2.3 que a “letra **D**” indica a direção de dependência. Foram representados os quatro tipos de dependências estratégicas: (a) por meta, (b) por tarefa, (c) por recurso e (d) por meta flexível.

Os tipos das dependências refletem o grau de liberdade existente no relacionamento. Na dependência por meta, cabe ao *dependee* toda e qualquer decisão para o cumprimento da meta. Na dependência por tarefa, o *dependee* executa a tarefa da maneira como o *dependor* deseja e, por isso, cabe ao *dependor* as decisões de como a tarefa deve ser executada. Na dependência por recurso, o grau de liberdade é nulo, ou seja, o *dependee* deve fornecer o recurso exatamente como o *dependor* deseja. Na dependência por meta flexível, o *dependor* tem a decisão final de aceitar ou não a meta alcançada, porém ele usa o benefício do “know-how” (habilidades e conhecimentos) do *dependee*.

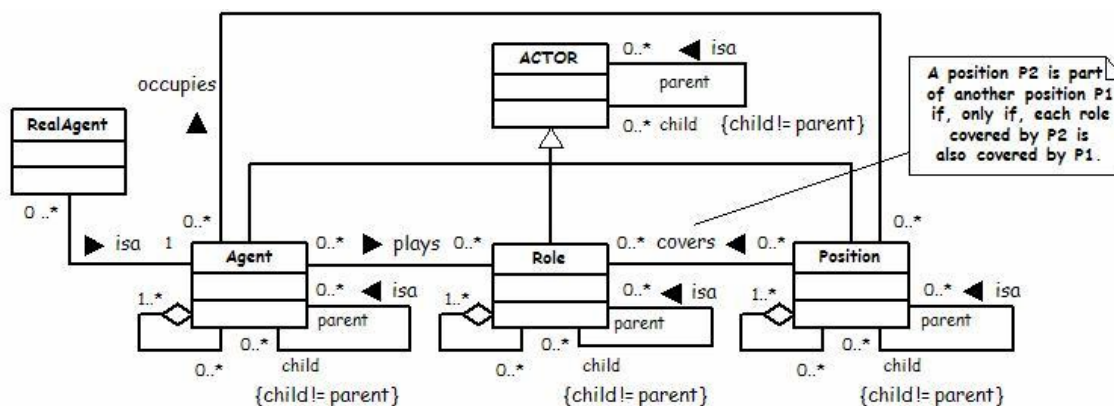
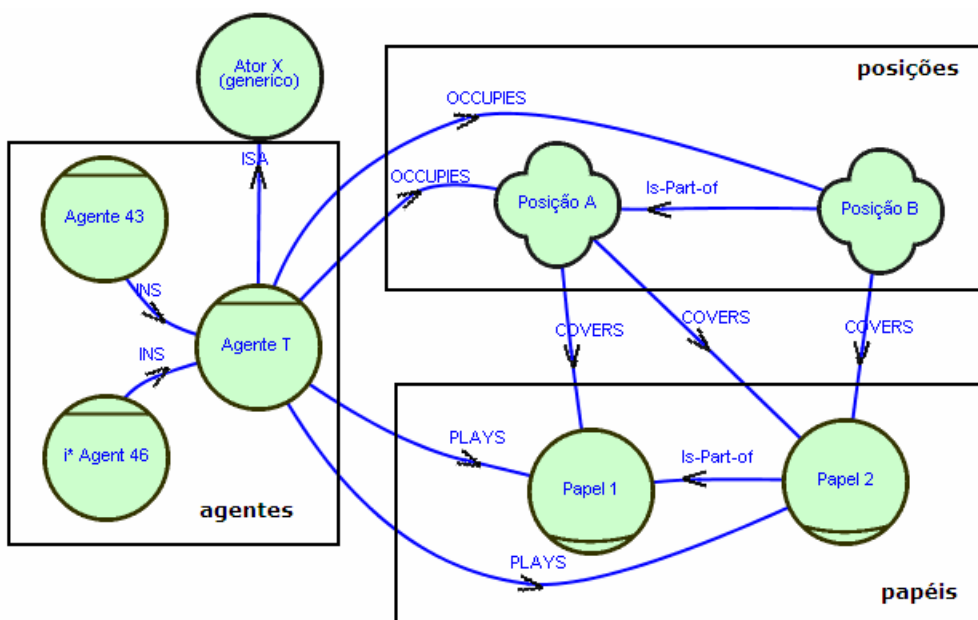


Figura 2.4 – Meta modelo do Modelo SA - adaptado de [Leite 07]

### Modelo SA:

A Figura 2.5 apresenta um esquema do Modelo de Atores Estratégicos, Modelo SA. No i\* um **ator** é uma entidade ativa que executa atividades para atingir metas pelo exercício do "know-how" próprio. Um **agente** (que pode ser uma pessoa) é um ator com manifestações físicas concretas tal qual um ser humano [Yu 95]. Agente, papel e posição são especializações de ator, sendo que um agente ocupa posições e desempenha papéis e uma posição cobre papéis. Uma **posição** é formada pelo conjunto de papéis desempenhados por um agente. Um **papel** é uma caracterização abstrata do comportamento de um ator social. Finalmente, papéis, posições e agentes podem ter partes. Leite et al. [Leite 07] apresenta um estudo detalhado e rigoroso que exemplifica todos esses conceitos sobre a especialização e a representação do Modelo SA.



**Figura 2.5** – Ilustração de um Modelo SA

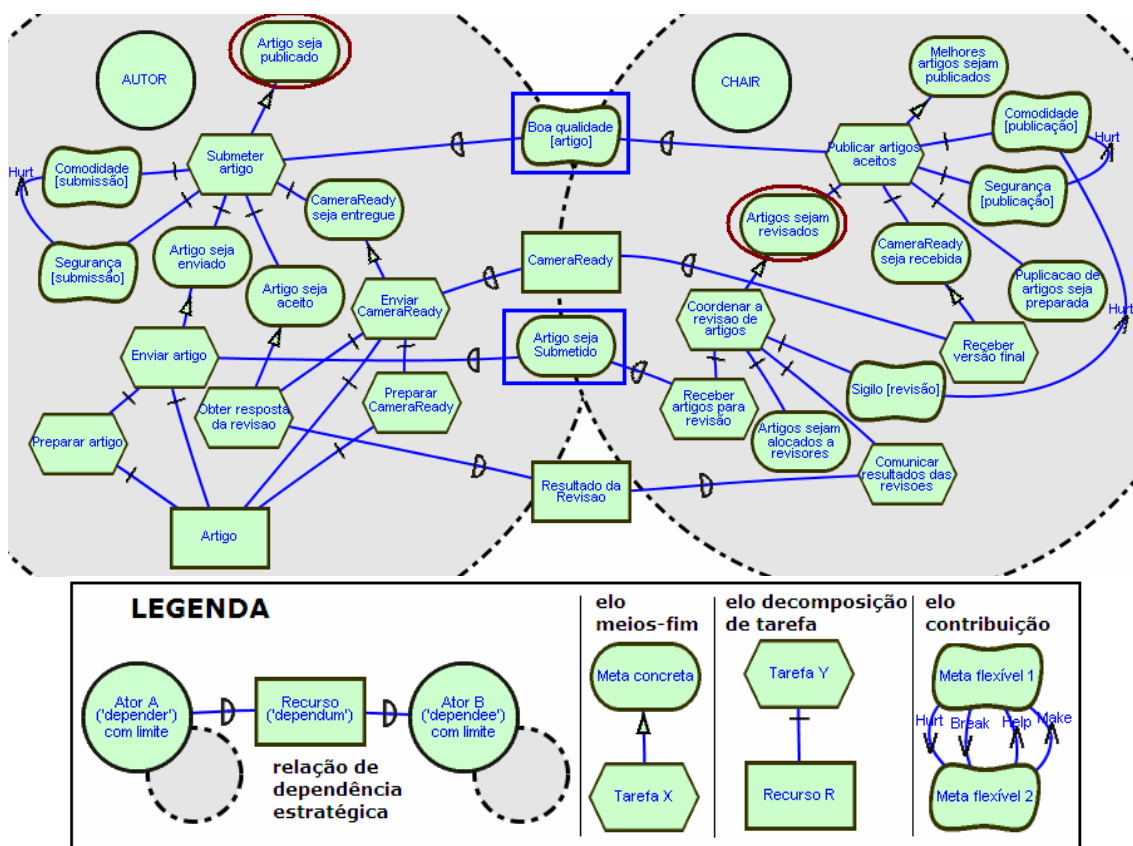
Na parte esquerda da ilustração da Figura 2.5 está representado que o “Agente T” é um **ator** pela relação “is a” (ISA). Do mesmo lado foram representados dois **agentes**, o “Agente 43” e o “i\* Agent 46”, que são instâncias (INS) do “Agente T”. Foi representado que o “Agente T” pode ocupar (OCCUPIES) duas **posições**: a “Posição A” e a “Posição B”, e também que a “Posição B” é parte (is Part of) da “Posição A”. Por outro lado, está representado que “Posição A” cobre (COVERS) os **papéis** “Papel 1” e “Papel 2”, enquanto “Posição B” cobre (COVERS) apenas “Papel 2”.

Na Figura 2.5 não seria necessário representar a relação “desempenha” (PLAYS) porque existe a lei: “Para todo agente que ocupa posição X  $\rightarrow$  agente desempenha os

*papéis cobertos pela posição X*” (Lei do Modelo SA) [Leite 07]. Isso significa que “Agente T” desempenha “Papel 1” e “Papel 2” quando ocupa “Posição A”. Também foi representado que “Papel 2” é subparte de “Papel 1”.

### **Modelo SR:**

O Modelo SR tem como objetivo representar o detalhamento do “rationale” interno de cada ator (inclusive com alternativas) em termos dos elementos do processo. Os elementos do processo usados para essa modelagem são: a **relação meios-fim**, que explicita as decisões que envolvem o alcance das metas do ator; a **decomposição de tarefa**, que detalha como as tarefas são elaboradas e realizadas e como os recursos são utilizados e disponibilizados; além do **relacionamento de contribuição** que, principalmente, explicita a contribuição positiva (+) ou negativa (-) entre metas flexíveis.



**Figura 2.6** – Exemplo de um Modelo SR (parte)

O Modelo SR tem a forma de um “grafo” com alguns tipos de nós e arestas (ou elos). A relação meios-fim é mapeada graficamente por um vetor direcionado para o nó fim, o qual geralmente é uma meta a partir dos nós meio, que são sempre tarefas (mais



de uma tarefa no caso de existirem alternativas mutuamente exclusivas - tipo “xor” - para o alcance da meta). A decomposição de tarefa é mapeada graficamente pela representação dos nós subcomponentes da tarefa, os quais são ligados ao nó principal (a tarefa maior) através de um elo representado por uma reta com um pequeno segmento de reta que corta esse elo. Os nós dos subcomponentes podem ser: metas, tarefas, recursos e metas flexíveis. Esses subcomponentes têm o mesmo significado daqueles já utilizados no Modelo SD. Uma meta é uma condição ou estado de desejo no mundo que o ator pretende alcançar e uma tarefa especifica um modo particular de realizar algo. Quando uma tarefa é especificada como uma subtarefa de outra, a subtarefa restringe a tarefa maior para um curso de ação em particular. Um recurso é uma entidade física ou informacional que não é considerada problemática pelo ator. A principal característica neste caso é se o recurso está disponível, e por quem foi disponibilizado no caso de uma dependência externa. Uma meta flexível é também uma condição no mundo que o ator deseja alcançar, mas diferentemente de uma meta concreta, o critério para essa condição ser atingida não é definido *a priori*, sendo sujeito à interpretação. Quando uma meta flexível é um componente em uma tarefa decomposta, ela serve como um atributo de qualidade para aquela tarefa, guiando ou restringindo a seleção entre alternativas para a decomposição da mesma.

A Figura 2.6 obtida de [Oliveira 06a] é um exemplo de um Modelo SR incompleto. Um Modelo SR completo deve, segundo o Framework i\*, representar todos os atores com os respectivos “rationales”. Como é pouco prático mostrar um Modelo SR completo em uma folha A4, no diagrama está sendo representada parte do “rationale” de apenas dois atores. É possível observar que as dependências estratégicas são também representadas no Modelo SR. Existem diferentes exemplos: do elo meios-fim, da decomposição de tarefa e da contribuição entre metas flexíveis. Na Figura 2.6, o AUTOR tem como meta principal que seu “Artigo seja publicado” e para atingir esse fim o meio é “Submeter artigo”. A tarefa “Submeter artigo” tem na decomposição duas metas flexíveis: “Comodidade” e “Segurança” e três metas: “Artigo seja enviado”, “Artigo seja aceito” e “CameraReady seja entregue”. No diagrama está representada a contribuição negativa (“Hurt”), pois a segurança reduz a comodidade da realização da tarefa. É importante observar que, embora não apareça na tese de Eric Yu [Yu 95], uma meta flexível, quando componente de uma tarefa, também exerce influência sobre todos os subcomponentes da tarefa. Ou seja, “Comodidade” também fica requerida pelas metas “Artigo seja enviado”, “Artigo seja aceito” e “CameraReady seja entregue”.

## 2.2 Técnicas, Métodos e Conceitos Adaptados ou Criados

Nesta seção apresentamos as técnicas, métodos e conceitos elaborados ou adaptados para serem utilizados pelo Método **Eri\*c**. São eles: “Os Conceitos de Ação Concreta e de Ação Flexível”, “As Estruturas Canônicas do Framework i\*”, “O Painel da Intencionalidade - Diagrama IP” e “Diagnósticos i\* (“i\* Diagnoses”)”.

O termo *ação concreta* foi apresentado pela primeira vez por Sá Carvalho em [Carvalho 88], e para ser aplicado nesta tese, foi adaptado para ter um sentido mais amplo. O termo *ação flexível* está sendo criado nesta tese como uma complementação de *ação concreta*, desse modo, sempre que existir uma ação ela será concreta ou flexível. As estruturas canônicas *SDsituation* [Oliveira 07] [Oliveira 08a] e *SRconstruct* [Oliveira 08b], bem como o *Diagrama IP* [Oliveira 08a] e a estratégia *Diagnósticos i\**, [Oliveira 08b] foram elaboradas pelo autor e divulgadas nos trabalhos assinalados nas referências.

### 2.2.1 Os Conceitos de Ação Concreta e de Ação Flexível

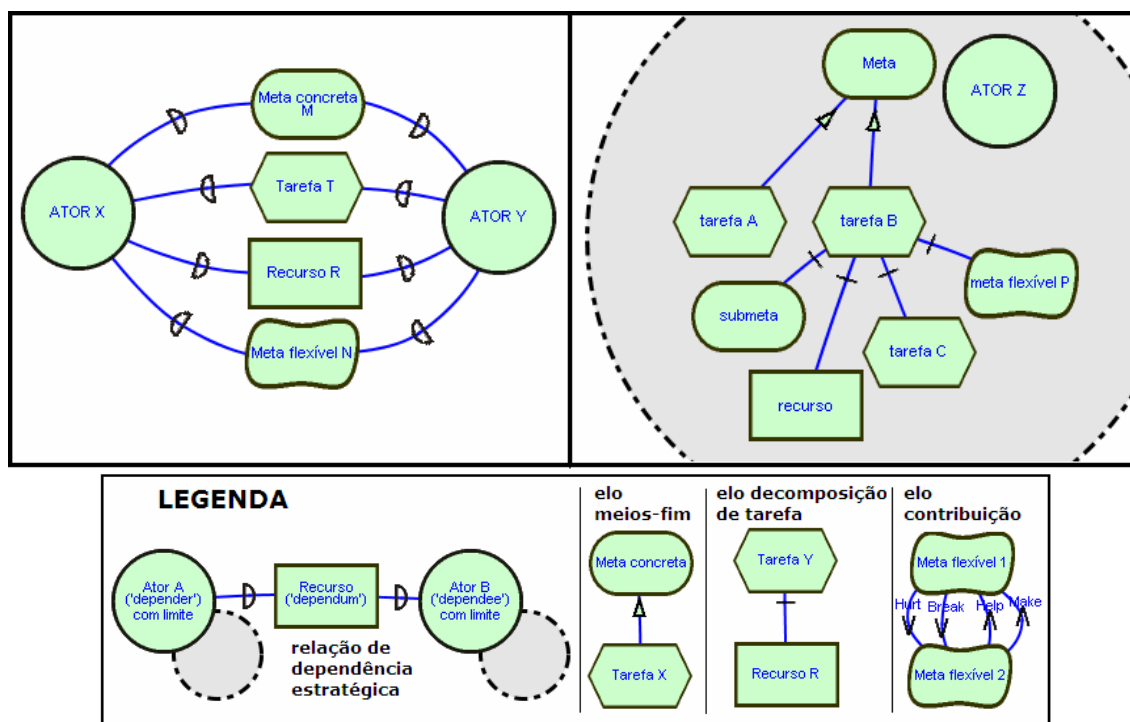
Sá Carvalho em [Carvalho 88] definiu que uma *ação concreta* é uma ação executada por um usuário do Sistema de Informação apoiado pelas informações desse sistema. Segundo Carvalho [Carvalho 88], uma *ação concreta* não pode ser descrita por um verbo pouco preciso (como controlar, apurar, administrar, acompanhar, avaliar, etc.) e sim por verbos ativos concretos bem definidos (comprar, vender, cobrar, multar, produzir, treinar, alocar, etc.) que especificam com clareza a ação final que será apoiada pelo Sistema de Informação (S.I.). Em [Oliveira 94a] foi apresentada uma experimentação dessa idéia em um sistema de software de uma grande empresa. O trabalho mostrou que o uso de ações concretas especifica com clareza a ação final a ser apoiada pelo Sistema de Informação. Para esta tese o termo *ação concreta* engloba não só as ações finais apoiadas pelo S.I. mas toda e qualquer ação concreta executada dentro do contexto da organização em estudo, ou seja, inclui também ações concretas intermediárias que são executadas no UdI [Oliveira 94b].

Neste trabalho, a qualificação do termo flexível em *ação flexível* tem o mesmo significado ou interpretação que o usado para metas flexíveis (“softgoals”). *Ações flexíveis* são ações pouco precisas, para as quais não se pode identificar um resultado concreto *a priori* e, além disso, a confirmação da execução da *ação flexível* pode

dependem de interpretação. Uma *ação flexível* (exemplos de “ações flexíveis”: analisar, apurar, avaliar, agilizar, conferir, controlar, gerenciar, verificar, validar, etc.) é diferente de uma *ação concreta* (exemplos de “ações concretas”: comprar, pagar, planejar).

### 2.2.2 As Estruturas Canônicas do Framework $i^*$

O Framework  $i^*$  [Yu 95] modela contextos organizacionais baseados no relacionamento de dependência entre atores. A idéia central do  $i^*$  é representar através de modelos os atores e as dependências que os atores têm uns com os outros para que metas próprias sejam atingidas.



**Figura 2.7** - Ilustração das estruturas canônicas do  $i^*$

(esquerda: SDsituation – direita: SRconstruct)

Podem ser observadas duas estruturas básicas que sempre estão presentes em modelos  $i^*$ . A primeira aparece em Modelos SD e a segunda aparece nitidamente em Modelos SR. Elas são respectivamente: a situação de dependência estratégica – SDsituation [Oliveira 06b] e o construto de razão estratégica – SRconstruct [Oliveira 08b]. A Figura 2.7 ilustra esses conceitos. A apresentação detalhada dessas estruturas do Framework  $i^*$  é feita nas seções 2.2.2.1 e 2.2.2.2 a seguir.

No Modelo SR da Figura 2.6 [Oliveira 06a] foram indicados com formas ovais dois SRconstructs, um no limite do ator AUTOR e o outro no limite do ator CHAIR.

São eles: “Artigo seja publicado” e “Artigos sejam revisados”. Com retângulos foram marcadas as dependências que fazem parte de uma SDSituation chamada de SUBMISSÃO DE ARTIGO (“Article Submission”).

### 2.2.2.1 Situações de Dependência Estratégica - SDSituations

Definição: *Uma SDSituation é uma estrutura de dependências estratégicas mútuas entre atores de um contexto organizacional, uma SDSituation define um bloco de elementos de dependência com intencionalidade situacional compartilhada.*

Da definição entendemos que uma SDSituation é uma parte, ou um bloco, de um modelo intencional maior, que através da reciprocidade de dependências estratégicas entre atores, tem a capacidade de concluir uma meta situacional.

O lado esquerdo da Figura 2.7 mostra que uma SDSituation é delimitada pelos múltiplos elementos de dependência que podem existir entre dois ou mais atores em uma Situação de Dependência Estratégica.

Situações de dependência estratégica ocorrem no ambiente organizacional e o ponto central de SDSituations é que cada elemento de dependência (***meta***, ***meta flexível***,  ***tarefa*** ou ***recurso***) que envolve os atores não está isolado, cada dependência é parte de uma situação de colaboração bem definida chamada de “situação de dependência estratégica” ou SDSituation [Oliveira 06b].

Uma SDSituation pode ser identificada separadamente das outras SDSituations formando uma cadeia de interdependências entre essas situações. Uma SDSituation pode ser caracterizada como parte da unidade de negócio. É necessário identificar diversas SDSituations separadas, apesar de cada uma depender das outras criticamente. Por exemplo: um processo eleitoral somente pode ser executado se os candidatos tiverem sido definidos previamente. Isso significa que é necessário identificar duas situações separadas, mas uma depende da outra criticamente: A ELEIÇÃO DE CANDIDATOS depende da DEFINIÇÃO DOS CANDIDATOS.

Uma SDSituation é temporalmente limitada e as interdependências entre elas podem ser do tipo física, lógica ou temporal, e elas podem ser representadas em um diagrama específico [Oliveira 06b].

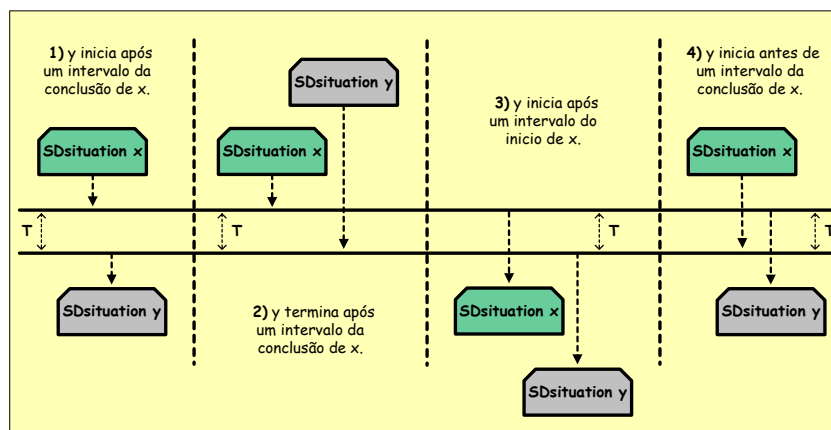
As SDSituations podem ser percebidas em modelos SR, porém são identificadas mais facilmente no modelo SD.

Como descrito anteriormente, existem três tipos de interdependências entre SDsituations: física, lógica ou temporal, entretanto mais de um tipo de interdependência pode ocorrer simultaneamente. Os três tipos de interdependências entre SDsituations estão descritos a seguir:

- ♦ Física - quando um recurso é preparado por uma SDsituation e requisitado por outra SDsituation;
- ♦ Lógica - quando uma ou mais SDsituations necessitam da conclusão de outra SDsituation para a iniciação, ou quando uma ou mais SDsituations necessitam da conclusão de outra SDsituation para a conclusão.
- ♦ Temporal – quando uma ou mais SDsituations necessitam esperar algum tempo após o início de outra SDsituation, ou quando uma ou mais SDsituations necessitam esperar algum tempo após a conclusão de outra SDsituation.

A Figura 2.8 [Oliveira 06b] mostra as ligações temporais possíveis. Essas ligações podem ser consideradas como uma especialização da ligação lógica. Uma ligação temporal se torna ligação lógica quando o intervalo de tempo é nulo. A figura mostra cada caso em que:

1. A SDsituation y para ser iniciada precisa esperar um intervalo “T” após a conclusão da SDsituation x.
2. A SDsituation y para ser terminada precisa esperar um intervalo “T” após a conclusão da SDsituation x.
3. A SDsituation y para ser iniciada precisa esperar um intervalo “T” após o início da SDsituation x.
4. A SDsituation y precisa iniciar um intervalo “T” antes da conclusão da SDsituation x.



**Figura 2.8** – As quatro variantes de interdependências lógica e temporal

Ligações de dependência entre as SDSituations são de vital importância para a análise organizacional. Considere, por exemplo, um “Sistema de Gerenciamento de Pedidos”. A SDSituation ENTREGA DO PEDIDO deve somente acontecer depois da SDSituation ENCOMENDA DO PEDIDO, porque existe uma ligação lógica. A SDSituation PAGAMENTO DO PEDIDO poderia acontecer antes da SDSituation ENTREGA DO PEDIDO mas nunca antes ou sem que a SDSituation ENCOMENDA DO PEDIDO tivesse acontecido.

No estudo de caso, quando modelamos o sistema “Expert Committee”, nós tivemos que responder a perguntas “por quê” tais como: Por que os atores submetem artigos? Por que os revisores revisam artigos? Por que os coordenadores indicam revisores? Por que pesquisadores aceitam ser revisores? Respostas a perguntas como as anteriores podem ser obtidas com mais facilidade quando o engenheiro de requisitos está trabalhando com uma “situação de negócio” bem definida.

A Figura 2.9 mostra o relacionamento entre os componentes da SDSituation. Uma SDSituation reflete somente uma “Situação de Dependência Estratégica”, que pode ser formada por uma única dependência (com um elemento) ou por mais dependências. Os elementos da dependência podem ser: metas, tarefas, recursos ou metas flexíveis. Adicionalmente uma meta flexível qualifica a SDSituation do mesmo modo que uma meta flexível qualifica os elementos da dependência, significando que os elementos da dependência não podem ser considerados sem aquela qualificação. A qualificação através de uma meta flexível lembra ao engenheiro de requisitos que uma operacionalização futura precisará ser desenvolvida [Chung 00].

O diagrama da Figura 2.9 também mostra que atores (dependers e dependees) participam da SDSituations e cada SDSituation possui um grau de relevância (crítico, mediano ou indiferente). A elicitação de que as SDSituations possuem ligações importantes permite ao engenheiro de requisitos e aos interessados descobrir metas de nível mais alto. A questão a ser respondida é: “Por que estas SDSituations precisam estar ligadas?”.

A identificação das SDSituations e da cadeia de dependências entre elas traz vários benefícios; são eles [Oliveira 06b]:

(a) A descoberta das SDSituations antes de modelá-las é uma boa maneira de lidar com a complexidade, porque desse modo não precisamos lidar com todas as dependências ao mesmo tempo.

(b) As relações de dependência entre os atores guiam o engenheiro de requisitos para a identificação de metas individuais e metas compartilhadas.

(c) A validação dos requisitos usando uma representação que os interessados possam ler é útil porque estes sentem maior conforto quando usam modelos centrados em linguagem natural. Além disso, a validação pode ser adaptada para que as SDSituations utilizem mais de um ponto de vista (pontos de vista de dependers e dependees).

(d) As SDSituations podem ajudar a Gerência de Requisitos mantendo a rastreabilidade (“backward-from & forward-to”) durante o processo de elicitação e também podem fazer parte de uma “baseline”, Leite e Oliveira [Leite 95], de maneira que possibilite o registro da evolução dos requisitos.

(e) Nas SDSituations percebe-se com clareza que os atores podem ter especializações como agentes, papéis e posições [Cunha 07] e [Leite 07].

(f) As SDSituations podem ter ligações tipo “bridge” umas com as outras. “Uma “bridge” é uma camada de dependência entre domínios” [Sztipanovits 97] [Mellor 02], o que significa que uma SDSituation pode ser implementada em um ambiente diferente de outra SDSituation, por exemplo: uma na Web e a outra em um dispositivo móvel.

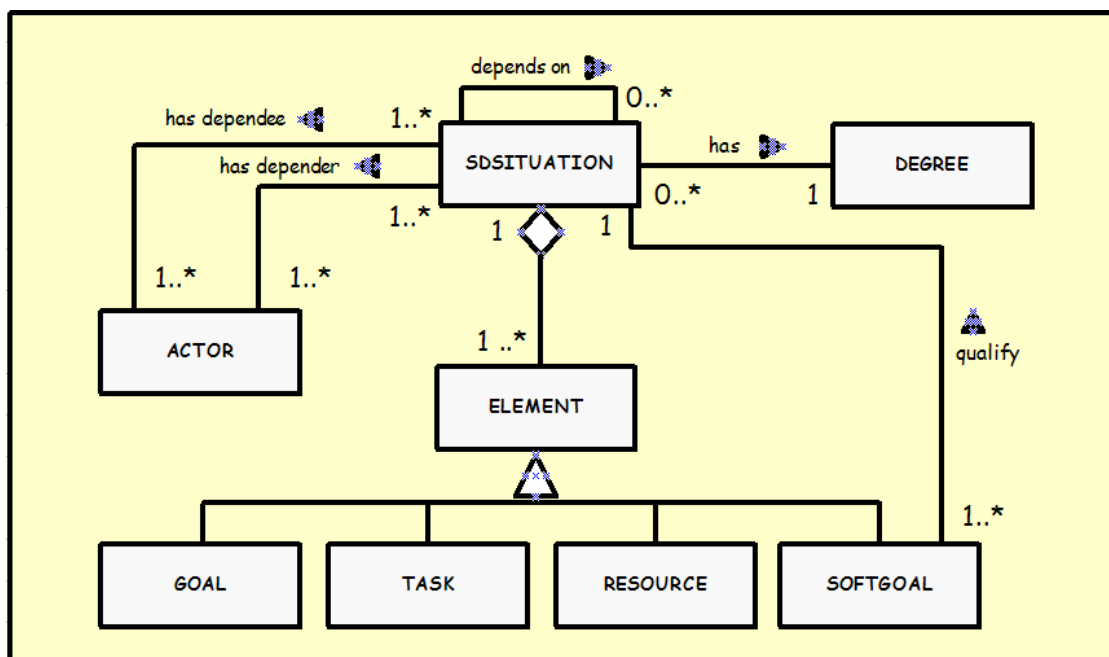


Figura 2.9 – Diagrama de Classes da SDSituation

### 2.2.2.2 Construtos de Razão Estratégica - SRconstructs

Definição: *Um SRconstruct é uma parte da estrutura do “rationale” de um ator, e modela os elementos necessários para o alcance da meta alvo do próprio construto.*

O lado direito da Figura 2.7 ilustra a estrutura básica do SRconstruct. O SRconstruct foi criado com base nos conceitos do modelo SR.

Os construtos de razão estratégica aparecem dentro da linha de fronteira do ator (a linha limítrofe pontilhada). Um SRconstruct é formado por uma meta (como o fim “end”), por pelo menos uma tarefa (como o meio de atingir o fim) e, além da meta e das “tarefas meio”, todos os componentes (ou subcomponentes) necessitados pelas “tarefas meio” que fazem parte da estrutura. Os subcomponentes são: tarefas, recursos, metas flexíveis e metas concretas que formam a decomposição da tarefa-meio. Apesar de a meta ser somente uma parte do SRconstruct, nós identificamos cada SRconstruct pelo nome da meta que ele preenche. Isso porque existe somente uma meta como sendo o fim em cada SRconstruct. O ponto central do SRconstruct é que em cada unidade de um SRconstruct existem, de modo a alcançar uma determinada meta, os componentes do SRconstruct que estão estruturalmente ligados para ajudar os meios a atingir o fim da estrutura.

### 2.2.3 O Painel da Intencionalidade - Diagrama IP

Nosso método propõe que a intencionalidade seja mapeada em partes, guiada pelas SDSituations [Oliveira 06b]. Propõe que, primeiramente, só a intencionalidade seja representada, ou seja, sejam representadas exclusivamente as metas em um diagrama chamado *Painel de Intencionalidade* (“Intentionality Panel”) ou *Diagrama IP*.

O Diagrama IP é uma redução do Modelo SR. Nele são considerados somente os atores, as metas (concretas e flexíveis) e as relações entre elas. O Diagrama IP é um diagrama de transição de estados que também representa os atores. Ele é um diagrama de estados por ter estados (que são as metas) conectados pelas transições [Oliveira 07].

Além da simplificação se comparado a um diagrama de transição de estados convencional, a principal motivação da criação do Diagrama IP é a representação da intencionalidade em um único e homogêneo diagrama. O Diagrama IP, além de não representar tarefas e recursos, que aparecem no Modelo SR, adota também uma simplificação se comparado ao diagrama de estados convencional, já que a notação da



transição de estados foi suprimida pela adoção de uma convenção que retrata somente três tipos de transição (a correlação, a contribuição e a dependência; veja a contextualização na seção a seguir), os quais são semanticamente os mesmos adotados por Modelos SR.

A idéia não é substituir os Modelos SR, mas apenas usar um diagrama preliminar para dar maior ênfase à intencionalidade. Nossa proposta, como está detalhada nas próximas etapas do método, tem a intenção de minimizar a dificuldade de compreensão de Modelos SR [Pastor 06] e [Estrada 06]. De fato, o Modelo SR é complexo por natureza, pois ele modela um ambiente organizacional inteiro em apenas um diagrama. Algumas abordagens já trouxeram idéias para particionamento do Modelo SR como de [Zheng 04] e [Alencar 07], que são apoiadas pelo conceito de visões, diferentemente de nossa proposta, que defende a idéia do Modelo SR ser dividido em vários sub-modelos de situações de dependência estratégica. Seguindo a mesma idéia do particionamento do Modelo SR, Oliveira et al. [Oliveira 07] indica que se deve elaborar um Diagrama IP para cada SDsituation.

A apresentação detalhada do relacionamento entre as metas e do Diagrama IP é feita, respectivamente, nas seções 2.2.3.1 e 2.2.3.2 a seguir.

### **2.2.3.1 As Relações entre as Metas**

As relações entre as metas aparecem de maneira indireta em modelos SD e de maneira direta ou explícita em Modelos SR. Em Modelos SR podemos observar quatro tipos principais de relações entre metas (concretas e flexíveis). Elas foram chamadas de relação de correlação, de contribuição, de dependência e de equivalência e foram usadas para representar as ligações entre as metas em Diagramas IP - Painéis de Intencionalidade.

Um Diagrama IP é composto por nós (ou vértices) e arestas (ou arcos). No Diagrama IP as metas concretas e as metas flexíveis são os nós e os quatro tipos de relações entre as metas são as arestas. Atores foram incluídos no diagrama apenas para indicar os proprietários das metas.

Tomando o Framework  $i^*$  como base, pode-se observar que nele existem relações entre as metas. São três as variantes da relação de correlação {(1) meta concreta para meta concreta, (2) meta flexível para meta concreta e (3) meta concreta para meta

flexível}; uma para tipo contribuição {meta flexível para meta flexível} e uma para tipo dependência {meta concreta para meta concreta de atores diferentes}.

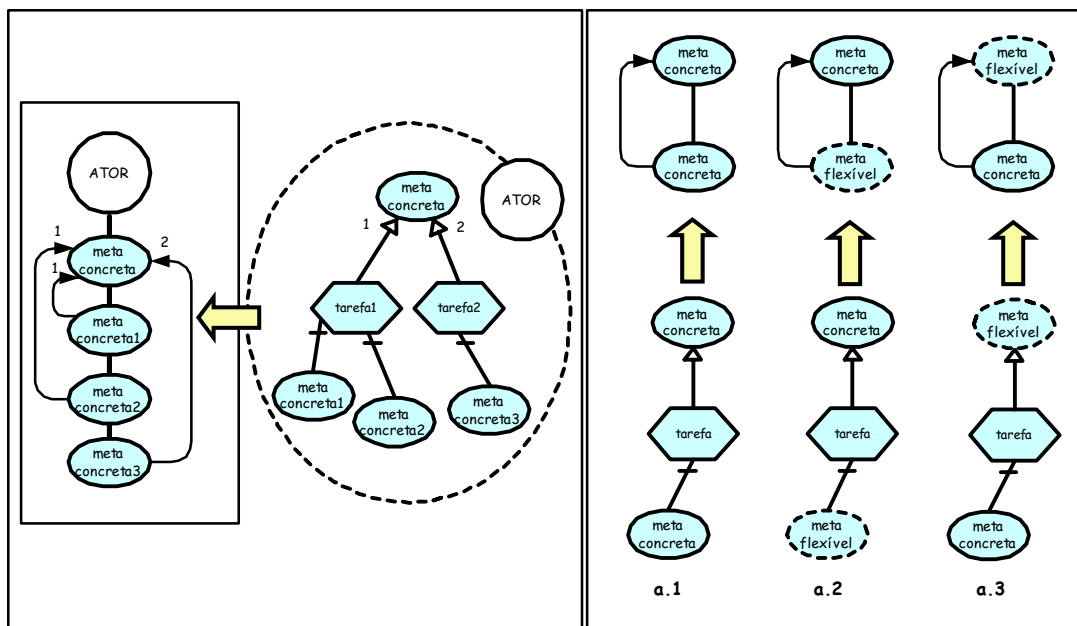


Figura 2.10 – Ilustração da relação de correlação entre as metas

Na parte direita da Figura 2.10 [Oliveira 07] estão representadas as correspondências de notação entre o Modelo SR e o Diagrama IP para a relação de **correlação**. Na parte esquerda, alternativas meios-fim estão representadas: enquanto por uma alternativa, as duas metas concretas (meta concreta1 e meta concreta2), juntas, possuem correlação com a meta concreta principal, por uma segunda alternativa, meta concreta3 tem uma correlação representada para a meta flexível principal a ser atingida. Nesse exemplo as correlações alternativas foram identificadas com 1 e 2 para mostrar que uma independe da outra.

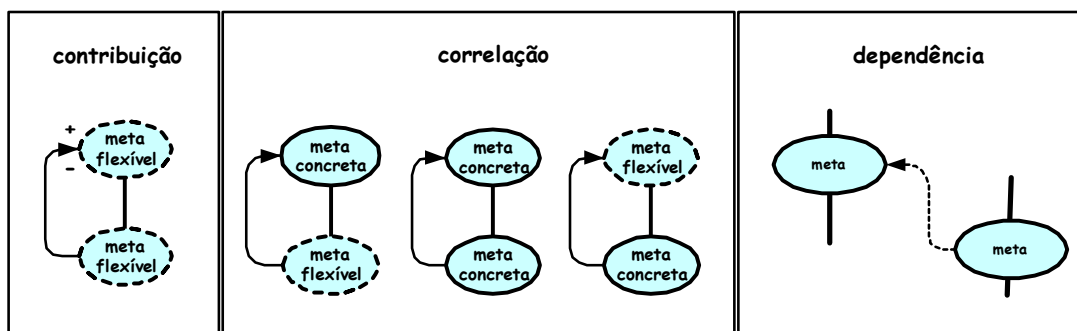


Figura 2.11 – Ilustração das relações entre as metas em um diagrama IP

a) **Correlação** entre duas metas:

A relação de correlação ocorre entre duas metas de um mesmo ator. Uma das metas, chamada de meta inicial, é subcomponente da tarefa, que é o meio para a meta principal. A correlação indica que o sucesso da meta principal é influenciado pelo alcance da meta inicial.

a.1) **Caso de correlação entre duas metas concretas:**

A meta concreta principal possui a meta concreta como um subcomponente da tarefa-meio.

a.2) **Caso da correlação entre uma meta flexível e uma meta concreta:**

A meta concreta principal possui a meta flexível como um subcomponente da tarefa-meio.

a.3) **Caso da correlação entre uma meta concreta e uma meta flexível:**

A meta flexível principal possui a meta concreta como um subcomponente da tarefa-meio.

b) **Contribuição** de meta flexível para meta flexível:

A relação de contribuição ocorre entre duas metas flexíveis de um mesmo ator. Uma das metas, chamada de meta inicial, pode contribuir de forma positiva (+), neutra (?) ou negativa (-) para a meta flexível principal. A contribuição afeta a meta flexível principal que é influenciada pela meta flexível inicial, com a mesma semântica usada em Modelos SR.

Na Figura 2.11 estão ilustradas: à direita, a ***dependência*** entre duas metas concretas de dois atores; ao centro, as três variações possíveis da relação de ***correlação*** entre as metas de um mesmo ator; e à esquerda, a relação de ***contribuição*** entre duas metas flexíveis de um mesmo ator.

c) **Dependência** entre duas metas:

A relação de dependência ocorre entre duas metas de atores diferentes. Ela representa a necessidade do suprimento de uma dependência entre dois atores exatamente com a mesma semântica usada no Modelo SD. As dependências embutidas e não-representadas na relação podem ser uma tarefa, um recurso, uma meta flexível ou uma meta concreta, e serão mapeadas posteriormente em outra etapa do método. A relação de dependência indica que o sucesso da meta

principal (do “depende”) é influenciado pelo alcance da meta inicial (do “dependee”).

d) **Equivalência** entre metas concretas ou entre metas flexíveis:

Não significa uma transição de estados. Sinaliza simplesmente, quando esse relacionamento existe, que as metas flexíveis ou as metas concretas são equivalentes para atores diferentes.

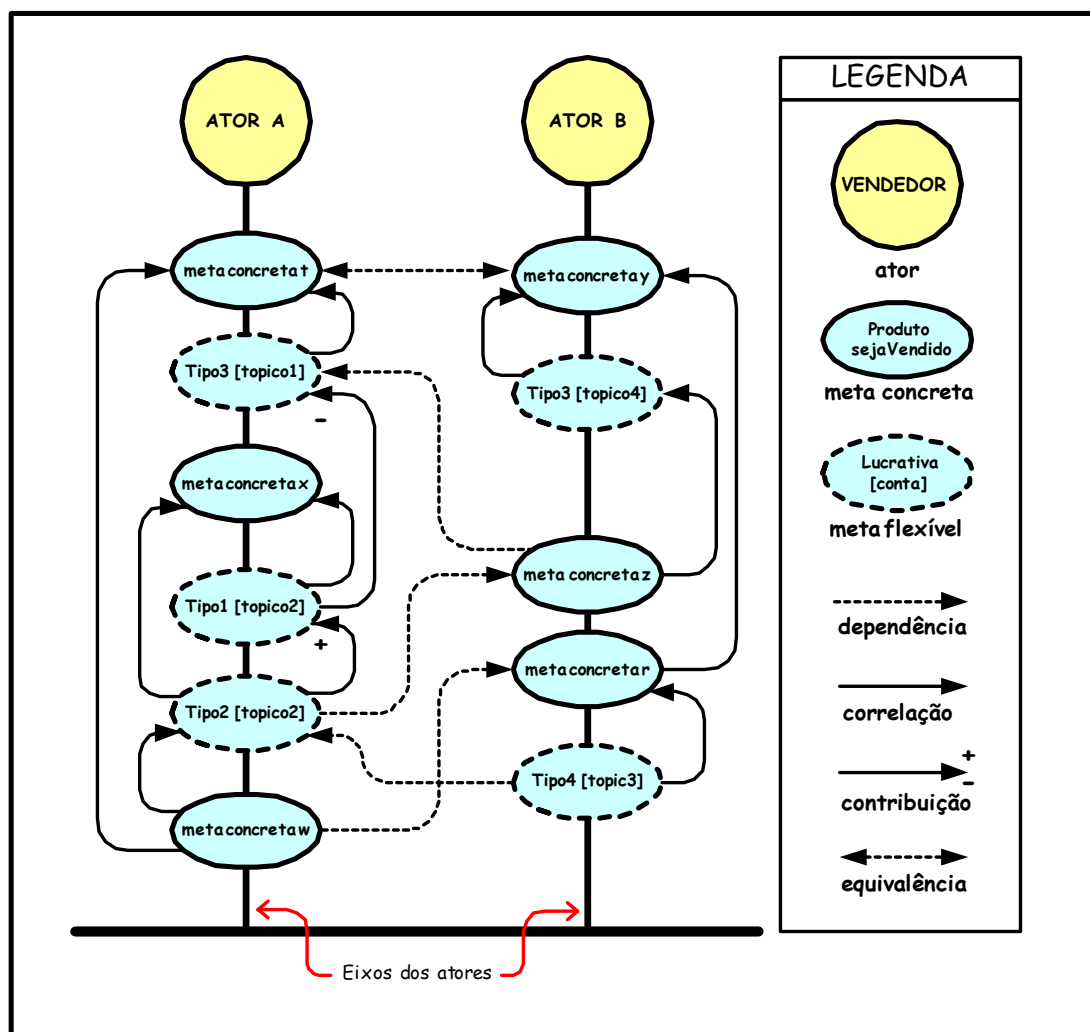


Figura 2.12 – Ilustração de um Painel de Intencionalidade

Na Figura 2.12 estão ilustrados todos os tipos de relação possíveis entre as metas. Observe que apenas para se manter o controle da complexidade, cada Diagrama IP deve retratar apenas uma SDsituation.

### 2.2.3.2 Diagrama de Classes do Diagrama IP

A Figura 2.13 apresenta as relações entre os elementos do Diagrama IP - “Intentionality Panel” através de um diagrama de classes.

O Diagrama IP é formado pelos atores e pelas metas desses atores. Cada meta de um ator é representada em apenas um diagrama, enquanto os atores podem aparecer em diversos.

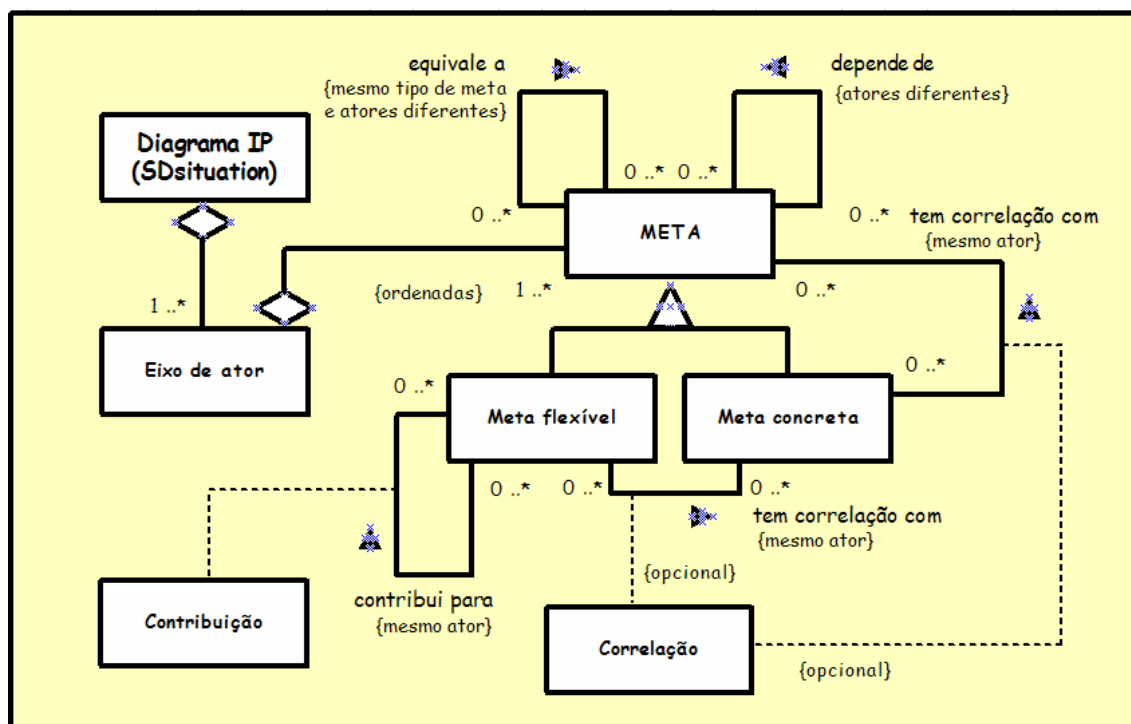


Figura 2.13 – Diagrama de classes do Painel de Intencionalidade

As metas de cada ator são representadas de forma cronologicamente ordenada em um eixo de metas com a identificação do ator. As metas podem ser de dois tipos: meta flexível ou meta concreta. Uma meta flexível pode dar ou receber contribuição (positiva ou negativa) de outras metas flexíveis, sendo que nessa associação todas as metas flexíveis pertencem ao mesmo ator (mesmo eixo). Uma meta concreta pode estar correlacionada a diversas metas flexíveis, assim como a diversas metas concretas. Uma meta flexível pode estar correlacionada a diversas metas concretas. Nos três casos todas as metas pertencem ao mesmo ator (mesmo eixo). Uma meta concreta pode depender de outra meta concreta, porém elas não pertencem ao mesmo ator.

Observe que os dois tipos de metas podem ser representados no Diagrama IP e que as metas são organizadas em SDsituations através do “Eixo de ator” porque um

conjunto de metas interconectadas define uma SDsituation. As metas dos atores podem ter correlação entre elas e a correlação de uma meta pode mostrar as alternativas através de uma identificação, quando existirem alternativas. Quando uma meta flexível contribui para outra meta flexível, é necessário indicar o tipo da contribuição (+, -). Dependências estratégicas entre metas de atores são representadas por uma associação “depends on”.

## 2.2.4 Diagnósticos i\* (“i\* Diagnoses”)

A maioria de trabalhos com o Framework i\* focaliza o aspecto da modelagem e como usar os modelos i\* nos estágios de produção de software. O foco dessa seção é a análise de modelos i\* com a intenção de melhorar a qualidade.

Este capítulo propõe uma abordagem aplicada às técnicas de verificação e validação já existentes na engenharia de software. A abordagem aplica a técnica de leitura e a de revisão “tipo walkthrough”, ambas usadas para inspeção de software. A abordagem considera o “inquiry based requirements analysis” [Potts 94] para o início do processo.

### 2.2.4.1 A Motivação para Diagnósticos i\*

Como processo de qualidade, o foco é prevenir em vez de procurar defeitos na fase de testes. Assim, a técnica é de análise com a finalidade de ter qualidade nos modelos. Apesar de o termo “análise” significar validação e verificação, nossa abordagem é de verificação.

São duas as motivações para a elaboração de um framework de diagnósticos i\*:  
(a) não existe um guia para a avaliação da qualidade de modelos i\* e, conseqüentemente, (b) não existe um acordo com base no bom senso sobre as melhores práticas para a elaboração de modelos i\*.

O objetivo de alcançar melhores modelos i\* contextualiza nosso framework como pertencente à área de avaliação da qualidade da modelagem conceitual. Lindland et al. [Lindland 94] é uma referência nessa área. Nesse trabalho os autores defendem a importância de frameworks de qualidade para modelos conceituais, além de argumentarem que os frameworks devem ser mais específicos para não serem modelos gerais de qualidade de software, centrados em uma lista de atributos de qualidade desejados.

Considerando o framework conceitual de Moody [Moody 05], percebemos que a nossa técnica poderia ser classificada como analítica “Analytical (Synthesis)” já que partimos de estratégias de qualidade conhecidas. Não estamos propondo, como classifica Moody [Moody 05], um “common standard for conceptual model quality”, e sim uma estratégia orientada ao Framework  $i^*$  com ênfase em um procedimento de avaliação.

Estamos apenas preocupados com uma avaliação específica de qualidade para modelos  $i^*$ , diferente de Easterbrook et al. [Easterbrook 05], que usou um processo orientado a pontos de vista para construir modelos  $i^*$  e comparou-o a outro não orientado. Esses autores relataram que o processo de comparação e casamento de pontos de vista trata um nível detalhado de entendimento no domínio do problema e melhora a possibilidade de rastreamento de requisitos para trás. Essas afirmações confirmam a tese de Leite [Leite 88] de que a análise de pontos de vista é uma efetiva maneira de encontrar “deltas”, que são os “feedbacks” para conduzir o processo de verificação da qualidade. Entretanto, como os autores afirmaram, o processo foi extremamente demorado com tendência a fazer uma análise superficial das escolhas de modelagem, uma vez que não foi usada uma ferramenta automatizada para analisar pontos de vista

[Leite 88], porém uma ferramenta de software facilitaria muito o processo.

Para a organização das atividades do processo de melhoria de qualidade adotamos o conhecido framework padrão usado na garantia da qualidade, o ciclo PDCA [Shewhart 39].



A figura ao lado foi obtida de <http://en.wikipedia.org/wiki/PDCA>.

No primeiro estágio (P) do ciclo PDCA são feitos o plano e a preparação da atividade, no segundo estágio (D) a atividade de aplicação das perguntas do diagnóstico é realizada, no terceiro estágio (C) a atividade de verificação toma lugar e no estágio (A) a atividade de correção dos problemas potenciais (alteração de modelos e especificações) ficou representada pela “volta” nas etapas, Figura 3.1 do Método **Eri\*c**.

Modelos  $i^*$  são normalmente muito grandes e complexos [Pastor 06]. O modelo SD representa em um único diagrama todos os atores do contexto organizacional e todas

as dependências que existem entre esses atores. O modelo SR é ainda maior e mais complexo porque além da representação de todos os tipos já representados no modelo SD, o modelo SR retrata também o detalhamento do “rationale” que cada ator carrega. Assim, para tornar possível o desenvolvimento de diagnósticos, nossa proposta adota a estratégia de particionar ou subdividir cada um dos dois modelos  $i^*$  em pequenas e significativas estruturas que tornem possível a aplicação dos diagnósticos.

As *SDsituations* que povoam o modelo SD e os *SRconstructs* presentes no modelo SR são estruturas canônicas pois são composições intencionais elementares (unidades intencionais) que aparecem repetidamente nos modelos  $i^*$ . O desmembramento de modelos  $i^*$  nessas estruturas ajuda o engenheiro de requisitos a apreender mais facilmente modelos complexos. As estruturas canônicas do Framework  $i^*$  são as unidades básicas do processo de diagnósticos. O framework de investigação foi denominado Diagnósticos  $i^*$  (“ $i^*$  Diagnoses”).

Aplicar a estratégia de Diagnósticos  $i^*$  traz diversos benefícios, entre eles podemos salientar: (i) encontrar as *SDsituations* do “melhor” sistema desejado; (ii) elaborar um esquema de priorização dos requisitos de modo a criar um plano para a implementação e entrega do sistema; (iii) verificar sistematicamente se o desenho do sistema está de acordo com a construção do sistema certo.

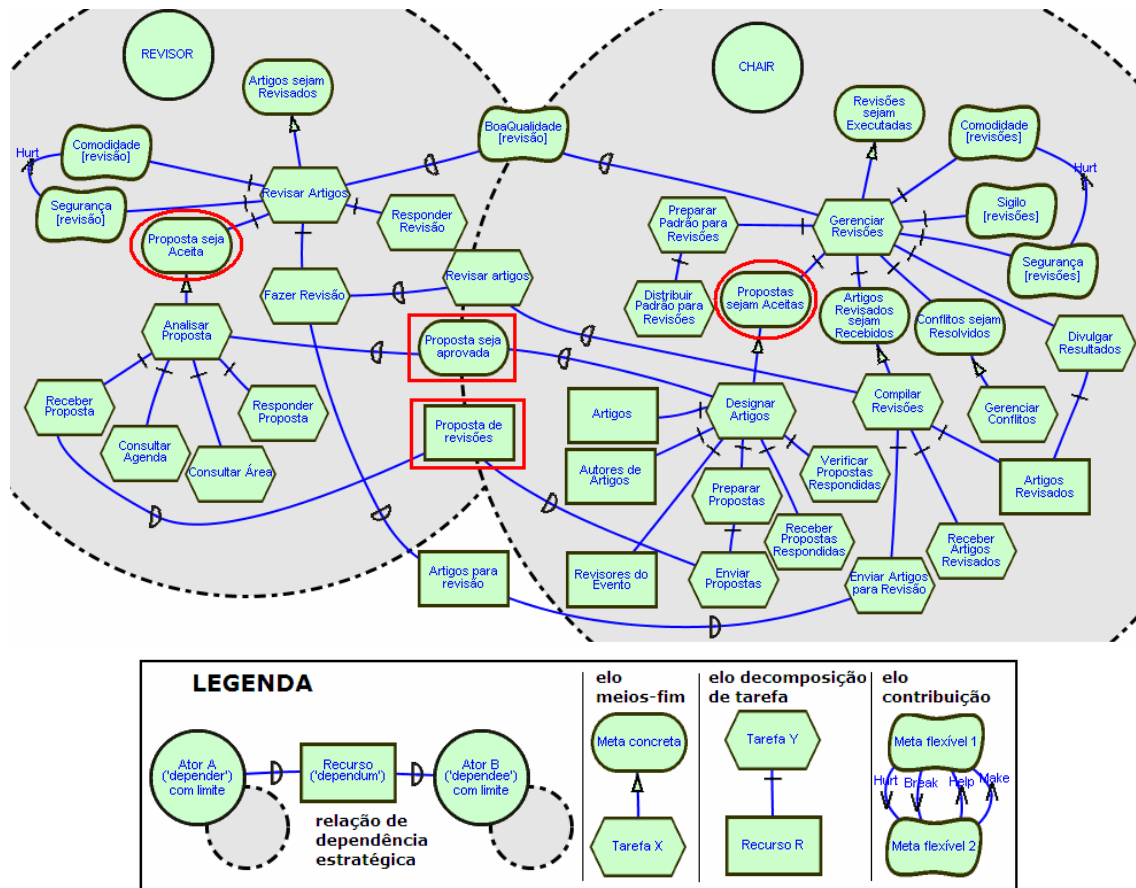
Além desses benefícios, Diagnósticos  $i^*$  produz resultados. Podemos dizer que ajuda a encontrar novos atores, ou especializações desses, novas dependências, problemas e deficiências em dependências, deficiências dentro do “rationale” dos atores e idéias de melhorias tecnológicas e operacionais para o sistema desejado.

No modelo SR da Figura 2.14 [Oliveira 06a] foram indicados com formas ovais dois *SRconstructs*, um no limite do ator REVISOR (PROPOSTA SEJA ACEITA) e o outro no limite do ator CHAIR (PROPOSTAS SEJAM ACEITAS). Com retângulos foram marcadas duas dependências que fazem parte de uma *SDsituation* chamada de **Aceitação de Propostas** (“PROPOSALS ACCEPTANCE”). Essas três estruturas estão inter-relacionadas, possuem forte acoplamento, pois os elementos de dependência que formam a *SDsituation* são compartilhados pelos dois *SRconstructs*.

O *SRconstruct* mostrado do lado esquerdo PROPOSTA SEJA ACEITA é um construto do ator REVISOR (“REVIEWER”). Ele tem uma meta concreta principal (fim), que dá o nome ao *SRconstruct* e uma única tarefa (meio) Analisar Proposta para alcançar a meta concreta principal. Essa tarefa tem uma submeta concreta Proposta seja aprovada, e quatro subtarefas: Receber Proposta, Consultar Agenda, Consultar Área e Responder



Proposta. Duas metas flexíveis *Comodidade [revisão]* e *Segurança [revisão]* estão ligadas à tarefa Revisar Artigos, também devem ser consideradas para o SRconstrut porque a meta seja concreta principal do SRconstrut é um dos componentes daquela tarefa.



**Figura 2.14** - Modelo SR (parte) - uma SDsituation e dois SRconstructs

O SRconstruct que aparece do lado direito PROPOSTAS SEJAM ACEITAS pertence ao ator CHAIR. O SRconstruct tem uma meta concreta principal (fim) que dá o nome ao SRconstruct e uma única tarefa (meio) Designar Artigos para alcançar a meta concreta. Essa tarefa possui três subtarefas Preparar Propostas, Receber Propostas Respondidas e Verificar Propostas Respondidas e três recursos Artigos, Autores de Artigos e Revisores do Evento. A tarefa Preparar Propostas possui uma subtarefa denominada Enviar Propostas. As três metas flexíveis: *Comodidade [revisões]*, *Sigilo [revisões]* e *Segurança [revisões]*, ligadas à tarefa Gerenciar Revisões, também devem ser consideradas para o SRconstruto porque a meta concreta principal do SRconstruto é um dos componentes daquela tarefa.

A SDsituation **Aceitação de Propostas** (“PROPOSALS ACCEPTANCE”) é delimitada externamente pelas dependências estratégicas Proposta seja aprovada e Proposta de Revisões. O REVISOR depende de o CHAIR disponibilizar o recurso Proposta de Revisões e CHAIR depende do REVISOR pela meta concreta Proposta seja aprovada.

No capítulo 4, seção 4.2 Caso: Expert Committee, exemplificamos o processo de diagnóstico dessas três estruturas seguindo a orientação das heurísticas definidas na seção 2.2.4.2 a seguir.

#### **2.2.4.2 A Estratégia de Perguntas do Framework Diagnósticos i\***

O framework Diagnósticos i\* examina cada estrutura canônica (SDsituations e SRconstructs) de um dado modelo com a intenção de cobrir cada construto com perguntas que desafiam a consistência, a completeza, a eficiência e a eficácia de cada um deles.

A Figura 2.15 [Oliveira 08], que descreve o Framework i\* sob a forma de meta modelo, foi a fonte das perguntas que compõem nossa estratégia de diagnósticos. Diversas perguntas apresentadas no capítulo do método se basearam nesse modelo; por exemplo: (a) se o risco envolvido em uma relação de dependência estratégica pode ser minimizado, (b) devido ao relacionamento meios-fim deve ser questionado se não podem existir alternativas para o alcance do fim. O “meta” modelo UML da Figura 2.15 também destaca uma característica recursiva pelo fato de que uma submeta componente de uma tarefa-meio é a meta concreta fim de outra estrutura meios-fim.

O meta modelo pode também ser usado como um guia na elaboração de modelos i\*, além de ajudar na derivação dos controles de qualidade. Devido ao fato de o modelo i\* ser o componente-chave, a estratégia se baseia em questões sugeridas na análise de requisitos baseada na investigação (“inquiry based requirements analysis [Potts 94]), o que fornece tipos de perguntas para ajudar no processo. Elas são: what-is (o que), how-to (como), who (quem), what-kinds-of (que tipos), when (quando), perguntas de relacionamento e what-if (e se). De modo a completar com questões do 5w2h [Leite 05], adicionamos o why (por que) e where (onde) na estratégia.



## 2.3 Conclusão

O objetivo deste capítulo foi apresentar um resumo das técnicas, métodos e conceitos que são utilizados pelo método proposto, o qual tem a intencionalidade dos atores de uma organização como a origem dos requisitos do sistema de informação.

Na primeira seção o capítulo fez uma recuperação das abordagens já divulgadas pela literatura e mostrou como essas abordagens são úteis. “O Léxico Ampliado da Linguagem”, “A Técnica de Cenários” e “O Framework de Modelagem i\*” foram as estratégias apresentadas na primeira seção.

Na segunda seção foram detalhados os métodos, técnicas e conceitos elaborados pelo autor da tese durante a pesquisa para o Método ERi\*c. Primeiro foram definidos e exemplificados os conceitos de *Ação Concreta* e de *Ação Flexível*, ao salientar-se que o termo *ação flexível* estava sendo criado como uma complementação de *ação concreta*. Esses termos foram influenciados pelos conceitos de meta concreta e meta flexível anteriormente definidos pelo Framework i\* [Yu 95]. Em segundo lugar definimos e apresentamos exemplos dos instrumentos que definimos para o trabalho com diagramas i\* reduzidos. Por isso foram definidas as estruturas Canônicas do Framework i\*: a *SDsituation* e o *SRconstruct*. Em seguida, para lidar preliminar e exclusivamente com a intencionalidade, apresentamos o instrumento gráfico Painel da Intencionalidade ou *Diagrama IP*. Por último definimos a motivação e a idéia de fazer diagnósticos em modelos i\* a partir das duas estruturas canônicas que tínhamos definido, com o objetivo de elaborar Modelos SD e SR de melhor qualidade.

O capítulo 3 a seguir mostra as etapas do método ERi\*c com o devido encadeamento desses conceitos, técnicas e métodos.