

2

O Framework de Modelagem i*

Este capítulo descreve a abordagem de orientação a metas através do *framework* i*, base de toda a dissertação. Ao longo do capítulo, será apresentada a visão geral do *framework* i*, seus modelos SD e SR, além de novas técnicas, métodos e conceitos que estendem o supracitado *framework*: o modelo SA (*Strategic Actor*), situações de dependências estratégicas (*SDsituations*) e painéis de intencionalidade (diagramas IP).

2.1. Visão geral do framework i*

O framework de modelagem i* (i-estrela), originalmente proposto por Yu [1], trata da modelagem de contextos organizacionais tomando como base os relacionamentos de dependência entre os atores participantes.

O principal objetivo no i* é representar, através de modelos, os atores participantes e as dependências entre os mesmos, para que suas metas próprias sejam alcançadas, recursos sejam fornecidos, tarefas sejam realizadas e metas flexíveis sejam “satisfeitas a contento” ou “razoavelmente satisfeitas” [1], [2]. Mais além, o uso do referido framework possibilita o entendimento das razões internas dos atores (*rationale*), uma vez que as mesmas são expressas explicitamente, melhorando assim o entendimento do engenheiro de requisitos durante a modelagem do sistema [3].

O framework i* apresenta dois modelos diferentes: o modelo SD (modelo de dependências estratégicas – *Strategic Dependency*) e o modelo SR (modelo de razões estratégicas – *Strategic Rationale*). Esses modelos serão detalhados a seguir.

2.2. O Modelo SD – *Strategic Dependency*

O modelo SD exhibe os relacionamentos de dependência estratégica entre os atores da organização, utilizando para isso uma rede de nós, representando os atores (agentes, posições ou papéis) e arestas, representando as dependências entre os mesmos.

Neste modelo, uma dependência simboliza um relacionamento baseado em cooperação entre dois atores, onde o primeiro é denominado “*dependor*” ou “dependente” e o segundo “*dependee*” ou “de quem se depende”, unidos por um elo de dependência, denominado “*dependum*”, representando uma meta a ser atingida, uma tarefa a ser realizada, um recurso a ser fornecido ou uma meta flexível a ser satisfeita a contento. Yu [1] define os quatro tipos de *dependum* da seguinte forma:

Uma **meta** é uma condição ou estados de desejos no mundo que o ator deseja alcançar. Através das metas, podemos representar a intencionalidade dos atores organizacionais. Não existe a especificação de como a meta deve ser alcançada, possibilitando assim uma série de alternativas.

Uma **tarefa** especifica uma maneira particular de realizar algo. Quando uma tarefa é especificada como subtarefa de outra, ela restringe a tarefa superior para um curso de ação em particular. A decomposição de tarefas será explicada com maior profundidade na seção 2.3.

Um **recurso** é uma entidade (física ou informacional) que não é considerada problemática pelo ator. Sua principal característica é se está disponível (e por quem, no caso de uma dependência externa).

Uma **meta flexível** (*softgoal*) é uma condição ou estado no mundo que o ator deseja alcançar, porém diferentemente de uma meta (rígida ou concreta), o critério para a condição de ser alcançada não é definido a princípio, estando sujeito a interpretação. Para metas flexíveis, utiliza-se o termo “satisfeita a contento” ou “razoavelmente satisfeita”, pois servem para designar metas que não possuem critérios claramente definidos para sua satisfação. Essa subjetividade para a satisfação de metas flexíveis é inerente a requisitos não funcionais, por se tratar de aspectos de qualidade [3]. O termo “satisfeita a contento” é traduzido do termo em inglês *satisficy*, definido por Simon [4].

Metas, tarefas, recursos e metas flexíveis fazem parte dos elos de dependência entre atores, conforme mostra a Figura a seguir:

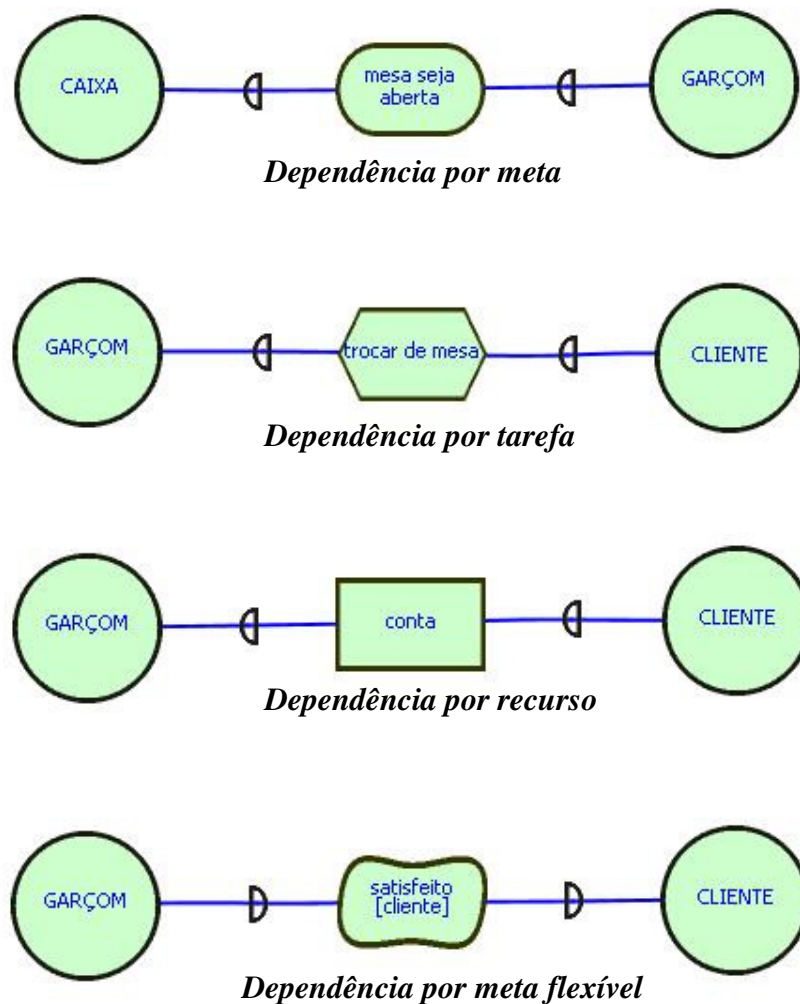


Figura 2.1 – Tipos de dependências entre atores no modelo SD.

Com base na Figura 2.1, pode-se perceber quatro tipos de dependências possíveis: dependência por meta, por tarefa, por recurso e dependência por meta flexível. Tais dependências serão descritas a seguir:

- **Dependência por meta:** o ator (*dependor*) depende de outro ator (*dependee*) para alcançar um determinado estado no mundo. Cabe ao *dependee* toda e qualquer decisão para que a meta seja alcançada. Neste caso, o *dependor* possui a habilidade de decidir que condição ou estado no mundo será alcançado. Porém, o *dependee* pode falhar na realização de tal condição, tornando assim o *dependor* vulnerável.

- **Dependência por tarefa:** o ator (*dependor*) depende de outro ator (*dependee*) para que uma tarefa seja desempenhada. Uma dependência por tarefa determina “como” e não “por que” uma tarefa deve ser desempenhada. Neste caso o *dependor* também fica vulnerável, pois o *dependee* pode falhar ao desempenhar a tarefa.
- **Dependência por recurso:** o ator (*dependor*) depende de outro ator (*dependee*) para a disponibilização de uma entidade (física ou computacional). Neste caso, o *dependor* possui a habilidade de usar a entidade como um recurso. Porém, o *dependor* torna-se vulnerável, pois a entidade pode ficar indisponível.
- **Dependência por meta flexível:** o ator (*dependor*) depende de outro ator (*dependee*) para que este desempenhe alguma tarefa para “satisfazer a contento” uma meta flexível. Neste caso, o *dependor* tem a decisão final em aceitar ou não a meta alcançada, usando o benefício de habilidades e conhecimentos do *dependee* [2]. Diferentemente do que ocorre em relação às metas, as condições para que as metas flexíveis sejam alcançadas são elaboradas de acordo com o desempenho das tarefas.

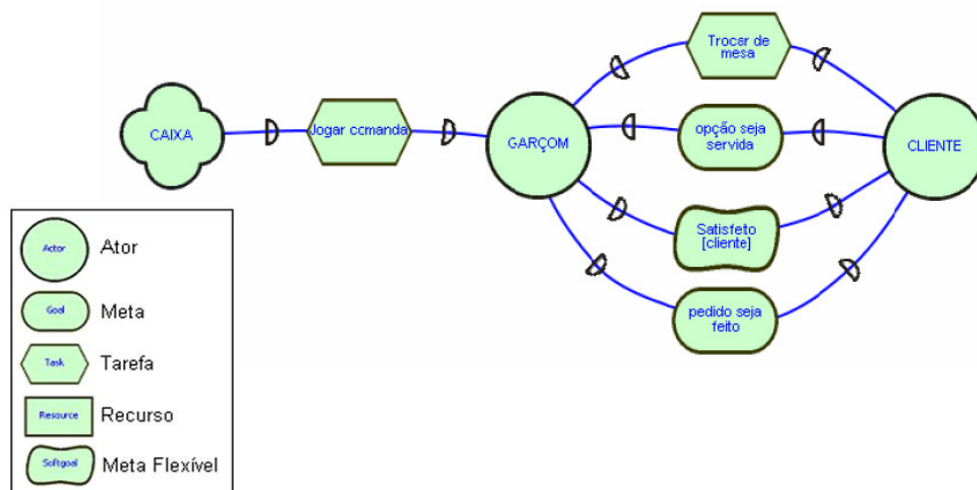


Figura 2.2 – Exemplo de um modelo SD. Retirado de [2].

De acordo com [1] e [2], os tipos dependências refletem o grau de liberdade que existe no relacionamento. Na dependência por meta, cabe ao *dependee* toda e

qualquer decisão para que a meta seja satisfeita. Na dependência por tarefa, o *dependee* executa a tarefa de acordo com o desejo do *dependor*, cabendo assim ao *dependor* tomar as decisões. No caso da dependência por recurso, o grau de liberdade é nulo, ou seja, o *dependee* fornece o recurso exatamente como o *dependor* necessita. Por fim, em uma dependência por meta flexível, o *dependor* toma a decisão final, sendo beneficiado pelo *know-how* (conhecimento prévio adequado) do *dependee*.

2.3.O Modelo SR – *Strategic Rationale*

O modelo SR tem como objetivo representar o detalhamento das estratégias internas (*rationale*) dos atores participantes, em termos dos elementos do processo, das alternativas e das decisões por trás do processo.

Estes modelos são elaborados de modo a expressarem o processo pelo qual as metas são alcançadas, as tarefas são elaboradas, os recursos são disponibilizados e as metas flexíveis são refinadas e operacionalizadas [3].

Os elementos do processo usados para representar os relacionamentos intencionais internos aos atores são: os *relacionamentos “meios-fim”*, que possuem o papel de explicitar as decisões que foram tomadas para que as metas do ator fossem alcançadas, a *decomposição de tarefas*, que detalha a elaboração e realização das tarefas, além de mostrar como os recursos são disponibilizados e utilizados, e os *relacionamentos de contribuição*, que exibem o tipo de contribuição (positiva ou negativa) entre metas flexíveis. A seguir, os elementos supracitados serão detalhados.

Decomposição de tarefa – uma tarefa (definida anteriormente como uma maneira particular de realizar algo) é mapeada graficamente pela representação de nós subcomponentes ligados a tarefa superior através de um segmento de reta cortado. Os nós subcomponentes podem ser: metas, tarefas, recursos e metas flexíveis. A Figura 2.3 representa um exemplo de decomposição de tarefa.

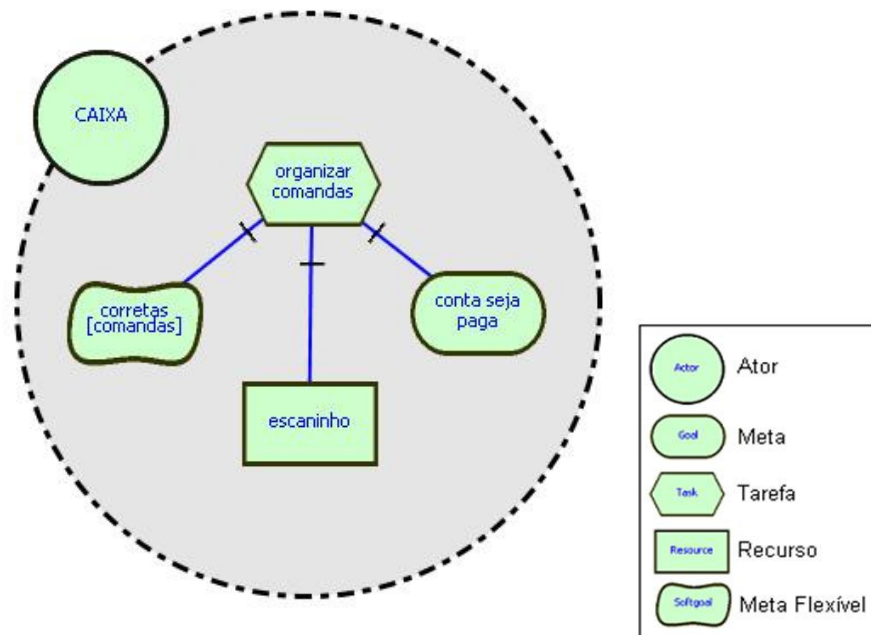


Figura 2.3 – Decomposição de tarefas (parte). Baseado em [2].

O exemplo de decomposição de tarefa presente na Figura 2.3 apresenta uma tarefa (organizar comandas) decomposta em uma meta (conta seja paga), um recurso (escaninho) e uma meta flexível.

Relacionamentos meios–fim – os relacionamentos do tipo meio fim são mapeados graficamente através de uma seta direcionada para o nó fim, que pode ser uma meta a ser alcançada, uma tarefa a ser desempenhada, um recurso a ser produzido ou uma meta flexível a ser “satisfeita a contento” [3]. Os meios são geralmente tarefas. Os tipos de relacionamentos meios–fim são descritos a seguir.

- Elo Meta–Tarefa – neste caso, o “fim” é especificado como uma meta e o “meio” como uma tarefa. A tarefa especifica “como” através da decomposição em seus componentes.
- Elo Recurso–Tarefa – neste caso, o “fim” é especificado como recurso e o “meio” como uma tarefa.

- Elo Meta Flexível–Tarefa – neste caso, o “fim” é especificado como uma meta flexível, e o ‘meio” é especificado como uma tarefa.
- Elo Meta Flexível–Meta Flexível – nos elos deste tipo, tanto o “fim” quanto o “meio” são metas flexíveis, permitindo o desenvolvimento de uma hierarquia meios–fim de metas flexíveis. Isso permite que as metas flexíveis sejam cada vez mais refinadas, até chegar a metas flexíveis mais fáceis de operacionalizar [3].

Os elos Meta Flexível–Meta Flexível possuem um relacionamento de contribuição, exibindo o tipo de contribuição entre as metas flexíveis. Em [1], Yu apresenta três valores para a contribuição: positiva (“+”), negativa (“-”) e neutra (“?”). Já em Chung et al. [5], são exibidos cinco valores para a escala de contribuição: *make* (“++”), *help* (“+”), *unknown* (“?”), *hurt* (“-”) e *break* (“--”).

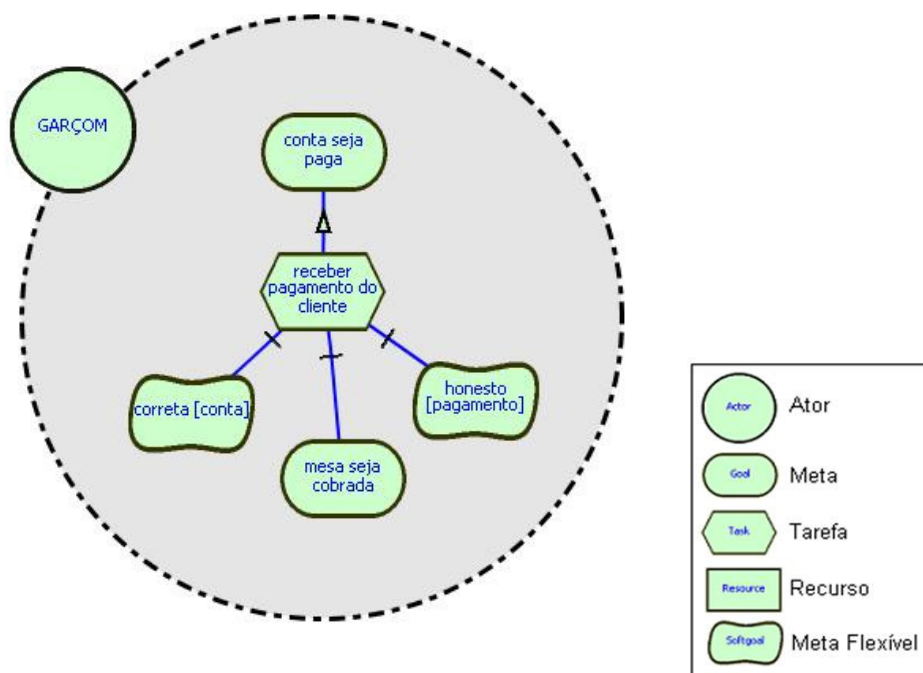


Figura 2.4 – Elos meios–fim (parte). Baseado em [2].

Na Figura 2.4 existe um exemplo de elo do tipo Meta–Tarefa, onde a meta “fim” é “conta seja paga” e a tarefa “meio” é “receber pagamento do cliente”. Cabe ressaltar que esta última tarefa se decompõe em uma submeta “mesa seja

cobrada” e duas metas flexíveis “correta [conta]” e “honesto [pagamento]”. A Figura a seguir, retirada de [2], mostra um exemplo de um modelo SR.

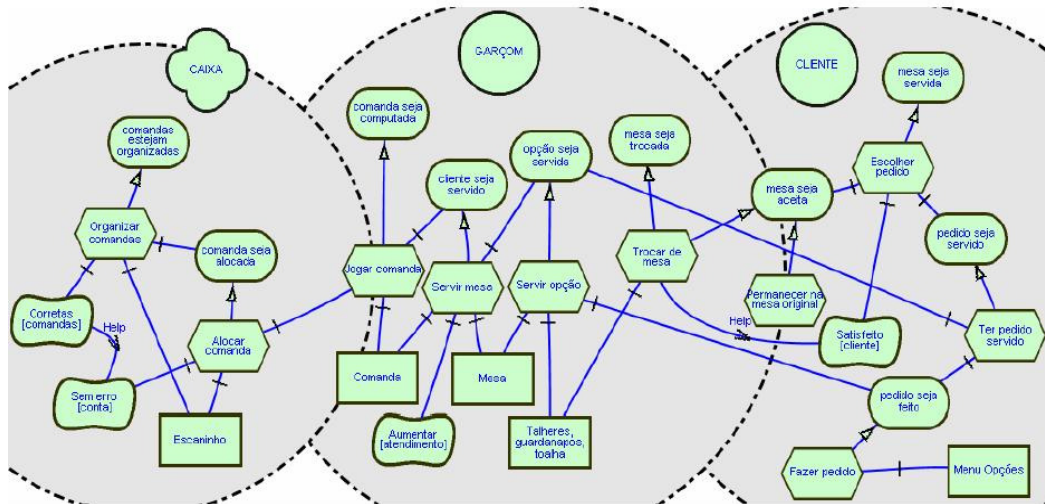


Figura 2.5 – Exemplo de Modelo SR. Retirado de [2].

2.4. Extensões do *framework i**

Este trabalho fará uso de algumas extensões sobre o *framework i** original: o modelo SA (*Strategic Actor*), desenvolvido em [6]; situações de dependências estratégicas (*SDsituations*), apresentado por Oliveira et al. em [7]; e painéis de intencionalidade (*Diagramas IP*), presente em [2]. Essas extensões serão detalhadas a seguir.

2.4.1. O Modelo SA – *Strategic Actor*

O modelo SA [6] tem como objetivo modelar os atores em *i**. A adoção de um modelo SA auxilia no entendimento dos atores e seus relacionamentos, do ponto de vista estrutural [3].

O modelo SA também usa os conceitos de agente, posição e papel, presentes em [1], para refinamento de atores e os relacionamentos entre si. Cabe ressaltar

que Leite et al. [1] apresentam um estudo detalhado sobre a especialização e representação do modelo SA. A seguir serão descritos alguns importantes conceitos:

Ator: segundo [1], um ator é uma entidade ativa que executa atividades para atingir suas metas pelo exercício de seu conhecimento próprio (*know-how*). Usa-se o termo ator para fazer referência a qualquer unidade à qual se possa atribuir dependências intencionais [1], [3].

Agente: um agente é um ator com manifestações físicas concretas, tal qual um ser humano [1]. Pode-se usar o termo agente tanto para humanos quanto para agentes artificiais (hardware/software). Um agente possui características normalmente intransferíveis para outros indivíduos, como habilidades, limitações e experiências [1].

Posição: uma posição é formada por um conjunto de papéis desempenhados por um agente [2]. Encontra-se em um nível intermediário de abstração entre um papel e um agente.

Papel: segundo [1], um papel é uma caracterização abstrata do comportamento de um ator social em algum contexto ou domínio especializado. Suas características podem ser transferidas para outros atores sociais.

Existe ainda a figura do agente real. O agente real é mais específico que o agente normal, possuindo identificação única. Em [6], Leite et al. descrevem os relacionamentos entre os conceitos supracitados, que serão apresentados resumidamente a seguir:

É um (IS A) – relacionamento de ator para ator, papel para papel, posição para posição e/ou agente para agente. É uma relação de especialização.

Instância (INS) – relacionamento de agente real para agente. Um agente pode ser instância de mais de um agente real.

Ocupa (OCCUPIES) – relacionamento de agente para posição. Um agente pode ocupar mais de uma posição e uma posição pode ser ocupada por mais de um agente [3].

Cobre (COVERS) – relacionamento de posição para papel. Uma posição cobre mais de um papel e um papel pode ser coberto por mais de uma posição.

Desempenha (PLAYS) – relacionamento de agente para papel. Um agente pode desempenhar mais de um papel e um papel pode ser desempenhado por mais de um agente.

Parte de (is-Part-of) – relacionamento de agente para agente, posição para posição e de papel para papel. Agentes, posições e papéis podem ser ou podem ter mais de uma subparte.

A Figura seguinte mostra um exemplo de um modelo de atores estratégicos e seus relacionamentos do jogo para ensino de engenharia de software, o SimuLES, que será abordado no estudo de caso da seção 4.

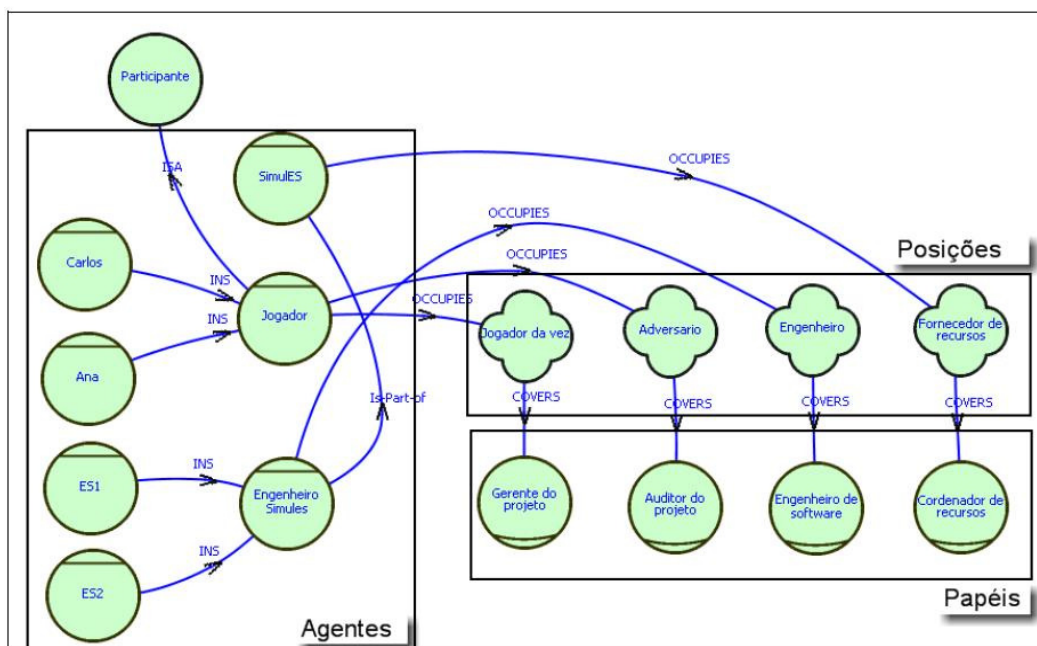


Figura 2.6 – Exemplo de um modelo SA.

2.4.2. Situação de Dependência Estratégica – *SDsituations*

SDsituation é uma representação estruturada de uma situação de dependência estratégica entre atores dentro do contexto organizacional. Segundo [2], uma *SDsituation* define um bloco de dependência com intencionalidade situacional compartilhada. As situações de dependência estratégica foram propostas por Oliveira et al. em [7].

O ponto central das *SDsituations* é que cada elemento de dependência (meta, tarefa, recurso ou meta flexível) envolvendo os atores não está isolado, ou seja, cada dependência é parte de uma situação bem definida de colaboração chamada de “situação de dependência estratégica” ou *SDsituation* [7].

Uma *SDsituation* pode ser identificada separadamente das outras *SDsituations*, formando uma cadeia de interdependências entre elas. Por exemplo, considerando o contexto de um congresso, um artigo só poderá ser revisado se ele tiver sido submetido anteriormente. Neste caso, são identificadas duas *SDsituations* separadamente: *Submissão de Artigos* e *Revisão de Artigos*. Apesar destas *SDsituations* terem sido identificadas de maneira separada, percebe-se uma interdependência crítica entre as duas.

Existem três tipos de interdependências entre *SDsituations*: física, lógica e temporal. Entretanto mais de um tipo de interdependência pode ocorrer simultaneamente. Os três tipos de interdependências serão descritos a seguir [7]:

Física – ocorre quando um recurso é preparado por uma *SDsituation* e é solicitado por outra *SDsituation*.

Lógica – ocorre quando uma ou mais *SDsituations* necessita da conclusão de outra *SDsituation* para sua iniciação, ou quando uma ou mais *SDsituations* dependem da conclusão de outra *SDsituation* para sua(s) conclusão(ões).

Temporal – ocorre 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*.

As relações de interdependências entre as *SDsituations* possuem fundamental importância para a análise organizacional. A Figura a seguir representa um exemplo adaptado de [3], sobre um sistema para suporte ao gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou *workshop*, o *Expert Committee*.

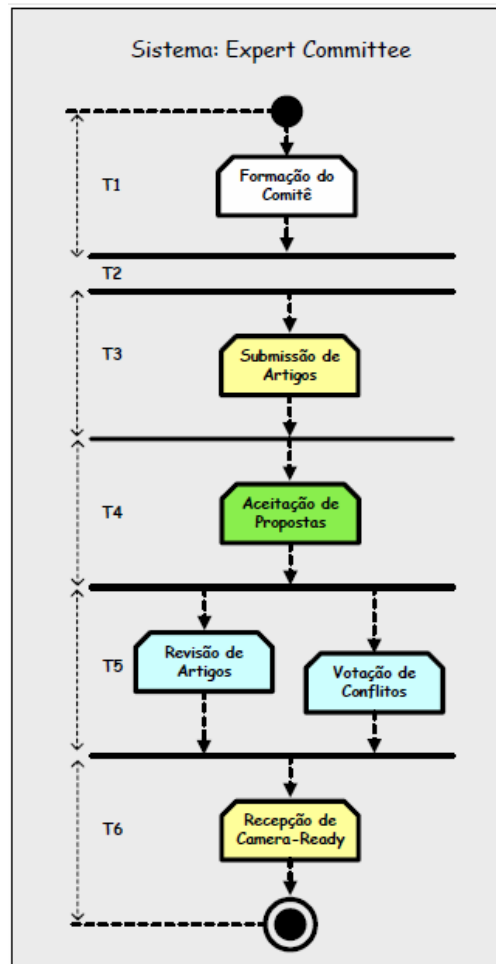


Figura 2.7 – Exemplo de interdependências entre *SDsituations*. Adaptado de [3].

Observado o exemplo supracitado, pode-se perceber que a *SDsituation* *Aceitação de Propostas* deve acontecer somente após a situação *Submissão de artigos*, devido a uma ligação lógica entre elas. Podemos perceber também que a *Recepção de Camera-Ready* jamais poderá vir antes de *Revisão de Artigos* ou *Submissão de Artigos*, por exemplo.

Uma situação de dependência estratégica é caracterizada principalmente por seu conjunto de elementos de dependência (metas, tarefas, recursos e/ou metas flexíveis). A Figura a seguir, retirada de [2] apresenta todos os relacionamentos de uma *SDsituation*.

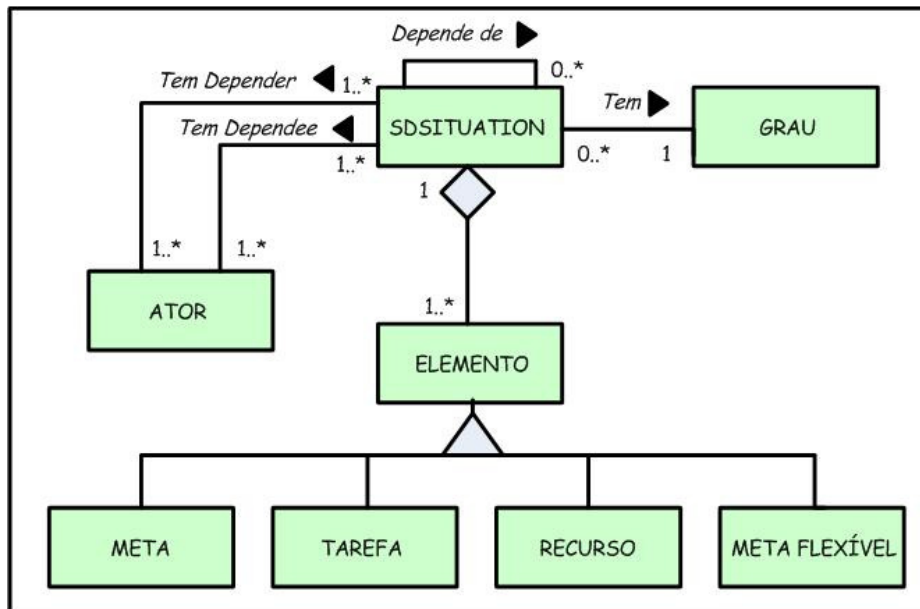


Figura 2.8 – Diagrama de classes de uma *SDsituation*. Traduzido de [2].

A Figura 2.8 mostra que atores (*dependers* e *dependees*) participam das *SDsituations* e que a mesma possui um grau de relevância (crítico, mediano ou indiferente). O diagrama também mostra que a situação de dependência estratégica pode ser formada por uma ou mais dependências, através de seus elementos (meta, tarefa, recurso e/ou meta flexível).

As *SDsituations* serão apresentadas com maior profundidade nos capítulos subsequentes.

2.4.3. Painel de Intencionalidade – Diagrama IP

O painel de intencionalidade tem como objetivo mapear a intencionalidade em partes, guiando-se pelas *SDsituations*, ou seja, somente representando as metas em um diagrama chamado “Painel de Intencionalidade” ou diagrama IP.

De acordo com [2], um diagrama IP é uma redução do modelo SR, onde são considerados somente os atores, as metas (concretas e flexíveis) e as relações entre elas. Toda a intencionalidade é representada em um único diagrama, visando principalmente, diminuir a dificuldade de entendimento do diagrama SR, que é complexo por natureza.

Um diagrama IP é composto por nós e arestas. Os nós são metas concretas ou metas flexíveis, enquanto as arestas são os tipos de relações entre metas (quatro relações no total). Os atores estão presentes no diagrama para indicar os responsáveis pelas metas. A seguir, serão descritos os tipos de relações [2]:

Correlação – a correlação ocorre entre duas metas de um mesmo ator. Indica que uma das metas é subcomponente de uma tarefa, que é meio para alcançar a meta principal.

Contribuição – relação que ocorre entre duas metas flexíveis de um mesmo ator. Uma meta flexível pode contribuir de forma positiva (“+”), negativa (“-”) ou neutra (“?”) para uma outra meta flexível.

Dependência – esta relação ocorre entre duas metas de atores diferentes. Ela representa a necessidade de satisfazer uma dependência entre dois atores, exatamente como em um diagrama SD. A relação de dependência indica que o sucesso da meta principal (do *dependor*) depende do alcance da meta inicial (do *dependee*).

Equivalência – relação entre metas concretas ou metas flexíveis. Indica apenas que, quando este relacionamento existe, as metas concretas ou flexíveis são equivalentes para atores diferentes.

A Figura a seguir, apresenta um exemplo de um diagrama IP, baseado em um sistema de controle caixa de restaurante, retirado de [2].

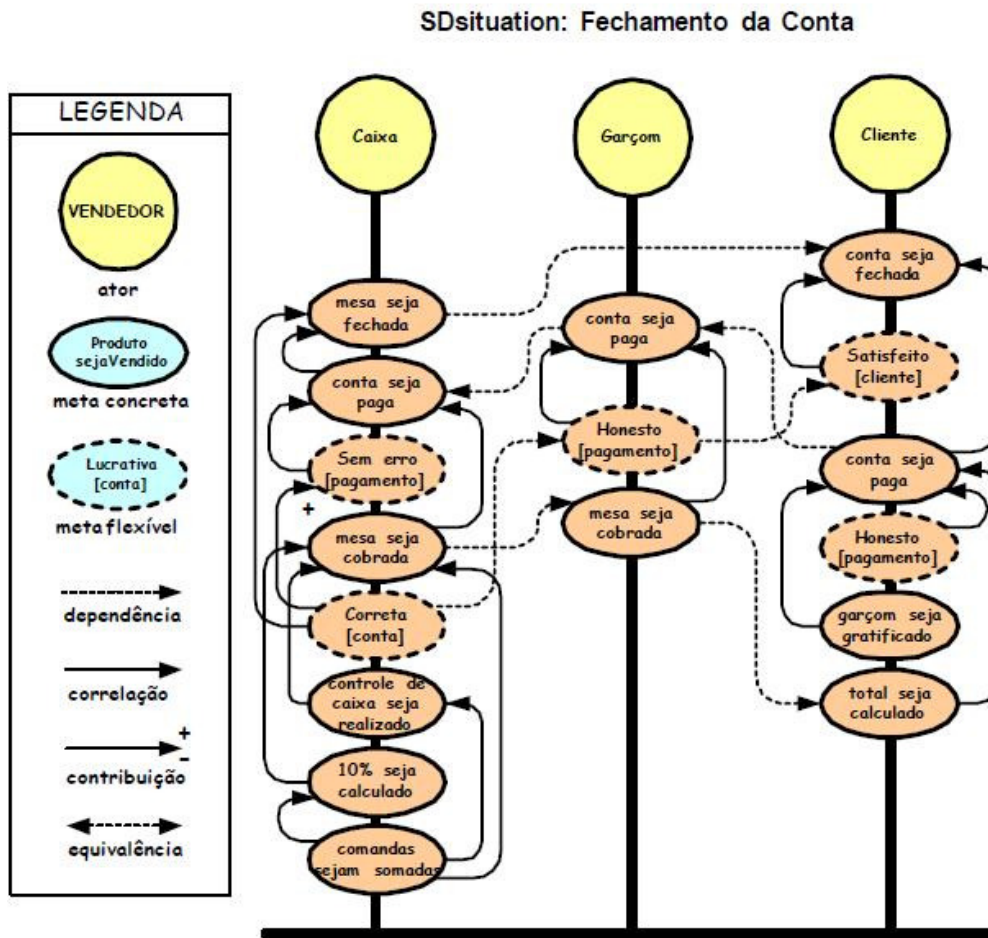


Figura 2.9 – Exemplo de um painel de intencionalidade [2].

No exemplo da Figura 2.9, pode-se observar os relacionamentos entre metas descritos anteriormente. Por exemplo, existe um relacionamento de dependência entre os atores caixa e cliente, através das metas *mesa seja fechada* e *conta seja fechada*. Além disso, a meta flexível *Correta [conta]* contribui positivamente para um pagamento sem erros (*Sem erro [pagamento]*).

Os painéis de intencionalidade, ou diagramas IP serão utilizados com maior profundidade nas seções 3 e 4 deste trabalho.