



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA

GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

**INTEGRAÇÃO BIDIRECIONAL ENTRE OS MODELOS I\* E BPMN  
NO CONTEXTO DA GESTÃO DE PROCESSOS DE NEGÓCIO**

REBECA DE SOUZA ALVES

TRABALHO DE GRADUAÇÃO

Recife  
ABRIL DE 2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA

REBECA DE SOUZA ALVES

**INTEGRAÇÃO BIDIRECIONAL ENTRE OS MODELOS I\* E BPMN NO CONTEXTO DA  
GESTÃO DE PROCESSOS DE NEGÓCIO**

*Trabalho apresentado ao Programa de GRADUAÇÃO EM  
ENGENHARIA DA COMPUTAÇÃO do CENTRO DE  
INFORMÁTICA da UNIVERSIDADE FEDERAL DE  
PERNAMBUCO como requisito parcial para obtenção do  
grau de Bacharel em ENGENHARIA DA COMPUTAÇÃO.*

Orientadora: *Profª Drª Carla Taciana Lima Lourenço Silva Schuenemann*

Recife  
ABRIL DE 2013

*A minha tia Jemima,  
exemplo vivo de altruísmo.*

# Agradecimentos

Sou imensamente grata

A Deus, porque sempre está comigo

Aos meus pais, por toda dedicação

Ao meu noivo, pelo apoio e compreensão

As minhas irmãs, pelo companheirismo

A minha orientadora, pela sua determinação em me passar seu conhecimento

E a minha família, por todo amor que foi fundamental para essa realização.

# Resumo

Existem várias abordagens para descoberta e especificação dos requisitos de um sistema e uma delas é a abordagem orientada a objetivos, que prioriza as metas que os usuários esperam do sistema, além de modelar os objetivos organizacionais relacionados aos processos de negócios de uma organização. Uma abordagem orientada a objetivos usada para este fim é o *i\**, que representa os relacionamentos sociais e as dependências entre os *stakeholders* e o sistema. Os processos de negócio dentro de uma organização precisam ser modelados para facilitar a comunicação e entendimento entre participantes do processo e pela própria organização. Este trabalho argumenta em favor do uso complementar da modelagem orientada a objetivos com o Framework *i\** e a modelagem baseada em fluxogramas como o BPMN, de forma a obter modelos BPMN diretamente de modelos *i\**. O objetivo desse trabalho é definir heurísticas de mapeamento para obter sistematicamente um modelo de processo de negócio com a notação BPMN, a partir de informações extraídas de modelos orientados a objetivos descritos com a linguagem *i\**. Desta forma, o modelo BPMN obtido estará alinhado com aos objetivos da organização.

**Palavras-chave:** Framework *i\**, modelos BPMN, BPM, modelagem orientada a objetivos, modelos de processos de negócios, gestão de processos de negócio.

# Sumário

<b>1</b>	<b>Introdução</b> .....	<b>9</b>
	1.1 Considerações Iniciais .....	9
	1.2 Motivação .....	10
	1.3 Objetivos do trabalho .....	11
	1.4 Estrutura do documento.....	11
<b>2</b>	<b>Fundamentação Teórica</b> .....	<b>13</b>
	2.1 Engenharia de Requisitos .....	13
	2.1.1 Classificação dos Requisitos .....	14
	2.1.2 Processos da Engenharia de Requisitos.....	15
	2.2 Engenharia de Requisitos Orientada a Objetivos .....	15
	2.2.1 Framework i* .....	17
	2.3 Gestão de Processos de Negócio .....	23
	2.3.1 BPMN.....	24
	2.4 Trabalhos Relacionados .....	27
	2.4.1 Abordagem da Metodologia KAOS com BPMN.....	27
	2.4.2 Abordagem do Framework i* com BPMN .....	33
	2.4.3 Desafios Práticos no Mapeamento i* - BPMN .....	36
<b>3</b>	<b>Aplicação do Método Atual de Mapeamento</b> .....	<b>40</b>
	3.1 Descrição do Processo.....	40
	3.2 Aplicação do Método .....	41
	3.2.1 Mapeamento do i* para o BPMN.....	41
	3.2.2 Mapeamento do BPMN para o i* .....	45
	3.2.3 Limitações do Método Aplicado .....	46
<b>4</b>	<b>Proposta de um Novo Método de Mapeamento</b> .....	<b>48</b>
	4.1 Heurísticas de Mapeamento .....	48
	4.1.1 i* para BPMN.....	52
	4.1.2 BPMN para i*.....	53
	4.2 Avaliação do método proposto.....	54
	4.2.1 Propriedades Metodológicas .....	55
	4.2.2 Propriedades Organizacionais .....	55
	4.2.3 Resultado da avaliação .....	56
<b>5</b>	<b>Conclusão</b> .....	<b>58</b>
	5.1 Contribuições e Limitações .....	58
	5.2 Trabalhos Futuros.....	59
	5.3 Considerações Finais .....	59
	<b>Referências Bibliográficas</b> .....	<b>60</b>

## Lista de Figuras

Figura 2.1 - Elementos do $i^*$ . .....	19
Figura 2.2 - Modelo de Dependência Estratégica (SD) (Yu, 1995). .....	20
Figura 2.3 - Modelo de Razão Estratégica (Yu, 1995). .....	22
Figura 2.4 - Objetos de Fluxo (WHITE, 2004, adaptado pelo autor). .....	25
Figura 2.5 - Objetos de Conexão (WHITE, 2004, adaptado pelo autor). .....	26
Figura 2.6 – Swimlanes (WHITE, 2004, adaptado pelo autor). .....	27
Figura 2.7 - Contexto atual do modelo BPMN (Koliadis e Ghose, 2006). .....	29
Figura 2.8 - Processo ‘Sorting Package’ (Koliadis e Ghose, 2006). .....	30
Figura 2.9 - Rastreabilidade de objetivos (Koliadis e Ghose, 2006). .....	31
Figura 2.10 – Efeitos, Trajetória e Satisfação dos Objetivos (Koliadis e Ghose, 2006). .....	32
Figura 2.11 – Adição de Objetivos (Koliadis e Ghose, 2006). .....	33
Figura 2.12 - Processo Modificado (Koliadis e Ghose, 2006). .....	33
Figura 3.1 - Modelagem $i^*$ do processo de indenização de seguros (YU, 1995). .....	41
Figura 3.2 - Rotina (YU, 1995, adaptado pelo autor). .....	42
Figura 3.3 - Escopo do processo (YU, 1995, adaptado pelo autor). .....	43
Figura 3.4 - Modelo BPMN gerado do modelo $i^*$ . .....	44
Figura 3.5 - Modelo $i^*$ gerado do BPMN. .....	45
Figura 4.1 - Modelo BPMN gerado do modelo $i^*$ da Fig. 3.1. .....	52
Figura 4.2 - Modelo $i^*$ obtido do modelo BPMN da Fig. 4.1. .....	54

## **Lista de Tabelas**

Tabela 3.1 - Anotação de satisfação para cada tarefa/dependência.....	43
Tabela 4.1 - Comparação dos métodos.....	57

# Capítulo 1

## Introdução

O presente capítulo está dividido em quatro partes: (i) introdução de uma visão geral sobre a modelagem orientada a objetivos e a modelagem de processos de negócio; (ii) apresentação da importância do trabalho que forneceu a motivação necessária para a construção do documento; (iii) definição dos objetivos do estudo; e, (iv) detalhamento da estrutura do documento.

### 1.1 Considerações Iniciais

A gestão de processos de negócios (BPM – *Business Process Management*) e a modelagem de processos de negócios são essenciais para manter a compreensão e a comunicação entre esses processos em uma organização, como também, ajudam na análise e a melhoria dos processos de negócio.

O BPM tem sido cada vez mais utilizado pela indústria. Na verdade, as técnicas de modelagem de processos de negócios focam em capturar o fluxo de atividades do processo, como por exemplo, o BPMN (*Business Process Modeling Notation*) (Miers e Stephen, 2008), que é baseado em conceitos, como: atividades, eventos, artefatos de controle de fluxo, mensagens de fluxo, desvios, entre outros. No entanto, essas técnicas não têm como capturar os objetivos estratégicos da organização e, portanto, não tornam explícito o alinhamento (ou a falta de alinhamento) entre os processos e o cumprimento desses objetivos.

Além disso, os processos de negócios acontecem em ambientes sociais da organização. As organizações são compostas de atores sociais que possuem objetivos e interesses que são atendidos através de uma rede de relações com outros atores. Um modelo mais rico de processos de negócios deve incluir, não somente como, quando e por quem as atividades dos processos são realizadas, mas também os objetivos dos atores responsáveis pela execução dessas atividades, além das dependências entre esses atores, a fim de atingir seus objetivos (YU e MYLOPOULOS, 1994).

A modelagem de processos de negócios através de uma abordagem orientada a objetivos, como o Framework i\*, ajuda no alinhamento da melhoria dos processos de negócios a fim de satisfazer os objetivos estratégicos da organização. Modelos i\* capturam o

nível organizacional com ênfase na motivação e intencionalidade dos atores no ambiente organizacional. O  $i^*$  traz a análise social que complementa a visão do fluxo de atividades e se concentra nos objetivos que a organização quer atingir (Yu, 1995). Desta forma, os modelos orientados a objetivos poderiam complementar a modelagem de processos de negócio tradicional, usando abordagens baseadas no fluxograma, como o diagrama de atividades de UML (Booch et al., 1999) e o BPMN.

Esse trabalho argumenta em favor do uso conjunto destas abordagens, para modelar os processos de negócio. Em particular, será apresentada uma abordagem para obter modelos BPMN de modelos  $i^*$  e vice-versa, a fim de assegurar que os processos de negócios estejam alinhados com os objetivos estratégicos organizacionais. Esta abordagem é uma extensão da proposta apresentada por Koliadis et al. (2006), já que refina as heurísticas existentes no mapeamento de modelos  $i^*$  para modelos BPMN e vice-versa e adiciona novas heurísticas para este mapeamento. A integração bidirecional entre o  $i^*$  e o BPMN tem como objetivo incentivar o uso combinado de ambas as técnicas de modelagem no contexto do BPM.

## **1.2 Motivação**

Os processos de negócios são os que caracterizam as ações da organização e representam o que a organização faz, a fim de alcançar um propósito ou objetivo específico. Estes processos são suportados por outros processos internos, resultando em um produto ou um serviço que é recebido por um cliente externo. Por isso, a gestão adequada dos processos de negócio é importante porque uma organização deve ser tão eficaz e eficiente, quanto seus processos de negócios.

Com base nos fundamentos apresentados anteriormente, um dos problemas que existem no mapeamento de modelos  $i^*$  em BPMN e vice-versa, é a falta de um método sistemático que guie o processo de mapeamento. Este método iria garantir a consistência entre os modelos mapeados, e assim, asseguraria que os processos de negócio estão alinhados aos objetivos da organização.

Neste contexto, esse trabalho consiste na elaboração de uma abordagem sistemática para obter modelos BPMN a partir de modelos  $i^*$  e vice-versa, diminuindo as etapas do processo em que o analista precisa usar o seu julgamento e experiência pessoal.

### **1.3 Objetivos do trabalho**

O objetivo geral deste trabalho é melhorar a abordagem de Koliadis et al. (2006) para mapear sistematicamente um modelo de processo de negócio descrito com a notação  $i^*$  para um modelo BPMN, e vice-versa.

Como objetivos específicos temos:

- Analisar trabalhos existentes que foquem na obtenção sistemática de modelos BPMN a partir de modelos orientados a objetivos;
- Aplicar a abordagem de Koliadis et al. (2006) a um exemplo de modelo  $i^*$  de forma a levantar as limitações desta abordagem de mapeamento;
- Elaborar novas heurísticas de mapeamento entre modelos  $i^*$  e BPMN;
- Aplicar as novas heurísticas a um exemplo para destacar as contribuições deste trabalho.

### **1.4 Estrutura do documento**

Este documento está organizado da seguinte forma:

Capítulo 2 – Fundamentação Teórica. Este capítulo apresenta uma introdução à Engenharia de Requisitos, com destaque para a Engenharia de Requisitos Orientada a Objetivos e o Framework  $i^*$ . Apresenta também, uma introdução à Gestão de Processos de Negócio, com destaque para a Modelagem de Processos de Negócio e a notação BPMN. Por fim, apresenta alguns trabalhos relacionados com o mapeamento entre modelos orientados a objetivos e o BPMN.

Capítulo 3 – Aplicação do Método Atual de Mapeamento. Um exemplo é usado para a demonstração do método de mapeamento entre  $i^*$  e BPMN proposto por Koliadis et al. (2006). Em seguida, apresenta-se as limitações da abordagem identificadas após a aplicação do método ao exemplo.

Capítulo 4 – Proposta de um Novo Método de Mapeamento. Nesse capítulo propõe-se melhorias para o método de Koliadis et al. (2006) e, em seguida, o novo método é aplicado a um exemplo. No final do capítulo, realiza-se uma análise da melhoria proposta e uma comparação entre o método antigo e o novo.

Capítulo 5 – Conclusão. As considerações finais, contribuições, limitações e possíveis trabalhos futuros são apresentados nesse capítulo.

## Capítulo 2

# Fundamentação Teórica

Nesse capítulo são apresentados os conceitos da Engenharia de Requisitos, e da Engenharia de Requisitos Orientada a Objetivos, enfatizando o Framework i\*. Além disso, são apresentados os conceitos da Gestão de Processos de Negócio e da Modelagem de Processos de Negócio com a Notação BPMN. O capítulo é finalizado com a exposição dos trabalhos relacionados, que contribuíram para elaboração desse trabalho.

### 2.1 Engenharia de Requisitos

Os sistemas de informação são fundamentais para o progresso das atividades da organização, tanto em âmbito estratégico como também funcional. Os desenvolvimentos dos softwares precisam está de acordo com os negócios da empresa, com isso, deve-se estabelecer uma ligação direta entre os requisitos que são necessários para o desenvolvimento correto do software e o processo da empresa.

Os sistemas possuem como função principal: apoiar o processo de negócio da empresa, por isso que os requisitos da organização precisam ser identificados adequadamente e entendidos para que os sistemas que irão apoiar esse processo reflitam de maneira correta os negócios da organização.

Segundo Kotonya e Sommerville (1998), a obtenção dos requisitos é um processo que reúne informações sobre os usuários do sistema e informações sobre o sistema proposto. O processo da engenharia de requisitos inclui uma avaliação sobre a utilidade do sistema para a empresa, a obtenção dos requisitos, a transformação dos requisitos em algum padrão e a validação dos requisitos de acordo com o desejo do cliente.

De acordo com Paula Filho (2001), a engenharia de requisitos consiste nas técnicas de levantamento, documentação e análise dos requisitos. É uma parte da engenharia de software que aborda a fase inicial do desenvolvimento do sistema, e tem como atividades: a elicitação, a análise, a negociação, a especificação, a validação, a documentação e o gerenciamento dos requisitos, que contribuem para que os requisitos sejam completos, consistentes e não ambíguos.

A fase inicial do desenvolvimento do sistema consiste em um dos problemas mais básicos da engenharia de software, porque os requisitos descrevem os serviços do sistema e suas restrições, e como eles refletem as necessidades dos usuários. A elaboração dos requisitos precisa ser entendida por todos os *stakeholders*, para que não exista inconsistência de informações no desenvolvimento do software (KOTONYA e SOMMERVILLE, 1998).

Requisitos indicam o que um sistema deverá fazer, por isso que a compreensão do contexto organizacional é fundamental, pois contribui para o sucesso contínuo do sistema. É nesse contexto que o uso das técnicas de modelagem de requisitos auxilia na análise dos interesses dos *stakeholders* e também na modelagem dos requisitos obtidos nessa análise (YU, 1997).

### **2.1.1 Classificação dos Requisitos**

Segundo Kotonya e Sommerville (1998), existem dois níveis de descrição dos requisitos: requisitos de usuários e requisitos de sistema.

- Requisitos de usuário – são declarações em linguagem natural, sobre os serviços e restrições do sistema. Os requisitos de usuário constituem os requisitos abstratos de alto nível.
- Requisitos de sistema – constituem uma descrição detalhada do que o sistema deve fazer, como as funções, os serviços e as restrições operacionais do sistema.

Além disso, os requisitos também são classificados em relação ao tipo de requisito: requisitos funcionais, não funcionais e de domínio (KOTONYA e SOMMERVILLE, 1998).

- Requisitos funcionais – descrevem os serviços oferecidos pelo sistema, o comportamento que o sistema deve ter quando receber uma determinada entrada e também especifica o que o sistema não deve fazer.
- Requisitos não funcionais – são descrições sobre a restrição de funções e serviços do sistema, como restrição de *timing*, desenvolvimento e padrões de projeto.
- Requisitos de domínio – se referem ao domínio da aplicação do sistema e caracterizam as restrições e características do domínio. Os requisitos de domínio se referem a requisitos funcionais e não funcionais.

### **2.1.2 Processos da Engenharia de Requisitos**

O processo da engenharia de requisitos é formado por atividades, divididas em fases, que criam e mantêm um documento de requisitos do sistema. A divisão é feita em quatro sub-processos que estão explicados a seguir (KOTONYA e SOMMERVILLE, 1998).

- Estudo de viabilidade – é um estudo breve, que envolve coleta e avaliação de informações e a elaboração de um relatório com foco na possibilidade de desenvolvimento do sistema.
- Elicitação e análise de requisitos – é a atividade em que os engenheiros de software interagem com os usuários do sistema e outros *stakeholders* para entender o domínio da aplicação e os serviços que devem ser fornecidos no sistema.
- Validação de requisitos – permite confirmar se os requisitos realmente estão de acordo como desejo do cliente. Nessa etapa, o foco é encontrar problemas nos requisitos para que erros sejam detectados antes do desenvolvimento completo do sistema, diminuindo assim, os custos com retrabalho.
- Gerenciamento de requisitos – é uma etapa do processo para compreender e controlar as mudanças de requisitos do sistema. Um planejamento é feito para diminuir o impacto das mudanças de requisitos.

### **2.2 Engenharia de Requisitos Orientada a Objetivos**

Um dos problemas da Engenharia de Software é o levantamento dos requisitos, porque os requisitos são geralmente implícitos e difíceis de serem modelados. Por isso, que o estudo da Engenharia de Requisitos Orientada a Objetivos é importante, pois, ela destaca a razão do sistema a ser modelado (YU, 1997).

Deste modo, objetivos são as metas que os usuários têm em relação ao sistema e são os objetivos que guiam o desenvolvimento do sistema. Os requisitos por serem proeminentes no desenvolvimento dos sistemas têm como medidas relevantes para o sucesso do sistema, o grau de satisfação no qual eles atendem aos objetivos que foram levantados para a construção do projeto (LAMSWEEERDE, 2001).

Os requisitos geralmente são entendidos como uma afirmação do que o sistema deve fazer, e não de como deve realizar. Porém, a compreensão do contexto organizacional fornece o entendimento correto dos requisitos do sistema, e isso contribui para o sucesso contínuo do sistema (Yu, 1997). As técnicas de modelagem auxiliam na obtenção do conhecimento e do raciocínio necessário nessa fase inicial do reconhecimento dos requisitos, que consiste em modelar e analisar os interesses dos *stakeholders*.

As metodologias orientadas a objetivos se baseiam na análise e definição das metas. A análise é feita em nível organizacional, operacional e técnico, apontando os problemas e as oportunidades do sistema a ser desenvolvido. Com isso, os objetivos são identificados e refinados para resolver tais problemas e encontrar os requisitos para atingir essas metas (LAMSWEERDE, 2001).

A Engenharia de Requisitos Orientada a Objetivos se baseia na aquisição dos requisitos, considerando as metas que os usuários desejam alcançar. A modelagem dos requisitos respeita os objetivos dos *stakeholders* dentro do ambiente organizacional, porque os objetivos são importantes em vários aspectos. Eles conduzem à incorporação dos requisitos, justificam e explicam a presença dos requisitos, que não são necessariamente compreensíveis pelos usuários. Além disso, ajudam na atribuição das responsabilidades dos atores no sistema, ou seja, dão a base para a melhor definição de quais atores devem executar as ações para que as restrições sejam atendidas (DARDENNE et al., 1993).

Como o objetivo é uma meta que o cliente deseja alcançar, as propriedades que são formuladas para atender aos objetivos se encontram em diversos níveis de abstração. Desde o alto nível, como requisitos estratégicos, até o baixo nível, como requisitos técnicos. Além de abranger os interesses funcionais, ou seja, os serviços que o sistema oferece, e os interesses não funcionais, que se referem à qualidade do software (LAMSWEERDE, 2001).

Ao contrário dos requisitos, um objetivo, requer, em geral, a colaboração de vários atores para alcançá-lo (Dardenne et al., 1993), por isso, um dos resultados mais importantes do processo da Engenharia de Requisitos é a decisão sobre quais as responsabilidades que cada ator deve ter no ambiente da organização.

Como a documentação dos requisitos na maioria das vezes não é bem entendida pelo cliente, cada vez mais a abordagem orientada a objetivos é usada na engenharia de requisitos, porque facilita o entendimento dos requisitos por meio dos objetivos (LAMSWEERDE,

2001). Ela aborda a modelagem organizacional e, por conseguinte, os requisitos funcionais, não funcionais e organizacionais são modelados. A abordagem orientada a objetivos facilita a modelagem dos requisitos, pois ajuda na elicitaco e validao dos requisitos, contribuindo para que os objetivos principais do sistema sejam satisfeitos.

Na subseo a seguir, uma abordagem orientada a objetivos, a modelagem *i\**,  apresentada.

### **2.2.1 Framework *i\****

O crescimento da complexidade dos sistemas e a juno cada vez maior com o ambiente social da organizao dificultam o entendimento correto dos *stakeholders*. Por isso,  necessria uma modelagem que reflita os relacionamentos sociais e as dependncias entre os *stakeholders* e o sistema (YU, 2009).

A modelagem *i\**  de fcil entendimento dos *stakeholders* porque, a modelagem  feita em nvel organizacional, ou seja, ela apresenta a relao dos *stakeholders* entre si e entre o sistema, com nfase na intencionalidade e na motivao no ambiente da organizao (YU, 1995).

O framework *i\**  fundamentado na engenharia de requisitos baseada em atores, ou seja, aborda os atores e suas dependncias sociais, que inclui “dependncia de metas, *softgoals*, tarefas e recursos associados com as intences e as necessidades dos atores no ambiente organizacional” (KOLIADIS et al., 2006).

Essa tcnica de modelagem organizacional ajuda a identificar os requisitos do sistema e tambm auxiliar em futuras mudanas que possam vir a ocorrer. O *i\** procura entender o motivo pelo qual o requisito  necessrio e o modo como ele  especificado.

De acordo com Yu (2009), o ator  o conceito central na abordagem *i\**, porque, ele possui intences, objetivos, realiza tarefas e dispe de recursos. Para alcanarem os objetivos ou realizarem algumas tarefas, os atores precisam de outros atores ou do sistema, com isso, estabelece uma relao de dependncia entre os atores ou entre o sistema.

Os sistemas computacionais dependem de muitos fatores alm das tcnicas de sistema. Eles tambm dependem do ambiente organizacional no qual esto envolvidos. Os *stakeholders* possuem uma grande gama de conceitos sobre os sistemas que podem mudar o

ambiente de trabalho. Isso quer dizer que, na introdução de um novo sistema de informação, pode existir mudanças nos papéis da organização, ou mudanças de tarefas que podem ser aceitas ou não, gerando conflito entre os *stakeholders* e o sistema (YU, 1995).

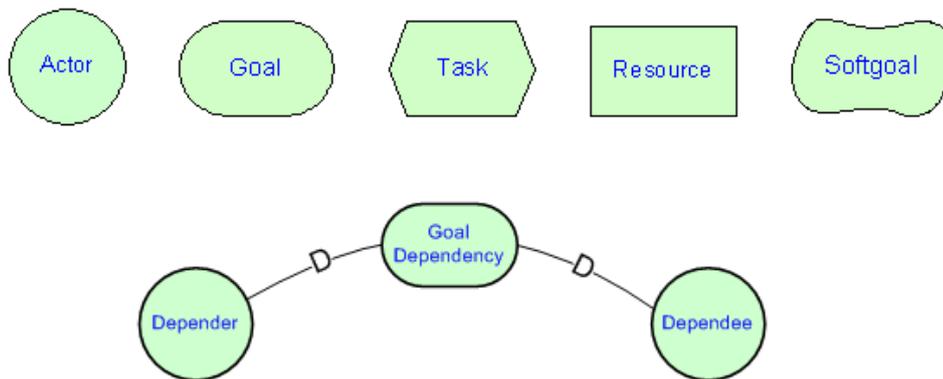
Por essa razão, a modelagem *i\** se preocupa com as descrições da organização social, diferente dos modelos convencionais que dão importância às técnicas para o desenvolvimento do sistema. Com a técnica do *i\**, os conceitos organizacionais são integrados ao sistema, facilitando seu desenvolvimento como também o entendimento dos *stakeholders* (DECREUS et al., 2009).

O *i\** traz a concepção social para dentro do processo da engenharia de sistemas, destacando os conceitos sociais que são representados pelas atividades diárias da organização. A análise social é a base para a boa condução do desenvolvimento do sistema, porque supera as habituais visões do fluxo de atividades e informações e passa a se preocupar com o que os atores querem, como eles querem, e de quem eles dependem para alcançar algo.

Yu (1995) descreveu os elementos do *i\** que são apresentados a seguir:

- Ator – É uma entidade ativa que realiza ações para atingir um objetivo. É representado por um círculo (Fig. 2.1).
- Objetivo (*Goal*) – O objetivo é uma condição ou estado que os *stakeholders* desejam alcançar. Sua representação é feita por uma forma oval (Fig. 2.1).
- *Softgoal* – O *softgoal* é uma condição ou estado que os *stakeholders* gostariam de viver, são os requisitos não funcionais do sistema. Os *softgoals* são representados por uma forma de “nuvens” (Fig. 2.1).
- Tarefa (*Task*) – É um modo específico de se fazer alguma coisa. São soluções operacionalizadas para satisfazer os objetivos e *softgoals*. As tarefas são representadas por hexágonos (Fig. 2.1).
- Recurso (*Resource*) – É uma entidade física ou não que tem como requisito não funcional a disponibilidade. Os recursos são representados por retângulos (Fig. 2.1).

- Dependência – Representa uma relação intencional entre dois atores, em que um dos atores depende (*dependor*) do outro ator (*dependee*) para alcançar algo (*dependum*) (Fig 2.1).



**Figura 2.1 - Elementos do i\*.**

O i\* possui dois modelos: um que descreve as relações de dependência entre os atores e o outro que representa as razões que os atores têm que justificam serem as configurações de um jeito e não de outro (Yu, 1995). São esses modelos respectivamente, o modelo de Dependência Estratégica, do inglês, *Strategic Dependency Model* (SD) e o modelo de Razão Estratégica, do inglês, *Strategic Rationale Model* (SR).

### **Modelo de Dependência Estratégica**

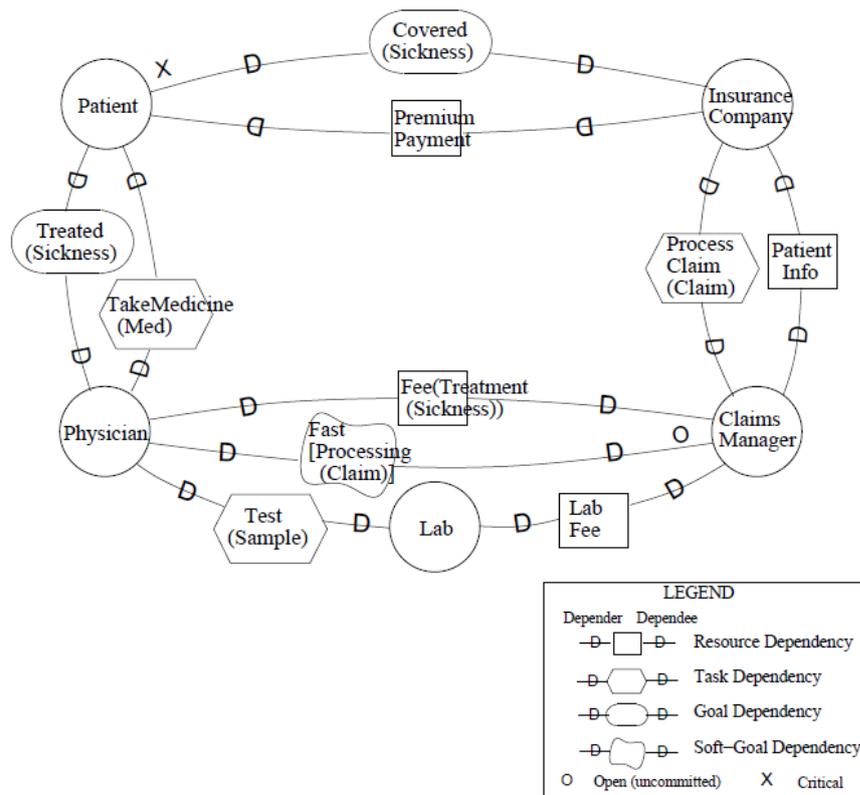
Segundo Yu (1995) o modelo SD é uma rede de dependências entre os atores. Dependência significa que o ator *dependor*, depende de alguém (*dependee*) para alcançar algo, como: um objetivo, tarefa, recurso ou *softgoal*.

O modelo SD analisa quem depende de quem, direto ou indiretamente. Ele tem como objetivo capturar a estrutura intencional do processo, ou seja, ele aborda características de mais alto nível, deixando de fora os detalhes que não são essenciais (Yu, 1995).

Ao contrário de modelos de processos, tais como diagramas de fluxo de dados ou de atividades que indicam fluxo de informação ou de controle de fluxo, o modelo SD tem um maior nível de abstração que descreve o que os atores querem um do outro, e as liberdades

que cada ator tem. Portanto, O modelo SD reconhece a autonomia dos atores em um mundo social (Yu, 2009).

A Figura 2.2 mostra um exemplo do modelo SD que representa a modelagem de um domínio de assistência médica, no qual existem cinco atores, que são: *Patient*, *Physician*, *Lab*, *Insurance Company* e *Claims Manager*. Suas relações são representadas por meio das dependências: o *Patient* depende do *Physician* para o tratamento, e o *Physician* depende do *Patient* para que ele tome os medicamentos e do *Lab* para realizar os exames. O *Physician* depende do gerente para receber o pagamento referente ao tratamento e o laboratório depende do *Claims Manager* para o pagamento da taxa do *Lab*. A *Insurance Company* depende do *Claims Manager* para processar as solicitações. O *Claims Manager* depende da *Insurance Company* para obter informações do *Patient*. A *Insurance Company* para receber o pagamento depende do *Patient*. E o *Patient* depende da *Insurance Company* para cobri-lo em caso de doença.



**Figura 2.2 - Modelo de Dependência Estratégica (SD) (Yu, 1995).**

## Modelo de Razão Estratégica

A análise realizada com o modelo SD é limitada porque esse modelo representa apenas as relações externas dos atores. As motivações e intenções dos atores não são modeladas, porém essa análise também é necessária para obter o conhecimento das intenções internas de cada ator.

No modelo SR para cada ator são atribuídos objetivos, *softgoals*, tarefas e recursos como elementos internos essenciais para alcançar algo (Yu, 2009). O modelo SR é complementar ao modelo SD, porque o modelo SR permite compreender com mais detalhes o sistema a ser desenvolvido. Ele concede um entendimento bem formado sobre as razões estratégicas de cada ator em relação ao processo da organização.

As relações intencionais internas dos atores são descritas no modelo SR como um meio para alcançar determinados objetivos do processo, ou seja, é representado explicitamente como os objetivos são alcançados (Yu, 1997). O modelo permite que as razões associadas ao ator e as suas dependências estejam relacionadas, pois cada objetivo ou tarefa interna justifica as dependências com os outros atores.

Os mesmos elementos de dependência do modelo SD são usados no modelo SR. Além disso, no modelo SR são adicionados as ligações de meio-fim e as ligações de decomposição de tarefas.

Segundo Yu (1995) as ligações de meio-fim indicam um relacionamento entre um fim, por exemplo, um objetivo a ser alcançado ou um *softgoal* a ser satisfeito, e um meio que alcança o fim. Os meios são expressos na forma de tarefas, que representam o modo de como fazer algo. A tarefa é decomposta e seus componentes são ligados a ela por meio de ligações de decomposição de tarefas.

Yu (1995) também expõe o conceito de rotina no  $i^*$  como um subgrafo que representa as racionalidades de um processo, ou seja, uma combinação particular de elementos que constituem meios para alcançar algum fim.

A Figura 2.3 apresenta um exemplo de modelo SR do mesmo domínio do modelo SD apresentado anteriormente. Nesse caso, será explorado o interior do ator *Claims Manager* para mostrar algumas das configurações que a *Insurance Company* pode adotar para processar o plano de tratamento apresentado.

Esse modelo SR (Fig. 2.3) mostra um ambiente de reivindicações de seguro de saúde, onde o *Physician* precisa submeter um plano de tratamento para a *Insurance Company* com o intuito de obter uma aprovação prévia, ou então o tratamento pode não ser reembolsado. A *Insurance Company* verifica que o tipo de tratamento é coberto pela política, e que o tratamento proposto é razoável de acordo com a opinião médica.

O gerente (*Claims Manager*) é quem autoriza a aprovação do tratamento (*ApprovalOfTreatment*) por meio da tarefa Aprovar Tratamento (*ApproveTreatment*). Essa tarefa consiste em dois componentes: um objetivo que é o tratamento ser avaliado (*TreatmentBeAssessed*) e uma tarefa de assinar o documento de aprovação (*SignApprovalDocument*). Uma alternativa para avaliar o plano de tratamento é deixar que um funcionário realize a avaliação do tratamento. Ou, que o próprio *Claims Manager* faça a avaliação. Essa última alternativa requer que o *Claims Manager* verifique a política da *Insurance Company* do *Patient* e avalie o plano de tratamento (*MedicallyAssessed*). Esse objetivo pode ser alcançado por algum *Medical Assessor* que tenha conhecimento para fazê-lo, ou pelo próprio *Manager*, fazendo uso de casos anteriores que estão em um repositório (*ClaimsCasesRepository*).

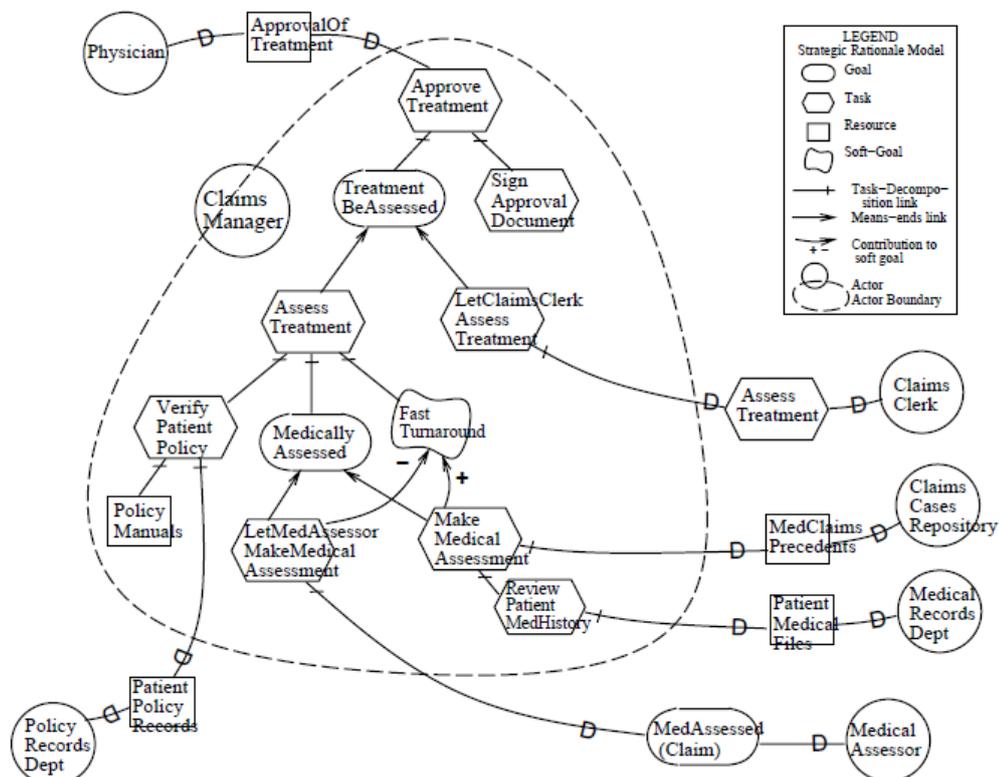


Figura 2.3 - Modelo de Razão Estratégica (Yu, 1995).

## 2.3 Gestão de Processos de Negócio

Nos últimos anos o conceito de empresa colaborativa vem se destacando no meio social e com, a necessidade de comunicação rápida entre os setores e também entre as empresas, consolidou ainda mais o uso da abordagem de processos de negócio (PAIM, 2007).

Davenport (1994) definiu processo como sendo uma união ordenada de atividades específicas de trabalho através do tempo e do espaço, com início, fim e um conjunto definido de entradas e saídas bem especificadas.

Um Processo de Negócio é um conjunto de atividades coordenadas dinamicamente por um número de participantes no ambiente social da organização (Koliadis e Ghose, 2006). Um processo pode ser definido como um conjunto de atividades mensuráveis e específicas que alcançam um objetivo final, podendo ser esse objetivo, um produto ou um serviço.

Os processos de negócio se distinguem pelo setor de atuação da organização e são apoiados por outros processos internos e externos que produzem um produto ou serviço oferecido ao cliente. A Gestão de Processos de Negócio (BPM – *Business Process Management*) se submete a gerenciar os processos e alinhá-los às metas estratégicas da organização. O BPM tem como principal objetivo alcançar as metas organizacionais por meio de melhorias continuadas e interativas na gestão e no controle dos processos da organização.

A gestão de processos de negócio permite que a empresa mantenha uma estrutura cultural e organizacional de comunicação entre os envolvidos no processo, seja dentro ou fora da organização (Paim, 2007). Por isso que a gestão de processos é importante para a organização, pois, é ela que estabelece as estratégias para alcançar os objetivos da empresa.

A gestão de processos consiste em entender o ambiente interno e externo à organização. Ela busca entender, selecionar e priorizar os processos que são importantes para as estratégias de negócio. Essas estratégias são modeladas para atingirem os objetivos estratégicos da organização e as abordagens para alcançar esse fim são gerenciadas por meio do BPM (PAIM, 2007).

Entretanto, a modelagem dos processos só é feita depois de um entendimento adequado da complexidade organizacional. As ferramentas e técnicas de modelagem auxiliam na gestão dos processos como instrumentos de apoio no gerenciamento da organização. A seguir, é apresentada uma notação usada para modelagem de processos de negócio.

### 2.3.1 BPMN

Uma notação de modelagem de processos de negócio bastante conhecida no meio industrial é o BPMN (*Business Process Modeling Notation*) que foi desenvolvido pela BPMI (*Business Process Management Initiative*). A BPMI foi criada para desenvolver o uso do BPM, através do uso de padrões de projetos de processos na implantação, execução, manutenção e otimização dos processos (OWEN e RAJ, 2003).

O principal objetivo da criação do BPMN era que fosse uma notação facilmente compreendida por todos *stakeholders*: usuários, analistas de negócios, desenvolvedores técnicos e gerentes. E, além disso, que diminuísse a diferença entre o projeto de processos de negócio e a implementação dos processos (WHITE, 2004).

O BPMN define um diagrama do processo de negócios, chamado de BPD (*Business Process Diagram*), que se baseia numa técnica de fluxograma adaptados para a criação de modelos gráficos de processos de negócios. Ou seja, um modelo de processos de negócios, é um conjunto de objetos gráficos que estão coordenados por um controle de fluxo que define uma ordem.

Um BPD é constituído por um conjunto de elementos gráficos, que permite um fácil desenvolvimento de diagramas simples que são entendidos por todos os analistas de negócios. Os elementos escolhidos para compor o diagrama são bastante familiares aos modeladores e se enquadram em quatro categorias básicas (WHITE, 2004):

- Objetos de fluxo
- Objetos de conexão
- *Swimlanes*
- Artefatos

Como o desenvolvimento do BPMN é criar um mecanismo simples para a criação de modelos de processos de negócio, e, ao mesmo tempo, ser capaz de lidar com a complexidade inerente aos processos de negócio. O diagrama corresponde a esses dois requisitos conflitantes, pois organiza os elementos gráficos da notação em categorias específicas, e isso ajuda os *stakeholders* a entender o diagrama e seus elementos.

Para modelar um fluxo de processo de negócio, primeiramente são identificados os eventos iniciais do processo, os processos que são realizados, e os resultados finais do fluxo do processo. As decisões e ramificação dos fluxos são modeladas em seguida, usando desvios. E, além disso, um processo pode conter um sub-processo (OWEN e RAJ, 2003).

Os elementos usados na notação BPMN para a modelagem dos processos de negócio são apresentados com mais detalhes a seguir, de acordo com a classificação mostrada anteriormente.

### Objetos de Fluxo

O BPMN possui três elementos de fluxo: evento, atividade e desvio (*gateway*) que estão mostrados na Figura 2.4.



**Figura 2.4 - Objetos de Fluxo (WHITE, 2004, adaptado pelo autor).**

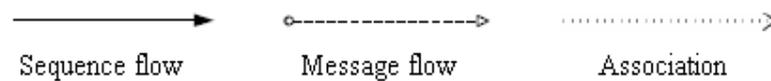
Um evento é representado por um círculo e é algo que acontece durante o curso de um processo. Estes eventos afetam o fluxo do processo e, normalmente, são ativados por uma causa ou resultam em algo. Existem diversos eventos que são sinalizados com diferentes marcadores colocados dentro do círculo. Os tipos de eventos são: eventos iniciais, eventos intermediários e eventos finais.

Uma atividade é representada por um retângulo arredondado e representa um termo genérico para o trabalho que a organização realiza. A atividade pode ser atômica ou composta. Os tipos são: tarefas e sub-processos, que se distingue, na representação, por um sinal de mais no centro do retângulo.

Um desvio é representado por um losango e sua função é controlar a divergência e convergência da sequência de fluxo. As decisões tomadas em um processo determina a fusão, união ou separação do fluxo de atividades.

## Objetos de conexão

Os objetos de fluxo são ligados no diagrama pelos objetos de conexão. Essas ligações estabelecem a estrutura dos processos de negócio. Existem três tipos de objetos de conexão (Fig. 2.5), e estão descritos a seguir.



**Figura 2.5 - Objetos de Conexão (WHITE, 2004, adaptado pelo autor).**

Um fluxo de sequência é representado por uma linha cheia com uma seta sólida e é usado para mostrar a ordem em que as atividades são realizadas em um processo.

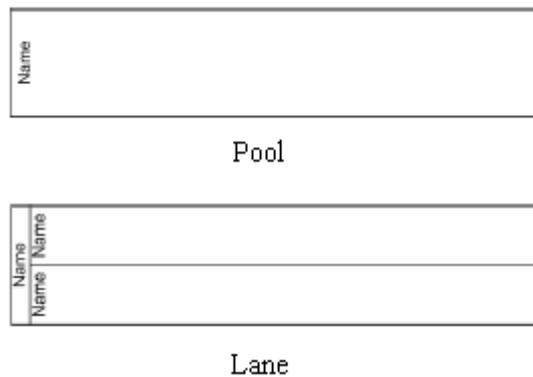
Um fluxo de mensagem é representado por uma linha tracejada com uma seta aberta e é utilizada para mostrar o fluxo de mensagens entre dois participantes do processo que não estão inseridos na mesma entidade empresarial.

Uma associação é representada por uma linha pontilhada e simula a associação de dados, textos e outros artefatos com os objetos de fluxo.

## Swimlanes

O BPMN utiliza o conceito de *swimlanes* como um mecanismo para organizar as atividades em diferentes categorias visuais, que ilustram diferentes responsabilidades funcionais. As *swimlanes* são de dois tipos: *lanes* e *pools* (Fig. 2.6).

Um *pool* representa um participante em um processo e também separa um conjunto de atividades em *lanes*. Enquanto que uma *lane* é uma subpartição dentro de um *pool* e é utilizada para organizar e categorizar as atividades.



**Figura 2.6 – Swimlanes (WHITE, 2004, adaptado pelo autor).**

### **Artefatos**

A notação BPMN tem a capacidade de ampliar o contexto para que a representação do processo seja mais específica. Com isso, qualquer quantidade de artefatos pode ser adicionada em um diagrama, conforme o contexto dos processos de negócio que estão sendo modelados. Os artefatos são: objetos de dados, grupos e anotações.

Os objetos de dados são usados para representar a produção de dados que são necessários para as atividades. Eles estão ligados às atividades por meio de associações. Um grupo é representado por um retângulo arredondado de linhas tracejadas. Ele agrupa as atividades, sem afetar o fluxo do processo. E por fim, as anotações são usadas pelo modelador do processo com o intuito de fornecer informação adicional para o leitor do diagrama.

## **2.4 Trabalhos Relacionados**

### **2.4.1 Abordagem da Metodologia KAOS com BPMN**

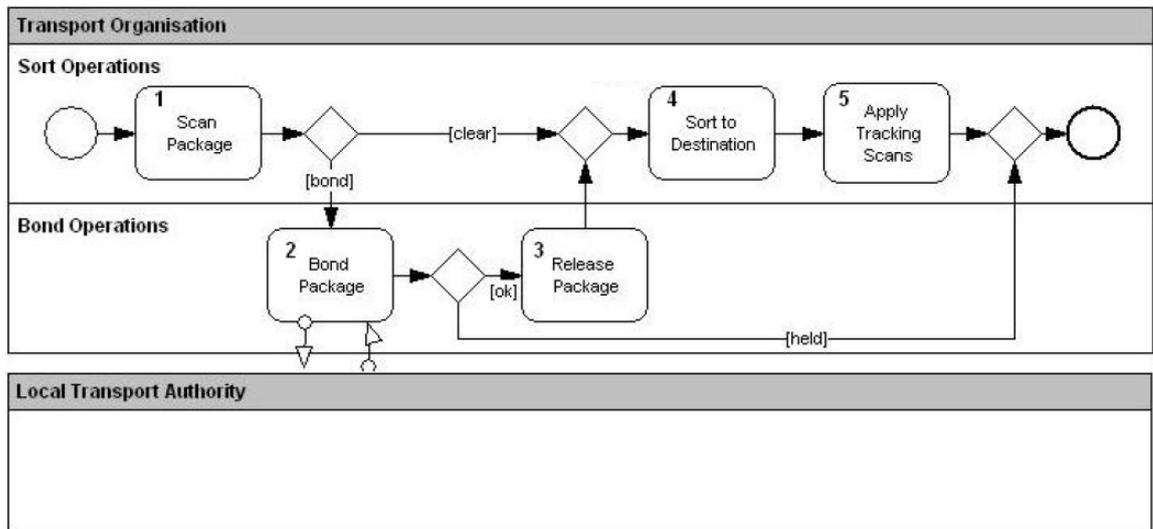
Devido às mudanças e evoluções que ocorrem no ciclo de vida dos processos, é fundamental que as metas dos stakeholders estejam bem especificadas para que quaisquer mudanças propostas só venham a ser implementadas se satisfizerem estas metas. Com o crescimento imprevisível nas organizações, o BPM visa apoiar as evoluções dos processos de negócios. Entretanto, os objetivos de interesse estratégico também devem ser assegurados para que o crescimento organizacional caminhe na direção certa. Com isso, Koliadis e Ghose (2006) propuseram uma metodologia chamada GoalBPM, que apoia a evolução controlada de

processos de negócios através da modelagem explícita das metas dos stakeholders, seus relacionamentos e suas evoluções prováveis. Esse método fornece uma ligação entre os modelos de processos de negócios, usando a modelagem BPMN e modelos orientados a objetivos, usando a metodologia KAOS.

Segundo Dardenne et al. (1993) KAOS (*Knowledge Acquisition in autOated Specification*) é um modelo que consiste no conceito de níveis de metas, suas relações, atributos e restrições. O modelo fornece uma base para uma definição genérica de arquitetura de ambientes, onde só se tem o conhecimento dos objetivos organizacionais pertinentes aos processos de negócios. Uma definição formal é estabelecida em KAOS, permitindo que os objetivos sejam especificados precisamente, apoiando a identificação e resolução de conflitos entre os objetivos.

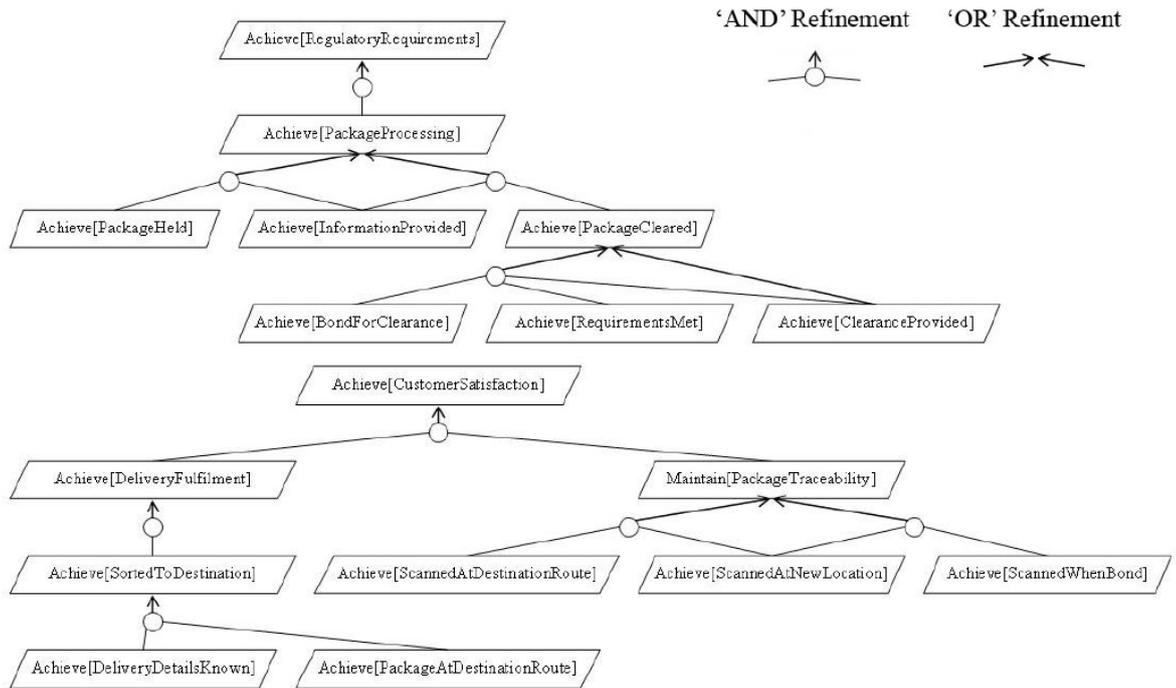
Nessa abordagem, os objetivos são declarados em termos de desejos alcançados. Os objetivos são modelados numa hierarquia, em que objetivos de alto-níveis são mais importantes na organização. A satisfação desses objetivos é complexa, por isso, eles são refinados em sub-objetivos, que são mais operacionais e atribuídos a um grupo menor de indivíduos que serão responsáveis pela sua satisfação. Os refinamentos podem ser dos tipos ‘AND’ ou ‘OR’, em que um refinamento ‘AND’, o objetivo pai só é satisfeito se todos os seus sub-objetivos forem satisfeitos. Um refinamento ‘OR’ afirma que o objetivo pai é satisfeito se pelo menos um sub-objetivo é alcançado (CAILLIAU e LAMSWEERDE, 2012).

A modelagem GoalBPM estabelece o relacionamento entre modelos de objetivos e modelos de processos. Como exemplo, o GoalBPM é aplicado ao contexto de uma Organização de Transporte, especificamente ao núcleo operacional ‘*Sorting Package*’ (Fig.2.7).



**Figura 2.7 - Contexto atual do modelo BPMN (Koliadis e Ghose, 2006).**

Nesse exemplo, a organização precisa fazer uma reformulação para manter a eficácia na triagem dos pacotes. Essa mudança impacta na classificação interna dos pacotes dentro da organização que deve ser realizada por um representante na chegada e na saída dos pacotes, a fim de, facilitar a classificação para a entrega ou encaminhamento do pacote. Mas o objetivo operacional do processo de *'Sorting Package'* é que o encaminhamento dos pacotes seja imediato, d chegada ao centro de triagem, até os seus respectivos destinos. Existem três participantes do processo cujos objetivos devem ser atendidos: A Organização de Transporte (*Transport Organization*) - cujo interesse é o eficiente roteamento dos pacotes para os destinos através da atribuição de responsabilidade a operações internas de classificação; os clientes (*Customers*) - cujo interesse é a entrega rápida e a possibilidade de rastreabilidade da encomenda; e, a Autoridades de Transporte (*Transport Authorities*) – cujo interesse é a manutenção de um alto nível de integridade dos pacotes de triagem. Estes requisitos são representados no modelo KAOS de objetivos da Figura 2.8.



**Figura 2.8 - Processo 'Sorting Package' (Koliadis e Ghose, 2006).**

Koliadis e Ghose (2006) propuseram o relacionamento entre os modelos em dois estágios. Primeiramente, são estabelecidas, por análise, ligações de rastreamento entre objetivos, no modelo de objetivos, para atividades ou eventos finais no modelo de processos (Fig. 2.9). Em um segundo estágio, é estabelecido ligações de satisfação entre os objetivos e os processos (Fig. 2.10). O relacionamento entre objetivos e processos é definido, considerando as condições antes e depois da execução do processo. Uma ligação de satisfação relaciona o processo e o objetivo que ele satisfaz, podendo ser de dois tipos: ligações de satisfação normativas, que relacionam as atividades aos estados desejados; e ligações de satisfação descritivas, que descrevem o “as-is”.

Goal Type	Goal Declaration	Concerned Objects	Antecedent	Consequent
Achieve	PackageProcessing	Package, SortFacility, TransportAuthority	Arrives(p, sf)	$\Leftrightarrow$ Processed(p, ta)
Achieve	InformationProvided	Package, SortFacility, TransportAuthority	Arrives(p, sf)	$\Leftrightarrow$ Provided(p.DeliveryDetails, ta)
Achieve	PackageCleared	Package, TransportAuthority	Provided(p.DeliveryDetails, ta)	$\Leftrightarrow$ Cleared(p, ta)
Achieve	PackageHeld	Package, TransportAuthority	Provided(p.DeliveryDetails, ta)	$\Leftrightarrow$ Held(p, ta)
Achieve	BondForClearance	Package, SortFacility, TransportAuthority	Provided(p.DeliveryDetails, ta)	$\Leftrightarrow$ Bond(p, sf) $\wedge$ Passed(p, ta.Requirements)
Achieve	RequirementsMet	Package, SortFacility, TransportAuthority	Bond(p, sf)	$\Leftrightarrow$ Passed(p, ta.Requirements)
Achieve	ClearanceProvided	Package, TransportAuthority	Provided(p.DeliveryDetails, ta) $\wedge$ Passed(p, ta.Requirements)	$\Leftrightarrow$ Cleared(p, ta)
Achieve	DeliveryFulfillment	Package, SortFacility, Customer	Arrives(p, sf)	$\Leftrightarrow$ Delivered(p, c)
Achieve	SortedToDestination	Package, SortFacility	Arrives(p, sf)	$\Leftrightarrow$ Sorted(p, p.Destination)
Achieve	DeliveryDetailsKnown	Package, SortFacility	Arrives(p, sf)	$\Leftrightarrow$ Known(p.DeliveryDetails, sf)
Achieve	PackageAtDestinationRoute	Package, TransportAuthority	Known(p.DeliveryDetails, sf)	$\Leftrightarrow$ Sorted(p, p.Destination)
Achieve	ScannedAtNewLocation	Package, SortFacility	Arrives(p, sf)	$\Leftrightarrow$ Scanned(p, sf)
Achieve	ScannedAtDestinationRoute	Package, SortFacility	Sorted(p, p.Destination)	$\Leftrightarrow$ Scanned(p, sf)
Achieve	ScannedWhenBond	Package, SortFacility, TransportAuthority	Bond(p, sf)	$\Leftrightarrow$ Scanned(p, sf)

**Figura 2.9 - Rastreabilidade de objetivos (Koliadis e Ghose, 2006).**

Para determinar as ligações de satisfação, são necessárias três etapas. Os autores introduziram o termo: ‘anotações de efeito’, que inicialmente, os modelos de processos são assinalados com essas anotações de efeito. Em seguida, é identificado um conjunto de trajetórias críticas e depois, é identificado um subconjunto de ligações de rastreamento que representa ligações de satisfação pela análise de trajetórias críticas relativas às anotações de efeito do processo. Essas ligações são denominadas descritivas. Por fim, é realizada uma comparação entre ligações de satisfação normativas e ligações de satisfação descritivas para guiar as mudanças do processo.

O conceito de ‘efeito’ foi introduzido por Koliadis e Ghose (2006) como sendo um resultado, ou consequência de uma atividade que é executada por um ator, ou seja, existe um relacionamento de causa entre uma atividade e um efeito que é declarado como um fato, enquanto que a atividade é declarada como um desejo. O efeito foi designado como uma consequência esperada, ou excepcional, ou seja, não esperada.

Koliadis e Ghose (2006) consideraram uma trajetória como um caminho de atividades executadas sequencialmente que resultam em um único resultado. Com isso, a trajetória pode ser tanto normal como excepcional em relação ao efeito acumulado, ou seja, em relação ao seu estado final alcançado. Uma trajetória normal conduz a satisfação de todos os objetivos, enquanto que uma trajetória excepcional satisfaz um número limitado de objetivos.

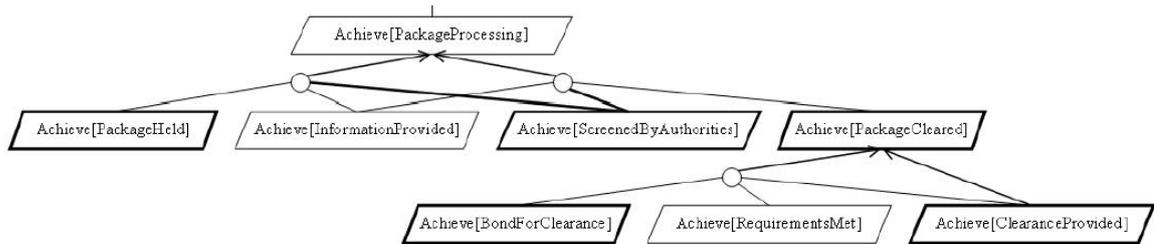
#	Activity	Effect Annotation	T#	A#	Cumulative Effect Assessment	T#	A#	Goal Satisfaction	
	Pre Conditions	Arrives(p, sf) AND Provided(p.DeliveryDetails, ta)	1	0	Arrives(p, sf) AND Provided(p.DeliveryDetails, ta) AND Passed(p, ta Requirements) AND Cleared(p, ta)	1	Pre	InformationProvided	
	Possible Effects	Passed(p, ta Requirements) AND Cleared(p, ta)		1	Known(p.DeliveryDetails, sf) AND Scanned(p, sf)				PackageCleared
1	Scan Package	Known(p.DeliveryDetails, sf) AND Scanned(p, sf)		4	Sorted(p, p.Destination)			1	DeliveryDetailsKnown
				5	Scanned(p, sf)			4	ScannedAtNewLocation
								5	PackageAtDestinationRoute
2	Bond Package	Bond(p, ta) AND Scanned(p, sf)	2	0	Arrives(p, sf) AND Provided(p.DeliveryDetails, ta)	2	Pre	InformationProvided	
		Bond(p, ta) AND Scanned(p, sf) AND Held(p, ta)		1	Known(p.DeliveryDetails, sf) AND Scanned(p, sf)			1	DeliveryDetailsKnown
				2	Bond(p, ta) AND Scanned(p, sf) AND Held(p, ta)			2	ScannedAtNewLocation
3	Release Package	Passed(p, ta Requirements) AND Cleared(p, ta)	3	0	Arrives(p, sf) AND Provided(p.DeliveryDetails, ta)	3	Pre	InformationProvided	
				1	Known(p.DeliveryDetails, sf) AND Scanned(p, sf)			1	DeliveryDetailsKnown
				2	Bond(p, ta) AND Scanned(p, sf)			2	ScannedAtNewLocation
				3	Passed(p, ta Requirements) AND Cleared(p, ta)			3	ScannedWhenBond
				4	Sorted(p, p.Destination)			4	PackageCleared
4	Sort to Destination	Sorted(p, p.Destination)						RequirementsMet	
								ClearanceProvided	
5	Apply Tracking Scans	Scanned(p, sf)					4	SortedToDestination	
								PackageAtDestinationRoute	
							5	ScannedAtDestinationRoute	

**Figura 2.10 – Efeitos, Trajetória e Satisfação dos Objetivos (Koliadis e Ghose, 2006).**

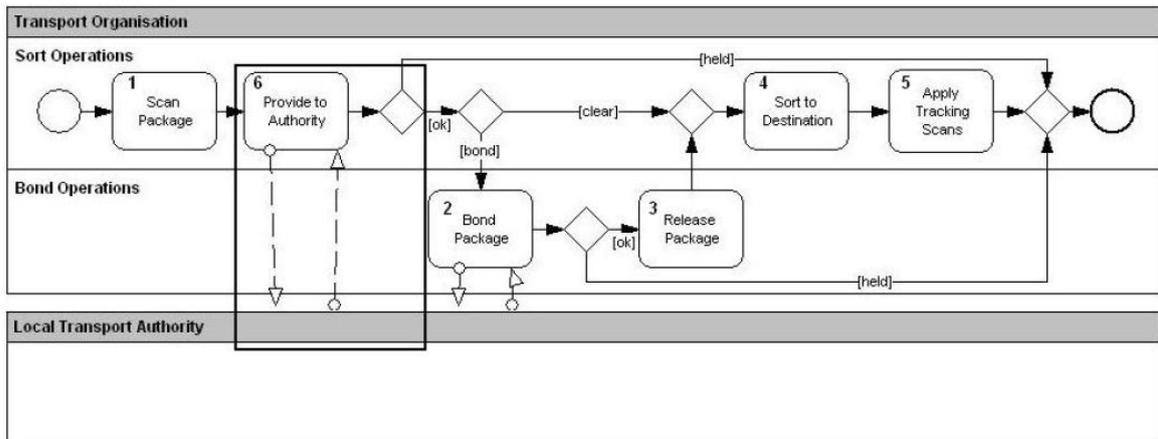
Dado essas considerações, a metodologia GoalBPM consiste em, inicialmente, percorrer cada trajetória do processo e comparar os efeitos com os objetivos de satisfação no modelo de objetivos, seguindo as ligações normativas na ordem temporal dos efeitos. As ligações de satisfação descritivas são estabelecidas através da progressão da trajetória e cada trajetória é classificada como normal ou excepcional. Para que uma relação de satisfação exista entre o modelo de objetivo e o modelo de processo, deve haver pelo menos uma trajetória normal.

Por fim, o resultado do processo é analisado, identificando se o processo alcança os objetivos. O relacionamento de satisfação entre o processo e o modelo de objetivo é classificado como forte, fraco ou insatisfeito. Um relacionamento é classificado como forte quando todas as possibilidades de trajetórias são ‘normal’. Um relacionamento fraco existe quando há pelo menos uma trajetória ‘excepcional’ e uma trajetória ‘normal’. E um relacionamento insatisfeito é o resultado de não existir uma única trajetória ‘normal’ no processo.

Aplicando essas heurísticas ao exemplo apresentando anteriormente, com a adição do novo requisito que foi descrito, que é a seleção dos pacotes, a Figura 2.11 mostra a mudança dos objetivos, por causa do requisito adicionado e conseqüentemente, introduziu um novo efeito desejado no modelo de processos, que é a avaliação de todos os pacotes pela Autoridade de Transporte (Fig. 2.12).



**Figura 2.11 – Adição de Objetivos (Koliadis e Ghose, 2006).**



**Figura 2.12 - Processo Modificado (Koliadis e Ghose, 2006).**

Um trabalho semelhante e realizado pelos mesmos autores, porém usando o Framework i\*, é explicado a seguir.

### 2.4.2 Abordagem do Framework i\* com BPMN

A modelagem de processos de negócios exerce influência sobre o desenvolvimento da organização, tanto na gestão organizacional como nos sistemas presentes na empresa. Por isso, a modelagem dos processos precisa ser bem formulada. Com base nessa premissa, Koliadis et al. (2006) propuseram uma metodologia que visa auxiliar o desenvolvimento de modelos de processos de negócios a partir de modelos conceituais de alto nível, nesse caso, o

i\*. O trabalho desses autores expõe conceitos e métodos que auxiliam a co-evolução da modelagem de processos de negócios e da modelagem orientada a objetivos.

Primeiramente, foram introduzidos dois conceitos para auxiliar no desenvolvimento da metodologia: condições de satisfação e anotações de efeito (KOLIADIS et al. ,2006).

Um efeito é classificado com um resultado ou consequência de uma ação. Sendo assim, anotação de efeito é uma afirmação relativa a uma consequência de uma atividade que está associada a um estado de alteração.

Condições de satisfação estão relacionadas com as tarefas e os objetivos atribuídos aos atores em um diagrama SR do i\*. A condição de satisfação é uma afirmação que deve ser alcançada na finalização de uma ação. As condições de satisfação que são anotadas no modelo i\* reconhecem os efeitos no modelo de processos de negócio.

Para realizar o mapeamento entre o i\* e BPMN, é necessário definir uma rotina que irá definir qual configuração do processo será modelada. De acordo com (Yu, 1995), uma rotina é um sub-diagrama de um modelo SR que representa um determinado curso entre as alternativas. O escopo inclui as sub-tarefas da rotina, as dependências ligadas a essas sub-tarefas e os atores ligados a essas dependências. Assim, Koliadis et al. (2006) forneceram algumas regras para a projeção do escopo do modelo i\* que é baseado na identificação do processo de negócio como uma rotina atribuída ao ator. As regras estabelecidas são essas:

- O nó raiz da rotina e todas as tarefas no seu primeiro estágio de decomposição são para estar dentro do escopo;
- Todas as dependências que estão associadas a uma tarefa dentro do escopo da rotina, onde o ator no controle da rotina é o *dependor* estão dentro do escopo do processo, bem como as tarefas atribuídas ao ator *dependee*;
- Todas as dependências que estão associadas a uma tarefa dentro do escopo da rotina, onde o ator no controle da rotina é o *dependee*, estão dentro do escopo do processo se a tarefa atribuída ao *dependor* é parte de alguma decomposição de uma tarefa no escopo do processo conforme a regra anterior; e também as tarefas atribuídas ao ator *dependor*.

Depois de aplicar essas três regras para determinar o escopo do modelo  $i^*$ , é necessário estabelecer regras de consistência para assegurar a correta relação entre os modelos  $i^*$  e BPMN. Estas regras estão descritas a seguir:

- Cada ator no modelo  $i^*$  deve existir como um participante no BPMN e também suas tarefas, aplicando as regras da projeção do escopo;
- Deve existir uma trajetória no modelo de processo, em que o objetivo prático da rotina é alcançado, e a sequência de atividades é consistente com os requisitos da rotina, descritos a seguir:
  - a) O efeito acumulado de todas as tarefas e objetivos da rotina devem alcançar condições de satisfação acumuladas em pelo menos uma trajetória no modelo de processos;
  - b) A execução da tarefa no lado *depend* de uma dependência não deve ser realizada antes da finalização da dependência em consequência da acumulação de efeitos durante o caminho da trajetória.

Com base nas regras de consistências expostas anteriormente, Koliadis et al. (2006) forneceram quatro passos para o desenvolvimento do modelo BPMN baseado no  $i^*$ :

- 1) Identificar os atores no diagrama  $i^*$ ;
- 2) Mapear os elementos para relacioná-los dentro do modelo BPMN:
  - a) Mapear os participantes: A organização ou ator externo no modelo  $i^*$  é representado com um *pool* em BPMN. Os atores internos são representados como *lanes* dentro do *pool*;
  - b) Mapear as atividades. Tarefas dentro do  $i^*$  representam ou um sub-processo ou uma atividade atômica no BPMN.
- 3) Introduzir as ligações de sequência de fluxo por análise das condições de satisfação considerando a segunda regra de consistência. Ligações de fluxo de mensagem (entre *pools*) podem existir, analisando, também, as condições de satisfação.

- 4) Elaborar um sub-processo. A escolha para introduzir tarefas ou sub-processos no diagrama BPMN para especificar tarefas no modelo  $i^*$  é feita no passo 2.

Koliadis et al. (2006) ainda propuseram os passos para a partir do BPMN, gerar o modelo  $i^*$ . A seguir, têm-se os passos para derivar o  $i^*$  a partir do BPMN.

- 1) Os pools e raias no BPMN representam atores no  $i^*$ .
- 2) Representar no  $i^*$  as atividades e sub-processos que são ‘primitivamente executáveis’.
- 3) Aplicar o raciocínio intencional para identificar objetivos e *softgoals*:
  - a. Consultar a intenção das tarefas para identificar objetivos;
  - b. Consultar a intenção das ligações de fluxo. Analisando o fluxo de mensagem entre fronteiras dos atores para identificar objetivos, tarefas e recursos dependentes.
- 4) Identificar *softgoals* no  $i^*$  que não estão no escopo da notação BPMN.

A abordagem de Koliadis et al. (2006) apresenta muitas lacunas no mapeamento entre os modelos  $i^*$  e BPMN e que dependem da experiência e julgamento dos analistas. Por isso, o objetivo desse trabalho é aprimorar a abordagem apresentada, para que as heurísticas de mapeamento entre os modelos sejam o mais sistemáticas possível, a fim de obter modelos de processos de negócio que estejam alinhados aos objetivos da organização.

### **2.4.3 Desafios Práticos no Mapeamento $i^*$ - BPMN**

Nessa subseção, vamos apresentar as métricas usadas por Decreus et al. (2009) para comparar os métodos de transformação de modelos  $i^*$  para modelos BPMN.

Os autores analisaram seis métodos que obtém modelos BPMN a partir de modelos  $i^*$  e identificaram alguns desafios no mapeamento entre os modelos. Esses desafios precisam ser abordados para que a transformação seja efetiva e o propósito para obter um modelo de processos de negócio alinhado aos objetivos da organização seja alcançado.

Visto que o mapeamento entre os modelos ainda é bastante imaturo, foram analisados os problemas que cada método enfrenta para garantir a consistência entre os modelos

mapeados. Para essa análise, Decreus et al. (2009) se concentraram nos métodos e em um conjunto de variáveis (detalhadas a seguir).

O primeiro pressuposto é que o método seja claro. Assim, as variáveis avaliam as propriedades metodológicas da transformação entre os modelos, como: nível de detalhe fornecido por passo, nível de formalidade, mapeamentos conceituais entre os objetivos e os processos, e consistência entre os modelos. Em segundo lugar, também deve estar claro o que o passo deve realizar.

Um segundo conjunto de variáveis é utilizado para avaliar as propriedades organizacionais da transformação do modelo  $i^*$  para BPMN. São elas: a atribuição de responsabilidade em cada passo do método, o grau em que os modelos de processo são orientados a negócio, o nível em que os modelos de processos são considerados técnicos, à medida que a estrutura organizacional é suportada, ou seja, as hierarquias nos departamentos e o nível de atenção dada à modelagem do contexto organizacional, como as relações entre os atores.

Decreus et al. (2009) classificou a avaliação dos métodos usando 0%, como pior caso, 25%, como um resultado baixo, 50%, na média, 75%, foi considerado um resultado alto e 100%, o melhor resultado. A seguir, vamos detalhar apenas os resultados do método proposto por Koliadis et al. (2006).

### **Propriedades Metodológicas**

Os detalhes das propriedades metodológicas da transformação dos modelos de objetivos para os modelos de processos de negócio são explicados a seguir.

O nível de detalhes por passos é uma variável que classifica os métodos de acordo com as informações que são mostradas do início ao fim do processo de transformação. Os autores classificaram o método de Koliadis et al. (2006) com uma pontuação de 75%, pois afirmaram que a abordagem foi elaborada com muitos detalhes no mapeamento entre os modelos.

Na variável, formalidade do algoritmo, nenhum dos métodos forneceu detalhes considerados para uma transformação formal. O método de Koliadis et al. (2006) obteve pontuação de 75%, devido as suas regras e etapas semiformais. Porém o método propõe a aplicação do "raciocínio intencional", sem especificar como realizar essa etapa, por isso, não ficou claro como sequenciar as atividades do processo usando as condições de satisfação.

Os mapeamentos conceituais consistem na relação entre os objetivos e os processos de negócios, ou seja, devem existir regras de mapeamento que sejam claras para orientar no mapeamento. Koliadis et al. (2006) fornece apenas mapeamentos semiformais, por isso recebeu 75% nessa variável.

A consistência entre os modelos é uma variável de verificação da credibilidade dos processos de negócios obtidos em relação ao modelo de objetivos original. Como Koliadis et al. (2006) estabeleceu regras semiformais de verificação, então o seu método recebeu nessa variável a nota de 75%, embora essa verificação possua várias lacunas que deixam a critério do raciocínio do analista.

### **Propriedades Organizacionais**

Do ponto de vista organizacional, é fundamental saber de quem é a responsabilidade em cada passo do método. Koliadis et al. (2006) afirmam que um analista deve executar o método, mas não fornecem detalhes do passo a passo e, por isso obteve pontuação de 50%.

A variável modelo de processos orientados a negócio avalia o grau em que os métodos são independentes de tecnologia. Todos os métodos analisados ganharam nota 100%. Enquanto que a variável modelo de processos orientados a tecnologia, avalia o grau de dependência a uma plataforma específica. O método de Koliadis et al. (2006) obteve nota de 25%, pois não fornece detalhes sobre tecnologia, mas inclui detalhes técnicos de rotinas.

A estrutura organizacional avalia à medida que o método utiliza a estrutura da organização dos atores, ou seja, a hierarquia dos relacionamentos entre os atores. O método dos autores estudados recebeu nota de 25%, pois não organiza os objetivos hierarquicamente. E a modelagem do contexto avalia o nível de atenção dada à modelagem dos atores e os relacionamentos entre os atores. O método de Koliadis et al. (2006) recebeu 75%.

### **Resultados**

A partir da análise realizada, Decreus et al. (2009) concluiu que o método proposto por Koliadis et al. (2006) foi um dos métodos mais eficazes de um ponto de vista global. Ele obteve melhor pontuação nas propriedades metodológicas, enquanto que o método proposto por Bleistein et al. (2006) obteve melhor pontuação nas propriedades organizacionais.

Decreus et al (2009) ainda identificou as limitações que precisam ser superadas pelos métodos de mapeamento estudados, que são: falta de atribuição de responsabilidade de funções, mapeamento insuficiente de conceitos , informalidade nas regras de mapeamento, nenhum suporte para estrutura organizacional e falta de checagem nas consistências dos modelos envolvidos no mapeamento.

Nosso estudo irá se basear no método proposto por Koliadis et al. (2006). Vamos abordar algumas das limitações apresentadas por Decreus et al. (2009), identificar outras limitações e, com base nelas, vamos propor melhorias na abordagem para que modelos BPMN sejam modelados de uma forma sistemática a partir de modelos i\* e vice-versa.

## Capítulo 3

# Aplicação do Método Atual de Mapeamento

Nesse capítulo, vamos aplicar o método proposto por Koliadis et al. (2006) a um exemplo já conhecido do meio acadêmico. Primeiro vamos aplicar a abordagem, descrevendo com detalhes o processo e, em seguida, vamos apontar os problemas existentes no mapeamento realizado segundo a abordagem. A metodologia dos autores citados já foi explicada no Capítulo 2.

### 3.1 Descrição do Processo

Vamos aplicar o método a uma modelagem de gerenciamento de indenização de seguros de saúde (Fig. 3.1), em que o médico (*Physician*) precisa obter uma pré-aprovação da companhia de seguros (*Insurance Company*) para tratar o paciente (*Patient*), a fim de receber, posteriormente, o pagamento pelo tratamento. O processo é explicado a seguir.

Do lado paciente, o objetivo geral é estar bem (*BeWell*). Uma forma para estar bem é comprando o seguro (*Buy insurance*) e recebendo tratamento quando ficar doente (*Get Treated*). O paciente depende do médico para receber tratamento (*Treated*) e da companhia de seguros para cobrir as despesas durante o tratamento (*Covered*).

Para o médico, a prática da medicina inclui o objetivo de curar os pacientes (*PatientBeCured*) e um *softgoal* (*Viable Practice*) para que a prática seja viável. Existem muitas maneiras para alcançar o objetivo de curar os pacientes. O processo de tratamento modelado envolve o diagnóstico da doença (*Diagnose Sickness*), o tratamento da doença (*Treat Sickness*) e o pagamento pela companhia de seguros (*Bill InsurCo*). O tratamento da doença depende do recebimento da pré-aprovação do tratamento pela companhia de seguros (*PreApproval*), e do paciente tomar os medicamentos prescritos (*Take Medication*).

Para a companhia de seguros, uma preocupação importante na gestão dos seguros de saúde é que sejam rentáveis. Isso envolve políticas de vendas (*Sell Policy*) para os pacientes, como a venda de apólices *premium*, dependendo dos pacientes para receber o pagamento (*Premium Payment*), e o processamento de reivindicações (*Process Claim*).

O processamento de reivindicações inclui a pré-aprovação do tratamento (*Approve Treatment*) e o reembolso do médico (*Reimburse Treatment*) quando o tratamento terminar. A necessidade da pré-aprovação contribui negativamente ao processo de reivindicações rápido (*Fast*), que é importante para a diminuição dos custos administrativos (*LowAdminCosts*). Por





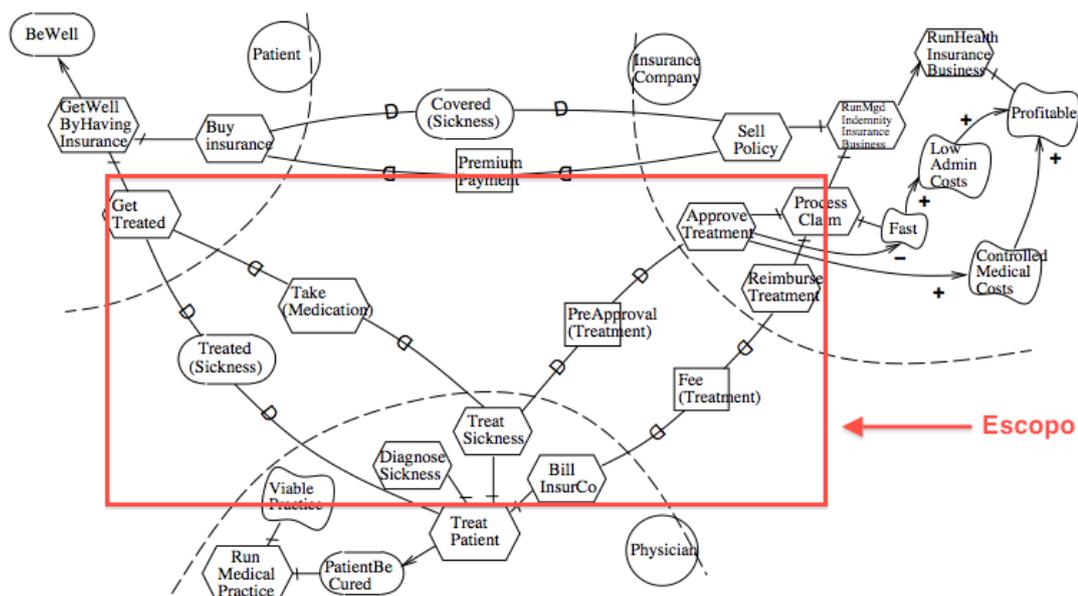


Figura 3.3 - Escopo do processo (YU, 1995, adaptado pelo autor).

<i>Tarefa/Dependência</i>	<i>Condições de Satisfação</i>	<i>Anotação da Tarefa</i>
PH: TreatPatient	Paciente esteja curado	1
PH: Diagnose Sickness	Paciente esteja curado	1
PH: Treat Sickness	Paciente esteja curado	1
PH: Bill InsurCo	Fazer conta dos gastos médicos	5
	Entregar conta dos gastos médicos	5
PA: Get Treated	Estar bem	4
IC: Approve Treatment	Tratamento Aprovado	2
PH -> PA: Take(Medication)	Doença seja curada	3
PH -> IC: PreApproval (Treatment)	Tratamento Aprovado	2 (mensagem)
PA -> PH: Treated (Sickness)	Estar bem	4

Tabela 3.1 - Anotação de satisfação para cada tarefa/dependência.

Respeitando as dependências, ou seja, uma tarefa que é dependente de outra tarefa, não deve ser realizada primeiro, devido ao conceito de efeito acumulado, definido por Koliadis et al. (2006). Por exemplo, a tarefa de aprovar o tratamento (*Approve Treatment*) deve ser realizada antes da tarefa de tratar a doença (*Treat Sickness*), pois no modelo  $i^*$ , existe uma dependência entre essas tarefas, em que, *Treat Sickness* é dependente de *Approve Treatment*.

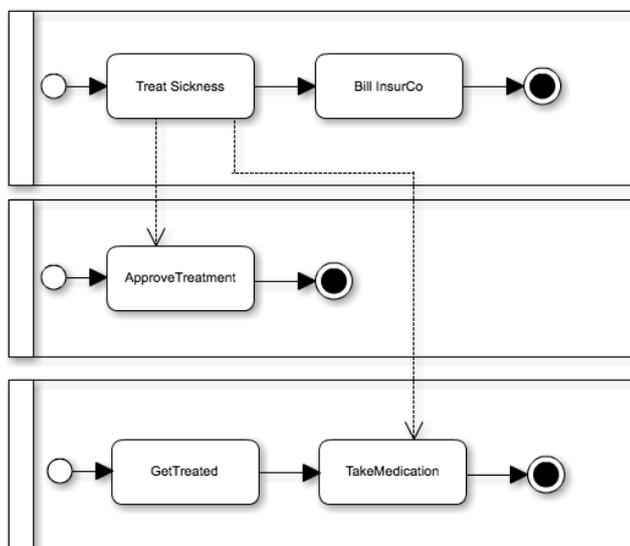
As anotações das tarefas foram adquiridas analisando o processo descrito por Yu (1995) e respeitando o conceito de efeito acumulado. Como o processo se caracteriza pela obtenção de uma indenização para o médico pelos cuidados realizados ao paciente, o processo se inicia com o ator médico tratando o paciente, descobrindo a doença, pedindo a aprovação e por fim, enviando à companhia de seguros a conta do tratamento.

Para finalizar o mapeamento aplicamos as regras de consistência da evolução para garantir que o processo foi mantido entre os dois modelos.

Primeiro, verificamos se cada ator presente no modelo  $i^*$  está presente no modelo BPMN de acordo com o a regra do escopo. Além disso, as tarefas pertencentes a cada ator também devem está no modelo BPMN.

Devemos garantir que exista pelo menos uma trajetória no modelo de processo, em que o objetivo geral deve ser alcançado, ou seja, os efeitos acumulados de todas as tarefas e objetivos devem alcançar as condições de satisfação para garantir o objetivo final. Além disso, a execução da tarefa no lado *dependor* não deve ser realizada antes da finalização da dependência.

O modelo BPMN gerado está representado na Figura 3.4.



**Figura 3.4 - Modelo BPMN gerado do modelo  $i^*$ .**

Os atores representados no  $i^*$  ficaram cada um em um *pool*, pois eles não fazem parte da mesma organização. As atividades foram colocadas no seu respectivo *pool*, e as dependências de tarefas são colocadas do lado *dependee*, mas o fluxo da mensagem vai do lado *dependor* para o *dependee*.

Como pode ser visto no diagrama BPMN gerado (Fig. 3.4), o processo modelado possui poucos detalhes e algumas tarefas foram excluídas do processo. As mensagens entre os *pools* não estão bem definidas e existe pouca informação sobre o fluxo do processo.

### 3.2.2 Mapeamento do BPMN para o i\*

Primeiramente mapeamos os participantes do BPMN diretamente para atores no i\*. Suas atividades também são mapeadas diretamente e aplicando o ‘raciocínio intencional’ identificamos as dependências e a motivação dos atores, para identificar os objetivos, recursos e possíveis *softgoals* que não são modelados no BPMN.

A Figura 3.5 representa o modelo i\* obtido do modelo BPMN da Figura 3.4. Como os passos fornecidos por Koliadis et al. (2006) são muito primitivos, o i\* gerado difere bastante do i\* original do processo. Além disso, como o modelo BPMN possui pouca informação, tivemos que inferir algumas dependências, como, as dependências de recurso: *PreApproval* e *Fee*.

Como os *softgoals* não são modelados em BPMN, não existe uma regra a ser seguida para incluí-los no modelo i\* gerado. Eles podem ser inferidos pelo analista, através da inspeção da intenção dos atores. Desse mesmo modo, os objetivos podem ser atribuídos aos atores, observando a intenção das atividades e sub-processos.

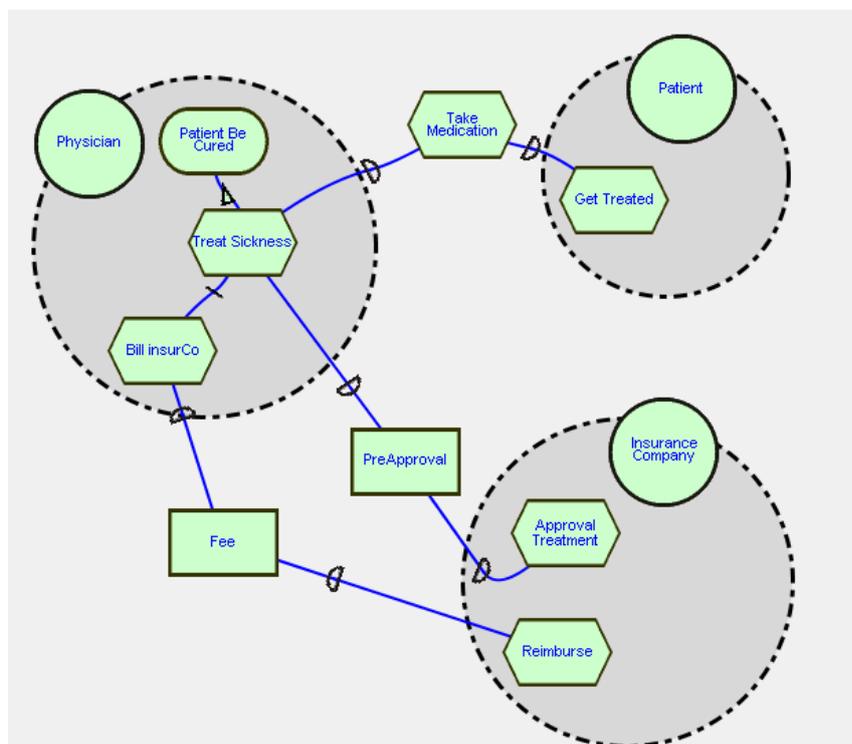


Figura 3.5 - Modelo i\* gerado do BPMN

A seguir, vamos listar as lacunas encontradas no mapeamento apresentado e no próximo capítulo iremos propor melhorias na abordagem.

### 3.2.3 Limitações do Método Aplicado

Nessa seção vamos apresentar as partes do mapeamento proposto por Koliadis et al. (2006) que não estão bem definidas ou não existe um procedimento sistemático. Nossa proposta é obter um método o mais sistemático possível, por isso, vamos nos concentrar em encontrar essas falhas e depois apontar as possíveis melhorias.

(i) Na identificação da rotina não fica claro qual rotina deve ser escolhida. Visto que, a rotina é um caminho alternativo, pode existir mais de uma rotina no processo. Nesse caso, o analista deve conhecer o processo para que a escolha da rotina a ser modelada primeiro seja a melhor possível.

(ii) Na definição do escopo, os autores só incluem as dependências, em que o proprietário da rotina é o *dependee*, se a tarefa do lado *dependee* faz parte de alguma decomposição de tarefa que já está dentro do escopo. Como vimos na seção anterior, uma dependência não foi incluída no mapeamento, e isso pode ser um problema, pois, o processo se torna incompleto, devido à falta dessa dependência. Se a tarefa do lado *dependee* está dentro do escopo, isso significa que o ator da rotina precisa realizar essa tarefa para alcançar seu objetivo principal.

(iii) As anotações de efeito e as condições de satisfação são obtidas através de inferência por parte do analista. Um analista pode interpretar de uma maneira e outro analista de uma maneira totalmente diferente. No nosso exemplo apresentado na seção anterior, a atividade *Treat Patient* poderia ter como condição de satisfação ‘Paciente Tratado’ e a tarefa *Treat Sickness* poderia ter como condição de satisfação ‘Doença Curada’. Sendo assim, o processo modelado em BPMN poderia conter as duas atividades, ou apenas a primeira atividade como um sub-processo.

O conceito de anotações de efeito é importante para garantir que uma tarefa dependente de outra não seja executada primeiro, mas quando o analista está modelando o processo, sua interpretação pode modificar o processo. Isso também ocorre com as condições de satisfação atribuídas a cada tarefa e dependência. Uma condição de satisfação pode ser interpretada de diversas formas. Pode ser um objetivo a ser alcançado ou um estado temporário que deve ser atingido para que a próxima tarefa seja executada.

No trabalho publicado por Koliadis et al. (2006) podemos observar que depois das condições de satisfação, muitas dependências ficaram de fora do processo, devido as anotações das tarefas serem as mesmas.

(iv) Não existe um critério definido para transformar as dependências do modelo i\* para o BPMN. Não tem uma regra para dependências de recurso, objetivo e tarefa.

(v) Não tem uma regra para definição de sub-processo. Transformar a tarefa em atividade atômica ou sub-processo depende da interpretação do analista.

Esse método possui várias partes em que o analista deve usar de seu conhecimento e seu raciocínio para concluir as etapas da transformação. Como Decreus et al. (2009) identificaram, o método possui limitações que precisam ser superadas, como: a falta de definição do propósito dos passos descritos, o mapeamento insuficiente de conceitos entre os modelos e a informalidade das regras de mapeamento.

## Capítulo 4

### Proposta de um Novo Método de Mapeamento

Nesse capítulo será apresentada a contribuição desse trabalho que é a melhoria do método definido por Koliadis et al. (2006) para obter um modelo BPMN diretamente de um modelo  $i^*$  e vice-versa. As heurísticas para obter um modelo BPMN de um modelo  $i^*$  estão descritas na seção seguinte. Depois de definidas as heurísticas de mapeamento, aplicaremos o método a um exemplo, primeiro transformando um modelo  $i^*$  em um modelo BPMN e, em seguida, fazemos o processo inverso. Esse capítulo finaliza com uma comparação entre o nosso método e o método proposto por Koliadis et al. (2006).

A metodologia desse trabalho foi identificar os pontos fracos do método já existente e, com base nisso, propor melhorias no método para obter um mapeamento mais sistemático e menos dependente da experiência do analista.

#### 4.1 Heurísticas de Mapeamento

Com base no método de Koliadis et al. (2006), propomos heurísticas de mapeamento do modelo  $i^*$  para o BPMN e vice-versa. As três primeiras heurísticas foram baseadas no método proposto pelos autores citados, e as sete últimas foram adicionadas para obter um método mais sistemático. O conjunto completo de heurísticas é detalhado a seguir:

- (i) Primeiramente identificamos a rotina e o escopo do processo. O conceito de rotina já foi explicado anteriormente. Porém, o escopo do processo deve ser obtido, incluindo as sub-tarefas da rotina no primeiro nível da decomposição, as dependências ligadas a essas sub-tarefas, independente se o proprietário da rotina é o ator *dependor* ou *dependee* e os atores que participam dessas dependências, como também as tarefas conectadas a elas. Cada rotina identificada no processo criará um modelo de BPMN diferente.
  - a. Se na decomposição da rotina existir um sub-objetivo ou sub-recurso que é alcançado por uma tarefa, essa tarefa deve estar dentro do escopo e sua transformação no BPMN será de uma atividade atômica. Se a tarefa for decomposta, no BPMN, ela será um sub-processo.
- (ii) Cada ator presente no escopo será transformado em um participante no modelo BPMN.

- a. Atores que não pertencem à mesma organização ficarão em *pools* diferentes.
  - b. Atores que pertencem à mesma organização ficarão no mesmo *pool*, mas em *lanes* diferentes.
- (iii) As tarefas internas dos atores presentes no escopo são incluídas como atividades atômicas nas *lanes/pools* dos participantes correspondentes no modelo BPMN.
- (iv) Se a tarefa dentro do escopo é decomposta, essas sub-tarefas devem ser analisadas:
- a. Se as sub-tarefas devem ser realizadas em paralelo, elas tornam-se atividades paralelas dentro da *lane/pool* do participante correspondente.
  - b. Se as sub-tarefas devem ser realizadas em sequência, elas tornam-se atividades conectadas através do fluxo de sequência.
  - c. Se a tarefa em questão não for a rotina escolhida do processo e sua decomposição possui mais de um nível de decomposição, no modelo BPMN esta tarefa se tornará um sub-processo e suas sub-tarefas se tornarão o detalhamento deste sub-processo. O detalhamento do sub-processo deve ser feito de acordo com a regra (iv.a) e (iv.b).
- (v) Se existir algum recurso interno dos atores dentro do escopo, no BPMN esse recurso será um artefato gerado pela atividade atômica ou sub-processo correspondente a tarefa que alcança esse recurso.
- (vi) Uma dependência de tarefa é incluída como uma atividade na *lane* correspondente do ator *dependee* e a mensagem de fluxo ou a ligação de controle de fluxo é direcionada a atividade correspondente do ator *depender*.
- (vii) Uma dependência de recurso é transformada em um artefato produzido pela atividade presente no participante que representa o ator *dependee*. Duas mensagens de fluxo ou controle de sequência de fluxo são adicionadas entre as atividades presentes nos participantes mapeados. O ator *depender* requisita o artefato e o ator *dependee* fabrica e envia o artefato para o ator *depender*. Essas duas ligações são conectadas em sentido opostos e foram adicionadas ao BPMN, porque, quando um ator necessita de um recurso, ele requisita a outro ator e esse ator, realiza a atividade que fabrica o recurso e responde a solicitação, enviando o artefato.

- (viii) Um objetivo torna-se um evento final porque é o estado que os atores participantes do processo querem alcançar.
  - a. Se o objetivo é uma dependência, o evento final é incluído na *lane/pool* do ator *dependender* correspondente.
  - b. Se o objetivo é um elemento interno de um ator, o evento final é incluído na *lane/pool* do ator correspondente.
- (ix) A tarefa raiz relacionada com a rotina escolhida torna-se o evento inicial que desencadeia o processo.
- (x) Qualquer tarefa que é decomposta em mais de um nível de decomposição, será um sub-processo no modelo BPMN.

Com base nessas heurísticas apresentadas, propomos algumas regras de consistência de elementos entre os modelos.

1. Todo ator é necessariamente um participante no modelo BPMN.
2. Todas as tarefas internas dos atores, são atividades internas na *lane/pool* do participante correspondente.
3. Toda dependência deve ter uma ligação de mensagem ou controle de fluxo.
  - a. Se for uma dependência de recurso, deve ter duas ligações entre as atividades e um artefato gerado.
4. Todo objetivo é um evento final não vazio.
5. O evento inicial do processo será a tarefa que realiza o objetivo geral.

O processo inverso é aplicado ao modelo BPMN para obter o modelo  $i^*$ . As heurísticas estão apresentadas a seguir.

- (i) Cada participante que corresponde a *lane* ou *pool* no modelo BPMN é um ator no modelo  $i^*$ .
- (ii) As atividades atômicas dentro da *lane* ou *pool* deve ser uma tarefa interna do ator.
- (iii) Ligações de fluxo de mensagem ou ligações de controle de fluxo entre *pools/lanes* se tornarão dependências entre os atores.

- a. Se a ligação entre as atividades gerará um artefato, essa dependência no modelo  $i^*$  será uma dependência de recurso. O ator *dependee*, dessa dependência, é o ator que produz o recurso.
- (iv) Um evento final não vazio pode tornar-se, dependendo do julgamento feito pelo analista, um objetivo interno relacionado com a rotina que está sendo modelada ou pode ser transformado em uma dependência de objetivo.
  - a. No primeiro caso, o objetivo é interno ao ator que possui o evento final.
  - b. No último caso, o ator *dependee* da dependência de objetivo no modelo  $i^*$  é o ator que possui o evento final. Como o evento final é do ator, isso é um objetivo que ele deseja alcançar, mas ele depende de outro ator (*dependee*) para atingir o seu objetivo.
- (v) A sequência de atividades no modelo BPMN deve ser analisada, e dependendo do julgamento feito pelo analista, pode se tornar sub-tarefas de alguma decomposição de tarefa ou uma tarefa sem um nó pai e filhos.
- (vi) Os sub-processos são transformados em decomposição de tarefas. E suas tarefas são transferidas para o modelo  $i^*$  de acordo com as regras acima, dependendo da interpretação do analista.
- (vii) O evento inicial, que desencadeia o processo, será transformado na tarefa que alcança o objetivo geral do processo.
- (viii) Como os *softgoals* não são modelados no BPMN, eles podem ser inferidos pela pesquisa de atributos de qualidade associados às atividades desenvolvidas pelos participantes.

As regras de consistência de elementos do modelo BPMN para o  $i^*$  são:

1. Todo participante é necessariamente um ator no modelo  $i^*$ .
2. Todas as atividades internas dos participantes são tarefas internas dos atores correspondentes.
3. Toda ligação de mensagem ou controle de fluxo deve ter uma dependência correspondente no modelo  $i^*$ .
  - a. Se forem duas ligações no sentido oposto entre atividade e uma atividade gera um artefato, a dependência será dependência de recurso.
4. Todo evento final não vazio é um objetivo.

- O evento inicial do processo será a tarefa que realiza o objetivo geral.

#### 4.1.1 i\* para BPMN

A abordagem proposta foi aplicada no processo de "*Managed Indemnity Insurance*" apresenta e no Capítulo 3. A Figura 4.1 representa o modelo BPMN gerado a partir do modelo i\* da Figura 3.1.

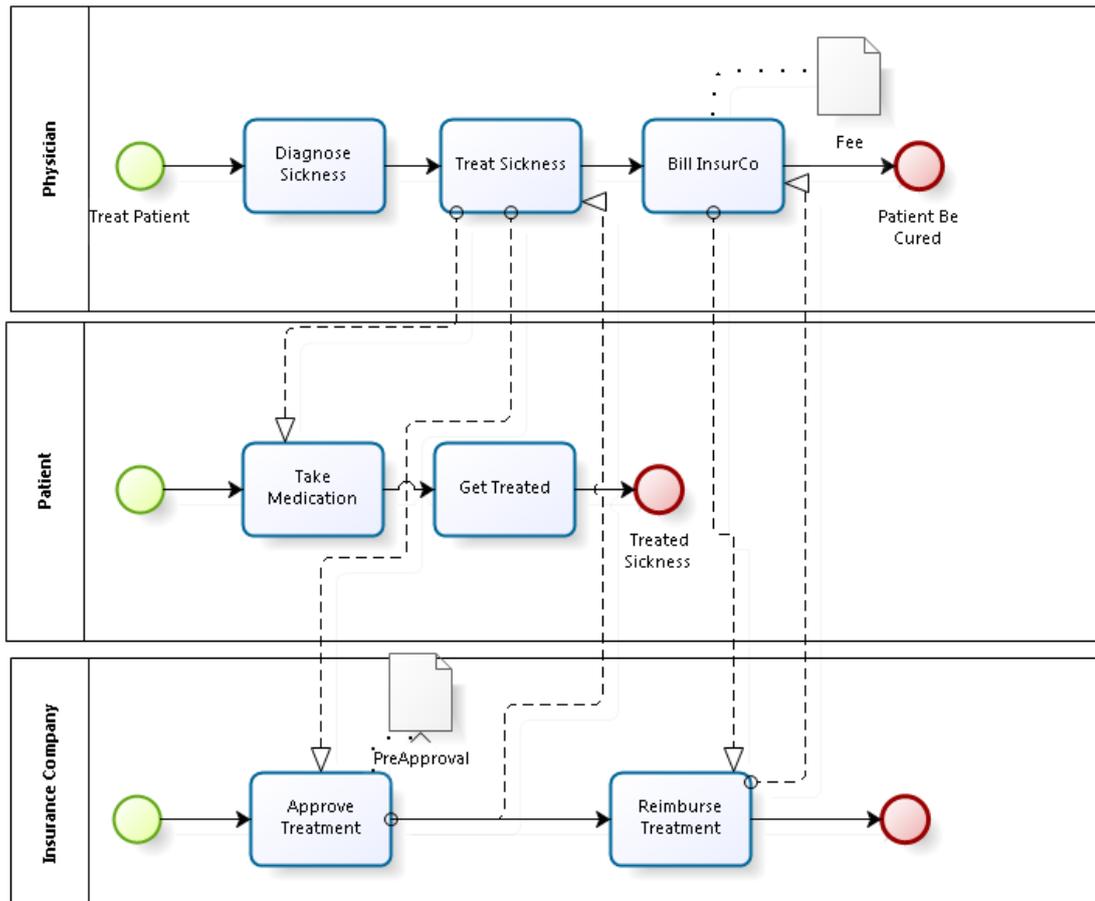


Figura 4.1 - Modelo BPMN gerado do modelo i\* da Fig. 3.1.

Esse modelo BPMN (Fig.4.1), foi obtido após a realização das heurísticas de mapeamento propostas. Nós escolhemos a rotina *TreatPatient* presente no ator médico (*Physician*). As sub-tarefas dessa rotina foram incluídas como atividades internas no *pool* do participante médico. Os outros atores: paciente (*Patient*) e a companhia de seguros (*Insurance Company*) foram incluídos no modelo BPMN como *pool* cada um, porque eles não fazem parte da mesma organização.

A rotina escolhida, no modelo BPMN é o evento que inicia todo o processo e os objetivos foram transformados nos eventos finais não vazios. O objetivo geral está no pool do ator proprietário da rotina, porque essa rotina é o sub-diagrama que realiza esse objetivo. Como o processo só deve terminar quando o objetivo é alcançado, o evento final se caracteriza pelo objetivo atingido. Já o objetivo presente em paciente, significa que é um objetivo apenas do participante e não um objetivo geral. Por essa razão, que o objetivo é colocado no *pool* correspondente ao participante paciente.

As dependências presentes no modelo *i\** estão presentes no modelo BPMN na forma de ligações de mensagem de fluxo. As duas dependências de recurso: *Fee* e *PreApproval* estão como artefatos gerados nas atividades dessas ligações.

De acordo com o conhecimento prévio que o analista tem do processo, é preciso inferir a ordem das atividades internas de cada participante. Essa etapa é a única que não possui um passo sistemático a ser seguido, dependendo apenas do julgamento do analista.

Nesse exemplo, para que o médico trate a doença do paciente, ele precisa antes, de uma pré-aprovação da companhia de seguros para realizar o tratamento. Em seguida, o médico deve encaminhar à companhia, a conta dos gastos médicos para que possa ser indenizado. No caso do paciente, ele precisa tomar os medicamentos antes, para tratar a doença e assim, ficar bem.

Com um conhecimento antecipado do processo de indenização das companhias de saúde, podemos afirmar que elas já conhecem cada tipo de tratamento para determinada doença, por isso, que elas podem pré-aprovar o tratamento, pois já sabem os gastos médicos, e só depois, que o médico recebe o pagamento pelo tratamento realizado.

#### **4.1.2 BPMN para *i\****

Aplicando o método inverso para obter um modelo *i\** a partir de um modelo BPMN, obtemos o modelo mostrado na Figura 4.2. O exemplo usado foi o mesmo, e o modelo BPMN utilizado para aplicar as heurísticas da transformação foi o da Figura 4.1.

Como podemos observar, o modelo *i\** atual difere do modelo *i\** original (Fig. 3.1), porém, a integridade do processo é mantida. Os objetivos estão presentes no modelo gerado, bem como as dependências.

Como o *dependee* é quem realiza a tarefa, então a dependência tem que está na direção do *dependee*, ou seja, a direção que está no BPMN.

De acordo com o conhecimento do analista, as atividades internas dos participantes, devem ser analisadas para organiza-las em tarefas decompostas ou não.

Os *softgoals* não são modelados em BPMN, mas devem ser inseridos modelo i\* gerado. Por meio da análise de qualidade e atributos das tarefas, o analista tem condições de identificar os *softgoals* do processo. Como exemplo: podemos adicionar como uma dependência, o *softgoal* ‘agilidade’, entre o médico e a companhia de seguros. O médico precisa da aprovação do tratamento pela companhia, e quanto mais rápido essa aprovação sair, mais rápido o tratamento é realizado.

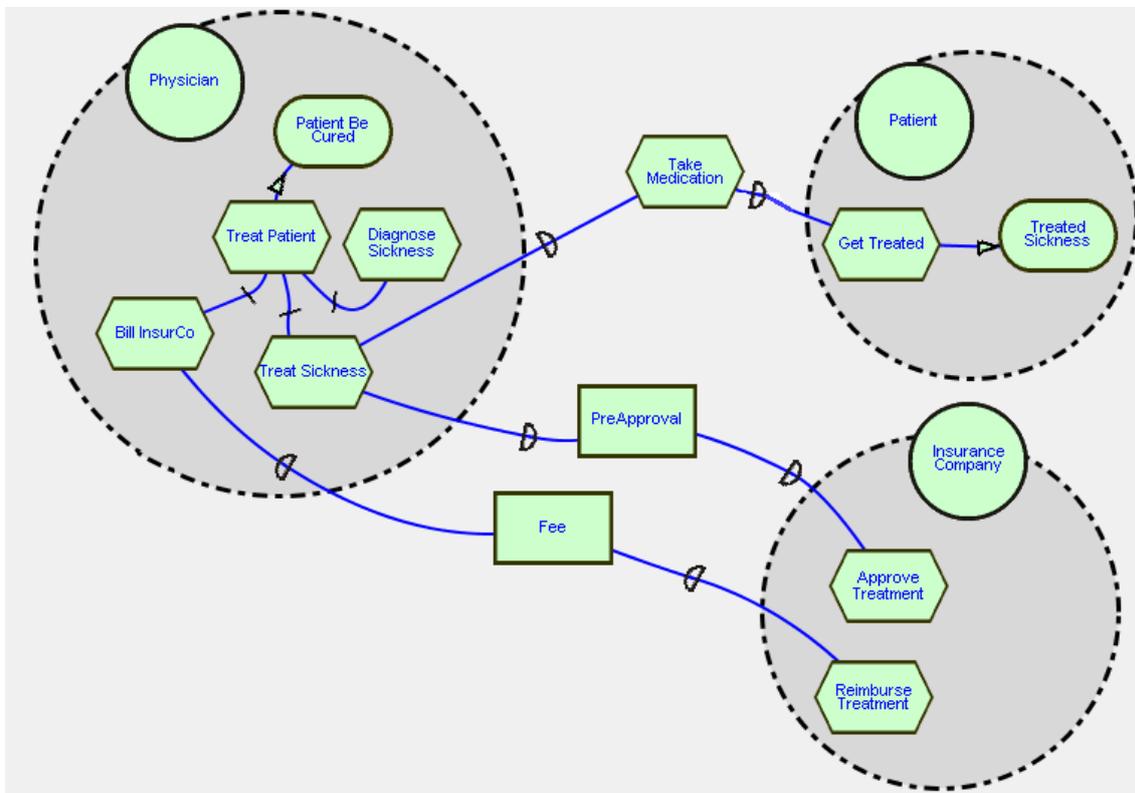


Figura 4.2 - Modelo i\* obtido do modelo BPMN da Fig. 4.1.

## 4.2 Avaliação do método proposto

Vamos usar os critérios usados por Decreus et al. (2009), apresentados no Capítulo 2, para avaliar a abordagem sugerida.

A primeira etapa da avaliação averigua a clareza do método. Assim, a primeira etapa é avaliar as variáveis das propriedades metodológicas da transformação entre os modelos, que

são: o nível de detalhe fornecido por passo, o nível de formalidade, os mapeamentos conceituais entre o  $i^*$  e o BPMN, e a consistência entre os modelos.

O segundo passo é avaliar as propriedades organizacionais do método proposto. As variáveis dessa etapa são: a atribuição de responsabilidade em cada passo do método, o grau em que os modelos de processo são orientados a negócio, o nível em que os modelos de processos são considerados técnicos, à medida que a estrutura organizacional é suportada e o nível de atenção dado a modelagem do contexto organizacional.

#### **4.2.1 Propriedades Metodológicas**

O nível de detalhes por passos é a variável que classifica o método de acordo com as informações que são fornecidas do início ao fim do processo da transformação. Nossa abordagem detalha sistematicamente os passos do mapeamento, justificando cada passo a ser seguido. Em relação ao método de Koliadis et al. (2006), nossa abordagem está melhor detalhada e justificada, pois descrevemos cada passo e sub-passo da transformação.

Descrevemos regras de mapeamento simples entre os modelos, que mantém a consistência, tanto no mapeamento  $i^*$ - BPMN, quanto no mapeamento BPMN –  $i^*$ . As regras de mapeamento mais formais garantem uma aplicação sistemática e uma possível transformação automatizada entre os modelos. O método de Koliadis et al. (2006) possui regras e etapas semiformais, pois propõe em alguns passos o ‘raciocínio intencional’. Por outro lado, no nosso método, o analista só não tem passos sistemáticos na modelagem das atividades internas (heurística iv).

Como Koliadis et al. (2006), também estabelecemos regras de verificação da consistência entre os modelos, porém nossas regras estão detalhadas e não deixam espaços para o julgamento do analista.

#### **4.2.2 Propriedades Organizacionais**

A avaliação organizacional é necessária para saber qual a função de cada passo do método. Nossa abordagem descreve em detalhes o que cada passo faz e qual seu papel no mapeamento. Já, Koliadis et al. (2006) não fornecem estes detalhes e apenas afirmam que o analista deve aplicar os passos.

A variável modelo de processos orientados a negócio avalia o grau em que os métodos são independentes de tecnologia. Tanto nossa abordagem como a abordagem de Koliadis et al. (2006) são independentes de tecnologia, porém incluem detalhes técnicos de rotinas.

A hierarquia dos relacionamentos entre os atores é respeitada tanto no nosso método quanto no método de Koliadis et al. (2006). A modelagem do contexto mantém os atores e os relacionamentos entre os atores, mas não organiza hierarquicamente os objetivos.

### **4.2.3 Resultado da avaliação**

Como podemos observar na descrição das propriedades, nossa abordagem possui mais detalhes, tanto nas heurísticas de mapeamento como na consistência entre os modelos.

Nas variáveis em que não empatamos com o método proposto por Koliadis et al. (2006), superamos a sua abordagem devido ao grau de detalhes e passos sistemáticos do processo de mapeamento aqui proposto.

Nas variáveis metodológicas, superamos o método de Koliadis et al. (2006) em níveis de detalhe e consistência entre os modelos, pois como nosso objetivo é propor um método mais sistemático, as heurísticas estão bem definidas e existe uma relação direta entre os elementos dos modelos mapeados.

Nas variáveis organizacionais, tivemos melhor resultado nas responsabilidades dos passos, pois, cada passo é especificado, e isso deixa evidente sua função no mapeamento. Na estrutura organizacional, nos destacamos um pouco, devido a manutenção das relações entre os atores, ou seja, as dependências e ligações entre os participantes são mantidas nos modelos, garantindo a integridade organizacional.

Nossa abordagem ainda possui etapas em que o julgamento do analista precisa ser usado, porém, os passos descritos garantem a integralidade do processo e a satisfação dos objetivos organizacionais. Em relação ao método de Koliadis et al. (2006), nossa abordagem está metodicamente definida e pouco suscetível a interpretações divergentes, porque estabelecemos regras que relacionam diretamente os elementos dos modelos i\* e BPMN. Ao contrário deles, que não possuem regras de mapeamento sistemáticas e deixam a critério do analista a maior parte da transformação.

A Tabela 4.1 resume nossa comparação entre os métodos. Com base na pontuação definida por Decreus et al. (2009), classificamos os dois métodos, como segue.

<i>Propriedades</i>	<i>Método Proposto</i>	<i>Método Koliadis et al. (2006)</i>
Nível de detalhe por passos	5	4
Formalidade do algoritmo	4	4
Mapeamento de conceitos	4	4
Consistência entre os modelos	5	4
Responsabilidade dos passos	5	3
Orientados a negócio	5	5
Orientados a tecnologia	2	2
Estrutura Organizacional	3	2
Modelagem do Contexto	4	4

**Tabela 4.1 - Comparação dos métodos.**

Nosso critério de pontuação é: 1 para o pior caso, 5 para o melhor caso. Como Decreus et al. (2009) usou cinco níveis de pontuação, seguimos esse mesmo padrão.

## Capítulo 5

# Conclusão

Nesse último capítulo são apresentadas as considerações finais sobre o trabalho apresentado, as contribuições proporcionadas pela pesquisa, além de possíveis estudos futuros. As limitações encontradas durante a realização do trabalho também são apresentadas nesse capítulo.

### 5.1 Contribuições e Limitações

A modelagem orientada a objetivos prioriza as metas que os *stakeholders* desejam alcançar, e é por isso, que essa abordagem é usada para modelar os objetivos organizacionais que estão relacionados aos processos de negócios de uma organização.

O *i\** traz a concepção social para dentro dos processos de negócio, enfatizando os conceitos sociais que são representados pelas tarefas rotineiras da organização. Por essa razão, que o trabalho conjunto da abordagem orientada a objetivos e dos modelos de processos de negócio garante uma estrutura organizacional que está alinhada aos objetivos da organização.

O uso conjunto desses dois modelos contribui para a difusão dos conceitos em áreas diferentes, ou seja, com a integração dos modelos, os analistas especializados no Framework *i\**, pode também, entender e modelar processos de negócio com o BPMN. Assim como, os analistas experientes na modelagem com a notação BPMN podem usar o *i\** para modelar os objetivos organizacionais.

Uma contribuição deste trabalho foi a aplicação do método de Koliadis et al. (2006) a um exemplo mais complexo do que o exemplo utilizado pelos autores do trabalho. Com isso, foi possível identificar as limitações nas heurísticas de mapeamento definidas no método.

Com base nestas limitações, o presente trabalho propôs melhorias no método de Koliadis et al. (2006) de forma a tornar o mapeamento bidirecional entre modelos *i\** e BPMN mais sistemático. O novo método foi aplicado ao mesmo exemplo utilizado para analisar a proposta de Koliadis et al. (2006). Como resultado, percebeu-se que o novo método apresenta menos limitações que o método original. Esta comparação entre os dois métodos foi realizada usando as mesmas variáveis de comparação consideradas no trabalho de Decreus et al. (2009).

Como limitação da abordagem proposta, afirmamos que ainda não foi possível definir heurísticas de mapeamento que sejam complementemente livres da interferência do analista. Por

exemplo, na obtenção de modelo BPMN a partir de modelos  $i^*$ , as heurísticas para obter *softgoals* e decomposição de tarefas ainda são completamente dependentes do analista.

Apesar desta limitação, pelo fato do novo método ser mais sistemático do que o método original, ele poderá ser usado mais facilmente por usuários iniciantes das notações  $i^*$  e/ou BPMN. Além disso, a obtenção sistemática de um modelo de processos de negócios a partir de um modelo orientado a objetivos permite a compreensão das atividades do processo dentro da organização e a consistência dessas atividades com os objetivos estratégicos da organização.

## **5.2 Trabalhos Futuros**

Considerando as limitações apresentadas no trabalho, os próximos estudos precisam ir em direção a um método sistemático que reduza ao máximo a inferência e a interferência humana durante o mapeamento dos modelos e, assim, possibilitando a geração automática de um modelo a partir do outro modelo.

A aplicação do novo método a exemplos mais complexos e estudos de caso reais também é necessária para possibilitar a identificações de limitações que direcionem a melhorias futuras no método.

## **5.3 Considerações Finais**

O desenvolvimento desse trabalho foi muito importante porque trouxe melhorias para o método de mapeamento bidirecional de modelos  $i^*$  e BPM. Apesar do trabalho ter obtido bons resultados, ele ainda apresenta oportunidades de melhorias como foi dito na seção de trabalhos futuros.

A realização desta pesquisa contribuiu consideravelmente para o desenvolvimento acadêmico e científico da autora, visto que durante a realização do trabalho foram aplicadas técnicas de pesquisa, como planejamento e metodologia científica, detecção de problemas, interpretação dos resultados e concretização de conclusões.

## Referências Bibliográficas

BLEISTEIN, S.; COX, K.; VERNER, J.; PHALP, K.: B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process, *Information and Software Technology*, vol. 48, pp. 846-868, 2006.

BOOCH, G., J. RUMBAUGH; I. JACOBSON. *The Unified Modeling Language User Guide*, Addison Wesley, Reading, MA. 1999.

CAILLIAU, A.; LAMSWEERDE, A.: A Probabilistic Framework for Goal-Oriented Risk Analysis Proc. RE'2012: 20th IEEE International Conference on Requirements Engineering, Chicago, September 2012.

DARDENNE, A.; LAMSWEERDE, A.; FICKAS, S.: Goal-directed requirements acquisition, *Science of Computer Programming*, v.20 n.1-2, p.3-50, April 1993.

DAVENPORT, T. H.: *Reengenharia de processos: como inovar na empresa através da tecnologia da informação*. Rio de Janeiro: Campus, 1994.

DECREUS, K., SNOECK, M.; POELS, G.: Practical Challenges for Methods Transforming i\* Goal Models into Business Process Models. In: *Seventeenth IEEE International Requirements Engineering Conference (RE'09)*, Atlanta, GA, USA. 2009.

KOLIADIS, G.; GHOSE, A. Relating Business Process Models to Goal-Oriented Requirements Models in KAOS. *Proceedings of the 9th Pacific Rim Knowledge Acquisition international conference on Advances in Knowledge Acquisition and Management*. 2006.

KOLIADIS, G.; VRANESEVIC, A.; BHUIYAN, M.; KRISHNA, A.; GHOSE, A.: Combining i\* and BPMN for Business Process Model Lifecycle Management. In: *Business Process Management Workshops*, 2006, pp. 416-427.

KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering: processes and techniques*. Chichester, John Wiley & Sons, 1998.

LAMSWEERDE, A.: *Goal-Oriented Requirements Engineering: A Guided Tour*. *Requirements Engineering, (RE'01)*. *Proceedings. Fifth IEEE International Symposium*. 2001.

MIERS, D.; STEPHEN, W. *BPMN Modeling and Reference Guide*. Future Strategies Inc. 2008.

OWEN, M.; RAJ, J.: BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard, 2003. Disponível em: <[www.bpmn.org/](http://www.bpmn.org/)>. Acesso em: 2 fev. 2013.

PAIM, R. As tarefas para gestão de processos. Rio de Janeiro: Tese de Doutorado em Engenharia de Produção, UFRJ, 2007.

PAULA FILHO, W. P. Engenharia de Software. Rio de Janeiro: LTC, 2001.

WHITE, S. A. Introduction to BPMN. IBM Corporation, 2004. Disponível em: <[www.bptrends.com](http://www.bptrends.com)>. Acesso em: 15 de jan. 2013.

YU, E.; MYLOPOULOS, J.: From E-R to A-R - Modelling Strategic Actor Relationships for Business Process Reengineering. Proceedings of the 13th International Conference on the Entity-Relationship Approach (ER'94) Lecture Notes in Computer Science no. 881, Springer-Verlag. pp. 548-565 1994.

YU, E.: Modeling Strategic Relationships for Process Reengineering. Ph.D. Thesis, Graduate Dept. of Comp. Science, University of Toronto. 1995.

YU, E.: Social Modeling and i\*. Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos, Springer-Verlag, Berlin, Heidelberg, 2009.

YU, E.: Towards modelling and reasoning support for early-phase requirements engineering. Requirements Engineering, Proceedings of the Third IEEE International Symposium on, 1997.