2
Marco Teórico

Este capítulo apresenta análises sobre as principais linguagens de modelagem atuais e demonstra as comparações e características consideradas na escolha das linguagens selecionadas para integração.

2.1. Introdução

Diversas linguagens estão disponíveis para a modelagem de processos de negócio e para a modelagem de objetivos, no entanto, a maioria das linguagens são específicas dentro de seu contexto de aplicação, não apresentando meios de relacionar seus elementos. [List&Korherr06] lista um conjunto de linguagens de modelagem de processos de negócio e apresenta uma comparação de suas características. Conforme demonstra a Figura 3, nenhuma das linguagens apresenta a partir do contexto de sua perspectiva a visibilidade para os objetivos dos processos modelados. Isso demonstra uma forte tendência do mercado no desenvolvimento de soluções para a modelagem da visão operacional, sem destaque ao relacionamento dos processos e objetivos organizacionais, em uma visão estratégica.

BPML Element	AD		BPDM		BPMN		EPC		IDEF3		Petri Nets		RAD	
Business Process Context Perspective														
Business Process	-/+	Activity	-/+	Stereotype SubProcess	-/+	Sub Process	-/+	Complex Function	-/+	Unit of Behaviour	-/+	Transition Hierarchy	-/+	Activity
Core, Support, Management	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Customer	-/+	Activity Partition	-/+	Role/ Participant	-/+	Pool	-/+	Organi-sational Role	-/-		-/-		-/+	Role
Deliverable	-/+	Object N.	-/-		-/-		+/+	Input/Output	-/+	Object	-/-		-/+	Resource
Service	-/+	Object N.	-/-		-/-		-/+	Input/Output	-/+	Object	-/-		-/+	Resource
Product	-/+	Object N.	-/-		-/-		-/+	Input/Output	-/+	Object	-/-		-/+	Resource
Process Owner	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Goal	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Process	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Enterprise	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Measure	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Quantitative	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Qualitative	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
To Be Value	-/-		-/-		-/-		-/-		-/-		-/-		-/-	
Unit	-/-		-/-		-/-		-/-		-/-		-/-		-/-	

Legend: +/+ Notation available / possible to present
-/+ Notation not available / possible to present
-/- Notation not available / not possible to present

Figura 3 – Avaliação da perspectiva no contexto de processos de negócio [List&Korherr06]

Ainda em uma tabela semelhante, [Pourshahid09] apresenta uma comparação de um conjunto de notações para modelagem de processos e objetivos (Figura 4 -

apresenta três sinais representando três níveis, sendo o sinal de visto representando "disponível", o sinal x representando "indisponível", e o sinal +/- representando um estado "intermediário, incompleto"). O trabalho demonstra a ausência de rastreabilidade entre processos e objetivos na grande maioria das linguagens. Esse é um forte indício da ausência do alinhamento entre os modelos nas principais linguagens utilizadas.

	BPMN	UML	EPC	YAWL	IDEF3	*i**	NFR	EEML	URN
Sequence Flow	✓	✓	✓	✓	✓	+/−	+/−	✓	✓
Roles	✓	✓	✓	✓	✗	✓	✗	✓	✓
Activities	✓	✓	✓	✓	✓	✗	✗	✓	✓
Events	✓	✓	✓	✓	✗	✗	✗	✓	✓
Process Hierarchies	✓	✓	✓	✓	✓	✗	✗	✓	✓
Goal modeling	✗	✗	✗	✗	✗	✓	✓	+/−	✓
Goal model Evaluation	✗	✗	✗	✗	✗	✗	✓	✗	✓
Goal/Process Traceability	✗	✗	✗	✗	✗	✗	✗	✗	✓

Figura 4 – Comparação de notações e características de modelagem de processos [Pourshahid09]

Neste trabalho integramos apenas duas linguagens, sendo uma voltada para a modelagem de processos e outra para a modelagem de objetivos. O objetivo é cobrir lacunas existentes em outras linguagens e possibilitar a análise do processo em relação aos seus objetivos através do uso de indicadores e, assim, conseguir avaliar o alinhamento entre os modelos.

Segundo [Pourshahid09], para apoiar o alinhamento de processos de negócio com objetivos de processos de negócio, a notação de modelagem de processo deve oferecer a modelagem de processo, modelagem de objetivo, rastreabilidade entre os modelos de objetivo e processo, e mecanismos de avaliação do modelo de objetivos. Caso contrário, não seria capaz de demonstrar o impacto dos processos de metas organizacionais. Além disso, papéis coadjuvantes (ou atores / organizações), atividades (ou funções) e eventos irão aumentar as capacidades de modelagem de processos de negócios, e enriquecer o significado das relações entre processos de negócio e objetivos.

Em uma análise preliminar, identificamos a partir da tabela de [Pourshahid09] (Figura 4) a ausência da modelagem de objetivos na maioria das linguagens de modelagem de processo apresentadas, sendo ainda possível verificar que o inverso também é verdade, ou seja, as linguagens de modelagem de objetivo não oferecem recursos para modelagem de processo ou não oferecem a rastreabilidade entre os modelos (exceto pela linguagem URN, conforme os autores apontam). No entanto, na arquitetura organizacional, processos de negócio e objetivos são intrinsecamente

interdependentes, apesar das linguagens de modelagem atuais apresentarem deficiências no alinhamento entre processos e objetivos.

Além do problema de alinhamento entre os modelos, observa-se grande ênfase no desenvolvimento de linguagens que ofereçam suporte para a modelagem da visão operacional, uma vez que são poucas as linguagens de modelagem de objetivos em comparação com as linguagens de modelagem de processos. É possível que este fato seja resultado da antiga visão de *workflow* [WMC95], aplicada de forma tradicional ao longo dos anos, ou ainda, pelo excessivo número de ferramentas que por muito tempo ofereceram (e algumas ainda oferecem) somente a modelagem de processos, por exemplo, Arpo BPMN ++, *Oryx*, Bizagi, Intalio|BPMS, Signavio, Microsoft Visio, Visual Architect e ARIS [Arpo12], [*Oryx*12], [Bizagi12], [Intalio12], [Signavio12], [Visio12], [VisualArchitect12], [ARIS12]. Assim, o uso do ferramental disponível também dificulta sobremaneira a tarefa de identificar se os processos utilizados para gerar serviços e produtos, verdadeiramente atingem os objetivos da organização, nem o impacto que as mudanças nos objetivos causariam nos processos de negócio [Cardoso10], [Cardoso11].

Para identificar as melhores características das linguagens e aplicar o reuso de seus elementos para o desenvolvimento de uma nova linguagem integrada, as seguintes linguagens foram analisadas: UCM e GRL (que compõem a URN), EPC e a modelagem de objetivos do *framework* ARIS, UML, BPMN, i* e NFR. As próximas subseções apresentam cada uma das linguagens e ao final a conclusão.

2.2. UML

A UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos [Guedes07]. É um padrão adotado pela OMG utilizado a nível internacional no contexto da engenharia de software e atualmente encontra-se na versão 2.0 [OMG11a].

Esta versão é composta por 13 tipos de diagramas [Melo04], com o objetivo de fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando-se assim atingir a completitude da modelagem, permitindo que cada diagrama complemente os outros. Alguns diagramas focam o sistema de forma mais geral, apresentando uma visão externa do sistema, como é o objetivo do Diagrama de Casos de Uso, ao passo que outros oferecem uma visão de uma camada mais profunda do software, apresentando um enfoque mais técnico ou

ainda visualizando apenas uma característica específica do sistema ou um determinado processo [Gudes07].

No entanto, o metamodelo da UML não contempla elementos específicos para tratar com diagramas de processos de negócio [Villarroel06], [Mili03]. Os usuários da UML utilizam o diagrama que mais se aproxima a este propósito que é o diagrama de atividades [Pourshahid09]. O Diagrama de Atividade se preocupa em descrever os passos a serem percorridos para a conclusão de uma atividade específica, muitas vezes representada por um método com certo grau de complexidade, podendo, no entanto, modelar um processo completo [Guedes07].

No entanto, muitos estudiosos concordam que existe a falta de vocabulários para expressar processos de negócio de uma forma intuitiva e natural na UML [Mili03]. Além disso, os diagramas de atividade oferecem a modelagem de fluxo de sequência, papeis, atividades, eventos e hierarquias de processos, mas não oferecem modelagem de objetivos e rastreabilidade entre os modelos de objetivos e modelos de processos [List&Korherr06]. Além disso, certos conceitos frequentemente usados na modelagem de processos não fazem parte do padrão UML [Eriksson&Penker00].

Para atender à demanda de modelagem de processos de negócio com uma solução mais abrangente, os mecanismos de extensão da própria UML definidos pela OMG foram utilizados (também conhecido como *built-in extensions*) [Villarroel06], [Eriksson&Penker00].

[Eriksson&Penker00] propuseram um conjunto de extensões que agregados ao padrão UML permitem a modelagem de processos de negócio. Estas extensões não alteram o padrão UML criando uma nova linguagem, mas apenas cria extensões baseadas nos mecanismos de extensões da UML.

Dentro de todos os diagramas e elementos propostos por [Eriksson&Penker00] (*Vision Statement diagram, Conceptual model, Goal model, Process diagram, Assembly Line diagram, Use-Case diagram, Resource model, Organization model, Information model, Statechart diagram, Interaction diagrams e System Topology diagram*), neste trabalho apenas interessa os diagramas de processos (*Process diagram)* e de objetivos (*Goal model)*, e os mecanismos de interação entre eles.

O diagrama de processos, diagrama de objetivos, seus elementos e apresentação de exemplos são descritos a seguir.

2.2.1.Extensões do diagrama de processo

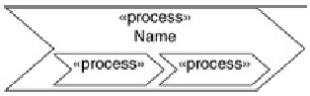
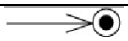
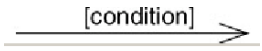
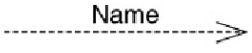
O diagrama de processo proposto por [Eriksson&Penker00] é baseado no diagrama de atividades da UML, somado a um conjunto de estereótipos que

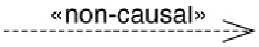
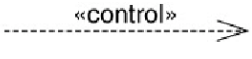
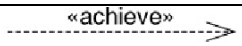
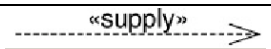
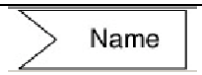
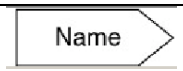
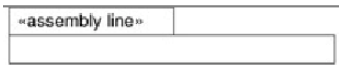
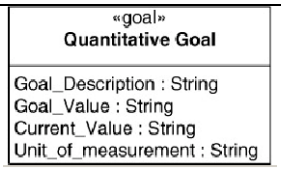
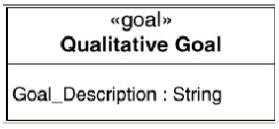
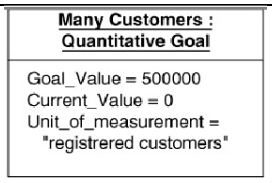
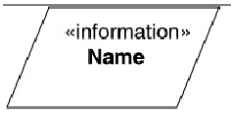
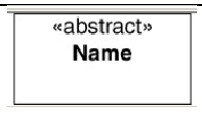
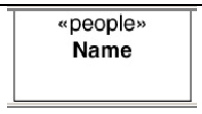
descrevem a execução das atividades dentro de um processo; como elas interagem; os objetos de entrada e saída; os recursos de suprimento e controle que participam no processo; e o objetivo do processo.

O processo é representado por um objeto de atividade com estereótipo de <<*process*>>, formado pelo ícone tradicional conforme descrito na Tabela 1 (além da regra de negócio, apresentada na Tabela 3, que está disponível em todos os diagramas). Um processo pode conter outros processos (subprocessos). A atividade é representada por um retângulo com os lados arredondados, e também é chamada na linguagem de *processo atômico,* ou seja, um processo que não pode ser decomposto.

Todos os elementos que estão envolvidos com o processo são modelados ao seu redor. O objetivo ou *problema* alocado ao processo é modelado acima do diagrama de processo, relacionado através do *link* de dependência que contém o estereótipo <<*achieve*>>, partindo do processo para o objetivo. Os outros objetos são típicos da modelagem de processos de negócio e encontram-se detalhados na Tabela 1.

Tabela 1 - Elementos de um diagrama de processo de negócio em UML

Extensões de processo			
Nome	Estereótipo	Símbolo	Definição/Descrição
Processo	Atividade		Um processo é uma descrição de um conjunto de atividades relacionadas que, quando executadas corretamente, irão satisfazer um objetivo explícito.
Atividade (Processo atômico)	Atividade		Um processo deve ser dividido em mais processos. Se estes processos são atômicos, eles são chamados de atividades.
Inicio do processo	Inicio		Inicia um processo
Fim do processo	Fim		Finaliza um processo
Objeto-para-Assembly Line	Objeto		Um objeto entregue por um processo para a *Assembly Line*.
Objeto-vindo de-Assembly Line	Objeto		Um objeto que vai da *Assembly Line* para um processo.
Fluxo de processo	Fluxo de controle		Um fluxo de controle de processo com uma condição
Fluxo de recurso	Fluxo de objeto		Fluxo de objeto mostra que um objeto é produzido por um processo e consumido por outro.

Fluxo de recursos não-causal	Fluxo de objeto		Fluxo de objeto não-causal mostra que um objeto pode ser produzido por um processo e consumido por outro.
Controle de processo	Fluxo de objeto		Mostra que um processo é controlado por um objeto.
Conexão de objetivo	Dependência		Aloca um objetivo a um processo.
Processo de suprimento	Fluxo de objeto		Mostra que um processo é suprido por um objeto.
Processo de decisão	Decisão		Ponto de decisão entre dois ou mais processos.
Processos de união e bifurcação	Bifurcação e União		Bifurcação e união de processos.
Receber eventos de negócio	Recepção de sinal		Mostra um recebimento de evento de negócio.
Enviar eventos de negócio	Emissão de sinal		Mostra um envio de evento de negócio.
Assembly Line	Pacote		As *Assembly Lines* sincronizam e suprem os processos em termos de objetos.
Objetivo quantitativo	Objetivo		Um objetivo pode ser descrito com um valor alvo em uma unidade especifica de medida (a quantidade).
Objetivo qualitativo	Objetivo		Um objetivo normalmente descreve em linguagem natural. Um objetivo qualitativo envolve julgamento humano, no processo de determinar se ele foi cumprido.
Instância de um objetivo qualitativo	Objetivo qualitativo		Ambos os objetivos qualitativos e quantitativos podem ser instanciados.
Recurso	Classe		Recursos podem ser produzidos, consumidos, usados ou refinados por processos. Os recursos são também informação ou coisas que podem ser abstratas ou físicas.
Recurso abstrato	Classe		Um recurso abstrato é algo intangível, por exemplo, conceitos matemáticos.
Pessoa	Classe		Um recurso físico que específica um ser humano.

Recurso físico	Classe	«physical» **Name**	Um recurso físico que especifica documentos, máquinas (não humano).
Evento de negócio	Sinal	«business event» **Name**	Uma ocorrência significante no tempo ou espaço. Um evento de negócio causa algum impacto ao negócio.

A Figura 5 apresenta um exemplo de modelo de processo de negócio dividido em raias (*swimlanes*), padrão do diagrama de atividades. Cada raia demonstra as atividades executadas pelos respectivos responsáveis descrito no topo. As trocas de recursos (entradas e saídas) são explicitadas entre os processos.

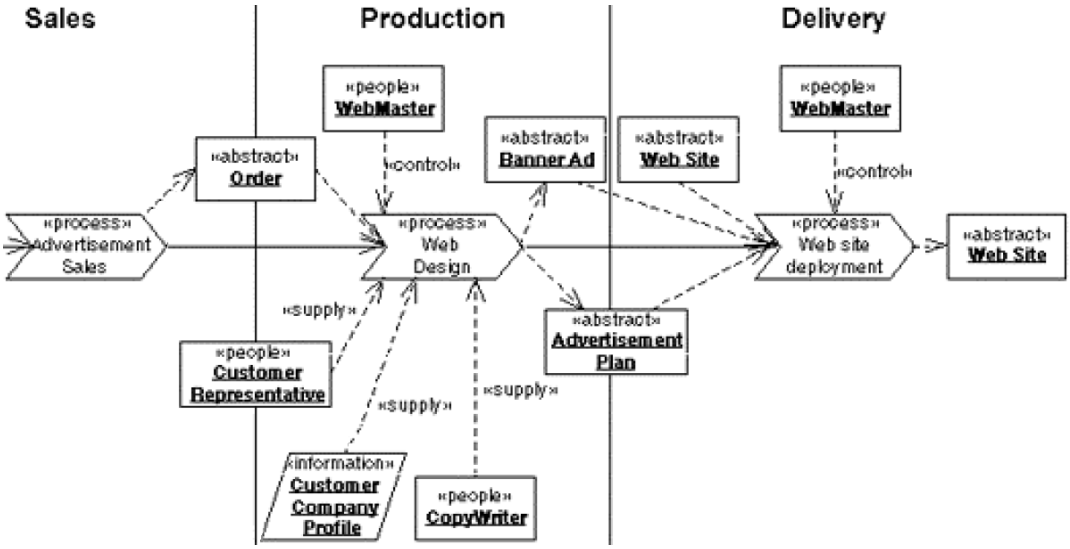


Figura 5 - Diagrama de processo de negócio em UML [Eriksson&Penker00]

A Figura 6 apresenta um exemplo de modelo de processo de negócio mais simples, porém demonstra o relacionamento de um objetivo ao elemento processo.

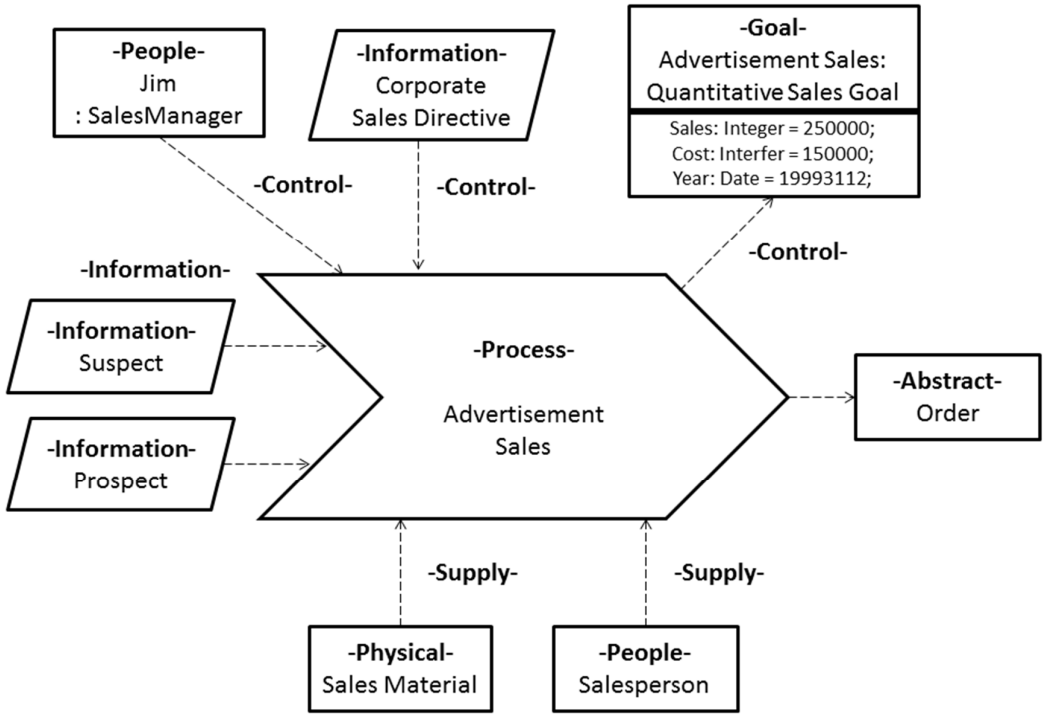


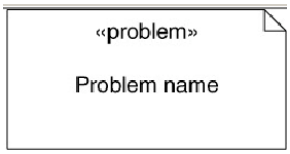
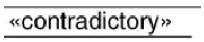
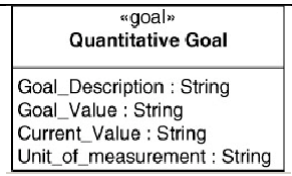
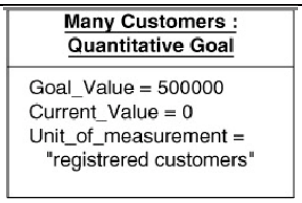
Figura 6 – Exemplo de objetivo relacionado ao processo [Eriksson&Penker00]

2.2.2.Extensões do diagrama de objetivos

O diagrama de objetivos proposto por [Eriksson&Penker00] reutiliza o diagrama de objetos da UML. Os objetivos podem ser representados em diferentes níveis, sendo cada nível um diagrama de objetos. Os objetivos são descritos como classes de objetos com o estereótipo de <<*goal*>>. Os elementos que compõem o diagrama são os objetivos, relacionamento de dependência e associação entre objetivos, e os problemas os quais o objetivo resolve (Tabela 2, além dos elementos da Tabela 3 que estão disponíveis para todos os diagramas).

Tabela 2 – Elementos de um diagrama de objetivos em UML

Extensões de objetivo			
Nome	Estereótipo	Símbolo	Definição/Descrição
Objetivo	Classe	«information» **Name**	Denota os estados de desejo, o que significa que os objetivos motivam ações para alterar estados na direção desejada.

Problema	Nota		Algo que impede o alcance do objetivo. Cauda, medida e pré-requisito são outros estereótipos que são uteis quando se modela problemas. Uma causa leva aos problemas. Um problema pode ser resolvido se uma causa é removida. A causa pode ser removida se uma medida é alcançada e certos pré-requisitos são validos.
Dependência de objetivo	Dependência		Objetivos são organizados em hierarquias de dependência nas quais um ou alguns objetivos são dependentes de subobjetivos.
Objetivo contraditório	Associação		Objetivos podem ser contraditórios, mas devem ser cumpridas.
Decomposição de objetivo incompleta	Dependência		Objetivos são organizados em hierarquias de dependência que algumas vezes são incompletas.
Decomposição de objetivo completa	Dependência		Objetivos são organizados em hierarquias de dependência que algumas vezes são completas.
Objetivo quantitativo	Objetivo		Um objetivo pode ser descrito com um valor alvo em uma unidade especifica de medida (a quantidade).
Objetivo qualitativo	Objetivo		Um objetivo normalmente descreve em linguagem natural. Um objetivo qualitativo envolve julgamento humano, no processo de determinar se ele foi cumprido.
Instância de um objetivo qualitativo	Objetivo qualitativo		Ambos os objetivos qualitativos e quantitativos podem ser instanciados.

Existem duas classes de objetivos predefinidas no diagrama de objetivos: objetivo quantitativo e objetivo qualitativo. Um objetivo quantitativo pode ser medido através de valores específicos que são atingidos, um objetivo qualitativo depende do julgamento de quem avalia, tornando subjetiva a sua medição. O objetivo quantitativo

é composto pelos atributos descrição (*goal description*), valor esperado (*goal value*), valor atual (*current value*) e unidade de medida (*unit of measurement*). O objetivo qualitativo também possui descrição, porém o valor esperado, valor atual e unidade de medida ficam a cargo do avaliador que os autores separam como "instância de um objeto qualitativo".

O relacionamento de dependência entre objetivos denota que um objetivo é subobjetivo de outro, ou que um depende do outro. O relacionamento de associação denota *links* entre objetivos, tais como contradições [Eriksson&Penker00].

Outro conceito apresentado é o "*problema*", que representa um obstáculo para a satisfação do objetivo. Identificar *problemas* pode levar ao descobrimento de novos objetivos que são satisfeitos com a eliminação dos *problemas*. A modelagem de *problemas* pode ser hierárquica, com a subdivisão em subproblemas. Os *problemas* são relacionados com os seus respectivos objetivos, mas podem ser eliminados pela inclusão de ações de solução.

Um plano de ação pode ser projetado no modelo de objetivos como uma forma de resolver os *problemas*. Os objetivos, por sua vez, são relacionados aos processos de negócio. Os processos projetados no plano de ação devem ser posteriormente formalizados como processos e relacionados aos seus objetivos.

A Figura 7 apresenta um exemplo de modelo de objetivo. O objetivo geral é ter muitos consumidores, e o valor desejado é de 500.000. Este objetivo foi dividido em 3 subobjetivos que também descrevem as categorias dos consumidores (visitantes da internet desconhecidos, consumidores registrados, consumidores inscritos. Três problemas estão relacionados a cada um dos subobjetivos. Por exemplo, o problema relacionado ao objetivo de atrair mais visitantes na internet é que os usuários podem desconhecer o site. Para solucionar este problema, são gerados novos subobjetivos que almejam a publicação do site para o conhecimento dos usuários, por exemplo, "Garantir que o site esteja visível mas máquinas de busca mais comuns".

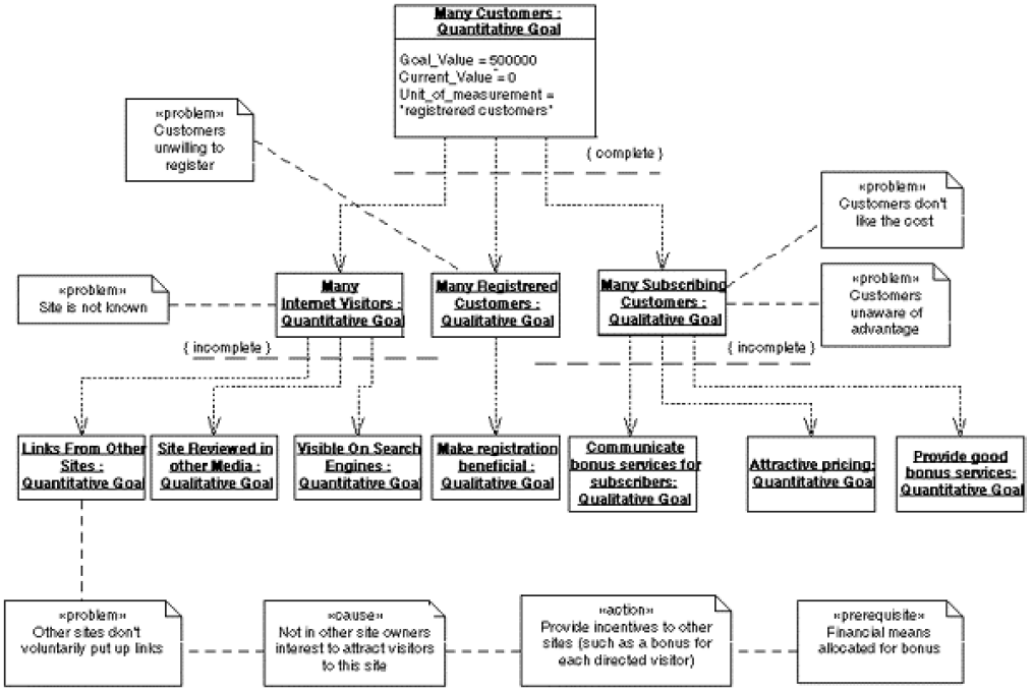


Figura 7 – Diagrama de objetivos em UML [Eriksson&Penker00]

A Tabela 3 apresenta o elemento regra de negócio que se apresenta em todos os modelos. As regras são elementos especiais que definem restrições/condições ao negócio, e por serem importantes devem sempre ser ligados aos elementos que são relacionados e/ou sofrem tais regras, por exemplo, objetivos e processos.

Tabela 3 – Recursos e regras compõem todos os diagramas

Extensões de regras			
Nome	Estereótipo	Símbolo	Definição/Descrição
Regra de negócio	Nota	«business rule» Rule statement	Regras restringem, derivam e estabelecem condições de existência. As regras de negócio são usadas para especificar estados de desejo, incluindo os estados alvos permitidos ao negócio.

Esta seção apresentou a modelagem de processos de negócio e objetivos utilizando um *built-in* para a linguagem UML, proposta por [Eriksson&Penker00]. A extensão apresentada utiliza modelos UML e objetos com novos estereótipos para representação de novos conceitos, o que permite a fácil inclusão de novos elementos a partir do reuso dos objetos e diagramas UML.

Na próxima seção é apresentada a notação BPMN.

2.3. BPMN

BPMN (*Business Process Model and Notation*), versão 2.0, é um padrão desenvolvido pela OMG (*Object Management Group*) com o principal objetivo de oferecer uma notação que fosse legível e compreensível para os usuários do negócio [OMG11a].

A notação foi desenvolvida a partir do conhecimento e experiências dos membros da OMG que procuraram consolidar as melhores ideias para alcançar uma única notação padrão. Muitas metodologias e notações foram revistas durante o desenvolvimento da BPMN (por exemplo, UML *Activity Diagram*, UML EDOC *Business Process*, IDEF, ebCML BPSS, *Activity-Decision Flow* (ADF) *Diagram*, RosettaNet, LOVeM e *Event-Process Chains* (EPCs)) para que fosse desenvolvido uma especificação que representasse a combinação das melhores práticas da comunidade de modelagem de processos [OMG11a].

A intenção da BPMN é padronizar o modelo de processo de negócio e sua notação em face das diversas notações de modelagem e pontos de vista distintos. Outro fator padronizado dentro da BPMN é a definição de arquivos de processo BPMN, o que permite a exportação e importação dos modelos para ferramentas de diferentes fabricantes.

A BPMN suporta somente conceitos de modelagem que são aplicáveis a processos de negócio. Isso significa que outros tipos de modelagem realizados por organizações, mesmo com propósitos do negócio, estão fora do escopo da BPMN, tais como modelagem organizacional, modelagem de dados, modelagem de objetivos/estratégia e modelagem de regras de negócio.

A notação oferece uma variedade de recursos para o desenvolvimento de um modelo de processo de negócio e classifica os diagramas BPMN em três tipos básicos baseado na perspectiva na qual o processo é modelado: Processo de negócio privado não executável; Processo de negócio privado executável e Processo público.

Ainda existem os diagramas de Colaboração, Coreografia e Conversação, sendo que esses dois últimos não são detalhados aqui por fugirem ao objetivo de descrição do processo em si.

2.3.1. Processos privados

Os *Processos de negócio privados* são processos internos a uma organização, antigamente chamados de *workflow ou processos BPM*. Existem dois tipos de processos privados: executável e não executável. Um processo *executável* é um

processo que foi modelado com propósito de ser executado, utilizando em sua semântica processos, atividades, eventos, operadores lógicos, fluxos, *loops* e manipulação de dados. Um processo *não executável* é um processo que foi modelado com o propósito de documentar o comportamento do processo a um nível de detalhe de modelagem definido. Desta forma, a informação necessária para execução, tal como a condição formal das expressões são tipicamente excluídas em um processo não executável.

A Figura 8 mostra um exemplo de um modelo de processo de negócio privado.

Figura 8 – Exemplo de modelo de processo de negócio privado [OMG11a].

2.3.2.Processos públicos

Um processo de negócio público representa as interações entre um processo de negócio privado com outro processo ou participante (Figura 9). Somente o processo público torna-se visível, enquanto o processo privado é representado por uma raia vazia. Assim, o processo público mostra o lado de fora do mundo em que o fluxo de mensagens e o comando destas mensagens de fluxo são necessários para interagir com o processo.

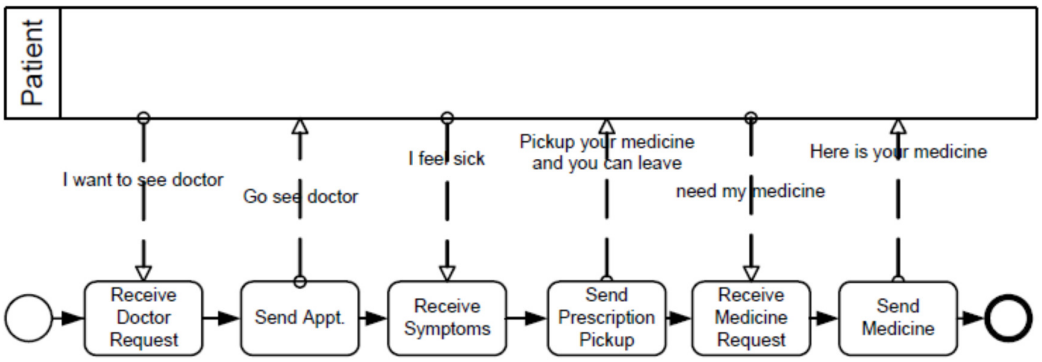


Figura 9 – Exemplo de processo de negócio público [OMG11a].

2.3.3.Processos colaborativos

Uma colaboração ilustra as interações entre duas ou mais entidades do negócio. Uma colaboração normalmente contém duas ou mais "*pools*", representando os participantes na colaboração. A troca de mensagem entre os participantes é mostrada pelo fluxo de mensagem que conecta as "*pools*" (ou os objetos dentro das *"pools"*). As mensagens associadas com o fluxo de mensagens estão descritas no corpo das setas

de fluxo de mensagem. A colaboração pode ser mostrada como dois ou mais processos públicos se comunicando (Figura 10).

Com um processo público, as atividades de colaboração dos participantes podem ser consideradas o “ponto de toque” entre os participantes. O correspondente interno no processo (executável) é suscetível a ter muito mais atividades e detalhes do que é mostrado nos processos públicos. Coreografias podem ser mostradas "entre" as “*pools*” como se elas dividissem o fluxo de mensagem entre elas. Todas as combinações de pools, processos, e coreografia são permitidas em uma colaboração.

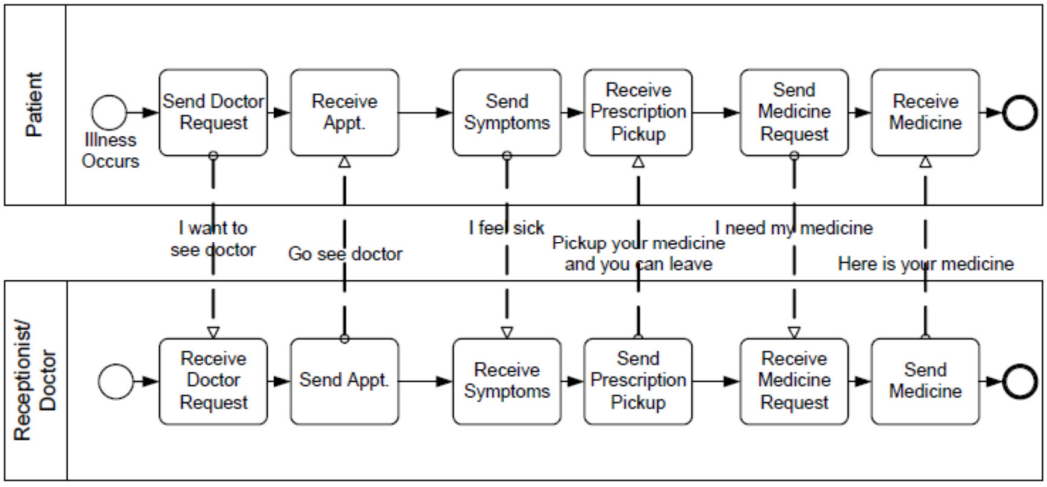
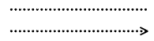
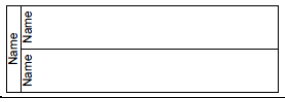


Figura 10 – Exemplo de processo colaborativo

2.3.4. Notação

Os elementos gráficos utilizados nos diagramas BPMN foram desenvolvidos intencionalmente para facilitar o entendimento dos modelos de forma fácil e ao mesmo tempo permitir expressar as complexidades inerentes aos processos de negócio. A abordagem adotada para lidar com esses dois requisitos conflitantes foi organizar aspectos gráficos da notação em categorias especificas de forma que o usuário possa facilmente reconhecer os tipos básicos de elementos e compreender o diagrama. Dentro das categorias básicas dos elementos, podem ser adicionadas variações e informações para suportar novas necessidades, porém sem mudar dramaticamente o visual básico e percepção do diagrama. As cinco categorias básicas de elementos são: Fluxo de objetos; Dados; Objetos de conexão; *Swimlanes* e Artefatos. Os elementos básicos que se encaixam nesses grupos são descritos na Tabela 4. Ainda são oferecidos outros elementos que são variações dos elementos básicos para expressar casos específicos com maior nível de detalhe (não demonstrados aqui).

Tabela 4 – Elementos básicos da BPMN

Nome	Símbolo	Definição/Descrição
Evento		Eventos são fatos que ocorrem durante a execução do processo. Pode ser classificado em evento inicial, intermediário e final.
Atividade		Uma atividade pode ser atômica ou não atômica (composta).
Operador lógico		Um operador lógico é usado para controlar divergência e convergência nos fluxos sequenciais de um processo.
Fluxo de sequencia		Um fluxo de sequencia é usado para mostrar a ordem das atividades que serão executadas no processo.
Fluxo de mensagem		Um fluxo de mensagem é usado para mostrar a passagem de mensagens entre os participantes.
Associação		Uma associação é usada para ligar informação e artefatos de forma gráfica.
Pool		Representação gráfica de um participante em um modelo.
Lane		Subpartição dentro de um processo ou Pool que estende toda a extensão do processo.
Objeto de dados		Objetos de dados que proveem informação sobre o que as atividades necessitam para serem executadas ou o que elas produzem.
Mensagem		Representa uma mensagem com conteúdo de comunicação entre dois participantes.
Grupo		Agrupa elementos gráficos de uma mesma categoria.

A Figura 11 apresenta um exemplo de modelo de processo de negócio em um diagrama BPMN. O processo é composto por quatro raias que representam os papéis executores das atividades que se encontram em suas respectivas raias. O modelo utiliza apenas elementos simples, tais como eventos, atividades, operadores lógicos e artefatos.

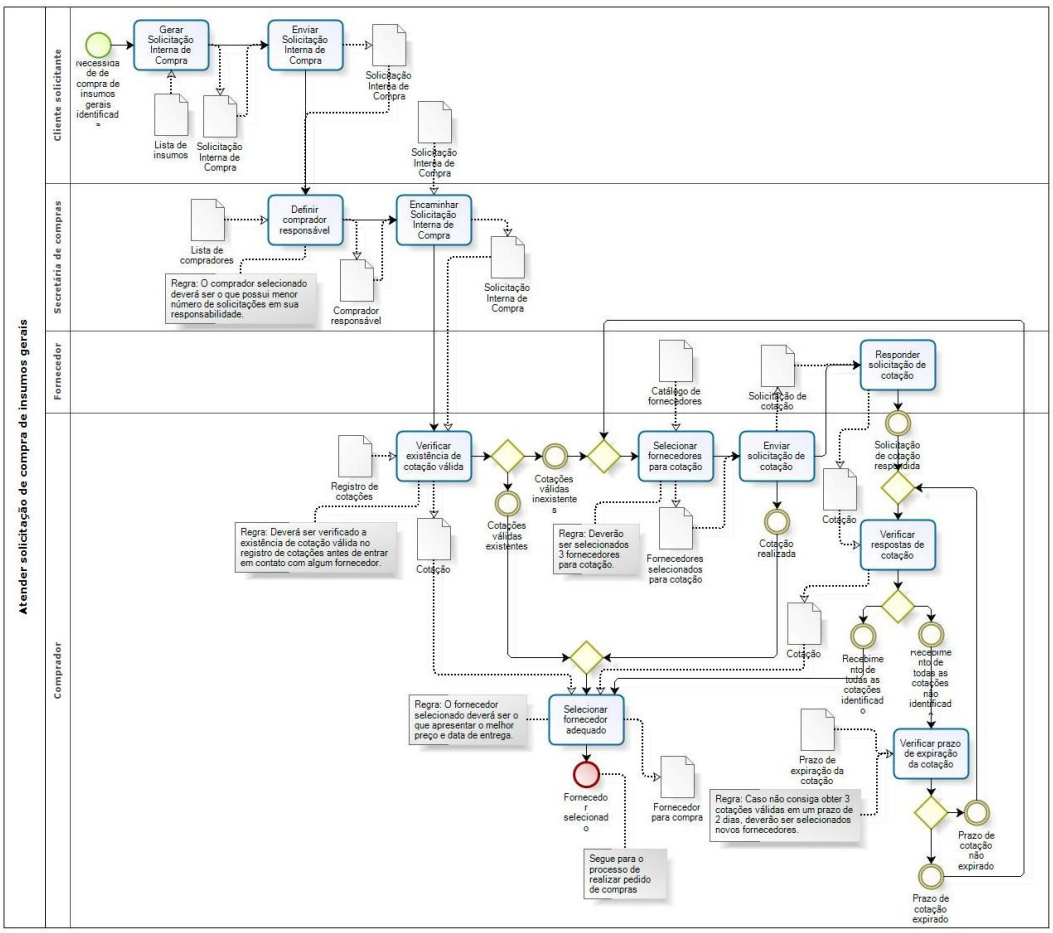


Figura 11 – Exemplo de modelo de processo de negócio em um diagrama BPMN

Esta seção apresentou a notação BPMN. Atualmente em sua versão 2.0, é um padrão publicado pela OMG que consiste em um resultado do esforço de diversos pesquisadores da área para desenvolver uma notação específica para a modelagem de processos de negócio. Portanto, a BPMN não aborda outros conceitos, como por exemplo, a modelagem de objetivos.

Na próxima seção será apresentado o *framework* i*, específico para a modelagem de metas intencionais.

2.4. *Framework* i*

O *framework* de Modelagem i* (i-estrela) [Yu95] modela contextos organizacionais com base nos relacionamentos de dependência entre os atores participantes [Napolitano09]. A ideia central do i* é representar através de modelos os atores e as dependências que eles têm uns com os outros para que metas próprias sejam atingidas [Oliveira08c].

O conceito mais relevante no i* é o conceito de objetivo: “Um objetivo é uma condição ou estado de interesse no mundo que um ator gostaria de alcançar” [Tradução livre, [Yu95] e [13] apud [Oliveira08c]]. Dessa forma, atores dependem uns dos outros para que seus objetivos sejam alcançados, recursos sejam fornecidos, tarefas sejam realizadas e metas-flexíveis sejam “razoavelmente satisfeitos” [Oliveira08c].

A proposta de modelagem do *framework i** consiste de dois componentes de modelagem, chamado *Strategic Dependency (SD)* e *Strategic Rationale (SR).*

O modelo SD descreve um processo em termos dos relacionamentos de dependência intencional entre os agentes. Os agentes dependem um do outro para que os objetivos sejam alcançados, tarefas sejam executadas e recursos compartilhados. O modelo SR prove uma descrição intencional do processo em termos de seus elementos e das suas razões estratégicas internas dos atores [Yu95]. As subseções seguintes detalham os modelo SD e SR.

2.4.1. *Strategic Dependency* (Modelo SD)

No modelo SD, os atores possuem dependência entre si para alcançar objetivos, executar tarefas e fornecer recursos. Ao depender de outro ator, o ator dependente obtém vantagens das oportunidades que são disponibilizadas através dos atores que prestam o suporte [Yu09]. Ao mesmo tempo em que o dependente é beneficiado, ele torna-se vulnerável caso não sejam atendidas as suas expectativas. Essas dependências são consideradas estratégicas para os atores interessados, já que eles podem ser beneficiados ou prejudicados em seu bem-estar. Atores poderiam escolher que dependências ter, de acordo com seu julgamento em relação a potenciais ganhos e perdas [Yu09].

O modelo SD é um grafo onde os nós representam atores (agentes, posições, papéis), e cada dependência representa um relacionamento de cooperação entre dois atores, onde um ator chamado de *depender* depende de outro chamado de *dependee*. O elo da dependência, chamado de *dependum*, é o objeto da dependência que pode ser: um objetivo, uma meta-flexível, uma tarefa ou um recurso, e esse é sempre uma entidade física ou informacional. Conforme mencionado anteriormente, as relações de dependência mostram as vulnerabilidades do *depender*, pois o *dependee* pode falhar no cumprimento do acordo e, por isso, cada dependência tem um grau de importância (“*critical*”, “*committed*” ou “*open*”, em ordem decrescente de importância) para refletir sua relevância [Oliveira08c].

Dentro do diagrama SD, é possível expressar quatro tipos de dependências: dependência de meta; dependência de tarefa; dependência de recurso; e dependência de meta-flexível. A Figura 12, Figura 13, Figura 14 e Figura 15 ilustram os quatro tipos de relacionamento (nestes casos, o *dependee* encontra-se sempre à esquerda e o *depender* sempre à direita):

Figura 12 – Exemplo de relacionamento de dependência de meta

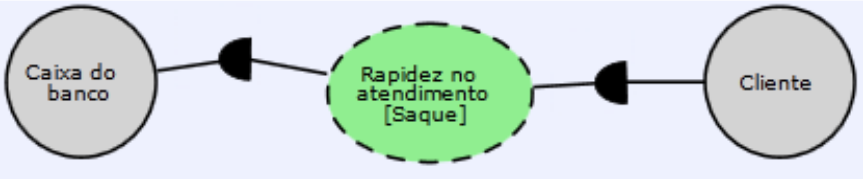


Figura 13 - Exemplo de relacionamento de dependência de meta-flexível

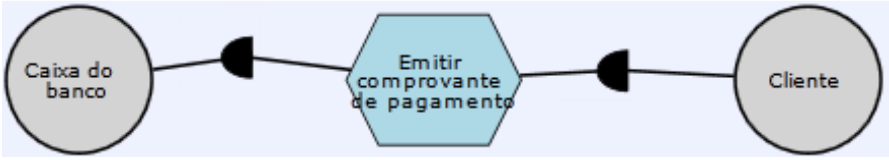


Figura 14 - Exemplo de relacionamento de dependência de tarefa

Figura 15 - Exemplo de relacionamento de dependência de recurso

Os exemplos fazem uma ilustração do significado de uma dependência entre dois atores: um ator “quer” alguma coisa que outro ator “pode” suprir. A Figura 12 apresenta uma dependência de objetivo, em que o cliente depende do caixa do banco para realizar o saque do dinheiro. A Figura 13 apresenta uma dependência de meta-flexível, em que o cliente depende do caixa do banco para ser atendido com rapidez ao realizar o saque. A Figura 14 apresenta uma dependência de tarefa, em que o cliente depende que o caixa do banco execute a tarefa de emitir o comprovante de

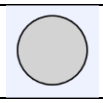
pagamento. A Figura 15 apresenta uma dependência de recurso, em que o cliente depende do caixa do banco para obter o seu comprovante de pagamento. Figura 15

Os tipos das dependências refletem o grau de liberdade existente no relacionamento. Na dependência por objetivo, cabe ao *dependee* toda e qualquer decisão para o cumprimento do objetivo. Na dependência por tarefa, o *dependee* executa a tarefa da maneira como o *depender* deseja e, por isso, cabe ao *depender* as decisões de como a tarefa deve ser executada. Na dependência por recurso, o grau de liberdade é nulo, ou seja, o *dependee* deve fornecer o recurso exatamente como o *depender* deseja. Na dependência por meta-flexível, o *depender* tem a decisão final de aceitar ou não a meta alcançada, porém ele usa o beneficio do "*know-how*" (habilidades e conhecimentos) do *dependee* [Oliveira08c].

Dede que um autor seja autônomo, ele pode optar por não se tornar dependente de outro. Ao analisar a rede de dependências, é possível concluir que algumas dependências parecem ser mais viáveis do que outras e que uma falha dentro de uma dependência pode propagar para uma grande cadeia de dependências [Yu09].

A Tabela 5 apresenta os elementos de um modelo SD.

Tabela 5 – Elementos de um modelo SD

Nome	Símbolo	Definição/Descrição
Ator/Agente		Representa um ator interessado em alcançar seus objetivos.
Meta		Representa um objetivo que só pode ser completamente satisfeito ou não satisfeito.
Meta-flexível		Representa um objetivo que pode ser satisfeito em diferentes níveis, dependendo da visão do avaliador.
Tarefa		Representa uma unidade de trabalho a ser executada.
Recurso		Representa um recurso produzido, compartilhado ou consumido pelos atores para alcançar seus objetivos.
Dependência		Representa um relacionamento de dependência.
Marcação de crítico		Marca o elemento como crítico. Significa que o ator/agente acredita que não existe outra forma para suceder se a dependência não for satisfeita.
Marcação de aberto		Marca o elemento como não comprometido/independente. Significa que a rotina pode ser afetada, mas não necessariamente falhar caso a dependência não seja satisfeita.

A Figura 16 apresenta um exemplo de um modelo SD.

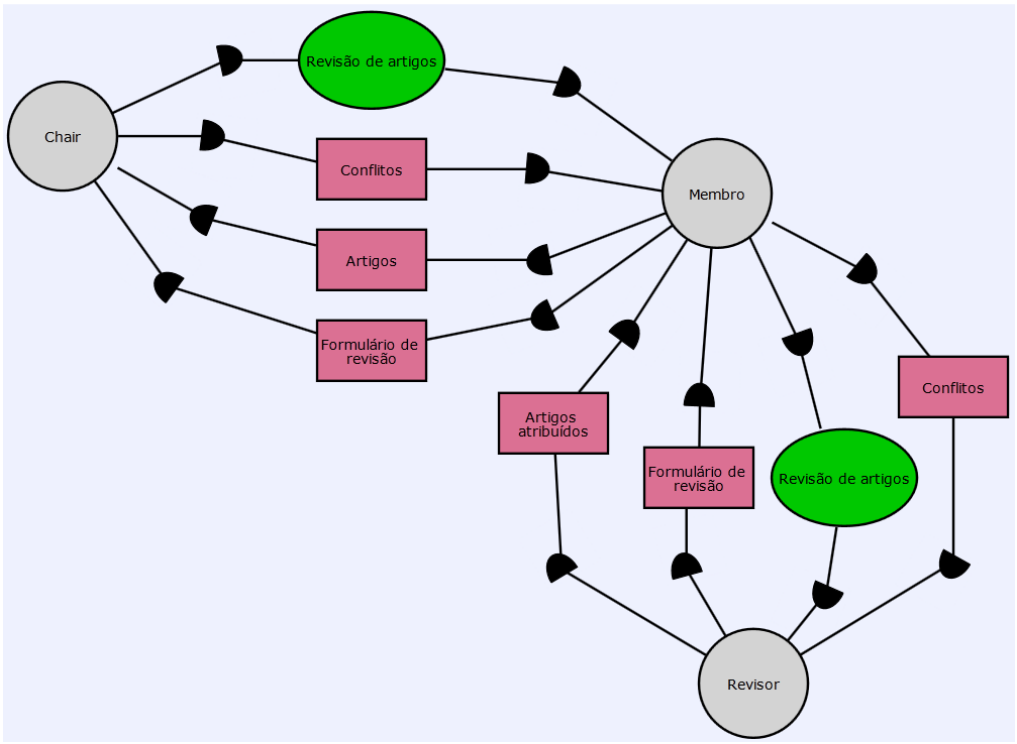


Figura 16 – Exemplo de diagrama SD [Adaptação de [Susi05]]

2.4.2. *Strategic Rationale* (Modelo SR)

O modelo SR oferece uma descrição intencional dos processos em termos de seus elementos e o raciocínio por trás deles. Enquanto o modelo SD mantém um nível de abstração por modelar somente os relacionamentos externos entre atores, o modelo SR ignora a abstração para permitir um profundo entendimento sobre os raciocínios estratégicos dos atores em relação aos processos. O modelo SR descreve os relacionamentos intencionais que são internos aos atores, tais como relacionamentos "*meios-fim*" que relacionam os elementos dos processos provendo representações explícitas do "Porque", "Como" e alternativas. Os raciocínios encontram-se no nível estratégico, logo as alternativas do processo que está sendo fundamentado também são relacionamentos estratégicos [Yu05].

Os elementos do processo usados para representar os relacionamentos intencionais internos aos atores são: os *relacionamentos "meios–fim"*, que possuem o papel de explicitar as decisões que foram tomadas para que as metas do ator fossem alcançadas, a *decomposição de tarefas*, que detalha a elaboração e realização das tarefas, além de mostrar como os recursos são disponibilizados e utilizados, e os

relacionamentos de contribuição, que exibem o tipo de contribuição (positiva ou negativa) entre metas flexíveis [Napolitano09].

[Oliveira10] Definem um metamodelo contendo os elementos e relacionamentos possíveis no i*. A Figura 18, Figura 20 e Figura 19 detalham a aplicação dos relacionamentos em elementos gráficos.

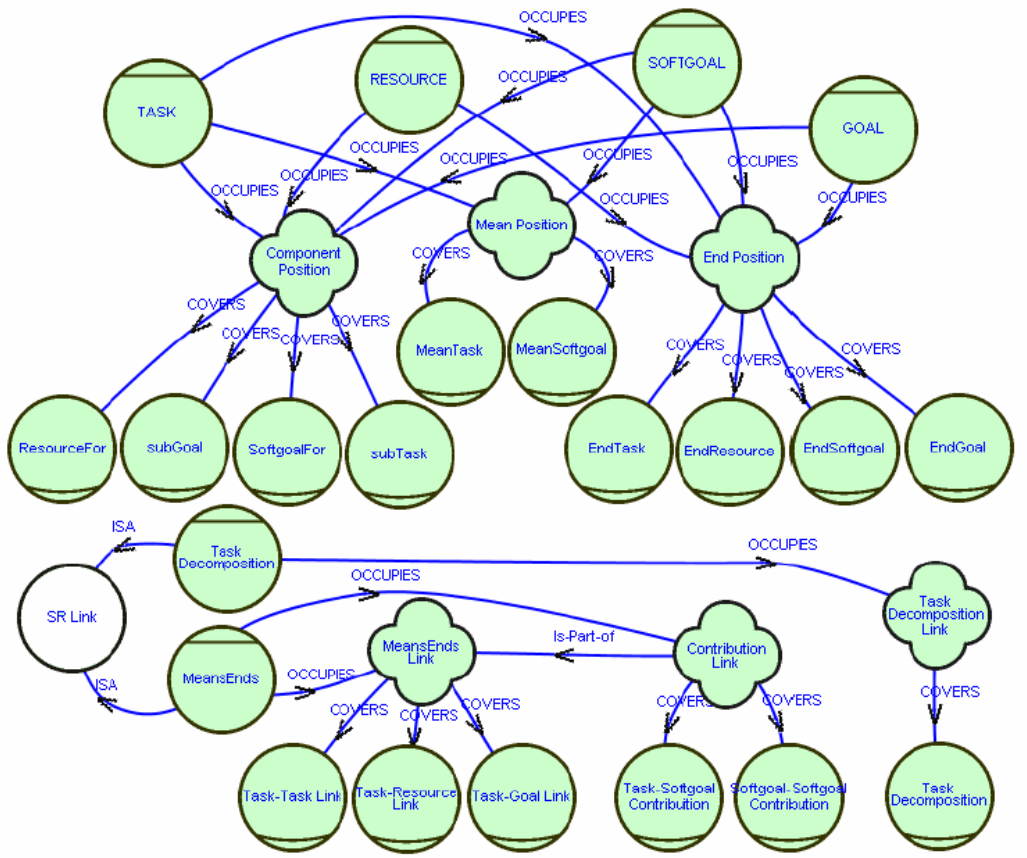


Figura 17 – Metamodelo i* [Oliveira10]

No relacionamento de decomposição de tarefa, é possível relacionar uma tarefa a subelementos que devem estar disponíveis para a execução da tarefa decomposta. Por exemplo, a decomposição de uma tarefa em uma meta implica que essa meta deve ser satisfeita primeiramente para que a tarefa possa ser executada. Se fosse uma decomposição de recurso, ele deveria estar disponível primeiramente para consumo da tarefa.

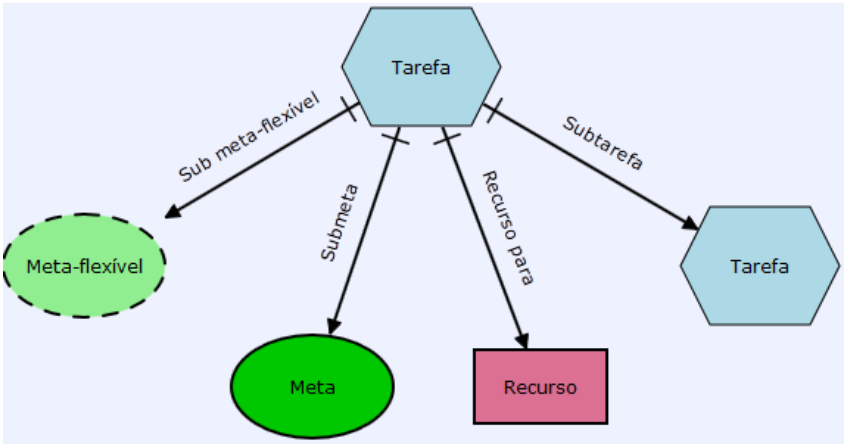


Figura 18 – Ligações para o relacionamento de Decomposição

O relacionamento *means-end* (meios-fim) registra "como" e "o que" foi executado (*means*) para que um dado "fim" pudesse ser realizado. Por exemplo, para um recurso ser produzido, alguma atividade foi realizada. Neste caso a atividade tem o papel de "*means"* e o recurso, o papel de "*end"*. Somente tarefas possuem o papel de "*means"*, porém os elementos recurso, tarefa e meta podem ter o papel de "*end"*.

Ainda existe o caso específico de relacionamento "*means-end"* para metas-flexíveis que é explicado a seguir. A Figura 19 apresenta as possibilidades de aplicação do relacionamento "*means-end"* de forma gráfica.

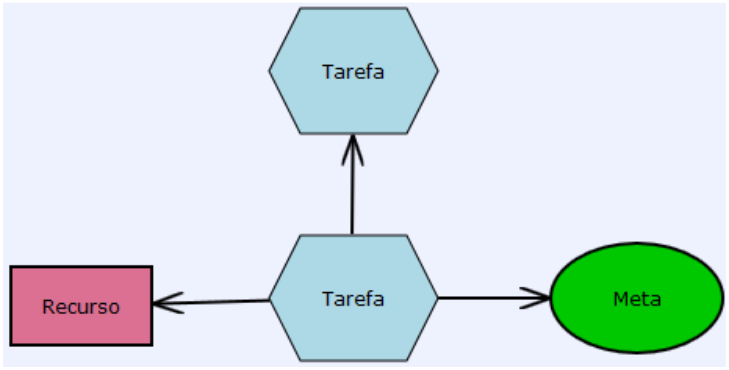


Figura 19 - Ligações para os relacionamentos de "meios-fim"

O relacionamento do tipo *Contribuição* é um sub-relacionamento do tipo *Means-end* e herda as suas características [Oliveira10], entretanto, por relacionar especificamente os elementos *Tarefa* e meta-flexível a outro elemento do tipo meta-flexível, recebe *labels* que expressam o significado de contribuição. A Figura 19 apresenta esses relacionamentos.

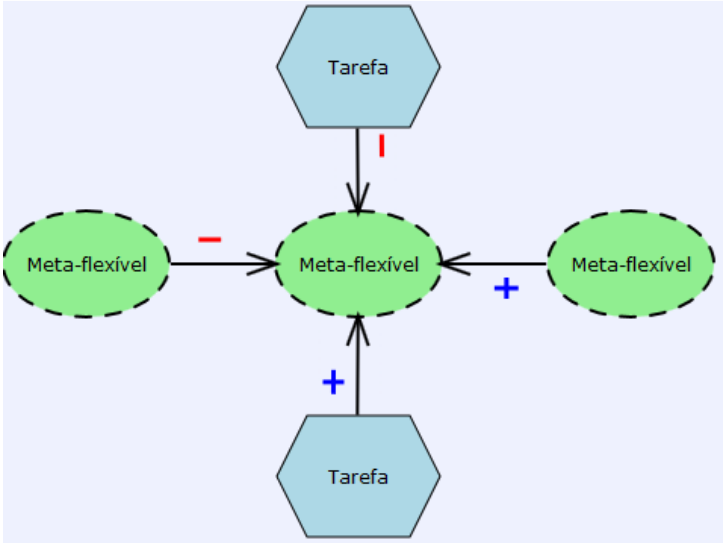
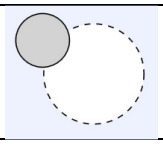


Figura 20 – Ligações para os relacionamentos de contribuição positiva e negativa

A Tabela 6 apresenta os elementos de um modelo SR e sua respectiva descrição.

Tabela 6 – Elementos do modelo SR

Nome	Símbolo	Definição/Descrição
Ator/Agente		Representa um ator interessado em alcançar seus objetivos.
Meta		Representa um objetivo que só pode ser completamente satisfeito ou não satisfeito.
Meta-flexível		Representa um objetivo que pode ser satisfeito em diferentes níveis, dependendo da visão do avaliador.
Tarefa		Representa uma unidade de trabalho a ser executada.
Recurso		Representa um recurso produzido, compartilhado ou consumido pelos atores para alcançar seus objetivos.
Dependência		Representa um relacionamento de dependência.
Marcação de crítico		Marca o elemento como crítico. Significa que o ator/agente acredita que não existe outra forma para suceder se a dependência não for satisfeita.
Marcação de aberto		Marca o elemento como não comprometido/independente. Significa que a rotina pode ser afetada, mas não necessariamente falhar caso a dependência não seja satisfeita.
Decomposição de tarefa		Representa um relacionamento que expressa a decomposição de uma tarefa em outros elementos.
Meios-fim		Representa um relacionamento que expressa a ligação entre um fim e um meio

		de alcançá-lo.
Contribuição positiva		Representa um relacionamento que expressa contribuição positiva para um meta-flexível.
Contribuição negativa		Representa um relacionamento que expressa contribuição negativa para um meta-flexível.
Pool do ator/agente		Representa um ator/agente em forma de *pool*.

A Figura 21 exemplifica um modelo misto (SD + SR). O conteúdo do modelo SR encontra-se na *pool* do ator central, onde é modelado o seu *rationale*, ou seja, a estratégia utilizada para que ele alcance suas metas. Ao redor, apresentam-se as modelagens das dependências de recursos, tarefas e meta com outros atores.

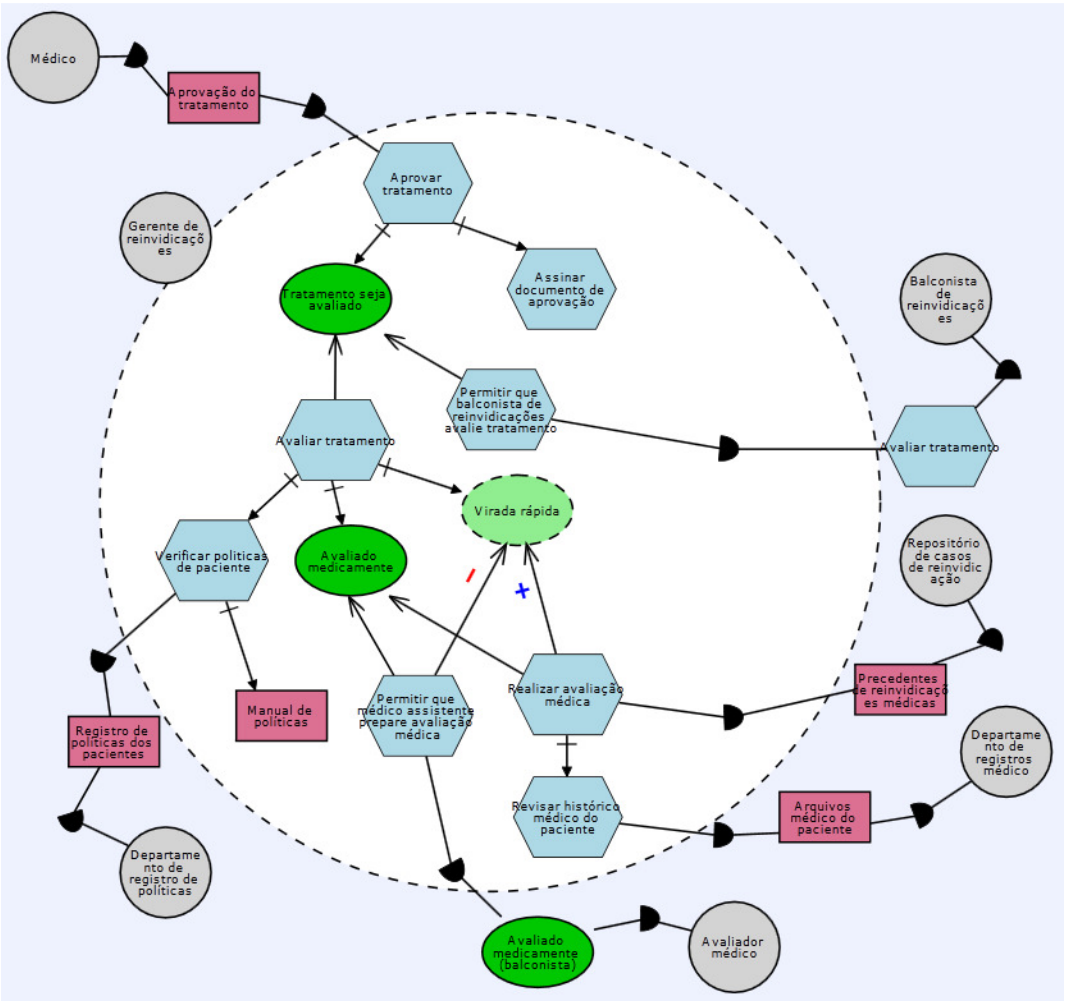


Figura 21 – Exemplo de modelo SR (adaptação de [Yu95])

Esta seção apresentou o *framework* i*, que oferece uma linguagem para a modelagem de dependências de recursos e metas entre atores, bem como a

modelagem das estratégias de cada ator para alcançar suas metas. Observa-se que a modelagem de objetivos do i* ocorre em um plano de visão distinto das modelagens tradicionais de objetivos organizacionais que são definidos em alto nível, trazendo uma nova noção que permite análises mais apuradas dos riscos e estratégias envolvidas na execução dos processos organizacionais.

A próxima seção apresenta a URN, que é uma notação composta por uma linguagem de modelagem de processos (UCM) e uma linguagem de modelagem de objetivos (GRL).

2.5. URN (UCM e GRL)

A URN (*User Requirement Notation*) é uma notação composta por duas sublinguagens, uma específica para modelagem de processos e outra para modelagem de objetivos. Essa notação é um padrão nomeado como Z.151 e recomendado pela ITU-T (*International Telecommunication Union)* no contexto das Técnicas de Descrições Formais (FDT) [ITU08].

A URN foi projetada para oferecer recursos para elicitação, análise, especificação e validação de requisitos. Entre os objetivos do desenvolvimento do padrão URN, temos: capturar os requisitos do usuário quando apenas poucos detalhes de design estão disponíveis; focalizar os estágios iniciais de design; avaliação antecipada de desempenho e detecção antecipada de interações indesejadas [Amyot&Mussbacher02].

A URN é composta por duas notações que representam conceitos de modelagem e notações para objetivos (com foco em requisitos não funcionais e atributos de qualidade) e cenários (com foco nos requisitos operacionais, requisitos funcionais, desempenho e raciocínio (*reasoning*) arquitetural) [Sikandar03].

A notação com foco em objetivo é chamada de GRL (*Goal-oriented Requirements Language*) e a notação de cenários é chamada de UCM *(Use Case Map*). A partir dessa combinação, a UCM permite a captura de objetivos e raciocínios de decisão (utilizando GRL) e a capacidade de modelar sistemas dinâmicos que possuem comportamento variável em tempo de execução (utilizando UCM) [Sikandar03].

As próximas subseções detalham a GRL e a UCM.

2.5.1.GRL

A GRL é uma extensão da linguagem i* [Oliveira08c] que foi desenvolvida para apoiar a modelagem orientada a metas e a representação do raciocínio (*reasoning*) de atores, especialmente para lidar com requisitos não funcionais. Ele fornece elementos para expressar vários tipos de conceitos que aparecem durante o processo de requisitos. Existem três categorias principais de conceitos: elementos intencionais, links, e atores. Os elementos intencionais na GRL são tarefa, objetivo, meta-flexível e recurso. Eles são intencionais porque são usados por modelos que permitem identificar o motivo de determinados comportamentos, o motivo de aspectos estruturais e informacionais serem escolhidos para serem incluídos nos requisitos do sistema, quais alternativas foram consideradas, quais critérios foram usados para deliberar entre as alternativas, e quais as razões para a escolha de uma alternativa e não outra [GRL12].

A GRL suporta o raciocínio sobre cenários, estabelecendo correspondências entre elementos GRL intencionais e não intencionais em modelos de cenários da URN. Modelar ambos os objetivos e cenários é complementar e pode ajudar na identificação de objetivos e cenários adicionais (e as etapas de cenário) importante para os interessados, contribuindo assim para a integralidade e exatidão dos requisitos.

Alguns dos elementos intencionais da GRL são reutilizados do i*, tais como, ator/agente, *goal*, *task*, meta-flexível e *goal* (meta). Entre os relacionamentos reutilizados encontram-se a *contribuição*, *dependência e decomposição*. Entretanto, novos elementos são adicionados, tais como, *belief,* link de *correlação*, níveis de satisfação e tipos de contribuição. A Figura 22 apresenta os níveis de satisfação e tipos de contribuição (também presentes no i*, provêm do NFR *framework* [Chung00]). A Tabela 7 apresenta os outros elementos do modelo GRL.

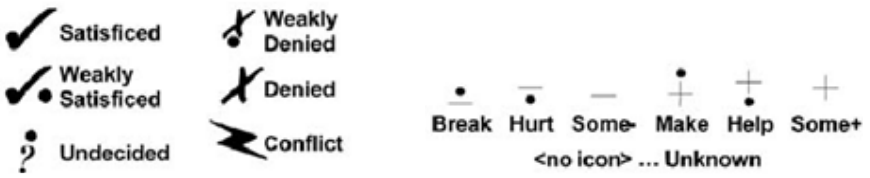
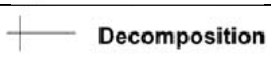


Figura 22 – Níveis de satisfação (à esquerda) e tipos de contribuição (à direita) da GRL

Tabela 7 – Elementos do modelo GRL

Nome	Símbolo	Definição/Descrição
Ator/Agente		Representa um ator interessado em alcançar seus objetivos.
Meta		Representa um objetivo que só pode ser completamente satisfeito ou não satisfeito.
Meta-flexível		Representa um objetivo que pode ser satisfeito em diferentes níveis, dependendo da visão do avaliador.
Tarefa		Representa uma unidade de trabalho a ser executada.
Recurso		Representa um recurso produzido, compartilhado ou consumido pelos atores para alcançar seus objetivos.
Crença		Raciocínio ou argumentação associada a uma contribuição ou uma relação.
Dependência		Representa um relacionamento de dependência.
Decomposição		Define o que é necessário para uma tarefa ser executada.
Contribuição		Descreve coma metas-flexíveis, tarefas, crenças, e relações contribuem entre si. Cada contribuição pode ser qualificada por um nível: *equal, break, hurt, some-, some+, undetermined, help ou make* (Figura 22).
Correlação		Contribuição que indica efeitos colaterais em outro elemento intencional.

A Figura 23 apresenta um exemplo de modelo em GRL utilizando grande parte dos elementos da linguagem.

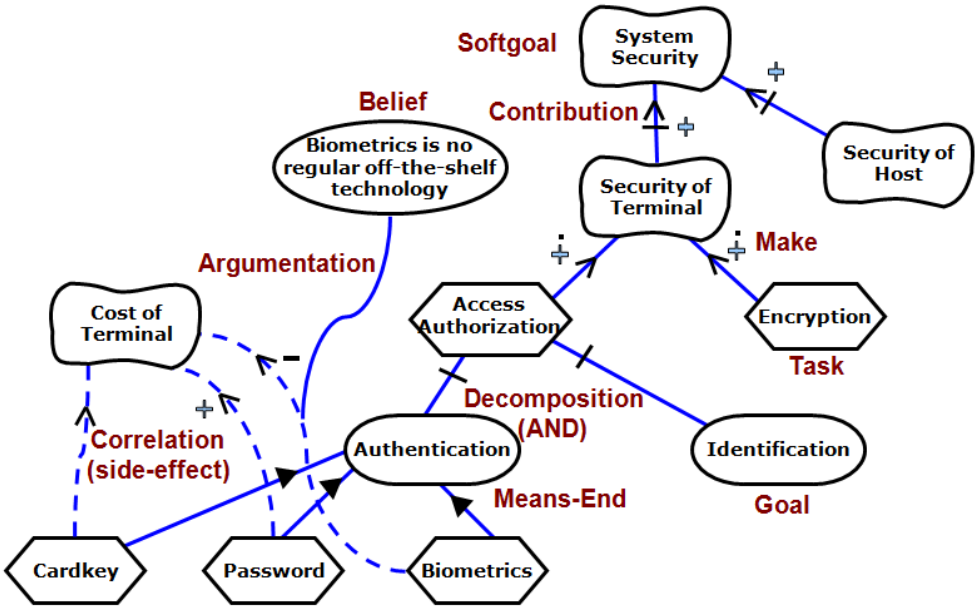


Figura 23 – Exemplo de modelo GRL [Amyot02]

2.5.2.UCM

A modelagem de requisitos funcionais de sistemas complexos, muitas vezes implica em uma ênfase inicial nos aspectos comportamentais, tais como interações entre o sistema e o seu ambiente (e usuários), de forma a definir os relacionamentos entre suas interações e sobre as atividades intermediárias realizadas pelo sistema. Cenários representam e maneira excelente e útil de descrever esses aspectos. [Amyot&Mussbacher02].

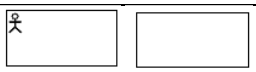
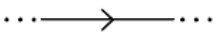
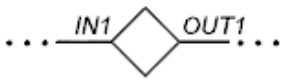
Os modelos UCM são usados como uma notação visual para a descrição de relações causais entre responsabilidades e um ou mais cenários. São mais úteis nas fases iniciais de desenvolvimento de software e são aplicáveis para captura e elicitação e validação de cenários, bem como design arquitetônico de alto nível e geração de casos de teste. Também fornecem um *framework* comportamental para avaliação e tomada de decisões arquiteturais em um alto nível, opcionalmente, com base em análise de desempenho dos modelos UCM. Além disso, fornecem aos seus usuários a capacidade dinâmica (em tempo de execução) do refinamento de variações de cenários e de estrutura como forma de permitir o desenvolvimento incremental e integração de cenários complexos [Amyot&Mussbacher02].

As principais vantagens da UCM são de que ela pode modelar refinamentos dinâmicos para diversos comportamentos e estruturas, além de integrar componentes

estruturais e comportamentais em uma única visão, realizando assim, a ponte entre requisitos e projeto [Amyot&Mussbacher01].

Os elementos principais da linguagem UCM são descritos a seguir (Tabela 8):

Tabela 8 – Elementos do modelo UCM

Nome	Símbolo	Definição/Descrição
Ator/Time		Representa o responsável ou grupo de responsáveis no processo.
Ponto inicial		Captura precondições e eventos de disparo.
Ponto final		Representa eventos resultantes e pós-condições.
Responsabilidades		Locais onde, por exemplo, procedimentos, atividades e funções são necessários.
Caminhos		Conecta pontos iniciais a pontos finais e pode ligar responsabilidades em um caminho causal.
Stub		Representa relacionamentos com outros diagramas.

Cenários grandes e complexos podem ser decompostos e estruturados graças aos submapas chamados plugins, usado (e reutilizados) em recipientes de mapas chamados stubs e exibidos em formato de diamantes (Tabela 8).

A Figura 24 apresenta um exemplo de diagrama UCM. Este diagrama retrata um cenário de alto nível (comumente chamado de *Root Map*), enquanto os dois outros modelos UCM da Figura 25 e Figura 26, são plugins para o *stub Authenticate*.

Neste exemplo do diagrama principal, um contribuinte quer acessar o contador eletrônico através do sistema de segurança. Após a identificação de contribuinte (CheckId), o sistema necessita autenticar o solicitante, que pode ser aceito ou rejeitado. Para realizar a autenticação, pode ser utilizado o *plugin* biométrico ou senha. Uma vez autenticado, o contador eletrônico cria uma nova sessão, no primeiro caso e em seguida, torna-se pronto para processar as transações (todos os do presente em paralelo com uma notificação de aceitação).

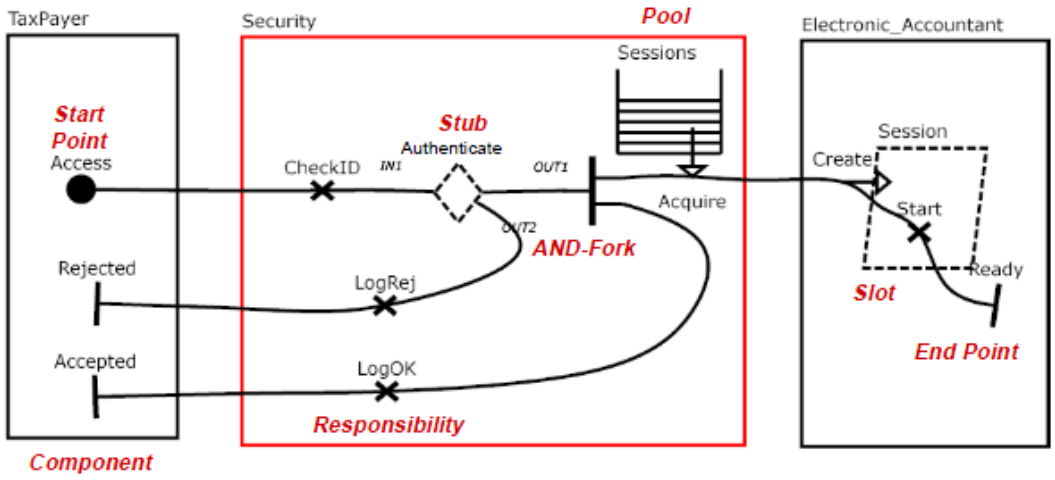


Figura 24 – Exemplo de diagrama UCM (diagrama principal) [Amyot&Mussbacher02]

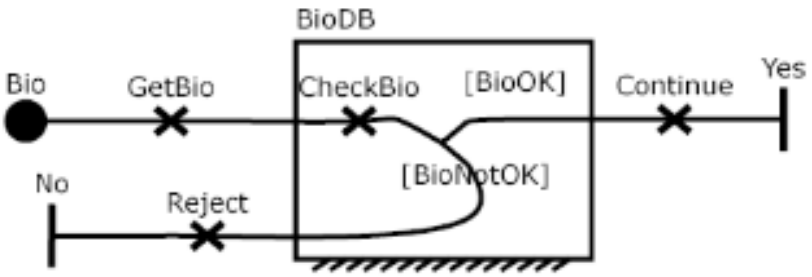


Figura 25 – *Plug-in* Biométrico [Amyot&Mussbacher02]

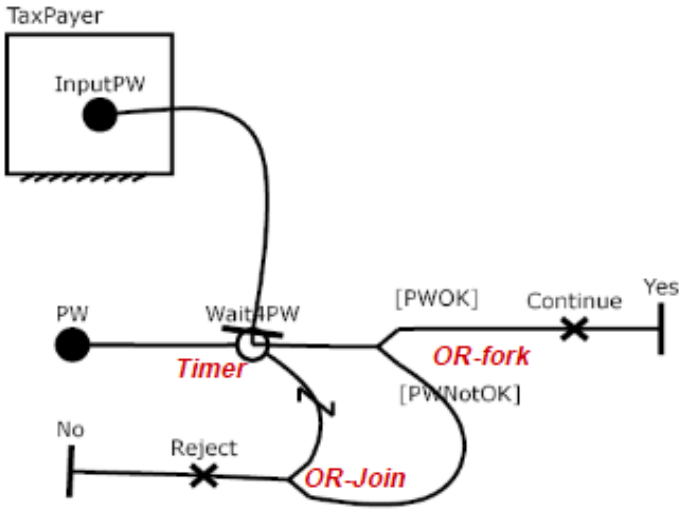


Figura 26 – *Plugin* Senha [Amyot&Mussbacher02]

Os candidatos a alternativas para a autenticação (biometria ou senha), se incluidos no modelo GRL, podem ser operacionalizados baseado no *reasoning,* desta forma é possível ter maior controle estratégico, além de possibilitar uma análise mais detalhada do contexto nos modelos.

2.5.3.Rastreabilidade entre UCM e GRL

A ferramenta jUCMNav é o principal recurso computacional para a criação de modelos UCM e GRL. Esta ferramenta foi desenvolvida como um *plugin* do Eclipse e encontra-se em constante evolução.

A integração dos modelos UCM e GRL dentro da UCM ocorre de várias formas. A forma mais simples é através do relacionamento de rastreabilidade (t*raceability)* [Ghanavati09] que cria um tipo de "atalho" entre o modelo UCM e o GRL através de um ícone no formato de um triângulo amarelo na horizontal (Figura 27).

Outra forma é o uso de relacionamentos específicos, tais como links de realização [Behnam10] (Figura 28), responsabilidade, rastreabilidade e complacência [Ghanavati09].

Diversos trabalhos são desenvolvidos para ampliar a capacidade de alinhamento e análise dos modelos UCM e GRL, por exemplo, com o uso de indicadores, *frameworks* de alinhamento [Ghanavati09] e uso de *templates* [Behnam10].

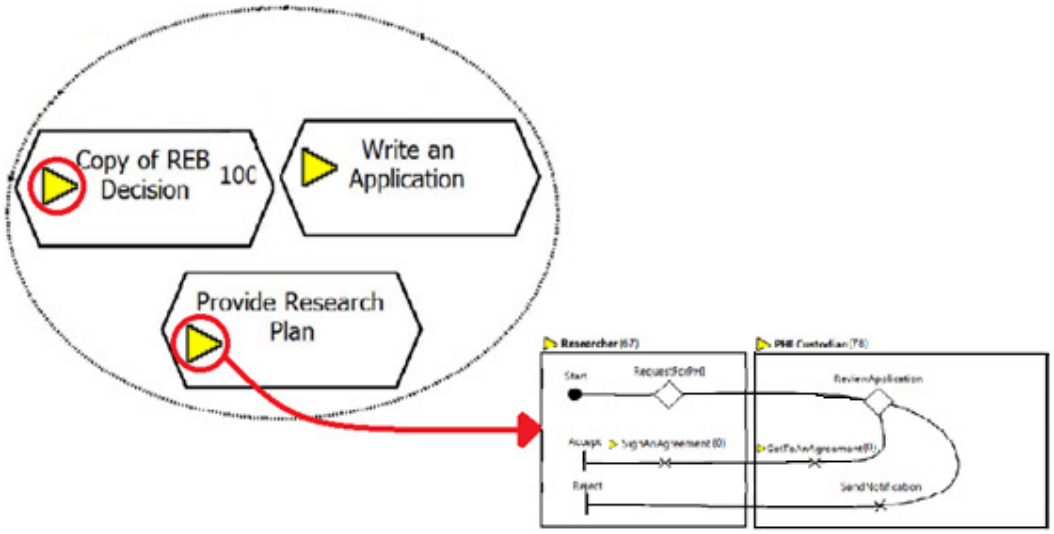


Figura 27 – Exemplo de relacionamento do tipo *Atribuição* entre GRL e UCM [Adaptação de Ghanavati09]

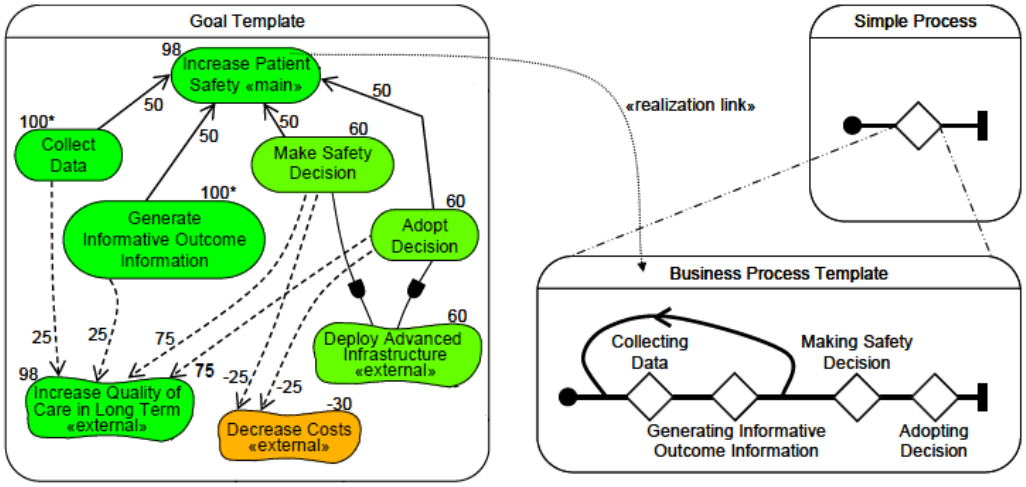


Figura 28 – Exemplo de relacionamento do tipo *realização* entre modelos GRL e UCM

Esta seção apresentou o padrão Z.151, conhecido como URN, proposto pela ITU-T (*International Telecommunication Union).* Este padrão oferece a modelagem em alto nível de objetivos utilizando a linguagem GRL, e a modelagem de cenários, utilizando o padrão UCM. Um dos pontos positivos desse padrão é a existência de rastreabilidade entre o modelo de objetivos e o modelo de execução. Além disso, diversos estudos expandem a linguagem incluindo novos elementos e relacionamentos de forma a ampliar a capacidade da linguagem no sentido de garantir que os objetivos sejam cumpridos.

Na próxima seção será apresentado o *framework* ARIS, considerado um dos mais importantes no contexto da modelagem de processos de negócio.

2.6.*Framework* ARIS

A Arquitetura de Sistemas de Informação Integrado (ARIS – *Architecture of Integraterd Information Systems)* é um conceito (e não de uma simples ferramenta) desenvolvido pelo professor August-Wilhelm Scheer, na Alemanha. Seu objetivo é prover um *framework* que ofereça meios de expressar os conceitos do negócio de forma precisa e, assim, permitir análises detalhadas e prover um ponto inicial não ambíguo para o desenvolvimento de sistemas de informação computacionais [Davis07].

A partir da definição do *framework* ARIS, foi desenvolvida a ferramenta ARIS *Toolset*, lançada em 1992, sendo utilizada em larga escala para a modelagem de processo de negócio por empresas de diversos tamanhos, sendo uma ferramenta líder no ramo [Davis02].

No contexto da modelagem de processos de negócio, a ferramenta oferece 3 níveis de abstração que são implementados em 3 modelos distintos, com objetos específicos. O primeiro nível é formado pelos processos que estão diretamente envolvidos na criação de valores para a companhia, que são modelados no diagrama Cadeia de Valor Agregado (VAC – *Value-added chain*) [Scheer&AG05]. O segundo nível é basicamente formado pelo fluxo de atividades, recursos, eventos e regras, que formam o diagrama de Cadeia de Processos e Eventos (EPC – *Event-Driven Process*), considerado o modelo central de toda a modelagem de negócio no ARIS [Davis07]. Por último, no nível mais baixo, estão modelados os elementos que representam a execução das atividades com o objetivo de oferecer informações adicionais sobre a atividade, em particular, sobre a transformação dos dados de entrada e saída [Davis07]. Esses elementos são modelados no Diagrama de Alocação de Função (FAD – *Function Allocation Diagram*).

Além desses modelos que são específicos para modelar os processos de negócio em diferentes níveis de abstração, o ARIS também possui um modelo específico para a modelagem de objetivos onde é possível definir e relacionar objetivos de forma hierárquica juntamente com os processos e Fatores Críticos de Sucesso [SeildImeier&Storr04].

Esses diagramas são detalhados nas subseções seguintes.

2.6.1.VAC – Valued Added Chain

O diagrama VAC especifica as funções em uma organização as quais influenciam diretamente o real valor agregado da organização. Estas funções podem ser ligadas a outras, de forma a sequenciá-las e então formar a cadeia de valor agregado [ARIS06].

Os processos podem ser relacionados aos respectivos objetivos e indicadores. Um modelo VAC pode ser detalhado em outros macroprocessos do mesmo tipo (VAC). Em um diagrama VAC, as funções podem ser dispostas em uma hierarquia ou similar a uma árvore de função. A orientação do processo subordinado ou superior é sempre ilustrada. Assim como a representação da função subordinada ou superior, um diagrama de cadeia de valor representa as ligações entre as funções e as unidades organizacionais e as funções e objetos informativos [ARIS06].

A Figura 29 exemplifica um modelo VAC. Este modelo possui o macroprocesso “Realizar empréstimos para pessoa física”, composto por outros três macroprocessos que estão ligados aos seus respectivos objetivos e indicadores (KPI).

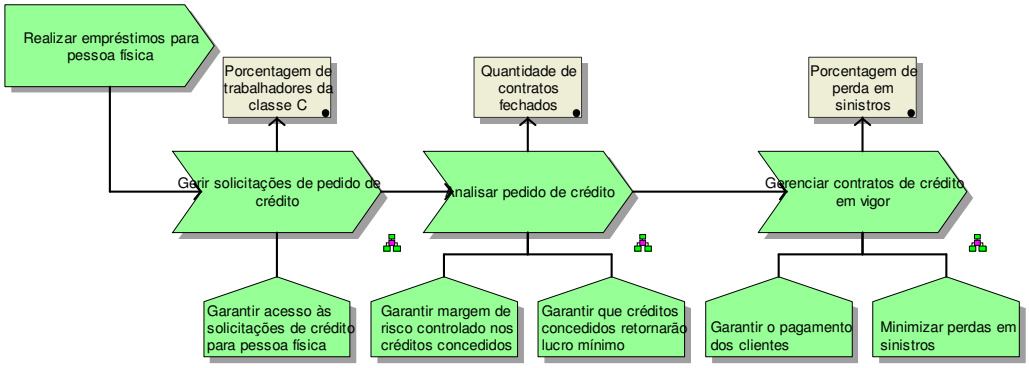


Figura 29 – Exemplo de modelo VAC [Sousa10]

A visão de macroprocesso é conhecida como uma visão gerencial devido a sua abstração. Essa visão é moldada durante a primeira fase do ciclo de vida da modelagem de processos de negócio com o propósito de adquirir conhecimento necessário para calcular tempo e custo de um projeto de modelagem de processos do negócio [Sousa10].

*2.6.2.*EPC – *Event-driven Process Chain*

O diagrama EPC é o modelo central para toda a modelagem de negócio. É um modelo dinâmico que traz consigo os recursos estáticos do negócio (por exemplo, sistemas, organizações e dados) e os organiza para conceber uma sequência de tarefas ou atividades (‘o processo’) que agrega valor ao negócio [Davis02].

Este modelo possui similaridades em relação ao modelo desenvolvido a partir do uso da notação BPMN, já que ambos detalham os processos de negócio ao nível de atividades utilizando elementos semelhantes. O fluxo de trabalho detalhado em um modelo EPC engloba informações como papéis executores e suas unidades organizacionais, raias que organizam as atividades de acordo com seus papéis executores, elementos de interface com outros processos, eventos que demarcam o início e o fim do processo, eventos intermediários que sinalizam circunstâncias importantes para a continuidade do processo, principalmente nos fluxos de decisão, operadores lógicos para inserir regras no fluxo e as próprias atividades que serão executadas ao longo do processo [Sousa10].

A Figura 30 ilustra um exemplo de EPC. Este modelo possui atividades executadas pela área de “Crédito e taxas contratuais” e pelo sistema “Crédito direto”. As atividades de responsabilidade de cada papel estão respectivamente em suas raias. As linhas que cruzam a linha das raias representam *handoffs* entre os papéis executores do processo.

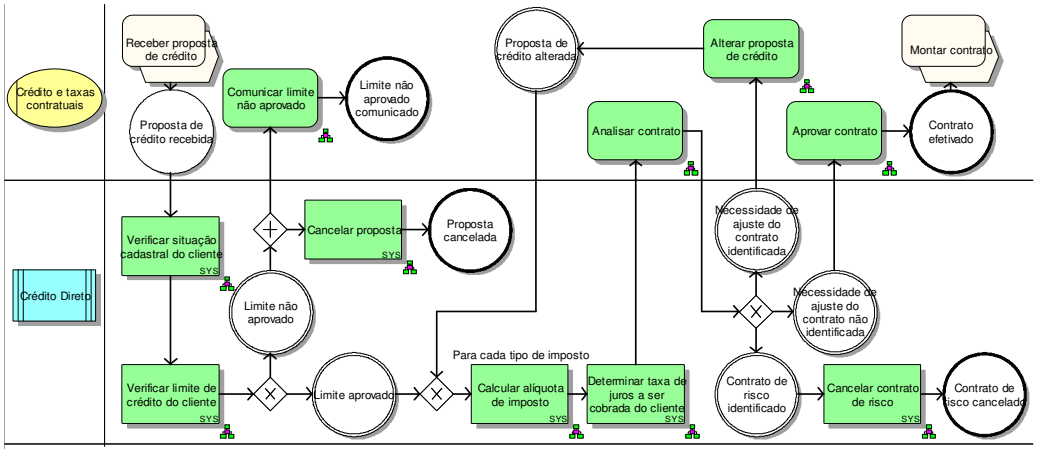


Figura 30 – Exemplo de modelo EPC [Sousa10]

*2.6.3.*FAD – *Function Allocation Diagram*

O diagrama FAD é um modelo que possui informações sobre uma dada atividade do ponto de vista operacional. É utilizado para apresentar uma visão mais detalhada dos recursos disponíveis e necessários, que são relevantes para as atividades. O FAD também é utilizado para reduzir a complexidade dos processos de negócio (neste caso, considere o modelo EPC), representando elementos como, por exemplo, funções, áreas, sistemas de apoio, entradas e saídas de dados, documentos e riscos envolvidos nas atividades [Sousa10]. Observe que a notação BPMN não possui modelos FAD, e que o fluxo de informações é modelado no próprio diagrama principal.

A Figura 31 exemplifica um modelo FAD. Esta atividade é executada pelo sistema “Crédito Direto”, caracterizando uma atividade automatizada. As informações que a atividade consome são “Proposta de crédito”, “Créditos concedidos” e “Cadastro de cliente”, sendo esta última extraída da base de dados “BDCliente”. O resultado da atividade é a informação “Proposta de crédito” atualizada com um status de aprovação que depende da regra de negócio “Verificação de limite de crédito” e que é implementado de acordo com o requisito de negócio “Verificar limite de crédito do cliente”.

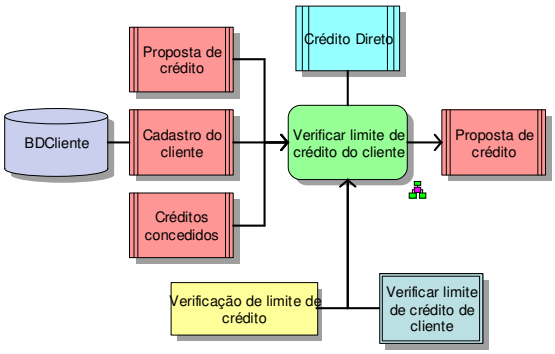


Figura 31 – Exemplo de modelo FAD [Sousa10]

2.6.4.Diagrama de objetivos

Além da possibilidade de relacionar o objetivo ao processo na cadeia de valor, ainda é possível detalhar os objetivos de negócio em um diagrama específico. O diagrama de objetivos trata os objetivos como elementos centrais e permite o relacionamento com processos, produtos/serviços, Fatores Críticos de Sucesso e o relacionamento entre objetivos de forma hierárquica. A Tabela 9 e Tabela 10 apresentam respectivamente os elementos do diagrama de objetivos e os tipos de relacionamentos disponíveis entre os elementos. Observe que os relacionamentos são limitados e não podem ser customizados (apenas é possível substituir o nome de representação, mas não o tipo).

Tabela 9 – Elementos de um diagrama de objetivos

Nome	Símbolo	Definição/Descrição
Objetivo	Objective	O objetivo pode ser decomposto em subobjetivos e deve estar relacionado direta ou indiretamente a um processo.
Processo	Process	Um processo deve ter um ou mais objetivos relacionados.
Fator crítico de sucesso (FCS)	Critical factor	O fator crítico de sucesso é relacionado ao objetivo e representa algo que deve ser tratado para que o objetivo seja alcançado.
Produto/Serviço	Product/Service	Um Produto/Serviço é consumido ou produzido por um processo.

Tabela 10 – Conexões permitidas entre os elementos do diagrama de objetivos

Conexão x Objeto				
	Objetivo	Processo	Fator crítico	Produto/ Serviço
Objetivo	Belongs to	Supports	N/A	N/A
Processo	Supports	N/A	N/A	Has output of
Fator crítico	Is critical fator for	N/A	N/A	N/A
Produto/ Serviço	Supports	is input for	N/A	N/A

A Figura 4 apresenta um exemplo de modelo de objetivos. O objetivo "*Improve Efficiency*" é composto pelos objetivos "*Reduce Costs*" e "*Increase returns*". Este último encontra-se relacionado ao Fator Crítico de Sucesso "*Returns*". Os processos "*Execute advertising Campaign*" e "*Check price strategy*", estão relacionados ao objetivo "*Increase market share*".

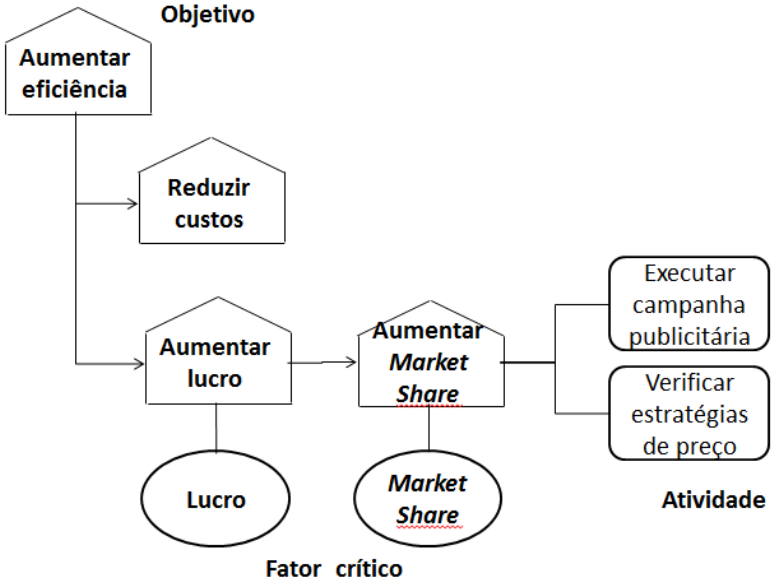


Figura 32 – Modelo de objetivos

Esta subseção apresentou a modelagem de processos de negócio e objetivos utilizando a ferramenta ARIS, que implementa o *framework* ARIS. A ferramenta oferece 3 níveis de modelagem de processos de negócio (VAC, EPC e FAD) além de um diagrama específico para a modelagem de objetivos.

Por ser uma ferramenta comercial, a customização de elementos no ARIS é bastante limitada de forma a garantir sempre as regras de seu *framework* original. Portanto, existem dificuldades para representar modelos mais detalhados na semântica dos relacionamentos entre os elementos, por exemplo, relacionamentos de medidas qualitativas/quantitativas, que não existem na linguagem. Essas limitações tornam difíceis análises de alinhamento entre os processos e objetivos. Por exemplo, não é possível identificar as atividades chave de um processo sem a rastreabilidade

entre as necessidades de um objetivo e as atividades executadas. Também não é possível utilizando-se apenas da notação disponível, verificar através dos modelos se o processo é capaz de satisfazer um objetivo, ou ainda, medir de forma quantitativa o seu impacto na camada estratégica do negócio.

Particularmente, cabe observar que apesar da plataforma ARIS possibilitar uma hierarquização dos objetivos por intermédio do diagrama de objetivos, o consequente alinhamento a diferentes processos é de difícil execução principalmente pela ausência de relacionamentos que possuam semântica para criar a rastreabilidade entre os objetivos e atividades que visam satisfazê-los.

A próxima seção apresenta uma rápida análise das linguagens vistas e justificativas para a seleção das linguagens utilizadas na integração.

2.7. Seleção de linguagens

O objetivo desta seção é ponderar sobre cada linguagem analisada nas subseções anteriores, de forma a realizar uma análise breve sobre seus pontos positivos e negativos e então justificar a escolha das linguagens para o trabalho de integração.

O que se deseja é uma linguagem para modelagem de objetivos e outra para modelagem de processos de negócio. O trabalho consiste em criar a rastreabilidade entre as linguagens, adicionando os elementos necessários, além de propor um método de verificação do alinhamento no nível de modelagem.

Um dos objetivos de seleção das linguagens é o reuso de seus elementos e definições de interface, uma vez que seria um retrabalho projetar novos elementos representativos, até porque muitas das definições utilizadas na modelagem já possuem certas representações padronizadas mesmo entre diferentes ferramentas e notações. Por outro lado, não vamos no ater às regras de cada linguagem, mas realizar as adaptações que forem necessárias para melhor alinhar os modelos. O conjunto de alterações expressivas na união das linguagens resultará em uma nova solução, porém derivada das linguagens escolhidas.

Nesta avaliação, não somente a capacidade da linguagem será importante, mas todo o contexto de aceitabilidade do mercado e a importância da linguagem dentro de seu domínio. Iniciaremos as análises no domínio dos processos de negócio e posteriormente, no domínio dos objetivos.

2.7.1.Domínio de processos de negócio

Seguindo de acordo com a sequência de apresentações das linguagens neste capítulo, temos a seguinte lista: UML, BPMN, UCM e EPC.

A modelagem de processos de negócio na UML inicialmente era desenvolvida utilizando o diagrama de atividades padrão que não possuía recursos suficientes para modelar a complexidade de um processo de negócio. No entanto, com a proposta de extensão de [Eriksson&Penker00], foram incluídos novos conceitos, aumentando muito a capacidade de representação da linguagem.

A UML é uma linguagem fortemente consolidada no contexto do desenvolvimento de software e seu diagrama de atividades já é utilizado há anos pelos analistas, entretanto, o fato da extensão de processos de negócio não ser um elemento nativo da UML é um ponto negativo. O diagrama de atividades em si permite apenas o desenho simples do fluxo de atividades, não permitindo, por exemplo, o registro de insumos e produtos das atividades, tais como informações e artefatos. Outro fator agravante é que, apesar desta proposta surgir em 2000, a OMG lançou a BPMN em 2001, alcançando rapidamente um grande número de usuários.

A BPMN é um padrão atualmente desenvolvido pela OMG, que é um consórcio aberto, em nível internacional, que foi fundado em 1989 com o objetivo de criar padrões através do apoio dos diversos membros participantes ao redor do mundo [OMG12]. Entre as grandes empresas que participaram do desenvolvimento da BPMN 2.0, podemos citar Oracle, SAP AG, Unisys, Accenture, IDS Scheer, *Red Hat*, TIBCO e *Hewlett-Packard*.

Com o auxilio de diversos usuários, é possível desenvolver os padrões unindo a experiência de vários grupos, além de facilitar a aceitabilidade e uso dos conteúdos gerados. Com isso, a BPMN é atualmente a notação mais utilizada para a modelagem de processos de negócio [OMG11a].

Isso também se justifica pelo grande número de ferramentas comerciais e principalmente gratuitas (incluindo ferramentas de código aberto) disponíveis no mercado. Como exemplo de ferramentas gratuitas (algumas livres) que oferecem a BPMN podemos citar ARIS Express, Barium Live, BizAgi *Process Modeler*, BonitaSoft, Eclipse BPMN *Modeler*, Intalio BPMS, *Interfacing Free Vision* BPMN *Modeler*, ITP Commerce BPMN Visio *plug-in*, *Oryx*, Processmaker *Open Source* BPM *and Workflow*, Questetra BPM Suite, TIBCO *Business Studio*, Visio BPMN *Modeler* e ainda um vasto número de ferramentas que podem ser encontradas em [Louw12]. Isso demonstra a grande aceitação da notação e a maturidade de difusão, já que temos ferramentas desenvolvidas por empresas de diversos países.

Na capacidade da notação em representar elementos de processos de negócio, é a linguagem que mais oferece recursos gráficos. Identifica-se um grande número de componentes, muitas vezes de um mesmo tipo, mas com elemento gráfico levemente distinto para representar situações específicas. Um exemplo são os eventos, que podem ocorrer temporalmente por diversos motivos, entretanto, dentro de um processo de negócio, é comum certos eventos serem comuns, por exemplo, a emissão de uma mensagem. Esses elementos são chamados de *triggers* (Figura 33).

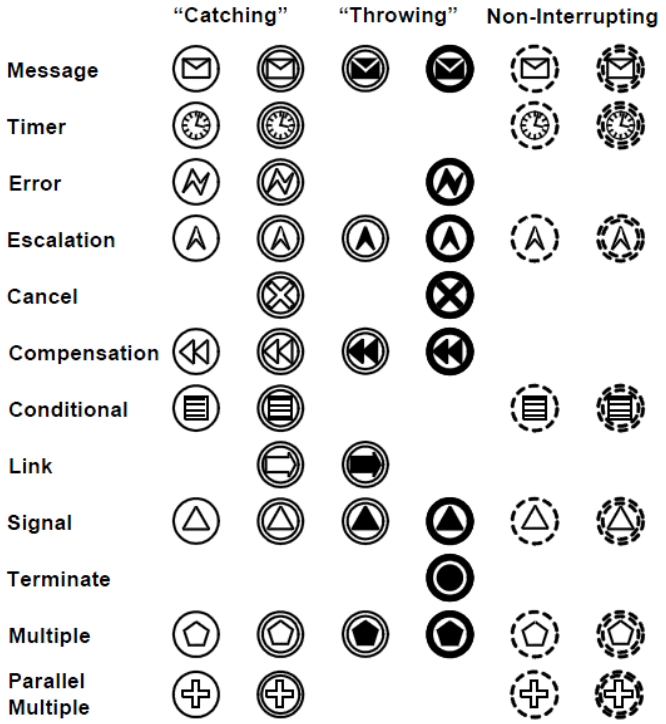


Figura 33 – Tipos de eventos da BPMN

A BPMN não oferece recursos para a modelagem de objetivos, no entanto, isso não é considerado um problema neste trabalho. Porém, isso talvez justifique a ausência de regras de negócio na notação, que é um conceito importante já que podem atuar diretamente no contexto dos objetivos da organização.

Outro ponto importante é que a BPMN, em relação à extensão da UML para processos de negócio, é adotada pela OMG, o que garantirá evolução contínua de acordo com as necessidades reais do mercado e maior garantia de longevidade. No contexto da extensibilidade, ambos permitem a inserção de novas definições.

A outra linguagem analisada foi a UCM, que integra a URN, juntamente com a GRL. A UCM é uma linguagem para modelagem de cenários que são usados para descrever processos de negócios ou fluxos de trabalho.

Apesar da sua importância dentro da URN representando os processos (descritos como cenários) relacionados aos objetivos, no contexto específico da modelagem de processos de negócio ela torna-se menos adequada, seja pela ausência de representação de elementos do negócio como pelo seu padrão gráfico de modelagem.

Portanto entende-se que a UCM é importante no contexto da URN quando trata de rastreabilidade e alinhamento entre objetivos e processos, mas para representar modelos de negócio de forma abrangente, ela não é a melhor opção. Outros autores corroboram com esse pensamento como [Pourshahid09] e [Sikandar03].

Como última linguagem analisada, temos o EPC que foi desenvolvido em 1992 pelo grupo do professor Scheer, na Universidade de Saarland. Assim como a BPMN, o EPC possui vasta capacidade de representação de elementos de processos de negócio e considerável aceitação do mercado. Um exemplo de importância da linguagem é que os sistemas ERP da SAP utilizam os diagramas EPC para documentar os processos do sistema SAP R/3.

O EPC é integrante de um dos *frameworks* mais avançados no contexto da modelagem de processo: o *framework* ARIS. Esse *framework* inspirou o desenvolvimento da ferramenta mais avançada no contexto de processos de negócio conhecida como ARIS (*Architecture of Integrated Information Systems*), da empresa Software AG. Obviamente a ferramenta ARIS também oferece o modelo EPC, porém muitas outras soluções foram desenvolvidas contendo o diagrama, por exemplo, ARIS *Express*, ADONIS, Mavim *Rules*, *Business Process Visual ARCHITECT*, Microsoft Visio, Semtalk, Bonapart e *Oryx*.

Apesar de ser uma linguagem sólida e reconhecida, não se tornou um padrão internacional como a BPMN, principalmente por ter sido desenvolvido em um contexto *proprietário* e ser o principal modelo de uma das soluções mais complexas e de alto custo do mercado, não atingindo as empresas de pequeno porte e usuários simples.

Baseado nestas considerações, escolhemos a BPMN como notação para a modelagem de processos de negócio neste trabalho porque consideramos sua capacidade de modelagem de processos de negócio suficiente e com elementos gráficos amigáveis, além de ser um padrão reconhecido e adotado por um grupo internacional com constantes atualizações e estar consolidado entre os usuários e desenvolvedores de ferramentas.

2.7.2.Domínio de objetivos

No domínio de objetivos, de acordo com a sequência, apresentamos a extensão da linguagem UML [Eriksson&Penker00], *framework* i*, GRL e modelo de objetivos do *framework* ARIS.

Assim como os elementos de modelagem de processos de negócio são propostos através da extensão da UML por [Eriksson&Penker00], o mesmo trabalho inclui a extensão de elementos para a modelagem de objetivos. A UML oferece uma abordagem parecida com a do *framework* ARIS que permite o relacionamento dos objetivos com os processos e a decomposição de objetivos em um diagrama específico, entretanto, é a única linguagem (entre as analisadas) que oferece o conceito de "problema".

O *framework* i* é o precursor da modelagem intencional distribuída e no registro do raciocínio (*reasoning*) para o alcance dos objetivos. A modelagem intencional é voltada para objetivos, definidos como metas e metas-flexíveis. A modelagem tradicional de processos utiliza objetivos, mas de uma maneira pouco profunda o que acarreta uma série de problemas.

O i* permite uma análise aprofundada sobre os objetivos organizacionais e como os objetivos provocam impactos uns nos outros e nas tarefas executadas pela organização, uma vez que o i* lida com a intencionalidade de forma mais completa, através da dependência estratégica entre os atores [10], evidenciando maior nível de informações e detalhamento o que contribui positivamente para a transparência do negócio.

Além disso, é um dos frameworks mais referenciados e discutidos da atualidade, servindo de inspiração para novos métodos, tais como GRL (*Goal*-Oriented Requirements Engineering) e TROPOS [[Bresciani04], [Castro02]], ERi*c [Oliveira08c] e extensões, tais como, Diagrama de IP [Oliveira08a], SD Situation [Oliveira08a], SR Construct [Oliveira08b] e Diagnósticos i* [Oliveira08b]. Isso demonstra a importância acadêmica do i*, principalmente como precursor da abordagem intencional.

Conforme visto, a GRL compõe a URN, que é um padrão internacional aprovado pela União Internacional de Telecomunicação (ITU) e é composta pelos elementos disponíveis no i*, incluindo extensões que ampliam a capacidade da linguagem tanto para a modelagem de agentes (em sistemas multiagentes) como maior capacidade semântica de relacionamentos, já que a GRL aborda especialmente requisitos não funcionais.

Em uma visão completamente distinta de modelagem de objetivos (mais próxima da UML), encontra-se o modelo de objetivos do *framework* ARIS. Consideramos este

modelo como o mais simples, principalmente no que tange a capacidade semântica dos relacionamentos. Além de não diferenciar meta de meta-flexível e de não possuir relacionamentos de contribuição, oferece apenas um tipo de relacionamento entre objetivos (*belongs to*) e outro entre objetivos e processos (*supports*). Além disso, não permite a extensão da linguagem, ou seja, do *framework* ARIS, para inclusão de novos elementos. Entretanto, é a única linguagem a explorar o conceito de FCS (Fator Crítico de Sucesso) [[Rockart78], [Grunert&Ellegard92]].

Duas linguagens se destacam entre as linguagens analisadas pela capacidade representativa dos elementos e semântica dos relacionamentos: i* e GRL. Na escolha entre as duas linguagens, optamos pelo i*, uma vez que a capacidade da GRL é justificada pelo uso do i* e, além disso, acreditamos que o i* possui maior foco acadêmico.

Sabemos que existem muitas outras linguagens além das analisadas neste trabalho, por exemplo, somente entre as linguagens discutidas por trabalhos que utilizamos como referências [[Pourshahid09], [List&Korherr06]], temos: IDEF, Petri Nets, YAWL e EEML, entretanto, descartamos estas linguagens porque entendemos serem menos expressivas no contexto da modelagem de processos do que as outras. Além disso, os trabalhos de [List&Korherr06] e [Pourshahid09] já demonstram que estas linguagens não oferecem mais benefícios do que as linguagens abordadas.

Esta subseção apresentou uma avaliação das linguagens de modelagem de processos e objetivos abordados neste capítulo e justificou a escolha da BPMN e do *framework* i* para o trabalho de integração das linguagens. Cabe salientar que ao reutilizar os conceitos dessas linguagens, a capacidade e restrições referentes à modelagem de processos de negócio e objetivos serão as mesmas das linguagens BPMN e i*, respectivamente. Entretanto, neste trabalho não atentamos a possíveis imperfeições das linguagens, mas apenas focamos na comparação de suas qualidades nos quesitos que consideramos importantes.

A Tabela 11 e Tabela 12 sintetizam de forma comparativa, com notas de 0 a 4, sendo o menor valor representando ausência da capacidade descrita, e o maior valor representando a máxima pontuação de satisfação. Os valores foram definidos baseados na experiência prévia e conhecimento adquirido durante o estudo.

O quesito *Interface* refere-se à capacidade representativa dos elementos gráficos; *Representação* refere-se à capacidade de englobar as diversas definições dos elementos envolvidos no respectivo domínio; *Popularidade* refere-se às frentes de pesquisa existentes, artigos, citações, porém, principalmente à percepção de volume de usuários utilizando a linguagem; *Integração* refere-se à presença de elementos que relacionam objetivos e processos de negócio; *Padronização* refere-se à forma como a

linguagem foi projetada, propósito, autores e uso como linguagem padrão dentro de seu domínio; *Ferramentas* referem-se ao número de softwares disponíveis que oferecem a modelagem da respectiva linguagem; *Evolução* refere-se à projeção de atualização e continuidade da linguagem.

Tabela 11 – Comparação entre as linguagens de modelagem de processos

	Diagrama de processos (UML)	BPMN	UCM (URN)	EPC (ARIS)
Interface	2	4	1	4
Representação	2	3	1	4
Popularidade	1	4	2	3
Integração	2	0	3	0
Padronização	1	4	4	4
Ferramentas	1	4	1	2
Evolução	2	4	3	4

Tabela 12 – Comparação entre as linguagens de modelagem de objetivos

	Diagrama de objetivos (UML)	I*	GRL (URN)	Diagrama de objetivos (ARIS)	NFR
Interface	3	4	4	3	4
Representação	2	4	4	2	3
Popularidade	2	3	3	2	3
Integração	2	0	3	0	0
Padronização	1	1	2	2	2
Ferramentas	1	2	1	1	2
Evolução	2	4	3	2	4

O próximo capítulo apresenta a integração das duas linguagens, os novos elementos, relacionamentos e visões propostos.