

Engenharia de Requisitos de Software

Jordana S. Salamon

jssalamon@inf.ufes.br

jordanasalamon@gmail.com

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Modelagem de Objetivos e de Casos de Uso

- ▶ A análise de requisitos é um processo que envolve a construção de diversos modelos.
- ▶ É importante construir, primeiro, um modelo que descreva o sistema em termos de suas tarefas, seu ambiente e como sistema e ambiente estão relacionados.
- ▶ Tal modelo deve ser passível de compreensão tanto por desenvolvedores - analistas, projetistas, programadores e testadores - como pela comunidade usuária - clientes e usuários.
- ▶ Modelos de casos de uso (use cases) são uma forma de estruturar essa visão.

Modelagem de Objetivos e de Casos de Uso

- ▶ Como o próprio nome sugere, um caso de uso é uma maneira de usar o sistema. Usuários interagem com o sistema, interagindo com seus casos de uso. Tomados em conjunto, os casos de uso de um sistema definem a sua funcionalidade. Casos de uso são, portanto, os “itens” que o desenvolvedor negocia com seus clientes.
- ▶ Como se pode perceber, casos de uso têm uma estreita relação com requisitos funcionais e podem ser derivados diretamente deles. Assim, quanto mais completa e correta for a identificação de requisitos funcionais, melhor será o modelo de casos de uso resultante.
- ▶ Obviamente, novos requisitos funcionais podem ser identificados quando se elabora um modelo de casos de uso, já que o processo é essencialmente iterativo. Contudo, se for possível utilizar outros mecanismos para apoiar a identificação de requisitos funcionais, tanto melhor.

Modelagem de Objetivos e de Casos de Uso

- ▶ Neste contexto, a análise de objetivos pode ser uma importante aliada.
- ▶ Objetivos tanto proveem razões para os requisitos quanto direcionam a identificação de requisitos para suportá-los. Assim, antes de se realizar a modelagem de casos de uso é interessante fazer uma análise de objetivos.
- ▶ Vale destacar que a análise de objetivos é uma abordagem que ainda não alcançou plenamente a indústria de software, sendo considerada, portanto, parte ainda do estado da arte.

Análise de Objetivos

- ▶ Como vimos anteriormente, o trabalho de ER pode ser estruturado em termos de três dimensões/questões: Por quê? O que? E quem?
- ▶ A análise de objetivos focaliza o porquê, tendo objetivo como abstração central.
- ▶ Objetivos são declarações de intenções ou desejos de interessados a serem alcançados. Objetivos são declarações prescritivas, assim como requisitos.
- ▶ Portanto, objetivos são declarativos, ao contrário dos procedimentos operacionais para atingi-los, que são capturados nos modelos de casos de uso. A satisfação de objetivos requer a cooperação de diversos agentes, dentre eles pessoas, dispositivos, produtos de software existentes e o produto de software a ser desenvolvido (software to-be).



nemo

Análise de Objetivos

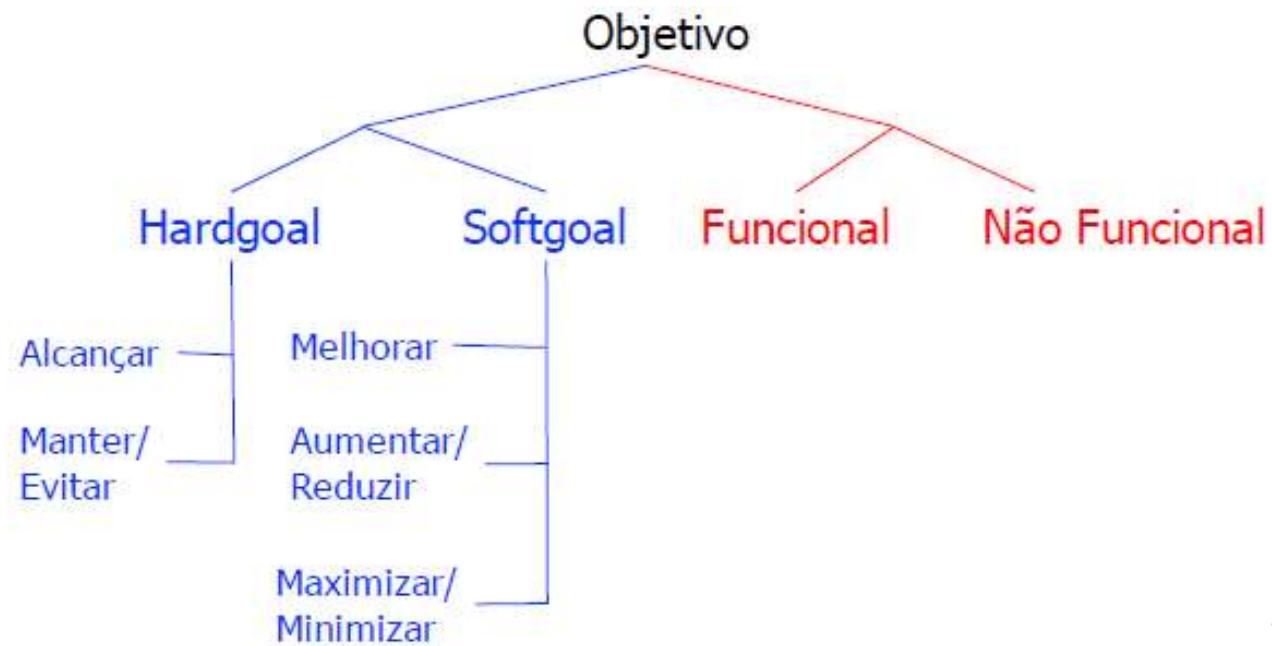
- ▶ Objetivos podem ser expressos em diferentes níveis de abstração/granularidade.
- ▶ Em níveis mais altos, há os objetivos mais gerais declarando metas estratégicas relacionadas ao negócio ou à organização.
- ▶ Nos níveis intermediários, há os objetivos táticos.
- ▶ Nos níveis mais baixos, há os objetivos mais específicos tratando de metas técnicas relacionadas ao sistema, ou seja, os requisitos para o sistema.
- ▶ Essa diferença entre níveis e granularidade de objetivos sugere um mecanismo de estruturação de objetivos baseado em ligações de contribuição (contribution links) entre objetivos.
- ▶ Um objetivo de nível mais alto pode ser refinado em objetivos de granularidade mais fina, de modo que se possa partir de metas organizacionais em direção a requisitos do software a ser desenvolvido.

Análise de Objetivos

- ▶ Essa noção de granularidade de objetivos nos permite deixar mais claras algumas noções importantes da ER.
- ▶ Quanto mais fina a granularidade de um objetivo, menos agentes são necessários para satisfazê-lo. Um requisito de software pode ser visto como um objetivo sob responsabilidade de um único agente: o software a ser desenvolvido (software to-be).
- ▶ Por outro lado, uma expectativa é um objetivo sob responsabilidade do ambiente do software to-be. Ao contrário de requisitos, expectativas não podem ser garantidas pelo software to-be.
- ▶ Em suma, um objetivo pode corresponder a um requisito de software ou não, dependendo dos agentes envolvidos em sua satisfação.

Análise de Objetivos

- ▶ Em GORE, objetivos são tipicamente classificados ao longo de duas dimensões principais, ortogonais entre si:



Análise de Objetivos

- ▶ A primeira dimensão refere-se ao fato de ser possível, ou não, estabelecer claramente um critério de satisfação para o objetivo.
- ▶ Hardgoals possuem um critério de satisfação claramente estabelecido. Um hardgoal é uma proposição que é objetivamente satisfeita por um estado de coisas. Assim, é possível falar em satisfação de um hardgoal.
- ▶ Um softgoal, por outro lado, é uma expressão vaga de uma intenção para a qual não se consegue estabelecer claramente um critério de satisfação. Ele prescreve preferências em relação a comportamentos alternativos do sistema e, portanto, ele é “mais satisfeito” por certas alternativas e menos por outras.
- ▶ Assim, não faz muito sentido falar em “satisfação do objetivo”, pois, na verdade, um softgoal é “mais satisfeito” por uma alternativa do que por outra.

Análise de Objetivos

- ▶ Hardgoals podem ser objetivos relacionados a alcançar ou manter situações. Um hardgoal relacionado a alcançar uma situação prescreve comportamentos pretendidos onde se busca alcançar tal situação.
- ▶ Quando se deseja destacar o tipo do objetivo, pode-se preceder o nome do objetivo com a palavra “Alcançar”. Assim, um objetivo dessa natureza pode ser escrito da seguinte forma: [Alcançar] Situação Alvo.
- ▶ Esse tipo de objetivo indica que, dada a situação atual, mais cedo ou mais tarde a situação alvo deve ser alcançada.
- ▶ P.ex., o objetivo “Alcançar Requisição de Exemplar Satisfeita” (ou simplesmente “Requisição de Exemplar Satisfeita”) indica que, se um exemplar de um livro for requisitado (situação atual), então mais cedo ou mais tarde um exemplar do livro será emprestado para o usuário (situação alvo).

Análise de Objetivos

- ▶ Um hardgoal relacionado a manter uma situação prescreve comportamentos pretendidos que buscam manter tal situação. Um objetivo dessa natureza pode ser escrito da seguinte forma: [Manter] Situação.
- ▶ P.ex., o objetivo “Manter Classificação de Livros Correta” (ou simplesmente “Classificação de Livros Correta”) indica que, se um livro é registrado corretamente na biblioteca, então sua correta classificação deve ser mantida.
- ▶ Softgoals, por sua vez, podem ser relacionados a melhorar uma certa situação, aumentar/reduzir quantidades, ou maximizar/minimizar certos aspectos.
- ▶ Ex.: O trabalho dos bibliotecários na classificação de livros deve ser reduzido.

Análise de Objetivos

- ▶ A segunda dimensão refere-se à distinção entre objetivos funcionais e não funcionais (a mesma distinção já discutida para requisitos de software).
- ▶ Objetivos/requisitos funcionais referem-se a funções que se pretende que o sistema a ser construído proveja. Objetivos/ requisitos não funcionais referem-se a qualidades ou restrições sobre o desenvolvimento ou provisão de um serviço.
- ▶ Essas duas dimensões proveem meios independentes de classificação de objetivos, ou seja, são classificações ortogonais. Na prática, contudo, certos objetivos de uma dimensão são mais frequentemente classificados como sendo de um tipo específico da outra.
- ▶ Por exemplo, frequentemente objetivos funcionais são hardgoals relacionados a alcançar uma situação. Por outro lado, objetivos relacionados à usabilidade e manutenibilidade tendem a ser softgoals.

Modelando Objetivos usando a linguagem iStar

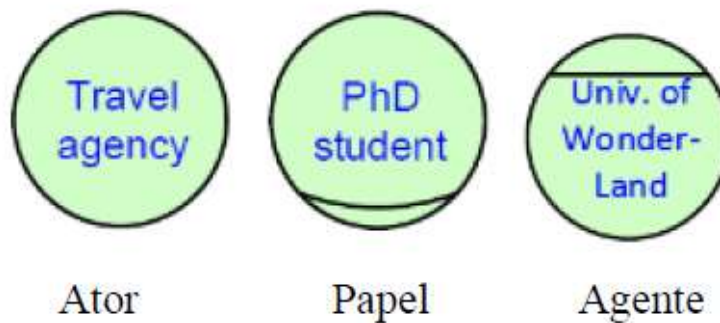
Atores e Ligações entre Atores

- ▶ Organizações são entidades sociais e sua operação depende da efetiva interação entre um número de atores.
- ▶ Um ator (actor) é uma entidade ativa e autônoma que visa alcançar seus objetivos exercendo seu know-how, em colaboração com outros atores.
- ▶ Há dois subtipos de atores: Agente e Papel.
- ▶ Um agente (agent) é um ator concreto, com manifestação física, tal como uma pessoa, uma organização ou um departamento.
- ▶ Um papel (role) é uma caracterização abstrata de um comportamento de um ator social dentro de um contexto ou iniciativa de esforço específico.

Modelando Objetivos usando a linguagem iStar

Atores e Ligações entre Atores

- ▶ O guia da linguagem provê as seguintes diretrizes:
 - ▶ • Use agente quando for possível identificar um indivíduo concreto.
 - ▶ • Use papel quando se deseja caracterizar uma classe abstrata.
 - ▶ • Use ator nos demais casos.



Modelando Objetivos usando a linguagem iStar

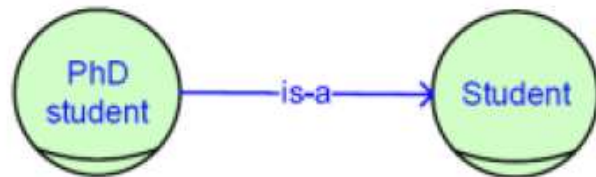
Atores e Ligações entre Atores

- ▶ Há dois tipos de ligação entre atores: is-a e participates-in.
- ▶ Is-a representa a noção de generalização/especialização e pode ser aplicado entre papéis (Papel a Papel) ou entre atores (Ator a Ator).
- ▶ Participates-in representa qualquer outro tipo de associação entre dois atores diferente de is-a. Participates-in tem diferentes significados dependendo dos elementos sendo ligados:
 - ▶ • Agente - Papel: representa que o agente desempenha (plays) o papel.
 - ▶ • Ligando elementos do mesmo tipo: representa uma relação parte-de (part-of).

Modelando Objetivos usando a linguagem iStar

Atores e Ligações entre Atores

► Exemplos:



PhD student é um subtipo de Student.



Mike White desempenha o papel de Phd student.



Univ. trip mgmt IS é parte de Univ. of Wonderland.

Modelando Objetivos usando a linguagem iStar

Elementos Intencionais

- ▶ iStar 2.0 enfoca intenções, i.e., coisas que atores querem. Há quatro tipos de elementos intencionais (intentional elements):
 - ▶ • **Objetivo** (Goal): é um estado de coisas (state of affairs) que um ator deseja alcançar e que tem um critério claro de satisfação. Ex.: ter artigo publicado, ter voos reservados.
 - ▶ • **Qualidade** (Quality): é um atributo para o qual um ator deseja algum nível de satisfação. Sendo atributos, qualidades estão sempre relacionadas a uma entidade. Qualidades guiam a busca por meios de se alcançar objetivos. Ex.: Reserva (de uma viagem) rápida.
 - ▶ • **Tarefa** (Task): representa ações que um ator quer que sejam executadas, usualmente dentro do contexto de se alcançar um objetivo. Ex.: Pagar pelos bilhetes comprados.
 - ▶ • **Recurso** (Resource): é uma entidade física ou de informação requerida por um ator para realizar uma tarefa. Ex.: Cartão de Crédito.

Modelando Objetivos usando a linguagem iStar

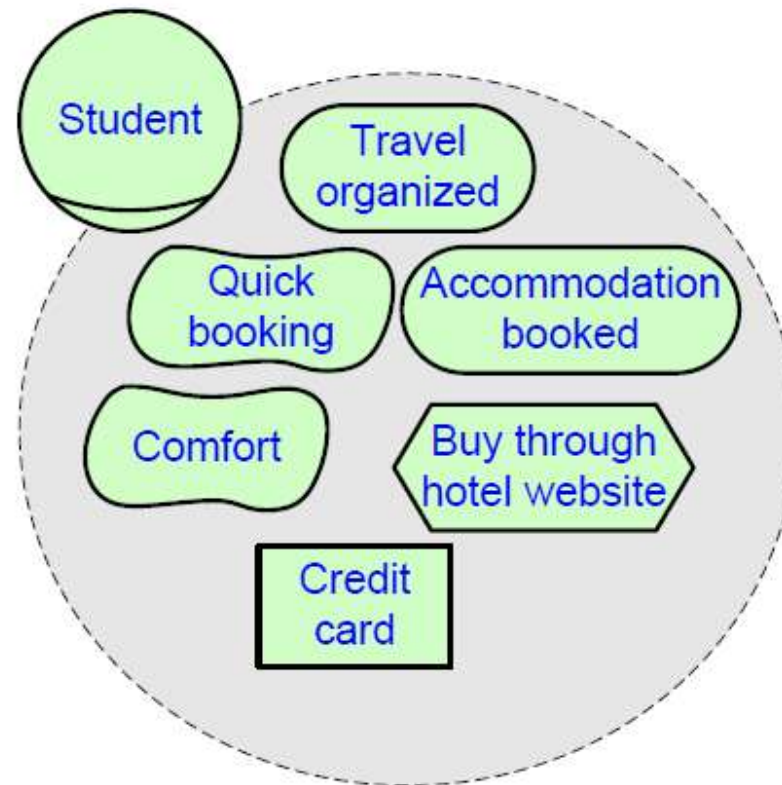
Elementos Intencionais



Modelando Objetivos usando a linguagem iStar

Elementos Intencionais

- ▶ Elementos intencionais aparecem dentro da fronteira do ator (actor boundary), representando a perspectiva do ator no modelo:



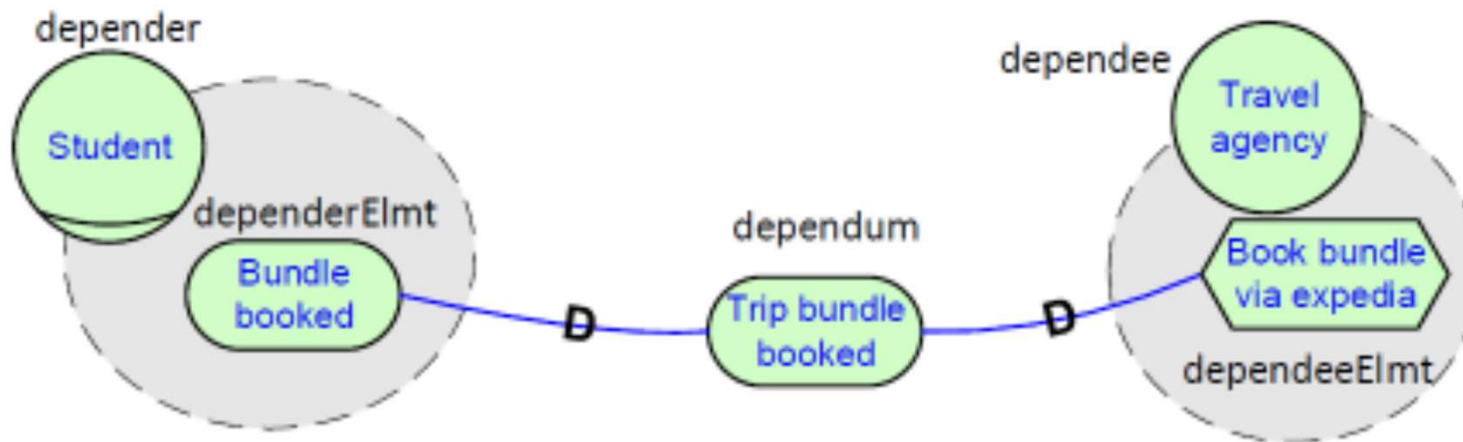
Modelando Objetivos usando a linguagem iStar

Dependências

- ▶ Relações sociais são representadas como dependências. Uma dependência é um relacionamento com cinco argumentos:
 - ▶ • Depender: ator que depende de alguma coisa (o Dependum) a ser provida;
 - ▶ • DependerElmt: elemento intencional dentro da fronteira do Depender onde a dependência inicia e que explica porque a dependência existe;
 - ▶ • Dependum: elemento intencional que é o objeto da dependência;
 - ▶ • Dependee: o ator que deve prover o dependum;
 - ▶ • DependeeElmt: o elemento intencional que explica como o dependee pretende atender o dependum.

Modelando Objetivos usando a linguagem iStar

Dependências



Modelando Objetivos usando a linguagem iStar

Dependências

- ▶ O tipo do dependum especializa a semântica da relação de dependência.
- ▶ Se o dependum é um objetivo, então o dependee é livre para escolher como atingir esse objetivo.
- ▶ Se o dependum é uma qualidade, o dependee é livre para escolher como satisfazer suficientemente a qualidade.
- ▶ Se o dependum é uma tarefa, espera-se que o dependee execute a tarefa da forma prescrita.
- ▶ Se o dependum é um recurso, espera-se que o dependee disponibilize o recurso para o depender.
- ▶ Assim, diferentes tipos de dependum dão ao dependee diferentes graus de liberdade.

Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais

- Os elementos intencionais dentro da fronteira de um ator são inter-relacionados. Em *iStar*, há quatro tipos de ligações entre elementos intencionais:

		Arrowhead pointing to			
		<i>Goal</i>	<i>Quality</i>	<i>Task</i>	<i>Resource</i>
Link starts from	<i>Goal</i>	Refinement	Contribution	Refinement	n/a
	<i>Quality</i>	Qualification	Contribution	Qualification	Qualification
	<i>Task</i>	Refinement	Contribution	Refinement	n/a
	<i>Resource</i>	n/a	Contribution	NeededBy	n/a

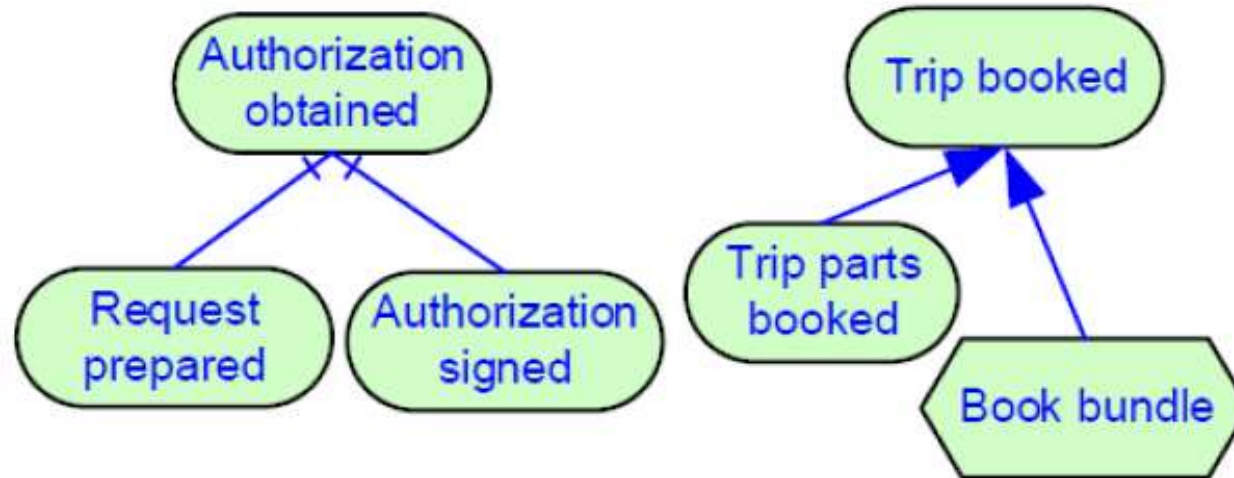
Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais

- ▶ Refinamento (Refinement) é um relacionamento genérico que liga objetivos e tarefas hierarquicamente.
- ▶ Há dois tipos de refinamento:
- ▶ **AND** - a realização de todos os n filhos ($n \geq 2$) torna o pai realizado;
- ▶ **Inclusive OR** - a realização de pelo menos um dos filhos torna o pai realizado.

Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais



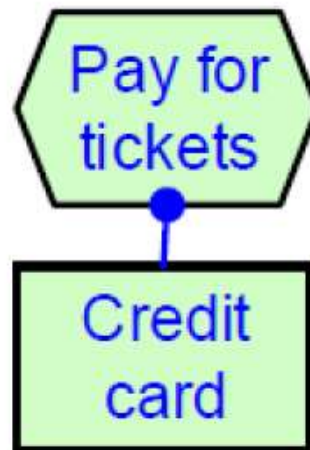
AND-Refinement

OR-Refinement

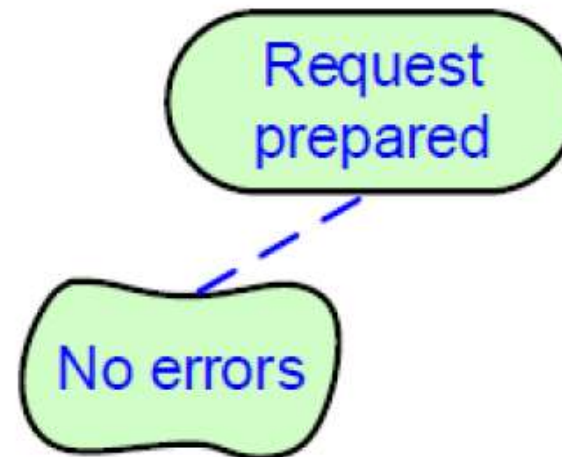
Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais

- ▶ A relação Requerido por (NeededBy) liga uma tarefa a um recurso e indica que o ator precisa do recurso para executar a tarefa. Essa relação, contudo, não detalha a razão para tal necessidade (consumo, leitura etc.).
- ▶ A ligação de Qualificação (Qualification) relaciona uma qualidade a uma tarefa, objetivo ou recurso.



Requerido por



Qualificação

Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais

- ▶ Por fim, ligações de contribuição (contribution links) representam os efeitos de elementos intencionais sobre qualidades.
- ▶ Qualidades podem ser cumpridas (ou satisfeitas), quando há evidências positivas suficiente, ou negadas, quando há evidências negativas.
- ▶ Sendo assim, essas ligações são qualitativas e são usadas para apoiar os analistas na tomada de decisão relativa a objetivos/tarefas alternativos.
- ▶ Note que as ligações de contribuição nada falam sobre como o cumprimento/negação é calculado.

Modelando Objetivos usando a linguagem iStar

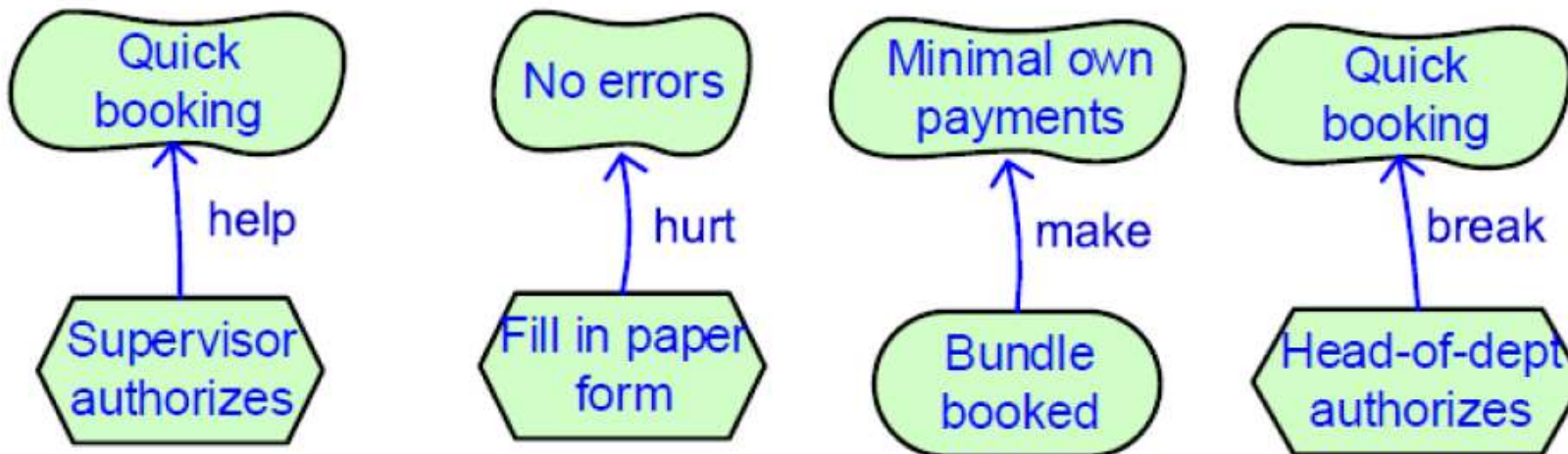
Ligações entre Elementos Intencionais

- ▶ Há quatro tipos de ligações de contribuição:
 - ▶ • **Make**: há evidência positiva suficiente para satisfazer o elemento alvo;
 - ▶ • **Help**: há evidência positiva fraca para satisfazer o elemento alvo;
 - ▶ • **Hurt**: há evidência fraca contra a satisfação (ou seja, para a negação) do elemento alvo;
 - ▶ • **Break**: há evidência suficiente contra a satisfação (ou para a negação) do elemento alvo.

Modelando Objetivos usando a linguagem iStar

Ligações entre Elementos Intencionais

- ▶ Há quatro tipos de ligações de contribuição:



Modelando Objetivos usando a linguagem iStar

Visões de Modelagem

- ▶ Quando usando iStar, o analista cria um modelo. Esse modelo pode ser visualizado via múltiplas perspectivas ou visões de modelo.
- ▶ As visões de modelo padrão de iStar são:
 - ▶ • Visão de Raciocínio estratégico (Strategic Rationale - SR): mostra todos os detalhes capturados no modelo: atores, ligações entre atores, elementos intencionais, dependências e ligações entre elementos intencionais.
 - ▶ • Visão de Dependência estratégica (Strategic Dependency - SD): mostra apenas a parte social do modelo, i.e., atores, ligações entre atores e dependências. Não mostra elementos intencionais e suas ligações.

That's all Folks!



nemo