

Método ERi*c - Engenharia de Requisitos Intencional

Antonio de Padua Albuquerque Oliveira^{1, 2}
Julio Cesar Sampaio do Prado Leite²
Luiz Marcio Cysneiros³

¹ UERJ – Universidade do Estado do Rio de Janeiro

² PUC-Rio – Pontificia Universidade Catolica do Rio de Janeiro

³ York University, Toronto

padua@ime.uerj.br; padua@inf.puc-rio.br
www.inf.puc-rio.br/~julio
cysneiro@yorku.ca

Abstract

*The aim of this paper is to present the Intentional Requirements Engineering Method - ERi*c, which is a contribution to Goal Oriented Requirements Engineering (GORE). ERi*c is based on the intentionality concept as applied by the i* Framework. Intentionality reflects organization actors' motivations and interests and is represented in the models by goals and softgoals. The singularity of the proposal consists in how to do the intentionality elicitation and how to model it in a way of mitigating i* models complexity. The ERi*c Method has six steps, which cover elicitation, modeling and analysis. The ERi*c Method is shown through "The Expert Committee System" (EC System) exemplar, a system to support the organization of a conference program.*

Resumo

*O objetivo deste artigo é apresentar o método Engenharia de Requisitos Intencional - ERi*c, o qual é uma contribuição à Engenharia de Requisitos Orientada a Metas (GORE). O Método ERi*c se baseia no conceito de intencionalidade como foi aplicado pelo Framework i*. Intencionalidade reflete as motivações e os interesses dos atores de uma organização e a intencionalidade é representada nos modelos através das metas. A singularidade da proposta está na elicitação da intencionalidade e como modelá-la de forma a reduzir a complexidade dos modelos i*. O Método ERi*c é composto por seis etapas, as quais cobrem a elicitação a modelagem e a análise. O Método ERi*c está apresentado através do exemplo do problema Expert Committee, um sistema para apoiar a organização de conferências.*

Introdução - Visão Geral do Método

Nesta seção mostramos uma visão geral do Método ERi*c [Oliveira 08a] e, depois, nas seções seguintes apresentamos as etapas com os objetivos desejados, as atividades e os artefatos (documentos e diagramas) preparados para alcançarmos os objetivos pretendidos.

A Figura 1 apresenta o esquema simplificado do método. O Método ERi*c está dividido em seis etapas. Na primeira etapa ele faz a identificação dos atores e a elicitação das metas, em seguida trabalha na identificação das situações de dependência estratégica, depois se ocupa com a modelagem das metas dos atores, na quarta etapa se preocupa com a modelagem da racionalização das metas, na quinta etapa faz a especificação das situações de dependência estratégica e, concluindo, na sexta etapa executa a análise da qualidade dos modelos SD e SR.

Na parte central do diagrama mostramos o termo baseline de Leite e Oliveira [Leite 95] para indicar que o Método ERi*c deve ser apoiado por ferramentas de software para tratar da evolução do software, ou seja ter instrumentos de gerência de configuração que apoiem as mudanças. As ferramentas também têm a missão de facilitar o trabalho do engenheiro de requisitos no processo e também garantir, de modo automatizado, a rastreabilidade dos requisitos [Ramesh 01].

A idéia é que o Método ERi*c contribua para melhorar a qualidade final dos requisitos. Melhor qualidade final dos requisitos significa: modelos de requisitos mais completos, que os requisitos possam ser apreendidos com mais facilidade e que, na representação, os requisitos estejam mais bem organizados e, conseqüentemente, que os modelos estejam menos sobrecarregados.

1. Elicitar as Metas dos Atores

Objetivo: "Elicitação e refinamento das metas para Sistemas Multi-Agentes" (SMA).

A primeira etapa é composta de três atividades: A) Preparar LAL - Léxico Ampliado da Linguagem, B)

Definir AGFL - Metas dos Atores Vindas do Léxico e C)
Refinar as Metas dos Atores.

1.A) Preparar LAL

Para a construção do LAL o engenheiro de requisitos primeiramente identifica as fontes de informação que podem ser utilizadas. Devem ser considerados todos os documentos disponíveis como também todas as pessoas que podem fornecer informações sobre o UdI – Universo de Informação. O engenheiro de requisitos deve seguir os seguintes passos recomendados por Leite et al. [Leite 00]: (i) identificar a lista de símbolos relevantes que são as palavras ou frases peculiares e mais usadas; (ii) classificar os símbolos como: sujeito, objeto, verbo, e estado; (iii) descrever os símbolos através da noção e do impacto; (iv) verificar o LAL através de inspeção; e (v) validar o LAL com os atores do UdI.

Durante os passos da atividade, siga as instruções que aparecem nas regras gerais fornecidas para a definição do nome, da noção e do impacto e obedeça aos princípios da circularidade e do vocabulário mínimo.

Para identificar a qual classe o símbolo pertence, em [Leite 00] são sugeridas as seguintes heurísticas: se o símbolo pratica uma ação, é classificado como sujeito; se é quem sofre a ação, é classificado como objeto; se é uma situação em um dado momento, é classificado como estado; se representa uma ação, é classificado como verbo.

1.B) Definir AGFL - Metas dos Atores vindas do Léxico

No LAL os atores são descritos como sujeitos. Atores também aparecem nas noções e nos impactos que descrevem os símbolos. Impactos mencionam as ações que acontecem na organização. Uma ação concreta muda um estado; muda o estado para outro estado diferente e uma ação flexível adiciona um atributo de qualidade ao estado.

O termo ação concreta foi apresentado pela primeira vez por Sá Carvalho em [Carvalho 88]. Para a tese de Oliveira [Oliveira 08a] ele foi adaptado para ter um sentido mais amplo. O termo ação flexível foi criado como uma complementação de ação concreta, desse modo, sempre que existir uma ação: ela é concreta ou flexível. Oliveira [Oliveira 08a] qualificou o termo flexível em ação flexível com o mesmo significado ou interpretação que o usado para metas flexíveis (“softgoals”). Ações flexíveis são ações pouco precisas, para as quais não se pode a priori identificar um resultado concreto e, além disso, a confirmação da execução da ação flexível pode depender de interpretação. Uma ação flexível (exemplos de “ações flexíveis”: analisar, apurar, avaliar, agilizar, conferir, controlar, gerenciar, verificar, validar, etc.) é diferente de uma ação concreta (exemplos de ações concretas: comprar, pagar, planejar).

Como ações mudam estados: então o ponto-chave da atividade é identificar a motivação (o porquê) que está por

trás de cada ação. (1) Quando a ação muda um estado (ação concreta) → a ação vai definir uma meta concreta. (2) Quando a ação fornece uma “qualidade” a um estado (ação flexível) → a ação vai definir uma meta flexível.

Três sub-atividades são requeridas neste ponto do método: Identificar os atores, Extrair as metas dos atores a partir dos impactos dos símbolos e Refinar as metas.

1.B.1) Identificar os atores

A idéia da atividade é identificar os atores, sejam eles especializados em agentes, desempenhando papéis ou mesmo cobrindo posições Leite et al. [Leite 07].

Os atores são classificados como sujeitos no LAL porque eles praticam ações. Os símbolos classificados como objetos sofrem ações praticadas por atores. Os símbolos do tipo sujeito e os praticantes das ações sobre os objetos são candidatos naturais a atores.

1.B.2) Extrair as metas a partir dos símbolos

Para capturar as metas usar o “template” específico para cada tipo de ação definida no impacto do símbolo do LAL. Veja os exemplos: Template A (caso de símbolo tipo sujeito no qual o impacto menciona uma ação concreta), Template B (caso de símbolo tipo objeto que sofre ação concreta) e Template C (caso de ação flexível). Cada impacto irá gerar uma meta em um formato pré-estabelecido pelos templates. Para definir a meta o engenheiro de requisitos deve responder uma pergunta simples: POR QUÊ? (“WHY?”).

Os impactos mencionam as ações que acontecem em uma organização e as ações mudam estados. As mudanças de estado são de duas modalidades: ações mudam um estado para outro estado, ou ações adicionam um atributo de qualidade a um estado.

Nas subseções a seguir ilustramos a definição das metas para todos os tipos de símbolos (sujeito, objeto, verbo e estado). Estes exemplos foram extraídos do estudo de caso EC – Expert Committee [Deloach 01], os quais foram apresentados em [Oliveira 08a].

1.B.2.a) Definir metas de símbolos tipo sujeito

As ações concretas provenientes de símbolos do tipo sujeito (Figura 2) podem refletir duas possibilidades da motivação de um ator: (1) um estado que o ator deseja e não dependa de outro ator; (2) um estado que o ator deseja e dependa de outro ator.

Se a resposta ao “por quê” definir uma meta que deve ser alcançada com uma dependência para outro ator então, devido a essa dependência, deve ser definida mais uma meta, a meta do outro ator (caso da meta reflexiva). Nesse caso a resposta da análise do impacto indica que o primeiro ator (“o dependente” – aquele que depende) depende ou necessita ajuda de outro ator (“o dependente” – daquele que se depende) para que a meta dele seja alcançada. Esse tipo de ocorrência indica que uma “meta reflexiva” precisa ser definida. Essa “corrente” de metas concretas deve ser elicitada até que apareça uma meta sem a dependência de um segundo ator.

- ⊕ Instruções: Para cada símbolo do tipo sujeito:
- ↻ Para cada impacto que mencionar uma ação flexível: definir uma meta flexível com o Template C(2). Porque uma meta flexível está associada (por contribuição) a pelo menos uma meta concreta, nossa recomendação nesse caso é para que a meta seja definida depois, quando o engenheiro for definir as metas provenientes de símbolos do tipo verbo.
- ↻ Para cada impacto que mencionar uma ação concreta: definir uma meta concreta (Template A) com quatro ou cinco componentes (ator, objeto, auxiliar, verbo e [ator]).
- ⊕ Heurísticas gerais: Sugestões para a definição das metas:

(1) na sentença de definição da meta, sempre use o nome principal do símbolo, não utilize os sinônimos do símbolo.
 (2) crie metas com sentenças simples e diretas. Por exemplo: ao invés de “processo de revisão seja executado” deve ser escrito “artigos sejam revisados”, porque a comunicação direta é mais fácil de ser compreendida.

No exemplo, para o símbolo pesquisador, a ação (submeter) no primeiro impacto foi considerada como concreta e a ação (participar) no segundo impacto foi considerada como flexível. O primeiro impacto nos levou a identificar duas metas concretas enquanto o segundo impacto será trabalhado mais adiante no método.

No primeiro impacto: Pesquisador submete artigo porque pesquisador deseja que artigo seja publicado por chair (pela conferência). Chair representa a conferência nesse relacionamento de cooperação. A segunda meta é devido ao relacionamento “reflexivo” entre os dois atores e está explicada no parágrafo a seguir.

1.B.2.b) Definir metas reflexivas

As metas reflexivas refletem a interação e a colaboração entre dois atores.

- ⊕ Instruções: Para cada meta concreta que teve indicado um segundo ator:
- ↻ Definir uma nova meta concreta que complemente a dependência entre os atores.
 - 1) Continuar até que na definição da meta não mais exista um segundo ator ou quando surgir uma dependência mútua.
 - 2) Criar a meta reflexiva com base nos passos anteriores de 1 até 5, como são elaboradas para símbolos do tipo sujeito. No exemplo anterior: pesquisador deseja que artigo seja publicado por chair e chair deseja que artigo seja submetido por pesquisador.

1.B.2.c) Definir metas de símbolos tipo objeto

Cada impacto que menciona uma ação sofrida por um símbolo do tipo objeto (Figura 3) indica um estado que algum ator deseja atingir.

- ⊕ Instruções: Para cada símbolo do tipo objeto:

- ↻ Para cada impacto que mencionar uma ação flexível: definir uma meta flexível com o Template C(2). Por que uma meta flexível está associada (por contribuição) a pelo menos uma meta concreta. Nossa recomendação nesse caso é para que a meta seja definida depois, quando o engenheiro for definir as metas provenientes de símbolos do tipo verbo.
- ↻ Para cada impacto que mencionar uma ação concreta: definir uma meta concreta (Template B) com quatro componentes (objeto, auxiliar, verbo e ator). O template usado para símbolos do tipo objeto é diferente do template usado para símbolos do tipo sujeito porque objetos sofrem as ações.

No exemplo, para o símbolo conferência, a ação (participar) no primeiro impacto foi considerada como flexível e a ação (submeter) no segundo impacto foi considerada como concreta. O segundo impacto nos levou a identificar uma meta concreta enquanto o primeiro impacto será trabalhado mais adiante no método.

1.B.2.d) Definir metas a partir de símbolos tipo verbo

Cada impacto que menciona uma ação referenciada por um símbolo do tipo verbo (Figura 4) indica um estado que algum ator deseja atingir.

- ⊕ Instruções: Para cada símbolo do tipo verbo:
- ↻ Para cada impacto que mencionar uma ação concreta: definir uma meta (Template C(1)) com quatro componentes (objeto, auxiliar, verbo e ator). Ou seja, aplicar as heurísticas do passo (b.3).
- ↻ Para cada impacto que mencionar uma ação flexível gerar uma meta flexível (Template C(1)), esta deve ser definida com três atributos (tipo, tópico e meta concreta associada).

No exemplo (Template C(1)) para o símbolo rever artigo a ação (executar) no primeiro impacto foi considerada como concreta e a ação (deve obedecer) no segundo impacto foi considerada como flexível. O primeiro impacto nos levou a identificar uma meta concreta enquanto o segundo impacto a uma meta flexível. A revisão dos artigos deve obedecer ao deadline para que não aconteça um atraso na apuração do resultado das revisões. A restrição gerou a meta flexível sem atraso [revisão]. Essa meta flexível foi associada à meta concreta do revisor “artigo seja revisado”.

Como pode ser observado no Template C(2), depois que a maioria das metas concretas foram elicitadas, as metas flexíveis provenientes de símbolos tipo sujeito, tipo objeto, tipo verbo e tipo estado podem ser elicitadas. Este ponto é considerado mais adequado porque as metas flexíveis precisam ser associadas a metas concretas. Então, metas concretas devem ser descobertas primeiro.

1.B.2.e) Definir metas a partir de símbolos tipo estado

Cada impacto que mencionar uma ação provocada por um símbolo do tipo estado (Figura 5) indica um estado que algum ator deseja atingir.

⊕ Instruções: Para cada símbolo do tipo estado:

✎ Para cada impacto que gerar uma meta concreta ou uma meta flexível, o engenheiro de requisitos escolhe o melhor das duas alternativas:

O Template C(3) mostra que um símbolo tipo estado também pode gerar uma meta flexível como uma meta concreta, depende da natureza da ação tratada no impacto do símbolo. No caso a ação (precisa receber uma outra avaliação) foi considerada flexível devido à necessidade de melhor qualidade na revisão de artigos, quando existe um conflito entre duas revisões.

1.C) Refinar as metas dos atores

A idéia da atividade é agrupar as metas (concretas e flexíveis) por ator e organizá-las em ordem cronológica.

⊕ Instruções:

✎ Converter as metas tipo objeto em metas tipo sujeito.

✎ Como o ator de cada meta oriunda de um símbolo do tipo objeto ficou no final do template, a tarefa é apenas remanejar o ator para o início da meta. A Figura 6 mostra o resultado da operação sobre a meta apresentada no Template B.

1) Grupar as metas por ator.

2) Ordenar temporalmente as metas.

✎ Iniciar com as metas que representem os primeiros estados alcançados ou modificados e depois colocar as metas avançando temporalmente.

✎ Colocar as metas de longo prazo no final. Metas de longo prazo apontam para resultados gerais ou resultados de sobrevivência do negócio.

3) Excluir as metas redundantes e/ou repetidas.

A Figura 7 mostra como exemplo o resultado final da atividade de refinar as metas para o ator autor. No exemplo aparecem 5 metas concretas e uma meta flexível, está mostrado também que duas metas do autor dependem de outros atores.

2. Identificar as SDSituations

Objetivo: “Refinamento e organização das metas de SMAs”.

A segunda etapa, Identificar as Situações de Dependência Estratégica, é composta por três atividades: A) Distinguir SDSituations, B) Reconhecer as interdependências entre as SDSituations e C) Construir o diagrama de SDSituations.

Uma SDSituation é um bloco de elementos de dependência com uma intencionalidade situacional compartilhada pelos atores, é formada pelos atores e pelas dependências estratégicas mútuas [Oliveira 08a].

2.A) Distinguir SDSituations

Arranjar as metas por situações de dependência estratégica. Os atores participam através das metas próprias de uma bem definida situação de colaboração chamada: uma “Situação de Dependência Estratégica”.

A atividade trabalha com a finalidade de perceber como as metas compõem conjuntos para formar situações de negócio. Sabemos que atores possuem metas, porém, as metas são fortemente ligadas ou associadas entre si de modo a concluir uma situação de negócio, uma situação estratégica de cooperação entre os atores, uma SDSituation.

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

A Tabela 1 mostra as sintaxes para a descrição dos elementos da SDSituation e as funções dos atributos na SDSituation. O nome de uma SDSituation deve ser escrito com a sintaxe: < **substantivo da situação + objeto** >, por exemplo: “Indicação de Revisores”.

2.B) Reconhecer Interdependências entre SDSituations

Observe cada SDSituation e reconheça situações em que há dependência lógica, temporal ou seqüencial com outra situação. Isso significa que você deve identificar duas situações distintas, mas uma depende da outra no processo do negócio. Reconheça também se existe algum paralelismo entre as SDSituations.

2. C) Construir Diagrama de SDSituations

Represente todas as SDSituations em um único diagrama [Oliveira 06b]. Mostre as interdependências entre as SDSituations e represente o encadeamento das SDSituations. Coloque a representação do fator tempo no diagrama quando for importante.

3. Modelar as Metas dos Atores

Objetivo: “Modelagem e avaliação das metas”

A terceira etapa é composta de duas atividades: A) Identificar Agentes, Posições e Papéis e B) Criar os Painéis de Intencionalidade.

Para o entendimento das atividades, como fizemos na seção anterior, foi apresentado em [Oliveira 07] a conceituação necessária para a elaboração dos Painéis de Intencionalidade com a representação das relações entre as metas.

A principal motivação da criação do Diagrama IP é a representação da intencionalidade em um único e homogêneo diagrama. A intencionalidade elicitada através das metas dá origem aos requisitos do sistema. O Diagrama IP (Figura 8) retrata para os atores apenas as metas e as relações de transição entre as mesmas, podendo ser considerado uma redução do Modelo SR. O Diagrama IP, além de não representar as tarefas e os recursos que aparecem no Modelo SR, adota também uma

simplificação em relação ao diagrama de estados tradicional porque a notação da transição de estados foi suprimida pela adoção de uma convenção que somente retrata três tipos de transição (a correlação, a contribuição e a dependência). As relações entre as metas são semanticamente as mesmas adotadas pelo Modelo SR [Yu 95]; veja a contextualização na seção a seguir.

3.A) Identificar Agentes, Posições e Papéis

Como em cada SDSituation atores trabalham e colaboram mutuamente para a resolução da meta da situação identificada, atores podem assumir posições e podem desempenhar papéis que são pertinentes a cada SDSituation. Por serem situações guiadas pela intencionalidade comum, as SDSituations agregam as metas dos atores que são dedicadas à resolução da meta comum aos envolvidos na situação de dependência estratégica. Devido ao fato que, com a abordagem de SDSituations, o Engenheiro de Requisitos lida com um número reduzido de metas por vez e que, além disso, elas são restritas à situação, ele pode perceber com mais clareza quando os atores têm algum comportamento que pode ser classificado como especialização do ator, como agente, papel ou posição.

Utilizando as metas concretas elicitadas para cada ator observe, e tente perceber, se existe algum papel ou alguma posição que pode ser identificada. Metas que são exclusivas de um ator e que não podem ser alcançadas por ninguém mais podem indicar um papel desempenhado pelo ator. Um ou mais papéis podem estar associados a um cargo coberto (posição) pelo agente na organização.

Enquanto os papéis geralmente são restritos a uma única SDSituation, uma posição coberta por um agente pode aparecer ou se repetir em mais de uma SDSituation.

3.B) Criar os Painéis de Intencionalidade

Prepare um diagrama para cada SDSituation. Dê preferência a colocar como vizinhos no diagrama, ou o mais próximo possível quando existir mais de dois atores, atores que tenham o maior número de relações de dependência. Decida sobre a ordenação das metas no eixo do ator, as metas iniciais devem ficar na parte inferior do diagrama. Metas concretas e metas flexíveis que necessitem aguardar algum outro estado ser alcançado devem ser colocadas na parte superior do eixo do ator. Coloque primeiramente apenas as metas concretas e em segundo lugar as metas flexíveis. O exemplo de Diagrama IP na Figura 8 mostra como é feito o detalhamento de uma SDSituation.

Trabalhe com um ator de cada vez. Primeiro represente as relações entre as metas flexíveis (a contribuição de uma meta flexível para outra meta flexível), depois as relações entre as metas flexíveis e as metas concretas (a correlação de uma meta flexível para uma meta concreta e a correlação de uma meta concreta para uma meta flexível).

Por último represente as relações de dependência entre as metas concretas dos atores, inicie a representação pela parte inferior do diagrama. Faça uma revisão, se metas de atores diferentes forem equivalentes e atingidas simultaneamente, então represente a relação de equivalência entre elas.

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

4. Modelar a Racionalização das Metas dos Atores

Objetivo: Modelagem do “rationale” dos atores.

A quarta etapa é composta de duas atividades: A) Construir Modelos SD e B) Construir Modelos SR.

4.A) Construir Modelos SD

Para a construção dos Modelos SD o engenheiro de requisitos necessita utilizar artefatos que foram preparados nas fases anteriores do Método Eri*c. São necessários o Diagrama de SDSituations e os Diagramas IP (Painéis de Intencionalidade) de cada SDSituation identificada.

4.A.1) Definir as dependências estratégicas

A idéia da atividade é definir as dependências estratégicas entre os atores usando os mesmos critérios de [Yu 95]. Para cada SDSituation do Diagrama de SDSituations, o engenheiro de requisitos deve observar as relações de dependência que foram assinaladas no Diagrama IP, e para cada uma delas definir entre os quatro tipos de dependência possíveis (por meta concreta, por meta flexível, por recurso e por tarefa) qual tipo de dependência estratégica é o mais indicado e vantajoso para o “dependee”.

Cada tipo de dependência tem um grau de liberdade diferente do “dependee” na relação de colaboração entre os atores. Na dependência por recurso o grau de liberdade do “dependee” é nulo, pois ele deve disponibilizar o recurso exatamente como é esperado pelo “dependee”. Nas duas dependências por meta o “dependee” fica mais vulnerável, pois não tem influência sobre o cumprimento da meta pelo “dependee”. Na dependência por meta flexível o “dependee” avalia e aceita, se achar conveniente, que a meta flexível foi razoavelmente satisfeita (“satisfied”). Na dependência por tarefa o “dependee” executa uma tarefa seguindo as orientações definidas pelo “dependee”. Na dependência por tarefa o “dependee” deve ter alguma motivação (ou compensação) por executar a tarefa da maneira que o “dependee” deseja e essa motivação deve ser mostrada no Modelo SD, na SDSituation atual, caso mais comum de acontecer, ou em outra SDSituation.

Para todos os tipos de dependências deve existir uma motivação para o cumprimento da dependência estratégica pelo “dependee”, pois se trata de um modelo de cooperação guiado pela intencionalidade dos atores, porém, a motivação pode não se tornar explícita pelo processo de elicitacão.

4.A.2) Preparar um Modelo SD para cada SDsituation

A atividade é simples: fazer um Modelo de Dependências Estratégicas – Modelo SD para cada conjunto de atores que participam da SDsituation, usando as dependências estratégicas definidas na atividade anterior.

⚙ Heurísticas para derivar modelos SD:

1) Represente os atores aplicando as especializações percebidas na SDsituation.

👉 Um ator representado pelos papéis desempenhados e pelas posições ocupadas dá maior representatividade ao detalhamento do processo de elicitacão.

2) Modele as dependências entre os atores.

👉 Use o Modelo SD para expressar os relacionamentos intencionais. Use as “Regras gerais para definição de dependências estratégicas” da Tabela 1 para definir as dependências estratégicas.

👉 Represente no Modelo SD as metas (concretas e flexíveis), as tarefas e os recursos importantes para cada SDsituation. Use a padronização de nomes da Tabela 1. Adotamos como guia a mesma sintaxe usada por Zheng [Zheng 04] para elementos do i*.

4.B) Construir Modelos SR

Para a construção dos Modelos SR o engenheiro de requisitos necessita dos Modelos SD construídos na atividade anterior como também utilizar artefatos que foram preparados nas fases anteriores do Método ERI*c. São necessários o Diagrama de SDsituations e os Diagramas IP (Painéis de Intencionalidade) de cada SDsituation identificada.

⚙ Heurísticas para derivar modelos SR:

O processo de modelagem MAS - Multi-Agent Systems, deve lidar com três (3) tipos de interações adicionais. Existem três novas interações (Ator ↔ Agente, Agente ↔ Sistema, e Agente ↔ Agente) em comparação com a modelagem tradicional de software (Ator ↔ Sistema) [Jennings 01] [Silva 04]. Jennings [Jennings 01] chama de ambiente (“environment”) o conjunto de todos os agentes e recursos que são compartilhados entre os agentes e os atores. Para gerenciar e controlar a complexidade, e explorar o uso apropriado das propriedades de sistemas multi-agentes, recomendamos a elaboração de quatro (4) Modelos SR ao invés de modelar todas as interações em um único diagrama [Oliveira 06a]. A técnica está apresentada em detalhes nas subseções i, ii e iii.

MODELAR AS INTERAÇÕES: ATOR – ATOR

Para cada SDsituation criar um Modelo SR para expressar como a Situação de Dependência Estratégica é resolvida por cada ator participante. Represente dentro da linha limite de cada ator as metas concretas, as metas flexíveis, os recursos e as tarefas como elementos para cumprir a intencionalidade do ator. Conecte esses elementos usando as ligações intencionais (meios-fim, decomposição e contribuição). Veja exemplo na Figura 9.

👉 Represente as mesmas dependências entre dois atores que aparecem no Modelo SD da SDsituation também no Modelo SR conectando os atores; os elementos da dependência aparecem fora das linhas limites dos atores, mas agora, no Modelo SR, as ligações aparecem ligando elementos que estão dentro da linha limite de cada ator.

👉 Identifique as metas principais para cada ator, elas são as metas que aparecem no Diagrama IP no topo de cada eixo do ator. Um ator pode possuir mais de uma meta principal e freqüentemente os atores possuem uma ou mais metas intermediárias. O Diagrama IP fornece a hierarquia das metas, mas o engenheiro de requisitos deve representar a hierarquia no Modelo SR da maneira correta.

👉 Represente as tarefas em ligações meios-fim de modo a cumprir o atingimento das metas concretas. Soluções alternativas devem ser mapeadas também como tarefas meio, se existir mais de um meio de alcançar a meta concreta.

👉 Represente as tarefas adicionais, ou especializadas, tipo subtarefas, que devem ser ligadas às tarefas (meio) usando a ligação de decomposição.

👉 Identifique os recursos necessários às tarefas como também os recursos necessários para serem compartilhados por tarefas e subtarefas.

👉 Mapeie as metas flexíveis conectando-as às tarefas quando elas adicionarem qualidade à intencionalidade mapeada pela meta concreta, ou conectado-as às outras metas flexíveis, usando a ligação de contribuição.

👉 Para nomear os elementos do modelo, use a padronização da Tabela 1.

MODELAR AS INTERAÇÕES: ATOR – AGENTE

(Estes modelos fornecem a interação de delegação)

(1) Prepare um Modelo SR para o relacionamento entre o ator e o agente de software que irá representar o ator no ambiente (“environment”). Esse diagrama é chamado de Modelo da Delegação [Oliveira 06a].

👉 Modele o relacionamento intencional entre o ator e o agente do ator. Precisam ser decididas quais responsabilidades o ator deseja delegar ao agente de software.

👉 Adapte as mesmas regras mostradas em (i) para esse caso de interação. Para cada meta concreta (na qual o

agente irá apoiar o ator), inclua uma meta concreta similar à do ator incluindo a palavra “apóia” no nome da meta concreta do agente ou inicie com o verbo “apoiar” para as tarefas do agente.

✎ Crie tarefas para a interação entre eles.

MODELAR AS INTERAÇÕES: AGENTE – AGENTE

(Representa as interações entre os agentes no ambiente “environment”).

(1) Modele o relacionamento intencional entre os agentes. Use o Modelo SR para expressar o relacionamento intencional. Prepare um modelo para cada relacionamento entre agentes que representarão os atores no ambiente.

✎ Modele o relacionamento intencional entre os agentes.

✎ Adapte as mesmas regras mostradas em (i) para este caso.

5. Especificar as SDSituations

Objetivo: Especificação das Situações de Dependência Estratégica.

A quinta etapa é composta por apenas uma atividade:

A) Descrever as Situações de Dependência Estratégica.

5.A) Descrever as SDSituations

Para a descrição de SDSituations através de Cenários, o engenheiro de requisitos necessita utilizar artefatos que foram preparados nas fases anteriores do Método ERi*c. São necessários: o Diagrama de SDSituations e os Modelos SR de cada SDSituation.

A atividade descreve SDSituations aplicando a Técnica de Cenários. A construção de Cenários deve seguir uma determinada técnica e dar ênfase aos conceitos de Sistemas Multi-Agentes (propriedades de agência). A ferramenta de software C&L, que é uma ferramenta de gerenciamento de léxicos e cenários, apóia essa atividade. O C&L foi desenvolvido pelo grupo de engenharia de requisitos da PUC-Rio e está disponível em [C&L – PUC-Rio].

✎ Heurísticas para Descrever SDSituations:

Para a descrição use o formato pré-definido com os elementos do “template” do cenário.

✎ Maximize o uso de símbolos do LAL na descrição de cada SDSituation.

✎ Dê ênfase aos elementos que lidem com as propriedades de agência como autonomia, pró-atividade, sociabilidade, adaptação e interação como também colaboração, aprendizado e mobilidade. É importante mostrar o que foi considerado na SDSituation quando descrever o cenário: (1) Atores em episódios recebem atenção especial porque eles participam da propriedade de interação; (2) O nome do objetivo do cenário deve obedecer as regras de descrição de elementos do i*, veja Tabela 1; (3) Os

episódios que não tenham relevância para lidar com propriedades de MAS devem ser grupados dentro de um episódio de mais alto nível (“Chair prepares proposals”). (4) Atributos de qualidade devem ser representados com a mesma sintaxe de metas flexíveis, veja Tabela 1.

✎ Colocar os atores envolvidos de maneira explícita facilita a visualização das dependências entre os atores.

6. Analisar a Racionalização das Metas dos Atores

Objetivo: Verificação dos modelos i* para a melhoria da qualidade.

A sexta etapa é composta de três atividades: A) Diagnosticar SDSituations, B) Diagnosticar SRconstructs e C) Preparar Quadro de Metas x Problemas.

Nosso processo de diagnósticos focaliza na prevenção de defeitos ao invés de detecção, considerando tanto a eficiência como a eficácia do desenvolvimento de Sistemas Multi-Agentes.

Para cada modelo SD e SR deve ser aplicado os templates de perguntas [Oliveira 08b]. Dada a estrutura básica das SDSituations e dos SRconstructs questões são formuladas com palavras em negrito as quais são “lacunas” para serem substituídas pelos nomes reais dos elementos que aparecem nos modelos SD e SR.

Após as duas atividades de diagnósticos, dentro da atividade “Preparar Quadro de Metas x Problemas”, a sub-atividade de integrar as respostas consiste em casar os problemas potenciais, para que estes possam ser analisados. Durante essa sub-atividade o engenheiro de requisitos deve re-classificar as respostas que tratam de um mesmo problema. Neste ponto também duplicações de diagnósticos que são naturais confirmações de um mesmo problema devem ser eliminadas. O objetivo geral do framework de diagnósticos é revirar compostos de uma SDSituation com os SRconstructs correlatos procurando por problemas, omissões, deficiências e também potenciais melhorias. A estratégia aqui é integrar as duas modalidades de diagnósticos. “Diagnoses are important to deeply understand the problem before looking for the solution” [Oliveira 08a].

6.A) Diagnosticar SDSituations

Considerando as SDSituations como unidades básicas nós preparamos os diagnósticos dos modelos SD. O objetivo do diagnóstico é identificar quais são as faltas e deficiências do modelo e dessa maneira poder encontrar recomendações de mudanças nas dependências atuais, nas ligações entre SDSituations e também no “rationale” dos atores de modo que as metas do sistema desejado (“system-to-be”) sejam alcançadas. Fazendo diagnósticos com base nas SDSituations do sistema atual (“system-as-it-

is”) nós podemos encontrar SDSituations do sistema desejado.

Uma SDSituation é formada por elementos de dependência, entretanto a finalidade de diagnosticar uma SDSituation é descobrir quais dependências são “problemáticas” e assim analisar se existe algum outro ator que poderia ser melhor colaborador naquela SDSituation.

6.B) Diagnosticar SRconstructs

Para diagnosticar SRconstructs a primeira sub-atividade é Identificar Construtos. Esta consiste em partir os diagramas i^* em construtos, partir o problema em pequenas peças que permitem seguir uma análise detalhada de cada parte do diagrama. A sub-atividade Aplicar Questionamento consiste em aplicar o framework de perguntas para cada construto por que cada SRconstruct precisa passar pelo processo de investigação de problemas como está estabelecido pelo framework de diagnósticos [Oliveira 08b].

Um SRconstruct possui um fim (end) a ser alcançado por uma ou mais tarefas que são meios (means). Por sua vez, as tarefas têm agregados elementos que são necessários para o alcance da meta pela tarefa. A finalidade de diagnosticar SRconstructs é descobrir quais componentes são “problemáticos” assim como descobrir se existe alguma outra maneira ou alternativa que poderia ser usada para o alcance de uma determinada meta.

6.C) Preparar Quadro de Metas x Problemas

Outro resultado que pode ser conseguido a partir dos diagnósticos é o quadro de Metas X Problemas [Oliveira 08a]. A estratégia de diagnósticos indica como resultado problemas potenciais ou deficiências as quais são colocadas na coluna número um do quadro e as metas afetadas pelos problemas que são identificadas nos SRconstructs e SDSituations são assinalados nas demais colunas do quadro. O quadro Metas x Problemas tem a propriedade de concisão, ele mostra a influência dos problemas sobre as metas.

Conclusão

A finalidade do artigo foi apresentar o método Engenharia de Requisitos Intencional - ERI^{*}c, que tem a intencionalidade dos atores de uma organização como norte. Para que esse objetivo fique completo precisamos identificar os proprietários das metas (concretas e flexíveis) isto é, os atores da organização. Apesar de existirem dificuldades para se descobrir papéis e posições ocupados, concordamos que a simples identificação de atores é na maior parte das vezes fácil de ser concluída, uma vez que os atores são mencionados diretamente em documentos e entrevistas. O núcleo do problema é: como elicitar e expressar a intencionalidade. No artigo apresentamos uma técnica para elicitar a intencionalidade dos atores sociais, de maneira sistemática, antes de modelar essa intencionalidade.

Nossa proposição pode ser considerada um processo indireto apoiado por questões [Potts 94] sobre o Léxico Ampliado da Linguagem – LAL [Leite 93]. Queremos chamar a atenção de que dentro de cada símbolo do LAL (veja os exemplos nas Figuras 2, 3, 4 e 5) o elemento impacto tem a finalidade de representar as ações: as que acontecem, as que são reflexos de alguma ação, as que são aplicadas e também aquelas que podem ocorrer.

Nossa idéia é simples: ações mudam estados e estados são metas. “A goal is a condition or state of affairs in the world that an actor would like to achieve” [Yu 95]. De modo a descobrir os estados desejados que uma ação quer mudar nós precisamos perguntar Por que a ação ocorre?

Em conseqüência, na nossa abordagem, as metas possuem uma sintaxe estruturada. A meta concreta deve seguir a forma: **objeto + seja(m)lesteja(m) + verbo na passiva**. E a meta flexível: **tipo + [tópico]**, seguindo a nomenclatura do “NFR Framework de Chung [Chung 00], onde tipo é um atributo de qualidade e tópico é um símbolo do léxico da linguagem. Nos templates adotados, nós fixamos algumas partes para facilitar o trabalho do Engenheiro de Requisitos. As metas devem ser descritas utilizando os símbolos do LAL e também utilizando os verbos (que denotam estados na voz passiva) do domínio da aplicação, os quais são normalmente repetitivos. A intenção dos “templates” é também facilitar o trabalho do engenheiro de requisitos ao expressar as metas na sintaxe adequada.

O método foi alvo de experimentação [Oliveira 08a] e os resultados alcançados reforçam o fato de que o método proposto preenche uma lacuna na literatura, isto é, a ausência de descrições de como tratar da elicitação da intencionalidade e de como modelá-la de forma a evitar a complexidade dos modelos, característica da modelagem do i^* tradicional [Pastor 06].

Oliveira, em [Oliveira 08a], faz uma comparação dos pontos comuns do Método ERI^{*}c com os métodos GBRAM (Goal Based Requirements Analysis Method) [Anton 97] e KAOS (Knowledge Acquisition in autOated Specification) [Dardenne 93]. Trabalhos mais recentes como RiSD [Grau 04] que fornece um guia para a construção de Modelos SD com heurísticas para a redução da complexidade e REUSE [Maiden 04] que trata problemas de situações de dependência com base na interação homem-máquina e na psicologia cognitiva, são importantes artigos com alguma similaridade com a nossa abordagem.

A contribuição fundamental desse trabalho pode ser resumida em três partes: um processo bem delineado de construção, com uma técnica para a elicitação de metas e uma estratégia de modelagem que possibilita a modularização dos modelos SD e SR do Framework i^* .

Anexo: Figuras e Templates Utilizados

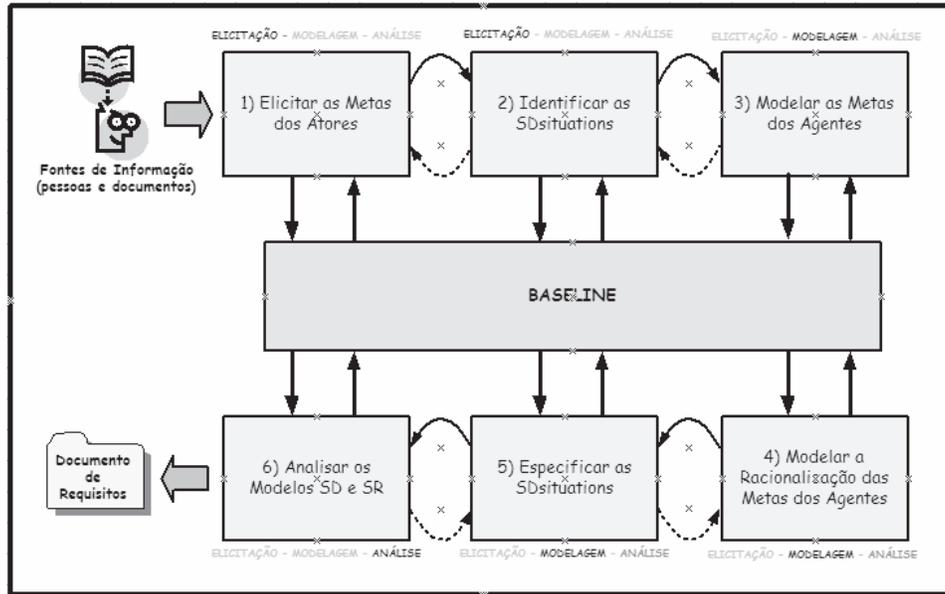


Figura 1 – Visão geral do método ilustrando o encadeamento das etapas

Nome:	pesquisador
Noção:	Pessoas que estudam e propoem trabalhos ou artigos sobre uma area de interesse da conferencia .
Classificação:	sujeito
Impacto(s):	- submetem artigos . - participam da conferencia .
Sinônimo(s):	pesquisadores.

Figura 2 – Exemplo de um símbolo do LAL do tipo sujeito

TIPO: SUJEITO	<meta concreta>			ATOR	
-- impacto	resposta ao por quê?	<sujeito / objeto LAL>	seja / esteja	<verbo>	<sujeito LAL>
PESQUISADORES					
-- submetem artigos.					
	Porque pesquisador deseja que	artigo	seja	publicado	por chair
	Porque chair deseja que	artigo	seja	submetido	por pesquisador
-- participam da conferência.		ação flexível			

Template A – Usando as ações concretas de símbolos do tipo sujeito

Nome:	conferencia
Noção:	Encontro periódico de pesquisadores experts em uma área de pesquisa.
Classificação:	objeto
Impacto(s):	- pesquisadores participam da conferência. - autores submetem artigos .
Sinônimo(s):	simpósio, congresso.

Figura 3 – Exemplo de um símbolo do LAL do tipo objeto

TIPO: OBJETO	<meta concreta>			ATOR	
-- impacto	resposta ao por quê?	<sujeito / objeto LAL>	seja/ esteja	<verbo>	<sujeito LAL>
CONFERÊNCIA					
-- pesquisadores participam da conferência.		ação-flexível			
-- autores submetem artigos.					
	Para que	artigos	sejam	revisos	por revisor

Template B – Usando as ações concretas de símbolos do tipo objeto

Nome:	rever artigo
Noção:	Trabalho de identificar erros e sugerir correções em artigos submetidos a conferencia .
Classificação:	verbo
Impacto(s):	- é executada por revisores . - deve ser obedecido o deadline de revisão .
Sinônimo(s):	revisões, revisão.

Figura 4 – Exemplo de um símbolo do LAL do tipo verbo

TIPO: VERBO	<meta flexível>			
-- impacto	<TIPO> atributo de qualidade	[TOPICO] sujeito/objeto LAL	<meta concreta associada>	<ator>
resposta ao por quê?				
REVER ARTIGO				
-- é executada por revisores .	ação concreta		artigo seja revisado	revisor
-- deve ser obedecido o deadline de revisão .				
Porque	sem atraso	[revisão]	artigo seja revisado	revisor

Template C(1) – Usando símbolos do tipo verbo

	<meta flexível>			
-- impacto	<TIPO> atributo de qualidade	[TOPICO] sujeito/objeto LAL	<meta concreta associada>	<ator>
resposta ao por quê?				
PESQUISADORES				
-- participam da conferência.	ação flexível			
Para que	reconhecimento	[comitê]	conferência seja executada por	coordenador geral

Template C(2) – Ações flexíveis de símbolos tipo sujeito e tipo objeto

Nome:	conflito
Noção:	Estado de indeterminação sobre a aceitação de um artigo .
Classificação:	estado
Impacto(s):	- artigo precisa receber uma outra avaliação.
Sinônimo(s):	conflitos.

Figura 5 – Exemplo de um símbolo do LAL do tipo estado

TIPO: ESTADO	<meta flexível>			
-- impacto	<TIPO> atributo de qualidade	[TOPICO] sujeito/objeto LAL	<meta concreta associada>	<ator>
resposta ao por quê?				
CONFLITO				
-- artigo precisa receber uma outra avaliação.	ação flexível			
Para que	qualidade	[revisão]	conflito seja resolvido	por comitê

Template C(3) – Usando símbolos do tipo estado

DEPENDER					DEPENDEE
revisor					
Porque revisor deseja que	artigo	seja	revisado	por	revisor

Figura 6 – Resultado da conversão de meta tipo objeto em meta tipo sujeito

DEPENDER					DEPENDEE
autor					
justa [revisão]	artigo	seja	submetido		
	artigo	seja	revisado	por	revisor
	artigo	seja	aceito		chair
	camera-ready	seja	enviada		
	artigo	seja	publicado	por	chair

Figura 7 – Exemplo com metas concretas, metas flexíveis e dependências

Tabela 1 – Descrevendo os elementos das SDSituations

	META CONCRETA	META FLEXÍVEL	TAREFA	RECURSO
NOME < sintaxe >	Substantivo + seja + verbo na passiva	Atributo de qualidade [tópico]	Verbo + objeto	Substantivo
DEFINIÇÃO	Sentença simples descrevendo a meta concreta.	Sentença simples descrevendo a meta-flexível.	Sentença simples descrevendo a tarefa.	Sentença simples descrevendo o recurso.
CONSEQÜÊNCIAS	Sentença simples	Sentença simples	Sentença simples	Sentença simples

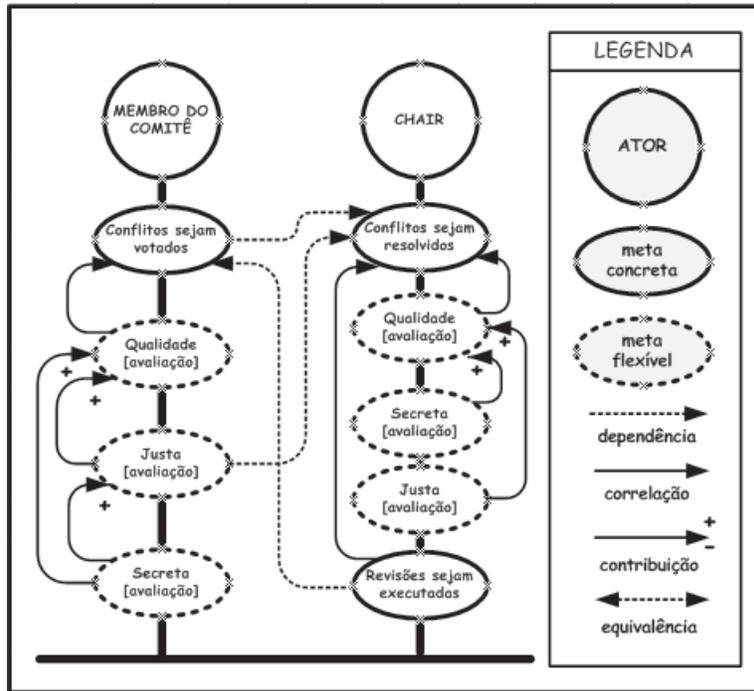


Figura 8 – Exemplo de Diagrama IP – SDSituation: VOTAÇÃO DE CONFLITOS

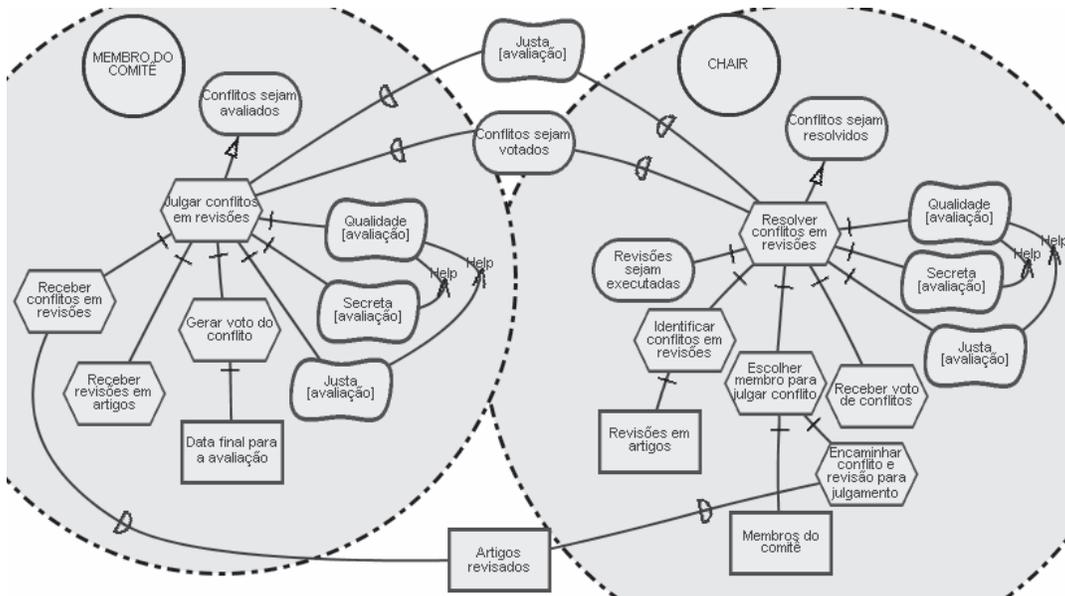


Figura 9 – SDSituation: Votação de Conflitos - Modelo SR

Referências Bibliográficas

- [Anton 97] Antón, A. I. Goal Identification and Refinement in the Specification of Software-Based Information Systems, Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, June 1997
- [Carvalho 88] Carvalho, Luiz C. de Sá; *Análise de Sistemas: o outro lado da informática*, LTC - Rio de Janeiro, 1988.
- [C&L – PUC-Rio] Cenários e Lexicos – PUC-Rio - Disponível em: <http://pes.inf.puc-rio.br/cel/>. Acesso: Outubro de 2007.
- [Chung 00] Chung, L.; Nixon, B.; Yu, E.; Mylopoulos, J.; *Non-Functional Requirements in Software Engineering* – Kluwer Academic Publishers 2000 – Massachusetts, USA.
- [Cunha 07] Cunha, Herbet “Uso de estratégias orientadas a metas para modelagem de requisitos de segurança”, *Dissertação (Mestrado em Informática) – PUC-Rio, 2007 - 145 f.*
- [Dardenne 93] Dardenne, A. van Lamsweerde and S. Fickas, “Goal-Directed Requirements Acquisition”, *Science of Computer Programming*, Vol. 20, 1993, 3-50.
- [Deloach 01] Deloach, S. et al.; *Multi-agent Systems Engineering*. International. In: *Journal of Software Engineering and Knowledge Engineering*, 11(3): 231—258 - 2001.
- [Grau 04] Grau, G; Franch, X; Mayol, E; Ayala, C; Cares, C; Haya, M; Navarrete, F; Botella, P; Quer, C; - *RiSD: A Methodology for Building i-Strategic Dependency Models*. SEKE 2005: 259-266
- [Jennings 01] Jennings, N., “An Agent-Based Approach for Building Complex Software Systems” *Communications of the ACM*, April 2001, Vol. 44 No 4.
- [Lamsweerde 01] Lamsweerde, Axel van; *Goal-Oriented Requirements Engineering: A Guided Tour*; *Proceedings RE'01, 5th IEEE International Symposium on Requirements Engineering*; Toronto; August 2001; 249-263.
- [Leite 93] Leite, Julio C. S. P.; Franco, Ana P. M.; *A Client Strategy for Conceptual Model Acquisition*; *Proceedings of the International Symposium on Requirements Engineering*, IEEE Computer Society Press, San Diego (1993), pp. 243-246.
- [Leite 95] Leite, Julio C. S. P.; Oliveira, A de Padua A.; *Client Oriented Requirements Baseline*, *Proceedings of the Second International Symposium on Requirements Engineering, RE95*, IEEE Computer Society Press, 1995, pp. 108-115.
- [Leite 00] Leite, J.C.S.P., Hadad, G., Doorn, J., Kaplan, G. A *Scenario Construction Process - RE Journal: Vol. 5, N. 1, Pags. 38 - 61, (2000), Springer-Verlag London Limited.*
- [Leite 07] Leite, Julio; Werneck, Vera; Oliveira, A. Padua A.; Capelli, Claudia; Cerqueira, Ana Luiza; Cunha, Herbert; Baixauli, Bruno; “Understanding the Strategic Actor Diagram: An Exercise of Meta Modeling” *The X Workshop on Requirements Engineering*; Toronto, Canada - May/2007.
- [Maiden 04] Maiden, N.A.M., & Jones, S.V., 2004, ‘*Dependability in RESCUE: A Concurrent Engineering Approach to the Specification of Requirements for Air Traffic Management*’, in *proceedings of the Dependability, Systems and Networks workshop on interdisciplinary approaches*, June 2004.
- [Oliveira 06a] Oliveira, A. Padua A., Cysneiros, L. M., Leite, J. C. do Prado, Figueiredo, E. M., and Lucena, C. J.; “Integrating scenarios, i*, and AspectT in the context of multi-agent systems”. In *Proceedings of the CASCON '06 (Toronto, Canada - Oct. /2006)*. ACM Press, New York, NY, 16.
- [Oliveira 06b] Oliveira, A. Padua A.; Cysneiros, L. M.; “Defining Strategic Dependency Situations in Requirements Elicitation”; *The IX Workshop on Requirements Engineering*; RJ, Brazil - July/2006.
- [Oliveira 07] Oliveira, A. Padua A.; Leite, J. C. S. P.; Cysneiros, L. M.; Cappelli, C.; “Eliciting Multi-Agents Systems Intentionality: From Language Extended Lexicon to i* Models”, *Proceedings of the XXVI International Conference of the Chilean Computer Science Society*. Los Alamitos: IEEE Computer Society Press, 2007. v. 16. p. 40-49.
- [Oliveira 08a] Oliveira, A. Padua A.; *Engenharia de Requisitos Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos*. PhD Tese, PUC-Rio, Março/2008.
- [Oliveira 08b] Oliveira, A. Padua A.; Leite, J. C. S. P.; Cysneiros, L. M.; Lucena, C. J.; *i* Diagnoses: A Quality Process for Building i* Models - Proceedings of the Forum at the CAiSE'08, Montpellier, France, June 18-20, 2008*. CEUR Workshop Proceedings 344 CEUR-WS.org 2008 pp. 9-12
- [Pastor 06] Pastor, O; Estrada, H; Martínez, A; *The Strengths and Weaknesses of the i* Framework: an experimental evaluation i*, its Applications, Variations and Extensions*. Eric Yu et als. (eds.) 2006 - MIT Press
- [Potts 94] Potts, Colin; Takahashi, Kenji; Antón, Annie I.; “Inquiry-Based Requirements Analysis”, *IEEE Software*, Volume 11, Issue 2 (March 1994), pp.: 21 – 32
- [Ramesh 01] Ramesh, B.; Jarke, M.; “Towards Reference Models for Requirements Traceability”, *IEEE TSE V. 21, N 1, 2001*
- [Silva 04] Silva, V.; Lucena, C. "From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language", In: Sycara, K., Wooldridge, M. (Edts.), *Journal of Autonomous Agents and MAS*, Kluwer Academic, 2004.
- [Wooldridge 02] Wooldridge, Michael; *An Introduction to Multi-Agent Systems*, John Wiley and Sons Limited: Chichester, 2002.
- [Yu 95] Yu, E. *Modelling Strategic Relationships for Process Reengineering*. PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada, 1995, pp. 124.
- [Zheng 04] Zheng You, *Using meta-model-driven views to address scalability in i* models*, Master of Science thesis, Graduate Department of Computer Science, University of Toronto, 2004, pp. 231