

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

ADRIANO DE OLIVEIRA CUNHA

CATALOGAÇÃO DO REQUISITO NÃO FUNCIONAL
TRANSPARÊNCIA

Rio de Janeiro
2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

INSTITUTO MILITAR DE ENGENHARIA

ADRIANO DE OLIVEIRA CUNHA

**CATALOGAÇÃO DO REQUISITO NÃO FUNCIONAL
TRANSPARÊNCIA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ricardo Choren Noya - D.Sc.

Co-orientador: Prof. Antonio de Pádua Albuquerque Oliveira - D.Sc.

Rio de Janeiro
2009

c2009

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do orientador.

C972 Cunha, A. O.
Catalogação do Requisito Não Funcional Transparência/ Adriano de Oliveira Cunha. – Rio de Janeiro: Instituto Militar de Engenharia, 2009.
90 p.:il.
Dissertação (mestrado) – Instituto Militar de Engenharia – Rio de Janeiro, 2009.

1. Engenharia de Software. 2. Engenharia de Requisitos. 3. Requisito Não Funcional. 4. Transparência. I. Título. II. Instituto Militar de Engenharia.

CDD 005.1

INSTITUTO MILITAR DE ENGENHARIA

ADRIANO DE OLIVEIRA CUNHA

**CATALOGAÇÃO DO REQUISITO NÃO FUNCIONAL
TRANSPARÊNCIA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ricardo Choren Noya - D.Sc.

Co-orientador: Prof. Antonio de Pádua Albuquerque Oliveira - D.Sc.

Aprovada em 22 de janeiro de 2009 pela seguinte Banca Examinadora:

Prof. Ricardo Choren Noya - D.Sc. do IME - Presidente

Prof. Antonio de Pádua Albuquerque Oliveira - D.Sc. da UERJ

Prof. Vera Maria Benjamim Werneck - D.Sc. da UERJ

Prof. Paulo Fernando Ferreira Rosa - Ph.D. do IME

Rio de Janeiro
2009

*Aos meus pais e irmão.
Obrigado pelo amor, carinho, orientação, incentivo e
paciência.*

AGRADECIMENTOS

À minha família, em especial a minha avó, pelo apoio incondicional ao longo da minha vida.

Aos amigos por compreenderem a necessidade de minha ausência durante o curso.

Aos CF(FN) Accioly e CF(FN) Cezar pelo incentivo e apoio fundamentais ao meu desempenho no curso.

Aos meus orientadores Ricardo Choren Noya e Antonio de Pádua Albuquerque Oliveira pelo apoio e incentivo na busca do meu melhor. Foram imprescindíveis ao sucesso do trabalho.

Aos professores Vera Maria Benjamim Werneck e Paulo Fernando Ferreira Rosa por terem contribuído com este trabalho, participando da banca examinadora e apontando melhorias pertinentes.

À todos os colegas de curso pela amizade e incentivo durante o curso.

À tripulação do IME, em especial aos membros da SE/8, pela fidalguia e apoio prestados durante o curso.

Ao Exército Brasileiro por mais uma vez contribuir com a minha formação.

Ao Corpo de Fuzileiros Navais da Marinha do Brasil por permitir a realização deste curso.

CT(FN) Adriano de Oliveira Cunha

“A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original”

Albert Einstein

SUMÁRIO

LISTA DE ILUSTRAÇÕES	9
LISTA DE TABELAS	11
LISTA DE ABREVIATURAS	12
1 INTRODUÇÃO	15
1.1 Objetivo	18
1.2 Contribuições	18
1.3 Organização do texto	19
2 CONCEITOS BÁSICOS	20
2.1 Transparência das Informações	20
2.2 NFR Framework	24
2.3 O Framework i - star	27
2.3.1 O Modelo SD - Strategic Dependency Model	27
2.3.2 O Modelo SR - Strategic Rationale Model	29
2.3.3 O Modelo SA - Strategic Actor Model	30
2.3.4 SDsituation - Strategic Dependency Situation	31
2.4 Conclusões	32
3 REQUISITOS DE TRANSPARÊNCIA	33
3.1 Conceitos sobre os Requisitos de Transparência	34
3.2 O Tipo Transparência	40
3.3 Princípios para incluir transparência das informações em sistemas	42
3.3.1 Princípio da Correta Identificação dos Atores	42
3.3.2 Princípio da Correta Classificação das Informações	43
3.3.3 Princípio da Correta Destinação das Informações	44
3.4 Notação para as metas flexíveis de transparência	46
3.5 Métodos de refinamento	46
3.5.1 Métodos de Decomposição por Tipo	46
3.5.2 Métodos de Decomposição por Tópico	49
3.5.3 Métodos de Operacionalização	54

3.5.4	Métodos de Argumentação	55
3.6	Conclusões	55
4	ESTUDO DE CASO	57
4.1	Sistema Expert Committee	57
4.2	Sistema Controle do Caixa do Restaurante	74
4.3	Conclusões sobre os Estudos de Casos	80
5	CONCLUSÃO	82
5.1	Contribuições	83
5.2	Trabalhos futuros	84
6	REFERÊNCIAS BIBLIOGRÁFICAS	85

LISTA DE ILUSTRAÇÕES

FIG.2.1	Exemplo de visões diferentes de uma conta.	23
FIG.2.2	Processo de transparência da informação.	23
FIG.2.3	Exemplo de elementos do SIG (CHUNG et al., 2000).	25
FIG.2.4	Exemplo de catálogo de Tipo de RNF (CHUNG et al., 2000).	26
FIG.2.5	Exemplo de catálogo de métodos de refinamento (CHUNG et al., 2000).	27
FIG.2.6	Tipos de dependências estratégicas.	28
FIG.2.7	Exemplo de modelo SR.	29
FIG.2.8	Escala de contribuições.	30
FIG.2.9	Exemplo de modelo SA.	30
FIG.2.10	Exemplo de SDsituation.	31
FIG.2.11	Exemplo de uma teia de SDsituations.	32
FIG.3.1	Triângulo da Transparência.	34
FIG.3.2	Combinações da taxonomia proposta para os atores.	36
FIG.3.3	Exemplo de um modelo SA para um banco.	38
FIG.3.4	Taxonomia indicada para as informações.	39
FIG.3.5	Catálogo do tipo.	40
FIG.3.6	Características das informações.	41
FIG.3.7	Subtipos da TD.	41
FIG.3.8	Subtipos da TP.	42
FIG.3.9	Cobertura das credenciais de acesso.	43
FIG.3.10	Métodos de refinamento da Transparência.	46
FIG.3.11	Exemplo de refinamento por tipo.	47
FIG.3.12	Exemplo de refinamento por tipo para a TD.	49
FIG.3.13	Exemplo de refinamento por tipo para a TP.	49
FIG.3.14	Exemplo de refinamento por subclasse no assunto.	50
FIG.3.15	Exemplo de refinamento por subclasse no destinatário.	51
FIG.3.16	Exemplo de refinamento por subclasse no processo.	52
FIG.3.17	Exemplo de refinamento por atributos individuais.	53
FIG.3.18	Catálogo de métodos de operacionalização.	54
FIG.3.19	Catálogo de métodos de argumentação.	55

FIG.3.20	Exemplo de argumentação.	56
FIG.4.1	Modelo SA para o ECS retirado de (OLIVEIRA, 2008).	58
FIG.4.2	Modelo SR para a SDsituation Submissão de Artigos.	60
FIG.4.3	Modelo SR para a SDsituation Aceitação de Propostas.	61
FIG.4.4	Modelo SR para a SDsituation Revisão de Artigos.	62
FIG.4.5	Modelo SR para a SDsituation Submissão de Artigos - adaptada de (OLIVEIRA, 2008).	64
FIG.4.6	Modelo SR para a SDsituation Aceitação de Propostas - adaptada de (OLIVEIRA, 2008).	65
FIG.4.7	Modelo SR para a SDsituation Revisão de Artigos - adaptada de (OLIVEIRA, 2008).	66
FIG.4.8	Refinamento da transparência do processo.	67
FIG.4.9	Operacionalização da transparência do processo.	67
FIG.4.10	Refinamento da transparência dos artigos.	68
FIG.4.11	Refinamento da transparência dos artigos (2º nível).	69
FIG.4.12	Refinamento da transparência dos artigos (3º nível).	70
FIG.4.13	Refinamento da transparência dos artigos (4º nível).	71
FIG.4.14	Refinamento da transparência dos artigos/Revisor (2º nível).	71
FIG.4.15	Refinamento da transparência dos artigos/Revisor (3º nível).	72
FIG.4.16	Refinamento da transparência dos artigos/Revisor (4º nível).	73
FIG.4.17	Operacionalização da transparência dos artigos.	74
FIG.4.18	Limitação da transparência dos artigos.	75
FIG.4.19	Modelo SR para a SDsituation Atendimento de Mesa (OLIVEIRA, 2008).	76
FIG.4.20	Modelo SR para a SDsituation Atendimento de Mesa - adaptada de (OLIVEIRA, 2008).	77
FIG.4.21	Refinamento da transparência do processo de atendimento.	78
FIG.4.22	Refinamento da transparência do Menu de Opções.	79

LISTA DE TABELAS

TAB.2.1	Exemplo de catálogo de independências implícitas (CHUNG et al., 2000).	27
TAB.4.1	Classificação dos atores quanto aos processos considerados no ECS.	59
TAB.4.2	Classificação dos atores quanto a organização e acesso ao sistema no ECS.	59
TAB.4.3	Credencial de acesso as informações dos atores do ECS.	59
TAB.4.4	Destinação das informações no processo Submissão.	60
TAB.4.5	Destinação das informações no processo Aceitação.	61
TAB.4.6	Destinação das informações no processo Revisão.	63
TAB.4.7	Destinação das informações no processo Revisão - atualizada.	73
TAB.4.8	Classificação dos atores do sistema.	75
TAB.4.9	Destinação das informações no processo Atendimento.	76
TAB.4.10	Destinação das informações no processo Atendimento - atualizada.	80

LISTA DE ABREVIATURAS

ABREVIATURAS

ECS	-	<i>Expert Committee System</i>
EEUU	-	<i>Estados Unidos da América</i>
FFAA	-	<i>Forças Armadas</i>
IME	-	<i>Instituto Militar de Engenharia</i>
PTP	-	<i>Política de Transparência do Patrocinador</i>
RF	-	<i>Requisito Funcional</i>
RNF	-	<i>Requisito Não Funcional</i>
SIG	-	<i>Softgoal Interdependency Graphs</i>
TD	-	<i>Transparência dos Dados do Sistema</i>
TDO	-	<i>Transparência dos Dados Ostensivos do Sistema</i>
TDR	-	<i>Transparência dos Dados Restritos do Sistema</i>
TP	-	<i>Transparência das informações sobre os processos do sistema</i>
TP	-	<i>Transparência dos Dados do Processo</i>
TPO	-	<i>Transparência dos Dados Ostensivos do Processo</i>
TPR	-	<i>Transparência dos Dados Restritos do Processo</i>

RESUMO

A transparência do sistema é importante para garantir a seus atores a correta visualização das informações. A elaboração de métodos, técnicas e ferramentas para tratar este Requisito Não Funcional é fundamental para o desenvolvimento de softwares no futuro. É importante pensar a transparência desde os momentos iniciais do projeto, pois a sua inclusão tardia, após a conclusão do mesmo, apresenta custos significativamente mais elevados.

Este trabalho apresenta um catálogo de conceitos e métodos de refinamento que contribuem para garantir a transparência dos sistemas desenvolvidos. A aplicação dos métodos propostos é demonstrada através de estudos de casos baseados em modelos de sistema amplamente utilizados na literatura. Através destes estudos de casos mostra-se a modelagem dos requisitos de transparência desde a sua definição até a sua operacionalização.

ABSTRACT

System transparency is important to assure to its actors the correct visualization of information. The elaboration of methods, techniques, and tools to treat this Non-Functional Requirement has utmost importance on the development of future software. It is important to take transparency into account from the beginning of the project, because its inclusion in a later stage of development presents significant cost.

We present a catalogue of refining methods and concepts that contribute to assure transparency on the developed systems. The application of the proposed methods is shown through case studies which are based on system models found in the literature. Through these case studies, we show the modelling of the transparency requirements from their definition to their operationalization.

1 INTRODUÇÃO

O direito de ser informado e de ter acesso às informações apropriadamente é uma tendência evidente nas sociedades democráticas modernas (LEITE, 2006b). De fato, a transparência das informações é uma demanda social (HOLZNER e HOLZNER, 2006), que só recentemente passou a ser considerada nas relações cotidianas das pessoas e organizações.

Para que haja transparência, é necessário que haja uma abertura que permita a visualização das informações por alguém (HOLZNER e HOLZNER, 2006). Assim, formam-se dois tipos de interessados¹. O primeiro se caracteriza pela intenção de permitir que suas informações sejam transparentes, considerando a importância das mesmas para si e o impacto, negativo ou positivo, que o seu conhecimento por outrem poderá causar. O segundo se caracteriza pela intenção de conhecer informações transparentes, considerando que as mesmas contribuirão positivamente para a realização de algum objetivo.

A caracterização dos interessados mostra que a decisão quanto ao sigilo de uma informação deve ser tomada por quem a detém (HOLZNER e HOLZNER, 2006). Porém, a sua transparência só fará sentido se existir alguém que queira conhecê-la, não sendo justificável o custo de tornar uma informação transparente sem um destinatário correto (RIBEIRO e BARRADAS, 1980). Desta forma, percebe-se que a transparência, respeitando as decisões de quem detém a informação, deve ser focada em quem vai visualizar a informação.

Alguns esforços estão sendo realizados por diferentes entidades da sociedade para garantir transparência nas informações. Estes esforços podem ser divididos sob dois conceitos. Um tem o objetivo de disponibilizar dados pré-definidos, organizados segundo diferentes taxonomias, sem um estudo específico sobre quem irá visualizar a informação e sem permitir uma interação além de consultas limitadas, como é o caso de algumas empresas (PETROBRAS, 2008; CSN, 2006), governos (DOBRASIL, 2005; ALRGS, 2005) e iniciativas independentes (CORRUPÇÃO). O outro tem o objetivo de tentar garantir um nível mínimo de transparência em alguns seguimentos da sociedade, guiado principalmente por interesses financeiros, como é o caso da Lei Sarbanes-Oxley nos EEUU (CONGRESS,

¹O conceito de interesse e interessado adotado neste estudo é o mesmo apresentado por Sommerville e Sawyer (SOMMERVILLE e SAWYER, 1997).

2002) e a Diretiva 2004/109/CE do Parlamento Europeu (EUROPEU, 2004).

Tais exemplos mostram a importância do assunto na atualidade, fato que pode ser justificado pelo impacto negativo que a falta de transparência, de certas informações, pode causar a diferentes áreas de interesse social, como por exemplo, nas pesquisas científicas (ROSSNER et al., 2007) e na contratação de serviços bancários (GREVE, 2007).

É evidente que, atualmente, as pessoas e as organizações encontram-se dependentes de diferentes sistemas de software. Esta dependência tem mostrado uma tendência de aumento, em vista da necessidade das organizações e governos em digitalizar seus processos (KALAKOTA e ROBINSON, 2003) e das pessoas em interagir com estes. A complexidade dos sistemas atuais, devido em grande parte a existência de requisitos de tempo real (PETERS e PARNAS, 2000), tem tornado o monitoramento da obediência dos requisitos, uma tarefa imprescindível para a credibilidade dos mesmos (ROBINSON, 2005; HOFFMAN, 2004). Assim, em um futuro próximo, a transparência não ficará restrita as informações gerenciadas por um sistema, pois a sociedade passará a demandar as informações sobre o software (LEITE, 2006a). Por isso, pode-se dizer que os engenheiros de software terão que lidar com a questão da transparência com mais frequência, nos desenvolvimentos de software no futuro (LEITE, 2006b).

Para permitir que a transparência seja considerada no desenvolvimento de um sistema, é necessário primeiramente a apreensão do ambiente do software a ser construído, ou seja, compreender as particularidades do mundo real que o software estará inserido quando automatizar os comportamentos funcionais de uma organização (BUBENKO, 1980; BORGIDA et al., 1985; ZAVE e JACKSON, 1997; LEITE e FRANCO, 1993; LEITE et al., 2000). Esta compreensão facilita a correta aquisição dos requisitos funcionais (RF) e não funcionais (RNF) necessários ao sistema, permitindo seu correto desenvolvimento (THEYER e DORFMAN, 1990; VANLAMSWEEERDE, 2000a).

A transparência já é reconhecida como um **RNF** a ser considerado nos desenvolvimentos de sistemas de softwares a bastante tempo (CHUNG et al., 2000). Porém, só recentemente passou a fazer parte do foco de estudos. Por isso, existem poucos trabalhos sobre o assunto. Um ponto em comum dentre os trabalhos existentes é que abordam a transparência de forma teórica, e apenas como parte da solução de outros problemas, como por exemplo: avaliação de processos de modelagem (CAPPELLI et al., 2007), processamento de imagens (SEINSTRÁ et al., 2002) e desenvolvimento de arquiteturas (ZADE et al., 2008).

Assim sendo, (LEITE, 2006a) considerando as características da nova coletividade existente nos dias de hoje, ressaltou a necessidade da institucionalização de conceitos, métodos e ferramentas que tratem deste RNF durante a produção de sistemas, visando o atendimento das futuras exigências da sociedade quanto aos softwares que ainda serão desenvolvidos. Tal fato evitará a atual necessidade de retrabalho sobre os softwares já terminados e, muitas vezes, já entregues aos clientes, para adaptá-los aos novos requisitos.

Segundo Boehm e Papaccio (apud (PFLEEGER, 2004)) o custo relativo para o conserto de um software devido a problemas na definição de requisitos é de \$ 1 na fase da análise de requisitos, \$ 5 na fase do projeto, \$ 10 na fase de codificação, \$ 20 na fase de teste de unidade e de \$ 200 após a entrega do sistema ao cliente. Somado a isto, falhas na definição de requisitos são responsáveis por um grande percentual no insucesso dos projetos de software (VANLAMSWEERDE, 2000b).

Portanto, é importante criar mecanismos que permitam o tratamento da transparência desde as fases mais iniciais do desenvolvimento de software (DARDENNE et al., 1993; CHUNG et al., 2000). Isto permitirá ao desenvolvedor projetar a transparência em um software, produzindo documentos de requisitos mais completos, organizados e acurados. Esta visão facilita o desenvolvimento, impede a propagação de erros para as fases posteriores do processo e facilita a rastreabilidade (KALINOWSKI et al., 2007; STOUT, 2001; PRESSMAN, 2005). Assim, diversos problemas de omissão, conflito, ambigüidade e dinamismo sobre os requisitos (CHRISTEL e KANG, 1992), seriam solucionados muito antes da entrega do executável.

“Análise de requisitos é um passo altamente crítico no ciclo de vida do software” (DARDENNE et al., 1993).

Devido à capacidade de capturar os diferentes níveis de abstração, sobre diversos objetivos de um sistema a ser desenvolvido, o uso de metas como base em métodos e técnicas para o gerenciamento de requisitos tem sido amplamente explorado nos últimos anos (YU e MYLOPOULOS, 1998; DARIMONT e VAN LAMSWEERDE, 1996; VANLAMSWEERDE e LETIER, 2000; VANLAMSWEERDE, 2001; LETIER, 2001).

1.1 OBJETIVO

Não existe um consenso sobre como lidar com os requisitos necessários a transparência. Isto se deve ao fato de que os engenheiros de software não possuem o conhecimento para compreender suas necessidades, nem uma forma de definir e organizar seus requisitos, permitindo sua operacionalização. Apesar disso, sabe-se que a transparência é um atributo de qualidade de software, e como tal, deve ser considerado um requisito não funcional do sistema (MYLOPOULOS et al., 1992; DARDENNE et al., 1993; CHUNG et al., 2000).

O objetivo desta dissertação consiste na definição dos conceitos necessários à compreensão da transparência das informações em sistemas de software e a definição da forma correta para aplicá-los em favor do desenvolvimento de um software transparente. Este trabalho será apresentado na forma de um catálogo, isto é, reunido em uma coletânea organizada e flexível, similar ao trabalho apresentado por (CHUNG et al., 2000).

O propósito de se criar um catálogo está relacionado ao fato de que a qualidade de um software é diretamente impactada pela qualidade na definição de seus requisitos (BELL e THAYER, 1976). Erros na especificação de requisitos podem gerar altos custos de manutenção, que podem superar em muitas vezes os custos de desenvolvimento, devido à necessidade de voltar-se aos artefatos iniciais do software para a realização de correções (BOEHM, 1981; BOEHM e PAPACCIO, 1988).

Assim, é importante desenvolver ferramentas específicas que permitam aos engenheiros de software lidar com a transparência desde as fases iniciais do projeto. No caso específico deste trabalho, o catálogo proposto deverá guiar os desenvolvedores na compreensão do ambiente computacional e do domínio do problema, colaborando com a definição de seus requisitos, descrevendo-os, refinando-os e operacionalizando-os (CHUNG et al., 1994, 2000).

1.2 CONTRIBUIÇÕES

- Produzir o conhecimento necessário à identificação dos requisitos não funcionais de transparência em um sistema;
- Apresentar este conhecimento na forma de um catálogo, similar ao modelo apresentado em (CHUNG et al., 2000); e
- Apresentar provas de conceito que exemplifiquem a modelagem dos requisitos de

transparência, desde a sua definição até sua operacionalização.

1.3 ORGANIZAÇÃO DO TEXTO

Este texto está organizado da seguinte forma. O capítulo 2 apresenta os conceitos necessários ao entendimento do trabalho. No capítulo 3, será apresentado o catálogo do requisito não funcional transparência, seguindo a técnica apresentada em (CHUNG et al., 2000). O capítulo 4 apresenta uma prova de conceito realizada para validar os conceitos apresentados no catálogo proposto, através da modelagem de um sistema com o requisito transparência. O capítulo 5 apresenta as conclusões e os trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo tem o objetivo de apresentar os conceitos necessários a compreensão deste trabalho.

Inicialmente, será apresentado a definição de elementos que serão amplamente utilizados, seguido da apresentação dos dois frameworks que serão utilizados na catalogação do RNF e no estudo de caso, respectivamente.

2.1 TRANSPARÊNCIA DAS INFORMAÇÕES

A transparência possui diferentes significados, de acordo com o contexto em que é aplicada.

- a semântica da palavra transparência significa a qualidade de algo que é transparente, ou seja, aquilo que permite que a luz o atravesse para que outros objetos sejam distinguidos através de sua espessura (BUENO, 1979);
- no senso comum, esta propriedade é entendida como sendo referente a algo evidente ou que deixa perceber um sentido oculto (WIKIPEDIA);
- na indústria do cinema, a transparência ocorre quando o espectador é incapaz de perceber a técnica utilizada para compor as cenas (XAVIER, 1977);
- no setor público, a transparência é considerada um fator vital para fortalecer as relações entre o governo e os cidadãos que pode se estabelecer através da completeza, objetividade, veracidade, excelência, acessibilidade e entendimento das informações (KONDO, 2002); e
- na indústria de software, software cuja execução exige praticamente nenhum conhecimento do operador (MICROSOFT, 2002).

A diversidade de significados mostra a amplitude dos assuntos que devem ser considerados para a compreensão da transparência. Tal compreensão não pode ser feita isoladamente, pois a análise dos significados identifica várias características, como completeza, objetividade, confiabilidade, acurácia, consistência, entre outras, que completam

e contribuem com o sentido a ser considerado na transparência (OLIVEIRA et al., 2007; CAPPELLI et al., 2007).

Diferentes interpretações levam a múltiplas perspectivas, fato que é comum as entidades de “*Qualidade*”, devido a sua característica multidimensional (HARRISON e NITHI). Porém, no caso da transparência, percebe-se a existência da dependência da modelagem de outros tipos de requisitos para dar suporte a sua satisfação, operacionalizando-a.

Algumas informações são consideradas vitais pelas organizações que as possuem, e por isso devem ser resguardadas da exposição indevida (HOLZNER e HOLZNER, 2006). A tarefa de resguardar as informações pode ser realizada de diversas maneiras, porém, no tocante ao uso de sistemas de informação, este controle sobre o conhecimento de determinadas informações será operacionalizado pela visualização controlada destas, por determinados atores do sistema.

A presença do antagonismo provocado pelo conflito entre a necessidade de abertura e a necessidade de sigilo, é uma questão marcante para a produção de um software transparente, ou seja, estas duas necessidades devem conviver em um equilíbrio satisfatório (HOLZNER e HOLZNER, 2006).

A ocorrência da transparência de alguma informação é verificada pela visualização, ou não, da mesma por alguém. Porém, a análise deste RNF deve sempre considerar o interesse do patrocinador² do sistema em tornar transparente uma informação, através da sua *Política de Transparência*, e o interesse do ator do sistema em conhecê-la, sendo que, o foco principal deve ser o ator. Isto porque, somente a visão do ator poderá validar se a informação está realmente transparente ou não.

Exemplificando isto, para operacionalizar o RNF Segurança, o desenvolvedor, atendendo as intenções do patrocinador do sistema, pode optar por usar no sistema uma criptografia de 64 ou 128 bits (CUNHA, 2007). Independente da decisão tomada, ela é irrelevante para os atores, pois a técnica de criptografia não é percebida por eles, sendo apenas uma ferramenta disponível no sistema, não tendo a necessidade de emitirem qualquer tipo de “*feedback*”, sobre a mesma, para o desenvolvedor. Diferentemente disso, a consideração de um “*feedback*” positivo sobre a transparência de uma informação, interessa ao desenvolvedor para que possa verificar se os requisitos foram atendidos, ou não,

²Os termos *patrocinador* e *organização patrocinadora* reference ao cliente para o qual o sistema está sendo construído.

segundo o ponto de vista do ator.

A Política de Transparência do Patrocinador (PTP) são as orientações que deverão ser utilizadas pelo engenheiro de software na elicitação dos requisitos não funcionais de transparência. Tais orientações são definidas baseadas nas características dos atores, nas características das informações, nas limitações e obrigações definidas pelas regras do negócio específico da organização patrocinadora, e na premissa de que, se não houverem implicações negativas quanto ao conhecimento de uma informação pelo ator do sistema, a mesma pode ser transparente a ele.

Portanto, para o desenvolvimento de um software com transparência, além da coleta de informações relevantes ao projeto e do desenvolvimento preocupado com a satisfação das PTP, deve-se ainda, identificar as necessidades oriundas dos interesses dos diferentes atores do sistema.

A informação é uma entidade que participa, obrigatoriamente, dos requisitos de transparência de um sistema. Existem duas formas de compreender seus conceitos (RIBEIRO e BARRADAS, 1980). A primeira utiliza o sentido comum, ou seja, o senso comum da vida cotidiana do indivíduo; e a outra, utiliza o sentido técnico. O sentido comum baseia-se no interesse pelo significado e valia da informação. O sentido técnico baseia-se na análise da utilização do sistema (*canal*) para tornar transparente (*transmitir*) informações que sejam compreendidas e tenham valia para determinado ator.

É fato que toda informação deve despertar algum interesse pelo seu significado em quem a recebe, porém a compreensão do significado de uma informação é diretamente ligado a capacidade do indivíduo que a recebe em associá-la a algum conhecimento prévio sobre o assunto em questão (RIBEIRO e BARRADAS, 1980). A valia da informação está associada ao conteúdo emocional despertado no indivíduo receptor da mesma (RIBEIRO e BARRADAS, 1980). A subjetividade destes conceitos torna necessário a criação de modelos técnicos específicos para permitir a compreensão e aplicação dos conceitos sobre a informação.

Uma informação pode ser composta por um único item de informação ou por uma combinação destes. A variedade nos arranjos de itens de informação pode gerar diferentes visões sobre a mesma informação ou uma informação diferente. A figura 2.1 apresenta diferentes visões sobre uma conta de restaurante, onde a *conta 2* permite que o cliente faça a conferência de seu *valor total*. Assim, tornar transparente uma informação para um ator não é uma tarefa simples e direta, pois uma informação enviada erradamente

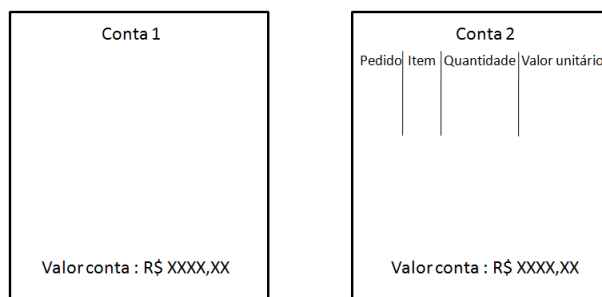


FIG. 2.1: Exemplo de visões diferentes de uma conta.

para um ator, provavelmente não será compreendida ou não terá valia, além de consumir desnecessariamente recursos do sistema, da mesma forma, uma informação errada ou incompleta pode gerar prejuízos ao ator e ao patrocinador, razões que exigem um estudo detalhado por parte do engenheiro de software.

A transparência das informações pode ser descrita como um processo de comunicação, e como tal, apresenta em sua estrutura dois pontos de vista a serem considerados. O ponto de vista da fonte, que imprime informação aos símbolos disponibilizados para visualização, e o ponto de vista do destinatário, que recolhe a informação pela identificação dos símbolos disponibilizados (RIBEIRO e BARRADAS, 1980).

“A natureza básica do fenômeno da comunicação consiste na transferência da informação da fonte ao destinatário” (RIBEIRO e BARRADAS, 1980).

A figura 2.2 representa os componentes do processo de transparência, sendo a fonte o próprio sistema a ser desenvolvido, o destinatário são os atores e os processos do sistema são os canais de ligação entre a fonte e os diferentes destinatários.

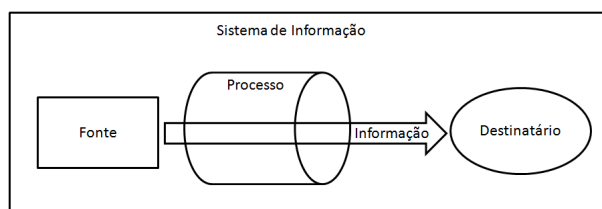


FIG. 2.2: Processo de transparência da informação.

Todo software produz algum tipo de informação, seja ela o cerne do seu propósito ou não. Existem diferentes tipos de informações sobre um software (LEITE, 2006a). As

informações manipuladas de acordo com o “*workflow*” dos processos de um sistema tem dois tipos de origens. O primeiro tipo é composto pelas informações originadas dos dados gerenciados pelo sistema (ex.: saldo da conta corrente). O segundo tipo é composto pelas informações sobre os processos, suas tarefas e etapas, executados pelo sistema (ex.: confirmação de depósito em conta corrente e como é feito o cálculo do juros sobre o saldo negativo), acabando com a idéia de sistemas *caixa preta*, onde o ator só conhece a entrada e a saída produzida.

Analisando os conceitos apresentados, define-se para este estudo que:

O software transparente é aquele que permite aos seus diferentes atores, ter diferentes visões, sobre as diferentes informações gerenciadas por ele, satisfazendo a contento as necessidades individuais de cada interessado.

A ausência de uma métrica específica para graduar a transparência de um sistema, dificulta a comparação entre sistemas diferentes. Apesar disto, a transparência ideal para qualquer sistema é aquela que atende integralmente a PTP. Assim, quanto mais branda for a PTP, mais transparente será o sistema. Por exemplo, a conta 2 é mais transparente ao cliente do que a conta 1 (figura 2.1).

2.2 NFR FRAMEWORK

Durante o desenvolvimento de um sistema de informação várias decisões devem ser tomadas pelo engenheiro de software. Tais decisões promovem diferentes impactos sobre os diversos aspectos do sistema (CHUNG et al., 1994).

Considerando o fato de que a satisfação dos RNF de um sistema é tão crucial quanto prover suas funcionalidades (CHUNG et al., 1994) e que abordar o desenho de sistemas através de seus RF é uma tarefa menos complexa (CHUNG et al., 2000), o NFR Framework (CHUNG, 1993b) veio preencher as necessidades para a representação dos RNF e apresentar uma metodologia para encontrar estes requisitos, resolvendo possíveis conflitos e facilitando a rastreabilidade dos mesmos (CHUNG, 1991). Além disso, utiliza uma abordagem orientada ao processo, complementando a tradicional abordagem orientada ao produto (MYLOPOULOS et al., 1992).

Este framework possui oito etapas e cinco componentes principais. As etapas não

são obrigatoriamente sequenciais, permitindo a interação do desenvolvedor com qualquer uma delas sempre que necessário, flexibilizando o uso de abordagens Top-down ou Bottom-up (CHUNG et al., 2000).

As etapas determinam a aquisição do conhecimento necessário ao desenvolvimento, a identificação dos RNF particulares ao domínio, a decomposição dos RNF identificados, a identificação das operacionalizações possíveis, além de prevê o estudo das ambiguidades, a negociação de prioridades, a identificação das interdependências entre RNF e operacionalizações identificadas, e a seleção entre as operacionalizações. Esta última é apoiada pelo uso do desenho para o raciocínio necessário a tomada de decisões e avaliação do impacto das mesmas (MYLOPOULOS et al., 1992; CHUNG et al., 2000).

Os componentes deste framework são metas, ligações, métodos genéricos, regras de correlação e procedimentos de rotulação. Estes elementos são utilizados em uma estrutura que permite representar e registrar o desenho e os raciocínios do processo em gráficos (CHUNG et al., 2000).

O Softgoal Interdependency Graphs (SIG) permite que o desenvolvedor possa, de forma incremental e interativa, ir passo a passo construindo, analisando e revisando o projeto em desenvolvimento, registrando suas considerações sobre as metas flexíveis e as interdependências entre elas, além de possibilitar a rastreabilidade no desenvolvimento do sistema (CHUNG et al., 2000).

Os requisitos principais definidos são mostrados como metas flexíveis no topo do gráfico (“parent”) e a partir daí decompostas em outras metas flexíveis (“offspring”) conectadas por ligações de dependência. As metas flexíveis são representadas por núvens e possuem rótulos que apoiam o processo de raciocínio durante o desenvolvimento do processo (CHUNG et al., 2000).

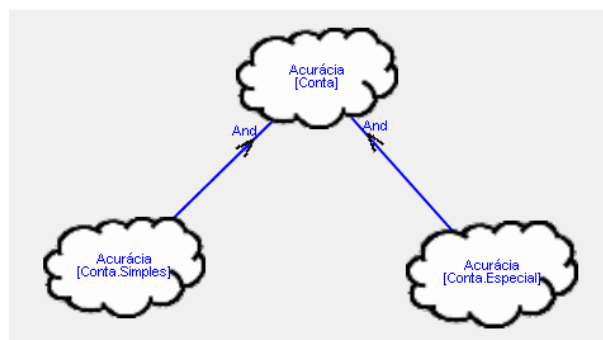


FIG. 2.3: Exemplo de elementos do SIG (CHUNG et al., 2000).

O mais importante deste framework é que durante todo o processo, o desenvolvedor tem controle sobre a declaração e o refinamento das metas flexíveis declaradas. Por isso, podemos constatar que o procedimento de avaliação é interativo, semi-automático e totalmente sobre o controle do desenvolvedor (MYLOPOULOS et al., 1992; CHUNG et al., 1994, 2000).

Outra contribuição deste framework é a catalogação do conhecimento sobre os RNFs, desenhos de sistema e técnicas de desenvolvimento (CHUNG et al., 2000).

O catálogo de um RNF permite o registro do conhecimento adquirido sobre os mesmos, de forma organizada e flexível, favorecendo a dissiminação, reuso e adaptação do “*Know-How*” pelo engenheiro de software. Porém, a principal vantagem no uso de um catálogo é proporcionar aos desenvolvedores o conhecimento necessário para facilitar as decisões que deverão ser tomadas, desde as fases iniciais do processo de desenvolvimento, visando as próximas fases do projeto (CHUNG et al., 2000).

Normalmente, um catálogo é composto de três partes, sendo: catálogo do tipo de RNF (figura 2.4), catálogo dos métodos de refinamento (figura 2.5) e catálogo das regras de correlação (tabela 2.1).

O primeiro codifica os conceitos particulares ao tipo de RNF considerado (especializações, características e terminologia).

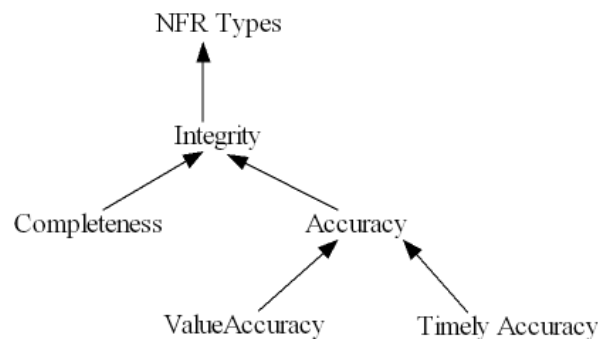


FIG. 2.4: Exemplo de catálogo de Tipo de RNF (CHUNG et al., 2000).

O segundo, codifica o conhecimento que ajuda no refinamento dos modelos, através de metas flexíveis de decomposição, considerando as operacionalizações.

O terceiro, codifica o conhecimento que ajuda a detecção das interdependências implícitas entre as metas flexíveis, apresentando o grau do impacto que as metas exercem sobre as outras.

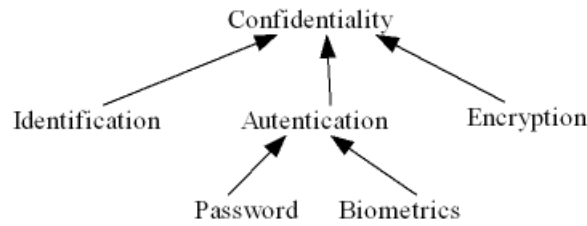


FIG. 2.5: Exemplo de catálogo de métodos de refinamento (CHUNG et al., 2000).

TAB. 2.1: Exemplo de catálogo de independências implícitas (CHUNG et al., 2000).

Metas Flexíveis Descendentes	Impacto Sobre as Metas Flexíveis Parentes	
Metas Flexíveis de Operacionalização	Acurácia	Confidencialidade
Validação	+	+
Compressão		
Indexação		
Autorização		+
ID Adicional		+

2.3 O FRAMEWORK I - STAR

O i^* (YU, 1995) é um framework que tem o propósito de modelar contextos organizacionais, tomando por base os relacionamentos de dependência entre os atores participantes, para que metas próprias sejam atingidas.

O uso deste framework facilita o entendimento dos relacionamentos organizacionais, contribuindo para a elicitación das metas nas fases iniciais do desenvolvimento, pois permite a compreensão das razões internas dos atores, favorecendo a escolha entre as alternativas dos modelos do software (CUNHA, 2007).

Para realizar a tarefa de modelar, o framework i^* utiliza dois modelos diferentes e complementares, originalmente definidos por (YU, 1995), sendo o modelo SD e o modelo SR.

2.3.1 O MODELO SD - STRATEGIC DEPENDENCY MODEL

Expressa exclusivamente relacionamentos de dependência estratégica entre os atores, onde cada relacionamento representa uma cooperação entre os atores. As dependências podem ser de quatro tipos diferentes (figura 2.6).

Na figura 2.6 podemos notar nós, que representam os atores, e arestas, que representam as dependências. Os atores são classificados como depender e dependee. O depender depende do dependee para alguma coisa. O objeto da dependência encontra-se na aresta,

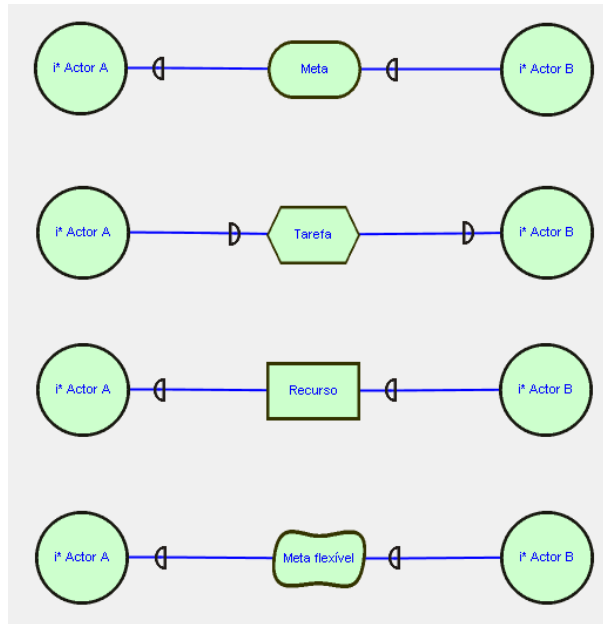


FIG. 2.6: Tipos de dependências estratégicas.

e é chamado de dependum.

O primeiro tipo de dependência entre os atores é por meta concreta. Nesta dependência, o ator B depende do ator A para que possa atingir algum estado no mundo real. A dependência por meta possui um grau de liberdade máximo, ou seja, o dependee é o único responsável pelo cumprimento da meta.

O segundo tipo de dependência, apresenta uma tarefa como dependum, onde o ator A depende que o ator B execute uma tarefa. Esta dependência possui um grau de liberdade menor, pois o ator B executa a tarefa da forma como o ator A determina.

Na dependência por recurso temos que o ator B depende do ator A fornecer o recurso, que pode ser uma entidade física ou informacional. Esta dependência possui um grau de liberdade nulo, ou seja, o dependee deve fornecer o dependum exatamente como o depender quer.

A última dependência é por meta flexível. Nesta dependência, o ator B depende que o ator A atinja alguma meta com determinada qualidade. Esta dependência permite ao dependee o grau máximo de liberdade para atingir a meta flexível, porém o depender também tem total liberdade para aceitar ou não se a meta foi atingida.

2.3.2 O MODELO SR - STRATEGIC RATIONALE MODEL

Detalha as razões estratégicas internas aos atores (figura 2.7), onde seus comportamentos são modelados por relações meio-fim (means-ends), decomposições de tarefas (task decompositions) e contribuições por meta flexível (softgoal contributions).

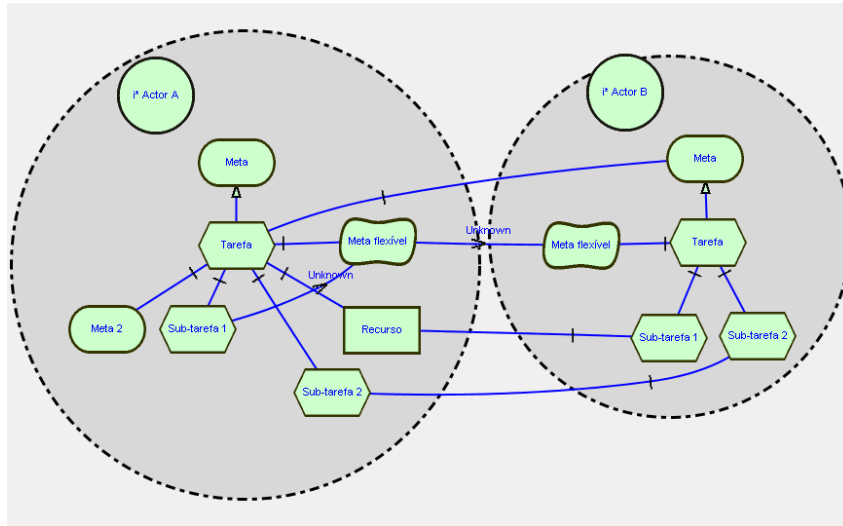


FIG. 2.7: Exemplo de modelo SR.

Os elementos deste modelo permitem representar o “rationale” dos atores. As relações meio-fim explicita as decisões que envolvem o alcance das metas do ator. A decomposição de tarefas mostra como estas são elaboradas e realizadas, além de mostrar como os recursos são utilizados e disponibilizados. A contribuição das metas flexíveis definem explicitamente a contribuição positiva ou negativa entre elas, ou entre outros elementos e as metas flexíveis.

A figura 2.7 mostra nós e arestas, sendo que os nós são formados por metas, tarefas, recursos e metas flexíveis, e as arestas são os elos meio-fim, os elos de decomposição de tarefas e os elos de contribuição.

Os elos de contribuição deverão ser identificados de acordo com o seu respectivo grau de contribuição, que representarão desde uma contribuição muito fraca, passando por desconhecida, até chegar a muito forte (figura 2.8). Esta escala de contribuições também é utilizada pelo NFR Framework.

Além dos modelos SD e SR, este trabalho necessita utilizar duas extensões para o framework originalmente proposto por Yu (YU, 1995). Estas extensões são o modelo SA (Strategic Actor) e a SDsituation (Situação de Dependência Estratégica).

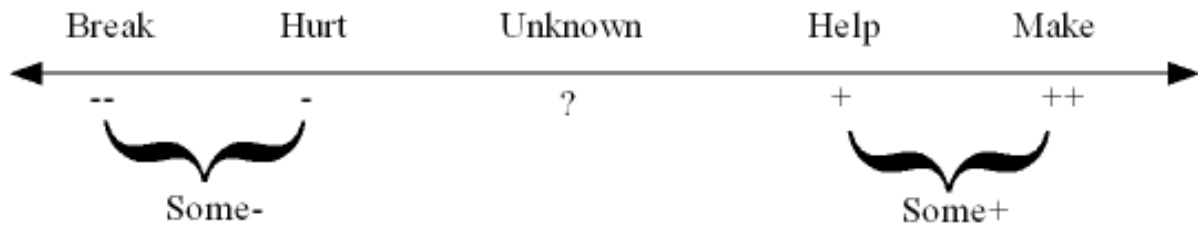


FIG. 2.8: Escala de contribuições.

2.3.3 O MODELO SA - STRATEGIC ACTOR MODEL

É específico para a modelagem de atores (LEITE et al., 2007), contribuindo para melhorar o entendimento dos atores e seus relacionamentos. Uma vantagem deste modelo é a possibilidade de visualizarmos os atores de acordo com a estrutura da organização que pertencem.

Este modelo permite estudar os conceitos de Ator, Posição, Papel e Agente, conforme definidos por Yu (YU, 1995), através dos relacionamentos Ocupa, Desempenha, Cobre, Parte de, É um e Instancia (figura 2.9).

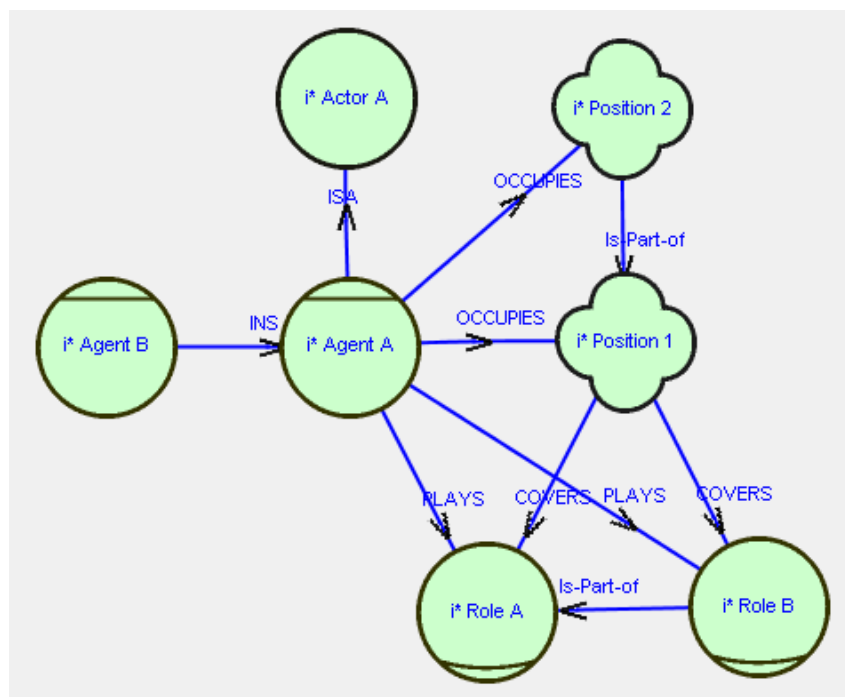


FIG. 2.9: Exemplo de modelo SA.

Segundo a figura 2.9, temos que o Agente A é (ISA) um Ator A; o Agente B instancia (INS) o Agente A; o Agente A ocupa (Occupies) as Posições 1 e 2; a Posição 2 é parte da

(Is-Part-of) Posição 1; a Posição 1 cobre (Covers) os Papeis A e B; o Papel B é parte do (Is-Part-of) Papel A; e o Agente A desempenha (Plays) os Papeis A e B.

2.3.4 SDSITUATION - STRATEGIC DEPENDENCY SITUATION

Foram definidas por Oliveira e Cysneiros (OLIVEIRA e CYSNEIROS, 2006) como estruturas ou constructos para representar uma situação de dependência estratégica. Estas situações podem ser identificadas dentro dos ambientes organizacionais, e elas definem que os elos que unem os diferentes atores não estão isolados, podendo ser mapeados dentro de uma situação de colaboração bem definida (figura 2.10).

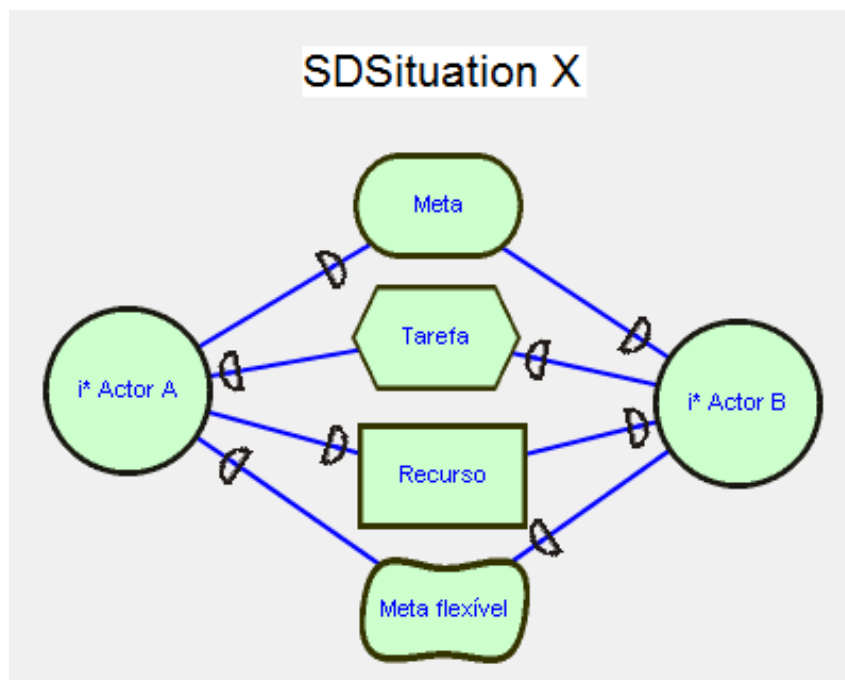


FIG. 2.10: Exemplo de SDSituation.

Cada SDSituation pode ser identificada independentemente de outras, apesar de pertencer a uma teia de interdependências, e podem conter um ou mais elementos de dependência estratégica, ou seja, dependências por meta, tarefa, recurso e meta flexível, juntas ou isoladas (figura 2.11).

As interdependências entre as SDSituations podem ser de três tipos, sendo: interdependência física, lógica ou temporal. Os tipos podem ocorrer simultaneamente em uma interdependência.

As SDSituations são caracterizadas pelas dependências que possuem e são definidas

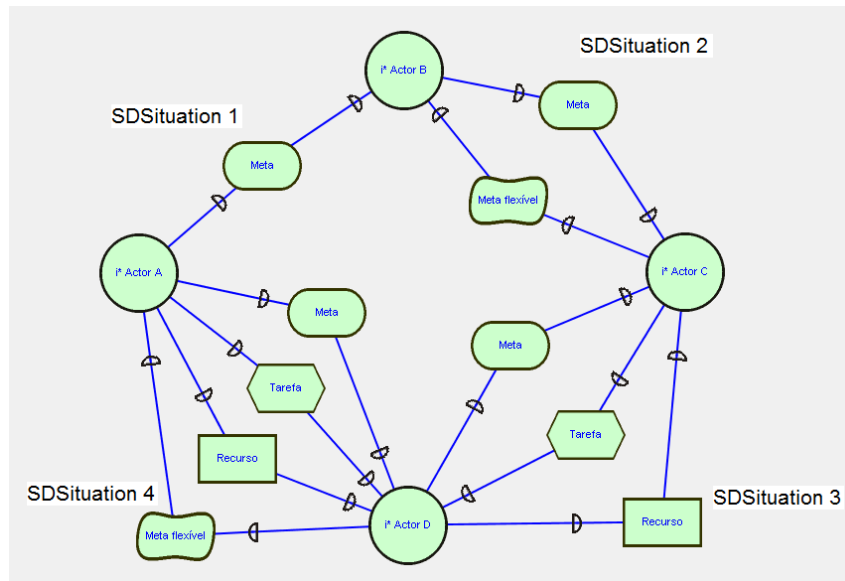


FIG. 2.11: Exemplo de uma teia de SDSituations.

através de três passos básicos, sendo: identificação dos elementos da SDSituation, composição da SDSituation e identificação do interrelacionamento entre as SDSituations.

2.4 CONCLUSÕES

Este capítulo mostrou que a transparência pode apresentar diferentes significados, conforme o contexto no qual está sendo considerada.

O “background” realizado sobre o que é uma *informação* considerou os conceitos da Teoria da Informação. Foi ressaltado a questão do antagonismo entre transparência e o sigilo das informações e apontando a necessidade de uma PTP bem definida.

A comparação do processo de transparência com um processo de comunicação possibilitou a identificação de duas origens para as informações em um sistema. Baseado nesta comparação apresentou-se uma definição para software transparente.

Além disso, foram mostrados os frameworks, i^* e NFR, já definidos na literatura, que serão utilizados no desenvolvimento deste trabalho.

3 REQUISITOS DE TRANSPARÊNCIA

Os sistemas de informação são artefatos sociais que devem servir a diferentes atores, com diferentes limitações e obrigações, envolvidas em diferentes transações do sistema (LIU e YU, 2003; CHUNG, 1991). Para desenhar tais sistemas é necessário lidar com os limites particulares ao ambiente do software e as tecnologias disponíveis (LIU et al., 2003). Isto torna imprescindível a compreensão do contexto social e organizacional do software em questão, além do ambiente que estará envolvido (LIU e YU, 2003; CHUNG, 1991; GOGUEN e LINDE, 1993).

Garantir a transparência de um sistema tem como propósito destinar corretamente as informações para que possam ser visualizadas pelos diferentes atores. Atingir este propósito tornou-se um desafio para os engenheiros de software devido à crescente demanda social por conhecimento (LEITE, 2006b; HOLZNER e HOLZNER, 2006).

Tal conhecimento pode ser relacionado a diversos assuntos e derivar de diferentes origens (LEITE, 2006a), mas o escopo tratado neste trabalho está limitado às informações originadas dos dados gerenciados pelo sistema e naquelas sobre os processos do sistema.

A transparência de uma informação é um problema inerente a um sistema de software, devido a necessidade de definir a sua correta destinação, ou seja, definir quais atores deverão visualizá-la. Por exemplo, em um sistema bancário, o nome do funcionário encarregado por avaliar o pedido de empréstimo de um correntista é uma informação que interessa tanto ao cliente que solicitou o empréstimo, quanto ao gerente de sua conta, cada um motivado por razões diferentes. Porém, considerando a PTP do sistema, definiu-se que o *nome do avaliador* de empréstimo é uma informação que só deve ser conhecida pelo gerente responsável pela conta do correntista que solicitou o empréstimo. Analisando este exemplo, verifica-se que dois atores tem interesse em conhecer uma informação, mas somente um poderá visualizá-la. Além disso, percebem-se limitações que deverão ser obrigatoriamente respeitadas pelo sistema, que no caso é o acesso restrito aos gerentes de contas e entre eles, o acesso restrito aos pedidos referentes às contas de sua responsabilidade. Assim, a transparência da informação *nome do avaliador* dependerá da sua confidencialidade entre os diferentes atores do sistema, para ser operacionalizada, e não terá valia para o ator se não expressar a verdade, ou seja, se não for uma informação

acurada. A dependência de outros RNFs para operacionalizar a transparência torna sua modelagem mais complexa, pois aumenta a possibilidade de ocorrerem interferências entre os RNFs do sistema.

No entanto, a transparência, como os demais RNFs (ex.: segurança (CHUNG, 1993a,b; CHUNG et al., 2000; CUNHA, 2007), acurácia (CHUNG, 1991, 1993b; CHUNG et al., 2000), desempenho (NIXON, 1992, 1993, 1994; CHUNG et al., 2000), etc) não aparece automaticamente no modelo de um software, sendo necessária uma metodologia para entendê-la e projetá-la. Portanto, o conhecimento prévio de seus conceitos, métodos de refinamento e de suas correlações, contribuem com a modelagem da transparência em um sistema qualquer (CHUNG et al., 2000).

Este capítulo consolida as informações definidas para o RNF Transparência, catalogando-as em um Framework específico, customizando o NFR Framework (CHUNG, 1993b) e complementando o trabalho iniciado por (CHUNG et al., 2000).

3.1 CONCEITOS SOBRE OS REQUISITOS DE TRANSPARÊNCIA

Para garantir o sucesso do desenvolvimento o engenheiro de software deverá considerar alguns fatores de entrada e saída. Estes fatores deverão ser ponderados considerando três aspectos principais (figura 3.1), que irão direcionar o desenvolvimento dos requisitos não funcionais de transparência para cada sistema de software, permitindo que as informações sejam destinadas aos diferentes atores, de forma justificada, contemplando as intenções do patrocinador e definindo as orientações da PTP do sistema.

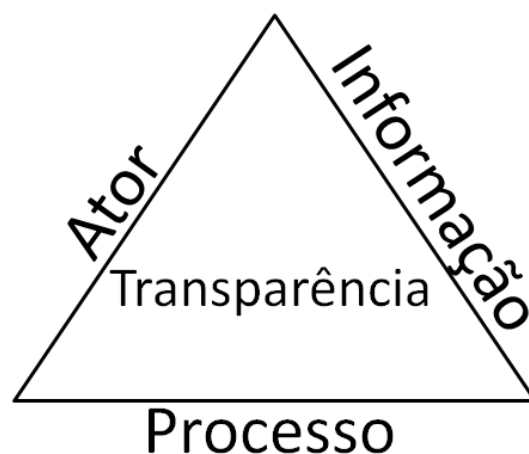


FIG. 3.1: Triângulo da Transparência.

A figura 3.1 representa em suas arestas os elementos fundamentais para a existência da transparência, de onde entende-se que cada ator do sistema visualiza uma determinada informação através de um ou mais processos específicos, ou seja, a transparência depende da interação destes elementos, onde o **Ator** é uma entidade ativa que desempenha atividades no sistema para atingir metas, a **Informação** é uma entidade passiva que possui significado compreensível e valia para algum ator e o **Processo** é um conjunto de atividades que serão desempenhadas por atores.

Assim, um primeiro fator a se considerar é a existência de duas origens para as informações em um sistema, obrigando a consideração das informações oriundas dos dados gerenciados pelo sistema, separadamente, das informações oriundas do andamento dos processos do sistema, pois a visualização de cada um destes tipos de informação será feita de forma diferente e servirá a diferentes propósitos. Os dados gerenciados pelo sistema têm o propósito de contribuir com o ator na execução de alguma tarefa junto à organização que utiliza o sistema. O andamento do processo tem o propósito de permitir ao ator acompanhar e entender as tarefas e etapas do processo ao qual está interagindo.

O segundo fator é a possível diversidade de atores em um sistema. Para facilitar o entendimento das características dos atores e as futuras decisões quanto à destinação das informações, inicialmente, indica-se a categorização de cada ator segundo a taxonomia a seguir, com o propósito de identificar e separar os atores em grupos, que de acordo com as particularidades de cada organização patrocinadora, terá implicações diferentes sobre as decisões tomadas para a destinação das informações.

- Quanto à organização;
- Quanto ao processo; e
- Quanto o acesso ao sistema.

A primeira categoria identifica, dentro dos possíveis atores de um sistema, aqueles que são internos ou externos à organização, ou seja, aqueles que fazem, ou não, parte de sua estrutura formal. O grupo de atores internos à organização é normalmente formado pelos seus funcionários e colaboradores. Os colaboradores, apesar de não serem funcionários da organização, são parceiros que também possuem seus vínculos bem definidos e formalizados com a organização, podendo inclusive ser outros sistemas (LIU et al., 2003). O

grupo de atores externos à organização é normalmente formado por seus clientes. A principal diferença entre estes dois grupos é que os atores internos possuem a sua identidade rigorosamente conhecida, diferente dos atores externos.

A segunda categoria identifica os atores que são internos ou externos aos diferentes processos da organização, ou seja, aqueles que participam, ou não, da execução de um processo. A interação do ator com o processo também tem que ser avaliada, pois um ator interno a um processo pode ter a capacidade de comprometer, ou não, a conclusão do mesmo.

A terceira categoria identifica os atores que interagem com o sistema através de conexões internas (controladas e próprias - intranet) e os que interagem através de conexões externas (não controladas e não próprias - internet).

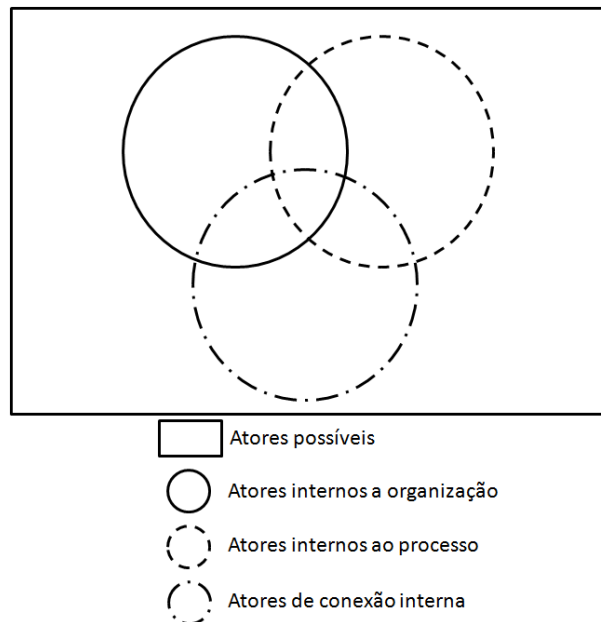


FIG. 3.2: Combinações da taxonomia proposta para os atores.

A figura 3.2 apresenta a taxonomia proposta para a classificação dos atores e suas combinações, cobrindo todo o universo de possibilidades para os tipos de atores. Assim, definem-se os seguintes tipos:

- Interno a organização, interno ao processo e com acesso interno (III);
- Interno a organização, interno ao processo e com acesso externo (IIE);
- Interno a organização, externo ao processo e com acesso interno (IEI);

- Interno a organização, externo ao processo e com acesso externo (IEE);
- Externo a organização, interno ao processo e com acesso interno (EII);
- Externo a organização, interno ao processo e com acesso externo (EIE);
- Externo a organização, externo ao processo e com acesso interno (EEI); e
- Externo a organização, externo ao processo e com acesso externo (EEE).

Assim, *professores* e *pais de alunos* são atores de um sistema de administração escolar, sendo que o primeiro é do *tipo III* e o segundo é do *tipo EIE*, em relação ao processo *consulta de notas*. Devido a esta classificação, é possível que os professores possam conhecer todas as informações sobre *notas* de alunos e os pais só possam conhecer as *médias* finais dos alunos pelos quais são responsáveis, fato que irá depender da PTP. Como a transparência, também, deve considerar a destinação correta das informações no andamento dos processos, existe a possibilidade de um ator ser destinatário deste tipo de informação, referente a outro processo, através de um processo do qual seja interno. No exemplo do empréstimo bancário, a solicitação de empréstimo é um processo no qual o cliente é interno, e a avaliação da solicitação do empréstimo é um outro processo, no qual o cliente é externo, porém, a informação de que a solicitação encontra-se sendo avaliada é de seu interesse, pois contribui com a transparência do processo *solicitação de empréstimo*.

Além destas características, os atores ocupam posições e desempenham papéis diferentes, que podem ser de acordo com a sua hierarquia dentro da organização (figura 3.3). Por isso, cada ator deverá ter sua própria credencial de acesso às informações do sistema. Esta credencial rotula os atores contribuindo com as futuras decisões quanto à destinação das informações.

O terceiro fator é que todo ator visualiza a uma determinada informação do sistema, através de pelo menos um de seus processos. Portanto, devem-se identificar os elementos (***Ator - Informação - Processo***) destas relações para que as possíveis destinações das informações sejam avaliadas e, conseqüentemente, ratificadas ou retificadas, contribuindo para evitar a ocorrência de conflitos no modelo final de transparência para o software.

O quarto fator implica na identificação e classificação das informações que deverão ser salvaguardadas do conhecimento indevido, contribuindo para evitar prejuízos à organização patrocinadora do sistema. Esta classificação depende exclusivamente da organização

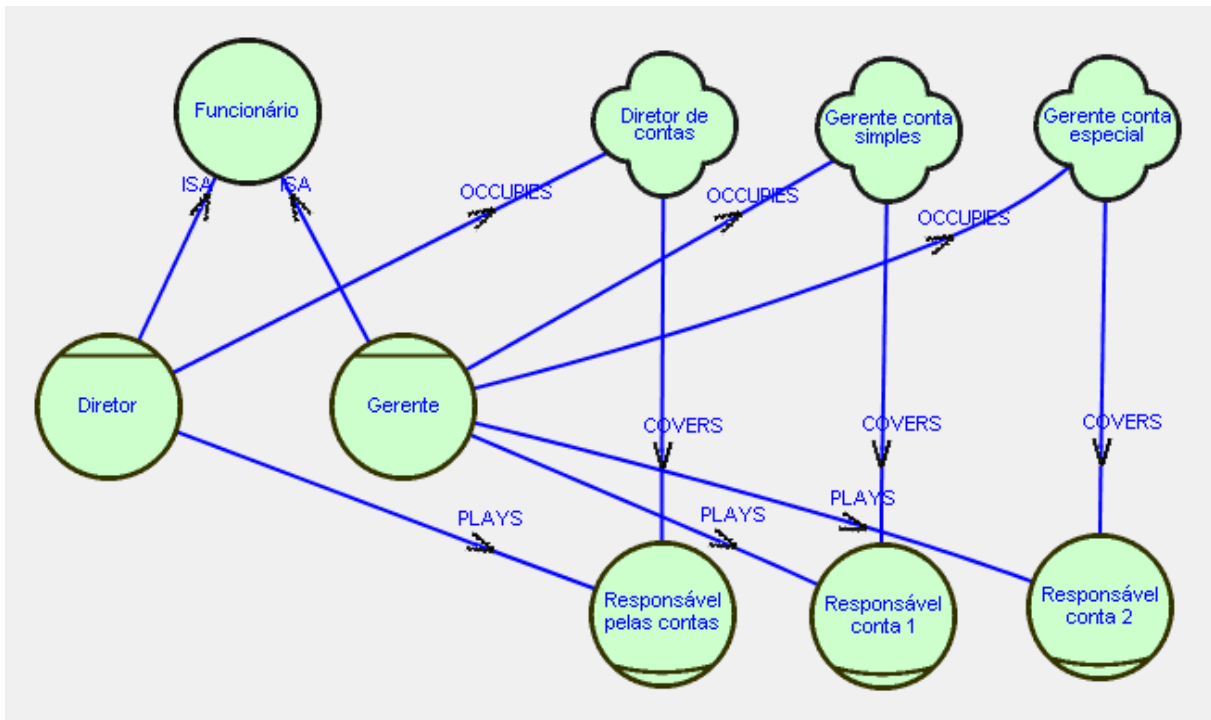


FIG. 3.3: Exemplo de um modelo SA para um banco.

patrocinadora, e deve seguir a taxonomia a seguir, que engloba de forma genérica todas as informações possíveis a um sistema.

- Ostensiva; e
- Restrita (não ostensiva).

Uma informação ostensiva é aquela que não necessita de restrições quanto ao seu conhecimento por qualquer ator do sistema. Já a informação restrita (não ostensiva) é aquela que necessita de restrições quanto ao seu conhecimento para alguns atores do sistema. Diferentemente da categorização dos atores, as categorias atribuídas às informações são excludentes, ou seja, não existe interseção entre os conjuntos (figura 3.4). Esta taxonomia pode ser mais especializada na categoria restrita (não ostensiva), dependendo da organização patrocinadora considerada (figura 3.4), devendo, também, ser utilizada na identificação da credencial de acesso às informações dos atores (ex.: FFAA).

O quinto fator é a verificação do momento no qual uma informação deve ser transparente a um ator para que tenha valia, pois não adianta o conhecimento de uma informação, em um momento do processo em curso, que não possa mais contribuir para a realização das tarefas do ator junto ao sistema, ou que não possibilite a satisfação de seus interesses

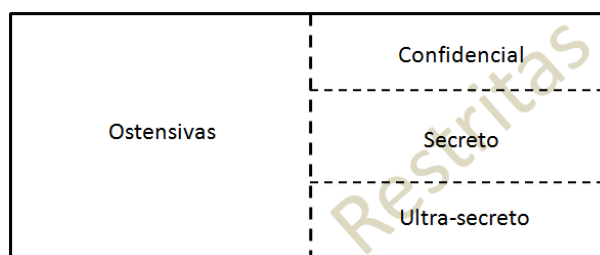


FIG. 3.4: Taxonomia indicada para as informações.

quanto ao acompanhamento do andamento de um processo. Isto ocorre, por exemplo, em um sistema de gerencia de pessoal, onde determinado funcionário é selecionado, junto ao sistema, para realizar um curso e somente durante o processo de matrícula que se verifica a falta da escolaridade mínima do funcionário selecionado, sendo necessário o retorno ao processo de seleção. Analisando esta situação, fica claro que a transparência das informações sobre escolaridade deve ocorrer no processo de seleção para cursos e não no processo de matrícula.

O sexto fator é a avaliação da justificativa do ator sobre seu interesse em conhecer determinada informação. Tal avaliação tem o propósito de reduzir o uso dos recursos do sistema para permitir a visualização de uma informação, devido às possíveis implicações que a transparência de uma informação pode causar aos outros RNF (ex.: desempenho e segurança).

O sétimo fator implica na determinação da forma na qual a informação deverá ser transparente a um determinado ator, ou seja, a aparência que a informação deverá ter ao ser visualizada pelo ator, possibilitando a compreensão de seu significado e garantindo a sua valia para o ator, sem comprometer a integridade do patrocinador. Por exemplo, um ator que necessita conhecer a idade de um funcionário de uma empresa, poderá visualizar a quantidade de anos que o funcionário possui (ex.: *32 anos*) ou a data de nascimento do mesmo (ex.: *01/01/1900*). Porém, se a data de nascimento for uma informação que não deva ser transparente a ele, o sistema será obrigado a destinar a primeira opção ao ator. Outro exemplo, é o caso da visualização, por um ator, que determinado funcionário da empresa possui o título de “Master of Science”, esta informação só terá o efeito desejado se o ator destinatário tiver conhecimento da Língua Inglesa e souber o que significa o título de mestrado.

O oitavo fator é a avaliação do impacto que poderá ser causado ao patrocinador, devido ao conhecimento, ou não, de uma informação por determinado ator. Isto porque,

os atores de um sistema poderão apresentar diversos interesses sobre as informações do sistema, porém o conhecimento de informações consideradas vitais a uma organização pode acarretar no seu comprometimento. Da mesma forma, o não conhecimento de uma informação vital, durante a execução de uma tarefa pelo ator, pode levá-lo a concluir erradamente um processo, causando prejuízos de diferentes naturezas.

3.2 O TIPO TRANSPARÊNCIA

Os requisitos não funcionais de transparência devem indicar a destinação das informações de um sistema para seus diferentes atores.

Como as informações de um sistema possuem origens e características diferentes, a transparência deverá focar cada caso. Desta forma, o refinamento do tipo permitirá sua subdivisão, contribuindo com o estudo e com as decisões a serem tomadas durante a modelagem.

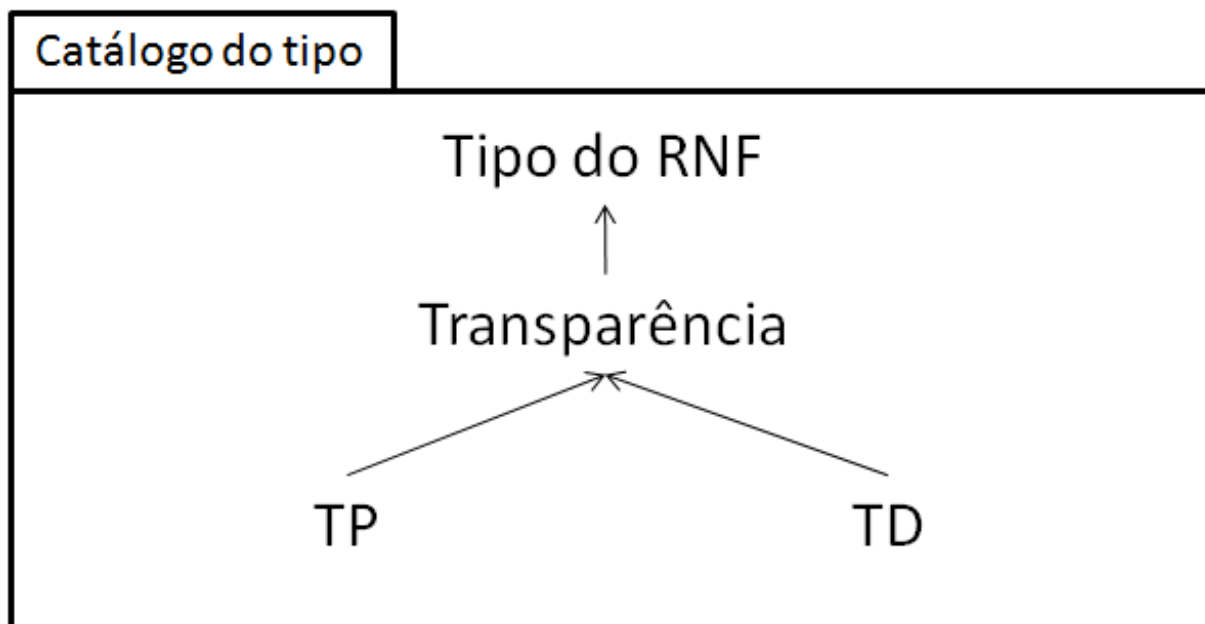


FIG. 3.5: Catálogo do tipo.

A figura 3.5 mostra a especialização inicial da transparência segundo a origem da informação em: transparência das informações originadas dos dados gerenciados pelo sistema ou simplesmente transparência dos dados do sistema (TD), e transparência das informações sobre os processos do sistema ou simplesmente transparência dos dados do processo (TP).

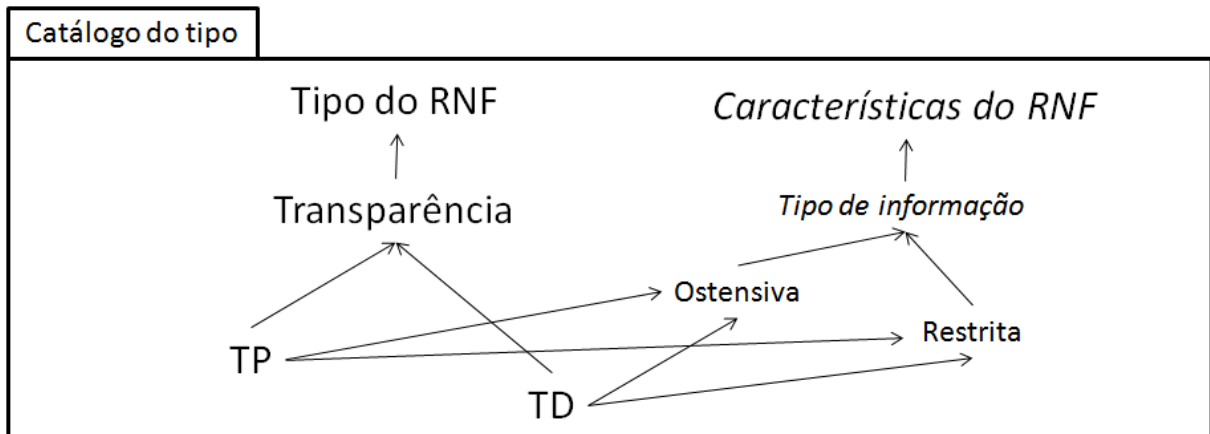


FIG. 3.6: Características das informações.

Estes dois subtipos podem ser mais específicos através de suas combinações com as características do RNF. A característica das informações considera a existência de informações classificadas como ostensivas e restritas (figura 3.6), gerando para a TD os subtipos (figura 3.7): transparência dos dados ostensivos do sistema (TDO) e transparência dos dados restritos do sistema (TDR).

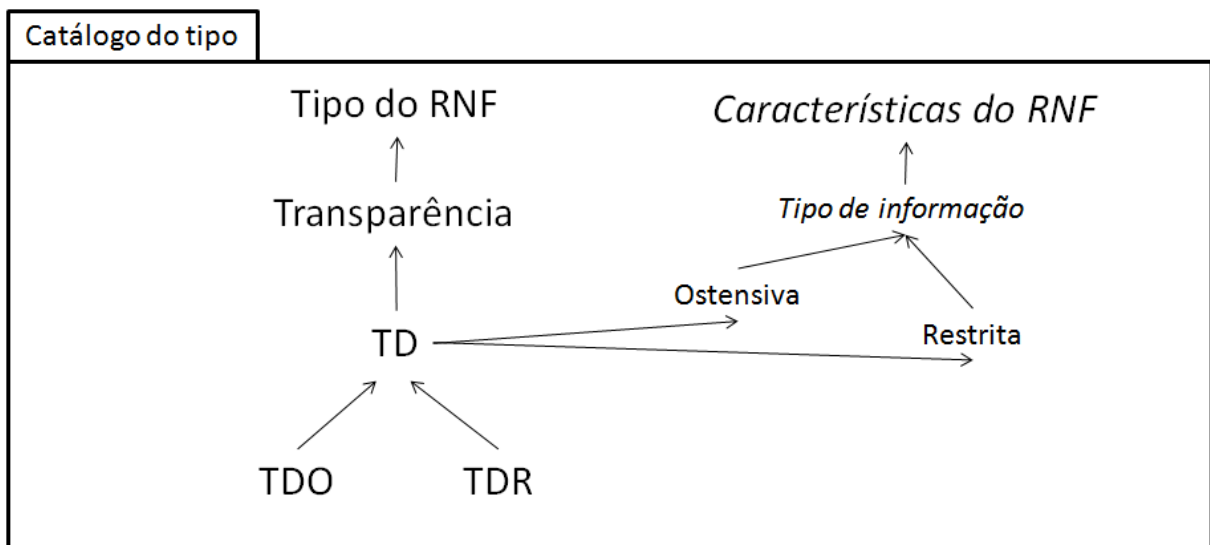


FIG. 3.7: Subtipos da TD.

Quanto à TP, são gerados os subtipos (figura 3.8): transparência dos dados ostensivos dos processos (TPO) e transparência dos dados restritos do processo (TPR).

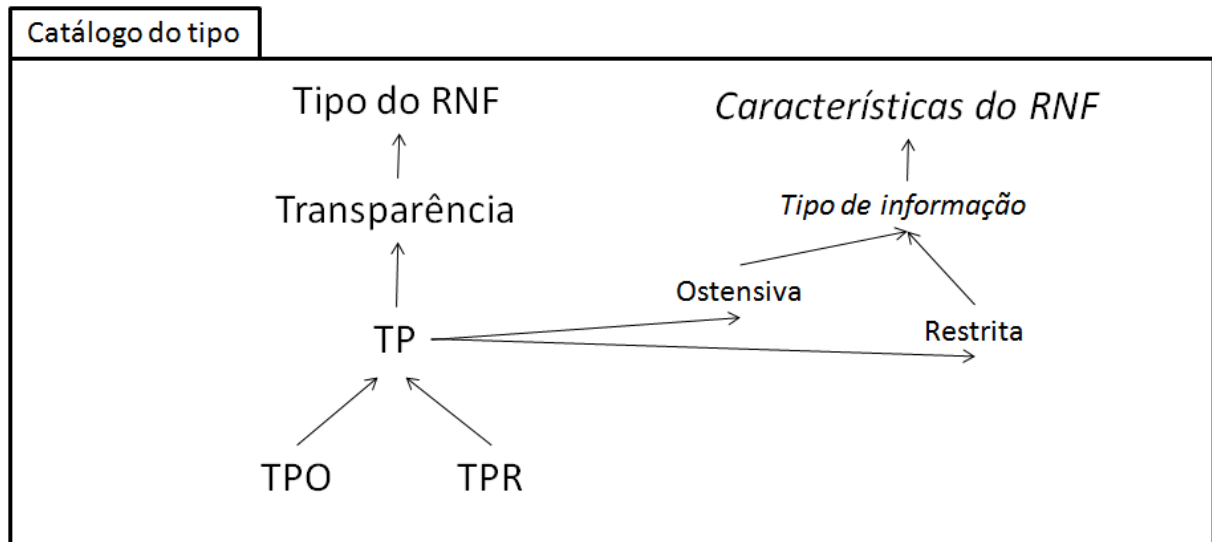


FIG. 3.8: Subtipos da TP.

3.3 PRINCÍPIOS PARA INCLUIR TRANSPARÊNCIA DAS INFORMAÇÕES EM SISTEMAS

A identificação dos conceitos de um RNF é fundamental para prover a terminologia necessária para expressá-lo e modelá-lo (CHUNG et al., 2000), fato que contribui com a definição destes para um software a ser desenvolvido.

Conforme as idéias apresentadas por Juran em (JURAN, 1988) e do Princípio de Pareto, a maior parte das prioridades que deverão ser consideradas pelos desenvolvedores serão referentes a poucos problemas a serem resolvidos.

Baseado nisso, identificou-se três princípios para apoiar o desenvolvimento da transparência em diferentes sistemas, fato que favorece o emprego e manutenção de um “baseline” específico para as decisões sobre a transparência dos mesmos (LEITE e OLIVEIRA, 1995) e (OLIVEIRA et al., 2008).

- Princípio da Correta Identificação dos Atores;
- Princípio da Correta Classificação das Informações; e
- Princípio da Correta Destinação das Informações.

3.3.1 PRINCÍPIO DA CORRETA IDENTIFICAÇÃO DOS ATORES

Este princípio baseia-se no fato de que os atores devem ser o foco do desenvolvimento de um sistema transparente, e que existe uma grande diversidade deles a ser considerada,

ou seja, os requisitos não funcionais de transparência visam a satisfação dos interesses dos atores, e não ao funcionamento do sistema, como no caso do RNF desempenho, apesar de serem definidos de acordo com a PTP. A correta identificação dos atores implica na definição de todas as características (categoria, posição, papel, credencial de acesso, etc) referentes a todos os atores envolvidos no ambiente do software a ser desenvolvido, possibilitando a construção de seus perfis e a compreensão de seus interesses. Os perfis podem ser referentes a um grupo de atores ou a um individualmente.

O perfil do ator deverá considerar inicialmente a sua relação com a organização, com o processo específico e com a forma de acesso ao sistema. Após isto, devem ser analisados os privilégios, limitações e obrigações inerentes as diferentes posições, papéis e a hierarquia, dos atores dentro da organização.

Por fim, deve ser atribuída aos atores identificados uma credencial de acesso a informação, de acordo com o entendimento da organização patrocinadora do sistema. Esta credencial deverá habilitar o ator a conhecer informações, segundo a classificação das mesmas, devendo referenciar a taxonomia definida para as informações. A habilitação para o conhecimento de informações com restrições ao conhecimento mais rígidas, automaticamente, habilita o conhecimento das informações com restrições mais brandas (figura 3.9).

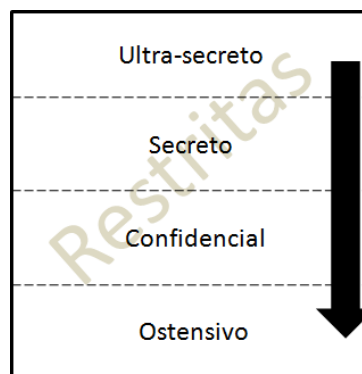


FIG. 3.9: Cobertura das credenciais de acesso.

A atribuição da credencial de acesso deve procurar avaliar os possíveis impactos, que podem ser causados a organização, pelo conhecimento de suas informações pelo ator considerado.

3.3.2 PRINCÍPIO DA CORRETA CLASSIFICAÇÃO DAS INFORMAÇÕES

O segundo princípio baseia-se no fato de que algumas informações não devem ser parcialmente, ou mesmo, totalmente transparentes, pois podem comprometer os interesses e a

integridade da organização patrocinadora (HOLZNER e HOLZNER, 2006). Portanto, o engenheiro de software terá que classificar as informações existentes no sistema de acordo com seu grau de importância para o patrocinador.

Para levantar as informações existentes no sistema a ser concebido, deverão ser mapeados todos os processos, inclusive suas etapas, tarefas e fluxo de informações associado, além de considerar a compreensão das necessidades dos interesses dos atores.

O fluxo das informações em um processo é pré-determinado e responsável por determinar como uma informação é criada, manipulada, gerenciada e destinada, internamente ou externamente a organização.

A classificação das informações deve seguir a taxonomia proposta, podendo ser evoluída conforme o caso. Tal classificação deve ser atribuída de acordo com as intenções do patrocinador, que também deverá ditar as restrições a serem consideradas.

3.3.3 PRINCÍPIO DA CORRETA DESTINAÇÃO DAS INFORMAÇÕES

Este princípio baseia-se no fato de que o conhecimento de qualquer informação, por qualquer ator de um sistema deve ser justificado. Esta justificativa deve ser fundamentada nas necessidades do ator em conhecer um ou mais itens de informação, para suprir seus interesses, contribuindo para a execução dos processos que participa.

Independente dos motivos que fundamentam a justificativa, a informação deve ser comprovadamente útil para o ator, que deverá ter capacidade de compreender seu significado e este deverá causar algum efeito desejado pelo ator. Portanto, se a informação não contribuir para a execução de alguma tarefa pelo ator ou possibilitar o acompanhamento de um processo, o mesmo não terá necessidade de conhecê-la. Da mesma forma, se o ator não tiver condições de perceber e compreender a informação, também não há sentido em a mesma ser destinada a ele.

Para verificar a destinação correta, deve-se confrontar os itens de informação levantados com os interesses dos atores, considerando sua credencial de acesso, a classificação dos itens de informação, o provável impacto causado a organização pelo conhecimento de uma informação por um ator e as limitações e obrigações impostas pelas regras do negócio do patrocinador do sistema. O confronto tem como objetivo definir a destinação das informações na PTP, evitando a existência de algum conflito.

Um conflito pode ser causado pelos seguintes fatos:

- O ator não necessita conhecer o item de informação destinado a ele, para satisfazer

seus interesses, contribuindo com a execução dos processos que participa ou com o acompanhamento de processos;

- O ator não possui capacidade para compreender o significado do item de informação destinado a ele, não tendo valia para o mesmo;
- O ator não possui credencial de acesso compatível com o item de informação de interesse;
- O item de informação de interesse não está relacionado ao processo que o ator participa ou do qual tem interesse em acompanhar; ou
- A PTP não admite tornar transparente um item de informação de interesse do ator.

A análise dos conflitos citados gera as seguintes possibilidades:

- Se o ator não necessita conhecer o item de informação, provavelmente seus interesses não foram compreendidos ou o item de informação foi destinado erradamente;
- Se o ator não possui capacidade para compreender o item de informação, provavelmente o item de informação foi destinado erradamente ou a forma do símbolo que o representa está errada;
- Se o ator não possui credencial de acesso compatível com a do item de informação, provavelmente a credencial do ator ou a classificação do item de informação está errada;
- Se o item não está relacionado com o processo, provavelmente o levantamento dos itens foi feito errado ou os interesses do ator não foram compreendidos; e
- Se a PTP não tornar o item de informação transparente, provavelmente deve-se reavaliar o impacto positivo e negativo causado pelo conhecimento do item de informação pelo ator.

Independente de qual seja o caso, a solução dos conflitos evitará a propagação de erros através dos modelos a serem construídos, contribuindo com a transparência adequada das informações do sistema.

3.4 NOTAÇÃO PARA AS METAS FLEXÍVEIS DE TRANSPARÊNCIA

Conforme (CHUNG et al., 2000), a notação de uma meta flexível deve refletir todas as informações necessárias ao seu entendimento pelo engenheiro de software.

No caso das metas flexíveis de transparência, define-se a seguinte notação:

TipoTransparência [Assunto, Destinatário, Processo]

Essa notação é iniciada pelo tipo da meta flexível, seguida pelo seu tópico. A primeira parte do tópico refere-se ao assunto tratado pela meta flexível, a segunda parte refere-se ao ator destinatário da informação relacionada com a meta flexível e a última parte refere-se ao processo no qual a informação está sendo transparente para o ator. Tal notação considera todos os elementos fundamentais a transparência conforme a figura 3.1.

3.5 MÉTODOS DE REFINAMENTO

O refinamento das metas flexíveis de transparência poderá ser feito em todos os seus componentes, ou seja, no tipo e no tópico (figura 3.10).

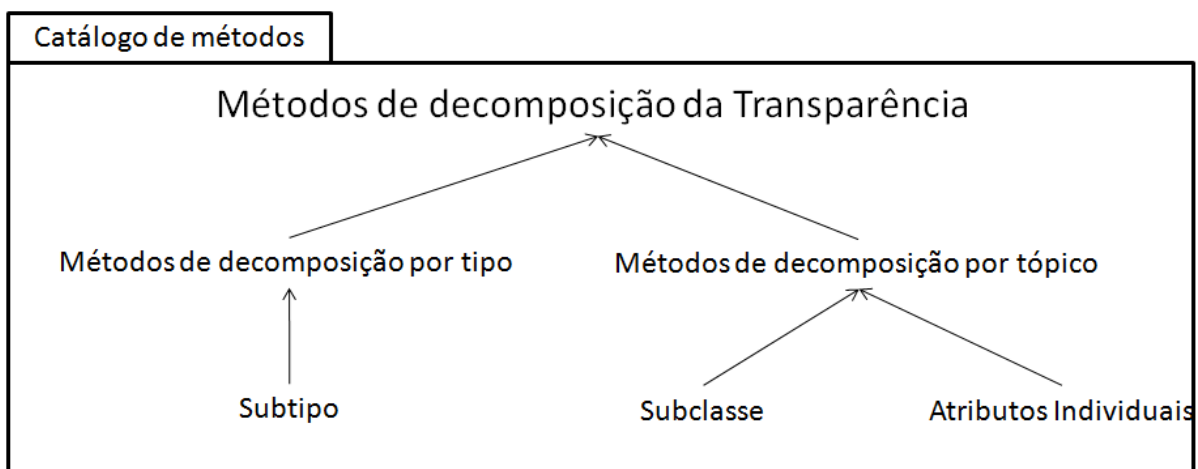


FIG. 3.10: Métodos de refinamento da Transparência.

3.5.1 MÉTODOS DE DECOMPOSIÇÃO POR TIPO

A decomposição por subtipo refina as metas flexíveis através do refinamento de seu tipo. A forma genérica deste método em relação à origem das informações é:

TD[Assunto, Destinatário, Processo]
AND
TP[Assunto, Destinatário, Processo]
SATISFICE
Transparência[Assunto, Destinatário, Processo]

O método indica que satisfazendo suficientemente (*Satisfice*³) a TD e a TP, de algum tópico em relação a um ator de um processo, irá satisfazer suficientemente a transparência das informações sobre o tópico para o determinado ator do processo.

No caso de um sistema para gerenciar as contas de um restaurante, pode-se decompor a meta flexível “*transparência da conta para o cliente no pagamento*” nos subtipos “*transparência dos dados da conta para o cliente no pagamento*” e “*transparência do processo de pagamento da conta para o cliente*” (figura 3.11). Assim, tem-se a seguinte instância:

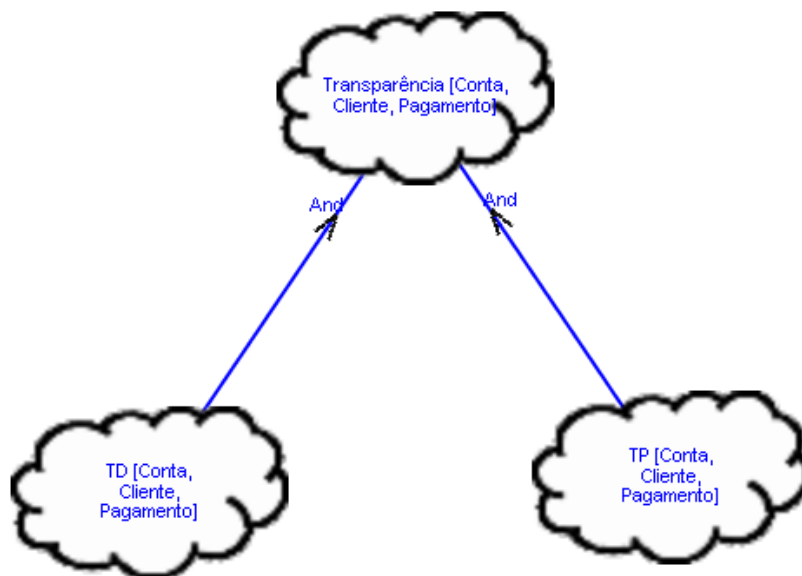


FIG. 3.11: Exemplo de refinamento por tipo.

TD[Conta, Cliente, Pagamento]
AND
TP[Conta, Cliente, Pagamento]
SATISFICE

³Este conceito foi definido em (SIMON, 1996).

Transparência[Conta, Cliente, Pagamento]

Tais metas flexíveis podem ser satisfeitas pelo conhecimento do valor final da conta e dos diferentes “status” possíveis ao processo de pagamento, pelo ator. No caso do pagamento da conta através de cartão de crédito, os “status” podem ser: conta pronta para pagamento, aguardando confirmação do cartão de crédito e pagamento concluído.

Considerando a classificação das informações, a forma genérica do método de refinamento por tipo é:

- Para a TD:

TDO[Assunto, Destinatário, Processo]
AND
TDR[Assunto, Destinatário, Processo]
SATISFICE
TD[Assunto, Destinatário, Processo]

- Para a TP:

TPO[Assunto, Destinatário, Processo]
AND
TPR[Assunto, Destinatário, Processo]
SATISFICE
TP[Assunto, Destinatário, Processo]

Seguindo o raciocínio do exemplo anterior, pode-se decompor a meta flexível “*transparência dos dados da conta para o cliente no pagamento*” nos subtipos “*transparência dos dados ostensivos da conta para o cliente no pagamento*” e “*transparência dos dados restritos da conta para o cliente no pagamento*” (figura 3.12). Assim, tem-se a seguinte instância:

TDO[Conta, Cliente, Pagamento]
AND
TDR[Conta, Cliente, Pagamento]
SATISFICE
TD[Conta, Cliente, Pagamento]

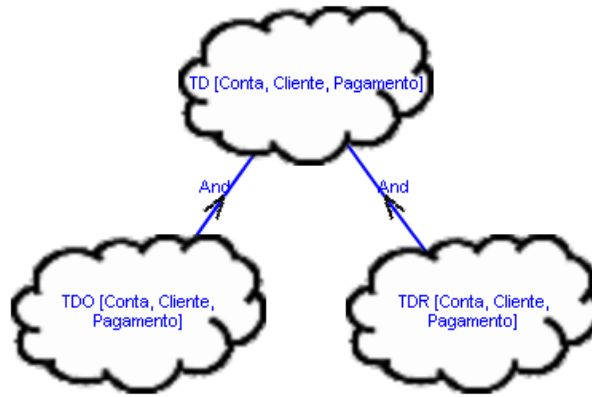


FIG. 3.12: Exemplo de refinamento por tipo para a TD.

Da mesma forma, a “*transparência do andamento do processo de pagamento da conta para o cliente*” pode ser refinado nos subtipos “*transparência das informações ostensivas sobre o processo de pagamento da conta para o cliente*” e “*transparência das informações restritas sobre o processo de pagamento da conta para o cliente*” (figura 3.13).

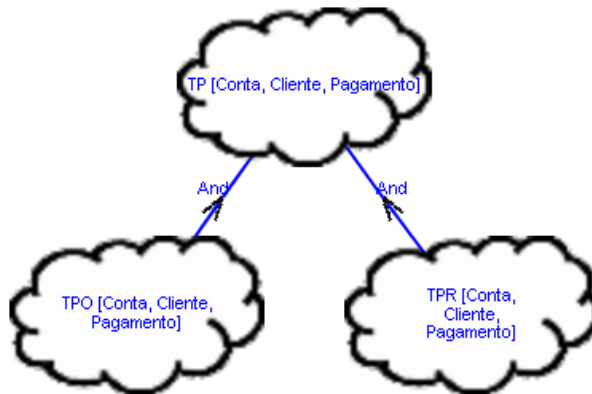


FIG. 3.13: Exemplo de refinamento por tipo para a TP.

TPO[Conta, Cliente, Pagamento]
AND
TPR[Conta, Cliente, Pagamento]
SATISFICE
TP[Conta, Cliente, Pagamento]

3.5.2 MÉTODOS DE DECOMPOSIÇÃO POR TÓPICO

Tais métodos utilizam o tópico para decompor as metas flexíveis. Como os tópicos das metas flexíveis de transparência são formados por três partes diferentes, a aplicação dos

métodos deve considerar cada parte individualmente, conforme o caso.

O método genérico de decomposição por subclasse (CHUNG et al., 2000) pode ser aplicado a todas as partes do tópico. Assim, pode-se definir as seguintes formas genéricas:

- Para o Assunto do tópico:

a) Definição:

TipoTransparência[Subclasse₁, Destinatário, Processo]
AND
TipoTransparência[Subclasse₂, Destinatário, Processo]
AND ...
SATISFICE
TipoTransparência[Classe, Destinatário, Processo]

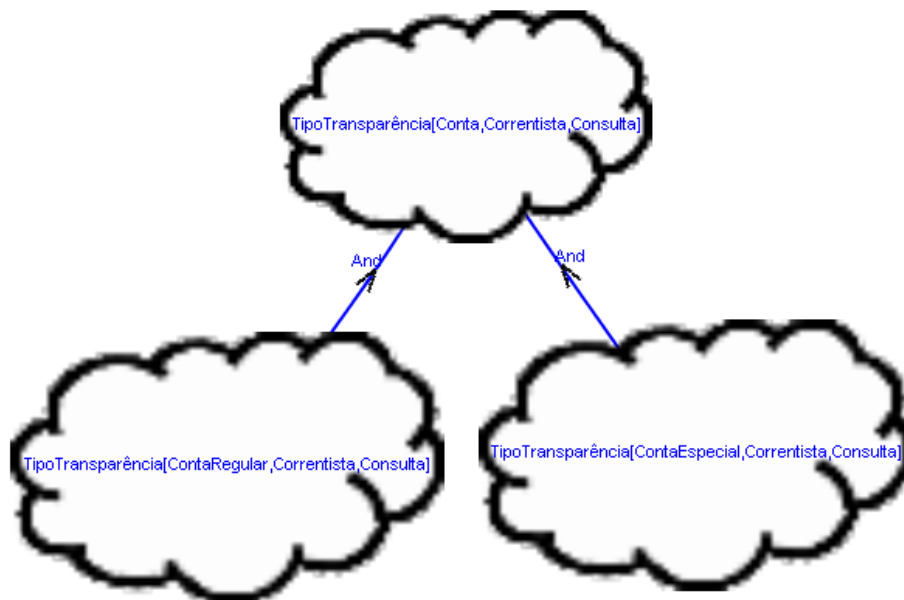


FIG. 3.14: Exemplo de refinamento por subclasse no assunto.

b) Exemplo (figura 3.14):

TipoTransparência[ContaRegular, Correntista, Consulta]
AND
TipoTransparência[ContaEspecial, Correntista, Consulta]
SATISFICE
TipoTransparência[Conta, Correntista, Consulta]

- Para o Destinatário do tópico:

a) Definição:

TipoTransparência[Assunto, Subclasse₁, Processo]
 AND
TipoTransparência[Assunto, Subclasse₂, Processo]
 AND ...
 SATISFICE
TipoTransparência[Assunto, Classe, Processo]

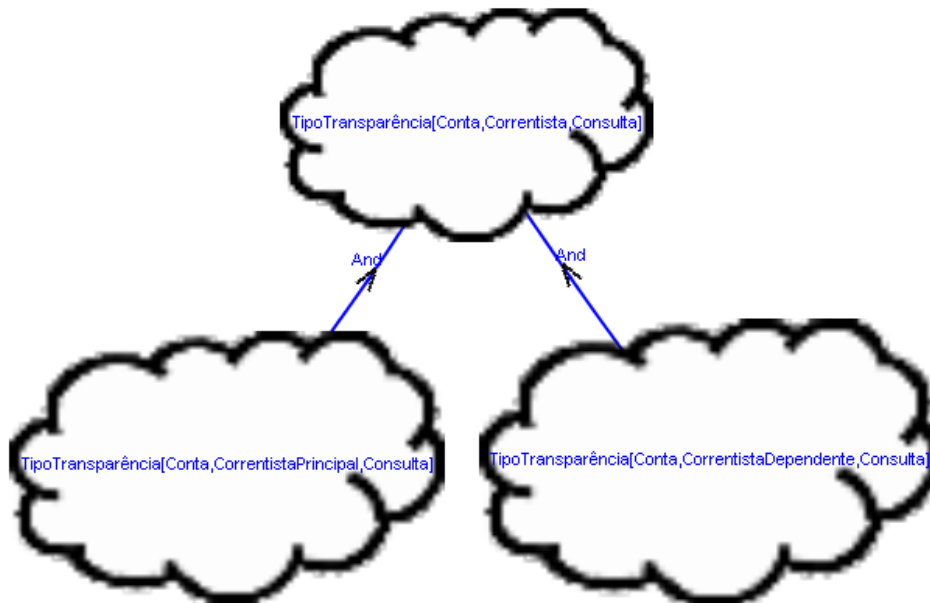


FIG. 3.15: Exemplo de refinamento por subclasse no destinatário.

b) Exemplo (figura 3.15):

TipoTransparência[Conta, CorrentistaPrincipal, Consulta]
 AND
TipoTransparência[Conta, CorrentistaDependente, Consulta]
 SATISFICE
TipoTransparência[Conta, Correntista, Consulta]

- Para o Processo do tópico:

a) Definição:

TipoTransparência[Assunto, Destinatário, Subclasse1]
 AND
TipoTransparência[Assunto, Destinatário, Subclasse2]
 AND ...
 SATISFICE
TipoTransparência[Assunto, Destinatário, Classe]

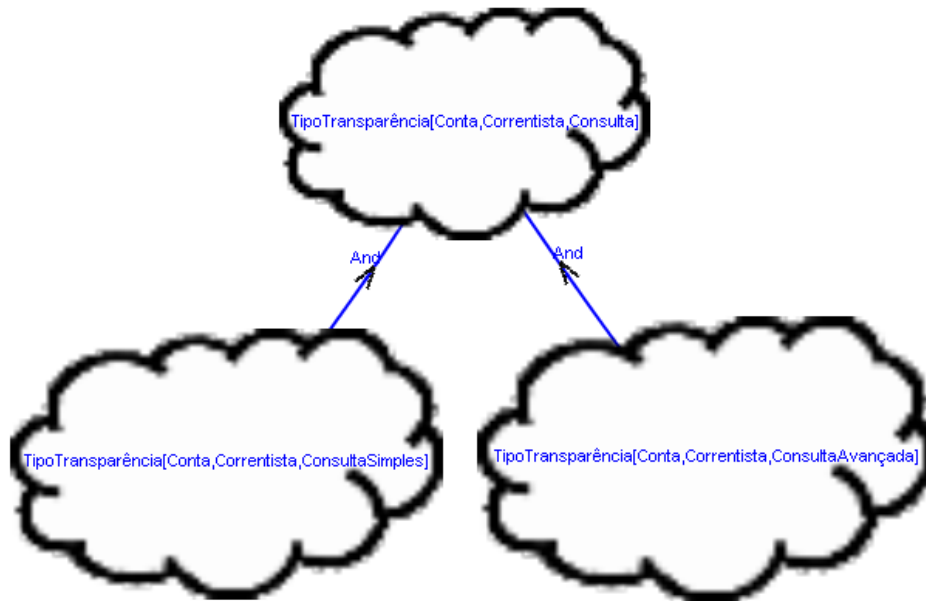


FIG. 3.16: Exemplo de refinamento por subclasse no processo.

b) Exemplo (figura 3.16):

TipoTransparência[Conta, Correntista, ConsultaSimples]
 AND
TipoTransparência[Conta, Correntista, ConsultaAvançada]
 SATISFICE
TipoTransparência[Conta, Correntista, Consulta]

De forma similar, o método genérico de decomposição por atributos individuais (CHUNG et al., 2000) pode ser aplicado ao assunto do tópico, visto que uma informação pode ser composta por diferentes itens de informação, os quais representam os atributos da informação.

a) Definição:

TipoTransparência[Assunto.Atributo1, Destinatário, Processo]
 AND
TipoTransparência[Assunto.Atributo2, Destinatário, Processo]
 AND ...
 SATISFICE
TipoTransparência[Assunto, Destinatário, Processo]

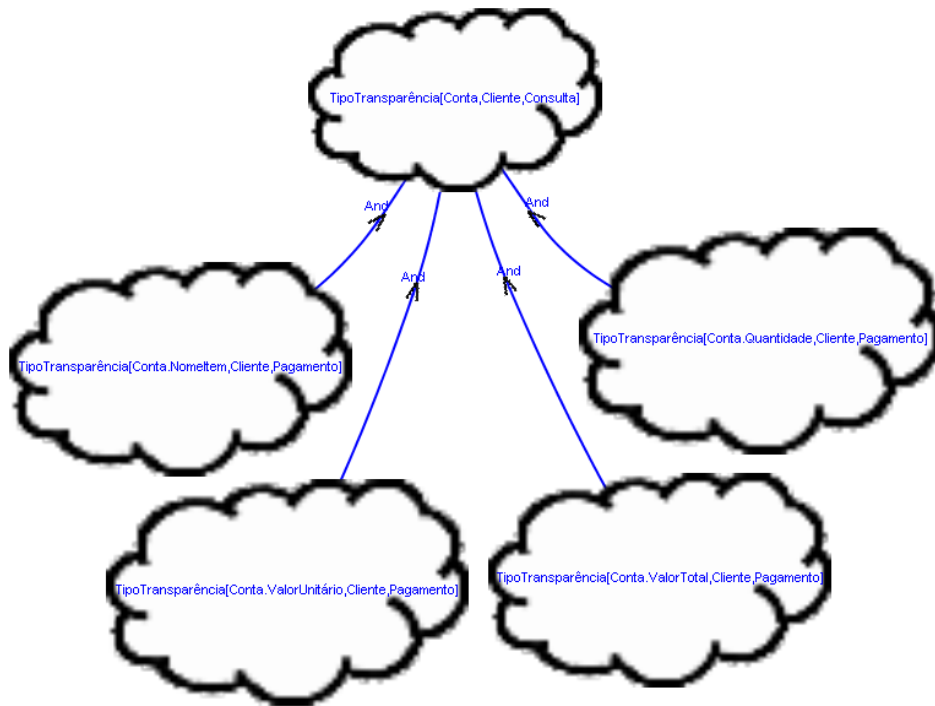


FIG. 3.17: Exemplo de refinamento por atributos individuais.

b) Exemplo (figura 3.17):

TipoTransparência[Conta.NomeItem, Cliente, Pagamento]
 AND
TipoTransparência[Conta.Quantidade, Cliente, Pagamento]
 AND
TipoTransparência[Conta.ValorTotal, Cliente, Pagamento]
 AND
TipoTransparência[Conta.ValorUnitário, Cliente, Pagamento]
 SATISFICE
TipoTransparência[Conta, Cliente, Pagamento]

3.5.3 MÉTODOS DE OPERACIONALIZAÇÃO

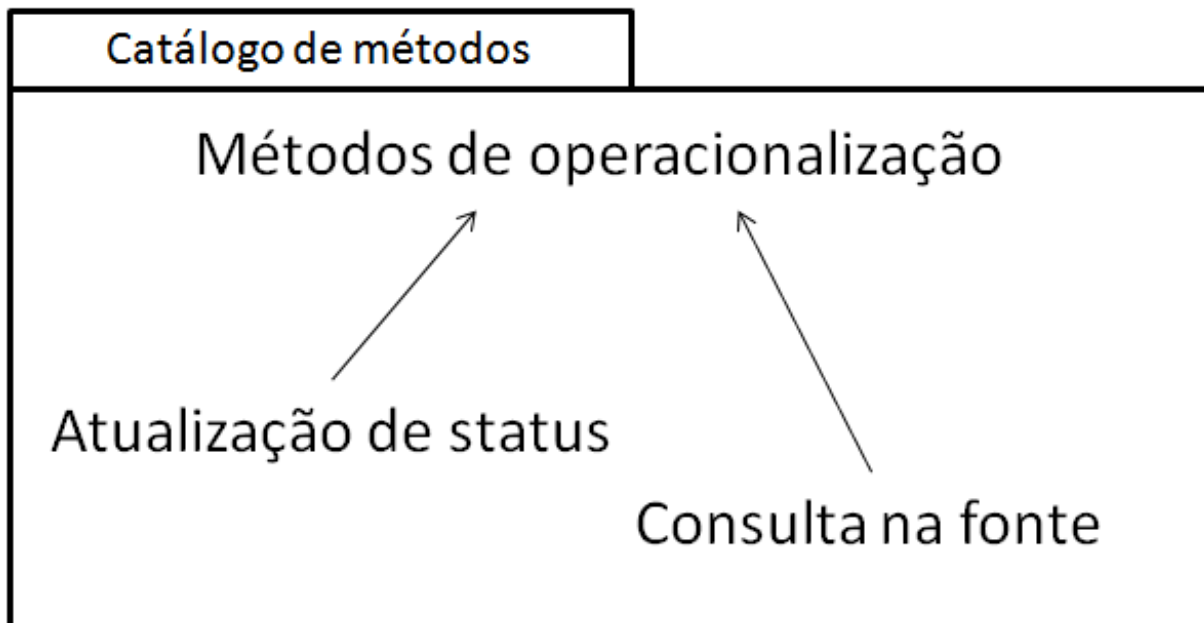


FIG. 3.18: Catálogo de métodos de operacionalização.

- Atualização de status
 - Definição:
Este método requer que o status do processo seja disponibilizado a cada alteração ocorrida.
 - Representação:
AtualizaStatus[Processo] *MAKES* TP[Assunto, Destinatário, Processo].

- Consulta na fonte
 - Definição:
Este método requer que o ator que tem interesse em alguma informação, adquira-a junto a fonte.
 - Representação:
ConsultaFonte[Assunto] *MAKES* TD[Assunto, Destinatário, Processo].

3.5.4 MÉTODOS DE ARGUMENTAÇÃO

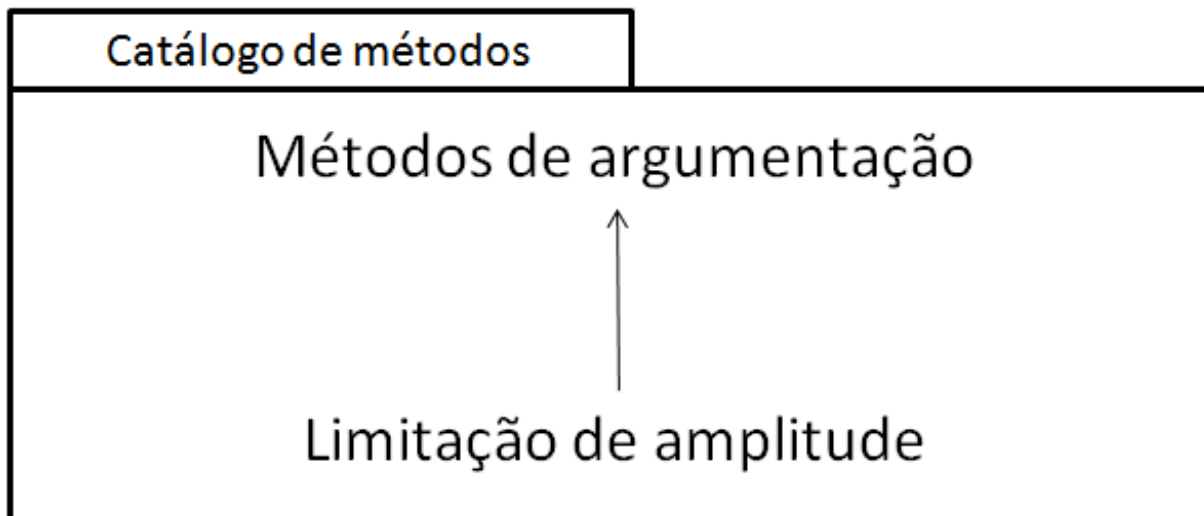


FIG. 3.19: Catálogo de métodos de argumentação.

- Limitação de amplitude

- a) Definição:

- Método usado para limitar a transparência de informações, complementando a destinação das informações de acordo com as características do domínio considerado.

- b) Exemplo (figura 3.20):

- LimitaçãoAmplitude:“A conta só poderá ser transparente para o gerente responsável por ela”.

3.6 CONCLUSÕES

Este capítulo, seguindo a estrutura de catalogação definida por (CHUNG et al., 2000), apresentou e discutiu conceitos específicos ao RNF Transparência, os elementos fundamentais a transparência, a taxonomia proposta para a divisão dos possíveis atores, a taxonomia indicada para a divisão das informações do sistema, os oito fatores a serem considerados pelos desenvolvedores na busca da transparência no software e os princípios para inclusão da transparência em sistemas.

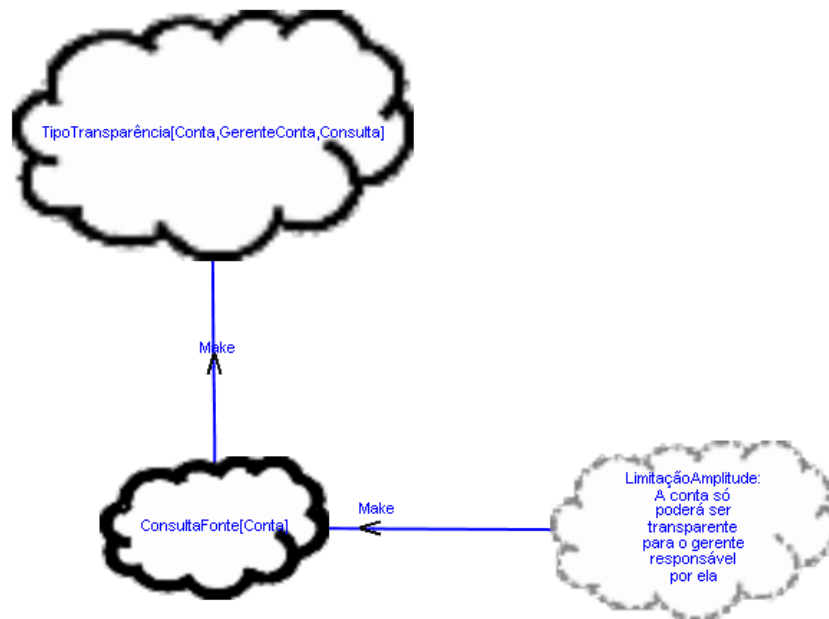


FIG. 3.20: Exemplo de argumentação.

Além disso, foram propostos e exemplificados métodos para o refinamento das metas flexíveis de transparência, possibilitando a compreensão e registro das necessidades do sistema, conforme os estudos de casos apresentados no capítulo seguinte.

4 ESTUDO DE CASO

Este capítulo tem o objetivo de apresentar dois estudos de casos utilizados para demonstrar a aplicação da modelagem do RNF Transparência de acordo com o catálogo proposto. Os casos escolhidos são o “EC System” (Expert Committee System - ECS) e o Controle do Caixa do Restaurante. O motivo da escolha destes casos é o fato de possuírem tamanho e complexidade adequados ao escopo dessa dissertação, além da existência de modelos intencionais prontos para serem utilizados, disponibilizados em exemplos de outros trabalhos (OLIVEIRA et al., 2006), (CUNHA, 2007) e (OLIVEIRA, 2008).

Os requisitos de um sistema que são referentes a transparência das informações tratam de entidades informacionais. O entendimento destas entidades deve considerar a forma como são produzidas e manuseadas. Assim, os modelos SR são adequados para apoiar a abordagem da transparência nos estudos realizados.

Como o catálogo apresentado no capítulo anterior (capítulo 3) não define uma metodologia específica para se lidar com a transparência das informações durante a modelagem de um sistema, o desenvolvimento dos estudos de casos foram feitos baseados nos modelos dos sistemas apresentados em (OLIVEIRA, 2008).

4.1 SISTEMA EXPERT COMMITTEE

O ECS tem o objetivo de apoiar o gerenciamento de submissões e revisões de artigos submetidos a uma conferência ou workshop (OLIVEIRA, 2008).

A modelagem deste sistema por (OLIVEIRA, 2008) definiu as seguintes SDsituations:

- Formação de Comitê
- Submissão de Artigos
- Aceitação de Propostas
- Revisão de Artigos
- Votação de Conflitos
- Recepção de Camera-Ready

Para facilitar o desenvolvimento deste estudo, os possíveis processos do ECS foram considerados conforme as SDSituations citadas⁴. De acordo com o trabalho de referência, foram definidos os seguintes atores para o sistema:

- Pesquisador
- Autor
- Chair
- Coordenador Geral
- Revisor
- Membro do Comitê de Programa

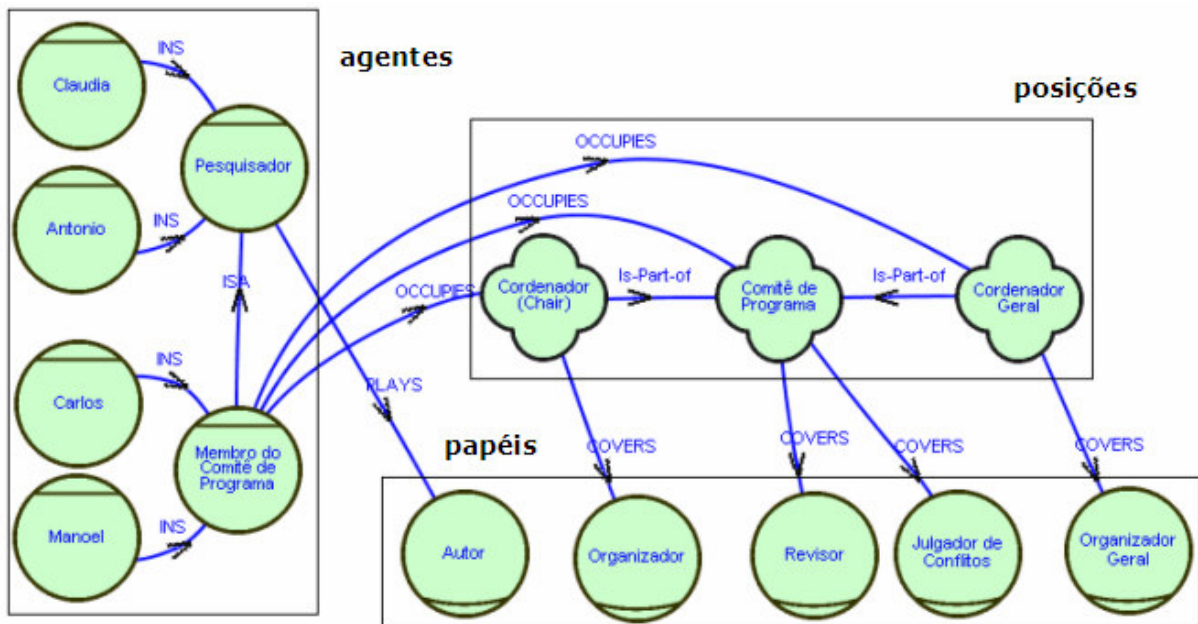


FIG. 4.1: Modelo SA para o ECS retirado de (OLIVEIRA, 2008).

Analisando as informações disponibilizadas e o modelo SA (figura 4.1) referente ao ECS, verifica-se que um Autor de artigo é um Pesquisador que submete pelo menos um artigo de sua autoria a conferência, para sua revisão e aceitação, se for o caso. Além disso,

⁴SDSituations e os processos de um sistema possuem significados diferentes. Cada processo do sistema pode conter uma ou mais SDSituations ou vice-versa.

para tornar-se um membro da conferência, o Pesquisador deve aceitar o convite enviado pelo Coordenador Geral.

O Chair e o Coordenador Geral são posições ocupadas por membros diferentes da conferência. Estas posições só podem ser ocupadas por um único membro, não definido através de convite.

A posição Comitê de Programa cobre os papéis de Revisor de artigos e de Julgador de conflitos. Esta última é representada pelo ator Membro do Comitê.

Classificando os atores do sistema quanto aos processos, obtem-se a tabela 4.1.

TAB. 4.1: Classificação dos atores quanto aos processos considerados no ECS.

Ator	Formação	Submissão	Aceitação	Revisão	Votação	Recepção
Autor	Externo	Interno	Externo	Externo	Externo	Interno
Chair	Interno	Interno	Interno	Interno	Interno	Interno
Coordenador	Interno	Externo	Externo	Externo	Externo	Externo
Revisor	Externo	Externo	Interno	Interno	Externo	Externo
Membro do comitê	Externo	Externo	Externo	Externo	Interno	Externo
Pesquisador	Interno	Externo	Externo	Externo	Externo	Externo

Quanto a organização e ao acesso, obtem-se a tabela 4.2.

TAB. 4.2: Classificação dos atores quanto a organização e acesso ao sistema no ECS.

Ator	Organização	Acesso
Autor	Externo	Externo
Chair	Interno	Interno
Coordenador	Interno	Interno
Revisor	Interno	Interno
Membro do comitê	Interno	Interno
Pesquisador	Externo	Externo

De acordo com a PTP pode-se atribuir a seguinte credencial de acesso as informações para os atores do sistema (tabela 4.3).

TAB. 4.3: Credencial de acesso as informações dos atores do ECS.

Ator	Credencial de acesso
Autor	Ostensiva
Chair	Restrita
Coordenador	Restrita
Revisor	Restrita
Membro do comitê	Restrita
Pesquisador	Ostensiva

Verificando os modelos SR referentes as SDSituations definidas para o ECS, é possível identificar as entidades informacionais de interesse para os atores e respectivas justificativas para o conhecimento das mesmas.

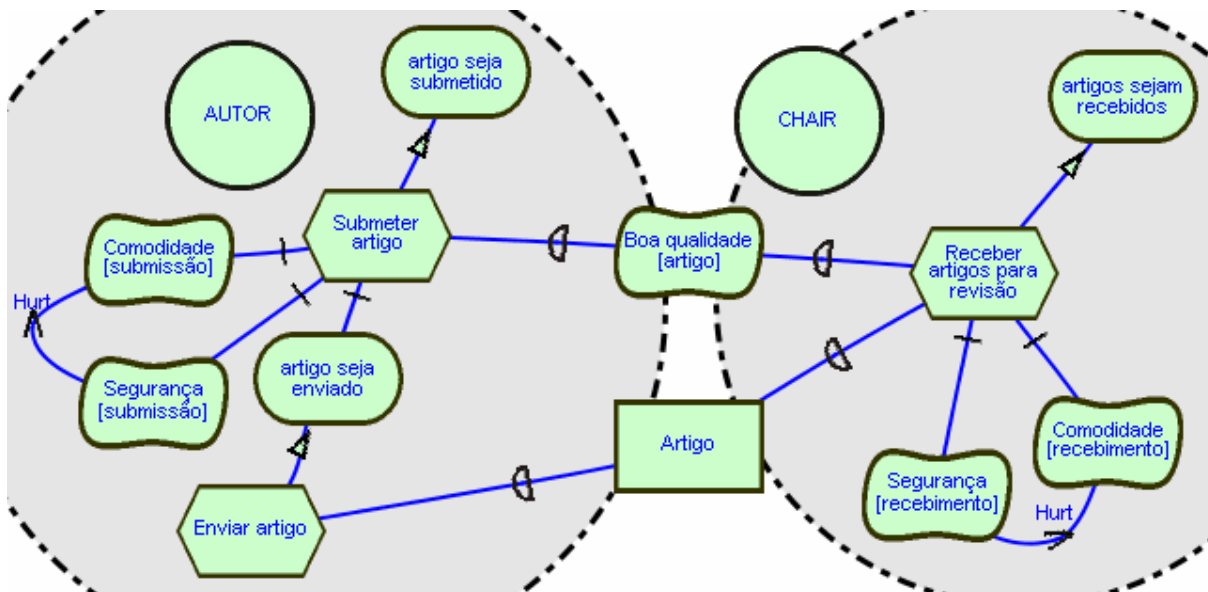


FIG. 4.2: Modelo SR para a SDsituation Submissão de Artigos.

Para simplificar a demonstração, serão utilizados apenas os modelos das SDsituations Submissão de Artigos (figura 4.2), Aceitação de Propostas (figura 4.3) e Revisão de Artigos (figura 4.4).

Além dos elementos (atores, metas, metas flexíveis, tarefas, recursos e links) explícitos nos modelos apresentados, pode-se identificar na figura 4.2 que o Chair tem interesse em conhecer o artigo submetido pelo Autor para que possa recebê-lo e enviá-lo para revisão, e que o Autor tem interesse em conhecer o status do processo de submissão para ter a confirmação de que o artigo foi recebido.

TAB. 4.4: Destinação das informações no processo Submissão.

Informação	Item de informação	Processo	Destinatário
Artigo submetido	Nome	Submissão	Chair
Artigo submetido	Instituição	Submissão	Chair
Artigo submetido	Contato	Submissão	Chair
Artigo submetido	Título	Submissão	Chair
Artigo submetido	Resumo	Submissão	Chair
Artigo submetido	Corpo	Submissão	Chair
Artigo submetido	Área	Submissão	Chair
Recebimento do artigo	X	Submissão	Autor

O artigo submetido é composto pelos nomes dos autores, as instituições que os autores são vinculados, o contato dos autores, o título do artigo, o resumo do artigo, o corpo do artigo e a área de interesse na qual o artigo pertence. O status do processo é representado somente pela informação de que o artigo foi recebido. Assim, pode-se definir a tabela de destinação das informações (tabela 4.4), considerando os elementos

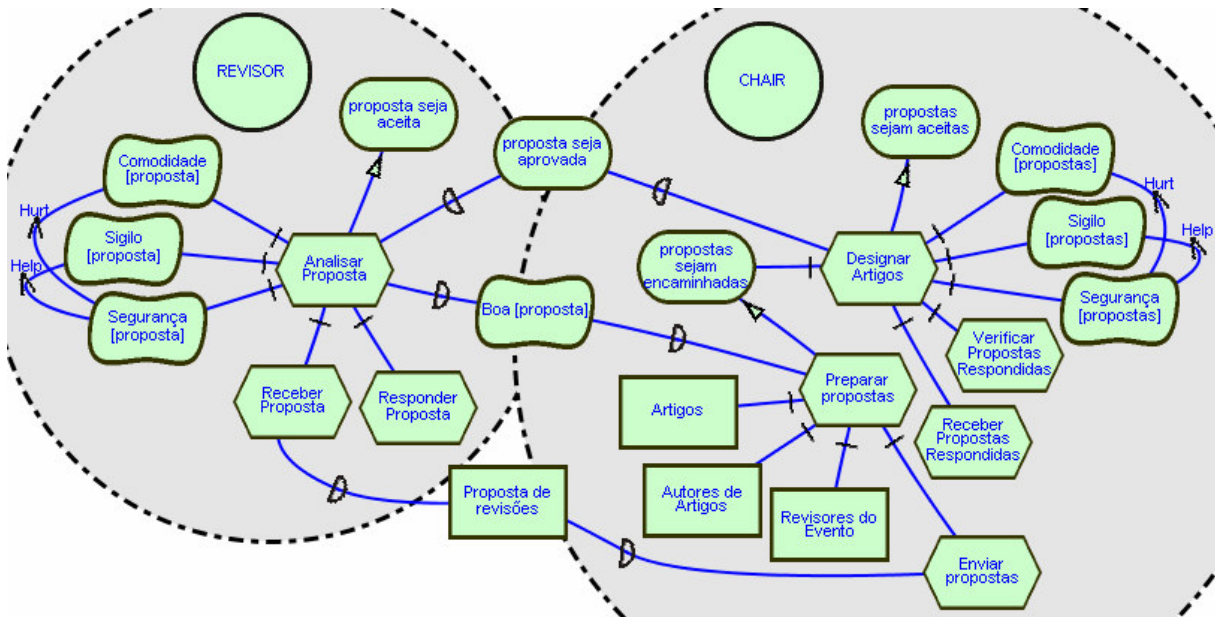


FIG. 4.3: Modelo SR para a SDsituation Aceitação de Propostas.

fundamentais (figura 3.1) presentes na SDsituation Submissão de Artigos.

Na figura 4.3 identifica-se que o Chair tem interesse em conhecer as informações dos artigos submetidos e dos revisores do evento para que possa preparar as propostas de revisões e enviá-las aos Revisores, além do status do processo de aceitação para poder saber quando o Revisor iniciou e finalizou a análise da proposta.

TAB. 4.5: Destinação das informações no processo Aceitação.

Informação	Item de informação	Processo	Destinatário
Artigo submetido	Nome	Aceitação	Chair
Artigo submetido	Instituição	Aceitação	Chair
Artigo submetido	Contato	Aceitação	Chair
Artigo submetido	Título	Aceitação	Chair
Artigo submetido	Resumo	Aceitação	Chair
Artigo submetido	Corpo	Aceitação	Chair
Artigo submetido	Área	Aceitação	Chair
Lista de revisores	Nome	Aceitação	Chair
Lista de revisores	Instituição	Aceitação	Chair
Lista de revisores	Área	Aceitação	Chair
Lista de revisores	Contato	Aceitação	Chair
Proposta de revisão	Nome	Aceitação	Revisor
Proposta de revisão	Instituição	Aceitação	Revisor
Proposta de revisão	Contato	Aceitação	Revisor
Proposta de revisão	Título	Aceitação	Revisor
Proposta de revisão	Resumo	Aceitação	Revisor
Proposta de revisão	Corpo	Aceitação	Revisor
Proposta de revisão	Área	Aceitação	Revisor
Proposta de revisão	Prazo	Aceitação	Revisor
Andamento da aceitação	X	Aceitação	Chair

As informações dos artigos submetidos estão definidas na tabela 4.4, já as informações dos revisores é composta pelos nomes dos revisores seleccionados, as instituições que estão

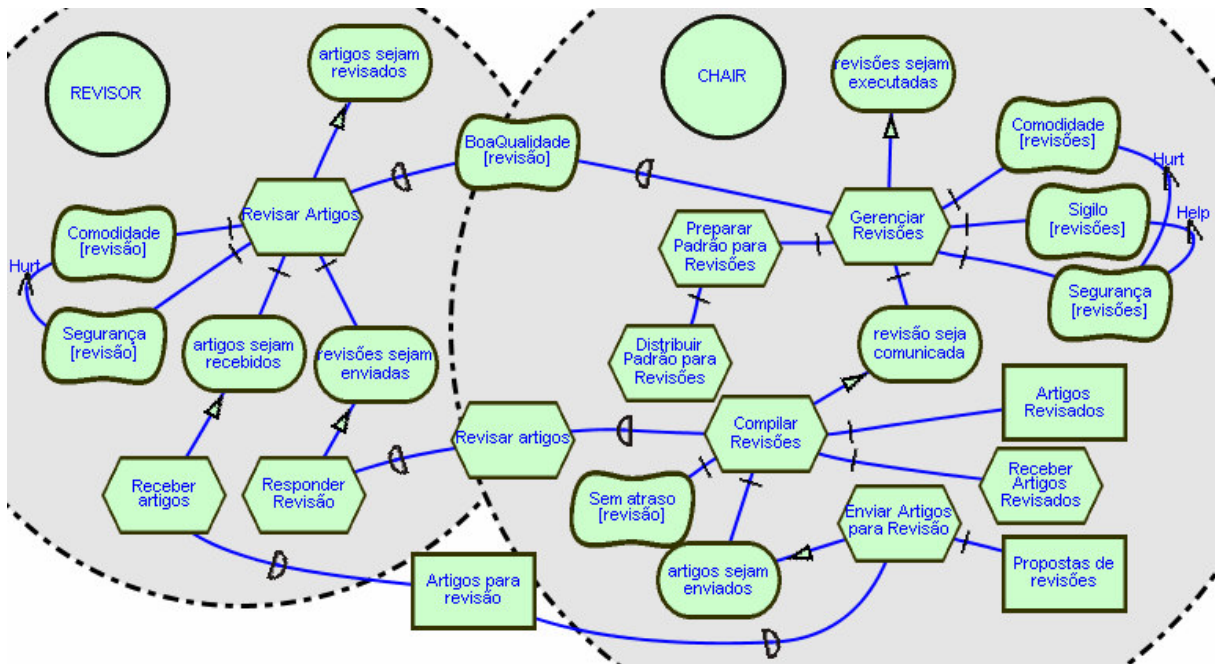


FIG. 4.4: Modelo SR para a SDsituation Revisão de Artigos.

vinculados, as áreas de interesse de cada revisor e o contato dos mesmos. O status do processo é representado pelas informações de que a análise encontra-se em andamento e finalizada. O Revisor tem interesse em conhecer as informações da proposta de revisão, que é composta pelas informações do artigo e o prazo para revisão. Com estas novas informações, define-se uma outra tabela de destinação das informações, agora considerando os elementos fundamentais presentes na SDsituation Aceitação de Propostas (tabela 4.5).

Analisando a figura 4.4 identifica-se que o Revisor tem interesse em conhecer as informações do artigo enviado para revisão, para que possa realizar a revisão do mesmo. O artigo enviado para revisão, a princípio é composto pelas mesmas informações que o artigo submetido. O Chair tem interesse em conhecer as informações do artigo revisado para que possa compilar as revisões e tem interesse em conhecer as informações das propostas respondidas para que possa enviar os artigos para revisão. O interesse do Chair em conhecer o status do processo de revisão, também é representado pelas informações de que a revisão encontra-se em andamento ou finalizada.

O artigo revisado possui além dos itens de informação já definidos para o artigo, o item de informação revisão do artigo. Da mesma forma, a proposta respondida possui o item de informação resposta do revisor, além dos itens de informação já definidos para a proposta de revisão. Assim, pode-se definir outra tabela de destinação das informações

(tabela 4.6), considerando os elementos fundamentais presentes na SDSituation Revisão de Artigos.

TAB. 4.6: Destinação das informações no processo Revisão.

Informação	Item de informação	Processo	Destinatário
Artigo para revisão	Nome	Revisão	Revisor
Artigo para revisão	Instituição	Revisão	Revisor
Artigo para revisão	Contato	Revisão	Revisor
Artigo para revisão	Título	Revisão	Revisor
Artigo para revisão	Resumo	Revisão	Revisor
Artigo para revisão	Corpo	Revisão	Revisor
Artigo para revisão	Área	Revisão	Revisor
Artigo revisado	Nome	Revisão	Chair
Artigo revisado	Instituição	Revisão	Chair
Artigo revisado	Contato	Revisão	Chair
Artigo revisado	Título	Revisão	Chair
Artigo revisado	Resumo	Revisão	Chair
Artigo revisado	Corpo	Revisão	Chair
Artigo revisado	Área	Revisão	Chair
Artigo revisado	Revisão	Revisão	Chair
Proposta de revisão	Nome	Revisão	Chair
Proposta de revisão	Instituição	Revisão	Chair
Proposta de revisão	Contato	Revisão	Chair
Proposta de revisão	Título	Revisão	Chair
Proposta de revisão	Resumo	Revisão	Chair
Proposta de revisão	Corpo	Revisão	Chair
Proposta de revisão	Área	Revisão	Chair
Proposta de revisão	Prazo	Revisão	Chair
Proposta de revisão	Resposta	Revisão	Chair
Andamento da revisão	X	Revisão	Chair

Após a identificação das informações existentes no sistema, dos atores aos quais devem ser destinadas e os processos que devem garantir que sejam visualizadas apropriadamente (tabelas 4.4, 4.5 e 4.6), deve-se classificar as informações conforme o grau de importância que o patrocinador atribui a cada uma. Tal classificação contribui com a verificação da destinação das informações do sistema, permitindo ao desenvolvedor atingir a transparência ideal, de acordo com a PTP.

Desta forma, para este estudo de caso, considera-se que os itens de informação contato do autor, contato do revisor, corpo do artigo e revisão do artigo são *restritos*. Assim, o modelo deverá registrar que tais itens deverão ter seu conhecimento restrito de alguma forma.

Conforme a compreensão atual do sistema, pode-se inferir que os seguintes requisitos não funcionais de transparência fazem parte da PTP:

- Transparência do processo de submissão para os membros da conferência.
- Transparência do processo de aceitação para os membros da conferência.
- Transparência do processo de revisão para os membros da conferência.

- Transparência dos artigos para os membros da conferência.
- Transparência da lista dos revisores para os membros da conferência.
- Transparência das propostas de revisão para os membros da conferência.

Portanto, adaptando os modelos SR do ECS é possível verificar as metas flexíveis de transparência e suas operacionalizações (figuras 4.5, 4.6 e 4.7).

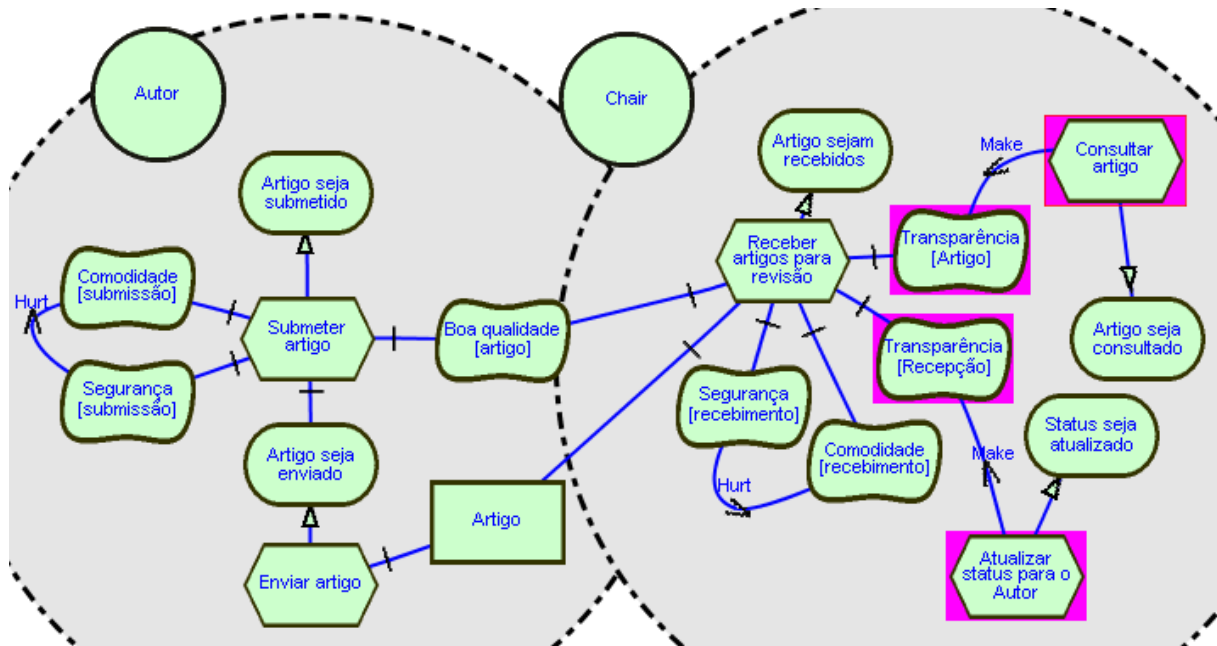


FIG. 4.5: Modelo SR para a SDsituation Submissão de Artigos - adaptada de (OLIVEIRA, 2008).

A meta flexível “Transparência[Recepção]” (figura 4.5) representa para o desenvolvedor a necessidade de que o processo de submissão seja transparente, e a meta “Status seja atualizado” representa a condição que o Chair deseja alcançar através da tarefa “Atualizar status para o Autor”, conforme ressaltado no modelo. De forma similar, a meta flexível “Transparência[Artigo]” representa a necessidade de que os artigos submetidos sejam transparentes para o Chair, atendendo seu interesse em recebê-los para que possa enviá-los para a revisão.

O modelo da figura 4.6 apresenta a meta flexível “Transparência[artigos]” que representa para o desenvolvedor a necessidade de que os artigos sejam transparentes para o Chair, atendendo seu interesse em conhecê-los para que possa preparar as propostas de revisão que serão enviadas aos revisores. As metas flexíveis “Transparência[autores]” e

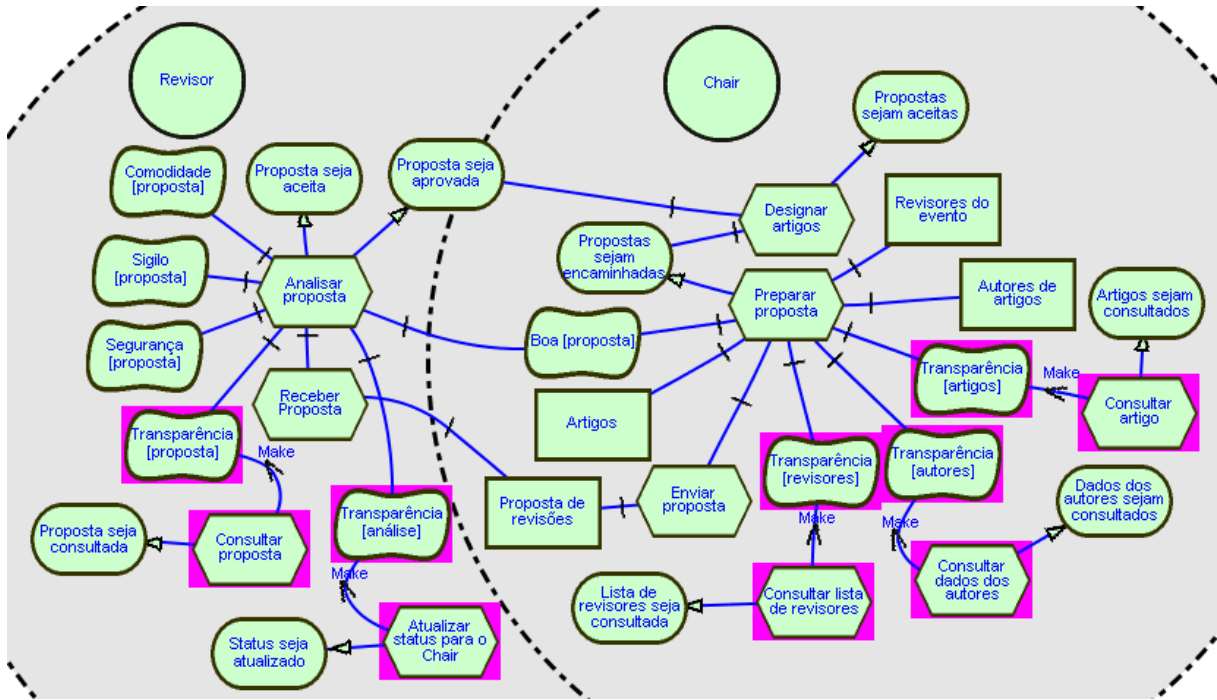


FIG. 4.6: Modelo SR para a SDsituation Aceitação de Propostas - adaptada de (OLIVEIRA, 2008).

“Transparência[revisores]” possuem o mesmo propósito. Como as informações sobre os autores estão incluídas nos artigos, a meta flexível “Transparência[autores]” poderia ser suprimida. A meta flexível “Transparência[proposta]” representa para o desenvolvedor a necessidade de que as propostas sejam transparentes para os revisores, atendendo seus interesses em conhecê-las para que possam analisá-las e respondê-las. Tais metas flexíveis são operacionalizadas por tarefas de consulta as fontes das informações (a lista de revisores, os artigos submetidos e as propostas elaboradas), conforme citado na tabela 4.5. A meta flexível “Transparência[análise]” representa para o desenvolvedor a necessidade de que o processo de aceitação seja transparente e é operacionalizada pela tarefa “Atualizar status para o Chair”.

Similar ao modelo anterior, o modelo da figura 4.7 apresenta metas flexíveis sobre a transparência dos artigos. A diferença sutil a ser considerada pelo desenvolvedor é que os artigos de interesse para o Chair são aqueles já revisados para que possa compilar as revisões, e no caso dos Revisores, os artigos de interesse são aqueles enviados para revisão para que possa realizar a revisão. Quanto a meta flexível “Transparência[Propostas]”, o interesse do Chair é referente as propostas respondidas para que possa enviar os artigos para revisão. A meta flexível “Transparência[revisão]” representa para o desenvolvedor a

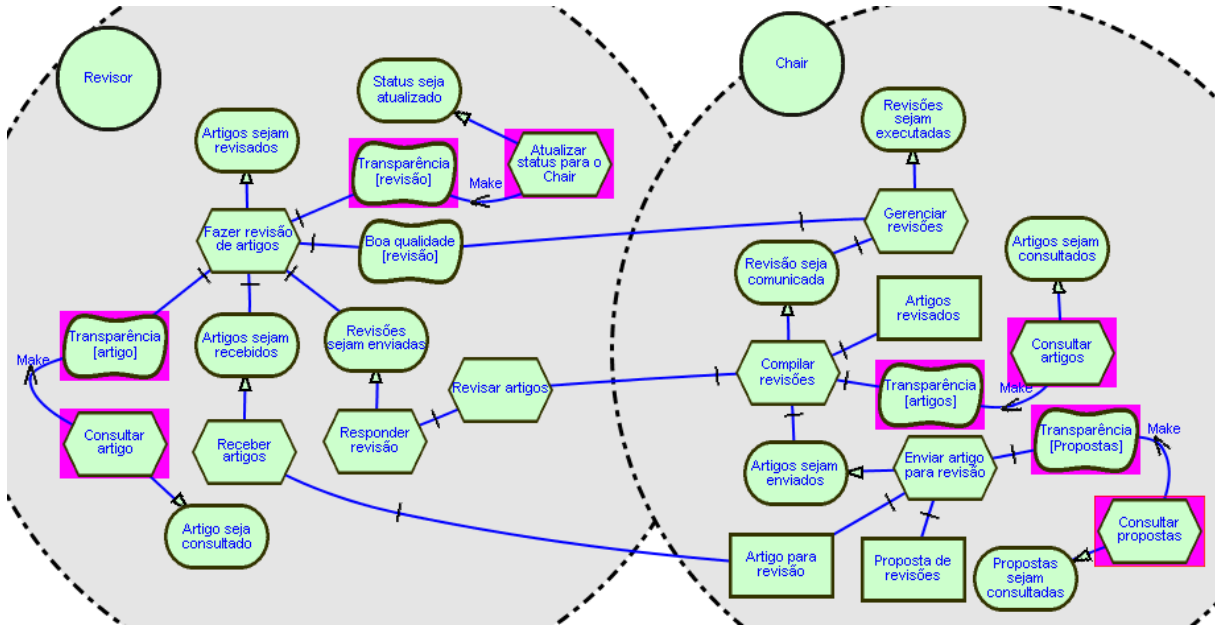


FIG. 4.7: Modelo SR para a SDsituation Revisão de Artigos - adaptada de (OLIVEIRA, 2008).

necessidade de que o processo de revisão seja transparente. Ambas as metas flexíveis são operacionalizadas, seja pela consulta as fontes ou pela atualização do status.

A utilização do NFR Framework na modelagem de um sistema, proporciona a construção de um SIG pelo desenvolvedor, permitindo o registro de seu raciocínio, a rastreabilidade das decisões tomadas e o apoio de um grafo para o refinamento do modelo. Porém, diferentemente dos modelos SR (figuras 4.5, 4.6 e 4.7), esta modelagem permite refinar as metas flexíveis de transparência em níveis mais específicos do que os apresentados, além de permitir a modelagem de limitações ou obrigações definidas na PTP.

Assim sendo, modelando o ECS através do NFR Framework, pode-se obter os grafos a seguir, que mostram o refinamento das metas flexíveis de transparência, de acordo com os requisitos inferidos para o sistema.

A modelagem da transparência dos processos para os membros da conferência pode ser exemplificada pelo modelo da figura 4.8. Este modelo apresenta a meta flexível “pai” do tipo **TP** e sua decomposição em duas metas flexíveis “filhas”. Este número de “filhas” é suficiente, conforme o modelo SR específico (figuras 4.3 e 4.7). Tais metas são interligadas por interdependências de decomposição que exercem uma contribuição do tipo **AND** para a suficiente satisfação da meta flexível “pai”. A decomposição presente no modelo utiliza o método de decomposição por subclasse, pois Chair e Revisor são subclasses de Membros

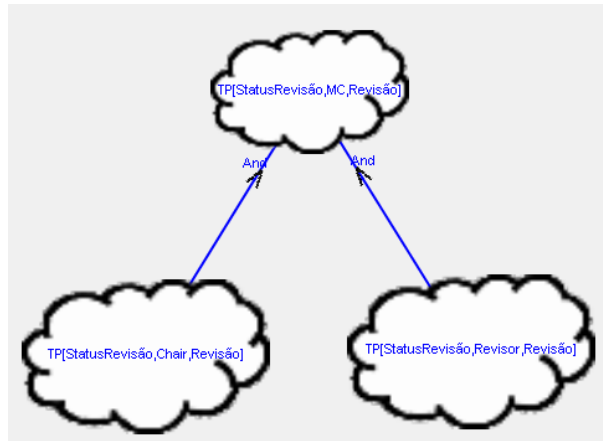


FIG. 4.8: Refinamento da transparência do processo.

da Conferência. Esta figura deve ser lida da seguinte forma:

TP[StatusRevisão,Chair,Revisão]
AND
TP[StatusRevisão,Revisor,Revisão]
SATISFICE
TP[StatusRevisão,MC,Revisão]

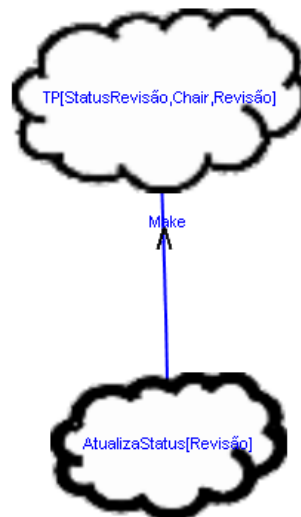


FIG. 4.9: Operacionalização da transparência do processo.

A operacionalização das metas flexíveis “filhas” pode ser demonstrada pelo modelo na figura 4.9, onde verifica-se a interdependência de operacionalização ligando a meta

flexível $TP[StatusRevisão, Chair, Revisão]$ a sua meta flexível de operacionalização $AtualizaStatus[Revisão]$. Esta interdependência exerce uma contribuição do tipo **MAKES** devido a utilização do método de operacionalização **Atualização de Status**. A leitura desta figura é:

AtualizaStatus[Revisão]
MAKES
TP[StatusRevisão, Chair, Revisão]

A modelagem da transparência dos artigos para os membros da conferência pode ser exemplificada pelo modelo da figura 4.10. Este modelo é voltado para o processo Revisão de Artigos, então, como somente dois tipos de atores do sistema interagem com o mesmo (figuras 4.3 e 4.7), o desenvolvimento do modelo pode considerar a possibilidade de existirem somente duas metas flexíveis “filhas” na decomposição da meta flexível “pai”.

Diferentemente do exemplo anterior, a meta flexível “pai” é do tipo **TD**. A existência dos tipos **TP** e **TD** na modelagem de um mesmo sistema, ratifica a necessidade de tratar separadamente as informações originadas nos dados gerenciados, das informações originadas no andamento dos processos, conforme descrito no capítulo 3.

A figura 4.10 mostra o refinamento inicial da meta flexível $TD[Artigo, MC, Revisão]$, através da decomposição do seu **tópico**, utilizando o método de decomposição por *sub-classe*, pois o Chair e o Revisor são subclasses de Membros da Conferência.

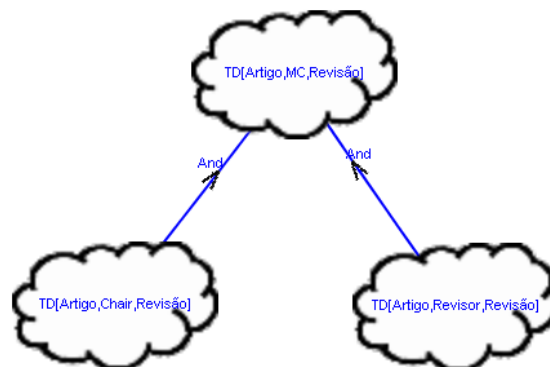


FIG. 4.10: Refinamento da transparência dos artigos.

A interligação destas metas flexíveis também são feitas por interdependências de decomposição (figura 4.10), onde a contribuição para a suficiente satisfação da meta flexível “pai” é do tipo **AND**. A leitura desta figura é:

TD[Artigo,Chair,Revisão]

AND

TD[Artigo,Revisor,Revisão]

SATISFICE

TD[Artigo,MC,Revisão]

O modelo apresentado (figura 4.10) pode ser refinado considerando as diferenças entre os artigos enviados para a revisão e os artigos revisados. Assim, a meta flexível *TD[Artigo,Chair,Revisão]* pode ser decomposta em duas outras, através da aplicação do método de decomposição por subclasse sobre o assunto do tópico da meta flexível “pai”, conforme mostrado na figura 4.11.

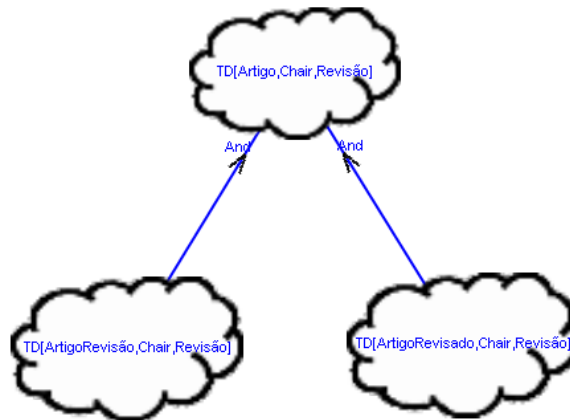


FIG. 4.11: Refinamento da transparência dos artigos (2° nível).

A figura deste modelo é lida da seguinte forma:

TD[ArtigoRevisão,Chair,Revisão]

AND

TD[ArtigoRevisado,Chair,Revisão]

SATISFICE

TD[Artigo,Chair,Revisão]

Segundo a PTP o artigo para revisão não deve ser transparente para o Chair, portanto esta parte do modelo não precisa ser desenvolvida. Seguindo com o refinamento, pode-se considerar que existem dados ostensivos e restritos na composição da informação *artigo revisado*, levando a decomposição do tipo da meta flexível *TD[ArtigoRevisado,Chair,Revisão]*, através da aplicação do método de decomposição por tipo (figura 4.12).

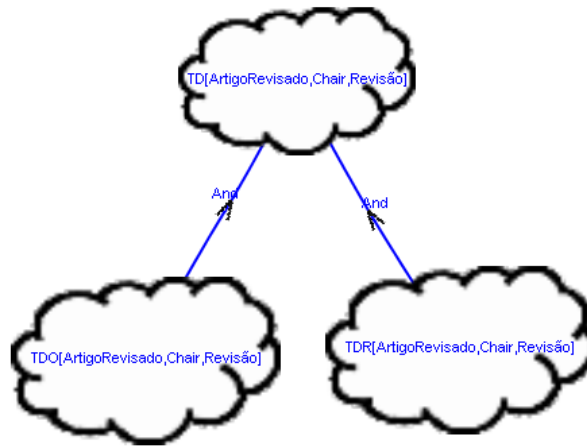


FIG. 4.12: Refinamento da transparência dos artigos (3° nível).

A decomposição apresentada no modelo da figura 4.12 mostra os subtipos **TDO** e **TDR**, conforme definido no capítulo 3. A leitura desta figura é:

$$\begin{aligned}
 & \mathbf{TDO[ArtigoRevisado,Chair,Revisão]} \\
 & \quad \mathbf{AND} \\
 & \mathbf{TDR[ArtigoRevisado,Chair,Revisão]} \\
 & \quad \mathbf{SATISFICE} \\
 & \mathbf{TD[ArtigoRevisado,Chair,Revisão]}
 \end{aligned}$$

Seguindo com o refinamento do modelo, a meta flexível $TDR[ArtigoRevisado, Chair, Revisão]$ pode ser decomposta considerando os itens de informação identificados na tabela 4.6 para a informação *artigo revisado*, através da aplicação do método de decomposição por atributos individuais (figura 4.13).

Este modelo (figura 4.13) deve ser lido da seguinte forma:

$$\begin{aligned}
 & \mathbf{TDR[ArtigoRevisado.Contato,Chair,Revisão]} \\
 & \quad \mathbf{AND} \\
 & \mathbf{TDR[ArtigoRevisado.Corpo,Chair,Revisão]} \\
 & \quad \mathbf{AND} \\
 & \mathbf{TDR[ArtigoRevisado.Revisão,Chair,Revisão]} \\
 & \quad \mathbf{SATISFICE} \\
 & \mathbf{TDR[ArtigoRevisado,Chair,Revisão]}
 \end{aligned}$$

A partir do modelo da figura 4.10, também é possível refinar a meta flexível $TD[Artigo, Revisor, Revisão]$ (figura 4.14). Este refinamento pode ser iniciado considerando-se a

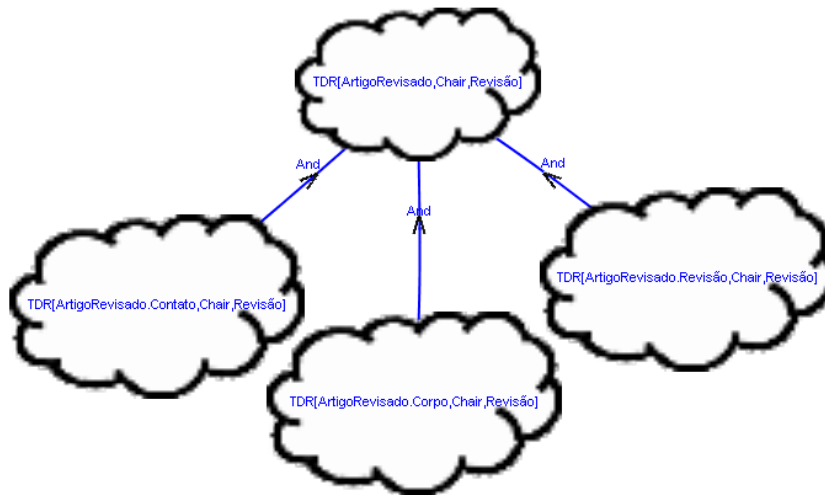


FIG. 4.13: Refinamento da transparência dos artigos (4° nível).

possível existência de itens de informação *ostensivos* e *restritos*. Então, a figura apresenta tal refinamento, atingido através da aplicação do método de decomposição por tipo sobre a meta flexível “pai”.

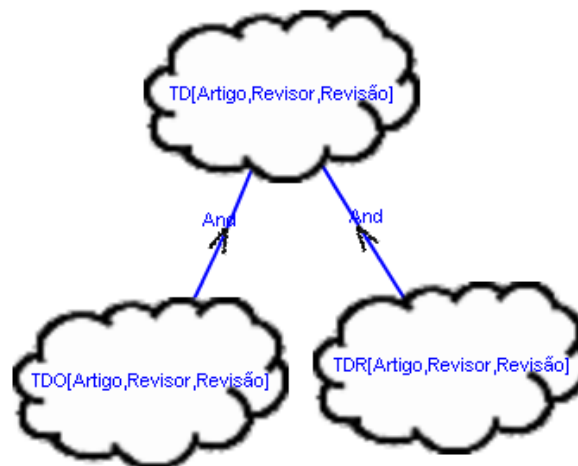


FIG. 4.14: Refinamento da transparência dos artigos/Revisor (2° nível).

Conforme os outros modelos, as interdependências entre as metas flexíveis são de decomposição, onde a contribuição para a suficiente satisfação da meta flexível “pai” é do tipo **AND**. A leitura desta figura é:

$$\begin{array}{c}
 \mathit{TDO}[\mathit{Artigo},\mathit{Revisor},\mathit{Revisão}] \\
 \mathit{AND} \\
 \mathit{TDR}[\mathit{Artigo},\mathit{Revisor},\mathit{Revisão}]
 \end{array}$$

SATISFICE

TD[Artigo,Revisor,Revisão]

O refinamento da meta flexível *TDO[Artigo,Revisor,Revisão]* define duas metas flexíveis “filhas” (figura 4.15), devido a aplicação do método de decomposição por subclasse sobre o assunto de seu tópico. A leitura do modelo desta figura é:

TDO[ArtigoRevisão,Revisor,Revisão]

AND

TDO[ArtigoRevisado,Revisor,Revisão]

SATISFICE

TDO[Artigo,Revisor,Revisão]

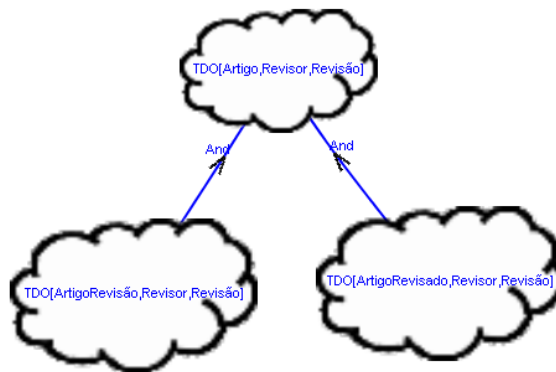


FIG. 4.15: Refinamento da transparência dos artigos/Revisor (3° nível).

Observando novamente a PTP, verifica-se que o artigo revisado não deve ser transparente para o Revisor, pois como no caso do Chair e o artigo para revisão, esta informação não possui utilidade para o Revisor, não havendo necessidade de seu conhecimento pelo mesmo, independente das características dos atores. Desta forma, a parte do modelo referente ao artigo revisado para este ator no processo de Revisão de Artigo, não precisa ser desenvolvido. Assim, o próximo refinamento será sobre a meta flexível *TDO[ArtigoRevisão,Revisor,Revisão]*, através da aplicação do método de decomposição por atributos individuais, conforme as informações da tabela 4.6. Porém, independente do Revisor ser do tipo **III** e possuir credencial de acesso às informações não ostensivas, a PTP define que os itens de informação referentes a identificação dos Autores dos artigos não devem ser transparentes aos Revisores por ocasião da revisão, para evitar constrangimentos e favorecimentos durante a execução desta tarefa. Tal regra implica que os itens

de informação nome do autor, instituição de ensino vinculada e contato do autor, não sejam transparentes ao Revisor, independente de não terem sido restringidas pelas orientações da transparência para o tipo do ator (tipo III) e de serem compatíveis com a sua credencial de acesso. Além disso, o item de informação área de interesse a qual o artigo pertence, também é desnecessária pois uma outra regra da PTP define que os Revisores somente receberão para revisão os artigos referentes as suas áreas de interesse.

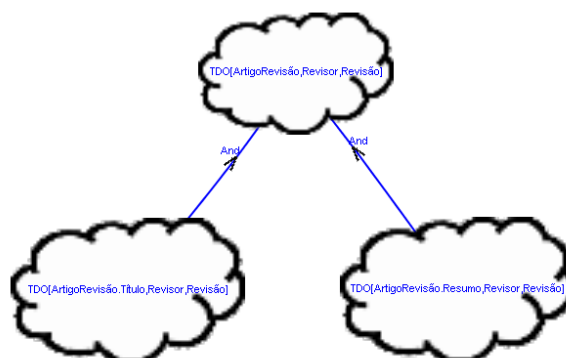


FIG. 4.16: Refinamento da transparência dos artigos/Revisor (4º nível).

A não transparência dos itens de informação citados, simplifica o modelo refinado pois diminui o número de metas flexíveis “filhas” a serem consideradas (figura 4.16). Esta simplificação retifica a tabela 4.6 visto que só será necessário tornar transparente os itens de informação título, resumo e corpo do artigo (tabela 4.7).

TAB. 4.7: Destinação das informações no processo Revisão - atualizada.

Informação	Item de informação	Processo	Destinatário
Artigo para revisão	Título	Revisão	Revisor
Artigo para revisão	Resumo	Revisão	Revisor
Artigo para revisão	Corpo	Revisão	Revisor
Artigo revisado	Nome	Revisão	Chair
Artigo revisado	Instituição	Revisão	Chair
Artigo revisado	Contato	Revisão	Chair
Artigo revisado	Título	Revisão	Chair
Artigo revisado	Resumo	Revisão	Chair
Artigo revisado	Corpo	Revisão	Chair
Artigo revisado	Área	Revisão	Chair
Artigo revisado	Revisão	Revisão	Chair
Proposta de revisão	Nome	Revisão	Chair
Proposta de revisão	Instituição	Revisão	Chair
Proposta de revisão	Contato	Revisão	Chair
Proposta de revisão	Título	Revisão	Chair
Proposta de revisão	Resumo	Revisão	Chair
Proposta de revisão	Corpo	Revisão	Chair
Proposta de revisão	Área	Revisão	Chair
Proposta de revisão	Prazo	Revisão	Chair
Proposta de revisão	Resposta	Revisão	Chair
Andamento da revisão	X	Revisão	Chair

A operacionalização da transparência dos artigos pode ser exemplificada pelo modelo da figura 4.17, onde verifica-se a interdependência de operacionalização entre a meta flexível $TDO[ArtigoRevisão.Título,Revisor,Revisão]$ e a meta flexível de operacionalização $ConsultaFonte[Título]$. Esta interdependência exerce uma contribuição do tipo **MAKES** devido a utilização do método de operacionalização **Consulta na Fonte**, similar ao ocorrido com o modelo da figura 4.9.

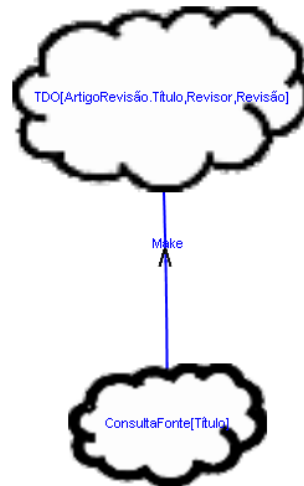


FIG. 4.17: Operacionalização da transparência dos artigos.

A PTP também define que os artigos enviados para revisão somente poderão ser transparentes aos Revisores selecionados que aceitaram realizar sua revisão, ou seja, aqueles revisores que responderam positivamente a proposta de revisão enviada sobre o artigo. Desta forma, a limitação da amplitude da transparência destes artigos é modelada conforme a figura 4.18.

4.2 SISTEMA CONTROLE DO CAIXA DO RESTAURANTE

A modelagem deste sistema por (OLIVEIRA, 2008) definiu as seguintes SDsituations:

- Liberação de Mesa
- Atendimento de Mesa
- Fechamento de Conta
- Rateio dos 10%

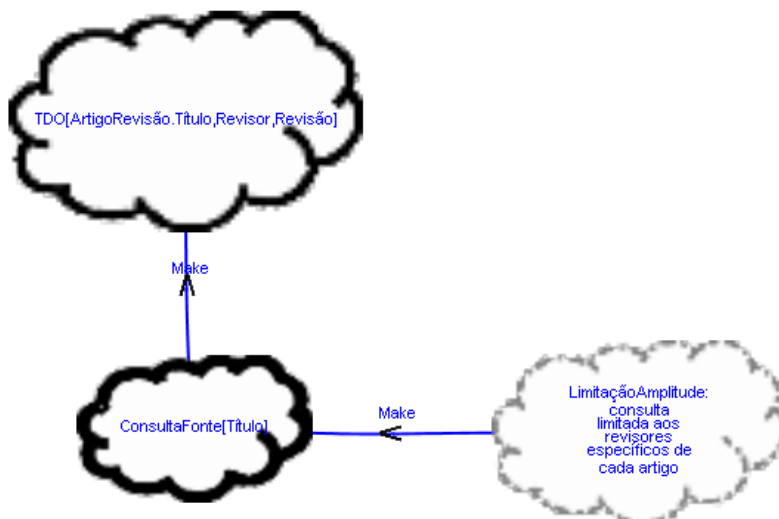


FIG. 4.18: Limitação da transparência dos artigos.

Da mesma forma que no estudo de caso anterior, os possíveis processos do sistema foram considerados conforme as SDSituations citadas. Conforme o trabalho de referencia, os seguintes atores foram definidos no sistema:

- Cliente
- Caixa
- Garçom

A análise das informações disponibilizadas mostra que o Cliente pode ser qualquer pessoa que ocupe alguma mesa do restaurante e consuma algum item do Menu de Opções. O Caixa é uma posição ocupada pelos empregados do restaurante que não são Garçons.

Para simplificar a demonstração da aplicação dos métodos de refinamento neste estudo de caso, será utilizado apenas o modelo SR referente a SDSituation Atendimento de Mesa (figura 4.19).

Classificando os atores definidos para o sistema, obtem-se a tabela 4.8.

TAB. 4.8: Classificação dos atores do sistema.

Ator	Organização	Processo	Acesso	Credencial
Cliente	Externo	Interno	Interno	Ostensivo
Caixa	Interno	Interno	Interno	Restrito
Garçom	Interno	Interno	Interno	Ostensivo

Além dos elementos explícitos no modelo da figura 4.19, pode-se identificar que o Cliente tem interesse em conhecer as informações do Menu de Opções para poder fazer o

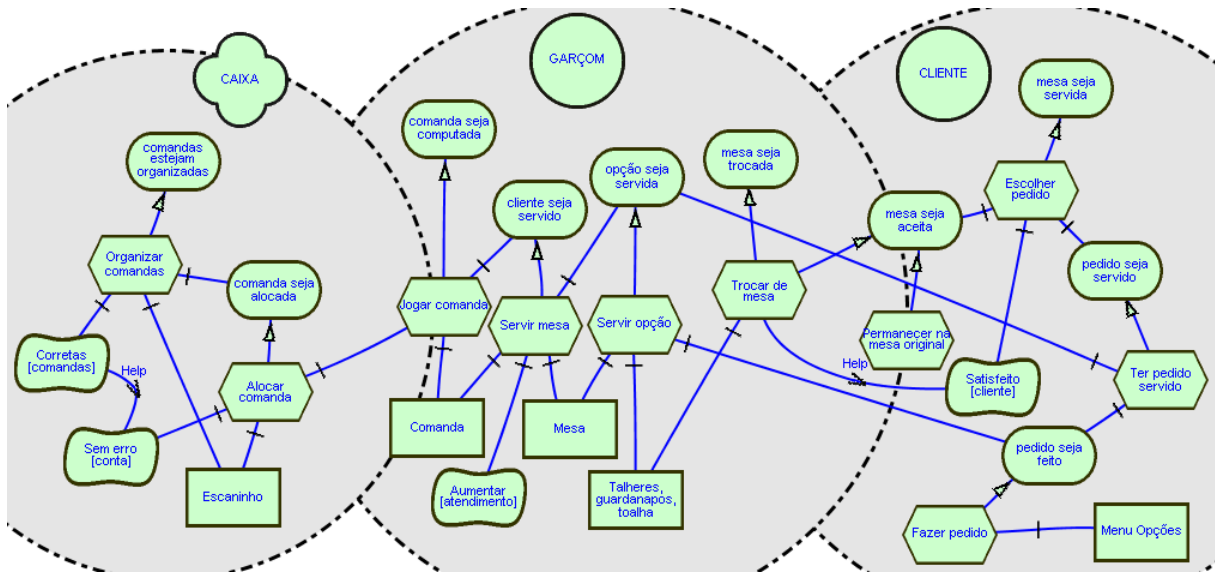


FIG. 4.19: Modelo SR para a SDSituation Atendimento de Mesa (OLIVEIRA, 2008).

pedido, o Garçon tem interesse em conhecer as informações do pedido e da mesa para que possa confeccionar as comandas e servir os Clientes, e o Caixa tem interesse em conhecer as informações das comandas para que possa organizá-las. Outro interesse do Cliente é referente ao status do processo de atendimento.

O Menu de Opções contém os identificadores dos itens de consumo disponíveis, o nome dos itens de consumo e seus valores unitários. O pedido é formado pelo nome dos itens e pelas quantidades desejadas para cada item de consumo pedido. A informação da mesa é composta da sua identificação. As comandas contém a identificação da mesa, o nome do item e a quantidade de cada item pedido. O status do processo é representado pela informação sobre o preparo na cozinha ou a busca no estoque, dos itens do pedido. Baseado nestas informações, pode-se definir a tabela 4.9.

TAB. 4.9: Destinação das informações no processo Atendimento.

Informação	Item de informação	Processo	Destinatário
Menu	Identificador item	Atendimento	Cliente
Menu	Nome	Atendimento	Cliente
Menu	PU	Atendimento	Cliente
Pedido	Nome	Atendimento	Garçon
Pedido	Quantidade	Atendimento	Garçon
Mesa	Identificador	Atendimento	Garçon
Comanda	Identificador da mesa	Atendimento	Caixa
Comanda	Nome	Atendimento	Caixa
Comanda	Quantidade	Atendimento	Caixa
Andamento do atendimento	X	Atendimento	Caixa

Baseado na tabela 4.9, deve-se classificar as informações de acordo com o seu grau de importância para o patrocinador. Neste estudo de caso, a PTP considera todos os itens de informação como ostensivos.

Conforme a compreensão atual do sistema, pode-se inferir que os seguintes requisitos não funcionais de transparência fazem parte da PTP considerada:

- Transparência do Menu de Opções para os atores
- Transparência do Pedido para os atores
- Transparência da Mesa para os atores
- Transparência das Comandas para os atores
- Transparência do processo de atendimento para os atores



FIG. 4.20: Modelo SR para a SDsituation Atendimento de Mesa - adaptada de (OLIVEIRA, 2008).

A adaptação do modelo SR original permite a inclusão das metas flexíveis de transparência e suas operacionalizações (figura 4.20).

Este modelo mostra através da meta flexível “Transparência[MenuOpção]” a necessidade de que o menu de opções seja transparente para o Cliente, atendendo seu interesse em conhecê-lo para que possa realizar o pedido. A meta flexível “Transparência[Pedido]” representa para o desenvolvedor a necessidade de que o pedido seja transparente para o Garçom, atendendo seu interesse em conhecê-lo para que possa servir o Cliente e confeccionar as comandas. Para a confecção das comandas, também é necessário a transparência

da mesa (“Transparência[Mesa]”) para que o Garçom possa identificar de quais mesas são os pedidos. A meta flexível “Transparência[Comandas]” representa para o desenvolvedor a necessidade de que a comanda seja transparente para o Caixa, atendendo seu interesse em conhecê-las para que possa organizá-las. A transparência do processo de atendimento (“Transparência[Atendimento]”) deve mostrar aos Clientes o estado em que encontram-se os pedidos realizados. A operacionalização das metas flexíveis referentes aos dados gerenciados pelo sistema é atingida através da consulta as fontes e a meta flexível referente ao andamento do processo é atingida através da atualização do status do processo.

A modelagem do sistema de Controle do Caixa do Restaurante utilizando o NFR Framework permite a visualização dos refinamentos das metas flexíveis de transparência, de acordo com os requisitos inferidos. A modelagem da transparência do processo de atendimento para os atores pode ser exemplificada pelo modelo da figura 4.21.

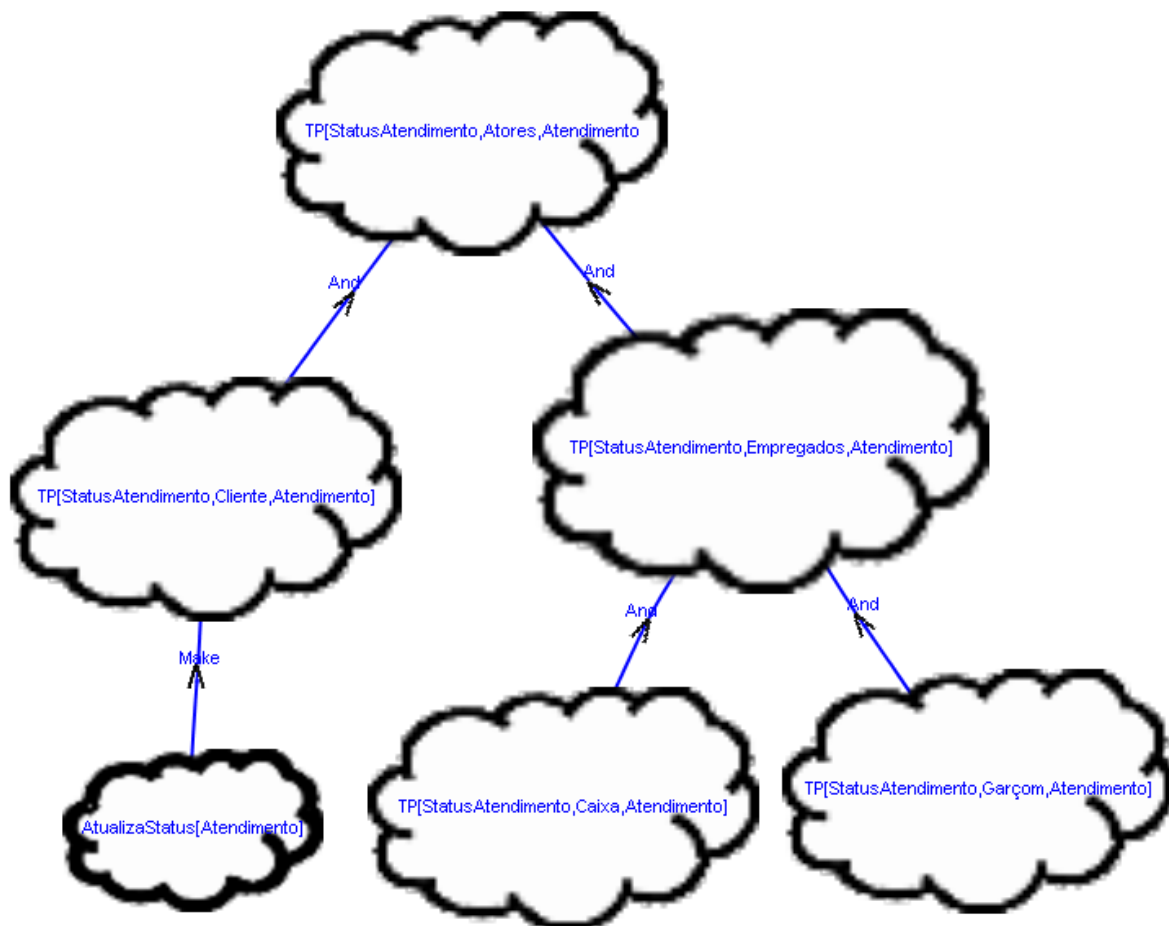


FIG. 4.21: Refinamento da transparência do processo de atendimento.

Este modelo (figura 4.21) apresenta a meta flexível “pai” do tipo **TP**. O refinamento do modelo é feito inicialmente pela aplicação do método de decomposição por subclasse. A aplicação do método de operacionalização Atualização de Status é feita somente sobre a meta flexível “TP[StatusAtendimento,Cliente,Atendimento]”, pois conforme a PTP, não é necessário a transparência deste processo para o Caixa e Garçon, independente de serem ambos do tipo III.

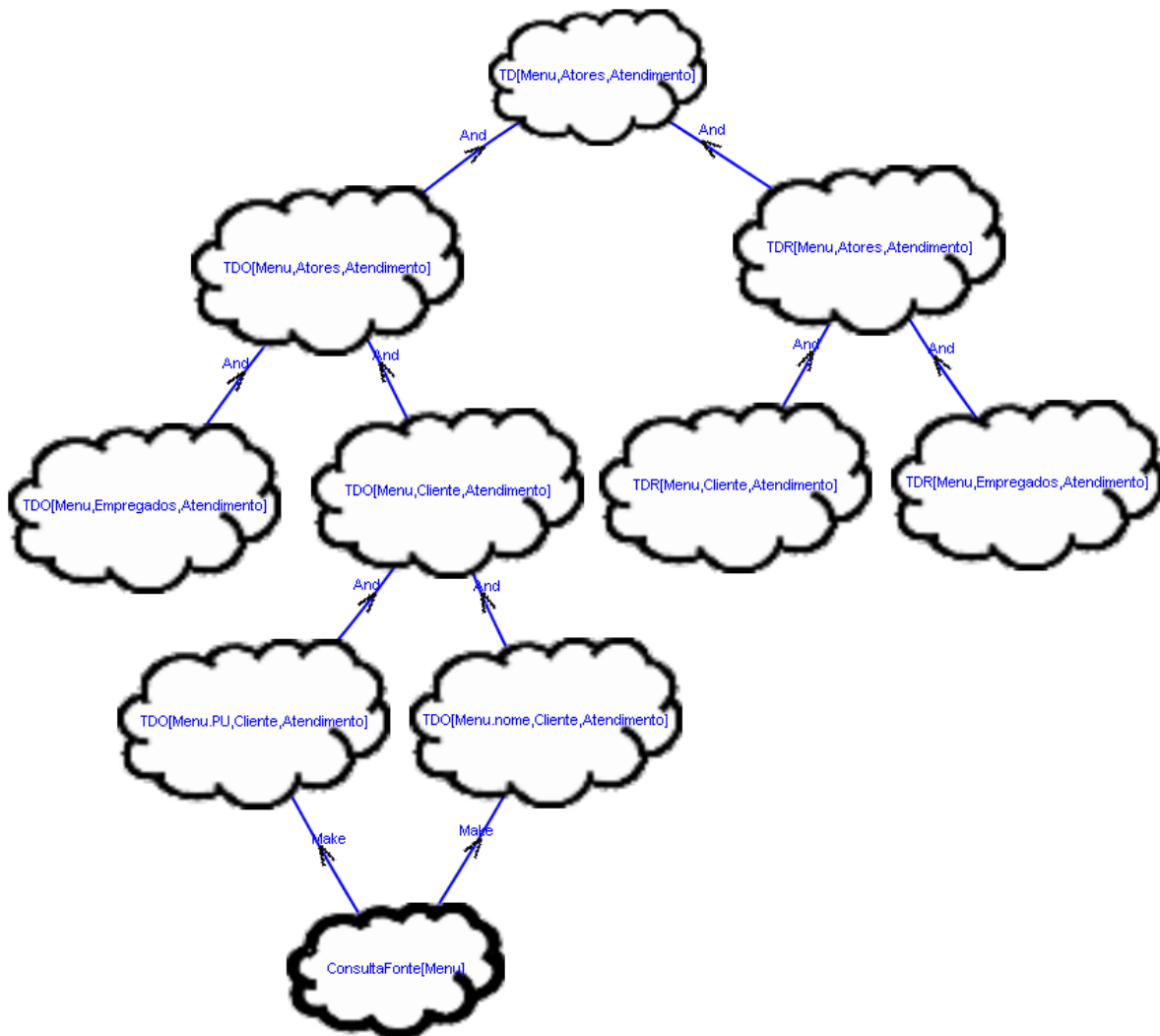


FIG. 4.22: Refinamento da transparência do Menu de Opções.

O modelo da figura 4.22 exemplifica o refinamento da meta flexível “TD[Menu, Atores, Atendimento]”. Este refinamento inicia-se com a aplicação do método de decomposição por tipo, seguido da aplicação do método de decomposição por subclasse. Segundo a PTP não deve haver transparência do Menu de Opções para os empregados através do processo de atendimento de mesa. Assim, as partes do modelo referentes aos empregados

não precisam ser desenvolvidos. Continuando o refinamento das partes do modelo referentes ao Cliente, deve ser aplicado o método de decomposição por atributos individuais, que segundo a tabela 4.9 deveria definir três metas flexíveis “filhas” para a meta flexível “TDO[Menu, Cliente, Atendimento]”. Porém, como a PTP define que os identificadores dos itens disponíveis para consumo não devem ser transparentes para os atores externos ao restaurante, assim os Clientes do restaurante (*tipo EII*) não deverão visualizar os identificadores. Tal consideração implica na retificação da destinação das informações inicial (tabela 4.10).

TAB. 4.10: Destinação das informações no processo Atendimento - atualizada.

Informação	Item de informação	Processo	Destinatário
Menu	Nome	Atendimento	Cliente
Menu	PU	Atendimento	Cliente
Pedido	Nome	Atendimento	Garçom
Pedido	Quantidade	Atendimento	Garçom
Mesa	Identificador	Atendimento	Garçom
Comanda	Identificador da mesa	Atendimento	Caixa
Comanda	Nome	Atendimento	Caixa
Comanda	Quantidade	Atendimento	Caixa
Andamento do atendimento	X	Atendimento	Caixa

4.3 CONCLUSÕES SOBRE OS ESTUDOS DE CASOS

A realização dos estudos de casos contribuíram positivamente para este trabalho, pois mostraram a aplicação dos métodos de refinamento e o raciocínio sobre os fatos relevantes a transparência, necessários para orientar e justificar os refinamentos, desde as metas flexíveis de alto nível, até sua operacionalização. Outra contribuição é a visualização do baseline voltado para permitir a rastreabilidade dos requisitos de transparência, representado pelas tabelas apresentadas em cada estudo de caso. Além disso, permitiram a utilização de modelos intencionais prontos, referentes aos casos escolhidos, como base de comparação para a modelagem dos sistemas através da utilização do NFR Framework.

Os modelos apresentados permitem confirmar a existência das duas origens para as informações de um sistema, conforme proposto no capítulo 2.

Apesar da necessidade de transparência de algumas informações estarem explícitas em alguns modelos, é importante definir como tais informações deverão ser compostas. Para isto são utilizadas as tabelas de destinação.

A representação das metas flexíveis de transparência segundo a notação proposta (capítulo 3) mostrou-se adequada pois identifica exatamente os elementos que devem ser con-

siderados pelo desenvolvedor, ou seja, os elementos da figura 3.1.

A aplicação dos métodos de decomposição não possui uma regra pré-definida, podendo ser realizada na ordem que melhor convier ao engenheiro de software.

A operacionalização das metas flexíveis permite visualizar possíveis correlações entre a Transparência e os RNF Informatividade e Acessabilidade, similar ao apresentado no segundo nível do arranjo proposto no SIG desenvolvido por (CAPPELLI et al., 2007).

Apesar do resultado positivo, verifica-se a necessidade da realização de outros estudos com o objetivo de continuar a exploração das possibilidades oferecidas pelos métodos e conceitos propostos, aprimorando-os.

5 CONCLUSÃO

Este trabalho teve como objetivo a definição dos conceitos necessários a compreensão da transparência das informações e a definição da forma para aplicá-los em favor do desenvolvimento do software transparente. Para isto, inicialmente, verificou-se que a transparência possui um sentido diferente para cada contexto em que é considerada, fato que dificulta a sua compreensão.

Definiu-se que para haver transparência de uma informação é necessário que a mesma seja visualizada por algum ator do sistema. Porém, devido à possibilidade de existirem informações vitais ao patrocinador, verificou-se a necessidade da construção de uma Política de Transparência que permita a correta destinação das informações, e conseqüentemente a definição de seus requisitos.

Para poder destinar corretamente uma informação, é preciso entender suas características. Desta forma, buscou-se a compreensão da entidade informação. Tal busca levou a identificação da necessidade dos conceitos de significado, valia e granularidade durante a destinação das informações.

A descrição da transparência como um processo de comunicação permitiu a definição de dois tipos de informações existente no software: as informações existentes nos dados gerenciados pelo sistema e as informações existente sobre os processos do sistema.

Baseado na definição de Software Transparente adotada por este trabalho, definiu-se três elementos fundamentais (**Ator - Informação - Processo**) para o tratamento da transparência em sistemas de informação. Estes elementos devem orientar a análise dos fatores de entrada e saída definidos, contribuindo com o desenvolvimento do software. Tais fatores foram divididos em oito categorias, onde destaca-se a importância da identificação dos atores e das informações, para que as decisões quanto a destinação das informações sejam tomadas.

Por fim, baseado nas informações anteriores, foram identificados alguns métodos de refinamento para as metas flexíveis de transparência, além da definição de princípios para incluir transparência das informações em sistemas.

Portanto, este trabalho tira a idéia de transparência em sistemas do nível da abstração teórico para um nível mais prático. Isto se dá, devido a definição do catálogo específico

para transparência, que além de catalogar o RNF e possibilitar a documentação das decisões tomadas durante o desenvolvimento, independe da forma como o sistema será implementado e pode ser usado por métodos de elicitação de requisitos diversos.

5.1 CONTRIBUIÇÕES

Produzir o conhecimento necessário à identificação dos requisitos não funcionais de transparência em um sistema.

A principal contribuição deste trabalho foi a produção de catálogos de métodos sobre o RNF Transparência. A utilização destes, nas fases iniciais do projeto, facilita ao engenheiro de software tomar as decisões relevantes ao projeto, que visem garantir o encontro dos requisitos não funcionais de transparência. Este conhecimento é composto por:

- Definição das origens das informações manipuladas em um sistema.
Esta definição divide o estudo em dois focos, facilitando o entendimento das características e propósitos de cada um.
- Definição da taxonomia para classificação dos atores.
Permite identificar características dos atores que podem ser utilizadas na destinação das informações do sistema.
- Definição da taxonomia para classificação das informações.
Permite identificar as informações que necessitam de restrições ao seu conhecimento em relação aos atores do sistema, contribuindo com a destinação das informações.
- Definição dos elementos fundamentais.
Define a necessidade da identificação das relações entre **Ator - Informação - Processo**, contribuindo com a análise dos fatores de entrada e saída e, conseqüentemente, com a destinação das informações.
- Definição de fatores de entrada e saída.
A análise destes fatores contribuem com a destinação das informações, fato que influencia diretamente na modelagem da transparência do sistema.

Apresentar este conhecimento na forma de um catálogo, similar ao modelo apresentado em (CHUNG et al., 2000).

Este objetivo foi alcançado com a apresentação dos conceitos descritos no capítulo 3, que foi organizado segundo o modelo citado. O uso de catálogos, facilita a consulta do conhecimento e flexibiliza a execução de atualizações. Além disso, complementa o trabalho iniciado por (CHUNG et al., 2000).

Apresentar provas de conceito que exemplifiquem a modelagem dos requisitos de transparência, desde a sua definição até sua operacionalização.

A aplicação dos conceitos e métodos propostos é demonstrada através de estudos de caso baseados em modelos de sistemas amplamente utilizados na literatura (capítulo 4).

5.2 TRABALHOS FUTUROS

Foram identificados os seguintes trabalhos futuros:

- a) Realização de outros estudos de casos para identificar outros métodos de refinamento, bem como melhorar a definição dos métodos propostos:
Permitirá a atualização do catálogo, aumentando e melhorando a qualidade do conhecimento disponibilizado.
- b) Realização de estudos para compreender as correlações entre a transparência e os outros RNF:
Permitirá a melhor avaliação dos impactos entre os RNF, contribuindo com a seleção das alternativas para o projeto.
- c) Definição de uma métrica que permita quantificar a transparência de um sistemas:
Permitirá a comparação entre sistemas diferentes, possibilitando um melhor controle de qualidade.
- d) Definição de uma metodologia para analisar as necessidades e elicitar os requisitos de transparência, incluindo o preparo da Política de Transparência específica do cliente e o desenvolvimento de um processo para a verificação da destinação das informações:
Permitirá a destinação correta e justificada das informações, considerando as intenções do patrocinador, e procurando atender os interesses dos atores quando possível.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- ALRGS. **Portal de transparência da assembléia legislativa do rio grande do sul**, 2005. URL www.al.rs.gov.br/Transparencia/. Último acesso em 23/08/2008 às 03h 32min.
- BELL, T. E. e THAYER, T. A. **Software requirements: Are they really a problem?** Em *ICSE '76: Proceedings of the 2nd international conference on Software engineering*, págs. 61–68, Los Alamitos, CA, USA, 1976. IEEE Computer Society Press.
- BOEHM, B. W. e PAPACCIO, P. N. **Understanding and controlling software costs.** *IEEE Trans. Softw. Eng.*, 14(10):1462–1477, 1988. ISSN 0098-5589.
- BOEHM, B. W. *Software Engineering Economics*. Prentice-Hall PTR, November 1981.
- BORGIDA, A., GREENSPAN, S., e MYLOPOULOS, J. **Knowledge representation as the basis for requirements specifications.** *IEEE Computer*, 18(4):82–91, 1985. ISSN 0018-9162.
- BUBENKO, J. A. **Information modeling in the context of system development.** Em *IFIP Congress*, págs. 395–411, 1980.
- BUENO, F. S. *Dicionário Escolar da Língua Portuguesa*. Fename, 11º edition, 1979.
- CAPPELLI, C., OLIVEIRA, A. P. A., e DO PRADO LEITE, J. C. S. **Exploring business process transparency concepts.** IEEE International Symposium on Requirements Engineering (RE'07), 2007.
- CHRISTEL, M. G. e KANG, K. C. **Issues in requirements elicitation.** Technical report, Carnegie Mellon University, September 1992.
- CHUNG, L. **Representation and utilization of non-functional requirements for information system design.** Em ANDERSEN, R., JR., J. A. B., e SØLVBERG, A., editores, *CAiSE*, volume 498 of *Lecture Notes in Computer Science*, págs. 5–30. Springer, 1991. ISBN 3-540-54059-8.
- CHUNG, L. **Dealing with security requirements during the development of information systems.** Em ROLLAND, C., BODART, F., e CAUVET, C., editores, *CAiSE*, volume 685 of *Lecture Notes in Computer Science*, págs. 234–251. Springer, 1993a. ISBN 3-540-56777-1.
- CHUNG, L. *Representing and Using Non-Functional Requeriments: A Process-Oriented Approach*. Tese de Doutorado, Dept. of computer Science, Univ. of Toronto, June 1993b.

- CHUNG, L., NIXON, B., YU, E., e MYLOPOULOS, J. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.
- CHUNG, L., NIXON, B. A., e YU, E. **Using quality requeriments to systematically develop quality software**. Em *Fourth International Conference on Software Quality*, McLean, VA, U.S.A, October 3-5 1994.
- CONGRESS, U. **Sarbanes-oxley act**. Pub. L. 107-204, 116 Stat. 745, 15 U.S.C., July 30 2002. USA Federal Law.
- CORRUPÇÃO, C. G. C. **Transparency international**. URL www.transparency.org. Último acesso em 20/05/2008, às 15h42min.
- CSN. **Portal de transparência csn**, 2006. URL www.csn.com.br.
- CUNHA, H. S. **Uso de estratégias orientadas a metas para modelagem de requisitos de segurança**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Abril 2007.
- DARDENNE, A., VAN LAMSWEERDE, A., e FICKAS, S. **Goal-directed requirements acquisition**. *Science of Computer Programming*, 20:3–50, 1993.
- DARIMONT, R. e VAN LAMSWEERDE, A. **Formal refinements patterns for goal-driven requirements elaboration**. Em *Proc. FSE'4*, págs. 179–190. Fourth ACM SIGSOFT Symp. on the Foundations of Software Engineering, 1996.
- DO BRASIL, G. F. **Portal da transparência**. Decreto nº 5.482, 30 Junho 2005. URL www.portaltransparencia.gov.br. Portaria Interministerial nº 140, de 16 de março de 2006.
- EUROPEU, P. **Ato 2004/109/ce**, 2004. URL <http://europa.eu/scadplus/leg/pt/lvb/l22022.htm>.
- GOGUEN, J. A. e LINDE, C. **Techniques for requirements elicitation**. págs. 152–164. IEEE Requirements Engineering '93, 1993.
- GREVE, J. M. **Ética x atingimento de metas - a transparência nas informações**, 2007. Trabalho exigido para a obtenção do Título de Especialista em Administração.
- HARRISON, R. e NITHI, R. **Specifying software quality requirements via unification and quantification of differing viewpoints**.
- HOFFMAN, T. **Sarbanes-oxley sparks forensics apps interest**. *Computerworld*, March 26 2004. URL <http://www.computerworld.com/>. Último acesso em: 10/06/2008 às 18h 24min.
- HOLZNER, B. e HOLZNER, L. *Transparency in Global Change: The Vanguard of the Open Society*. University of Pittsburg Press, 2006.
- JURAN, J. M. *Juran's Quality Control Handbook*. Mcgraw-Hill, 4th edition, AGO 1988.

- KALAKOTA, R. e ROBINSON, M. *Services Blueprint: Roadmap for Execution*. Addison-Wesley Professional, 1^o edition, June 2003.
- KALINOWSKI, M., SPÍNOLA, R. O., NETO, A. C. D., BOTT, A., e TRAVASSOS, G. H. **Inspeções de requisitos de software em desenvolvimento incremental: Uma experiência prática**. Em *Anais do SBQS 2007 - VI Simpósio Brasileiro de Qualidade de Software*, Porto de Galinhas - PE, Junho 2007. SBC.
- KONDO, S. *Transparência e Responsabilização no Setor Público: Fazendo Acontecer*. Ministério Público, SEGES, 2002.
- LEITE, J. C. S. P. **Sistemas de software transparentes**, Outubro 2006a. URL <http://www-di.inf.puc-rio.br/julio/Slct-pub/transp-sbes.pdf>. Palestra convidada para o 20^o SBES.
- LEITE, J. C. S. P., WERNECK, V. M., OLIVEIRA, A. P. A., CAPPELLI, C., CERQUEIRA, A., CUNHA, H. S., e GONZALEZ-BAIXAULI, B. **Understanding the strategic actor diagram: An exercise of meta modeling**. Em *Anais do Wer07 - 10th Workshop on Requirements Engineering, Toronto, Canadá, 2007*.
- LEITE, J. C. S. P. **Transparência de software**, Jul 2006b. URL <http://amazonnggg.blogspot.com/2006/07/software-transparency.html>. Último acesso: 26/06/2008 às 02:33h.
- LEITE, J. C. S. P. e FRANCO, A. P. M. **A strategy for conceptual model acquisition**. págs. 243–246, San Diego, 1993. International Symposium on Requirements Engineering, IEEE Computer Society Press.
- LEITE, J. C. S. P., HADAD, G. D. S., DOORN, J. H., e KAPLAN, G. N. **A scenario construction process**. *Requirements Engineering*, 5(1):38–61, July 2000.
- LEITE, J. C. S. P. e OLIVEIRA, A. P. A. **A client oriented requirements baseline**. Em *RE*, págs. 108–115. IEEE Computer Society, 1995.
- LETIER, E. *Reasoning about Agents in Goal-Oriented Requirements Engineering*. Tese de Doutorado, Université Catholique de Louvan, Maio 2001.
- LIU, L. e YU, E. **Designing information systems in social context: A goal and scenario modelling approach**. *Information Systems*, 29(02), 2003.
- LIU, L., YU, E. S. K., e MYLOPOULOS, J. **Security and privacy requirements analysis within a social setting**. Em *RE*, págs. 151–161. IEEE Computer Society, 2003. ISBN 0-7695-1980-6.
- MICROSOFT. *Microsoft Computer Dictionary*. Microsoft Press, 5th edition, 2002. ISBN 0-7356-1495-4.
- MYLOPOULOS, J., CHUNG, L., e NIXON, B. **Representing and using non-functional requirements: A process-oriented approach**. *IEEE Transactions on Software Engineering*, 18(6):483–497, jun 1992.

- NIXON, B. **Implementation of information system design specifications: a performance perspective.** Em *Proceedings of the third international workshop on Database programming languages : bulk types & persistent data*, págs. 149–168, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 1-55860-242-9.
- NIXON, B. A. **Dealing with performance requirements during the development of information system.** Em *Proceedings of the IEEE International Symposium on Requirements Engineering*, págs. 42–49, San Diego, CA, January 4–6 1993. IEEE, IEEE Computer Society Press.
- NIXON, B. A. **Representing and using performance requirements during the development of information systems.** Em *EDBT '94: Proceedings of the 4th international conference on extending database technology*, págs. 187–200, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 3-540-57818-8.
- OLIVEIRA, A. P. A. **Engenharia de Requisitos Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos.** Tese de Doutorado, PUC-Rio, Março 2008.
- OLIVEIRA, A. P. A. e CYSNEIROS, L. M. **Defining strategic dependency situations in requirements elicitation.** Em ALENCAR, F., SÁNCHEZ, J., e WERNECK, V., editores, *Anais do WER06 - Workshop em Engenharia de Requisitos, Rio de Janeiro, RJ, Brasil, Julho 13-14*, págs. 12–23, 2006. ISBN 1413-9014 (ISSN).
- OLIVEIRA, A. P. A., LEITE, J. C. S. P., e CYSNEIROS, L. M. **Método eri*c - engenharia de requisitos intencional.** Em *11th Workshop on Requirements Engineering (WER 2008)*, Barcelona, Catalonia, Spain, September 12-13 2008.
- OLIVEIRA, A. P. A., CAPPELLI, C., DE SOUZA CUNHA, H., DO PRADO LEITE, J. C. S., e WERNECK, V. M. **Engenharia de requisitos intencional: Tornando o software mais transparente.** Em *XXI Simpósio Brasileiro de Engenharia de Software*. Sociedade Brasileira de Computação, Outubro 2007. URL www.sbbd-sbes2007.ufpb.br/tuto3.pdf.
- OLIVEIRA, A. P. A., CYSNEIROS, L. M., DO PRADO LEITE, J. C. S., FIGUEIREDO, E. M. L., e DE LUCENA, C. J. P. **Integrating scenarios, i*, and aspectt in the context of multi-agent systems.** Em ERDOGMUS, H., STROULIA, E., e STEWART, D. A., editores, *CASCON*, págs. 204–218. IBM, 2006.
- PETERS, D. K. e PARNAS, D. L. **Requirements-based monitors for real-time systems.** Em *ISSTA*, págs. 77–85, 2000.
- PETROBRAS. **Portal de transparência da petrobras**, 2008. URL www.petrobras.com.br. Último acesso em 23/08/2008 às 03h 30min.
- PFLEEGER, S. L. **Engenharia de Software: Teoria e Prática.** Prentice Hall, 2° edition, 2004.

- PRESSMAN, R. S. *Engenharia de Software*. McGraw-Hill Interamericana do Brasil Ltda, sexta edição edition, 2005.
- RIBEIRO, M. P. e BARRADAS, O. C. M. *Telecomunicações: Sistemas Analógico - Digitais*. Embratel, 1980.
- ROBINSON, W. N. **Implementing rule-based monitors within a framework for continuous requirements monitoring**. Em *HICSS*. IEEE Computer Society, 2005. ISBN 0-7695-2268-8.
- ROSSNER, M., EPPS, H. V., e HILL, E. **Show me the data**. *The Rockefeller University Press, The Journal of Cell Biology - JCB*, 179(6):1091 – 1092, 17 Dez 2007. URL <http://www.jcb.org/cgi/doi/10.1083/jcb.200711140>. Downloaded from jcb.rupress.org on December 1, 2008.
- SEINSTRA, F. J., KOELMA, D., e GEUSEBROEK, J. M. **A software architecture for user transparent parallel image processing**. Em *Parallel Computing*, págs. 653–662, 2002.
- SIMON, H. A. *The Sciences of the Artificial*. MIT Press, 3^o edition, 1996.
- SOMMERVILLE, I. e SAWYER, P. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, Inc., New York, NY, USA, 1997. ISBN 0471974447.
- STOUT, G. A. **Requeriments traceability and the effect on the system development lifecycle**. *Spring Cluster*, 2001.
- THEYER, R. H. e DORFMAN, M. *System and Software Requirements Engineering*. IEEE Computer Society Press, 1990.
- VAN LAMSWEERDE, A. **Formal specification: a roadmap**. *ACM Press 2000*, 2000a. URL <http://store.acm.org/acmstore>. Taken From: "The Future of Software Engineering" , Anthony Finkelstein (Ed.).
- VAN LAMSWEERDE, A. **Requirements engineering in the 00: A research perspective**. Em *Proc. ICSE'2000*, págs. 5–19. 22nd International Conference on Software Engineering, ACM Press, 2000b.
- VAN LAMSWEERDE, A. **Goal-oriented requirements engineering: A guided tour**. Em *RE*, pág. 249. IEEE Computer Society, 2001. ISBN 0-7695-1125-2.
- VAN LAMSWEERDE, A. e LETIER, E. **Handling obstacles in goal-oriented requirements engineering**. *IEEE Transactions on Software Engineering, Special Issue on Exception Handling*, 2000.
- WIKIPEDIA. URL <http://en.wikipedia.org/>. Último acesso em 23/08/2008 às 03h 25min.
- XAVIER, I. *O discurso cinematográfico: A opacidade e a transparência*, volume 4. Editora Paz e Terra, 1977. Coleção Cinema.

- YU, E. e MYLOPOULOS, J. **Why goal-oriented requirements engineering.** Em DUBOIS, E., OPDAHL, A. L., e POHL, K., editores, *4th International Workshop on Requirements Engineering: Foundations of Software Quality*, págs. 15–22, Pisa, Italy, June 1998. Presses Universitaires de Namur.
- YU, E. S.-K. **Modelling strategic relationships for process reengineering.** Tese de Doutorado, Toronto, Ont., Canada, Canada, 1995. Adviser-John Mylopoulos.
- ZADE, A. T., RASULZADEH, S., e TORKASHVAN, R. **A middleware transparent framework for applying mda to soa.** Em *Proceedings of World Academy of Science, Engineering and Technology*, volume 32, 2008.
- ZAVE, P. e JACKSON, M. **Four dark corners of requirements engineering.** *ACM Transactions on Software Engineering and Methodology*, 6(1):1–30, jan 1997.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)