

5. Argumentação sobre Requisitos de Transparência

Esse capítulo descreve o terceiro grande desafio para se obter a transparência de software: capturar os argumentos dos interessados em relação à transparência do software descrito pelos requisitos intencionais. Esses argumentos são capturados e analisados formalmente para se determinar a validade relativa dos requisitos de transparência. Uma vez relativamente validados, os requisitos de transparência podem ser desenhados e implementados com o auxílio das heurísticas de transformação apresentadas no Capítulo 3.

Em (Leite e Cappelli 2010) e (Aló 2009), os autores consideram que a transparência de software envolve vários critérios de qualidade, tais como: acessibilidade, usabilidade, informatividade, entendimento e auditabilidade. Dessa forma, os autores lidam com a transparência considerando a mesma e os critérios de qualidade associados como requisitos não-funcionais (RNFs) ou metas flexíveis. Eles também provêem uma visão gráfica baseada na notação de *Softgoals Interdependency Graph* (SIG) do NFR Framework (Chung et al. 2000). Esse grafo contém algumas metas flexíveis relacionadas à transparência e suas contribuições e impactos na transparência de software.

Esforços têm sido realizados visando o desenvolvimento de um Catálogo de Transparência de Software (CTS) (CTS 2011) baseado em (Leite e Cappelli 2010) e (Aló 2009). Esse catálogo facilita a operacionalização da rede de metas flexíveis de transparência de software e, conseqüentemente, a análise e a avaliação do software. O principal propósito de se ter um CTS é melhorar a reutilização dos conceitos relacionados à transparência. A idéia que perseguimos (Serrano e Leite 2011a) é a reutilização de requisitos não-funcionais (critérios de qualidade) (Leite et al. 2005) através da instanciação e aplicação do catálogo em diferentes projetos de software. Entretanto, é difícil aplicar esse catálogo no sentido de esclarecer como alguém saberá se a transparência é atingida como resultado da aplicação do catálogo. Dessa forma, a questão consiste em como abordar a validação de um modelo enriquecido com o conceito de transparência.

Como previamente mencionado, a transparência é afetada por uma rede complexa de metas flexíveis interdependentes, a qual é composta por um número considerável de metas flexíveis correlacionadas. Como esperado, a validação de um modelo que incorpora o conceito de transparência e seus correlatos não é simples tão pouco já definida na comunidade científica, como ocorre com outros critérios de qualidade, tais como segurança e desempenho de um software.

Focando nossa atenção na dimensão do conceito de transparência, caso optemos por lidar com cada meta flexível usando um método específico, essa estratégia demandaria um esforço e uma dedicação considerável na atividade de validação. Além disso, e contra a nossa intenção, vários problemas provavelmente ocorreriam ao longo dessa atividade como, por exemplo, os efeitos colaterais causados pela interdependência das metas flexíveis. Adicionalmente, a validação da transparência precisa considerar diferentes pontos de vista (Leite e Freeman 1991). A transparência é avaliada com base no ponto de vista de cada interessado. Na concepção dos interessados, transparência comumente pode implicar em liberdade de informação, abertura e confiabilidade. Todavia, transparência é uma questão mais complexa e pouco explorada, a qual envolve – dentre outros desafios – portabilidade, uniformidade, intuitividade, atualidade, composabilidade e rastreabilidade.

Nesse cenário, como é possível dizer que um software é transparente? Visando responder essa questão, precisamos validar os requisitos de transparência de software considerando várias questões relacionadas e os diferentes pontos de vista dos interessados (Leite e Freeman 1991). Considerando os diferentes pontos de vista dos interessados, em (Jureta et al. 2009), os autores diferenciam validade absoluta e validade relativa. A validade absoluta almeja “obter artefatos da ER que capturam exatamente o que os interessados realmente precisam”; enquanto a validade relativa “está preocupada em se os interessados concordam com o conteúdo de um artefato da ER”. Adicionalmente, a validade relativa “é atingível na prática.”

Nesse capítulo, propomos aplicar a teoria de argumentação (Chesñevar et al. 2006) para validar os requisitos intencionais modelados com o *framework* i* (Yu 1995; Yu 1997; ISTARWIKI 2011) usando o Catálogo de Transparência de Software (CTS) (CTS 2011), que contém o SIG de Transparência (Leite e Cappelli 2010; Aló 2009) e suas operacionalizações. A teoria de argumentação é

baseada no *framework* ACE (Jureta et al. 2009), o qual lida com diferentes pontos de vista. Esse *framework* auxilia na realização de um acordo entre múltiplas partes, com base em argumentação. Visando atingir esse propósito: 1) enxergamos os modelos i^* e SIGs do NFR Framework contidos no CTS como proposições no *framework* ACE, 2) definimos como obter a condição de aceitabilidade (i.e., a condição de parada), 3) rastreamos os argumentos dos interessados com a linguagem ACE, e 4) analisamos se existe um acordo entre múltiplas partes quanto aos requisitos de transparência usando os algoritmos do *framework* ACE e a condição de aceitabilidade.

O processo de argumentação proposto nesse capítulo foi desenvolvido através de dois cenários de uso, descritos nas Seções 5.3 e 5.4. Esse processo foi, posteriormente, validado em um estudo de caso, a aplicação *Web Lattes-Scholar*. Mais detalhes no Capítulo 7.

Esse capítulo está organizado em seções. A Seção 5.1 descreve como representar modelos intencionais em grafos na linguagem proposta pelo *framework* ACE. A Seção 5.2 apresenta o conceito de Condição de Aceitabilidade (*Acceptability Condition* - AC) e como determiná-la para um grafo ACE traduzido a partir de modelos intencionais. A Seção 5.3 mostra como capturar argumentos dos interessados. A validação relativa de requisitos de transparência é explicada na Seção 5.4. A Seção 5.5 mostra o apoio ferramental para a construção de grafos ACE. Na Seção 5.6, são discutidas as contribuições científicas e alguns trabalhos relacionados. Finalmente, as considerações finais são apresentadas na Seção 5.7.

5.1. Representando Modelos Intencionais no *Framework* ACE

Adicionar critérios de qualidade relacionados à transparência para os artefatos de software, depois dos mesmos serem construídos, é uma tarefa difícil. A transparência de software precisa ser validada de antemão, nos documentos de requisitos, trazendo as questões de transparência para o começo da construção do software. De acordo com (Jureta et al. 2009), a linguagem ACE pode lidar com qualquer artefato de engenharia de requisitos (ER) caso o artefato seja expresso usando proposições. Dessa forma, propomos o uso da linguagem ACE para apoiar a validação de requisitos no que diz respeito à transparência.

Visando documentar/armazenar os argumentos dos interessados a favor ou contra os requisitos de transparência, precisamos, primeiramente, representar os requisitos modelados no *framework* i^* como proposições na linguagem ACE. Em um segundo momento, precisamos ainda representar os SIGs do Catálogo de Transparência de Software como proposições na linguagem ACE, no intuito de se ter um *framework* comum que permita ao engenheiro de requisitos identificar como os requisitos se conflitam com as questões de transparência.

Para representar modelos i^* como proposições na linguagem ACE, identificamos estruturas recorrentes do *framework* i^* e as expressamos como proposições. A **sutil** inversão das direções de setas em algumas regras de tradução é proposital e fundamental para a correta análise da condição de aceitabilidade do grafo.

As estruturas recorrentes do *framework* i^* são (i) estrutura meios-fim, identificadas por uma meta e uma ou mais tarefas associadas por um conjunto de elos meios-fim; (ii) estrutura de decomposição de tarefa, identificadas por uma tarefa decomposta em metas, metas flexíveis, sub-tarefas e recursos. A decomposição usa elos de decomposição de tarefa e tem a semântica de uma decomposição do tipo E (*AND-decomposition*); (iii) estrutura de contribuição, identificada por uma meta flexível sendo impactada por tarefas, metas e outras metas flexíveis através de elos de contribuição; e (iv) estrutura de dependência, identificada por dois atores (em diagramas SD) ou tarefas (em diagramas SR) conectadas pela mesma meta, meta flexível, tarefa ou recurso usando elos de dependências. Os diagramas SD e SR foram previamente discutidos no Capítulo 2 – Seção 2.1.

A Figura 5.1 mostra essas quatro estruturas i^* recorrentes e suas respectivas representações como proposições na linguagem ACE. A estrutura i^* meios-fim (Figura 5.1–A) consiste em uma ou mais tarefas associadas por elos meios-fim a uma meta. A meta $g1$ e as tarefas $t1$ e $t2$ são, respectivamente, expressas na linguagem ACE como vértices de informação $i(g1)$, $i(t1)$ e $i(t2)$. Todos os elos meios-fim são semanticamente idênticos, e uma regra de inferência I_{ME} é suficiente para expressar essa relação. A meta $g1$ é a entrada para a regra de inferência I_{ME1} e as tarefas $t1$ e $t2$ são saídas. Dessa forma, a regra de inferência I_{ME1} pode ser formalmente representada como $I_{ME1}(i(g1), \{i(t1), i(t2)\})$.

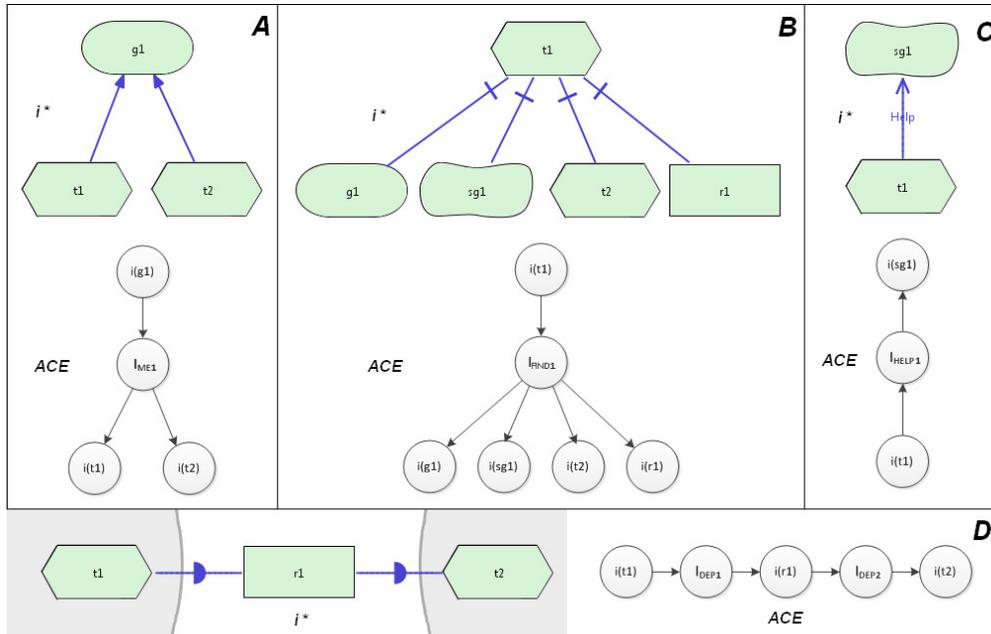


Figura 5.1 - As quatro estruturas i^* recorrentes representadas como proposições na linguagem ACE

A estrutura i^* de decomposição de tarefa (Figura 5.1–B) consiste em uma tarefa sendo decomposta em metas, metas flexíveis, subtarefas e recursos. Novamente, metas ($g1$), metas flexíveis ($sg1$), tarefas ($t1$ e $t2$) e recursos ($r1$) são, respectivamente, expressas na linguagem ACE como vértices de informações $i(g1)$, $i(sg1)$, $i(t1)$, $i(t2)$ e $i(r1)$. Todos os elos de decomposição de tarefas são semanticamente idênticos para uma decomposição do tipo E (*AND-decomposition*), expressos pela regra de inferência I_{AND} . A tarefa decomposta $t1$ é a entrada para a regra de inferência I_{AND1} e as decomposições $g1$, $sg1$, $t2$ e $r1$ são as saídas. Dessa forma, a regra de inferência I_{AND1} pode ser formalmente representada como $I_{AND1}(i(t1), \{i(g1), i(sg1), i(t2), i(r1)\})$.

A estrutura i^* de contribuição (Figura 5.1–C) consiste em uma meta flexível sendo impactada por tarefas, metas ou outras metas flexíveis. O exemplo na Figura 5.1 mostra uma meta flexível $sg1$ sendo impactada positivamente (elo de contribuição do tipo *Help*) pela tarefa $t1$. A meta flexível $sg1$ e a tarefa $t1$ são expressas, respectivamente, na linguagem ACE como vértices de informações $i(sg1)$, e $i(t1)$. A contribuição do tipo *help* é expressa pela regra de inferência I_{HELP} . A tarefa $t1$ é a entrada para a regra de inferência I_{HELP1} e a meta flexível $sg1$ é a saída. Dessa forma, a regra de inferência I_{HELP1} pode ser formalmente expressa como $I_{HELP1}(i(t1), i(sg1))$.

Finalmente, a estrutura i^* de dependência (Figura 5.1–D) consiste em um *dependor*, a tarefa $t1$, um *dependee*, a tarefa $t2$, e um *dependum*, o recurso $r1$. As tarefas $t1$ e $t2$ e o recurso $r1$ são, respectivamente, expressas na linguagem ACE como vértices de informações $i(t1)$, $i(t2)$ e $i(r1)$. Duas regras de inferência I_{DEP} são análogas aos elos de dependências. As regras de inferência podem ser formalmente expressas como $I_{DEP1}(i(t1),i(r1))$ e $I_{DEP2}(i(r1),(t2))$.

As traduções mostradas na Figura 5.1 podem ser usadas como diferentes *templates*, ou amostras, para facilitar o processo de tradução de i^* para ACE. Uma vez que os requisitos são representados como proposições na linguagem ACE, é possível validar os requisitos de transparência. Visando atingir esse propósito, precisamos ainda representar as preocupações de transparência como proposições.

Transparência, como um RNF (Leite e Cappelli 2010; Aló 2009), é descrita usando um SIG, modelada no NFR Framework e depois expressa como um padrão de requisitos do tipo objetivo (Supakkul et al. 2010). O padrão de requisitos “Objetivos de Transparência de Software” é apresentado na Figura 5.2. Visando traduzir o SIG de transparência para a linguagem ACE, precisamos identificar estruturas recorrentes do SIG e expressá-las como proposições na linguagem ACE. No NFR Framework, SIGs reutilizam três estruturas distintas: decomposições, contribuições e *claims*.

A Figura 5.3 ilustra essas três estruturas SIG recorrentes e suas respectivas representações como proposições na linguagem ACE. A estrutura decomposição é identificada por um RNF sendo decomposto por operadores do tipo E (*AND-decomposition*) ou OU (*OR-decomposition*) em um ou mais RNFs. A decomposição consiste em um detalhamento do RNF em outros RNFs mais específicos. A Figura 5.3-A mostra um RNF $nfr1$ sendo decomposto por um operador do tipo E nos RNFs $nfr2$ e $nfr3$. Os RNFs $nfr1$, $nfr2$ e $nfr3$ são expressos como vértices de informações na linguagem ACE. Os elos de decomposição do tipo E (o mesmo se aplica aos do tipo OU) são semanticamente idênticos e podem ser representados por uma única regra de inferência, I_{AND} . O RNF decomposto $nfr1$ é a entrada para a regra de inferência I_{AND1} e os RNFs mais específicos $nfr2$ e $nfr3$ são as saídas.

Objetivos de Transparência de Software

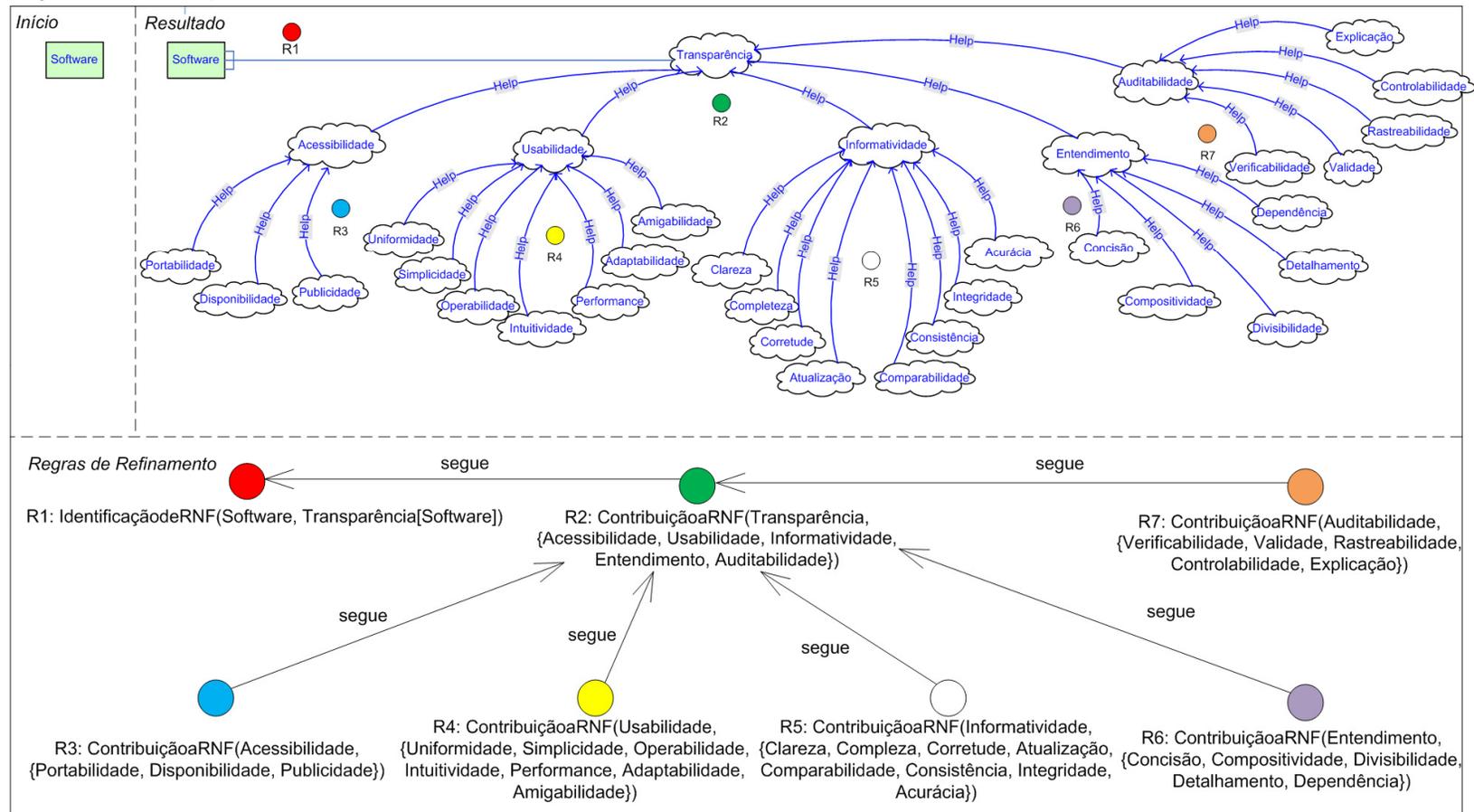


Figura 5.2 - Padrão de requisitos “Objetivos de Transparência de Software”, extraído do CTS (CTS 2011).

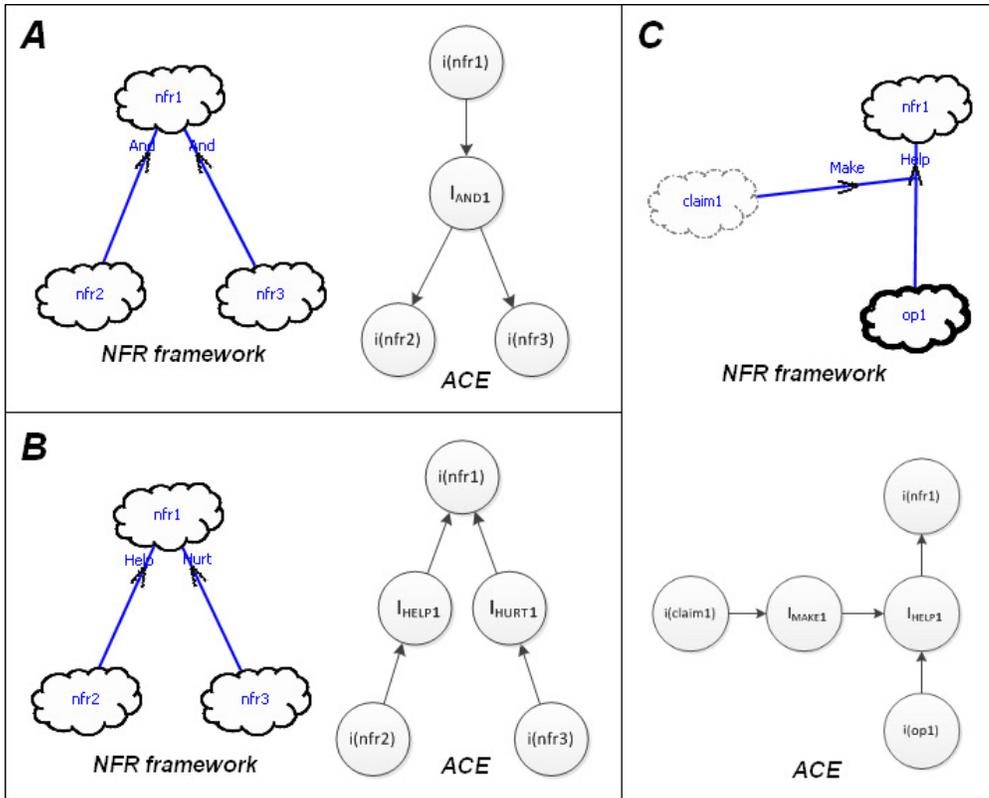


Figura 5.3 - As três estruturas SIGs recorrentes representadas como proposições na linguagem ACE

A estrutura de contribuição é identificada por um ou mais RNFs, operacionalizações ou *claims* que, positivamente, ou negativamente, impactam em um RNF. Essa estrutura consiste em enfatizar a interdependência entre RNFs e como as operacionalizações impactam nos RNFs. A Figura 5.3-B ilustra um RNF *nfr1* sendo impactado positivamente por um RNF *nfr2* e, negativamente, por um RNF *nfr3*. Os RNFs *nfr1*, *nfr2* e *nfr3* são expressos como vértices de informações na linguagem ACE. Os elos de contribuição não são semanticamente idênticos, e precisam ser representados por duas regras de inferência. O elo de contribuição *Help* é representado pela regra de inferência I_{HELP} . O elo de contribuição *hurt* é representado pela regra de inferência I_{HURT} . O RNF impactado *nfr1* é a entrada para a regra de inferência. O RNF *nfr2* é a saída para a regra de inferência I_{HELP1} , e o RNF *nfr3* é a saída para a regra de inferência I_{HURT1} .

A estrutura *claim* é identificada por um *claim* e um elo de contribuição para um elo de contribuição. Essa estrutura consiste em um *claim* argumentando a favor ou contra uma estrutura de contribuição. A Figura 5.3-C mostra um *claim* *claim1* argumentando a favor de um elo de contribuição *Help* de uma

operacionalização *op1* para um RNF *nfr1*. A estrutura de contribuição é representada por uma regra de inferência I_{HELP} e sua entrada *op1* e saída *nfr1*. O *claim claim1* é expresso como um vértice de informação $i(\textit{claim1})$ na linguagem ACE. Um elo de contribuição *Make* é representado pela regra de inferência I_{MAKE} . O *claim claim1* é a entrada para a regra de inferência I_{MAKE1} . A regra de inferência I_{HELP1} é a saída para a regra de inferência I_{MAKE1} , uma vez que o vértice de informação *claim1* representa um argumento a favor da estrutura de contribuição.

As traduções ilustradas na Figura 5.3 podem ser usadas como *templates*, ou amostras, para facilitar o processo de tradução do SIG do NFR Framework para o *framework* ACE. Uma vez que os requisitos e as preocupações de transparência estão representados como proposições na linguagem ACE, precisamos definir uma condição de aceitabilidade (a condição de parada da discussão), antes de capturar os argumentos dos interessados a favor ou contra os requisitos de transparência.

5.2. Condição de Aceitabilidade

A aceitabilidade dos requisitos de transparência sinaliza se existe um acordo entre os interessados que participaram na discussão e pode ser entendida como a condição de parada da discussão. Somente é possível discutir como definir a condição de aceitabilidade dos requisitos de transparência caso existam alguns requisitos modelados no *framework i** já traduzidos para a linguagem ACE.

Baseamos a discussão dessa seção em um exemplo usando a aplicação *Web Lattes-Scholar*. *Lattes-Scholar* (Lattes-Scholar 2011a; Lattes-Scholar 2011b) é um serviço de Internet composto por dois outros serviços de Internet: o bem conhecido serviço de Internet Google Scholar e o serviço de Internet Lattes (Lattes 2011), o qual é uma base de dados de currículos e instituições tecnológicas e científicas mantida pelo Governo Brasileiro. Veja o Capítulo 7 para mais detalhes sobre o *Lattes-Scholar*. O *Lattes-Scholar*: (i) busca o currículo de um pesquisador no Lattes, (ii) recupera do Lattes as publicações contidas no currículo do pesquisador, e (iii) ordena as publicações pelo número de citações obtidas usando o Google Scholar. A Figura 5.4 mostra os requisitos – *Late-Requirements* – do *Lattes-Scholar* para o passo (i), modelados no *framework i**.

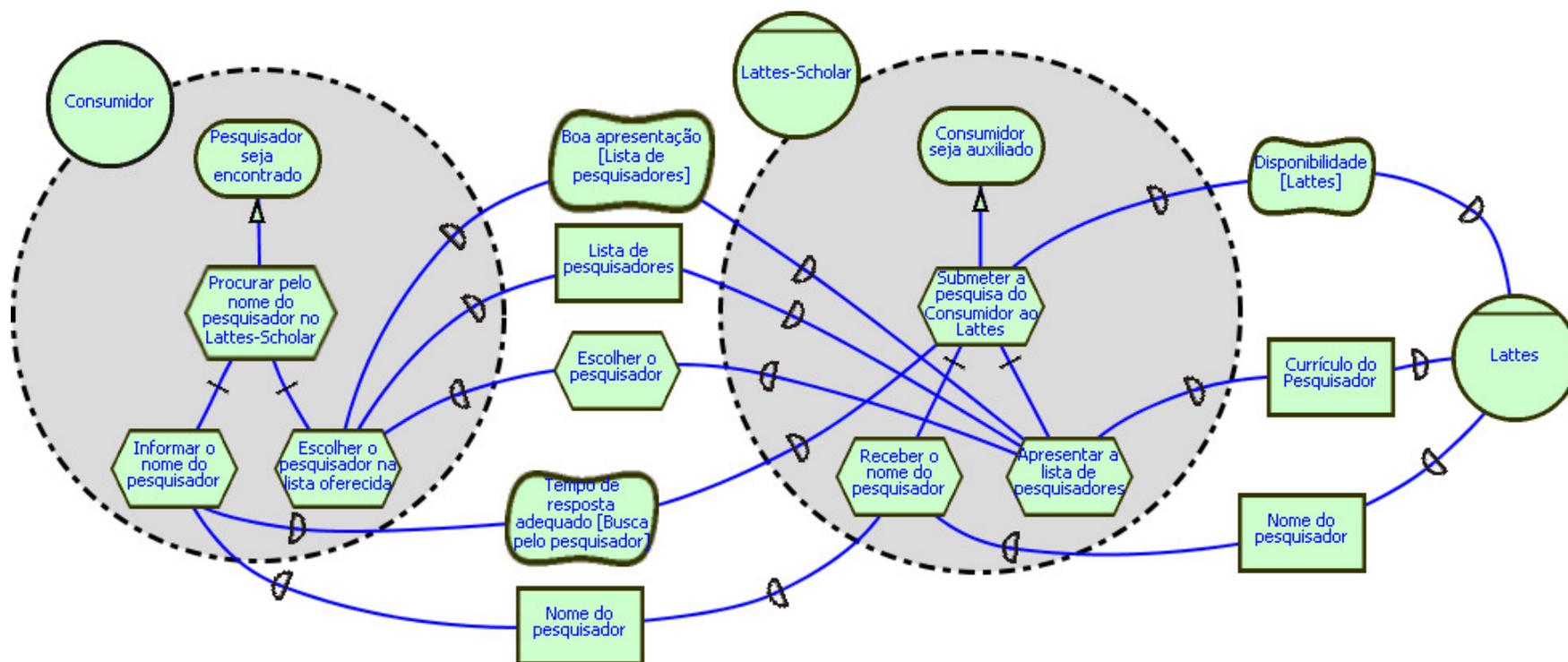


Figura 5.4 - Os late-requirements do Lattes-Scholar modelados no framework i*

A partir do modelo da Figura 5.4, escolhemos uma parte para ser utilizada como exemplo. O objetivo é analisar uma estrutura i^* meios-fim do ator “Consumidor” em termos de transparência. Em outras palavras, analisar o quão transparente é a descrição dessa estrutura. Dessa forma, um grupo de interessados provê argumentos a favor ou contra a transparência da estrutura i^* meios-fim. “Pesquisador seja recuperado” ($g1$ na Figura 5.5) é uma meta que pode ser atingida executando a tarefa “Procurar pelo nome do pesquisador no Lattes-Scholar” ($t1$ na Figura 5.5). Como previamente apresentado na Seção 5.1, a estrutura meios-fim do *framework* i^* pode ser traduzida para a linguagem ACE como uma proposição – Figura 5.5.



Figura 5.5 - A estrutura i^* meios-fim da Figura 5.4 traduzida para a linguagem ACE

A linguagem ACE (Jureta et al. 2009) define que a condição de aceitabilidade (*Acceptability Condition* - AC) é expressa por $AC(ID, T(ID), OD)$ se e somente se $AC(ID)$ E $AC(T(ID))$ E $AC(OD)$ [(i.e., a AC de um método de transformação da ER é a AC das entradas do método de transformação (ID) e a AC do método de transformação em si ($T(ID)$) e a AC da saída do método de transformação (OD)]. Na estrutura meios-fim da Figura 5.5, isso significa:

$AC(i(g1), I_{ME1}, \{i(t1)\})$ se e somente se $AC(i(g1))$ E $AC(I_{ME1})$ E $AC(i(t1))$.

Essa condição de aceitabilidade significa que a estrutura i^* meios-fim é aceitável se nem a regra de interência I_{ME1} , sua entrada $i(g1)$ ou sua saída $i(t1)$ foram atacadas/ameaçadas por conflitos não dominados (vide Capítulo 2 – Seção 2.4). Uma vez que temos a definição da condição de parada da discussão, podemos começar a capturar os argumentos dos interessados sobre os requisitos de transparência.

5.3. Captura dos Argumentos

Para que seja possível validar os requisitos de transparência, precisamos emaranhá-los de alguma forma com o modelo i^* que captura os requisitos do software. Também é necessário ligá-los aos argumentos dos interessados. Isso é feito com elos de contribuição que partem dos elementos do modelo de requisitos e que impactam nos elementos do modelo de transparência. Esses elos representam contribuições positivas ou negativas de decisões de desenho específicas em relação à transparência. Referimos-nos a esse grafo como Grafo D. Esse grafo será traduzido para a linguagem ACE e terá sua condição de aceitabilidade analisada para determinar se a validade relativa foi obtida (vide Seção 5.4).

Os elos de contribuição podem ser elicitados mostrando-se aos interessados suas especificações de metas/tarefas e os padrões de requisitos obtidos a partir do CTS. É pedido aos interessados que os mesmos marquem, a partir de sua própria perspectiva, as contribuições das tarefas para as metas flexíveis. Nesse estágio, cada interessado apresenta seu próprio ponto de vista sobre transparência. Esse ponto de vista será depois discutido com outros interessados nos estágios subsequentes. O Grafo D representa os pontos de vista dos interessados sobre a transparência das suas especificações de metas e tarefas. Para validar relativamente a transparência do Grafo D, aplicamos um processo iterativo e interativo onde os interessados ampliam o Grafo D com novos argumentos.

Referimos-nos à versão inicial do grafo D como grafo $D(0)$. No passo i cada interessado amplia $D(i-1)$ com novos argumentos, resultando em um novo grafo $D(i)$. O grafo $D(i)$ contém mais argumentos a favor ou contra à transparência do desenho proposto do que o grafo $D(i-1)$. O processo termina quando $D(n-1) = D(n)$, isto é, quando nenhum dos interessados possui novos argumentos para enriquecer o grafo $D(n-1)$.

A Figura 5.6 mostra um grafo inicial $D(0)$ e a sua evolução $D(1)$ durante uma curta discussão que termina quando $D(2)=D(3)$ (Figura 5.7). A árvore de NFRs representa parte do SIG de Transparência, enquanto a estrutura de metas e tarefas é a estrutura meios-fim do modelo i^* do ator “Consumidor” da Figura 5.4

(Seção 5.2). Os interessados A, B e C participam da discussão. Um conflito pode ser observado no grafo D(3), onde o interessado C discorda de A sobre a contribuição do tipo *Hurt* da tarefa “Pesquisar o pesquisador pelo nome” para o RNF Validade. Depois do argumento do interessado A, não existem novos argumentos, portanto: $D(2)=D(3)$.

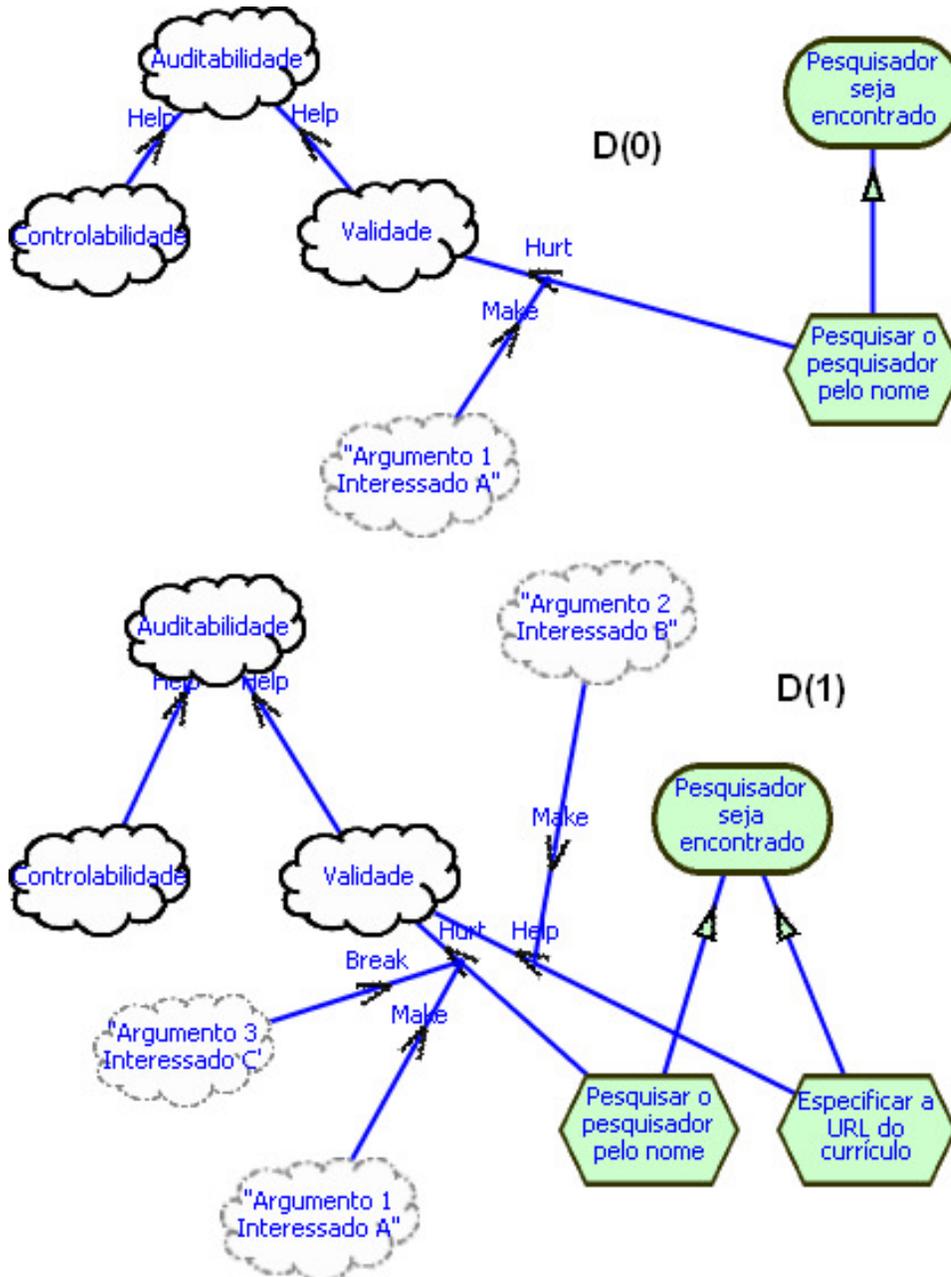


Figura 5.6 - A evolução do Grafo D de D(0) até D(1)

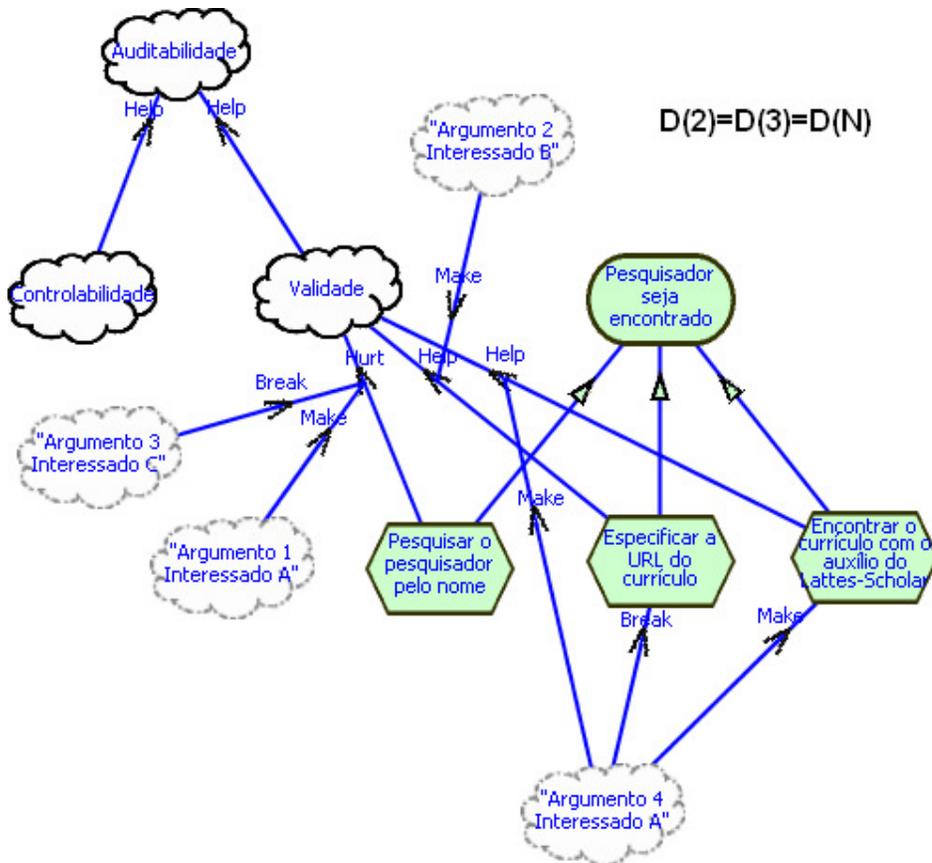


Figura 5.7 - Grafo resultante $D(2)=D(3)=D(N)$

O resultado desse processo é o Grafo $D(N)$. Na próxima seção, traduzimos os grafos para a linguagem ACE para analisar se existe um acordo entre as partes e analisamos a condição de aceitabilidade desses grafos.

5.4. Validação Relativa dos Requisitos de Transparência

Nessa seção, desenvolveremos nosso exemplo validando relativamente o grafo com os argumentos dos interessados sobre os requisitos de transparência. O SIG parcial de transparência, apresentado na Figura 5.8, foi extraído do padrão de requisitos “Objetivos de Transparência de Software”, mostrado na Figura 5.2. Esse padrão de requisitos foi obtido do Catálogo de Transparência de Software (CTS) e será usado como um guia para transparência pelos interessados. Nesse exemplo, os argumentos dos interessados estão focados no RNF Validade e no RNF Simplicidade. O RNF Clareza também aparece no SIG da Figura 5.6, pois é o foco do nosso segundo exemplo, a ser apresentado mais adiante nesse capítulo.

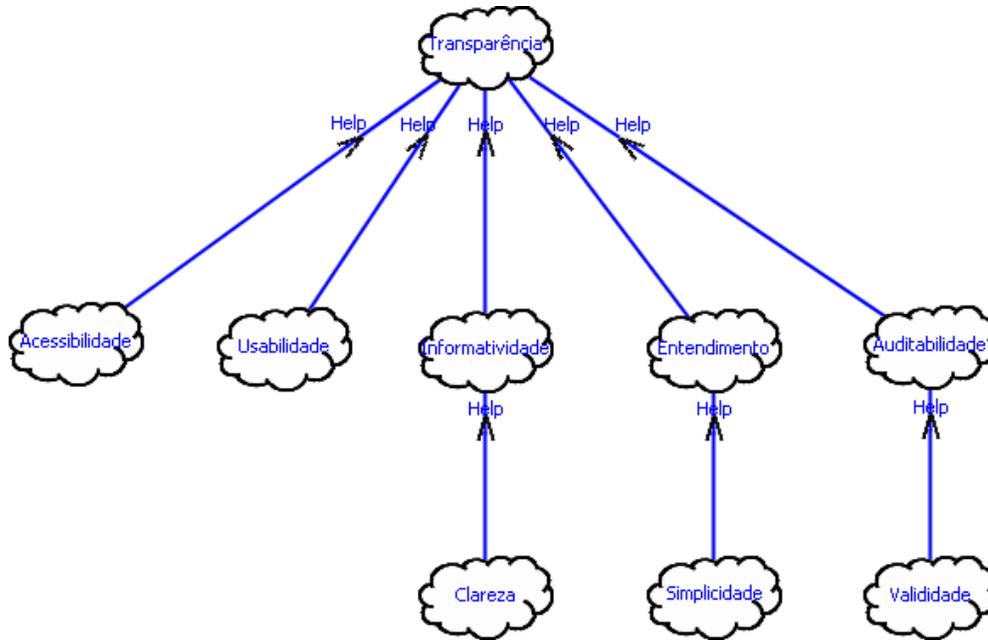


Figura 5.8 - Um SIG parcial de Transparência

A discussão dos interessados ocorre durante a validação relativa da estrutura i^* meios-fim da Figura 5.5. Alguns interessados argumentam que existe um conflito entre a tarefa $t1$ e o RNF Validade, uma vez que (i) a questão 1.5 para o RNF Validade (Seção 2.3, Tabela 2.1) verifica se as fontes de informação foram identificadas, (ii) uma pesquisa por nome no serviço de Internet Lattes pode recuperar mais de um currículo no caso de pesquisadores homônimos, e (iii) o ator “Consumidor” não pode validar se o currículo recuperado pelo Lattes é o currículo do pesquisador correto ou se pertence a um homônimo. Referenciaremos esse argumento como a proposição individual $p1$.

Visando representar essa discussão usando a linguagem ACE, o SIG parcial de Transparência da Figura 5.8 precisa ser expresso como proposições (vide Seção 5.1). Na Figura 5.9, a regra de inferência:

$$I_{\text{HELP1}}(\{i(nfr2),i(nfr3),i(nfr4),i(nfr5),i(nfr6)\}, i(nfr1))$$

representa uma contribuição do tipo *Help* associada aos RNFs (de $nfr2$ até $nfr6$) que contribuem para o RNF Transparência ($i(nfr1)$). $I_{\text{HELP2}}(i(nfr7),i(nfr4))$, $I_{\text{HELP3}}(i(nfr8),i(nfr5))$ e $I_{\text{HELP4}}(i(nfr9),i(nfr6))$ são análogos ao I_{HELP1} . A regra de inferência $I_{\text{ME1}}(i(g1),i(t1))$ representa uma decomposição meios-fim da meta $g1$ na tarefa $t1$. O argumento $p1$ do interessado é representado no grafo como um vértice de informação $i(p1)$. Esse argumento apóia (regra de inferência $I_{T1}(i(p1),C1)$) um

conflito (C1) entre o RNF Validade ($i(nfr9)$) e a tarefa $t1$ ($i(t1)$). Esse conflito está representado como um vértice de conflito $C1(i(nfr9),i(t1))$.

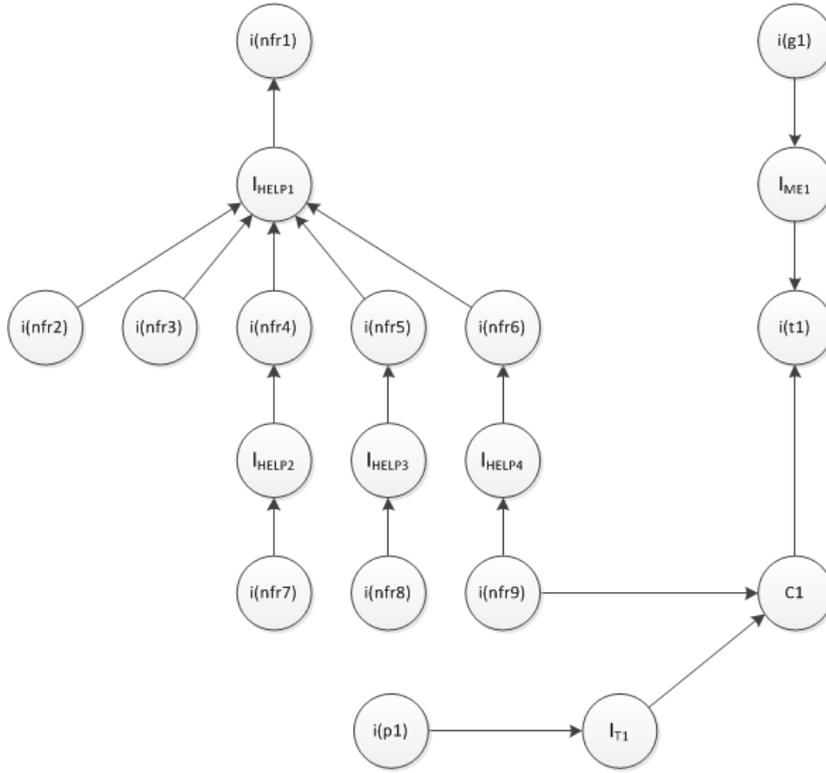


Figura 5.9 - Um conflito (C1) entre o RNF Validade ($i(nfr9)$) e a tarefa $t1$ ($i(t1)$)

O *framework* ACE provê um algoritmo para avaliar a aceitabilidade do grafo. O algoritmo identifica que a tarefa $t1$ ($i(t1)$) recebe uma aresta do vértice de conflito C1 e, por definição, está sendo atacada (i.e. *attacked*). O vértice de informação $i(t1)$ é etiquetado como rejeitado (i.e. *rejected*) e a $AC(i(t1))$ é falsa. Dessa forma,

$AC(i(gI), I_{ME1}, i(t1))$ se, e somente se, $AC(i(gI))$ E $AC(I_{ME1})$ E $AC(i(t1))$,

$AC(i(gI), I_{ME1}, i(t1))$ se, e somente se, verdadeiro E verdadeiro E falso,

$AC(i(gI), I_{ME1}, i(t1)) = \text{falso}$.

A condição de aceitabilidade significa que a estrutura i^* meios-fim não é aceitável e o modelo i^* não está transparente de acordo com os interessados. O grafo indica que serão necessárias discussões futuras visando evoluir o modelo i^* e melhorar sua transparência.

Em outra discussão, um interessado sugere (proposição individual $p2$) permitir ao ator “Consumidor” especificar exatamente a URL do currículo de um pesquisador. A proposição $p2$ infere (regra de inferência $I_{T2}(i(p2),\{i(t2),I_{ME2}\})$) uma nova tarefa $t2$ e uma estrutura meios-fim alternativa I_{ME2} visando atingir a

meta $g1$, como ilustrado na Figura 5.10. O ator “Consumidor” terá uma nova tarefa ($t2$) para atingir a meta $g1$. Com essa alternativa, o ator pode validar se o currículo encontrado pelo Lattes-Scholar é o currículo correto. A proposição ($p2$) também cria um conflito $C2(i(p2),\{C1, I_{ME1}\})$ com o conflito $C1$ e a estrutura i^* meios-fim original I_{ME1} . O algoritmo de avaliação de aceitabilidade agora etiqueta o vértice de conflito $C1$ e a regra de inferência I_{ME1} como rejeitados (i.e. *rejected*). Dessa forma, a $AC(I_{ME1})$ é falsa e a estrutura meios-fim original não é aceitável. Entretanto, se considerarmos a AC da segunda estrutura meios-fim,

$$AC(i(g1), I_{ME2}, \{i(t1),i(t2)\}) \text{ se, e somente se, } AC(i(g1)) \text{ E } AC(I_{ME2}) \text{ E } AC(\{i(t1),i(t2)\}),$$

$$AC(i(g1), I_{ME2}, \{i(t1),i(t2)\}) \text{ se, e somente se, verdadeiro E verdadeiro E verdadeiro,}$$

$$AC(i(g1), I_{ME2}, \{i(t1),i(t2)\}) = \text{verdadeiro.}$$

A condição de aceitabilidade significa que $p2$ é uma possível solução, uma vez que a segunda estrutura meios-fim é aceitável.

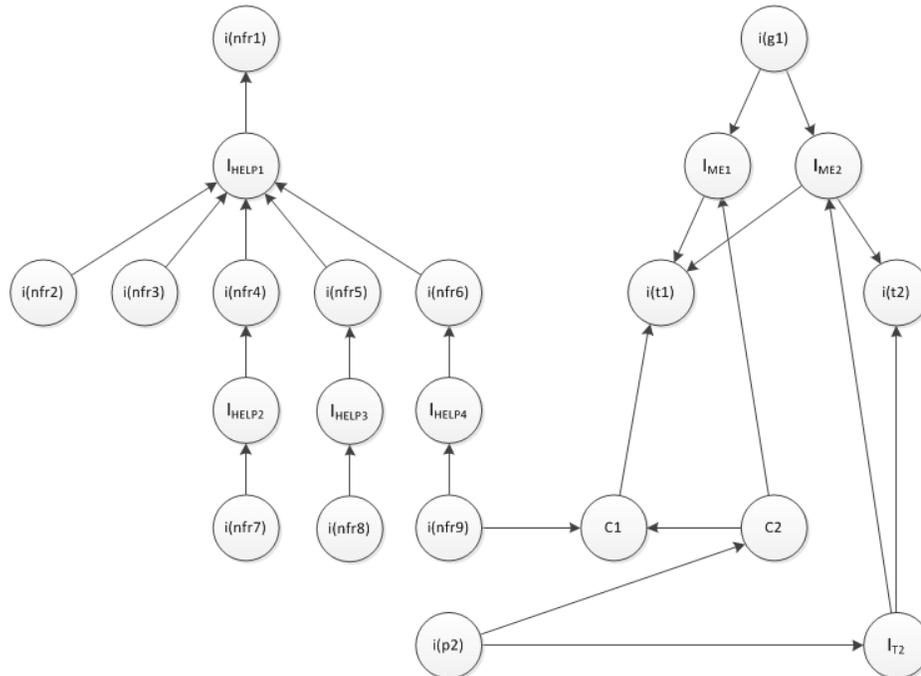


Figura 5.10 - O argumento $p2$ de um interessado a favor de uma estrutura meios-fim alternativa

Consideramos o argumento de um terceiro interessado (proposição individual $p3$) de que existe um conflito entre o RNF Simplicidade e a tarefa $t2$, uma vez que: (i) a questão 1.2 (do CTS) para o RNF Simplicidade verifica se não existe necessidade de acesso a recursos externos, e (ii) o Lattes-Scholar não ajuda

o ator “Consumidor” a obter a URL do currículo correto no Lattes. A Figura 5.11 mostra que a proposição $p3$ infere $(I_{T3}(i(p3),C3))$ um conflito $C3(i(nfr8),i(t2))$ entre o RNF Simplicidade $(i(nfr8))$ e a tarefa $t2$ $(i(t2))$. Um quarto interessado argumenta que o Lattes-Scholar poderia ajudar o ator “Consumidor” abrindo uma janela do tipo *pop-up* para a pesquisa no serviço de Internet Lattes. Essa nova proposição individual $p4$ infere $(I_{T4}(i(p4),\{i(t3),i(I_{ME3})\}))$ uma nova tarefa $t3$ e uma nova estrutura *i** meios-fim $I_{ME3}(i(g1),\{i(t1),i(t3)\})$. A Figura 5.11 também mostra que a proposição $p4$ conflita $(C4(i(p4), I_{ME2}))$ com a segunda estrutura meios-fim I_{ME2} . O algoritmo de avaliação da condição de aceitabilidade agora etiqueta a regra de inferência I_{ME2} como rejeitada (i.e. *rejected*). A $AC(I_{ME2})$ é falsa, e a segunda estrutura meios-fim não é aceitável. Entretanto, se considerarmos a AC da terceira estrutura meios-fim,

$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\})$ se, e somente se, $AC(i(g1))$ E $AC(I_{ME3})$ E $AC(\{i(t1), i(t3)\})$,

$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\})$ se, e somente se, verdadeiro eE verdadeiro E verdadeiro,

$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\}) = \text{verdadeiro}$.

A condição de aceitabilidade indica que essa também é uma solução possível, uma vez que a terceira estrutura meios-fim é aceitável.

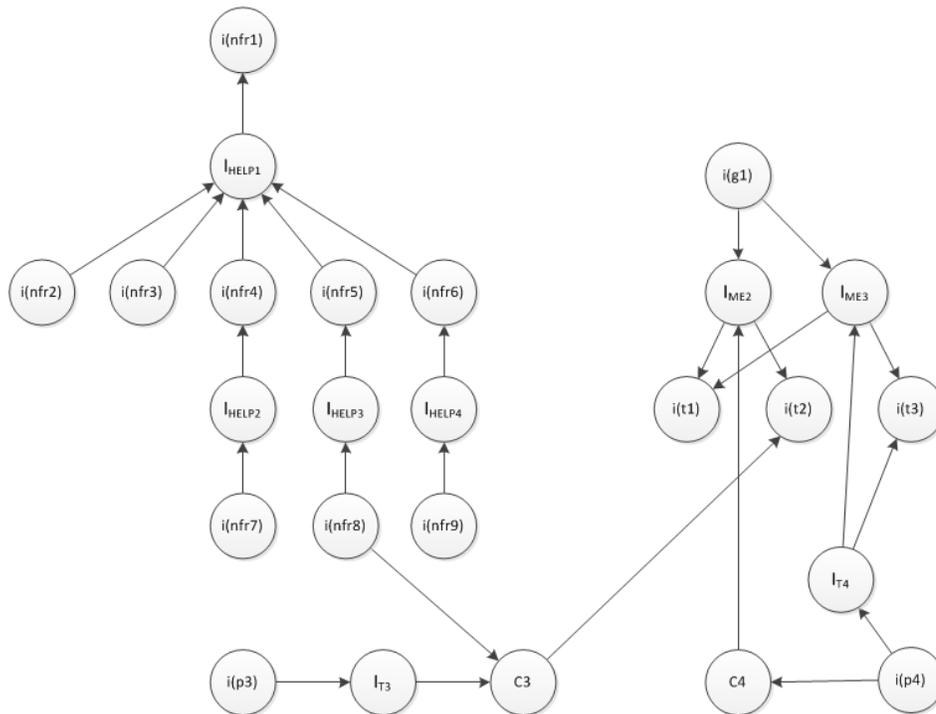


Figura 5.11 - Argumentos ($p3$ e $p4$) de dois interessados

Note que ambas as proposições $p2$ e $p4$ são mutuamente exclusivas. A Figura 5.12 mostra que essa exclusão mútua é representada no grafo como dois vértices de conflitos $C5(i(p4),i(p2))$ e $C6(i(p2),i(p4))$. Caso não existam mais argumentos a serem colocados pelos interessados, os interessados precisam agora escolher se eles preferem a proposição $p2$ ou a proposição $p4$. Eles precisam votar na proposição preferida, e esses votos podem ser ponderados, caso um grupo de interessados seja mais relevante que os demais para a validação dos requisitos de transparência. Assumimos nesse exemplo que todos os interessados possuem a mesma relevância. Se a maioria dos interessados prefere $p4$ à $p2$, isso é representado como um vértice de regra de preferência $P1(i(p4),i(p2))$. Se o argumento $p2$ é rejeitado, o conflito $C6$ precisa também ser rejeitado. Dessa forma, a regra de preferência $P1$ também implicaria em um vértice de conflito $C7(P1,C6)$ que rejeita (i.e. *rejects*) o conflito $C6$.

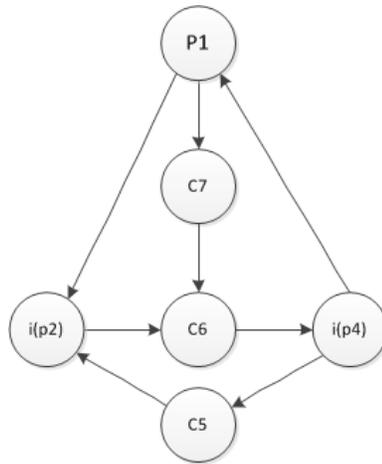


Figura 5.12 - A regra de preferência ilustrando a preferência pela proposição $p4$

Esse exemplo resulta na evolução do modelo i^* , mostrada na Figura 5.13.

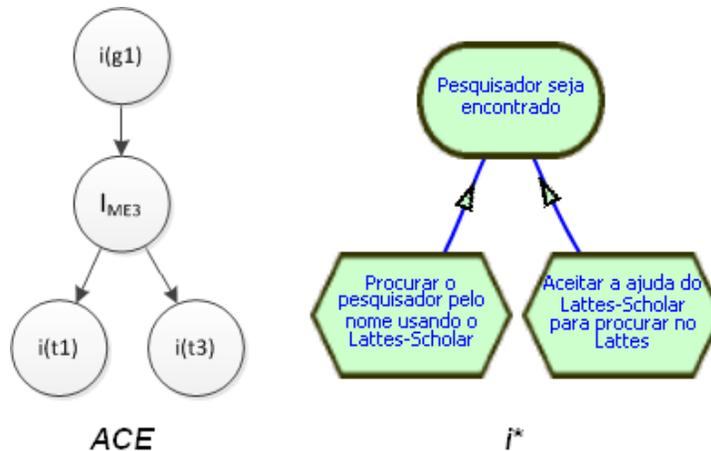


Figura 5.13 - A estrutura i^* meios-fim evoluída

Esse exemplo mostrou como validar relativamente os requisitos de transparência e como rastrear as proposições dos interessados que obtiveram uma validade aceitável. O grafo ACE resultante da Figura 5.13 foi finalmente traduzido para o *framework* i^* , visando obter o modelo evoluído. É importante notar que essa abordagem também pode ser usada para validar os requisitos de transparência das estruturas representacionais de linguagem. A seguir, apresentamos um segundo exemplo, no qual os interessados argumentam e contra-argumentam sobre a clareza da estrutura i^* meios-fim, uma decomposição implícita do tipo OU.

Vamos supor que os interessados estejam validando os requisitos de transparência da estrutura i^* meios-fim da Figura 5.13. Considerando que um interessado argumente (proposição individual $p5$) que existe um conflito entre a estrutura i^* meios-fim I_{ME3} e o RNF Clareza (Figura 5.15, $i(nfr7)$), uma vez que (i) a questão 4.2 (extraída do CTS) pesquisa se os modelos de sequencia são compreendidos para o caso do RNF Clareza, (ii) não é claro se um ator “Consumidor” pode fazer ambas as tarefas $t1$ e $t3$ ao mesmo tempo ou precisa escolher entre executar $t1$ ou $t3$, e (iii) o consumidor pode abrir o Lattes-Scholar em duas abas do navegador *Web* e realizar ambas as tarefas ao mesmo tempo. A Figura 5.14 ilustra o argumento de um interessado $i(p5)$. Esse argumento apóia (regra de inferência $I_{T5}(i(p5),C8)$) um conflito entre o RNF Clareza ($i(nfr7)$) e a estrutura i^* meios-fim (I_{ME3}). Esse conflito está representado como um vértice de conflito $C8(i(nfr7),I_{ME3})$.

O algoritmo de aceitabilidade identifica que o vértice da regra de inferência I_{ME3} recebe uma linha a partir do vértice de conflito $C8$ e, por definição, está sendo atacado (i.e. *attacked*). O vértice da regra de inferência I_{ME3} é etiquetado como rejeitado (i.e. *rejected*) e a $AC(I_{ME3})$ é falsa. Dessa forma,

$$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\}) \text{ se, e somente se, } AC(i(g1)) \text{ E } AC(I_{ME3}) \text{ E } AC(\{i(t1), i(t3)\}),$$

$$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\}) \text{ se, e somente se, verdadeiro E falso E verdadeiro, } AC(i(g1), I_{ME3}, \{i(t1), i(t3)\}) = \text{falso.}$$

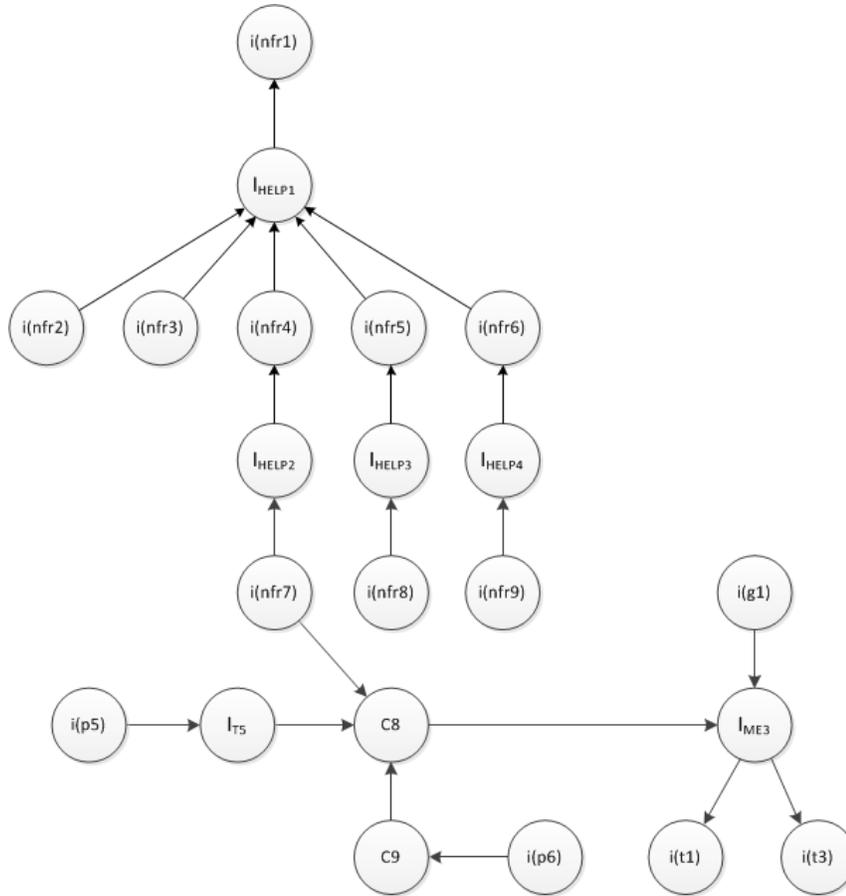


Figura 5.14 - Um conflito (C8) entre o RNF Clareza ($i(nfr7)$) e a estrutura i^* meios-fim (I_{ME3})

A condição de aceitabilidade significa que a estrutura meios-fim não é aceitável e o modelo i^* não é transparente de acordo com os interessados. É necessário continuar a discussão com os interessados no intuito de obter a validade relativa dos requisitos de transparência.

Supondo que outro interessado sugira (proposição individual $p6$) manter a mesma estrutura e incluir uma nota no modelo i^* que deixa mais claro que o ator “Consumidor” pode executar as tarefas $t1$ e $t3$ concorrentemente, visando atingir a meta $g1$. Esse argumento ($p6$) cria um conflito $C9(i(p6), C8)$ com o conflito C8, conforme ilustrado na Figura 5.14. O algoritmo de avaliação de aceitabilidade agora etiqueta o vértice de conflito C8 como rejeitado (i.e. *rejected*) e a regra de inferência I_{ME3} como aceita (i.e. *accepted*). Portanto, a $AC(I_{ME3})$ é falsa. Dessa forma,

$$AC(i(g1), I_{ME3}, \{i(t1), i(t3)\}) \text{ se, e somente se, } AC(i(g1)) \text{ E } AC(I_{ME3}) \text{ E } AC(\{i(t1), i(t3)\}),$$

$AC(i(gI), I_{ME3}, \{i(t1), i(t3)\})$ se, e somente se, verdadeiro E verdadeiro E verdadeiro,

$$AC(i(gI), I_{ME3}, \{i(t1), i(t3)\}) = \text{verdadeiro}.$$

A condição de aceitabilidade significa que $p6$ é uma solução possível, uma vez que a estrutura meios-fim é agora aceitável.

Considerando que outro interessado argumente (proposição individual $p7$) a favor da substituição da estrutura i^* meios-fim por uma estrutura OU-exclusiva (i.e. *exclusive-OR (XOR)*) – vide Figura 5.15. A possibilidade de executar as tarefas $t1$ e $t3$ concorrentemente é representada no modelo i^* inicial (Figura 5.13) com uma terceira tarefa “ $t1$ e $t3$ ”. Esse argumento ($p7$) infere (I_{T6}) a tarefa $i(t1 \text{ and } t3)$ e a regra de inferência $I_{ME4}(i(gI), \{i(t1), i(t3), i(t1 \text{ and } t3)\})$. Esse argumento também gera um conflito $C10(i(p7), I_{ME3})$ com a estrutura i^* meios-fim substituída (regra de inferência I_{ME3}). O algoritmo de avaliação de aceitabilidade agora etiqueta a regra de inferência I_{ME4} como aceita (i.e. *accepted*), e a regra de inferência I_{ME3} como rejeitada (i.e. *rejected*). A $AC(I_{ME3})$ é falsa, e a primeira estrutura meios-fim não é aceitável. Portanto, caso consideremos a AC da estrutura meios-fim do tipo OU-exclusiva (i.e. *exclusive-OR (XOR)*), I_{ME4} ,

$AC(i(gI), I_{ME4}, \{i(t1), i(t3), i(t1 \text{ and } t3)\})$ se, e somente se,

$$AC(i(gI)) \text{ E } AC(I_{ME4}) \text{ E } AC(\{i(t1), i(t3), i(t1 \text{ and } t3)\}),$$

$AC(i(gI), I_{ME4}, \{i(t1), i(t3), i(t1 \text{ and } t3)\})$ se, e somente se, verdadeiro E verdadeiro E verdadeiro,

$$AC(i(gI), I_{ME4}, \{i(t1), i(t3), i(t1 \text{ and } t3)\}) = \text{verdadeiro}.$$

A condição de aceitabilidade significa que essa é também uma solução possível, uma vez que a estrutura meios-fim do tipo Ou-exclusivo é aceitável.

É relevante notar que ambos os argumentos $p6$ e $p7$ são mutuamente exclusivos. A Figura 5.16 mostra que essa exclusão mútua é representada no grafo como dois vértices de conflitos $C11(i(p7), i(p6))$ e $C12(i(p6), i(p7))$. Caso a maioria dos interessados prefira $p7$ a $p6$, isso é representado como um vértice de regra de preferência $P2(i(p7), i(p6))$. Se o argumento $p6$ é etiquetado como rejeitado (i.e. *rejected*), o conflito $C12$ precisa também ser rejeitado (i.e. *rejected*). Dessa forma, a regra de preferência $P2$ também implica no vértice de conflito $C13(P2, C12)$ que domina (i.e. *dominates*) o conflito $C12$.

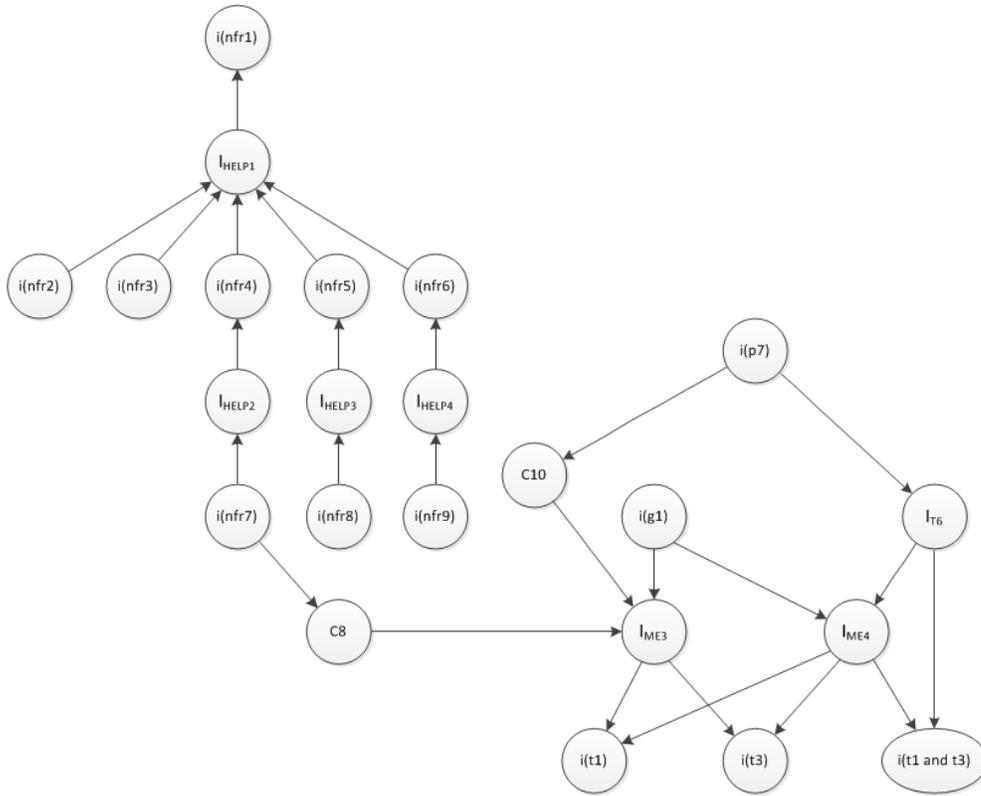


Figura 5.15 - Um argumento $p7$ de um interessado a favor da substituição da estrutura meios-fim I_{ME3} por uma estrutura meios-fim do tipo Ou-exclusivo

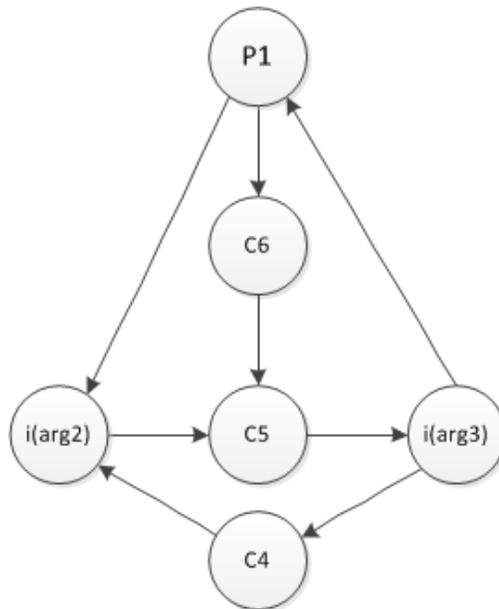


Figura 5.16 - A regra de preferência ilustrando a preferência pela proposição $p7$

Esse exemplo resulta em uma possível evolução do modelo i^* , mostrada na Figura 5.17.

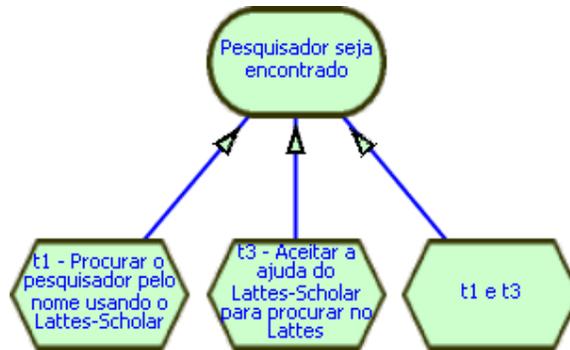


Figura 5.17 - A estrutura i* meios-fim evoluída

5.5. Apoio Ferramental

A ferramenta C&L (C&L 2011) foi utilizada para apoiar a construção do grafo ACE. A ferramenta C&L foi originalmente desenhada/projetada para ajudar na construção do Léxico Estendido da Linguagem (LEL). O LEL oferece a denotação e conotação dos termos da linguagem de ambiente da aplicação. O C&L ajuda na compreensão da aplicação usando linguagem natural. A ferramenta está disponível na *Web* com suporte ao trabalho colaborativo e automatizando na manutenção de elos entre os termos da aplicação.

O LEL apoiou a manutenção dos termos da linguagem ACE. As características hiper-textuais do C&L e da ferramenta de geração de grafo beneficiaram a geração automática dos grafos ACE. Cada símbolo do LEL é descrito por nome, noção, classificação, sinônimos e impactos. O nome do símbolo é usado como o identificador, ou o nome de um nó do grafo ACE. Exploramos o campo noção (a denotação) dos termos para explicar cada regra de inferência, regra de conflito, regra de preferência ou vértice de informação do ACE. Nossos símbolos foram classificados como objetos, caso eles fossem nós do grafo ACE; ou como sujeitos, caso eles representassem interessados que participaram da discussão dos requisitos de transparência. O campo “sinônimos” não foi explorado, uma vez que usamos a ferramenta C&L para manter termos da linguagem ACE, não da linguagem natural. Finalmente, o campo “impactos” é o mais importante campo do símbolo, uma vez que a conotação de um nó do grafo ACE representa os elos direcionais entre os nós. Visando obter elos direcionais com a direção correta, adaptamos a gramática do LEL, incluindo os termos “apontados” no campo “impactos”. A ferramenta C&L mostra os elos entre os

símbolos como elos hiper-textuais navegáveis. É possível navegar do símbolo “iv-p4” (mostrado na Tabela 5.1) para o símbolo “ir-t4” (mostrado na Tabela 5.2) apenas clicando-se no segundo hiper-elo. Na página do símbolo “ir-t4”, clicando no primeiro hiper-elo, é permitido navegar para o símbolo “ir-me3” (mostrado na Tabela 5.3).

Tabela 5.1 - A informação do símbolo “iv-p4”

Name:	iv-p4
Notion:	The information vertex iv-p4 represents the proposition p4 "Lattes-Scholar could help the actor Consumer by opening on a pop-up window the Lattes search interface" proposed by the stakeholder s4.
Classification:	object
Impact(s):	The proposition p4 conflicts with the means-end structure through the conflict rule cr-4 . The proposition p4 inferences a new task and an alternative means-end structure through the inference rule ir-t4 . The proposition p4 conflicts with other proposition through the conflict rule cr-5 . The proposition p4 is preferred by the stakeholders through the preference rule pr-1 .
Synonym(s):	

Tabela 5.2 - A informação do símbolo “ir-t4”

Name:	ir-t4
Notion:	The inference rule ir-t4 represents the inference of a new task and a new means-end structure by a proposition.
Classification:	object
Impact(s):	The inference rule ir-t4 has the outputs ir-me3 and iv-t3 .
Synonym(s):	

Tabela 5.3 - A informação do símbolo “ir-me3”

Name:	ir-me3
Notion:	The inference rule ir-me3 represents an alternative means-end structure to achieve the goal g1 "Researcher Be Found" of the actor "Consumer" in the i* model.
Classification:	object
Impact(s):	The inference rule ir-me3 has the outputs iv-t1 and iv-t3 .
Synonym(s):	

Optamos por manter os temas da linguagem ACE na ferramenta C&L visando o apoio ao trabalho colaborativo. Cada interessado pode fazer o *login* na ferramenta e inserir suas proposições como vértices de informações, enquanto um ou mais engenheiros de requisitos definem o impacto dessas proposições no grafo ACE. Outra funcionalidade muito interessante do C&L é a geração automática do grafo. O C&L usa algoritmos de visualização para, inicialmente, arranjar os nós do grafo. A Figura 5.18 ilustra um grafo ACE completo para o primeiro estudo de caso (conflitos de Auditabilidade e Simplicidade), rearranjado para, visualmente, assemelhar-se aos grafos das Figuras 5.9, 5.10, 5.11 e 5.12.

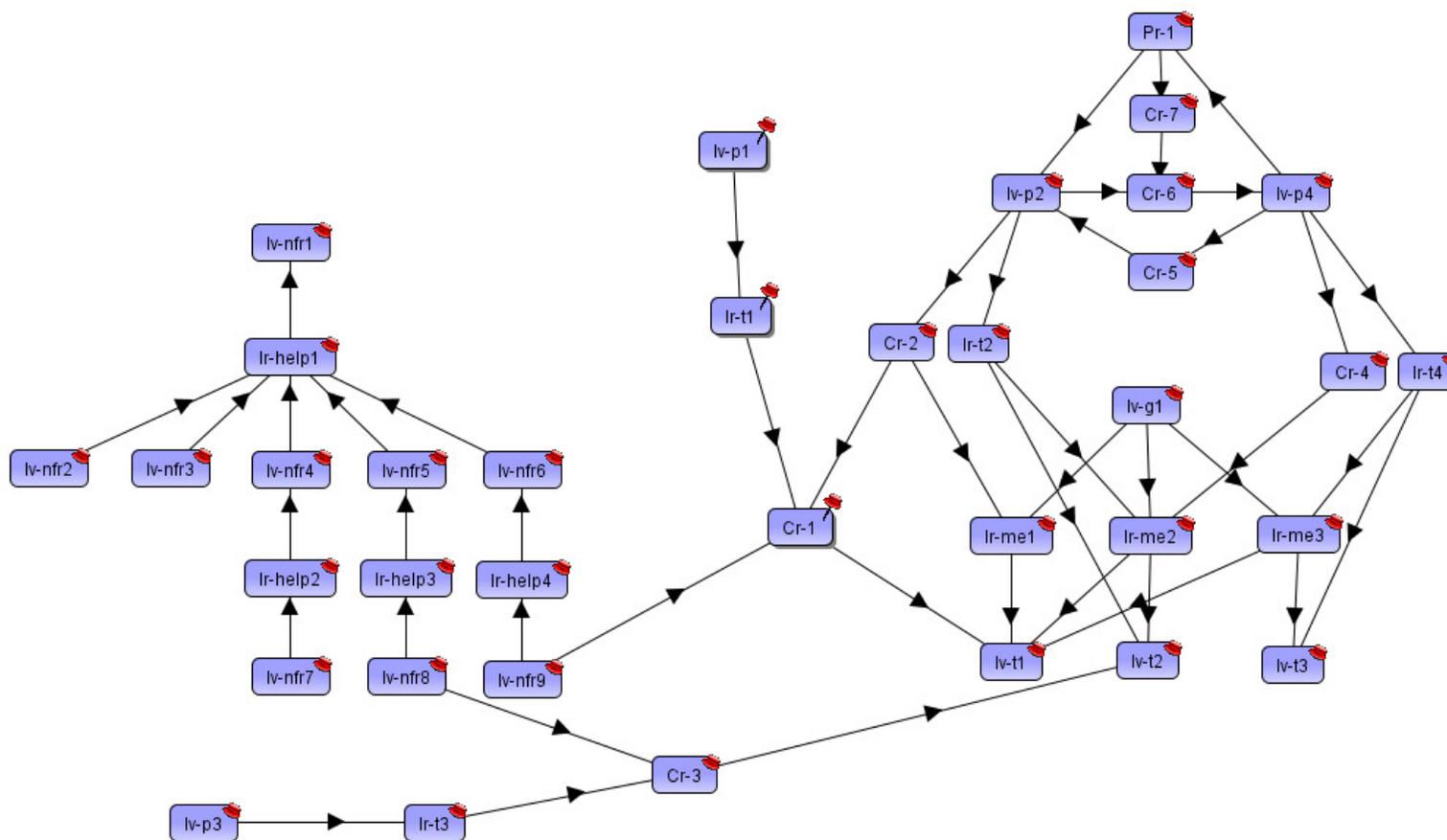


Figura 5.18 - O grafo ACE completo para o primeiro estudo de caso

Mantivemos ainda, na ferramenta C&L, o léxico dos termos do SIG de Transparência (Figura 5.19) e o léxico da linguagem ACE. Esses léxicos auxiliares ajudam a melhorar o léxico e a facilitar o entendimento do grafo ACE.

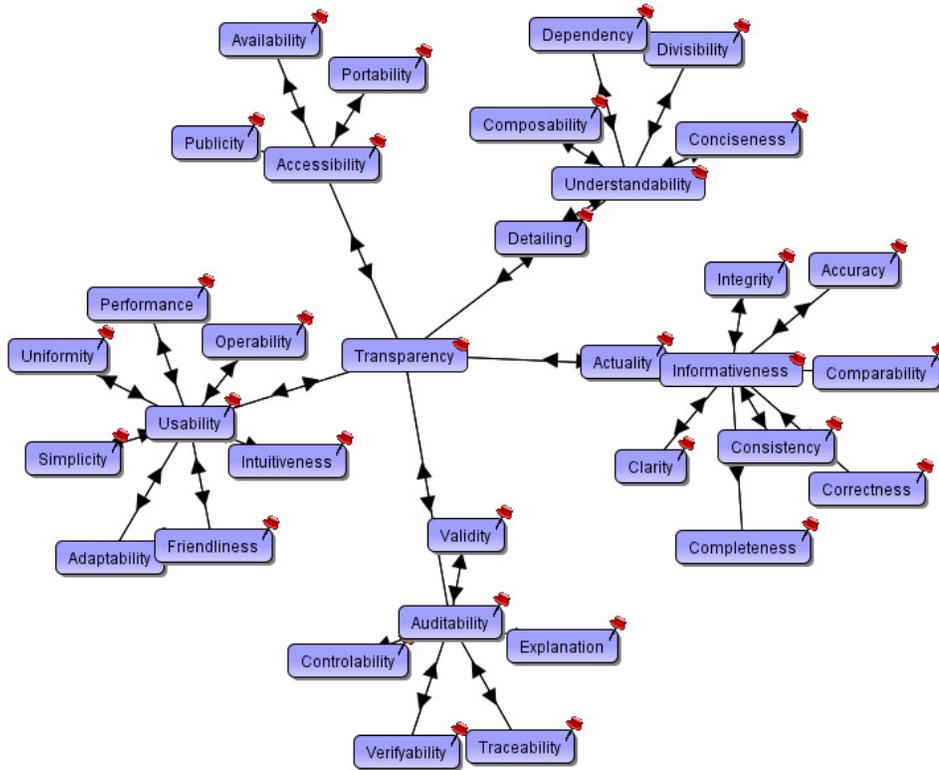


Figura 5.19 - O grafo do léxico do SIG de Transparência

A Figura 5.20 ilustra o símbolo do LEL “iv-nfr8” do nosso léxico do grafo ACE. Esse LEL foi enriquecido com elos para o LEL do SIG de Transparência (elo “b”), para o LEL da linguagem ACE (elos “a”, “e” e “h”), para o LEL do *framework* i* (elos “d” e “i”) e para o LEL do NFR Framework (elo “c”). Os elos “f” e “g” referem-se a outros símbolos do nosso LEL do grafo ACE.

Name:	iv-nfr8
Notion:	The information vertex iv-nfr8 represents the simplicity nfr/softgoal .
Classification:	object
Impact(s):	The information vertex iv-nfr8 is the input of the inference rule ir-help3 . The simplicity nfr/softgoal conflicts with a task , by the conflict rule cr-3 .
Synonym(s):	

Figura 5.20 - O símbolo “iv-nfr8” enriquecido com elos para outros símbolos do LEL

5.6. Trabalhos Relacionados

Uma possível abordagem para validação de requisitos de qualidade é apresentada por Elahi et al. (Elahi e Yu 2009). Os autores propõem um *framework* para elicitação e análise de requisitos de segurança centrado em vulnerabilidade. O *framework* apóia os desenhistas/projetistas na identificação de potenciais riscos de ataques e na análise dos comportamentos dos mesmos visando à confiabilidade do sistema. Os autores estendem o metamodelo do *framework* i* para incorporar entidades baseadas em segurança (ex. Vulnerabilidade, Impacto de Segurança, Ataque e Efeito). A abordagem é viável uma vez que o critério de qualidade segurança pode ser decomposto em poucos conceitos correlatos, com base na questão de vulnerabilidade. Além disso, a maioria dos interessados tem uma noção concreta e bem definida do que segurança significa. Outra proposta relacionada utiliza *early aspects* (Rashid et al. 2002) para promover a separação de *concerns*. Entretanto, a transparência é fortemente acoplada a outros requisitos não-funcionais (incluindo 33 interdependências com outras metas flexíveis). Nesse contexto, a validação não é tão simples quando lidando com requisitos de transparência. Em (Cysneiros e Werneck 2009), os autores argumentam que confiabilidade “é uma das características mais importantes para se atingir a transparência”. O ponto de vista deles difere de (Mercuri 2005), o que enfatiza a necessidade de se validar a transparência através de uma resolução de pontos de vista, como apresentado na Seção 5.4.

Cysneiros et al. (Cysneiros et al. 2001) utiliza um catálogo de RNFs para analisar modelos UML. Para atingir seus objetivos, os autores emaranham SIGs do NFR Framework com modelos de Casos de Uso da UML. Gross e Yu (Gross e Yu 2001) também emaranham modelos i* e SIGs de NFRs para analisar escolhas de modelagem. Adotamos uma abordagem similar de emaranhamento para analisar os requisitos de transparência.

O NFR Framework já provê um suporte básico para *claims*. O *framework* ACE estende esse suporte básico com o conceito da aceitabilidade. Gross e Yu (Gross e Yu 2009) fazem uso extensivo do operador “claim” para registrar informações assumidas durante o desenho. Registrar os argumentos dos interessados na linguagem ACE permitiu-nos não só registrar os argumentos, mas

também identificar conflitos, preferências e consensos, obtendo uma validação relativa do desenho escolhido.

Outros *frameworks* de argumentação já foram propostos nas áreas de Gerência de Conhecimento e Inteligência Artificial e poderiam ter sido utilizados para capturar os argumentos dos interessados, como (Pollock 2009) ou (Rahwan e Larson 2009). Entretanto, o *framework* ACE oferece todas as abstrações necessárias para capturar argumentos em uma discussão aberta: (1) vértices de informação para os argumentos dos interessados; (2) vértice de regras de inferência para o encadeamento de argumentos; (3) vértices para regras de conflito visando representar contra-argumentos, e (4) vértices para regras de preferência para representar a escolha dos interessados entre argumentos conflitantes. Além disso, o *framework* ACE oferece algoritmos para identificar impasses ou consensos e uma nova abordagem para a validação – a validação **relativa**. Na Inteligência Artificial, existe uma iniciativa para se definir o núcleo de uma ontologia para argumentação, o *Argument Interchange Format* (AIF) (Chesñevar et al. 2006). O *framework* ACE foi influenciado por essa iniciativa. Para atender aos nossos objetivos, nenhum outro *framework* de argumentação baseado no AIF pode ser comparado ao ACE, especialmente em relação aos algoritmos de análise.

5.7. Considerações Finais

Nesse capítulo, apresentamos nossos esforços no intuito de lidar com a questão da transparência no contexto de software. A transparência é um conceito abstrato, cuja interpretação depende das expectativas dos interessados e de suas próprias interpretações – suas noções em relação ao que transparência representa.

Baseados nesses desafios, a dimensão do conceito de transparência e a necessidade de se considerar pontos de vista diferentes, integramos propostas anteriores para a modelagem e análise de requisitos (Yu 1997; Chung et al. 2000) e validação de requisitos através de argumentação (Jureta et al. 2009). Estendemos trabalhos anteriores (Leite e Cappelli 2010; Aló 2009), oferecendo uma forma prática de se aplicar o CTS – usando construção com reutilização – e de se ampliar o catálogo através da captura de novos padrões de requisitos (Serrano e Leite 2011a). Esse capítulo também mostra como emaranhar modelos i^* , SIGs do

NFR Framework SIGs e os argumentos dos interessados para elicitar requisitos de transparência considerando múltiplos pontos de vista.

Focamos nossa pesquisa na validação relativa dos requisitos de transparência. Primeiramente, representamos os modelos i^* e os SIGs do NFR Framework na linguagem ACE, a qual é proposta em (Jureta et al. 2009) para lidar melhor com acordos entre múltiplas partes na validação de requisitos. Considerando as representações dos modelos i^* e do SIG de Transparência, validamos relativamente os requisitos de transparência através da análise da rede de metas flexíveis associadas à questão da transparência. Assim, o conhecimento contido no Catálogo de Transparência de Software pôde ser reutilizado. Além disso, a validação relativa dos requisitos de transparência foi realizada considerando várias metas flexíveis e aplicando a mesma técnica, uma abordagem baseada em argumentação. Capturamos os argumentos dos interessados sobre os requisitos de transparência considerando diferentes pontos de vista na validação desses requisitos. Definimos ainda como a condição de aceitabilidade é determinada, visando estabelecer um ponto de parada na discussão.

No intuito de demonstrar os benefícios da abordagem proposta, aplicamos a mesma em dois cenários de uso. Também aplicamos a abordagem em um estudo de caso, a aplicação *Web Lattes-Scholar* (mais detalhes no Capítulo 7).