

Uso de Sinônimos na Identificação de Atributos de Transparência

Joás Weslei Baía, José Luis Braga

Universidade Federal de Viçosa
Campus UFV, 36570-000 Viçosa, MG, Brasil.
joas.baia@gmail.com; zeluis@dpi.ufv.br;

Resumo. A transparência é um princípio democrático que permite aos cidadãos buscarem informações sobre fatos e processos. Nesse contexto, ocorre o aumento da demanda por transparência nas relações humanas. A transparência de *software* é um requisito não funcional que engenheiros de *software* precisarão demonstrar à medida que a sociedade exigir transparência em suas relações, pois essas relações são automatizadas pelos sistemas de *software*. Diante da importância de demonstrar a transparência nos processos de informação, propõe-se nesse trabalho verificar a presença de atributos de transparência em modelos de requisitos de *software* representados com o *Framework* iStar, utilizando os termos que representam os requisitos e também seus sinônimos mais conhecidos. O modelo de requisitos é analisado utilizando uma base de dados contendo sinônimos dos atributos de transparência, em adição aos conceitos do iStar para identificar esses atributos.

Palavras-chave: Framework iStar, transparência de *software*, verificação de modelos de requisitos.

1 Introdução

A transparência é um princípio democrático onde cidadãos desejam obter informações sobre fatos e processos. As organizações necessitam obter informações sobre o panorama mundial. Os investidores precisam de informações sobre oportunidades e riscos de novos mercados para decidirem sobre oportunidades de investimento. As organizações de proteção à saúde necessitam de fontes de informações sobre epidemias, e outras questões relativas ao bem estar dos seres humanos. Organismos internacionais necessitam de informações válidas sobre as intenções e estratégias políticas entre os países. Nesse contexto, ocorre o aumento da demanda por transparência nas relações humanas [1].

A transparência de *software* é um requisito não funcional que engenheiros de *software* precisarão demonstrar cada vez mais à medida que a sociedade exigir transparência nas relações com seus representantes, nas relações comerciais, sociais, enfim, nas relações humanas, pois essas relações são automatizadas pelos sistemas de

software [2]. Considerando os processos organizacionais a transparência é conceituada como a existência de processos que permitem aos indivíduos obterem informações sobre a organização através do acesso, uso, apresentação, entendimento e auditabilidade [3].

Diante da importância de demonstrar a transparência nos processos de informação, propõe-se nesse trabalho verificar a presença de requisitos de transparência a partir de modelos de requisitos de software representados com o Framework iStar [4]. Os requisitos são analisados através de uma base de sinônimos de atributos de transparência, que serão indicativos indiretos da presença dos atributos.

A Figura 1 apresenta o modelo de razões estratégicas do cenário *adicionar foto* em um dispositivo móvel, adaptado de [11]. O usuário deseja manipular mídias, nesse sentido ao manipular uma foto ele informa o local dessa foto, atribui seu nome e indica o álbum ao qual ela pertencerá. O sistema do dispositivo é o ator responsável pelo gerenciamento das mídias manipuladas pelo usuário. Para alcançar esse objetivo o sistema inicializa o aplicativo, disponibiliza a lista de álbuns, salva e armazena as fotos. Nessa especificação foi acrescentado ao modelo original o requisito *Ease* para ilustrar o uso de sinônimos na identificação de atributos de transparência. A tarefa *Save Automatically* contribui parcialmente para que o objetivo *Ease*, facilidade para manipular mídias, seja alcançado. Na seção 4 discutiremos o papel de sinônimos na identificação de atributos de transparência.

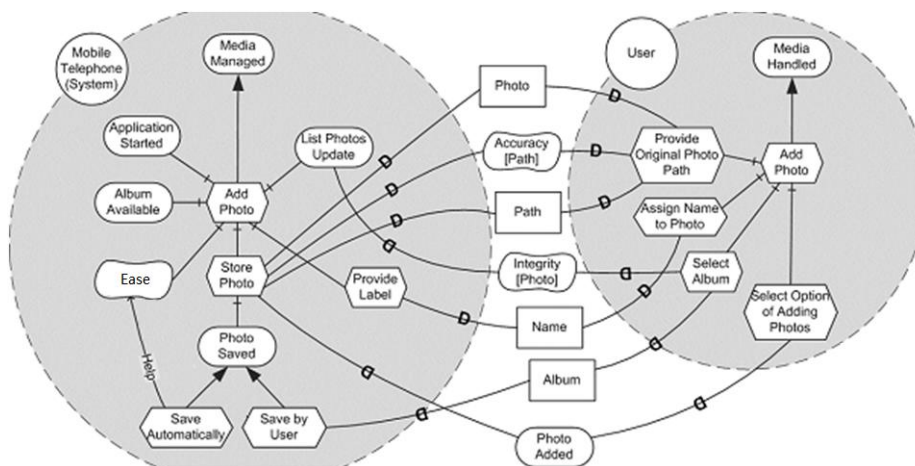


Fig. 1. Modelo de Razões Estratégicas da especificação de requisito (Mobile Telephone System). Adaptado de [11].

Nesse texto será apresentada a abordagem para analisar modelos de requisitos sob o ponto de vista da transparência. A estratégia apresentada para verificar a presença de atributos de transparência em modelos de requisitos apresentada nesse trabalho fornece:

- Uma base de dados de sinônimos dos atributos de transparência.
- Um mecanismo para transformar o modelo de requisitos representado com o Framework iStar em uma base de fatos (base de dados).

- Uma base de conhecimento para análise desses modelos de requisitos composta por regras de produção.
- A separação entre o mecanismo de controle (base de conhecimento) e a descrição do problema (especificação de requisitos) permite um sistema flexível e expansível para realizar a verificação de requisitos.

Este trabalho é um desdobramento do artigo [10], apresentado no WER2012 em Buenos Aires. A estrutura da solução baseada em Sistemas de Conhecimento está descrita naquele artigo, e será apenas referenciada neste artigo.

Esse texto está organizado em seis seções. Na segunda seção é apresentada a relação entre os atributos de transparência e as características do Framework iStar. A terceira seção apresenta a base de dados contendo os sinônimos dos atributos de transparência. A quarta seção discute a estratégia escolhida para a verificação da presença de atributos de transparência em especificações iStar. Na quinta seção discutimos os trabalhos correlacionados. Na sexta seção finalizamos com as conclusões.

2 A Transparência de Software e sua Relação com o Framework iStar

A transparência de *software* é um requisito não funcional, pois é caracterizada por atributos de qualidade, podendo ser julgada de forma diferente de acordo com a visão do indivíduo, dessa forma ela é considerada uma meta-flexível, ou um *softgoal*, ou seja, o *software* pode ser considerado transparente do ponto de vista de uma pessoa, enquanto pode não ser transparente ou ainda menos transparente para outra [5]. O *software* transparente deve possuir os atributos de *acessibilidade*, *usabilidade*, *informatividade*, *entendibilidade* e *auditabilidade*, pois nos processos organizacionais transparentes esses atributos devem estar presentes, e considerando que os sistemas de *software* automatizam tais processos, eles devem incorporá-los [6][7].

A Tabela 1 apresenta os atributos de transparência relacionados com as características do Framework iStar. Esse relacionamento entre os atributos que conceituam a transparência e o iStar é utilizado na implementação de regras de produção para verificar se uma especificação de requisitos é aderente aos conceitos da Transparência. A coluna Relação expressa a aderência das características do iStar identificados pelos algarismos de 1 a 9 na coluna ID, aos atributos de transparência. *Verificabilidade*, por exemplo, está relacionada com as características 1, 5.2, e 7 do Framework iStar.

Analisando a Tabela 1, observa-se que existem alguns atributos de Transparência que não têm relação com o Framework iStar. Por exemplo, o atributo *publicidade* não está relacionado a nenhuma característica do iStar. O mesmo ocorre com os atributos *portabilidade*, *uniformidade*, *simplicidade*, e outros nove atributos listados na primeira coluna da Tabela 1.

A partir dessa lacuna entre alguns atributos de Transparência e as características do Framework iStar foi adota a estratégia de utilizar sinônimos desses atributos para sinalizar sua presença em especificações de requisitos. Na próxima seção é apresentada essa lista de sinônimos.

Table 1. Relação entre as características do iStar e os atributos de transparência [6] [7].

Atributos de Transparência	Relação	ID	Característica iStar
Portabilidade	Ausente	1	Intencionalidade (<i>actor</i> e <i>goal</i>)
Disponibilidade	6	2	<i>Softgoal</i>
Publicidade	Ausente	3	Intencionalidade detalhada (decomposição de tarefas)
Uniformidade	Ausente	4	Alternativas de operacionalização (conectores <i>means-end</i>)
Simplicidade	Ausente		
Operabilidade	5.1	5	Dependência estratégica
Intuitividade	Ausente	5.1	Dependência de recurso
Desempenho	8	5.2	Dependência de meta
Adaptabilidade	Ausente	5.3	Dependência de meta flexível
Amigabilidade	Ausente	6	Classificação de atores (Associação entre atores INS e ISA)
Clareza	2, 5.2, 9	7	Estrutura organizacional (posições)
Completeza	2	8	Responsabilidades (papeis)
Corretude	Ausente	9	Contribuição positiva/negativas (<i>link contribution</i>)
Atualidade	Ausente		
Comparabilidade	Ausente		
Consistência	Ausente		
Integridade	4		
Acurácia	2		
Concisão	Ausente		
Compositividade	3, 5.3		
Divisibilidade	3, 8		
Detalhamento	4		
Dependência	5, 9		
Validável	4		
Controlabilidade	Ausente		
Verificabilidade	1, 5.2, 7		
Rastreabilidade	1, 5.2, 9		
Explicável	Ausente		

3 Sinônimos dos Atributos de Transparência

A partir da lista de atributos de transparência apresentados na Tabela 1 é possível observar que existem atributos que não possuem relação com os conceitos do iStar, inviabilizando sua identificação pelas regras de produção que exploram esse relacionamento. A partir desses atributos de transparência foram identificados seus sinônimos através da consulta à base de dados Wordnet [8]. A Tabela 2 contém a lista desses sinônimos e a partir dela foi criada a base de fatos contendo os atributos de transparência e seus respectivos sinônimos, exibida na Figura 2.

Table 2. Sinônimos dos atributos de Transparência, extraído da base Wordnet [8].

Atributos	Atributos	Sinônimo
Portável	portable	portable, portability
Disponível	available	availability, usable, useable, uncommitted
Publicidade	disclosure	revelation, revealing
Uniforme	uniform	consistent, undifferentiated, unvarying
Simples	simplicity	ease, easiness, simpleness
Operável	operable	operable, practicable, functional, usable, useable, operability, in working order
Intuitivo	intuitive	intuitive , nonrational, visceral
Desempenho	performance	execution, operation, functioning, carrying out, carrying into action
Adaptável	adaptable	adaptability, adaptable
Amigável	friendly	friendliness, friendly
Clareza	clarity	lucidity, lucidness, pellucidity, clearness, limpidity
Completa	complete	fill out, fill in, make out, completeness, full
Correto	correct	right, correctness, rightness
Atualizada	updated	update
Comparável	comparable	corresponding, like, comparison, compare, equivalence, comparability
Consistência	consistency	consistence
Integridade	integrity	unity, wholeness
Acurácia	accuracy	truth
Concisão	conciseness	concision, pithiness, succinctness
Compositividade	compound	compound
Divisível	divisible	divisibility
Detalhamento	detail	item, point, particular
Dependência	dependence	dependance, dependency
Validável	authenticate	authenticate
Controlável	controllable	controllable, controllability, governable
Verificável	verifiable	confirmable

Rastreável	traceable	trackable, traced
Explicável	explainable	understandable

A Figura 2 mostra parcialmente a base de dados de sinônimos dos atributos de transparência representada na sintaxe da linguagem CLIPS *C Language Integrated Production System* [9]. Através dessa base de dados, o mecanismo de inferência do sistema pode aplicar a regra de produção *ruleDenoteSynonym*, também exibida na Figura 2, para sinalizar uma possível ocorrência de sinônimos dos atributos de transparência no modelo de requisitos. Por exemplo, quando for citada a palavra *revelation* na especificação, a regra de produção será disparada informando que há o relacionamento com o atributo de transparência *disclosure*.

```
(deffacts words
  (word (id 1) (word "portable") )
  (word (id 2) (word "available") )
  (word (id 3) (word "disclosure") )
  (word (id 4) (word "uniform") )
  (word (id 5) (word "simplicity") )
  (word (id 6) (word "operable") )
  (word (id 7) (word "intuitive") )
  (word (id 8) (word "performance") )
  (word (id 23) (word "dependence") )
  (word (id 24) (word "authenticate") )
  (word (id 25) (word "controllable") )
  (word (id 26) (word "verifiable") )
  (word (id 27) (word "traceable") )
  (word (id 28) (word "explainable") )

  (word (id 30) (word "portability") (idAttribute 1))
  (word (id 31) (word "availability") (idAttribute 2))
  (word (id 32) (word "usable") (idAttribute 2))
  (word (id 33) (word "useable") (idAttribute 2))

)

(defrule ruleDenoteSynonym
  (element (id ?id) (name ?name) (type ?type) ); o elemento
  (word (id ?idAttribute) (word ?nameAttribute) ); o atributo
  (word (id ?idSim) (word ?name) (idAttribute ?idAttribute)); o sinônimo
  =>
  (printout t ?name " denote " ?nameAttribute crlf)
)

```

Fig. 2. Parte da base de dados composta de sinônimos.

Conforme mostra a Figura 2, a palavra *portability* está relacionada com o atributo *portable*. Quando a regra de produção *ruleDenoteSynonym* for aplicada ao modelo de requisitos ela sinalizará essa ocorrência. O funcionamento dessa regra de produção consiste na utilização de três informações na base de dados. Para cada elemento do modelo de requisitos é verificada sua relação com os sinônimos dos atributos, e a partir dessa relação é recuperada a relação do sinônimo com o atributo de transparência. Se essa condição for satisfeita será sinalizado a ocorrência de um sinônimo de atributo de transparência na especificação de requisitos.

A abordagem utilizando a base de sinônimos dos atributos de transparência é uma extensão do trabalho apresentado em [10]. Conforme relatado na Tabela 1, há

atributos de transparência que não possuem relação com os conceitos do iStar, inviabilizando a identificação desses atributos.

4 Verificação de Atributos de Transparência em Especificações de Requisitos Representadas em iStar

A estratégia para verificar a aderência de uma especificação de requisitos representada com o *Framework* iStar aos atributos de transparência é realizada seguindo os seguintes passos:

1. Representar a especificação de requisitos no formato de fatos CLIPS [9], formando assim a base de dados do sistema de verificação.
2. Inspeccionar essa base de dados com regras de produção que implementam a *relação* entre os conceitos do iStar e os atributos de transparência, conforme exibido na Tabela 1.
3. Inspeccionar essa base de dados com a regra de produção *ruleDenoteSynonym* para identificar a ocorrência de sinônimos dos atributos de transparência nessa especificação de requisitos.

No primeiro passo, a especificação de requisitos construída com o Framework iStar é representada no formato de fatos, formando assim a base de dados do sistema de verificação. No trabalho [10] foram apresentadas duas abordagens para formar a base de dados. Por exemplo, a especificação de requisitos apresentada na Figura 1, os elementos *User* e *Media handled* são representados nos seguintes fatos:

```
(element ( id "01" ) ( name "User" ) ( type "actor" ) )  
(element ( id "02" ) ( name "Media handled" ) ( type "goal" ) ( idActor "01" ) )
```

A partir da especificação de requisitos descrita na base de dados, é possível verificar sua aderência aos atributos de transparência utilizando a base de conhecimento apresentada em [10]. Essa base de conhecimento é formada pela relação entre as características do iStar com os atributos de transparência apresentada na Tabela 1. Por exemplo, a característica do iStar *intencionalidade* está relacionada com os atributos de transparência *rastreabilidade* e *verificabilidade*, portando na base de conhecimento há uma regra que indica essa ocorrência. Na especificação da Figura 1, o elemento *Media handled* é uma meta do ator *User*, essa relação indica a intencionalidade desse ator, portanto será sinalizada a presença dos atributos *rastreabilidade* e *verificabilidade* nesse parte da especificação.

Uma vez realizada a segunda etapa, a base de dados é verificada à procura de sinônimos dos atributos de transparência. Nesse sentido a regra *ruleDenoteSynonym* verifica a ocorrência desses sinônimos na especificação de requisitos. Por exemplo, na especificação de requisitos exibida na Figura 1 o elemento *Ease* é um sinônimo do atributo *simplicity*, portanto essa regra de produção sinalizará a ocorrência desse atributo nessa especificação, "*Ease denote simplicity*".

5 Trabalhos correlacionados

No trabalho [10] foi apresentada a abordagem para verificar a aderência de uma especificação de requisitos aos atributos de transparências a partir do relacionamento entre as características do iStar e os atributos de transparência. Nesse trabalho, [10], foram discutidas duas abordagens para formar a base de dados a partir de uma especificação de requisitos em iStar, além de apresentar a base de conhecimento que implementa a relação entre os atributos de transparência e as características do Framework iStar. Porém, a partir da análise da Tabela 1, observa-se que existem atributos de transparência que não têm relação com as características do iStar, conseqüentemente a base de conhecimento apresentada nesse trabalho não pode identificar sua presença. A partir dessa necessidade foi apresentada no presente texto uma base de dados composta por sinônimos dos atributos de transparência.

No trabalho desenvolvido em [12] foi apresentada uma abordagem para tornar o *software* mais transparente, onde foi discutida uma forma de documentação de sistemas de *software* fundamentada em aspectos técnicos e de uso. As informações técnicas são direcionadas aos engenheiros de *software*, tais como: arquitetura e linguagem de programação. As informações de uso são destinadas aos usuários finais do sistema, por exemplo, visão geral do *software*, suas funcionalidades e também os requisitos que não foram implementados. Essa abordagem recebeu o nome de *BuS* (Bula de Software) inspirada na bula de remédios que agrega informações sobre medicamentos.

Para organizar as informações sobre o *software* foi utilizada a tecnologia XML (*eXtensible Markup Language*). As informações técnicas e de uso são estruturadas em tópicos através da utilização de *tags*. A *tag* <CABECALHOSOFTWARE/> descreve a seção do documento contendo o nome do *software*, sua composição e seus objetivos. Na *tag* <INFORMACAOGERALCIDADA/> ficam estruturadas as informações sobre o uso do sistema e sobre o seu escopo. Por fim, na *tag* <INFORMACAOTECNICA/> são disponibilizadas informações sobre a tecnologia envolvida no desenvolvimento do *software* e que tem como público alvo os engenheiros de *software*.

A documentação de *software BuS* visa potencializar a transparência contribuindo para que o *software* seja aderente aos atributos de transparência *uniformidade*, *amigabilidade*, *simplicidade*, *intuitividade*, *operabilidade* e *clareza*. A abordagem apresentada nesse texto procura indícios da presença de atributos de transparência em especificações de requisitos iStar. As duas estratégias possuem objetivos semelhantes, ambas buscam tornar o *software* mais transparente.

O trabalho desenvolvido por [13] aborda a verificação de requisitos a partir de processamento de linguagem natural com o objetivo de identificar ambiguidades em especificações de requisitos. Os requisitos representados em linguagem natural são submetidos ao procedimento de medição de ambiguidades, onde as sentenças potencialmente ambíguas são verificadas. Além disso, é identificado o motivo que tornaria a sentença ambígua auxiliando o trabalho do engenheiro de *software* na remoção de tais ambiguidades. A abordagem de verificação de atributos de transparência desenvolvida nesse trabalho não utiliza processamento de linguagem

natural, porém procura identificar sinônimos desses atributos de transparência nos modelos de requisitos.

No trabalho de [14] a metodologia Tropos é empregada para verificar os aspectos dinâmicos de dependência entre atores estratégicos. O modelo de especificação de requisitos iStar é representado na linguagem *Formal Tropos* onde atores, metas, tarefas, recursos e dependências são declarados juntamente com as restrições usadas para verificar o modelo. O método de verificação de atributos de transparência difere dessa abordagem, pois o modelo a ser verificado é organizado em uma base de dados enquanto o conhecimento usado para a verificação é implementado nas regras de produção. Há dois módulos distintos para realizar a verificação: a base de fatos e a base de regras. A base de fatos é composta pela especificação de requisitos, enquanto a base de regras é formada pelas regras de produção que são aplicadas à base de fatos para inferir a presença ou ausência dos atributos de transparência.

A estratégia utilizada no trabalho de [15] emprega a tecnologia XML/XSLT para verificar especificações de requisitos sob os aspectos de ambiguidade, completude e rastreabilidade. A verificação dos requisitos é realizada através da aplicação de folhas de estilos sobre o modelo de requisitos representado em XML. Enquanto a abordagem aqui apresentada utiliza regras de produção para verificar se na especificação de requisitos representada com o Framework iStar há indícios de atributos de transparência.

A diferença entre a estratégia utilizada nesse trabalho e as abordagens utilizadas por [13], [14] e [15] está no objetivo da verificação. Ela busca identificar atributos de transparência em especificações de requisitos, enquanto as demais procuram verificar se essas especificações estão em conformidade com as necessidades e interesses dos envolvidos no ambiente organizacional onde o sistema irá operar. A transparência de *software* será mais uma entre as expectativas dos *stakeholders* à medida que ela for exigida pelos diversos segmentos da sociedade. Além disso o uso de uma base de sinônimos dos atributos de transparência procura expandir a abordagem empregada em [10].

6 Conclusões

A abordagem baseada em sistema de inferência permitiu a obtenção de um sistema flexível e expansível, que proporciona seu uso para verificar especificações de requisitos e para sugerir a inclusão de novos atributos em especificações analisadas pelo sistema. Essa estratégia permite também o uso do sistema como uma plataforma de ensino e disseminação das ideias de transparência e sua inclusão em sistemas de informação organizacionais. A representação do modelo de requisitos iStar em uma base de dados permite que outros tipos de verificação sejam realizadas. Por exemplo, a verificação sintática do modelo segundo as regras do Framework iStar e pode também ser integrada às ferramentas de modelagem iStar.

A separação entre a base de dados contendo a especificação de requisitos e as regras de produção proporcionou a obtenção de um sistema de verificação composto por módulos independentes. Novas regras de produção podem ser adicionadas à medida

que outras relações entre os atributos de transparência e o iStar forem propostas. A base de conhecimento desenvolvida pode ser utilizada por um agente inteligente baseado em conhecimento.

O sinônimos dos atributos de transparência utilizados para indicar a presença de palavras na especificação de requisitos que indicam os atributos de transparência é uma proposta inicial que pode ser melhorada a partir de sua utilização em modelos de requisitos disponíveis na literatura.

Referências Bibliográficas

1. HOLZNER B.; HOLZNER L. **Transparency in Global Change: The Value of Open society**. 1. ed. Pittsburgh, PA, EUA: Universidad of Pittsburgh Press, 2006.
2. CAPPELLI, C.; OLIVEIRA, A. PADUA; LEITE, J. C. S. P. **Exploring Business Process Transparency Concepts**, RE 207, IEEE Computer Society Press, pp. 389-390, 207.
3. CAPPELLI, C.; LEITE, J. C. S. P. **Transparência de Processos Organizacionais**. In: Simpósio Internacional de Transparência nos Negócios, 2008, Rio de Janeiro, Brasil. **Anais...** Rio de Janeiro: Universidade Federal Fluminense, LATEC, 2008. Disponível em: <<http://www.latec.uff.br/transparencia/anais.pt-br.php>>. Acesso em: 25 jan. 2011.
4. YU, Eric. **Modelling Strategic Relationships for Process Reengineering**. 1995. 124 f. Tese de Doutorado – Dept. of Computer Science, University of Toronto, Toronto, 1995.
5. LEAL, A. L. C.; SOUSA, H. P.; LEITE, J. C. S. P.; BRAGA, J.L. **Transparência aplicada a modelo de negócios**. In: XIV WER - Workshop em Engenharia de Requisitos (XIV CibSE - Congresso Ibero-Americano em Engenharia de Software), 2011, Rio de Janeiro. **Proceedings XIV CibSE**. Rio de Janeiro, 2011. v. XIV. p. 321-332. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/papers_by_conference.lp?conference=WER11>. Acesso em: 20 jan. 2012.
6. LEITE, J.C.S.P, CAPPELLI, C., Exploring i* Characteristics that Support Software Transparency. In Proceedings of the 3rd International i* Workshop, CEUR Workshop Proceedings, Vol. 322, 208, pp. 51-54. Disponível em: <<http://CEUR-WS.org/Vol-322/>>. Acesso em 24 set. 2011.
7. OLIVEIRA, A. P. A.; CAPPELLI, C.; CUNHA, H. S.; LEITE, J. C. P.; WERNECK, V. M. B. **Engenharia de software Intencional: Tornando o Software Mais Transparente**. In: Simpósio Brasileiro de Engenharia de Software, 2007, Rio de Janeiro. Tutorial apresentado no XXI Simpósio Brasileiro de Engenharia de Software, Rio de Janeiro, 2007. Disponível em: <<http://www.sbbd-sbes2007.ufpb.br/tuto3.pdf>>. Acesso em: 22 jan. 2011.
8. WORDNET: A lexical database for English. Disponível em: <<http://wordnet.princeton.edu/wordnet/>>. Acesso em: 02 maio 2012.
9. GIARRATANO, J. C. **CLIPS User's Guide**. Version 6.0. NASA Lyndon B. Johnson Space Center, Software Technology Branch, Houston, Texas, EUA, 1993.
10. BAIA, J. W.; BRAGA, J. L.; CARVALHO, L. F. **Verificação de Requisitos de Transparência em Modelos iStar**. In: XV WER - Workshop em Engenharia de Requisitos (XIV CibSE - Congresso Ibero-Americano em Engenharia de Software), 2012, Buenos Aires . **Proceedings XV CibSE**. Buenos Aires, 2012. v. XV. Disponível em:

<http://wer.inf.puc-rio.br/WERpapers/papers_by_conference.lp?conference=WER12>.

Acesso em: 20 jul. 2012.

11. SILVA, C.; BORBA, C.; CASTRO, J. **A Goal Oriented Approach to Identify and Configure Feature Models for Software Product Lines**. In: Workshop em Engenharia de Requisitos – WER11, (XIV CibSE - Congresso Ibero-Americano em Engenharia de Software), 2011, Rio de Janeiro. **Proceedings** XIV CibSE. Rio de Janeiro, Brasil, 2011. v. XIV. p. 321-332. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/papers_by_conference.lp?conference=WER11>. Acesso em: 20 jan. 2012.
12. LEAL, A. L. de C.; ALMENTERO, Eduardo; CUNHA, Herbert; SOUSA, H. P.; LEITE, J. C. S. P. **Bula de Software: Uma Estrutura Definida para Promover a Melhoria da Transparência em Software**. In: XV Workshop on Requirements Engineering - WER2012, 2012, Buenos Aires, Argentina. **Anais ...** Buenos Aires, Argentina: UNLaM - Universidad Nacional de La Matanza, 2012. v. 15. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/papers_by_conference.lp?conference=WER12>. Acesso em 11 jun. 2012.
13. KIYAVITSKAYA, Nadzeya; ZENI, Nicola; MICH, Luisa; BERRY, Daniel M. **Requirements for Tools for Ambiguity Identification and Measurement in Natural Language Requirements Specifications**. In: Workshop em Engenharia de Requisitos - WER07, 2007, Toronto. **Proceedings ...** Toronto: York University, 2007. p. 197 - 206. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/papers_by_conference.lp?conference=WER07>. Acesso em: 23 dez. 2011.
14. FUXMAN, A.; MYLOPOULOS, J.; PISTORE, M.; TRAVERSO, P. **Model Checking Early Requirements Specifications in Tropos**. In: RE-2001, 9th IEEE International Requirements Engineering Conference, 2001, Toronto, Canada. **Proceedings** on Fifth IEEE International Symposium. Toronto, Canada:[s.n.], 2001.
15. DURAN, A.; BERNARDEZ, B.; RUIZ, A.; TORO, M. **An XML-based approach for the automatic verification of software requirements specifications**. In: Fourth Workshop on Requirements Engineering, 2001, Buenos Aires, Argentina. **Proceedings ...** Buenos Aires: Universidad Tecnológica Nacional, 2001. p. 181-194. Disponível em: <<http://www.inf.puc-rio.br/wer01/>>. Acesso em 23 dez. 2011.