Capturing Transparency-Related Requirements Patterns through Argumentation

Maurício Serrano, Julio Cesar Sampaio do Prado Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brasil
mauserr@les.inf.puc-rio.br, www.inf.puc-rio.br/~julio

Abstract—**Transparency is a critical concern to democratic societies. As software permeates our social lives, Software Transparency is becoming a quality *criterion* that demands more attention from software developers. We present in this paper our approach to capture transparency-related requirements patterns through argumentation. We represented initial transparency knowledge as requirements patterns. These initial patterns stimulated stakeholders' arguments on an open discussion about transparency of a given software. We apply an argumentation framework to capture the stakeholders' arguments on argumentation graphs. Transparency-related requirements patterns are identified by formally analyzing consensus on the argumentation graph. Basically, the identified patterns are possible operationalizations to transparency-related softgoals.**

Keywords-capturing requirements patterns; argumentation; software transparency; transparency patterns; multi-party agreement

I. INTRODUCTION

Transparency is a critical concern to democratic societies. According to Holzner&Holzner [1], transparency consists of "*the social value of open, public and/or individual access to information held and disclosed by centers of authority*." We can understand that transparency is a key principle to establish trust between citizens and their governments. As software permeates our social lives, Software Transparency is becoming a quality *criterion* that demands more attention from software developers.

In the Software Engineering literature, few work address Software Transparency [2] [3] [4]. Cappelli and Leite [4] try to understand and define Software Transparency as a Non-Functional Requirement (NFR) by identifying its relation to other well-known NFRs. According to them, Software Transparency is helped by the Accessibility, Usability, Informativeness, Understandability and Auditability NFRs. They propose a Software Transparency Softgoal Interdependency Graph (SIG), following the NFR Framework [5].

The Software Transparency Group [6] initial approach was to extend the Transparency SIG with several operationalizations for each transparency-related softgoal. This approach proved to be counter-productive, as it would only offer a rigid set of operationalizations. We preferred to extend the Transparency SIG with a set of questions for each transparency-related softgoal. These questions were the result of an adaptation of the Goal-Question-Metric (GQM) [7] to a Goal-Question-Operationalization (GQO) method [8]. The questions should help the requirements engineer to identify good practices that operationalize the softgoals (NFRs).

It was necessary to provide a flexible mechanism to represent the initial transparency knowledge and the identified operationalizations. The Software Transparency Catalog (STC) [9] was proposed following an adaptation of Supakkul et al.'s [10] requirements patterns approach. The Transparency SIG and the GQO questions were represented as patterns in the catalog, thus providing an initial set of transparency knowledge. This initial knowledge supports the process of identifying good practices that operationalize the transparency-related NFRs. The identified good practices are then represented as requirements patterns and inserted into the catalog.

We are capturing the operationalizations requirements patterns on a series of weekly group meetings. These meetings focuses [11] on discussing the transparency of Lattes-Scholar [12], a Web application (software) that uses two Web-Services (Google Scholar and Lattes [13]) to sort the publications of a researcher by the number of citations. On these meetings, a mediator uses the transparency knowledge to support the stakeholders' discussion. The aim of the discussion is to let the stakeholders argue and counter-argue about possible operationalizations for the transparency-related NFRs that could answer the GQO questions.

The stakeholders' arguments are represented on an Argumentation Framework [14]. The ACE Framework [15] has a language to model argumentation graphs, were the arguments, conflicts, preferences and inferences between arguments can be explicitly represented. Moreover, ACE algorithms formally identify impasses or consensus. When there is a consensus about some operationalization to a transparency-related NFR, this operationalization is represented as a requirement pattern and is inserted into the catalog. All the group's meetings are recorded on video to facilitate the arguments transcription to ACE graphs.

We present in this paper our approach to capture requirements patterns through argumentation by identifying, on argumentation graphs, stakeholders' consensus about NFR operationalizations. The remainder of this paper is structured as follows: Section 2 presents how we represented the initial transparency knowledge as requirements patterns in order to support the stakeholders' discussions. Our approach to capture requirements patterns through argumentation is shown on Section 3. Some considerations about the relative validation of the identified requirements patterns are presented on Section 4. Section 5 discusses

978-1-4577-1022-3/11/$26.00 (c)2011 IEEE

related work. Finally, Section 6 presents the final considerations.

II. Supporting the discussion with Requirements Patterns

We support the stakeholders' discussion with transparency knowledge, represented as requirements patterns on an adaptation of Supakkul et al.'s approach [10]. They define four kinds of patterns: Objectives, Problems, Alternatives and Selection Patterns. We represented the Transparency SIG [4] as an Objective pattern. The Transparency Objective pattern was inserted in the STC.

Figure 1 shows the Transparency SIG represented as an Objective requirements pattern. Seven refinement rules are necessary to express the Transparency SIG. The first refinement rule, R1, relates the Transparency NFR to the software resource. R2 defines the contributions between the five main transparency-related NFRs and the Transparency NFR. R3 to R7 define the contributions from twenty eight NFRs to the five main transparency-related NFRs. For example, R2 states that the Accessibility NFR helps the Transparency NFR. Moreover, R3 states that Portability, Availability and Publicity help the Accessibility NFR.

This knowledge was used to (i) acknowledge that it was important to discuss the Lattes-Scholar accessibility, as the accessibility NFR is a key NFR to transparency, and (ii) guide the stakeholders' discussion about the Lattes-Scholar accessibility, suggesting that the stakeholders should argue about the Lattes-Scholar portability, availability and publicity.

The GQO questions could not be represented by any pattern kind suggested by [10]. Supakkul et al.'s patterns follow a problem/vulnerability-solutions approach, where the Problem pattern represents the vulnerabilities and the Alternative represents the solutions. As our Software Transparency Catalog follows a good-practices approach, we propose a Questions pattern to represent the GQO questions and its groups [8]. The operationalizations are represented on an adaptation of Supakkul et al.'s Alternative patterns.

Figure 2 shows the GQO questions for the Availability NFR represented as a Question requirements pattern. The GroupsIdentification refinement rule, R1, defines three questions groups for the Availability NFR. Each QuestionsIdentification refinement rule, R2, R3 and R4, lists the good practices questions pertinent to each group. For example, R1 states that "Apply security techniques to ensure uninterruptible information availability" is a question group of the Availability NFR. R3 lists five questions that help indentifying good practices, or operationalizations, that answer to these questions.

Question patterns provide points of discussion, as the stakeholders' argue and counter argue about possible operationalizations to a transparency related-NFR - the answers. For example, the stakeholders discussed about using a university's server to host the Lattes-Scholar, trying to answer the question "Was a dedicated server used to let the information available?" (first question of the second question group to the Availability NFR).

III. Capturing Requirements Patterns

We are discussing the Lattes-Scholar [12] transparency on a series of weekly group meetings. On these meetings we openly discuss operationalizations to the transparency-related NFRs. Accessibility is a key quality criteria to Software Transparency [4], as it is impossible to achieve the other four softgoals (Usability, Informativeness, Understandability and Auditability) without access to the software. The Software Transparency SIG also shows that the NFRs Availability, Portability and Publicity help the Accessibility NFR.

Two of the weekly group meetings were dedicated to discuss the accessibility of the Lattes-Scholar. Four requirements patterns (transparency objectives pattern plus Availability, Portability and Publicity questions patterns) were retrieved from the STC and guided the group's discussion (Section 2). The stakeholders argued and counter-argued about operationalizations that could answer the good practices questions. These meetings were recorded on video to facilitate the arguments identification and offer some degree of pre-traceability. The stakeholders' arguments were captured using an argumentation framework. We used the ACE Argumentation Framework [15] to represent the stakeholders' arguments as a graph. We attached a traceability tag to the arguments nodes to trace them back to the meetings video files, thus providing backwards traceability to the ACE graphs.

Our key approach to capture Requirements Patterns is to formally identify consensus about possible NFRs operationalizations on ACE Graphs by applying ACE algorithms [15]. The agreed operationalizations were further described in NFR Framework SIGs and finally represented as Alternative patterns.

Figure 3 shows a partial ACE graph of the discussion. ACE graphs are directed labeled graphs, where all edges have the semantical meaning "to". ACE model propositions with nodes using four possible labels: i, I, C and P. Nodes labeled ITn are inference nodes, a deductive or ampliative means to obtain the outputs (ends) from the inputs. Nodes labeled i are information vertices used as inputs or outputs of inference rules. The stakeholders arguments are represented as information vertices labeled as i(arg n). Each information vertex has a traceability tag that describes the argument, identifies the stakeholder that proposed it and traces the information vertex back to the video file of the discussion. Nodes labeled C are conflict vertices, denoting conflicts between stakeholders' arguments. Finally, P nodes represent preferences between two conflicting arguments. Figure 4 shows all the discussed operationalizations to the Publicity NFR represented as a NFR Framework SIG.

On the left branch of Figure 3, the arguments i(arg11) and i(arg13) analyze the contributions of the "Publish in Scientific Publications" operationalization to the Publicity NFR and to other transparency-related NFRs. Argument i(arg13) corrects i(arg11) (conflict rule C1) and is a consensus, denoted by the preference rule P1. Figure 5 shows the Alternative requirements pattern that represents this operationalization as a possible answer to the question

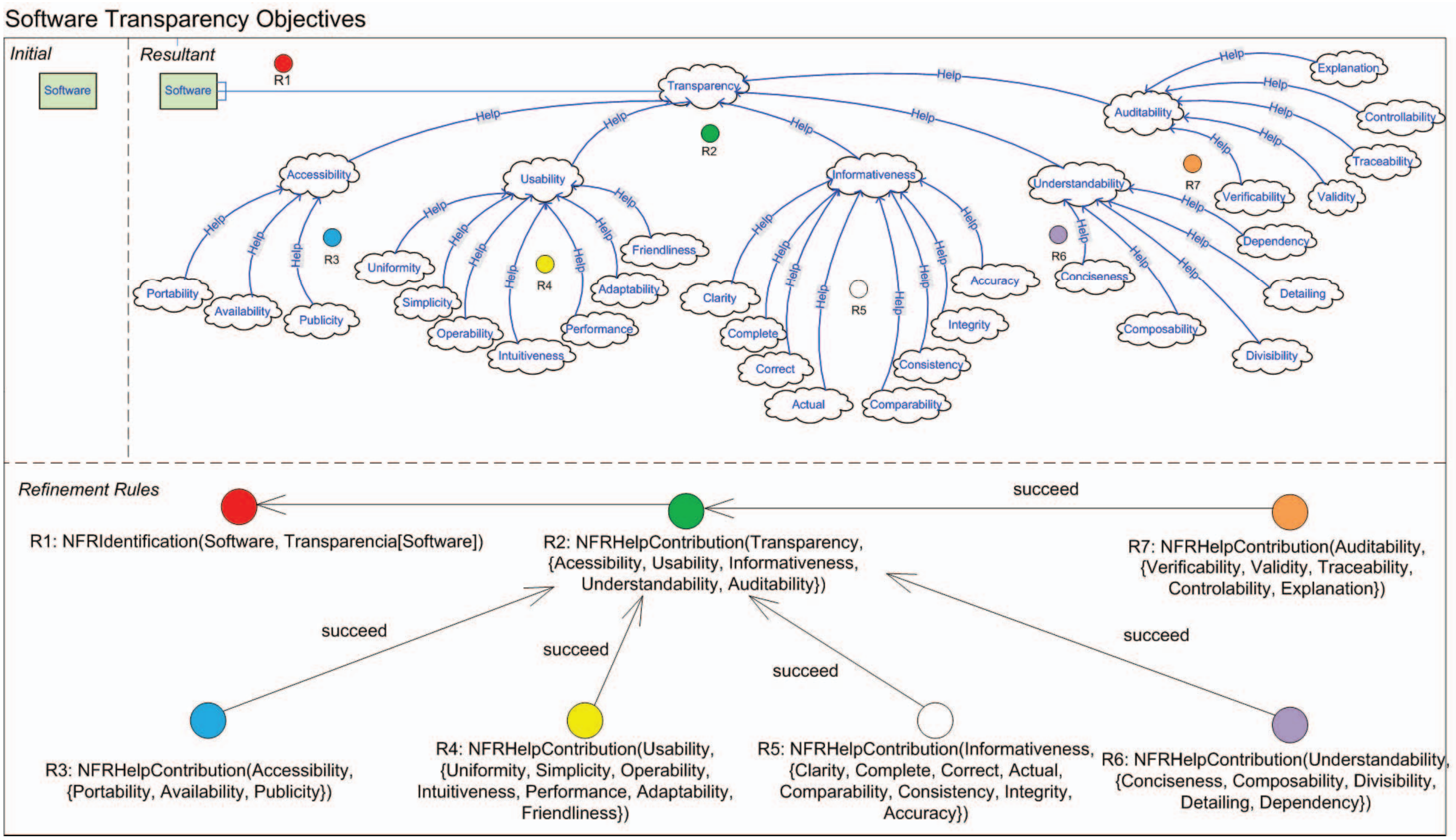
Software Transparency Objectives
Initial
Software
Resultant
Software
R1
Transparency
R2
Help
Accessibility
R3
Portability
Availability
Publicity
Usability
R4
Uniformity
Simplicity
Operability
Intuitiveness
Performance
Adaptability
Friendliness
Informativeness
R5
Clarity
Complete
Correct
Actual
Comparability
Consistency
Integrity
Accuracy
Understandability
R6
Conciseness
Composability
Divisibility
Detailing
Dependency
Auditability
R7
Verificability
Validity
Traceability
Controllability
Explanation
Refinement Rules
R1: NFRIdentification(Software, Transparencia[Software])
R2: NFRHelpContribution(Transparency, {Acessibility, Usability, Informativeness, Understandability, Auditability})
R7: NFRHelpContribution(Auditability, {Verificability, Validity, Traceability, Controlability, Explanation})
R3: NFRHelpContribution(Accessibility, {Portability, Availability, Publicity})
R4: NFRHelpContribution(Usability, {Uniformity, Simplicity, Operability, Intuitiveness, Performance, Adaptability, Friendliness})
R5: NFRHelpContribution(Informativeness, {Clarity, Complete, Correct, Actual, Comparability, Consistency, Integrity, Accuracy})
R6: NFRHelpContribution(Understandability, {Conciseness, Composability, Divisibility, Detailing, Dependency})
succeed

Figure 1. Software Transparency Objectives requirements pattern. Extracted from [8].

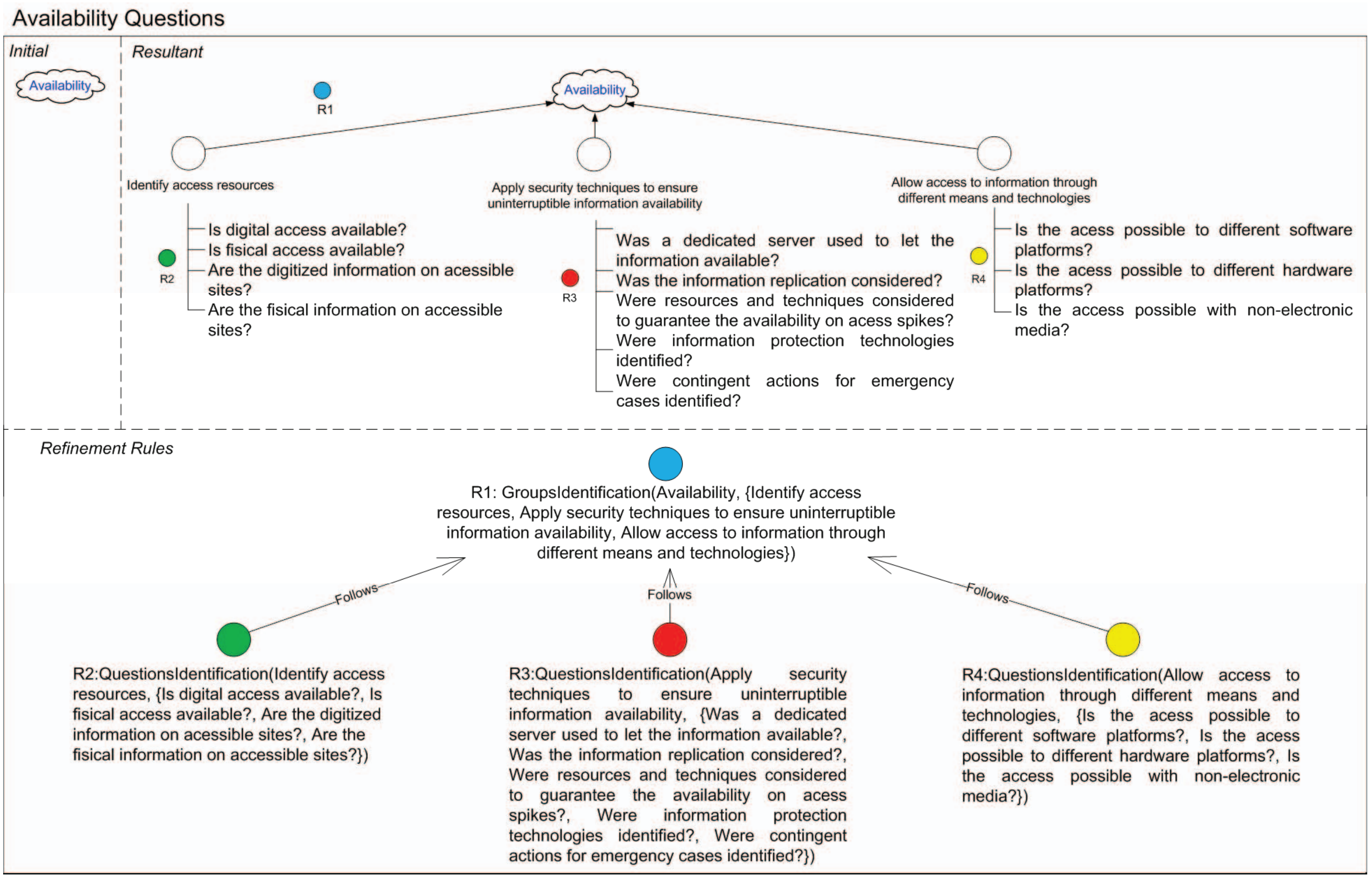
Availability Questions
Initial
Availability
Resultant
R1
Availability
Identify access resources
Apply security techniques to ensure uninterruptible information availability
Allow access to information through different means and technologies
R2
Is digital access available?
Is fisical access available?
Are the digitized information on acessible sites?
Are the fisical information on accessible sites?
R3
Was a dedicated server used to let the information available?
Was the information replication considered?
Were resources and techniques considered to guarantee the availability on acess spikes?
Were information protection technologies identified?
Were contingent actions for emergency cases identified?
R4
Is the acess possible to different software platforms?
Is the acess possible to different hardware platforms?
Is the access possible with non-electronic media?
Refinement Rules
R1: GroupsIdentification(Availability, {Identify access resources, Apply security techniques to ensure uninterruptible information availability, Allow access to information through different means and technologies})
Follows
Follows
Follows
R2:QuestionsIdentification(Identify access resources, {Is digital access available?, Is fisical access available?, Are the digitized information on acessible sites?, Are the fisical information on accessible sites?})
R3:QuestionsIdentification(Apply security techniques to ensure uninterruptible information availability, {Was a dedicated server used to let the information available?, Was the information replication considered?, Were resources and techniques considered to guarantee the availability on acess spikes?, Were information protection technologies identified?, Were contingent actions for emergency cases identified?})
R4:QuestionsIdentification(Allow access to information through different means and technologies, {Is the acess possible to different software platforms?, Is the acess possible to different hardware platforms?, Is the access possible with non-electronic media?})

Figure 2. Availability Questions requirements pattern.

"Is the information disseminated from information spreading mechanisms?" The Alternative requirements pattern template from Figure 5 is an adaptation of the Alternative Means pattern proposed by Supakkul et al. This adaptation contemplates the GQO good-practices questions, defining two new refinement rules: QuestionAnswering and UnintendedContribution. The QuestionAnswering refinement rule defines the operationalization as an answer to the pattern's question and its contribution to the NFR. The UnintendedContribution shows the collateral effects of applying this operationalization to other transparency-related NFRs.

On the middle-right branch of Figure 3, the arguments i(arg14) and i(arg15) analyze the contributions of the "Register a dedicated web-domain" operationalization to the Availability NFR and to other transparency-related NFRs. The argument i(arg15) states that his operationalization helps the Simplicity NFR and is a consensus, denoted by the preference rule P2. Figure 6 shows the Alternative requirements pattern (same template as Figure 5) that represents this operationalization as an possible answer to the question "Was digital access service provided for usage?"

The identified Alternative requirements patterns were included on the STC to support future accessibility discussions.

IV. Relative Validation

How can we establish that the identified requirements patterns are valid? As our requirements patterns can be justified by the ACE argumentation graphs, it is possible to validate the patterns by validating the argumentation graph itself. The ACE framework proposes an interesting approach to validation, where validity may be *absolute* or *relative*. *Absolute* validity aspires "*to arrive at RE artifacts that capture exactly what the stakeholders really need*," while *relative* validity "*is concerned with whether stakeholders agree on the content of an RE artifact*." Jureta et al. states that *relative* validity "*is achievable in practice*."

The ACE framework offers algorithms to reason about the relative validity of elicited requirements. Relative validity means that stakeholders agree on the content of an RE Artifact, i.e., the discussed operationalizations to the transparency-related NFRs. This framework consists of (i) a set of concepts for modeling stakeholder arguments as a graph (Figure 3), and (ii) an algorithm to evaluate the acceptability of the graph (to the stakeholders).

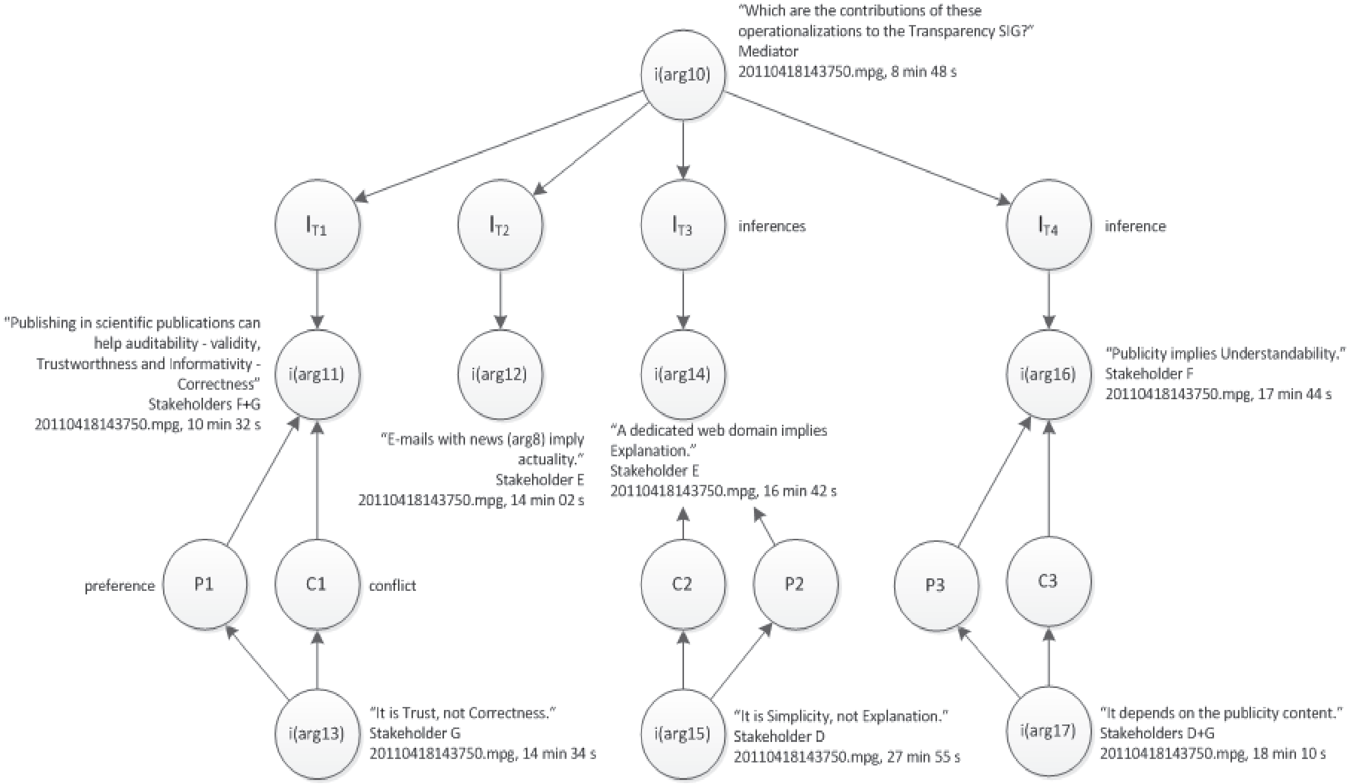
"Which are the contributions of these operationalizations to the Transparency SIG?"
Mediator
20110418143750.mpg, 8 min 48 s
i(arg10)
I_{T1}
I_{T2}
I_{T3}
inferences
I_{T4}
inference
"Publishing in scientific publications can help auditability - validity, Trustworthness and Informativity - Correctness"
Stakeholders F+G
20110418143750.mpg, 10 min 32 s
i(arg11)
i(arg12)
i(arg14)
i(arg16)
"Publicity implies Understandability."
Stakeholder F
20110418143750.mpg, 17 min 44 s
"E-mails with news (arg8) imply actuality."
Stakeholder E
20110418143750.mpg, 14 min 02 s
"A dedicated web domain implies Explanation."
Stakeholder E
20110418143750.mpg, 16 min 42 s
preference
P1
C1
conflict
C2
P2
P3
C3
i(arg13)
"It is Trust, not Correctness."
Stakeholder G
20110418143750.mpg, 14 min 34 s
i(arg15)
"It is Simplicity, not Explanation."
Stakeholder D
20110418143750.mpg, 27 min 55 s
i(arg17)
"It depends on the publicity content."
Stakeholders D+G
20110418143750.mpg, 18 min 10 s

Figure 3. Availability Questions requirements pattern.

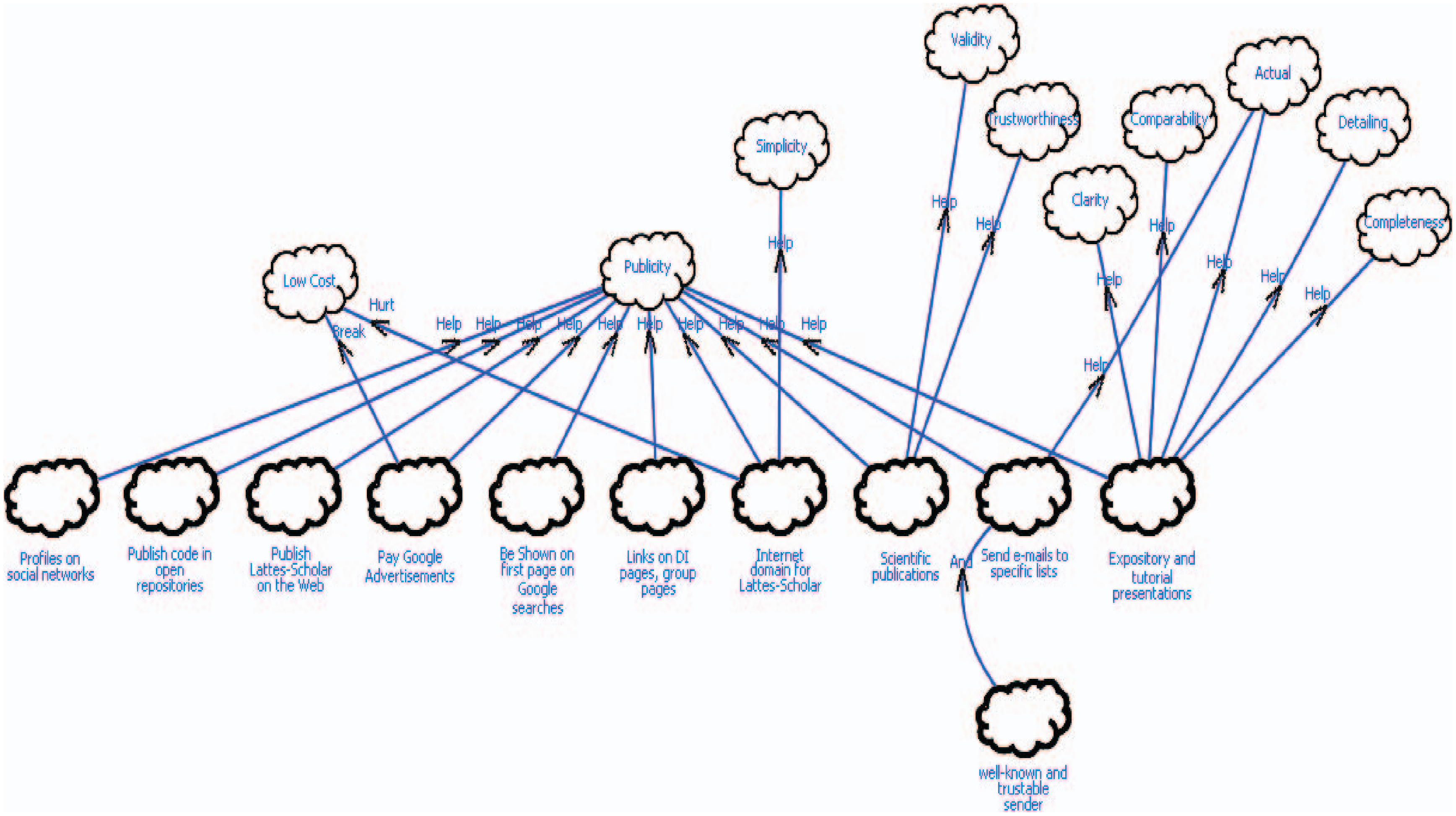
Validity
Trustworthiness
Actual
Comparability
Detailing
Simplicity
Clarity
Completeness
Publicity
Low Cost
Hurt
Break
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
Help
And
Profiles on social networks
Publish code in open repositories
Publish Lattes-Scholar on the Web
Pay Google Advertisements
Be Shown on first page on Google searches
Links on DI pages, group pages
Internet domain for Lattes-Scholar
Scientific publications
Send e-mails to specific lists
Expository and tutorial presentations
well-known and trustable sender

Figure 4. All the discussed operationalizations to the Publicity NFR represented as a SIG.

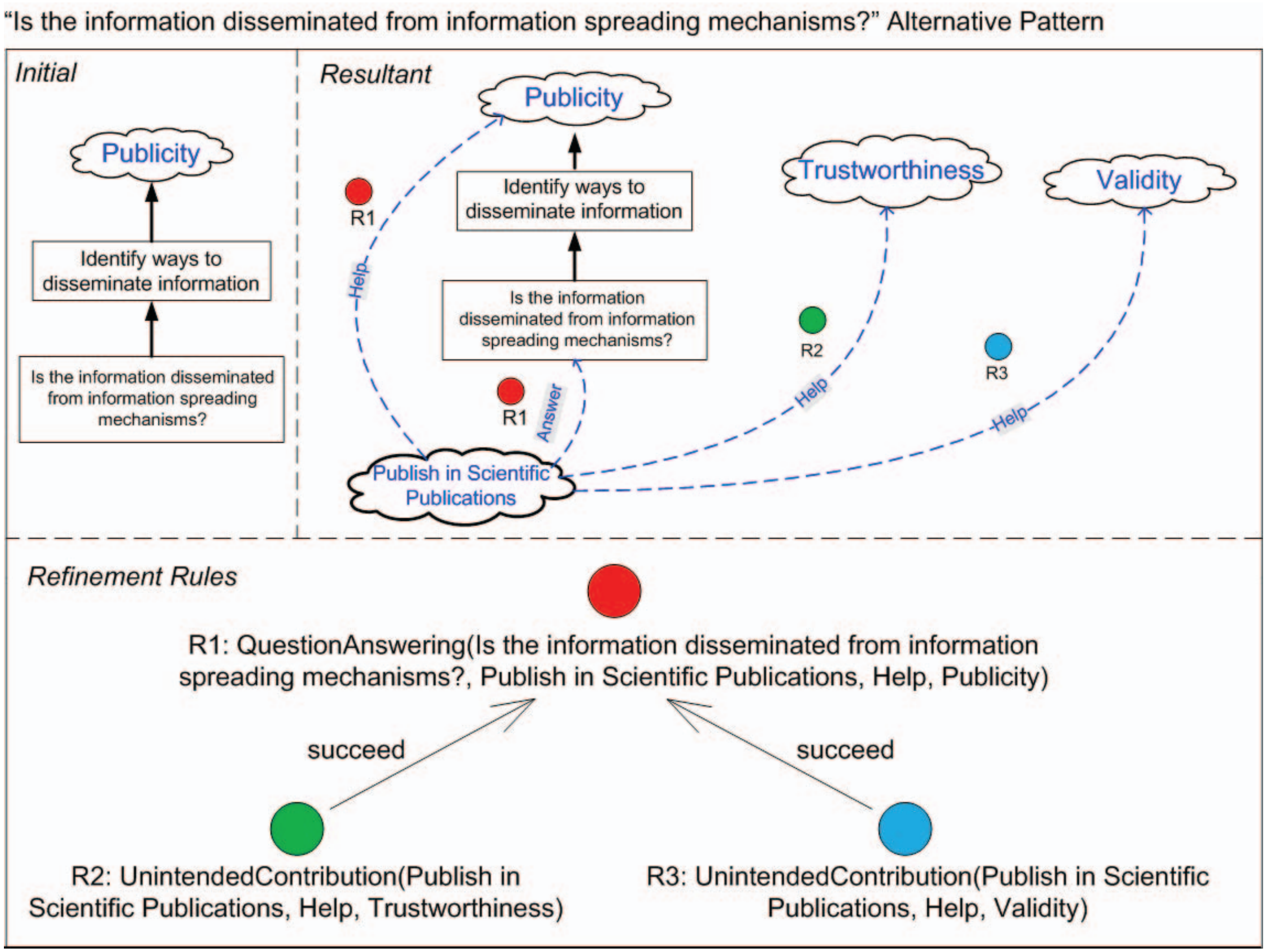
"Is the information disseminated from information spreading mechanisms?" Alternative Pattern
Initial
Publicity
Identify ways to disseminate information
Is the information disseminated from information spreading mechanisms?
Resultant
Publicity
Identify ways to disseminate information
Is the information disseminated from information spreading mechanisms?
R1
Help
R1
Answer
Publish in Scientific Publications
Trustworthiness
Validity
R2
Help
R3
Help
Refinement Rules
R1: QuestionAnswering(Is the information disseminated from information spreading mechanisms?, Publish in Scientific Publications, Help, Publicity)
succeed
succeed
R2: UnintendedContribution(Publish in Scientific Publications, Help, Trustworthiness)
R3: UnintendedContribution(Publish in Scientific Publications, Help, Validity)

Figure 5. "Is the information disseminated from information spreading mechanisms?" Alternative requirements pattern.

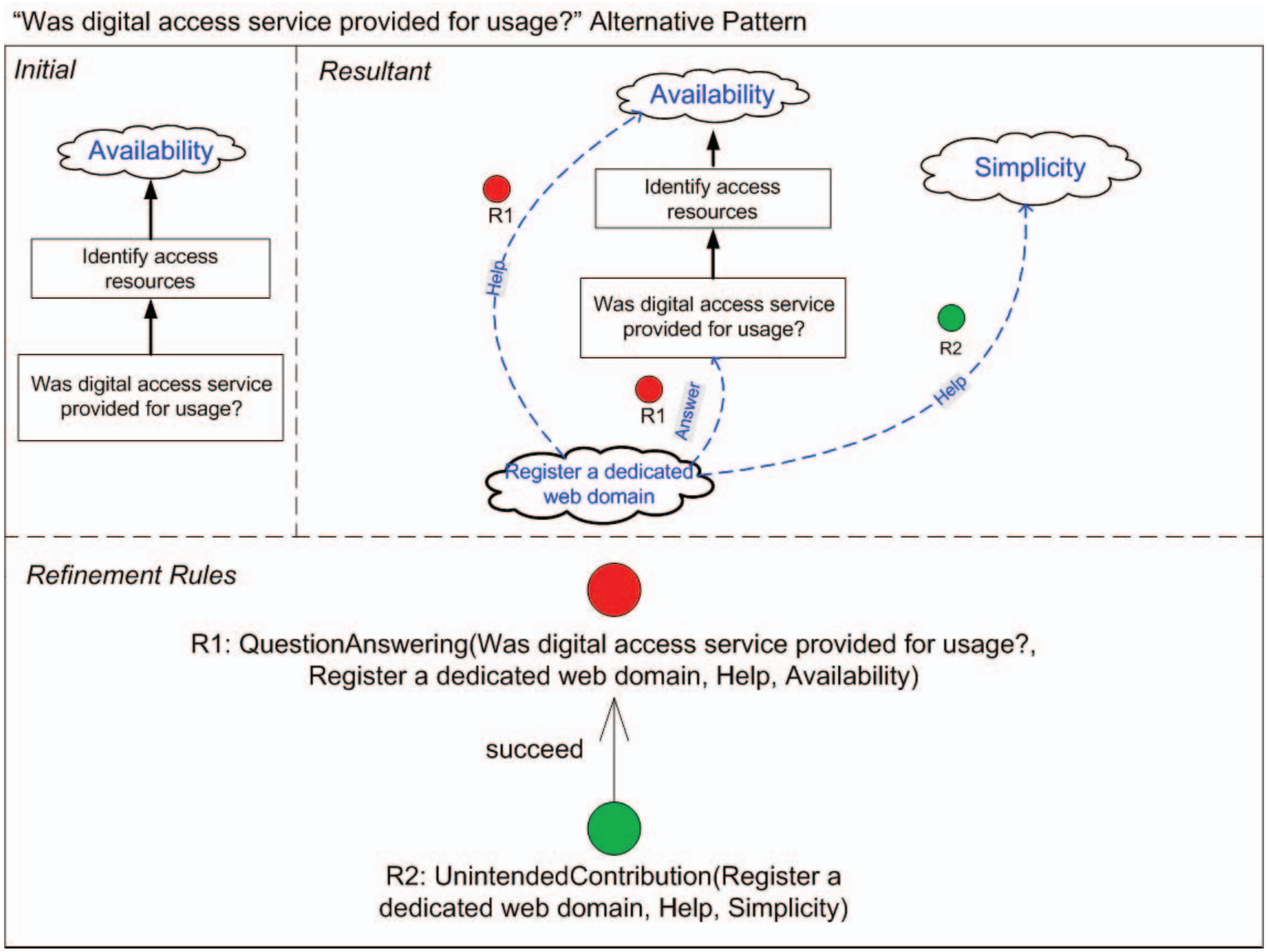
"Was digital access service provided for usage?" Alternative Pattern
Initial
Availability
Identify access resources
Was digital access service provided for usage?
Resultant
Availability
Identify access resources
Was digital access service provided for usage?
Simplicity
R1
Help
R1
Answer
R2
Help
Register a dedicated web domain
Refinement Rules
R1: QuestionAnswering(Was digital access service provided for usage?, Register a dedicated web domain, Help, Availability)
succeed
R2: UnintendedContribution(Register a dedicated web domain, Help, Simplicity)

Figure 6. "Was digital access service provided for usage?" Alternative requirements pattern.

Figure 7 shows an ACE algorithm application example. The ACE algorithm determines if a graph representing elicited requirements and associated arguments is acceptable to the participants in the construction of the graph, or whether it needs further discussion. Basically, each vertex receiving a link from a conflict vertex is being attacked. This attack can be dominated by preference rules. Figure 7 shows how to apply the algorithm: on the first step, the non-attacked vertices are marked “A” (accepted, no one is attacking them). All un-marked vertices are potentially attacked. During the second step, preference rules dominate attacked vertices, marking them as “AD” (accepted and dominated). During the third step, un-dominated attacked vertices are marked “R” (rejected). After applying the algorithm, if any proposition is rejected, the RE artifact is not valid relative to the stakeholders that participated in the discussion.

Although the relative validation guarantees that the stakeholders agree about the discussed operationalizations, there are some problems with this approach. The relative validation means that the RE artifact is valid to the group of stakeholders that participated on the discussion, not necessarily to all stakeholders. During the Lattes-Scholar accessibility discussion, for example, no participant suggested accessibility operationalizations that considered visual impairments.

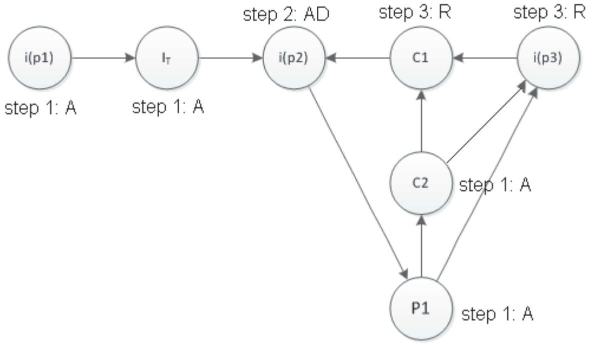

Figure 7. ACE language algorithm application example

There are work on the Human-Computer Interaction (CHI) literature that address usability for visual-impaired users [16]. The daSilva program [17], for example, analyzes if HTML content is compatible with accessibility rules defined by the Web Accessibility Content Guidelines (WCAG). “Writing WCAG compatible HTML content” is a possible operationalization to the Accessibility NFR for visual impaired users. If these operationalizations were available on the STC as Alternative patterns, they could have helped the stakeholders to achieve more inclusive operationalizations.

V. Related Work

We based our patterns templates on Supakkul et al.’s approach to NFR patterns and our customization using the GQO idea [8]. However, their pattern kinds follow a problem/vulnerability/solutions approach similar to other approaches focused on security NFR. As our STC follows an optimistic approach, were we use a Goal-Question-Operationalization method to identify good-practices, we adapted their templates and defined a new pattern kind – the Question pattern.

Konrad and Cheng [18] describe a requirements pattern template similar in style to that used by Gamma *et al.* [19]. They use problem frames and UML diagrams to give structural and behavioral information. We preferred to follow Supakkul et al. approach to requirements patterns templates, as a high abstraction level template is more suitable to be manipulated by stakeholders on a discussion. Konrad and Cheng paper focuses on the pattern template and its validation. They do not apply a formal method to identify requirements patterns. They state that they capture “common” automotive industry requirements.

Hagge and Lappe [20] require that patterns be relevant, applicable and reliable. They defend that patterns should not originate “from academic reasoning alone”. Our key insight to capture requirements patterns after a stakeholders’ consensus guarantees that our patterns originate from relatively validated operationalizations with a practical use on a real scenario.

We could not find on the literature research about capturing and identifying requirements patterns through argumentation or stakeholders’ multi-party agreement.

Other argumentation frameworks, as [21] or [22] could have been used to capture the stakeholders’ arguments. However, the ACE Framework offers all the necessary abstractions to capture arguments on an open discussion: (1) information vertexes for stakeholders’ arguments; (2) inference rules vertexes for arguments chaining; (3) conflict rules for counter-arguments, and (4) preference rules to represent the stakeholders’ choice between conflicting arguments. Moreover, the ACE Framework offers algorithms to identify impasses or consensus and a new approach to validation - *relative* validation.

VI. Final Considerations

We presented on this paper our approach to capture transparency-related requirements patterns through argumentation. Using an argumentation framework [15], we proposed a formal method to capture and identify requirements patterns. As our requirements patterns are captured on a stakeholders’ discussion [14] and are identified after consensus, we can state that our patterns originate from *relatively* validated NFR operationalizations with a practical use on a real scenario. As a second contribution, we improved our adaptation of Supakkul et al.’s approach to NFR patterns to an optimistic, good-practices perspective.

The initial transparency knowledge (namely the Transparency SIG [4] and the GQO questions for each of the twenty eight transparency-related NFR) were represented as instances of the Objective and Question patterns kind. These twenty-nine patterns (transparency objectives pattern plus twenty eight question patterns) were the initial content of the Software Transparency Catalog (STC) [9].

The Transparency Objectives and the accessibility-related Question patterns were applied to support the stakeholders' discussion about the Lattes-Scholar accessibility. The discussed availability, portability and publicity operationalizations were represented as instances of the Alternative pattern kind, evolving the transparency knowledge contained on the STC.

Section 4 showed that the relative validation approach has its problems. Relative validation means that the RE artifact is valid to the group of stakeholders that participated on the discussion, not necessarily to all stakeholders. We are now enriching the stakeholders' discussion with Alternative patterns obtained from the STC to mitigate some of these problems.

As ongoing work, we are identifying on argumentation graphs Alternative patterns to the Operability, Friendliness, Conciseness, Simplicity and Traceability NFRs. As further work, we plan to capture and identify Alternative patterns that operationalize other transparency-related NFRs on future group's meetings argumentation graphs.

ACKNOWLEDGEMENT

Julio Cesar Sampaio do Prado Leite acknowledges the support of FAPERJ. Julio Cesar Sampaio do Prado Leite and Maurício Serrano acknowledge the support of CNPq project 557.128/2009-9 and FAPERJ project E-26/170028/2008 (Programa INC&T - Projeto: Instituto Brasileiro de Pesquisa em Ciência da Web).

REFERENCES

[1] B. Holzner B and L. Holzner, "Transparency in global change: the vanguard of the open society," University of Pittsburgh Press, Pittsburgh, 2006.

[2] P. Meunier, "Software transparency and purity", Communications of the ACM, Doi: 10.1145/1314215.1314232, Vol. 51 Issue 2, February 2008, pp. 104.

[3] R. Mercuri, "Trusting in transparency", Communications of the ACM, Vol. 48 Issue 5, 2005, pp. 15-19.

[4] J.C.S.P. Leite and C. Cappelli, "Software Transparency," Business & Information Systems Engineering: Vol. 2, 2010, Iss. 3, 127-139. Available at: http://aisel.aisnet.org/bise/vol2/iss3/3

[5] L. Chung, B. Nixon, E. Yu and J. Mylopoulos, "Non-functional requirements in software engineering," Kluwer, 2000.

[6] Software Transparency Group's Wiki. Available in: http://www.er.les.inf.puc-rio.br/~wiki/index.php /Transparência_de_Software (last access January 2011)

[7] V.R. Basili, "Software Modeling and Measurement: The Goal Question Metric Paradigm," Computer Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, MD, September 1992.

[8] M. Serrano, S. Supakkul, C. Cappelli, E.K. Almentero, H.S. Cunha, E.S. Monsalve, A.L.C. Leal, H.P. Sousa, V.M.B. Werneck, L.M. Cysneiros and J.C.S.P. Leite, "An NFR Patterns Approach to Building a Software Transparency Catalog," Monograph, Department of Informatics, Pontifical Catholic University of Rio de Janeiro, 10 pages, March 2011.

[9] Catálogo de Transparência - Wiki do Grupo. Available at (in portuguese): http://www.er.les.inf.puc-rio.br/~wiki/index.php/Transparência_de_Software (last access July 2010)

[10] S. Supakkul, T. Hill, L. Chung, T. T. Tun and J.C.S.P. Leite, "An NFR Pattern Approach to Dealing with NFRs," Requirements Engineering, IEEE International Conference on, 2010, pp. 179-188, ISBN: 978-0-7695-4162-4.

[11] M. Serrano, J.C.S.P. Leite, "Elicitação de Requisitos de Transparência através de Argumentação", Monografia, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, 10 páginas, Abril 2011. (in portuguese)

[12] Lattes-Scholar: Home. Available at: http://transparencia.les.inf.puc-rio.br:8080/lattesscholarv2/index.lp (last access August 2010)

[13] Plataforma Lattes. Available at: http://lattes.cnpq.br/ (last access August 2010)

[14] M. Serrano, J.C.S.P. Leite and J. Mylopoulos, "Relative Validation for Transparency Requirements," Monograph, Department of Informatics, Pontifical Catholic University of Rio de Janeiro, 14 pages, February 2011.

[15] I. Jureta, J. Mylopoulos and S. Faulkner, "Analysis of Multi-Party Agreement in Requirements Validation," Proceedings of the 17th IEEE International Requirements Engineering Conference, IEEE Computer Society, 2009, pp. 57-66.

[16] H. Takagi, S. Saito, K. Fukuda and C. Asakawa, "Analysis of navigability of Web applications for improving blind usability," ACM Trans. Comput.-Hum. Interact. 14, 3, Article 13, September 2007, 37 pages. DOI = 10.1145/1279700.1279703 http://doi.acm.org/10.1145/1279700.1279703

[17] L. Ferreira, S. Bacellar e R.R. Nunes, "e-Usabilidade", Rio de Janeiro, LTC, 2008. (in portuguese)

[18] S. Konrad and B. Cheng, "Requirements patterns for embedded systems," Proc. of the IEEE Joint Intl. Conf. on Requirements Engineering (RE'02), 2002, pp. 127-136.

[19] E. Gamma, R. Helm, R. Johnson and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison Wesley, 1994.

[20] L. Hagge and K. Lappe, "Sharing requirements engineering experience using patterns," IEEE Software, IEEE, 2005, 22, pp. 24-31.

[21] J.L. Pollock, "A Recursive Semantics for Defeasible Reasoning," Argumentation in Artificial Intelligence, Springer, 2009, pp. 173-197. DOI: 10.1007/978-0-387-98197-0_9

[22] I. Rahwan and K. Larson, "Argumentation and Game Theory," Argumentation in Artificial Intelligence, Springer, 2009, pp. 321-339. DOI: 10.1007/978-0-387-98197-0_16