Using i* to Elicit and Model Transparency in the Presence of Other Non-Functional Requirements: A Position Paper

Luiz Marcio Cysneiros

School of Information Technology – York University, Canada
cysneiro@yorku.ca

Keywords: Software Transparency, Non-Functional Requirements, istar Framework

Abstract. Transparency has been, for long, a general requirement for democratic societies. The right to be informed as well as to have access to information has been an important issue on modern societies. Nowadays, Transparency has been elevated to a must have property to be delivered by governments and businesses. In an era when computer systems are ubiquitous and present in almost every aspect of our lives, it seems natural that Transparency becomes a key requirement in our software systems. We believe that Transparency can rarely be satisfied. The best we can do is to satisfy it within acceptable limits (satisfice). Therefore, we consider Transparency a Non-Functional Requirements (NFR) that should be elicited and modelled in the presence of other competing and synergistic NFR. Intentions behind the adoption of Transparency may play an important role while eliciting solutions for software to deliver appropriate levels of Transparency. Hence, we believe that the i* framework is ideal to elicit and model Transparency. This work will show initial ideas for using i* to support this effort.

1 Introduction.

Since 2005 we have been following the crescent number of claims from society for government and business to be transparent. In late 2007, the world faced the sub-prime mortgage crisis along with scandals due to enormous bonus for bankers involved in the administration of banks and industries that had to be bailed out by the US and Canadian government. This crisis, partially due to the lack of transparency in business and government regulatory schemas has contaminated other economies throughout the world. It all raised the cry for transparency to reach unprecedented levels.

In 2008, Canadians and Americans citizens were surprised to learn that when accessing Ticketmaster.com looking for tickets that were sold out, Tickemaster would alert they were being redirected to another site. However, it failed to alert that these customers were in fact facing ticket prices almost twice higher than in Ticketmaster. It called our attention that despite the fact that Ticketmaster was being transparent in its action it was not being transparent in its intentions. Driven by this episode, in 2009 we started to investigate the relationship between trust and transparency [2]. Contrary to what we expected, we could find many situations where trust would have a negative impact on transparency and vice versa.

We subscribe to the idea that in the near future the relationship among companies, clients, shareholders and government will be close to what Tapscott and Ticol [10] call "Naked Corporation": "Transparency must now be an explicit factor in nearly every management decision. Customers evaluate the worth of products and services at levels

never before possible. Employees share formerly secret information about corporate strategy, management behavior, and challenges. In a real-time global supply chain, companies and their business partners by necessity share competitive and operational secrets. With the rise of institutional investors, especially pension funds, share owners now scrutinize management like never before. Thanks to instant communications, *whistleblowers and inquisitive media, citizens and communities routinely put firms* under the microscope. And the Internet is a central locus and organizing force for all these activities".

Since software systems are now inherent to our society and part of our daily activities, it seems natural that such move will have to be followed by a change in our focus regarding software development. Systems will have to be developed since its early stages aiming at being Transparent. In fact, during the 2009 edition of the 17th IEEE International Requirements Engineering Conference, two Keynotes speakers[1] many times mentioned the importance of Software Transparency. Despite that, very little work has been done focusing this issue [1][6]

Many authors such as Holzner [5] and Henriques [4] provide different definitions for transparency but central to any definition remains the idea of *information disclosure.* We believe that for software to be Transparent the information which it deals with has to be transparent (Information Transparency). Software also needs to be Transparent itself. It should inform about itself, how it works, what it does and why (Process Transparency). Although approaches such as Open source and well defined and documented software development processes can contribute to software Transparency they alone do not come close to deliver Transparency.

We view Transparency as an NFR and hence, it may positively or negatively impact many other NFR and as a consequence it cannot be studied in isolation. Transparency should be treated as one of the most important contributing NFR since to satisfice Transparency it will be necessary to satisfice many other NFR hence, if transparency is not dealt as one of the leading NFR re-work may have to be done later to adapt existing solutions for NFR such as those illustrated in Figure 1 to cope with Transparency needs.

Moreover, as we could see from our study between Transparency and Trust [2], we need not only to model how, where and when systems need to be Transparent but also the intentions behind the intended Transparency. Every company *will have its own motivation to be transparent* and will deliver Transparency up to a point where it does not threat its business. Therefore, *modeling these intentions would help to clarify alternatives and to better reason about possible tradeoffs* during the elicitation and modeling process. We understand that *the i* framework provides an ideal environment* to support this process.

This work will show initial ideas on the use of i* to support eliciting and modeling Transparency requirements together with other NFR. We present a brief real life case study where Interoperability was studied together with Transparency as a proof of concept on how eliciting and modeling NFR correlations to Transparency may be critical to adequately achieve Transparency. Aside from that, we present other areas where we will investigate different issues to help developing software that will deliver Transparency such as which information should be kept, how it should be stored, how it should be retrieved, how it should be presented.

[1] Jim Herbsleb and Daryl Plummer

2 Objectives of the Research

Our short term goal is to better understand what does it mean to be Transparent and which information should be available to stakeholders to deliver Transparency without ignoring other NFR such as privacy, security, trust, and ethics among others. Note that Leite's work [6] heavily focus on transparency itself while in our focus the relationship between transparency and other NFR such as Trust, Privacy, Security, Ethics and Interoperability will concentrate the majority of our work.

We initially plan to capture this knowledge using catalogues in the form of Softgoal Interdependency Graphs (SIG) the same way we have been doing for capturing knowledge regarding NFR operationalizations in the past. However, we plan to further develop recent work aiming at providing a more efficient and flexible way to store and retrieve information gathered about Transparency and other NFR using ontology and semantic web techniques.

We also plan to investigate how we can categorize software regarding how much Transparency it delivers. We expect that such categorization will lead to standards to measure how Transparent one software is. Such classification may influence the decision process of acquiring one software.

Finally, we will investigate how to incorporate the knowledge for delivering Transparency with the process of eliciting and modeling software allowing the intentions behind adopting Transparencies solutions to be studied in conjunction with all the other intentional elements of the system being developed.

3 Scientific Contributions

Although Process Transparency may complement our research objectives, our main focus will be to investigate methods, techniques and tools to help requirements engineers to develop systems having Information Transparency in its core from the early stages of software development. We believe that the i* Framework will be ideal to provide the necessary structures and constructs to reason about ways to operationalize Transparency and co-related NFR as well as to capture the intentionality that drives individual business and governments to deliver transparency. I* intentional elements and its mechanisms to model intentions, its impacts and alternatives are ideal to deal with Transparency. Furthermore, modeling NFR plays a major role in i* and therefore is a perfect match to model Transparency where we will mostly be dealing with many NFR working together and intentions motivation companies to be and to offer transparency.

4 Ongoing and Future work

4.1 Investigating Transparency in the presence of other NFR

An initial approach to capture Transparency operationalizations can be found in the Transparency Ladder proposed by Leite [6]. In this Ladder we can see that central to deliver Transparency we can find other NFR, namely: Accessibility, Usability, Informativiness, Understandability and Auditability. Accessibility can then be decomposed into Portability, Availability, Operability and so on. Figure 1 Shows part of this Ladder. Of course any operationalizations that we may choose to satisfice Accessibility might contribute positively or negatively to operationalizations that can satisfice other NFR on the

Ladder. Moreover, other NFR outside the Ladder can also bring synergy or conflict to Transparency and vice versa. Hence, the challenge relies not only in finding out operationalizations to deliver Transparency but most importantly, how to harmonize these operationalizations in light of other much needed NFR. Initially, we will be registering our findings in catalogs using the NFR framework.

As a proof of concept, we have recently carried out *a case study* to evaluate the relationship between *Transparency and Interoperability*. This case study investigated particular types of messages exchanged between non-interoperable systems using a message broker within a Health Care Provider in Ontario. The model chosen is based on descriptive methods, particularly an observational case study used to analyze production data, other than laboratorial data, to create a baseline that may be used to verify if Interoperability can hurt or help Transparency. We used the Transparency Ladder proposed by Leite [6] to evaluate its proposed operationalizations for Transparency against possible problems brought by Interoperability mismatches. Figure 1 shows a *partial set* of the correlations that we found. The focus of this work was to evaluate how much incorrect data due to Interoperability issues occurred involving a system A and a system B after they have sent information to a system C. Analyzing these errors we tried to evaluate if despite the errors, Transparency could yet be satisficed at least in a timely manner. As we can see in Figure 1 we realized that there were many situations where errors due to Interoperability problems could have hurt Transparency. Since System C is used as repository of information to be manipulated later, these errors did not pose any risk to the daily operation of Systems A and B. However they would impact Transparency in a short term basis.

Other NFR such as Trust, Privacy, Security and Ethics will soon be investigated to evaluate its relationship with Transparency. Although at least intially we will continue to use SIGs to capture this knowledge, we plan to further develop a recent work using ontology and semantic web to facilitate NFRs knowledge reuse [7][8]. Further details will

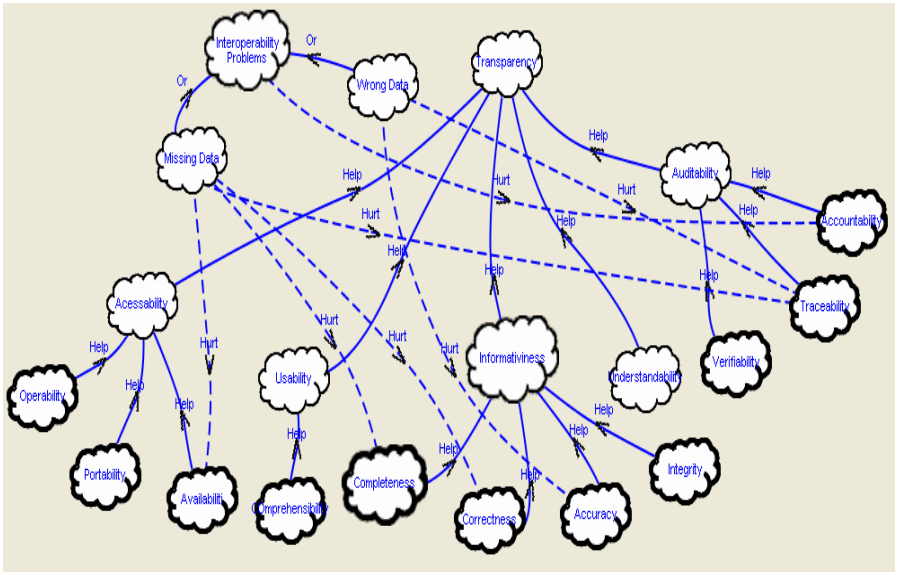

Figure 1 – Interoperability and Transparency

be shown in the section 4.2.

4.2 A Framework to Store and Retrieve knowledge on Software Transparency and Other NFR

SIGs have been used to represent quality attributes. Many complex projects can produce quite large and complex design graphs, which might complicate recovering alternatives, trade-offs and rationale information in SIGs. As an alternative, we started to look at ontologies and sematic web techniques as an alternative to manipulate NFR knowledge.

Ontologies are usually associated to description logic, and represented with languages like OWL [9], which provides vocabulary to describe classes, their properties, relations among them and cardinalities, and support inference rules to derive new information from input data. In operational terms, the Resource Description Framework (RDF) is widely used to describe ontologies (mainly at semantically enriched Web sites). We have developed the NFR and Design Rationale (NDR) Ontology where we implemented a separate faceted and multi-faceted search capability that can recover whole SIGs (or relevant SIG fragments) [8]. The NDR Ontology allows describing well-formed SIGs *in a machine-readable format* and processing the rationale knowledge embedded in SIGs by developing software applications. NDR describes well-formed SIG models: its classes represent SIGs concepts, and its properties describe feasible relationships among these concepts storing them in a central repository for future extraction. We are in the process of developing a tool that would semi-automate the storage process as well as provide query mechanisms to recover knowledge and, in a mid/long term basis, the design rational behind adopted solutions [7]. We expect that this tool would provide a way for retrieving Transparency and NFR related information both from a specific domain as well as from specific projects. This tool will be able to import from existing SIGs as well as to export to other Tools supporting SIGs manipulation. This repository may also be used to generate guidelines to classify software regarding its level of transparency as explained in Section 4.3

4.3 Supporting the Creation of Guidelines to Measure Software Transparency

We also aim at investigating the need for configuring software to deliver different levels of Transparency depending on the stakeholder using it. We anticipate that there will be situations where the levels of Transparency made available to some of the upper management people should not be available to every stakeholder for the sake of, among other reasons, having business strategies kept to those who need to know. Transparency may still have to be provided but with lower levels of Information Disclosure,

Based on the knowledge obtained during the previous phases of our research plan, we will gear towards a product-oriented approach to NFR classification to produce guidelines to be used by a semi-automated tool to attest how Transparent one software is. We will investigate how to create a tool that will support guided interaction with stakeholders that want to use a semi-automated approach based on this knowledge to classify one specific software. Different levels of Transparency will be achievable in a similar way one appliance can be classified for energy efficiency such as in the Directive 2010/30/EU. The use of GRL will be investigated as an option for providing weights schemas to help with the task of categorization. Standards based on these levels of

Transparency may guide software acquisition and Request For Proposal for outsourced software development

4.4 Integrate Transparency Knowledge to Development Models

Orthogonally to the goals mentioned above, we will investigate how to integrate the knowledge stored in the form of SIGs and/or using our tool into i* models reflecting the system being developed.

We believe that intentionality will play a key role to develop software delivering Transparency in accordance not only with the business/government needs but also taking into consideration citizen's needs. We expect that many tradeoffs will take place while addressing Transparency through the viewpoint of citizens, business and government together. We will need to integrate knowledge about Transparency with models depicting the functionality, non-functionality and the intentions related to the domain in question. We will be investigating systematic methods and possibly tools to integrate this knowledge into i* models. We also plan to evaluate if GRL can offer a richer environment when considering jUCMNav [3]. Particularly, we will be focusing on a GRL feature where intentional elements in an actor can be attached to importance factors. These factors can be linked to satisfaction levels to guide the evaluation of the overall satisfaction of an actor. We believe this characteristic may play an important role when evaluating different intentions behind Transparency and other functional and non-functional requirements. Other tools will be also evaluated.

References.

1. Cappeli,C. and Leite, J.C.S.P. "Exploring Business Process Transparency" in Proc. Requirements Engineering Conference, pp:389-390, 2007
2. Cysneiros, L.M. "An Initial Analysis on How Software Transparency and Trust Influence Each Other" 12th Workshop on Requirements Engineering on July 2009, Chile.
3. GRL – ITU-T – User Requirements Notation (URN) – Language Definition, ITU-T Recommendation Z.151 (10/12). Geneza, Switzerland(2012); http://www.itu.int/rec/T-REC-Z.151/en
4. Henriques A., Corporate Truth The Limits to Transparency, EARTHSCAN, UK (2007).
5. Holzner B., Holzner L., Transparency in Global Change: The Vanguard of the Open Society. University of Pittsburgh Press; 1 edition (2006).
6. Leite, J. and Cappelli, C. Software Transparency. Business & Information Systems Engineering, 2, 3 (2010), 127-139. DOI=10.1007/s12599-010-0102-z.
7. López,C., Codocedo,V., Astudillo,H., Cysneiros,L.M.. "Bridging the Gap between Software Architecture Rationale Formalisms and Actual Architecture Documents: An Ontology- driven Approach**.**. Science of Computer Programming Journal. *Volume 77 Issue 1, January 2012. Pg 66-80r*
8. López,C., Cysneiros,L.M., Astudillo,H.. **"**NDR Ontology: Sharing and Reusing NFR and Design Rationale Knowledge". First International Workshop on Managing Requirements Knowledge, Spain 2008. Workshop proceedings in the IEEE Digital Library, MaRK'08, p. 1-10. DOI: http://doi.ieeecomputersociety.org/10.1109/MARK.2008.7
9. McGuinness,D.L., van Harmelen,F. "OWL web ontology language overview," W3C recommendation, http://www.w3.org/TR/2004/REC-owl-features-20040210/
10. Tapscott, D. and Ticoli, D. "The Naked Corporation" The Wall Street Journal Ocotber 15th 2003