A Rich Traceability Model for Social Interactions

Maurício Serrano
Departamento de Informática,
Pontifícia Universidade Católica do Rio de Janeiro
mauserr@les.inf.puc-rio.br

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

ABSTRACT
In 1993, Goguen published a research note addressing the social issues in Requirements Engineering. He identified in the requirements process three major social groups: the client organization; the requirements team; and the development team. However, nowadays there is a lack of technological support that traces requirements to social issues on the requirements team or development team. From early published traceability metamodels to current requirements traceability literature, the client organization and the stakeholders are first-class citizens, but the software engineers and the interactions between these groups are not. In this paper we present a partially formalized RichPicture traceability model to fill this gap. ITrace is a flexible model to weave together the social network graph, the information sources graph, the social interactions graph, and the Requirements Engineering artifacts evolution graph. We empirically developed our traceability model tracking a Transparency catalogue evolution. We also compare our model structure to Contribution Structures.

Categories and Subject Descriptors
D.2.1 [**Software Engineering**]: Requirements/Specifications – *languages.*

General Terms
Documentation, Experimentation.

Keywords
Social issues; requirements traceability; rich picture; graph-based traceability; software evolution.

1. INTRODUCTION
Goguen published a research note [1] addressing the social issues in Requirements Engineering (RE). He identified in the requirements process three major social groups: the client organization; the requirements team; and the development team. He described social issues in each of these groups, and in the interactions between them. It is interesting to notice that nowadays if we search on Google Scholar using the key word "Requirements Traceability", we could not find any recent work that traces the Re artifacts to social issues on the requirements team or development team. Social issues between the client organization, the requirements team and the development team are also not linked to the requirements. The main focus seems to be the client organization and the social issues between the stakeholders. This makes further analysis of the other relevant social issues impacts on the RE process almost impossible.

Originally, traceability in requirements scope was only used to analyze if the required capabilities and software issues were implemented and tested according to the clients' expectative. In their paper, Gotel and Finkelstein [2] introduced the distinction between backward traceability and forward traceability. They also reported several problems in the industry attributed to the lack of pre-Requirements Specification Traceability. Moreover, they described social issues on the requirements team as poor communication, lack of commitment, and lack of accountability on distributed teams.

The requirements pre-traceability was the focus of the Contribution Structures [3], proposed by Gotel and Finkelstein to address the problems identified on their previous work. These dynamic structures are centered on artifacts-based traceability relations and aim the visibility of individuals and groups that contributed to the requirements development. The proposed approach traces information about the individuals roles and commitment to the requirements development and is the foundation for requirements pre-traceability on current traceability metamodels.

The backward traceability discussed by Gotel on [2] was among the issues treated by proposals dealing with Goal-Oriented Requirements Engineering (GORE) [4]. GORE focused on social issues of the client organization, more specifically on the stakeholders' social network. GORE also helps tracing the requirements to the Early Requirements, to the stakeholders' goals, and to the expected quality criteria (softgoals) for the system-to-be. It means that some important information, commonly lost during the traditional Software Engineering Development Process, will be traced, allowing further investigation and analysis.

We understand that GORE uncovered so many research possibilities (e.g. TROPOS [5]) that the social issues on the requirements team, development team, and in the interaction between these two groups and the stakeholders lost focus. This became evident on Ramesh's Traceability Metamodel [6], in which the stakeholders are first class citizens, but the software engineers are not. Ramesh's Traceability Metamodel used Contribution Structures to provide pre-traceability. It was also extended to aim interactive systems [7]. More recent work [8] [9] extended the TROPOS methodology to deal with requirements traceability in agent-oriented development.

In the RE community, two of the most used languages to model GORE are the NFR Framework [10] and the i* framework [11][12], as i* is the base model for the Early-Requirements and

TEFSE'11, May 23, 2011, Waikiki, Honolulu, HI, USA

Late Requirements Phases of the TROPOS Methodology. The construction or analysis of these models involves several interactions with the stakeholders. These interactions result in the evolution of these models. In [5] the authors emphasize that: "...what goals are associated with each actor is a decision of the corresponding stakeholder, not the design team." In this scenario, the interactions with the stakeholders are intrinsic and intense. Other researchers [7] [8] show the complexity of these interactions, but little attention has been given to the pre-traceability of goal-oriented requirements.

On this paper we present a graph-based traceability model to track social actors, social networks and social interactions on RE artifacts evolution – the ITrace. It also tracks the social interactions goals, the performed activities, the techniques applied on these activities, the produced documents and information sources. ITrace's graphical representation is based on RichPicture [13] to facilitate hand-drawing during the activities. ITrace models are partially formalized to offer a common platform as well as a flexible way to extend the models.

This paper is divided in Sections. Section 2 shows an overview about the ITrace model. Section 3 describes the ITrace graphical representation. Finally, Section 4 presents the final considerations and the further work.

2. ITRACE MODEL OVERVIEW

The ITrace model is based on graphs, on which each edge is a trace. ITrace is defined as a set of graphs weaved together. An ITrace model can be described as:

ITrace = (*B*, *I*, *A*, *W1*, *W2*), where:

B = Graph of the social network and the information sources, the base for RE;

I = Graph of the social interactions, the social interactions goals, the performed activities, the applied techniques and the produced artifacts;

A = Graph of the RE artifacts evolution;

W1 = List of weaving traces (edges) between *B* and *I*; and

W2 = List of weaving traces (edges) between *I* and *A*.

It is interesting to compare Gotel and Finkelstein Contribution Structures [3] to the ITrace's structure:

- The relations between the individuals and the artifacts are the Contribution Structures "first-class citizens". The relations between individuals or artifacts and information sources or resources are treated as "second-class citizens". On ITrace all these relations are "first-class citizens".
- Individuals and artifacts are directly related on the Contribution Structures. On ITrace, artifacts (on layer *A*) can only be manipulated by actors (on layer *B*) during social interactions on layer *I*;
- ITrace traces the social interactions goals, the key to understand the individuals' contributions. The Contribution Structures pay no attention to the reasons (the *why*) behind the interactions contributions;
- ITrace applies visual symbols, e.g. cartoon "thoughts," to represent the actors' concerns on a social interaction. The Contribution Structures pay no attention to the individuals' concerns while they contribute to the artifacts; and
- Gotel and Finkelstein apply text linguistics to obtain the artifacts connectivity relations. This approach analyzes the produced artifacts after the social interaction. ITrace benefits from RichPicture's "participatory design" [13] to obtain the relations while the artifact is being produced. These relations can be validated with the stakeholders "on the fly".

3. GRAPHICAL REPRESENTATION

ITrace graphical representation is based on RichPicture [13], which is a simple graphical notation tool to be used at the workplace. Monk and Horward state that RichPicture is a useful tool in "designing design", and encourages Participatory Design of the work-context.

We empirically developed the ITrace by tracing nine months of weekly meetings. On these meetings, at least five PhDs, five PhD students and six master degree students collaboratively contributed to the evolution of a Transparency Catalogue [14]. This catalogue models the Transparency concept as a non-functional requirement (NFR), decomposing it on a Softgoal Interdependence Graph (SIG) of five softgoals [15]: Accessibility, Usability, Informativeness, Understandability and Auditability. Each NFR/Softgoal decomposition provides a deeper understanding level of the Transparency concept.

Figure 1 shows the ITrace model of the meeting that decomposed the Accuracy NFR, an NFR related to the Informativeness NFR. The model was hand-draw by a modeler with less than one hour of training during the meeting, allowing it to be validated "on the fly". It was then redraw on a computer tool after the meeting and uploaded to the group's Wiki for public access. The ITrace model is draw using three layers: bottom layer, for ITrace's graph *B*; middle layer, for graph *I*; and upper layer, for graph *A*.

On the model's bottom layer (Figure 1, layer *B*), the Software Transparency Group is represented as a social network. This social network was elicited from work relationships and the way social actors contact each other. The relationships between the social actors are traced as directional links such as "student_of" and "collaborates_with", capturing more information than a simple meeting minuta participants list, for example. The social actors concerns or goals are represented as cartoon "thoughts", e.g. "Accuracy and Precision are different NFRs". We still recover to these models to observe the social actors concerns and remember the meetings "mood". The information sources and resources are also represented on the bottom layer: "Wordnet Search: Accuracy" is an information source consulted by the "Brainstorm" technique and "Internet" is a resource consumed by the "Decompose the Accuracy NFR" activity.

On the middle layer (Figure 1, layer *I*), the social interactions are draw following a timeline from left to right. The "Meeting" social interaction has the goal "Transparency Catalogue Be Evolved". The "Decompose the Accuracy NFR" activity applied a "Brainstorm" technique in order to achieve this goal. Initially, ITrace models did not capture the social interaction goals; this necessity was observed during some meetings where we had to consult previous models. Some social actors noticed that formally capturing this kind of information on a traceability model allowed us to trace back the why of the RE artifact evolution. The

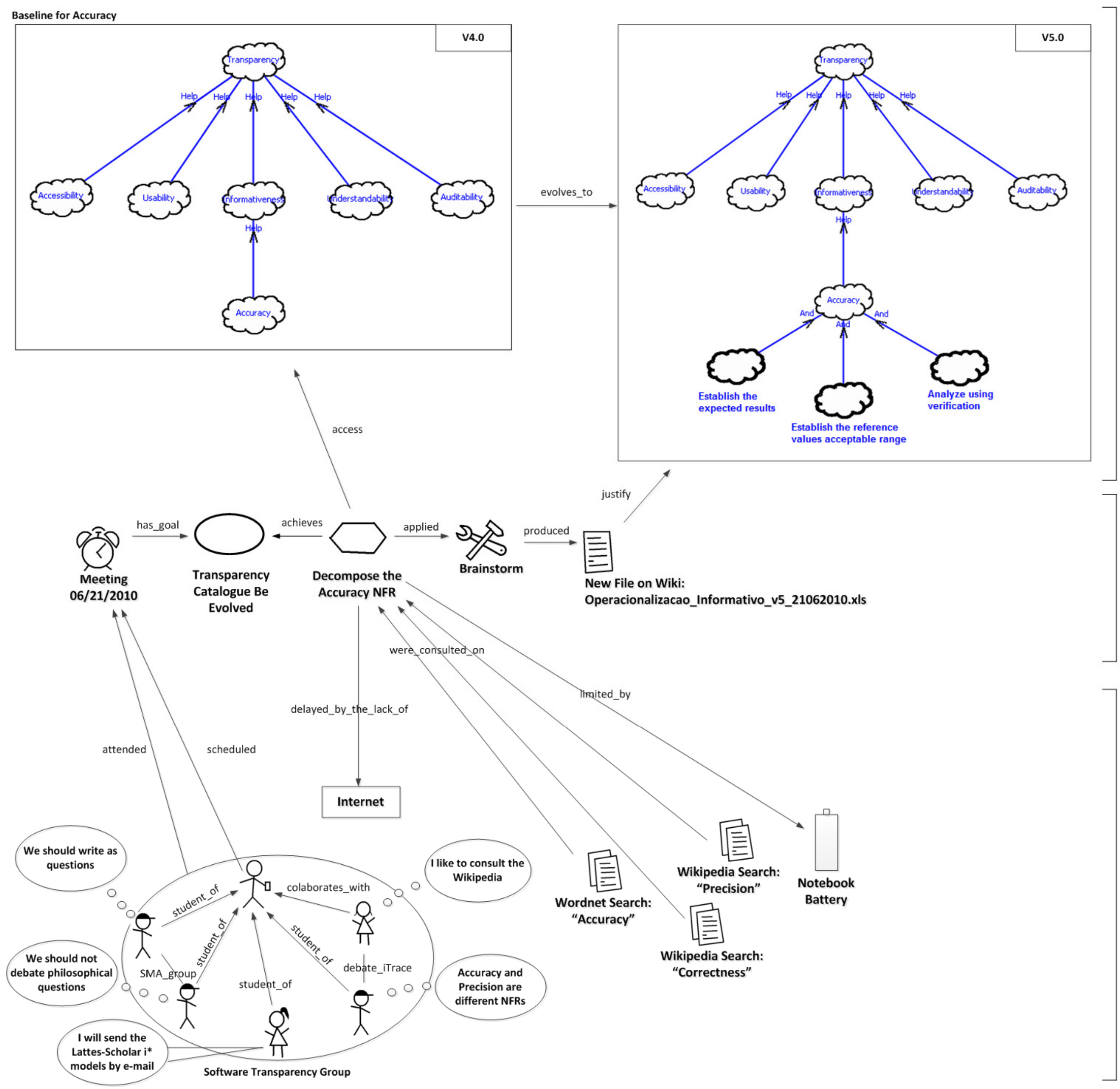

Figure 1. ITrace model of the Accuracy NFR decomposition

“Brainstorm” technique produced an RE artifact that shows how the Accuracy NFR was evolved.

On the upper layer (Figure 1, layer *A*), the evolution of the Transparency SIG is show. The baseline for the Accuracy NFR was evolved by decomposing the Accuracy NFR in three operationalizations: “Establish the expected results”, “Establish the reference values acceptable range” and “Analyze using verification”. The artifacts snapshots represent different versions or views of the artifacts. Thus, in this layer we also have the notion of timeline. It is possible to refine/evolve the RE artifacts as shown on Figure 2. An introductory overview about software evolution can be found on [16]. Figure 2 illustrates an example of model evolution. The original baseline i* model version 1.0 evolves to i* model version 1.1 that evolves to i* model version 1.2. Moreover, this original baseline also branches to i* model version 1.0.1 that evolves to i* model version 1.0.2.

Three social actors played the role of modelers during these nine months of weekly meetings. Slight differences can be observed on models produced by different modelers, as the nodes and links labeling expressiveness, nodes positioning, and the amount of captured information. Some stakeholders were creative and captured unusual information, such as the “Notebook Battery” resource limiting the “Brainstorm” technique shown on Figure 1. This was expected, as ITrace is semi-formalized and the graphical representation is based on RichPicture.

We avoid using weaving links between the bottom layer and the upper layer (graphs *B* and *A*) as a first good practice rule. We

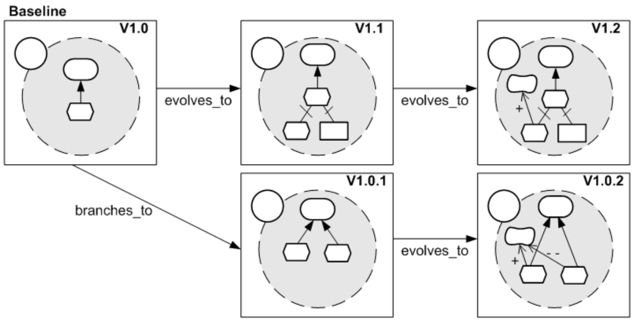

Figure 2: Example of a model evolution

understand that the interactions between social actors and objects always take place on activities performed on social interactions. Moreover, the layers separations should be clear, avoiding merging the Social Interaction Graph on layer *I* with the Information Sources Graph on layer *B*. As a second good practice rule, we use arrows to guide the reading direction of the traces, although traces are formally bidirectional edges on the graph.

4. FINAL CONSIDERATIONS

Current traceability models fail to trace the requirements back to social issues as described by Goguen and Gotel. In this paper, we present our efforts in order to deal with this concern and fill this technological gap using a richer traceability model, ITrace.

ITrace traces the RE artifacts evolution back to the activities that were performed on social interactions. ITrace can give an invaluable insight on how we evolve RE artifacts and, consequently, on how we evolve software.

ITrace is a practical and flexible model. It can be easily replicated and used by anyone without extensive training. We formalized only the minimum necessary to define a common base for all ITrace users, but the three interweaved graphs can be extended to comply with a RichPicture graphical representation. There is no such thing as a "correct" way to construct an ITrace model: the modeler can do what works for her/him. What really matters is to not lose the traces between the RE artifacts and the social interactions that produced/evolved these artifacts.

ITtrace graphical representation can become cumbersome as the model evolution is an ever-increasing graph. Graphical representations should be used only to show a few social interactions, producing a "snapshot" of the artifact at that point in time. Finally, ITrace can also trace other software engineering artifacts by extending the third graph *A*.

The success of Google [17] is based on the representation of the Internet - its sites, resources and links - as a graph. ITrace tries to apply this idea. As for future work, these obtained and maintained graphs will be analyzed in order to identify patterns for RE artifacts evolution. We also intend to analyze the use of tags on ITrace in order to determine if it helps on the elicitation of tacit knowledge.

REFERENCES

[1] Goguen, J. A. 1993. *Social Issues in Requirements Engineering*. In Proceedings of the IEEE International Symposium on Requirements Engineering, pp. 194-195, San Diego, California, January.

[2] Gotel, O. and Finkelstein, A. 1994. *An Analysis of the Requirements Traceability Problem*. In Proceedings of First International Conference on Requirements Engineering, pages 94-101.

[3] Gotel, O. and Finkelstein, A. 1995. *Contribution structures*. Proc. 2nd Intl. Symp. Requirements Engineering, York, pages 100-107.

[4] Lamsweerde, A. van, 2001. *Goal-Oriented Requirements Engineering: A Guided Tour*. Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01), Publisher IEEE Computer Society, ISBN: 0-7695-1125-2, p.249, Washington, DC, USA.

[5] Bresciani, P.; Perini, A.; Giorgini, P.; Giunc, F. and Mylopoulos, J. 2004. *Tropos: An Agent-Oriented Software Development Methodology*. Autonomous Agents and Multi-Agent Sytems, Kluwer Academic Publishers, 8, pp. 203–236.

[6] Ramesh, B. and Jarke, M. 2001. *Toward Reference Models for Requirements Traceability*. IEEE Transactions on Software Engineering, Vol. 27, No. 1, pages 58-93, January.

[7] Toranzo, M. A. 2002. *Um Framework para Melhorar o Rastreamento de Requisitos*. Dissertação de Doutorado, Centro de Informática da Universidade Federal de Pernambuco-UFPE, Brazil.

[8] Castro, J.; Pinto, R.; Castor, A. and Mylopoulos, J. 2003. *Requirements Traceability in Agent Oriented Development*. Software Engineering for Large-Scale Multi-Agent Systems, LNCS, Springer-Verlag Berlin Heidelberg, pp. 57-72.

[9] Pinto, R. C.; Silva, C. and Jaelson Castro, J. 2005. *A Process for Requirement Traceability in Agent Oriented Development*. In: Proceedings of WER'05. Workshop on Requirements Engineering, Portugal, pp. 221-232.

[10] Chung, L., Nixon, B. A., Yu, E. and Mylopoulos, J. 1999. Non-functional requirements in software engineering. ISBN: 978-0-7923-8666-7, Springer, 476 pages.

[11] Yu, E. 1995. Modelling strategic relationships for process reengineering. PhD Thesis, ISBN:0-612-02887-9, 181 pages, University of Toronto, Canada.

[12] i* Framework - Wiki of the Computer Science Group at UofT. Available at: http://istar.rwth-aachen.de/tiki-view_articles.php (last access July 2010)

[13] Monk, A. and Howard, S. 1998. *The Rich Picture: A Tool for Reasoning about Work Context*. Methods and Tools, ACM 1072-5220/98/0300, pp. 21-30, April.

[14] 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)

[15] Leite, J. C. S. do P. and Cappelli, C. 2010. *Software Transparency*. Journal Business & Information Systems Engineering 2(3), pp. 127-139.

[16] Lehman, M. and Ramil, J. 2003. *Software Evolution – Background, Theory, Practice*. Information Processing Letters, Vol. 88, Issues 1-2, pages 33-44, October.

[17] Brin, S. and Page, L. 1998. *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Computer Networks and ISDN Systems, Elsevier Science Publishers, pp. 107-117.