

# Software Transparency

Julio Cesar Sampaio do Prado Leite

Departamento de Informática

PUC-Rio

[www.inf.puc-rio.br/~julio](http://www.inf.puc-rio.br/~julio)

]“Software is deemed transparent if it makes the information it deals with transparent (information transparency) and if it, itself, is transparent, that is it informs about itself, how it works, what it does and why (process transparency).” [1]

Scholars in the area of social science have been exploring the concept of transparency and its importance to democratic societies [2] [3]. Most of these scholars believe that informing society of how processes are being conducted is beneficial to society as a whole. The right of being informed is essential to citizens, if they are to fully participate in a given society.

Transparency can be understood, in the broad sense, as described above, but it can also be more focused. Shareholders require transparency of how the business is being conducted. Regulatory agencies require that organizations under their span be transparent. External auditors need that financial records of a business be transparent. Participants in groups usually feel better when information is open to all participants.

As society becomes more and more digitalized, most of the information is being kept in digital repositories. As noted above, this digitalized information will also encompass the processes (software) that deal with this information. In this context, we posit that society will require that digitalized information and digitalized processes be transparent. Making information available to citizens is already a standard procedure in more organized and open societies, mainly regarding government information. However, digitalized processes are seldom transparent to citizens.

Demanding information and processes to be transparent, society will transfer to software professionals a chore that is highly demanding. One of the problems with transparency is that it is a fuzzy concept. What does exactly mean to have software that is transparent? How to implement transparency? What is the implication of producing transparent software? What type of methods and tools will be needed to fulfill this demand?

Our first result in studying the topic, which we consider a key achievement, is regarding transparency as a “softgoal”. Using the Toronto approach [4] we inherit a framework (The NFR Framework) built to deal with goals that are not satisfied, but satisfied (a term coined by Herbert Simon). Using an extensive survey and questionnaire driven validation we built a softgoal network to define transparency. The network has 27 leaf nodes and 4 clusters. Each node is a characteristic or a softgoal that is necessary as to achieve the transparency softgoal. We also built an initial set of operationalizations for each of these 27 leaf nodes, and a first version is already available as a questionnaire to gauge transparency of web sites.

In order to deal with the challenge of providing software transparency, we are adopting a strategy that focus on transparent requirements. In the words of Professor Mylopoulos:

*“Transparency is an interesting quality because it makes it necessary to attach requirements models to software.”*<sup>1</sup>. This statement posits that requirement models are the right vehicle for the openness necessary for transparency, but it says more. Demanding that requirements models be attached to software brings up the issue of trust, that is, it is understood that software has to behave as prescribed by the requirements models.

Assuming that the running code implements the requirements, it brings the problem of software transparency to a high level of abstraction. Note that by Mylopoulos’s observation the fact of attaching requirements models to code is necessary but, it does not say it is sufficient.

Even if we have means, like monitors, to guarantee that the running software is implementing the requirements, we need to deal with the issue of presenting the process to the citizen, depending on his/hers level of “curiosity”, which may imply, even, to look at the source code or in the certified monitoring that the code is being executed as described.

Our concerns are similar to the ones expressed by Weitzner et al.[5]; where transparency and accountability are terms frequently associated in the text. However, in our characterization of Transparency, accountability is one of the 27 leaf nodes elicited for defining transparency. In a report to the National Research Council, Jackson et al. [6] stress that software companies must make claims of dependability transparent to citizens so they may have an informed judgment about the products being offered. From our point of view dependability is a softgoal (one of the 27 leaf nodes). The different viewpoints expressed by [5] and [6] stress the challenge to deal with accountability and dependability in large software systems, which we believed is compounded by the fact that, in both cases, the focus is not on transparency, but on other qualities, that will need transparency as to be effective.

Adding to those challenges we foresee that the concerns of the software industry will be, in the future, targeted to **citizens**, and as such, demanding more sophisticated design models rather than the usual user or client oriented ones.

## References

- [1] 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, 2008, pp. 51-54 (<http://CEUR-WS.org/Vol-322/>).
- [2] Fung A., Graham M., Weil D., Full Disclosure, the Perils and Promise of Transparency, Cambridge University Press, 2007.
- [3] Holzner B., Holzner L., Transparency in Global Change: The Vanguard of the Open Society. University of Pittsburgh Press; 1 edition, 2006.
- [4] Mylopoulos J., Chung L., Nixon B., Representing and using nonfunctional requirements: a process-oriented approach - IEEE Transactions on Software Engineering, 1992, 18(6):483-497.
- [5] Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T., and Weitzner, D. 2008. Web science: an interdisciplinary approach to understanding the web. Commun. ACM 51, 7(Jul. 2008), 60-69. DOI= <http://doi.acm.org/10.1145/1364782.1364798>
- [6] Weitzner, D. J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., and Sussman, G. J 2008. Information accountability. Commun. ACM 51, 6 (Jun. 2008), 82-87. DOI= <http://doi.acm.org/10.1145/1349026.1349043>

---

<sup>1</sup> Personal communication