The Thing Itself Speaks

Accountability as a Foundation for Requirements in Sociotechnical Systems

Amit K. Chopra
Lancaster University
Lancaster LA1 4WA, UK
a.chopra1@lancaster.ac.uk

Munindar P. Singh
North Carolina State University
Raleigh, NC 27695-8206, USA
m.singh@ieee.org

***Abstract*—We consider sociotechnical systems (STSs) that facilitate social interaction among autonomous principals (either humans or organizations). Although accountability is a foundational concept in such systems, established requirements engineering methods do not support accountability in the broad sense of calling to account of one party by another. To address this shortcoming, we propose the notion of accountability requirement. Further, we claim that to model an STS means to precisely capture the accountability requirements between its principals.**

I. Overview

Autonomy and accountability are fundamental concepts in understanding sociotechnical systems. Autonomy means each principal is free to act as it pleases; accountability means that a principal may be called upon to account for its actions. In general, balancing autonomy and accountability is crucial for ensuring that an STS would not devolve into the extremes of chaos or tyranny. Accountability doesn't conflict with autonomy in that a principal can violate any expectation for which it is accountable: it would merely be held to account.

Our thesis here is that to model an STS is to precisely capture the accountability requirements between its principals. Anything less would lead to unsound solutions; anything more would lead to over-coupled solutions. Notice that we do not presume a black-box model for components. In many cases, internal implementation details may be subject to an accountability requirement and may need to be suitably exposed.

Our formulation of accountability is purely normative: accountability requirements describe how principals ought to act in each other's eyes, providing a basis for their mutual expectations. More precisely, an accountability requirement expresses that one party—the *account-taker*—expects certain behavior from another—the *account-giver*. The a-giver is accountable to the a-taker.

Let's consider some examples. In the meeting scheduling system of RE folklore, we may model that a meeting participant who accepts a meeting invitation from the meeting initiator is accountable for showing up. In a food safety systems, we may model that a food company is accountable to the regulator for maintaining specified tracking information and making it available to the regulator upon demand.

The foregoing view of accountability captures the key intuition of scholars outside of IT [1]. As a normative conception, accountability is independent from both support mechanisms (such as the traceability of actions to principals) and sanctioning processes.

At the risk of over-simplification, we observe that the RE literature approaches the engineering of STSs from two main perspectives. First, the information systems dominant analyses consider the social and organizational aspects of deploying software solutions in real-life organizations. These approaches concern themselves with themes that affect the success or failure of an IT deployment, including organizational culture and the motivations and incentives of the participants. Although these approaches introduce relevant concepts, they provide no clear computational logical path to reasoning about them, in essence, relegating the concepts to be no more than informal guidance for designers.

Second, the modeling dominant approaches provide a formal notation in which to express elements of requirements. Prominent among these are the goal-oriented approaches such as KAOS and Tropos. These approaches concern themselves with rendering selected social and organizational notions in formal terms to guide the design process. However, the particular concepts chosen in these approaches have either a purely functional or at best a cognitive underpinning, in essence, disregarding accountability. Notice that to simply say that the regulator *depends* on the company for tracking information—as one might do in Tropos—is too weak: Tropos dependencies are not relationships of accountability, and, as such, inadequately constrain the behavior of principals.

Important directions of future work include (1) expressive accountability requirements and their formalization, (2) methodologies for deriving and modeling STS specifications in terms of accountability requirements (partially addressed in [2]), and (3) methodologies for deriving software specifications from accountability requirements.

Acknowledgments

Munindar Singh was partially supported by U.S. Department of Defense under the Science of Security Lablet grant.

References

[1] E. J. Emanuel and L. L. Emanuel, "What is accountability in health care?" *Annals of Internal Medicine*, vol. 124, no. 2, pp. 229–239, Jan. 1996.

[2] A. K. Chopra, F. Dalpiaz, F. B. Aydemir, P. Giorgini, J. Mylopoulos, and M. P. Singh, "Protos: Foundations for engineering innovative sociotechnical systems," in *Proceedings of the 22nd International IEEE Conference on Requirements Engineering*, 2014.