On Designing Usable Policy Languages for Declarative Trust Aggregation

Michael Huth and Jim Huan-Pu Kuo

Department of Computing, Imperial College London, London, SW7 2AZ, United Kingdom
{m.huth,jimhkuo}@imperial.ac.uk

Abstract. We argue that there will be an increasing future need for the design and implementation of declarative languages that can aggregate trust evidence and therefore inform the decision making of IT systems at run-time. We first present requirements for such languages. Then we discuss an instance of such a language, $\text{Peal}^+$, which extends an early prototype $\text{Peal}$ that was researched by others in collaboration with us. Next, we formulate the intuitive semantics of $\text{Peal}^+$, present a simple use case of it, and evaluate to what extent $\text{Peal}^+$ meets our formulated requirements. In this evaluation, particular attention is given to the usability aspects of declarative languages that mean to aggregate trust evidence.

1 Introduction

There is little doubt that the advances in computing and information technology are transforming the manner in which we conduct our business and lead our personal lives. The use of small devices such as tablets and smart phones, the rapid pace with which such technologies evolve, and the increased reach of these technologies – to name smart meters for electric power supply – are prominent examples of this.

One consequence of this is that ever more things have programmable interfaces to which other things and processes may connect. In this Internet of Things, designers, programmers, and users alike need to be able to formulate constraints on the interactions across such interfaces that adequately reflect implicit trust assumptions, risk appetite, and other intentions. We think that trust management will play a key role in the articulation of such interaction constraints. Let us first state what we mean by the term “trust” in this paper. We say that an agent $A$ (a program, a user, a system, etc.) trusts another agent $B$ for a planned interaction $I$ with $B$, when agent $A$ has collected and inspected evidence that leads agent $A$ to believe that engaging in the interaction $I$ with $B$ is worth taking any risk reflected in the studied evidence. This asymmetric view of trust can be made symmetric by letting agent $B$ perform a similar inspection and decision process regarding the interaction $I$ with agent $A$ – typically based on evidence pertinent to agent $B$. If we think of the symmetric view as a logical and of the asymmetric view (a trust-mediated interaction would only take place if both agents agree to it), we can focus on the asymmetric view subsequently.
In this paper, we investigate how declarative languages can help with formalizing the process of collecting and studying indicators of trust in context-dependent interactions with other agents. We posit that such languages will be increasingly needed, and formulate requirements that they should satisfy. For sake of illustration, we will study an extension of a trust-aggregation language that we have designed with others [8,2,7,9] and assess its suitability against the requirements we will formulate further below. This will in part make use of a toy example written in that language. The paper concludes by identifying future work for the design of more usable trust aggregation languages.

2 Declarative Trust Aggregation through Policies

We refer to trust aggregation as a process in which agent A first collects observable indicators of trust in an interaction I with agent B, and then systematically combines such indicators to more compact or abstract expressions that can directly inform the decision making of agent A (e.g. whether or not to commit to interaction I with agent B). Note that observable indicators may also be estimates, for example, the estimated uncertainty in the computed reputation of an agent. Such indicators may themselves be talking about perceived trust, reputation scores, risk levels, or about things that influence trust perceptions indirectly – for example the financial risk to agent A in interaction I with agent B. We note that existing approaches to computing trust or reputation scores (including those that have only binary scores as in “trust” or “don’t trust”) acknowledge that such computations benefit from incorporating context-dependent information in the aggregation of trust evidence. For example, reputation systems for online trading sites that base reputation scores on the number of successful past interactions without taking into account the monetary values of these transactions are subject to active attacks. Such attacks can indeed be prevented or mitigated against by making reputation scores dependent on transaction values as well – see for example the nice discussion in [11].

We posit that the future Internet of Things will have an increased need for using such trust aggregations at many interfaces, and that this creates the need for a sort of trust calculus as the basis of computing the perceived trust in interactions across interfaces, which can then be enforced at run-time. No doubt will this lead to many dialects or variants of such a calculus. But there is evolutionary pressure to standardize such aggregation languages in order to get portability across platforms and technologies. Additionally, the creation of a more generic trust calculus will facilitate the development of robust analyses of aggregations formulated in such a calculus. It will also allow the decoupling of executable and analyzable such core languages from user-facing and domain-specific languages for expressing trust aggregation. With such a separation of concerns, one would for example only have to compile user-facing domain-specific languages into a (not user-facing) core language that can be implemented in systems and for which the desired analyses can be performed.