Languages for prototyping regulated systems: A case study

L. Thomas van Binsbergen – University of Amsterdam, The Netherlands, ltvanbinsbergen@acm.org

Lu-Chi Liu – University of Amsterdam, The Netherlands, l.liu@uva.nl

Abstract

Software systems may need to comply with certain laws and regulations and are expected to faithfully capture the (business or governmental) policies behind the services they implement. In data sharing systems, additional sharing agreements between data owners and data controllers need to be respected, as well as the privacy of the data subjects identifiable in data. In other words, software systems need to comply with norms from a variety of sources. Designing a software system to meet these requirements is complicated due to the increasing volume and dynamic nature of the relevant norms. For example, a system can be rolled out in different geographical areas in which different laws and regulations apply. Sharing agreements may be produced dynamically as the result of a negotiation process. Laws and regulations are frequently amended, modified or replaced.

As part of a collaboration between several research institutes and industry partners we are developing architectures for (data-sharing) systems in which norms from a variety of sources can be embedded as so-called regulatory services. The regulatory services are derived from (high-level) specifications written in domain-specific languages to achieve a high degree of trust in and transparency of the regulatory services, and by extension the regulated system itself. We opt for an actor-oriented approach in which norms are enforced by actors that deliver rewards or punishments, resembling regulated *social* systems, and compliance is not necessarily hard-coded into the system [**?**]. By enforcing compliance at runtime rather than statically, regulated systems are able to adept more easily to changes in both norms and execution context. Moreover, our approach makes it relatively easy to simultaneously apply sets of norms that have been constructed independently of each other. We consider the resulting potential for conflicts between sources of norms, e.g. being prohibited from doing A whilst simultaneously having the obligation to do A, as a strength of our approach rather than a weakness, demanding an explicit preference mechanism to determine which norms to adhere to in case of conflict.

The regulatory services of the architectures we propose can be used to *control* the behavior of actors in the system by preventing non-compliant actions and encouraging the satisfaction of obligations. The system is *monitored* to make inferences about the behavior of actors and to make judgements about their normative positions, i.e. whether actors posses permissions, obligations, prohibitions and powers. If non-compliant behavior is observed, *enforcement* actors can act upon these violations by executing certain powers, e.g. placing a penalty or demanding compensation. The regulatory services produce traces from which detailed reports can be constructed for the purposes of *diagnosis*. These reports can be presented to (both human and software) actors of the system and can be used to modify their behavior, e.g. by learning and avoiding non-compliance. Together with the formal specification of norms, such reports also simplify manual auditing processes.

In this presentation we discuss an outline of our general approach, the tools and languages we use to develop prototypes and parts of the specification and prototype of a case study within the financial domain related to the Know Your Customer (KYC) principle. In particular, we show how normative specifications written in the eFLINT language [**?**] and agent specifications written in the AgentScript language [**?**] form the basis of a prototype architecture with regulatory services that enforce norms from a variety of sources related to the KYC case.

Keywords: regulated systems, executable specifications, runtime compliance-checking, runtime enforcement, actor-oriented programming, distributed systems, regulations, law, privacy, contracts