Towards Reactive Semantic Web Policies: Advanced Agent Control for the Semantic Web

José Júlio Alferes, Ricardo Amador
CENTRIA, Universidade Nova de Lisboa, Portugal

Philipp Kärger, Daniel Olmedilla
L3S Research Center, Hannover, Germany

ABSTRACT

The Semantic Web envisions a distributed environment with well-defined data that can be understood and used by machines. This machine-understandable knowledge allows intelligent agents to automatically take decisions and perform tasks on our behalf. In the past, different Semantic Web policy languages have been developed as a powerful means to describe a system's behavior by defining statements about how the system must behave under certain conditions. With the growing dynamics of the Semantic Web the need for a reactive control based on changing and evolving situations arises. This paper presents preliminary results towards a framework for the specification and enforcement of reactive Semantic Web policies, which can be used in order to allow agents to automatically perform advanced and powerful tasks, which can neither be addressed by existing Semantic Web policy languages nor by recent efforts towards reactivity on the Semantic Web.

1. INTRODUCTION

In order to specify how Semantic Web agents reason and make decisions under different conditions, different Semantic Web policy languages have been developed. These Semantic Web policy languages vary in terms of the reasoning mechanisms they use (e.g., Description Logics or Logic Programming) as well as in the expressivity they provide, therefore being able to address different scenarios. Unfortunately, none of them provide means to model the *reactive* behavior of agents in order to make decisions or perform tasks triggered by external events. For example, a policy-driven agent could automatically re-book a flight given that the schedule of a meeting has changed, or redirect an incoming VoIP call to a mobile phone in case the receptor is offline.

Recent efforts towards reactivity on the Semantic Web focus on the evolution of data and how systems must react to changes, possibly triggering the execution of some actions. Some of these approaches provide the basic mechanisms required for a general-purpose Semantic Web reasoner for reactive rules, but they lack specific features required in order to be applied to behavior control for automated agents: for example specific modelling of the interactions between agents and delegation or contextual disclosure of information (including the policies themselves).

In this paper we present a reactive Semantic Web policy framework that integrates both approaches, Semantic Web Policies and reactive systems, in order to exploit their advantages. This need for a broader notion of *Semantic Web Policies*, including reactive rules, has already been stated in [4]. The benefits of such a reactive behaviour control on the Semantic Web include separation between the application logic and the implementation, interoperability, compact representations (possibly to be exchanged among agents), verifiability, reasoning about agent behavior, reusability, and context sensitivity. Thanks to this integration, our framework allows agents to automatically perform advanced and powerful tasks, which can not be addressed alone neither by existing Semantic Web policy languages nor by recent efforts towards reactivity on the Semantic Web. A proof-of-concept prototype of our framework has been implemented and is freely available.

2. MOTIVATION AND REQUIREMENTS

Reactive behaviour control on the Semantic Web is a need that results from the flexibility of the Semantic Web: decisions have to be made by taking events into account and consequences of decisions have to be turned into real actions. As an illustrative example, we envision a (fictitious) system called SWYPE which is a very flexible voice-enabled instant messaging solution. It allows for automated adaptive behavior control: SWYPE behaves differently tailored to the user's needs based on knowledge available on the Semantic Web, based on events/notifications and also based on information exchanged between communication peers. Moreover, a user's SWYPE client can act as his Semantic Web broker: it may initiate on-line transactions (e.g., purchase goods), automatically send notifications, redirect communication flow (e.g., calls or chat messages) and so on.

A system like SWYPE has to meet several requirements which will be shortly[1] listed in the following. SWYPE requires each node to be both, a communication capable peer (e.g., chats and voice calls), and a Semantic Web node (URI addressable, eventually discoverable, and capable of describing itself on a semantic level). A SWYPE-client's behaviour has to be *programmable* (i.e., user-defined) in two ways: *declarative* to let the user allow to describe what has to be done (and not how), and *interoperable* to share concept definitions among different entities. A definition of behaviour should allow for complex *transactions* (e.g., purchasing) and *agent interactions* (e.g., negotiations) and has to be *reactive* at the same time in order to allow for the triggering of actions. Such a client should also allow for integration of *external systems* (e.g., PIMs[2] storing personal events). Finally, basic *security and privacy requirements* have to be met, too; in order to, for example, establish trust for an

[1]An extended technical report of this article is available at `www.L3S.de/~kaerger/reports/reactive_policies.pdf`.
[2]Personal Information Manager (e.g., Outlook)

agent communication. Suitable mechanisms for behaviour descriptions must also account for *explanations*: in case a SWYPE client behaves unexpectedly, an explanation needs to be provided to the user. Even in case of proper behavior, an adequate explanation may help the user to understand it better (and improve it, if needed).

Neither current Semantic Web policy frameworks nor reactivity frameworks alone are able to cater for all these listed requirements. Moreover, a careful comparison of current Semantic Web policy frameworks and reactivity frameworks reveals that the two fields are complementary. We argue, that both approaches together cover the full set of requirements: on the one hand, policy frameworks provide expressive and trustful condition languages, but they do not address reactivity requirements; on the other hand, reactive frameworks are very general, focus their expressiveness on reactivity issues and do not address other agent related concerns.

3. A FRAMEWORK FOR REACTIVE SEMANTIC WEB POLICIES

In this section we propose a solution that bridges reactivity and policies. Our approach is to combine a powerful policy framework, namely PROTUNE [2], with a Semantic Web reactive rules framework, in our case r^3 [1, 3].

PROTUNE (PRovisional TrUst NEgotiation framework)[3] aims at combining distributed trust management policies with provisional-style business rules and access-control related actions. The PROTUNE policy framework offers a high flexibility for specifying any kind of policy, integrate external systems at the policy level and provide facilities for increasing user awareness, like for example, explanations of policies and policy evaluation in natural language. It is entirely developed in Java what permits its integration into web environments as an applet (without requiring the installation of any additional software).

r^3 (Resourceful Reactive Rules)[4] is a Semantic Web framework to express reactive behavior with ECA rules (Event-Condition-Action rules). r^3 defines a foundational ontology that provides an abstract vocabulary for expressing ECA rules and the languages used in those rules. The common functionality shared by different r^3 nodes is abstracted using a Java library that includes a Servlet, Client and Core components which interact with abstract language engines (and markup translators), thus facilitating the integration of different languages (either by implementing new engines, or by wrapping existing ones).

r^3 and PROTUNE form the basis for our reactive Semantic Web policy framework. On the one hand, r^3 provides general reactive rule reasoning for Semantic Web systems, on the other hand, PROTUNE is a powerful policy language that allows to define Semantic Web systems' behaviors in a declarative way. The heterogeniety approach of r^3 allows to integrate any language making it a part of the reactive framework. In general, for an expression in any language, r^3 selects an available implementation for the language involved and submits the expression to the corresponding engine. This engine returns a set of results. The r^3 Java library abstracts this flow (and communication requirements) and further validates the submitted expressions according to a given language. Therefore, the task of implementing an r^3 engine for a given language is focused on providing (or wrapping) an implementation of that specific language (PROTUNE in our case). PROTUNE offers a Java API, and both r^3 and PROTUNE use logical variables for communication. Therefore, a wrapper providing r^3 with an implementation for the PROTUNE language was developed enabling r^3 rules to integrate PROTUNE capabilities, i.e. PROTUNE is exposed as an r^3 language. The resulting policy language is reactive: r^3 triggers the actions and takes care of binding variables across the borders of events, conditions, and actions. It passes the bindings to PROTUNE and performs, if needed, the defined actions. PROTUNE provides all the needed means for a powerful automated agent behavior control. Our implementation including some example policies is available on-line at `http://rewerse.net/I5/r3/TST/protune`.

Looking back to the requirements summarized in Section 2, with this framework it is easy to model a SWYPE-client's behaviour: PROTUNE is declarative since it is rule-based, and interoperable since it allows to refer to ontological defined concepts. Agent control allowing for negotiations, transactions, and agent interaction are also provided by PROTUNE. The integration of external systems (as it was required for, e.g., including PIMs) as well as security features are also part of PROTUNE. Reactivity needed for notification, reaction to personal events, etc. is provided by r^3. Finally, explanations of rules and decisions based on rules in natural language are provided by our framework as well, since it is able to combine r^3 variable bindings with PROTUNE's explanation facilities.

4. CONCLUSIONS AND FURTHER WORK

Semantic Web policy languages provide the means to specify statements for automated agent behavior control. Unfortunately, Semantic Web policy languages are not able to model reactivity, therefore allowing agents to react to changes in the environment nor to automatically trigger actions. On the other hand, existing reactive systems are general-purpose reasoners that do not meet many of the basic requirements needed to use them with agents. This paper presents how both approaches, being complementary, can be integrated in order to combine the advantages of both. In such a way, very complex and advanced scenarios can be addressed, therefore giving users expressive and powerful agents able to help in situations that were not addressable nowadays. Our integration of the frameworks r^3 and PROTUNE has been developed and is beeing tested. We are currently working on a prototype application in order to demonstrate some of the scenarios introduced here. In particular, we are extending some current messaging and VoIP applications to allow for automated behavior control. We also plan to develop appropriate tools to specify such reactive policies as well as to visualize and monitor them.

[3] `http://www.L3S.de/web/PROTUNE`

[4] `http://rewerse.net/I5/r3/`

5. REFERENCES

[1] J. J. Alferes and R. Amador. r^3- A foundational ontology for reactive rules. In *ODBASE'07, LNCS 4803*. Springer.

[2] P. Bonatti and D. Olmedilla. Driving and monitoring provisional trust negotiation with metapolicies. In *POLICY'05*. IEEE.

[3] W. May, J. J. Alferes, and R. Amador. Active rules in the semantic web: Dealing with language heterogeneity. In *RuleML'05, LNCS 3791*. Springer.

[4] P. A. Bonatti, et al. Semantic web policies - a discussion of requirements and research issues. In *ESWC'06*. Springer.