Exploring new Frontiers of Social Computing with AThOS and ExceptionOWL

Matteo Baldoni, Cristina Baroglio, Roberto Micalizio, Gian Luca Pozzato

Università degli Studi di Torino — Dipartimento di Informatica
c.so Svizzera 185, I-10149 Torino (Italy)
{firstname.lastname}@unito.it

Abstract. We present the main ideas underlieing two projects, funded by Università degli Studi di Torino and Fondazione San Paolo: *Accountable Trustworthy Organizations and Systems* (AThOS), and *Nonmonotonic Extensions of Description Logics and OWL for defeasible inheritance with exceptions* (ExceptionOWL).

1 AThOS

Many private and public enterprises can nowadays be seen as Sociotechnical Systems (STSs): complex systems in which human beings interact by means of ICT infrastructures. Two main features characterize STSs: (1) human beings are essential to deliver the business goals of the enterprise; (2) humans and software components acting on behalf of humans are just partially predictable actors since (may) enjoy some form of autonomous behavior.

Because of the autonomy that characterizes the actors, *accountability* becomes a critical and predominant requirement that the ICT infrastructure has to support. Roughly speaking, accountability encompasses the tracking down of responsibilities for actions and decisions taken within the organization. Despite its importance, accountability has so far not received the attention it deserves, but most of the research effort has been put on the design and the management of the processes (i.e., workflows) taking place within an enterprise, whereby the rise of many Business Process Management (BPM) techniques.

AThOS aims at formalizing and developing a novel methodology for enforcing accountability in STSs. Specifically, when some activity fails in achieving its expected goals, an accountable enterprise is committed to understand what went wrong and why, and to find a proper course of repairing actions. Although the current workflow engines support the on-line monitoring of the business processes, such a monitoring is basically a reactive mechanism that, once recognized a particular condition, triggers a precompiled solution (e.g., issue alarms, send emails, etc.). However, this reactive mechanism fails in identifying the agents that were responsible for the unwanted conditions.

AThOS provides a completely new way of conceiving STSs, in which the perspective is reversed: the main element of the model is no longer a workflow of activities, assigned to different actors, but the *actors* and the *interactions* arising among them become the first-class entities around which an STS is modeled. The rationale is to promote the *autonomy of the actors*, which is considered as a value since actors have the capability of reasoning about their context, and take advantage of unexpected occasions or react to unexpected adversities. This is achieved by simply stating, in a declarative manner, the norms actors must abide by (what they can/cannot do), without imposing unnecessary constraints on their activities. This does not mean that workflows are no longer included within the system model, but rather that we need novel and flexible representations, where actors are left free to decide about the best way for achieving their own goals, and where such freedom is balanced by the fact that actors will be *liable for their decisions* and for the actions they have taken. In conclusion, AthOS aims at: (1) defining of a modeling framework for STS, centered around the notions of interaction and of social relationship between actors (social relationships will capture links as *commitments*, delegations, etc.); (2) developing a *new engine*, that enterprises can use to monitor on-going interactions and to enforce accountability.

2 ExceptionOWL

The Semantic Web is as an extension of the World Wide Web (WWW) that allows computers or agents to manipulate, search, integrate Web contents referring to the meaning that such content has to human beings. To this aim, Semantic Web languages allow to express the semantics of data in a machine-understandable syntax by means of metadata. Ontologies, intended as descriptions of knowledge about a domain of interest for a specific application, have become a key concept for the Semantic Web technologies.

One of the main limits of the existing languages for ontology engineering is that of not supporting the formalization of exceptions as well as related prototypical reasoning in taxonomies. The need of representing properties holding only for typical members of a given class, and not for all such members, emerges in several domain applications. This limit of the actually available formalisms for building ontologies is explained by the fact that the logical formalisms underlying these languages, known as the *Description Logics*, are monotonic.

Description Logics (for short, DLs) are reminiscent of the early semantic networks and of frame-based systems. They have found a great success in several applications and they have been used as the base of the languages of the Semantic Web such as the Web Ontology Language OWL. The OWL DL variant of OWL is the language usually adopted for practical applications in several research fields, just to mention a few: biology, medicine, geography, astronomy, agriculture, and defense.

Recently, a renewed interest in the definition of nonmonotonic extensions of DLs has been spurred by the research in ontologies and Semantic Web. This extensive practical experience has confirmed the benefits of using DLs in knowledge representation, but it has also highlighted the mentioned limitations, and finding a suitable nonmonotonic extension for inheritance reasoning with exceptions seems far from being solved. The main aim of ExceptionOWL is to extend DLs and OWL in order to reason about defeasible inheritance with exceptions in ontologies, by proposing a new approach which is technically well-founded, motivated both from the knowledge representation and philosophy points of view, computationally effective and usable in the context of practical applications. More in detail, ExceptionOWL intends to explore an innovative strategy, based on the combination of (i) a *typicality operator* $\mathbf{T}$, extending the language of standard Description Logics in order to express *typical* properties of concepts (given a concept C, $\mathbf{T}(C)$ singles out the instances of C that are typical) and (ii) the well established nonmonotonic mechanism of *rational closure*.

ExceptionOWL will also examine techniques for mining structured information from the social media ecosphere. This data comes usually in the form of unstructured information (tweets, posts, comments), that lack of machine-understandable metadata associated to them. ExceptionOWL will try to fill this gap and it will refer to the extensions of DLs and OWL with exceptions for the formalization of the data extracted from social media outlets. The developed techniques will be applied to the medical domain, where they will produce a structured knowledge base about conditions, symptoms, mental states, and opinions of patients that will be the main input of a new generation of medical expert systems based on ontologies with exceptions.