Multiagent and Grid Systems – An International Journal 11 (2015) 121–146
DOI 10.3233/MGS-150233
IOS Press

Automatic policy enforcement on semantic social data

Truc-Vien T. Nguyen*, Nicoletta Fornara and Fabio Marfia
Institute for Communication Technologies, Faculty of Communication Sciences, Università della Svizzera italiana, Lugano, Switzerland

Abstract. Web-based data collection of non-reactive data is becoming increasingly important for many social science fields. Being able to introduce and automatically enforce policies that regulate the collection and the use of those data is crucial for taking into account the privacy and confidentiality wishes of data providers. Those policies are currently expressed in natural language or in a language lacking a formal semantics, therefore, in order to comply with them researchers need to apply them manually. It is therefore clear that realizing new technologies for automatically supporting the enforcement and monitoring of various type of policies in Web-based data collection is an important open issue. In this paper, we propose an application independent formal model of policies that can be automatically enforced and whose specification allows to take into account the meaning of the managed data. Given that for enforcing the policies on the collected data, their semantics is crucial, in this paper we propose an OWL 2 Social Network Ontology to capture the nature of social network data collected from the Web. We also propose techniques for semantic enrichment of the data, which are then exploited in the specification and enforcement of the formalized policies.

Keywords: Policy, norm, obligations, semantic web technologies, social network, natural language processing, OWL, topic detection

1. Introduction

Web-based data collection of non-reactive data (that is data collection for social research, where studied people are not aware of it) is becoming increasingly important for many social science fields like sociology, market research, health studies, psychology, and communication sciences. Web-based data collection is not restricted to Web surveys, but it also includes non-reactive data collected by means of log file analysis, data mining, text mining, and data crawling from heterogeneous web sources (i.e. blogs, social networks, consumer reviews, folksonomies, and search results). Being able to introduce and automatically enforce policies that regulate the collection and the use of those data is crucial for taking into account the privacy and confidentiality wishes of data providers [26].

Those policies may be ethical guidelines, like for example the Code of Professional Ethics and Practices proposed by the American Association for Public Opinion Research;[1] legal constrains on the processing of personal data, like the European Union Directive 95/46/EC of the European Parliament and

*Corresponding author: Truc-Vien T. Nguyen, Institute for Communication Technologies, Faculty of Communication Sciences, Università della Svizzera italiana, via G. Buffi 13, Lugano 6900, Switzerland. E-mail: thi.truc.vien.nguyen@usi.ch.

[1]Available at http://www.aapor.org/AM/Template.cfm?Section=AAPOR_Code_of_Ethics&Template=/CM/ContentDisplay.cfm&ContentID=4248.

ISSN 1574-1702/15/$35.00 © 2015 – IOS Press and the authors. All rights reserved

of the Council of 24 October 1995;[2] policies on how the resources available on a social software can be used for automatic data collection, like for example the Automated Data Collection Terms specified by Facebook,[3] and rules on what pages of a web site can be crawled by a web robots, expressed for example with the `robots.txt` language.[4]

Even if such policies are not all enforced at this moment by the data publishers, and thus some of them can be ignored by data collection software, their fulfilment is crucial for social scientists wanting to use the available data correctly and follow an ethics in Internet Research [7].

Those policies are mainly expressed in natural language (e.g. English), or in a language lacking a formal semantics, and they come from different sources usually (the producer of the data, the publisher of the data, the community where those data will be used). Therefore, in order to comply with them, researchers need to read, understand, and finally apply them. Being compliant with those norms becomes very difficult when a big amount of data is treated for automatic extraction by means of specialized software (i.e. both site-specific and generic crawlers). Moreover, some policies/norms[5] of those express obligations ("we shall") and prohibitions ("we shall not", "shall prohibit") on how data can be collected, stored, used, disclosed, and so on. It is therefore clear that realizing new challenging technologies for supporting the automatic enforcement of various type of policies in Web-based data collection is an important open issue.

In this paper, we propose an application-independent formal model of policies, whose specification allows taking into account the meaning of the managed data. Those policies are mainly characterized by an *activation condition* and a *content*. Thanks to the formal specification of those policies by using Semantic Web Technologies [14] and to the formal representation of the meaning of the collected data by using the standard Web Ontology Language OWL 2,[6] those policies can be automatically enforced by using a software, together with one of the available OWL reasoners. Policies are enforced by first detecting if they are active and then by complying with the active ones.

This is an innovative approach indeed. It appears that there are no studies or tools, from our analysis of the state of the art, for collecting non-reactive data from the Web that take into account a formal specification and enforcement of policies regulating their usage. Moreover, the proposed application-independent model of policies, which is based on representing the semantics of the data by using Semantic Web Technologies, is novel.

In fact, as discussed in Section 2, in Multiagent Systems literature there are interesting studies on norms/policies representation, enforcement, monitoring, analysis, and deconflict, which are focussed on regulating generic actions of autonomous agents. Norms/policies can be used to regulate electronic institutions, electronic commerce, and to deal with security issues [25]. Among those studies, the most relevant for this paper are the one that use Semantic Web Technologies for norms formalizations and enforcement [10,12,30,33], but none of them is focussed on regulating the performance of actions on big amount of data.

Other relevant studies are in the area of Web Access Control, where access control policies are used to regulate the access to resources and data available on the Web [1,3,9,28], but they mainly take into

[2] Available at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.

[3] Available at http://www.facebook.com/apps/site_scraping_tos_terms.php.

[4] http://www.robotstxt.org/robotstxt.html.

[5] Taking into account the American tradition of using the term policies for expressing obligations, prohibitions and permissions and the European tradition of calling them norms, in this paper we use those two terms as synonymous.

[6] The standard Web Ontology Language http://www.w3.org/TR/owl2-overview/ which is a practical realization of a Description Logic (DL).

account only the roles (or, more generally, the attributes) of the actors involved, or the context from which a user asks to get access to certain data. However, they are intrinsically unable to consider the meaning of the data accessed. That because a rich semantic representation of the data meaning is not available very often. Therefore, another crucial aspect of this paper is that we propose some techniques for enriching the data extracted automatically from the Web with a formal semantics. We store them in an OWL 2 ontology, and, thanks to the use of Natural Language Processing Techniques (for automatic extraction and representation of knowledge from semi-structured and unstructured data), we enrich them with the semantics required for expressing certain policies.

In particular, we test and evaluate the proposed model of policies and the system (described in Section 3) required for collecting the data, performing the semantic analysis, and automatically enforcing the policies, by using the following use case. We extracted a reasonable amount of data automatically from Facebook and Twitter, then we manipulated and stored them in an OWL Social Network ontology, as described in Section 4. Therefore, we enriched the collected data semantically with named entity recognition, disambiguation, and topic detection techniques as described in Section 5, focussing on the detection of the information required for the formalization and enforcement of certain policies. In Section 6 we present an application-independent formal model of norms/policies able to take into account the semantics of the managed data and we present the Policy Enforcement Service a component able to enforce the formal specification of those policies. We also exemplify how to use the proposed model by formalizing and enforcing three obligations coming from EU Directive 95/46/EC. Section 7 presents the experiments we perform for investigating the applicability of the proposed approach. Finally, some conclusions are drawn in Section 8.

2. Literature review

The work described in this paper is related to different areas of research: (1) Web-based data collection of non-reactive data; (2) an area of multiagent systems, called normative multiagent systems (NorMAS) (3) the area of Web Access Control; and (4) the area of the Semantic Web that studies automatic extraction and representation of knowledge from semi-structured and unstructured web data by using Semantic Web technologies.

2.1. Web-based data collection of non-reactive data

Regarding the first area, to the best of our knowledge and on the basis of the MoU of the WEB-DATANET COST Action [26], there are no studies or tools nowadays for automatically collecting non-reactive data from the Web that combine formal models of norms and policies representation and enforcement, and techniques for automatic extraction and representation of knowledge from semi-structured and unstructured data, with the goal of being compliant with existing norms or policies.

2.2. Normative Multiagent Systems

There are various proposals in the literature on Normative Multiagent Systems (NorMAS) for the declarative specification of norms [6,11] and policies [30,31] and of frameworks for their management, enforcement, and monitoring using different languages [12,32]. The choice of using formal declarative languages (like logics or logic programming languages) has many important advantages, because it makes possible:

- To represent the policies/norms as data, instead of coding them into the software, with the advantage of making possible to add, remove, or change the policies/norms both when the system is off line, and at run-time, without the need to reprogram some components of the system used by the agents for interacting or of the software agents whose activities are regulated by the policies/norms;
- To realize an application-independent monitoring component able to keep track of the state of policies/norms on the basis of the events that happen in the system and of the agents' actions (this mechanism can also be able to react to norm fulfilments or violations);
- To develop agents able to reason and plan their actions by taking into considerations the correlations between their goals and external social constrains expressed also in terms of policies/norms, this by reasoning on what norms apply in a given situation, what activities are obligatory, permitted or prohibited, using for example some form of what-if reasoning [33] for deciding whether or not to comply;
- To develop agents able to interact within different systems without the need of being reprogrammed.

The choice of the formal language used for the declarative specification of policies/norms is difficult because many aspects have to be taken into account. The most important are: the expressivity of the language, its computational complexity, the fact that the underlying logic is decidable, the diffusion of the language among software practitioners and research communities, its feasibility for fast prototyping, and its adoption as an international standard. We believe that Semantic Web Technologies [14] may be successfully adopted for representing and managing in an efficient and effective way policies/norms for open interaction systems running on the Internet [9]. In fact, Semantic Web technologies are increasingly becoming a well known standard for Internet application and therefore are supported by many tools. For example, thanks to the fact that the OWL 2 language is decidable and it is an international W3C standard, it is supported by many efficient reasoners (like Fact++, Pellet, Racer Pro, HermiT), tools for ontology editing (like Protégé), and libraries for automatic ontology management (like OWL-API and JENA). Moreover, considering that policies/norms may be used for regulating different domains of application, an important advantage of using Semantic Web Technologies is the possibility of promoting re-use of existing ontologies and the integration of norms/policies coming from difference sources.

In particular, we think that the Semantic Web Technology that may be relevant for the specification of norms/policies on semantic data extracted from the web is the Web Ontology Language (in its OWL 2[7] version), the description logic language recommended by W3C for the definition of the ontologies of Semantic Web applications. The advantage of choosing this ontology language instead of the RDF Schema[8] ontology language is that, using OWL, it is possible to define a class as equivalent to an axiom which uses a rich set of logical operators (for example property restrictions, intersection, union, and complement of classes, and property chain to mention few) and therefore it is possible to better exploit the reasoning possibilities of OWL reasoners for deducing new knowledge on the data, which will result fundamental for the enforcement of complex policies/norms.

Taking into account those considerations, we will discuss the state of the art on norms and policies modelling and reasoning, focusing on approaches that use Semantic Web technologies.

The policy formalization proposed in this paper is strictly related to the one presented in [10,13], where the activation conditions (when a norm becomes active and therefore some agents start to be subject to the norm) and the content (the regulated action) of norms are expressed using OWL 2 classes that are defined using OWL 2 axioms. In particular, an OWL ontology of obligations, whose content is a class of

[7] The W3C standard language for ontology specification http://www.w3.org/TR/owl2-syntax/.
[8] http://www.w3.org/TR/rdf-schema/.

possible actions that have to be performed within a given deadline, is presented in [10]. The monitoring of those obligations (checking if they are fulfilled of violated on the basis of the actions of the agents) can be realized thanks to a specific framework required for managing the elapsing of time and to perform closed-world reasoning on certain classes. A similar ontological formalization of obligations has been also extended for being used in a wider OWL 2 model of artificial institutions instantiated at run-time by dynamically creating spaces of interaction [13].

We present an adaptation of such a model in this paper, in order to be able to express policies, and in particular obligations, on how some data extracted from the Web should be manipulated before using them for Internet Research [8]. In particular, in the model presented in Section 6, we express the *activation conditions* of policies by using OWL 2 axioms, which are written by using the classes and the properties of the Semantic Network Ontology used for representing the data extracted from the Web. Those activation conditions may be tested (in order to check if they are satisfied or not, and the corresponding obligations are active or not) by using the reasoning services of one OWL reasoner for retrieving all the individuals that belong to the activation condition class. If an obligation is active, it is possible to fulfill it by performing the obliged action, that is an action that must be performed usually on the data in order to filter them or to manipulate them. Given that it is not possible to describe an action in OWL operationally, we assume that this action belongs to a library of possible actions, whose parameters may be described using OWL classes.

Another interesting approach that uses Semantic Web Technologies for policies formalization and reasoning for regulating the behaviour of autonomous agents is the OWL-POLAR framework [30]. The framework investigates the possibility of using OWL ontologies for representing the state of the interaction among agents and SPARQL queries for reasoning on policies/norms activation, for anticipating possible conflicts among policies, and for conflicts avoidance and resolution. One important difference between such a proposal and the work proposed in this paper is that, in the OWL-POLAR framework, the activation condition and the content (what is prohibited, permitted or obliged) of policies are represented using *conjunctive semantic formulas*: the conjunction of atomic assertions expressed using the concepts (classes and relations) from an OWL-DL ontology on a vector of variables. In order to evaluate those formulaes they must be converted into SPARQL queries that are executed on the OWL ontology used to describe the state of the interaction. The main drawbacks of the OWL-POLAR approach are: (i) the expressivity of the formal language chosen: it is not possible to express in conjunctive semantic formulas the same conditions that it is possible to express with OWL axioms; (ii) the translation into SPARQL queries may slower the process of evaluation of the policies. An important contribution of the OWL-POLAR framework is the study of norm conflicts and of conflict avoidance and resolution. A future study on the management of conflicts among norms coming from different sources may depart from that work.

Another proposal, whose field of application is the regulation of the behaviour of autonomous agents, and where Semantic Web technologies are used for norm/policy specification and management, is the one proposed for the formalization of the well known policy management framework KAoS [2,33]. The KAoS framework is composed by three layers: (i) the human interface layer where policies, expressing authorizations and obligations, are specified in the form of constrained English sentences; (ii) the policy management layer used for encoding in OWL the policy-related information; (iii) the policy monitoring and enforcing layer used to compile OWL policies to an efficient format usable for monitoring and enforcement. KAoS policies have the following basic form: "*[Actor] is [constrained] to perform [controlled action] which has [any attributes]*". Therefore it is possible to express activation conditions on the actor constrained by the policy and on some attributes of the controlled action including some

aspects of its context. The main difference between the norms/policies model proposed in this paper and the one proposed in the KAoS framework is the field of application and the fact that we contextualize the regulated actions by defining special OWL classes and by defining a set of basic operations.

Semantic Web technologies are also partially used in the Rei policy language [18], for modelling the deontic concepts of rights, prohibitions, obligations and dispensations. Differently from the approach proposed in this paper Rei is implemented in the logic language Prolog, but it also includes some ontologies that enable the policy engine to interpret a subset of RDFS policies.

2.3. Web access control

Another relevant field of research where interesting models of policies and frameworks have been proposed, by using Semantic Web Technologies, is the field of *Web Access Control*. Even if the field of application of those approaches (regulating the access to data available on the Web of Data[9]) is different from the one studied in this paper, it is worth to compare them with the work presented in this paper.

In [28], the authors propose an extended version of the Web Access Control (WAC[10]), which consists in a complete framework for expressing and managing read, write, or control privileges in the Web of Data. They introduce a Privacy Preference Ontology (PPO) for Linked Data; as Linked Data are represented using RDF triples, PPO provides the privileges to restrict access in more complex situations, such as access to an RDF statement, an RDF graph, or an RDF resource. The PPO allows one to express access test queries for defining who is granted access to a given part of the data by means of SPARQL ASK queries use to specify which attributes or properties the user must satisfy. Such access test queries are executed by the Privacy Preference Manager on the FOAF profile of the requester, to check if a given user has or has not access to a given part of the data. This model assumes a private-by-default policy, that is, if a specific resource has no privacy preference, it cannot be accessed.

While the work proposed in [28] is focused on access control policies for RDF documents, in [3] the focus is on authorization mechanism for RDF graph stores; that is, on the definition of a fine-grained access control model for named graphs accessible by means of a SPARQL 1.1 endpoint. Since they aim at mobile devices their model of access control is integrated with a model of the mobile context of the users. The model of the policies is described in the S4AC vocabulary reuses concepts from other ontologies like SKOS and WAC. The general idea of the proposed framework is to protect RDF stores by changing the SPARQL queries issued by a user by adding some binding clauses to restrict their scope to the triples included in accessible named graphs only. The accessible name graphs are obtained evaluating pre-defined access conditions, which, similarly to the proposal presented in in [28], are expressed as SPARQL 1.1 ASK queries on the context of the requester instead of on its FOAF profile.

An important difference between those two proposals and the one described in this paper is the choice of the language for expressing the data and the access (or activation) conditions of the policies. In fact, the reasoning functionalities on RDF data, RDF Schema, and SPARQL queries are limited with respect to those on OWL ontologies. Another difference is the fact that the policies presented in those work are all focused on expressing permissions for accessing to certain data, whereas we focused on the formalization of obligations for transforming the data before using them, or sharing them with other organizations. Finally, the activation conditions of the model of obligations presented in this paper are evaluated on the content of semantic data formalized in OWL, differently, in [28], the access test queries

[9] http://www.w3.org/2013/data/.

[10] http://www.w3.org/wiki/WebAccessControl.

are checked on the FOAF profile of the requester and, in [3], the access conditions are evaluated on the context of the requester which is expressed in RDF. An advantage of the approaches described in [3,28] is that it is possible to evaluate them with experimental tests by using existing big RDF stores, SPARQL 1.1 query engines, and SPARQL benchmark datasets. An interesting future work will consist in proposing a model of policies/norms for integrating all those types of conditions together with the study of mechanisms for preventing the insertion of conflict policies.

In [19] a generic Social Network Ontology is presented (as discussed in Section 4) together with a rule-based access control policy model where rules are expressed with the Semantic Web Rule Language SWRL.[11] The idea of integrating OWL with rules [14] is very interesting because it makes possible to express much more conditions on which data are accessible or not. That because there are some interesting DL-safe rules that cannot be expressed as description logic (OWL) axioms and do not endanger decidability. This aspect is not discussed in the paper and may represent an interesting starting point for future improvement of the model presented in this paper. Similarly to the two previously cited proposals, the approach presented in [19] does not take into consideration the nature and semantics of the data, regulating the access according to information metadata as owner, author and user relations.

An important general-purpose XML specification for expressing access control policies in distributed contexts, that is considered the *de facto* standard, nowadays, is XACML.[12] XACML implementations do not make use of Semantic Web Technologies usually, making such frameworks poorly related to our work. However, Kolovski et al. [17] propose techniques for translating XACML policies into DL axioms. Their work is aimed at modeling a generic set of XACML policies using DL, in order to easily apply tasks of policy harmonization. Their approach covers a wide range of XACML expressivity. While their work's primary objective is the use of DL for access control in distributed system, our focus is on the possibility for researchers to abide by the different norms and laws automatically using DL-formalized policies. Semantic formalization of policies is useful in access control environments for complex tasks as policy harmonization and policy explanation, while the expression of obligations and permission in non-reactive data collection deals with the identification of specific pieces of data where to enforce the formalized policies.

Finally, it is worth mentioning one of the few existing works where access permissions are formalized for regulating access to data represented in the ABox and TBox of an OWL ontology [1]. Thanks to that choice, it is the only approach that makes it possible to exploit one of the available OWL reasoners for deducing if a given permission is or is not granted.

2.4. Semantic analysis

In the field of research related to semantic extraction from semi-structured and unstructured data, we developed our systems following the literature in natural language processing. The tasks we need to solve are: 1) Detection of private data, which can be divided into two subtasks: Named Entity Recognition (NER) and Disambiguation to Wikipedia (D2W); 2) Detection of sensive topic, which can be translated as the task Topic Detection (TD). Unlike previous works on these three tasks, in our work, the application must be carried on the Facebook data, which often contains various errors. For instance, the text from Facebook statuses/comments is often noisy – contains spelling errors, abbreviations, non-standard words, false starts, repetitions, etc. Therefore, it requires special computational techniques.

[11] http://www.w3.org/Submission/SWRL/.

[12] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

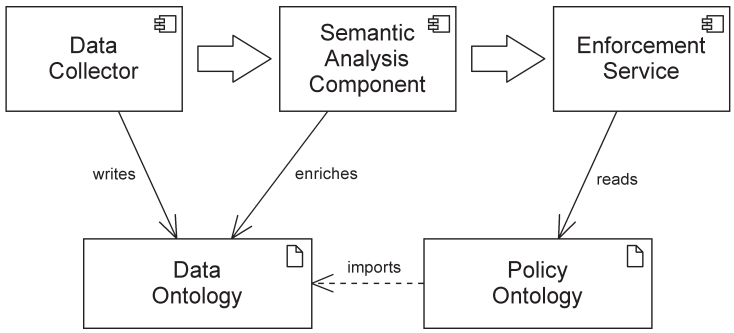

Fig. 1. Main components of the system.

We decided to develop our own systems for these three tasks. Specifically, to identify private data in Facebook statuses, we made use of our previous works presented in [23,24] and of a new method presented in [22]. In the former, we developed a system which recognizes named entities in text [23] and associate them with Wikipedia entries [24]. Recently we proposed the system in [22], which solves the *D2W* problem in a language-independent manner. Thus, our *D2W* system can be applied in any language available in Wikipedia, as opposed to one single language observed in all previous works [5,20,21,27]. Up to now, we have deployed it in four European languages: English, French, Italian, and Polish without any difficulty.

In order to detect if a Facebook post/comment is related to sensitive topics such as politics, religion, ethnicity, we implemented a vector space model with *tf-idf* as the weighting scheme. Following previous works, we consider a topic as a set of related terms, and the content of a post/comment as a document - set of words. Considering the *set of terms* and *set of words* as vectors, we computed the cosine similarity between the term vector and a document vector. If the similarity exceeds some threshold, the text is then associated to that topic. Moreover, with this approach, by exploiting Wikipedia as big data resource, we devised a new method to improve the accuracy of topic detection with semantic knowledge extracted from Wikipedia and injected into the traditional *tf-idf* model. In all three tasks *NER*, *D2W*, and *TD*, our systems reach state-of-the-art performance.

3. General system architecture

In this section, we describe the general architecture of the system that we realize for acquiring non-reactive data from the Web, while taking into account a formal specification of policies regulating the usage of such data, in its most general form. As it is shown in Fig. 1, we identified three main components:

1. The *Data Collector*: it is the component that deals with the acquisition of the data from the Web. It identifies information, collects it, and organizes it semantically, generating an ontology, the *Data Ontology*, from the original data.
2. The *Semantic Analysis Component*: it is the component that, using NLP Techniques, enriches the ontology with new semantic information about unstructured text. The added semantic information is eventually useful for inferring whether a policy has to be applied or not on a piece of data.
3. The *Enforcement service*: it is the service that evaluates if the formalized policies (stored in the *Policy Ontology*) are active and enforces them. It uses the reasoning service of an OWL reasoner in order to identify the pieces of data on which the policies apply.

The *Data Ontology* represents the ontological collection of acquired data. It is organized semantically thanks to an *a priori* specification of concepts and properties formalized in an OWL *TBox*. The *Policy Ontology* contains the concepts and the properties required for the formal specification of the policies and the collection of formalized policies. Given that the formalized policies are related to the semantics of the collected data, the *Policy Ontology* imports the *Data Ontology*. An analogous framework is proposed in [32]. While our focus is more on the data acquisition and semantic analysis, taking into account specific modules for data collection and semantic enrichment, they depict additional, dedicated modules for specific tasks as, e.g., policy mapping, policy storage policy distribution to different entities in a distributed system, that we do not present at this state of the work.

As mentioned in Section 1, we analyze a specific use case of acquisition and enforcement of policies in this paper. It consists in the collection of social network data, while taking into account some policies of the EU Directive 95/46/EC, stating the necessity of anonymization of personal data and of data revealing confidential information on people.

In such a use case, the *Data Collector* is represented by the algorithms for acquiring data from the Facebook and Twitter social networks, as presented in Section 4. The *Data Ontology* consists in a general ontology for capturing the main concepts and properties involved in a social network environment, the Social Network Ontology presented in Section 4. The *Semantic Analysis Component* is represented by the named entity recognition, disambiguation and topic detection techniques for enriching the acquired data, as described in Section 5. The *Enforcement Service* is the application independent component in charge of checking the activation and enforcing the cited, formalized policies coming from the EU Directive on the collected data, as described in Section 6.

4. The Social Network (SN) ontology

Privacy becomes more and more an important topic surrounding social media systems when web data proliferates. Depending on the kind of the data, setting up privacy preferences is a difficult task to deal with, and can lead to a lot of confusion. For example, if someone shares a picture and tags his/her friends in it, each of the tagged people can contribute additional policy constraints that can narrow access to it. A more refined taxonomy is very essential in open context like social networks, and could help us in moving toward better privacy controls for online social media systems, in line with the development of Semantic Web technologies and languages.

We present the formalization of the *TBox* of the OWL 2 ontology in this section, that can be used to capture the classes and the properties of a generic social network. As we are aware, there are different ontologies representing a general conceptualization of a Social Network (SN) System in literature. The BBC Core Concepts Ontology[13] represents entities such as people, places, organizations in the context of BBC broadcasting network. The SIOC (Semantically-Interlinked Online Communities) Ontology[14] provides the main concepts and properties required to describe information from different types of online communities, included SNs. Masoumzadeh and Joshi present in [19] an ontological representation of a SN, in order to provide complex access control for the contents, using OWL reasoning. While the first and the second ontologies represent too simple and general conceptualizations according to our purposes, the third presents an interesting portrait of social media with users, messages, categories, relationships

[13] http://www.bbc.co.uk/ontologies/coreconcepts.
[14] http://sioc-project.org/ontology.

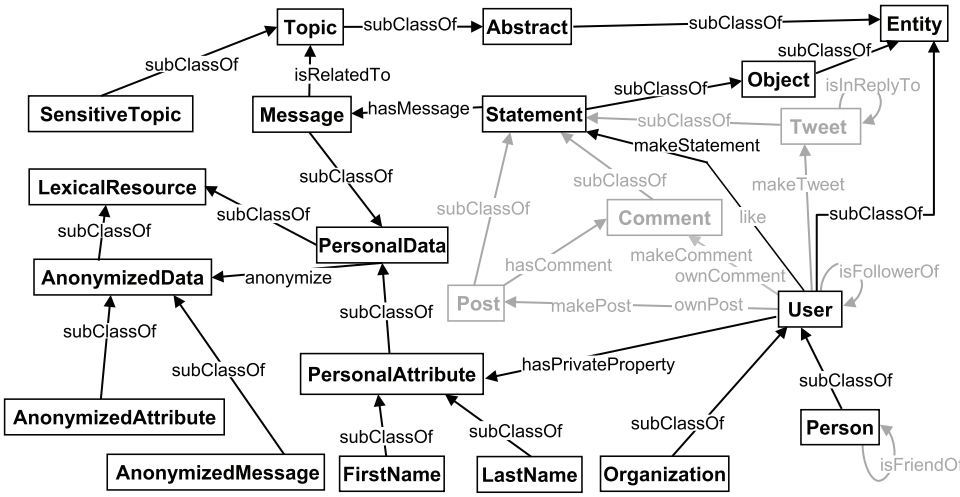

Fig. 2. Social Network (SN) ontology main classes and properties. General SN concepts are drawn in black, while concepts that are specific of a single SN (Facebook or Twitter) are drawn in grey.

and interactions between users. We started from that last representation for generating our own ontology, we named it SN Ontology. Its main classes and properties are shown in Fig. 2 and it is made available on-line.[15] The rationale behind it is to provide a minimal but sufficient ontology suitable for social networking environments.

There are two main branches in the SN Ontology: Entity $\rightarrow$ Object and LexicalResource $\rightarrow$ PersonalData. The first branch characterizes users, their statuses and interactions in the social network. A user can be a person, a location, or an organization (a community is considered as an organization). The second branch describes user's properties, such as first name, last name, or created messages. User's properties are represented as binary relations. For instance, the hasPrivateProperty property points to a generic personal attribute, as first name or last name; MakeStatement indicates that a person is the author of a statement. However, the main difference between our ontology and the one presented in [19] is the presence of the AnonymizedData class, representing data that are subjected to a process of anonymization, according to specific conditions, as presented in Section 6.

While classes and properties in black in Fig. 2 are general concepts suitable for every SN, concepts in grey are, instead, specific of a single SN and gain meaning only in that specific context.

Regarding the Facebook SN, it can be noticed that there are posts and photo updates that allow users to discuss their thoughts, whereabouts, or important information with their friends. A Post is created by a user on his/her own wall or the wall of a friend, and it may include any kind of content such as shared links, checkins, photos and status updates. The author of the post is the user who *makes* the post, while the owner of the wall that contains the post is considered the one who *owns* the post in our SN ontology. We use the term "post" to represent all kinds of updates, including status, photos and sharing updates. So, a connection with a makePost property between a User and a Post identifies the user as the author of the post. A connection with a ownPost property between a User and a Post identifies the user as the owner of the wall on which the post is published.

[15] http://www.people.usi.ch/fornaran/ontology/SemanticNetworkOntology.zip.

Regarding the Twitter SN, messages are published in the form of Tweets, that are max-140-characters texts that can include links to pictures, videos or other content. A Tweet can be written in reply to another Tweet (a relation represented by the property inReplyTo). A user can be the follower of another user (isFollowerOf property). That means that the Tweets written by the latter user are shown in the personal Twitter Home page of the former one.

The procedure for translating network data into an OWL 2 ontology, which has as TBox the SN Ontology, is illustrated in the next section.

5. Data collection and semantic analysis components

We describe the framework for collecting data from an online social network, for translating those data into an OWL 2 knowledge base, and for enriching the data with semantic knowledge in this section. After this process, the knowledge base is then modified to be compliant with a set of data protection policies as presented in Section 6.

5.1. Data collection procedure

We collected the data from Facebook and Twitter, two popular social network with a huge amount of users. The reason for choosing Facebook and Twitter is that they are two between the most popular social networking sites. Facebook reached 1.11 billion of active users in the world as of March 2013,[16] while Twitter reached 204 million of active users in the same period.[17]

Regarding the Facebook SN, we used RestFB[18] (a Facebook Graph API[19] written in Java) to collect users' information, statuses, posts and photos. The Graph API is the primary way in which data is retrieved from or posted to Facebook. It is a low-level HTTP-based API that you can use to query data, post new stories, upload photos and many other tasks. In order to access to Facebook data via the API, the system needs to obtain an access token which provides temporary, secure access to Facebook. With the access token, the system is then able to retrieve Facebook data in terms of different fields: personal information, statuses, posts and photos.

The collection procedure is illustrated in Algorithm 1. Starting from a set of "seed" users $\mathscr{U}$, we collected their public information, statuses, posts and photos. Then $\mathscr{U}$ is repeatedly expanded until we get some large amount of data. The expansion is realized as follow. For each user u in $\mathscr{U}$, we collect u's public statuses/posts/photos. For each status/post/photo p, we take the set of people who like p, the comments made in p, people who wrote or who liked those comments. Then we obtain a list of "new" users to add to $\mathscr{U}$. The collection and expansion are repeated until we get an amount of data, which, is defined by a threshold.

The procedure for translating Facebook SN data into an OWL 2 ontology, which has as TBox the SN Ontology, is illustrated in Algorithm 2. It is based on the idea of following the progress of how users interact with their network: (1) a user creates statuses, posts photos, shares links or posts information on his/her friend's wall; (2) next, his/her friends start liking the posts, commenting on them, and sharing them on their own wall. From this progress, two main steps should be taken in the algorithm: (1) create individuals in the ontology corresponding to person, posts, comments; (2) create binary relations between users/posts/comments.

[16] http://investor.fb.com/releasedetail.cfm?ReleaseID=761090.
[17] https://about.twitter.com/company.
[18] http://restfb.com/.
[19] https://developers.facebook.com/docs/graph-api.

Algorithm 1: FACEBOOK_SOCIAL_NETWORK_DATA_COLLECTION()

$\mathscr{U}$ = *set of seed users*
$\mathscr{DS} = \emptyset$
$\mathscr{PS} = \emptyset$
repeat
  $\mathscr{UT} = \emptyset$
  for each $\langle user : u \rangle \in \mathscr{U}$
    do
      $\mathscr{D} \leftarrow$ *personal information from* u
      $\mathscr{P} \leftarrow$ *set of public posts from* u
      $\mathscr{DS} \leftarrow \mathscr{DS} \cup \{u, \mathscr{D}\}$
      $\mathscr{PS} \leftarrow \mathscr{PS} \cup \{u, \mathscr{P}\}$
      for each $\langle post : p \rangle \in \mathscr{P}$
        do
          $\mathscr{T} \leftarrow$ *set of people who are tagged in* p
          $\mathscr{L} \leftarrow$ *set of people who like* p
          $\mathscr{UT} \leftarrow \mathscr{UT} \cup \mathscr{T} \cup \mathscr{L}$
          $\mathscr{C} \leftarrow$ *set of comments in* p
          for each $\langle comment : c \rangle \in \mathscr{C}$
            do
              $a :$ *user who wrote comment* c
              $\mathscr{K} \leftarrow$ *set of people who like* c
              $\mathscr{UT} \leftarrow \mathscr{UT} \cup \{a\} \cup \mathscr{K}$
  $\mathscr{U} = \mathscr{UT}$
until $\mathscr{DS} =$ *limit*
return ($\mathscr{DS}$ *and* $\mathscr{PS}$)

Regarding the Twitter SN, we used the Twitter REST API,[20] an interface for programmatic access to read and write Twitter data. It is an HTTP-based API that can be queried to acquire Tweets, publish new ones, read author profiles and follower data, and more. Access to Twitter REST API is granted under OAuth[21] authorization and responses are provided in JSON format. The collection procedure we used is simple: sending a single keyword in a request, the Twitter REST API can return a real-time streaming of all the Tweets containing that keyword, provided that they are not more than the 10% of all the generated Tweets in a specific instant. We provided some random keywords in a time-window of a week, and we obtained about 420,000 Tweets.

The procedure for translating Twitter SN data into an OWL 2 ontology, which has as TBox the SN Ontology, is illustrated in Algorithm 3. Every Tweet is provided with data about the user who generated it and the Tweet which it is in response to. A Tweet is added to the ontology, then, if the user who generated it is still not in the ontology, it is added also. Then, if the Tweet which the original Tweet is in response to is still not in the ontology, it is put also in the ontology, and their reply-relation is annotated.

5.2. Data preprocessing and semantic enrichment

In Section 6 we propose a model of policy that allows to take into account the meaning of the managed data. In particular the policies that we take as use cases come from the EU Directive 95/46/EC and state

[20] https://dev.twitter.com/rest/public.
[21] https://dev.twitter.com/oauth.

Algorithm 2: FACEBOOK_NETWORK_DATA_TO_ONTOLOGY()

```
𝒰 = set of users
𝒪 = ∅
for each ⟨user : u⟩ ∈ 𝒰
    do { create individual person u
         𝒪 ← 𝒪 ∪ {u}
         𝒫 ← set of public posts from u
         for each ⟨post : p⟩ ∈ 𝒫
             do { create individual post p
                  create objectProperty o1 = {u makePost p}
                  𝒪 ← 𝒪 ∪ {o1}
                  ℒ ← set of people who like p
                  for each ⟨person : l⟩ ∈ ℒ
                      do { create individual person l
                           create objectProperty o2 = {l likes p}
                           𝒪 ← 𝒪 ∪ {o2} }
                  𝒞 ← set of comments in p
                  for each ⟨comment : c⟩ ∈ 𝒞
                      do { create individual comment c
                           create objectProperty o3 = {p hasComment c}
                           𝒪 ← 𝒪 ∪ {o3}
                           a : user who wrote comment c
                           create objectProperty o4 = {a makeComment c}
                           𝒪 ← 𝒪 ∪ {o4}
                           𝒦 ← set of people who like c
                           for each ⟨person : k⟩ ∈ 𝒦
                               do { create individual person k
                                    create objectProperty o5 = {klikesc}
                                    𝒪 ← 𝒪 ∪ {o5} } } } }
return (𝒪)
```

the necessity of anonymization, define the notion of personal data and processing of personal data, and constraint personal data processing (they are literally cited and discussed in Section 6).

Our use case policies shall apply to personal data, which needs to be identified and used to enrich the data contained in the OWL SN ontology. In common sense, we consider "personal data" as pieces of information which can be used to identify a person. That information include, but is not limited to: user ID, user name, first name, last name, full name, birth date, home town, living place, job, job location, company website, personal website.

Personal data may appear extrinsically or intrinsically. For example, the knowledge base about users contains all the users with their private data. However, e.g., in the conversation within a Facebook status, the text often contains some tagged people, as well as some (un-tagged) names. While those tagged names should be taken into account as private data and should be protected, the un-tagged names should be identified and protected as well. The information which should be identified include: first name, last name, full name (of people), names (of organization, locations, which may be used to identify people).

Algorithm 3: TWITTER_NETWORK_DATA_TO_ONTOLOGY()

$\mathscr{T}$ = *set of Tweets*
$\mathscr{O} = \emptyset$
for each $\langle T\ weet : t \rangle \in \mathscr{T}$
do
- *create individual Tweet* t
- **if** $t \notin \mathscr{O}$
  - **then** $\mathscr{O} \leftarrow \mathscr{O} \cup \{t\}$
- u = *user author of* t
- **if** $u \notin \mathscr{O}$
  - **then** $\mathscr{O} \leftarrow \mathscr{O} \cup \{u\}$
- *create objectProperty* $o1 = \{u\ makeTweet\ t\}$
- $\mathscr{O} \leftarrow \mathscr{O} \cup \{o1\}$
- $t2$ = *Tweet in reply to which* t *was generated*
- **if** $t2 \notin \mathscr{O}$
  - **then** $\mathscr{O} \leftarrow \mathscr{O} \cup \{t2\}$
- *create objectProperty* $o2 = \{t\ isInReplyTo\ t2\}$
- $\mathscr{O} \leftarrow \mathscr{O} \cup \{o2\}$

return $(\mathscr{O})$

Moreover, according to Article 8 of the EU Directive 95/46/EC, the protection shall apply to data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs. We consider those some topics (ethnicity, politics, religion) that need to be identified in a text.

Therefore, as a prerequisite for enforcing the policies presented in Section 6, the SN Ontology has to be enriched with external knowledge. In fact, policy 1 requires the anonymization of personal data (such as first name, last name, web site). Therefore, in user's messages, it is needed: (i) to recognize names appearing in their content, (ii) to determine whether a name needs to be protected or not, since popular, famous names do not need to be anonymized. Policy 2 requires the removal of some content in case a message reveals racial or ethnic origin, political opinions, religious or philosophical beliefs. Thus, it is necessary to determine if the content of messages is related to such topics.

There are two main issues we need to tackle here. First, how to detect if texts are private data or popular data. For instance, *Washington, Switzerland, Microsoft* are proper names but they are famous, popular names. Popular names may appear in the content of user's messages but there is no need to make them anonymous. Second, how to detect if some content is related to sensitive topics, including, but not limited to, religion, ethnicity, or politics. If such is the case, those messages shall not be stored.

5.2.1. Detection of private data

To recognize proper names in a text and to determine whether those names are popular or not, we employ Natural Language Processing (NLP) techniques, in particular we use Named Entity Recognition (NER) and disambiguation to Wikipedia (*D2W*). Named entity recognition purposes the detection and classification of text segments into pre-defined categories, such as Person, Organization, and Location. Disambiguation to Wikipedia (*D2W*) refers to the task of detecting and linking expressions in text to their referent Wikipedia pages. For instance, given a text "John McCarthy, 'great man' of computer science, wins major award.", a *D2W* system is expected to detect the text segment "John McCarthy" and link to the correct Wikipedia page *http://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist)*,

instead of other *John McCarthy* who are ambassador, senator or linguist. Since Wikipedia mainly contains popular names of people, organizations, locations, if a name can be linked to Wikipedia, it is very likely that it is a popular name. We use the *NER* system of [23] and the *D2W* system developed in [22].

5.2.2. Detection of topics

For the second issue, we apply a popular algorithm to detect if a given text is related to a pre-defined topic. The topic categories include, but not limited to: religion, ethnicity, politics. For each topic we construct a set of related terms. In order to detect if a message is related to sensitive topic such as politics, religion, ethnicity, we implemented a vector space model, where a message d_j is represented as vector of words and a topic t is represented by a vector of related terms (features):

$$d_j = (w_{1,j}, w_{2,j}, w_{t,j})$$
$$t = (w_{1,t}, w_{2,t}, w_{t,t})$$

Then, we employ the cosine similarity between the vectors to determine how important a text is related to a topic (set of terms):

$$\textit{sim}(d_j, t) = \frac{d_j \cdot t}{\|d_j\| \, \|t\|}$$

If the similarity exceeds some threshold, the text is then associated to that topic.

As for the weighting scheme we used the popular *tf-idf* [29], which, stands for term frequency-inverse document frequency, and is often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. The *tf-idf* weighting scheme assigns to term t a weight in document d given by

$$\textit{tf-idf}_{t,d} = \textit{tf}_{t,d} \times \textit{idf}_t$$

where *tf* stands for term frequency and *idf* stands for inverse document frequency.

6. Formalization and enforcement of policies

In normative multiagent system (NorMAS) literature, norms are usually studied from two different perspectives: (i) the development of techniques for norms monitoring with the goal of keeping the interaction among autonomous agents in open systems within certain boundaries [10]; (ii) the study of the techniques for developing agents able to reason and plan their actions on the basis of the norms that regulate their behaviour [4]. This second use is the one on which our work is focused. In NorMAS literature, it is also widely recognized that norms have the following main components [6]: (1) a *type* for specifying if they are used to define obligations, prohibitions, or permissions; (2) a *role* for expressing a class of debtors; (3) a *content* for the specification of the *action* performance or state of affairs that are regulated by the norm; (4) they are *active* during a period of time that can be expressed through *activation* and *deactivation events* or *conditions*; (5) norms may specify *sanctions* for norms violations, *rewards* for norm fulfillment; (6) norms are usually defined in a given *context*.

In this section, we present an application-independent formal model of *norms/policies* which has two important characteristics:

- In the specification of the *activation condition* and of the *content* components, it is possible to take into account the meaning (the formal semantics) of the managed data. This aspect is fundamental for being able to formalize, with the proposed model, the class of norms/policies that regulate the management of the data collected from the Web. In particular, the focus of this paper is on the formalization of those norms/policies that regulate the process of automatically transforming the content of the data extracted from the Web into data that fulfill certain ethical and legal constrains on the basis of their meaning.
- The actions regulated may be *internal* or *external* actions. Actions are *internal* when they have to be executed on the Data Ontology (in particular on the Social Network ontology) created with the data collected from the Web. The procedure for executing those *internal* actions may be specified in all details and their correct execution may be evaluated on the Data Ontology directly. The actions regulated by the policies formalized in this sections (policy-1a, policy-1b, and policy-2) are examples of *internal* actions. Actions are *external* when they are executed on the environment of the agents and compliance with those norms/policies is evaluated on a knowledge base (an ontology) that has to be dynamically updated to represent the changes in the environment and the changes in the observable properties of the agents acting within it. Examples of *external* actions discussed in [10,30] are leaving a room, paying or delivering a product, sending a file. Examples of *external* actions in the context of data collection are sharing a given set of data with other agents or deleting certain collected data with-in a given deadline. When norms/policies regulate *external* actions the procedure for their concrete execution is not described by the norms/policies, but it has to be implemented by the agent in charge of fulfilling them.

Given that for using the proposed model of policies/norms it is necessary that the data have a formal semantics, we propose to formalize them by using the OWL 2 ontology language. In particular we model the data collected from the web, and semantically enriched, with the *SN Onology* (introduced in Section 4), and we store the various components of the formalized policies in the OWL 2 *Policy Ontology*. Given that the activation condition and part of the content of the formalized policies are OWL axioms expressed using the classes and properties defined in the *SN Ontology*, the *Policy Ontology* imports the *SN Ontology*. The model of norms/policies presented in this section is partially inspired from the model of obligations presented in [10,13]. In this Section we describe also the *Policy Enforcement Service* which is in charge of checking the activation of the formalized policies stored in the *Policy Ontology* and enforce them by exploiting the reasoning services of an OWL reasoner. We exemplify also how to use the proposed model of norms/policies by formalizing three obligations coming from two EU data protection policies.

6.1. The model of policies

A policy (represented as an individual in the OWL *Policy Ontology*) may express obligations or prohibitions. For distinguishing between those two different type of policies, we introduce the Obligation and Prohibition classes which are subclasses of the Policy class. Policies are characterized by an *activation condition* that should be satisfied in order that they become active and by a *content* that is the action that should or should not be performed on the data. In some situation the *role* of the agent that will use the collected data (for example the social scientist), and the *context* in which those data will be used (for example the type of activities where the data will be used) may be relevant in the definition of the policies, we will tackle these extensions of our model in our future works.

The *activation condition* of a policy is used to distinguish between active and inactive policies. When an agent is reasoning on a given set of policies, only the active ones are taken into account, the other

are ignored. Thanks to the fact that the data collected from the Web are stored in an OWL ontology, in our model the activation condition of a generic policy-n is specified using an OWL axiom. In particular by introducing in the *Policy Ontology* a specific class, the ActivationCond-n class, which is defined as equivalent to an OWL axiom. This choice makes it easily possible to evaluate if a given policy is active by submitting to the reasoning services of an OWL reasoner a *retrieve query* that asks to the reasoner to retrieve the set $\mathscr{X}$ of individuals that belong to the ActivationCond-n class. Then if there is at least one individual in the set $\mathscr{X}$ it means that the policy is active. In the *Policy Ontology* for connecting a policy (which is an individual) to its activation condition (a class) we define a specific property: hasPolicyActivation: Policy $\rightarrow$ Object. This is possible in OWL 2 thanks to *punning*[22] Class $\leftrightarrow$ Individual, which allows to use the same term for both a class and an individual.

The choice of formalizing the activation conditions of policies using an OWL class has many advantages. First, in defining the activation condition class, it is possible to use the rich set of OWL operators available for defining classes, like for example the intersection of classes or properties restriction. Therefore, in the process of retrieving the individuals that belong to the activation condition class, it is possible to exploit the reasoning capability of an OWL reasoner and deducing new knowledge from the data contained in the Data Ontology. Another advantage is that the activation condition is written in the ontology and therefore it is possible to change it by simply using and ontology editor without the need to re-code a software. An alternative may consist in checking the activation of a policy by performing a SPARQL-DL[23] query, but in this case the activation condition would have to be written as SPARQL-DL query directly in the program that will execute it and would be more difficult to change it. Moreover, currently, the SPARQL-DL query language is not an international standard.

The *content* of a given policy is the specification of the *action* that should or should not be performed. In the type of policies on which this paper is focused, the action is an operation on the SN Ontology. Given that it is not possible to update the content of an OWL ontology using OWL operators, we will use a specific procedure for every policy and an OWL library (in our code is OWL-API [15]) for accessing and operating on the OWL ontology. The updates that is safe to perform on an OWL ontology must have an impact only on the *ABox*, that is the assertion component of the ontology. The assertions contained in the *ABox* may be *concept assertions* of the form C(a) (which states that an individual a belongs to a class C), or *relation assertions* of the form R(a,b) (which states that two individuals a and b are connected by a property R). In order to make the content of policies as much as possible declarative and parametric we propose to specify the regulated actions by using the basic operations of *insert*, *delete*, and *replace* performed on concept assertions or on relation assertions. As discussed in more detail in next section, for every policy those operations are executed on the individuals that belong to the activation condition class and to the content class of the policy.

An important aspect of the policy model presented in this paper, which has not being modelled in other approaches, is that the *activation condition* of a policy may be strictly connected with the *content* of the policy. That is, the *activation condition* class of a policy is used also in the definition of the action that should or should not be performed on the data.

When a new policy must be inserted in the *Policy Ontology* it is necessary to specify its *activation condition* class and its *content* class by specifying OWL axioms. Moreover, if the regulated action is an *internal* action, it is also required to describe in detail the action associated to the policy. Differently from KAoS basic form of policies [33], where the controlled action is a variable that refers to an action

[22] http://www.w3.org/TR/owl2-new-features/#F12:_Punning.
[23] http://www.w3.org/2001/sw/wiki/SPARQL-DL.

class defined in the ontology, in our model of norms/policies the controlled action is defined using OWL axioms. The important advantage of this choice is that it is possible to express complex actions by combining various OWL operators available for defining classes. The disadvantage may be that in order to be able to write OWL axioms the user must be an expert of this language and in particular of Description Logic (DL). The syntactic correctness of an OWL axiom may be easily checked by inserting it in the *Policy Ontology* by using an OWL editor like Protégé where OWL axioms must be written using the Manchester Syntax [16].

6.2. The policy enforcement service

The Policy Enforcement Service is in charge of evaluating at run-time the *activation conditions* of the various policies stored in the *Policy Ontology*. As previously discussed the *activation condition* of a specific policy is an OWL class defined as equivalent to an OWL axiom, which is expressed using classes and properties defined in a given ontology (e.g. the class of personal information that are not popular stored in the SN Ontology). Therefore evaluating the *activation condition* of one specific policy-n consists in submitting to the reasoning service of an OWL reasoner a *retrieve* query, evaluated on the *Policy Ontology*, which asks the set $\mathscr{X}$ of individuals that belong to the ActivationCond-n class. policy-n is *active* if there is at least one individual in the set $\mathscr{X}$.

If a given policy-n is active, and it belongs to the Obligation class, the Policy Enforcement Service is in charge of executing the action specified in the *content* component of the policy. For the class of policies formalized in this paper the obliged action is an internal action, and it consists in a sequence of *insert*, *delete*, or *replace* operations performed on concept assertions or on relation assertions in the SN Ontology. Those operations are related on two set of individuals: (i) the individuals in set $\mathscr{X}$ that belong to the ActivationCond-n class of policy-n; (ii) the individuals in set $\mathscr{Y}$ that belong to the Content-n class of policy-n. Similarly to set $\mathscr{X}$, set $\mathscr{Y}$ is obtained submitting to the reasoning service of an OWL reasoner a *retrieve* query, evaluated on the *Policy Ontology*, which asks the set of individuals that belong to the Content-n class. Once set $\mathscr{X}$ and set $\mathscr{Y}$ are retrieved, the Policy Enforcement Service executes the action associated to policy-n by executing its *content procedure* on the sets $\mathscr{X}$ and $\mathscr{Y}$.

The pseudocode of the Policy Enforcement Service is illustrated in Algorithm 4.

Algorithm 4: POLICY_ENFORCEMENT ()

$\mathscr{P} \leftarrow$ *retrieve*(Policy)
for each policy-n $\in \mathscr{P}$
do
- $\mathscr{X} \leftarrow$ *retrieve* (ActivationCondition-n)
- **if** $\mathscr{X} \neq \emptyset$
  - **then**
    - **if** policy-n $\in$ Obligation
      - **then**
        - $\mathscr{Y} \leftarrow$ *retrieve* (Content-n)
        - *execute content procedure* $(\mathscr{X}, \mathscr{Y})$ *of* policy-n

6.3. Modelling and reasoning on policy examples

In this section we formalize some specific policies for exemplifying how the previously presented model can be concretely used. The policies that we take as use cases in this paper come from the EU Directive 95/46/EC and state the necessity of anonymization at point (26), define the notion of personal

data and processing of personal data in Article 2, and constraint personal data processing in Article 8. They are reported in the following:

> (26) Whereas the principles of protection must apply to any information concerning an identified or identifiable person;...; whereas the principles of protection shall *not apply to data rendered anonymous* in such a way that the data subject is no longer identifiable;...

> **Article 2**
> (a) 'personal data' shall mean any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly;
> (b) 'processing of personal data' ('processing') shall mean any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction;

> **Article 8**
> 1. Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life.

Starting from the aforementioned statements of the EU Directive, the following two policies can be formulated:

Policy 1. It is obligatory to make anonymous all personal data relating to an identified or identifiable natural person in order to store, retrieve, and use them. Those properties include: username, user ID, first name, last name, full name, web site.

Policy 2. It is obligatory to prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs. When applied to the context of social networks, this policy becomes: it is obligatory to anonymize or remove the content of a post or of a comment if it reveals racial or ethnic origin, political opinions, religious or philosophical beliefs.

The first policy is related to the properties of a user. Whereas first name and last name belong to people only, the other properties can belong to other entities like organization, community or location in social networks. The second policy should be considered in a context of human statements or sentences, like the content of user posts, statuses, or comments on each post.

6.3.1. Formal model of policy 1

Policy 1 requires that personal information (such as first name, last name, web site) have to be anonymized. It may be modelled by formalizing two obligations, which requires the anonymization of (i) user attributes and (ii) personal information that appear in the content of posts/comments.

6.3.1.1. Policy 1-obligation 1

$\mathsf{policy\text{-}1a} \in \mathsf{Obligation}$ is activated when in the SN Ontology there is a user personal data which is not popular. Its *activation condition* class is formalized with the following OWL axiom:

$$\mathsf{ActivationCond\text{-}1a} \equiv \mathsf{PersonalAttribute} \sqcap \mathsf{hasPrivateData} \ni \mathsf{True} \tag{1}$$

where relation assertions that involve the data property $\mathsf{hasPrivateData}$: $\mathsf{PersonalData} \rightarrow \{\mathsf{True},\mathsf{False}\}$ has been created by the algorithm described in Section 5.

The *content* of $\mathsf{policy\text{-}1a}$ is the action of retrieving all user's personal information and then anonymize them. This can be done in two steps: (i) first, by asserting that those personal information are connected

(by means of the object property anonymize: PersonalData $\rightarrow$ AnonymizedData) to an anonymous individual that belongs to the AnonymizedAttribute $\sqsubseteq$ AnonymizedData class; (ii) second, by retrieving all the users associated to those personal information and replace all the relations that associate a user to his own personal information with a relation from the user to the corresponding anonymous individual. In order to perform this action we need to define the Content-1a class as equivalent to the the set of users who possess the personal data defined by the ActivationCond-1a class:

$$\text{Content-1a} \equiv \text{User} \sqcap \exists \text{hasPrivateProperty.ActivationCond-1a} \tag{2}$$

The obliged internal action is realized by a procedure written in Java, which uses OWL-API to operate on the SN Ontology by means of the basic operations of *insert*, *delete*, and *replace*. The procedure receives from the Policy Enforcement Service the sets $\mathscr{X}$ and $\mathscr{Y}$ and executes the following operations:

1. For every personal data in $\mathscr{X}$ *insert* into the ontology a unique anonymous individual that belongs to the AnonymizedAttribute class.
2. *Insert* a relation assertion that connects every personal data with its corresponding anonymous individual by means of the anonimize object property. The correspondence between personal data and antonymous attribute is also stored in a file for possible future use.
3. *Replace* all the relations between a user in $\mathscr{Y}$ and its personal data in $\mathscr{X}$ with another relation between the user and the anonymous individual created for each specific personal data in step (1).

6.3.1.2. Policy 1-obligation 2

policy-1b $\in$ Obligation is activated when in the SN Ontology there is a message and it contains personal information. The Message class is used for the textual content of a statement. The Statement class is a generic class for representing posts, comments, or tweets of Facebook/Twitter users. Facebook statuses are considered as posts. A user may also make a post or comment on a post made by another user. The *activation condition* of policy-1b is expressed with the following OWL class:

$$\text{ActivationCond-1b} \equiv \text{Message} \sqcap \text{hasPrivateData} \ni \text{True} \tag{3}$$

The *content* of policy-1b is the action of anonymizing all personal information that appear in a message. To do this we follow a procedure, which is similar to the one used for fulfilling policy-1a. However, two more steps are needed: (i) detect personal information in the content of messages by using the algorithm described in Section 5.2 and (ii) reconstruct the content with the personal information replaced by anonymous values. In order to perform this action we need to define the Content-1b class as equivalent to the class of users who make the set messages defined by the ActivationCond-1b class:

$$\text{Content-1b} \equiv \text{User} \sqcap \exists \text{makeStatement}(\exists \text{hasMessage.ActivationCond-1b}) \tag{4}$$

The procedure that should be performed for fulfilling policy-1b, by using the sets $\mathscr{X}$ and $\mathscr{Y}$ received by the Policy Enforcement Service, is composed by the following operations:

1. For every message in $\mathscr{X}$ (which contains private data) *insert* in the ontology a unique anonymous individual that belongs to the AnonymizedMessage class, in which the private data in the message has been anonymized.
2. *Insert* into the ontology the assertions for connecting the content of a message to its anonymized message by means of the anonymize property introduced above (this is possible because Message $\sqsubseteq$ PersonalData and AnonymizedMessage $\sqsubseteq$ AnonymizedData).
3. *Replace* all the relations between the posts/comments/tweets of users in $\mathscr{Y}$ having content in $\mathscr{X}$ with another relation between the posts/comments/tweets and the anonymous message created for their content.

Table 1
Data statistics on Facebook

Number of seed users	Number of posts	Number of comments	Number of final users
20	19	26	95
50	89	58	220
100	221	136	419
200	450	346	892
300	681	729	1492
400	1000	976	2032
500	1208	1152	2517

6.3.2. Formal model of policy 2

policy-2 $\in$ Obligation is activated when in the SN Ontology there is a statement (post, comment, or tweet) whose content is related to a sensitive topic. The topic of the content of statements has been detected with the method described in Section 5.2. On the basis of the result of this method, the assertions for connecting the content of statements to their topic, by using the property isRelatedTo: Message $\rightarrow$ Topic, are inserted in the ontology. The *activation condition* of policy-2 is expressed with the following OWL class, which is equivalent to the statements whose content is related to a sensitive topic:

$$\text{ActivationCond-2} \equiv \text{Statement} \sqcap \exists \text{hasMessage}.(\exists \text{isRelatedTo}.\text{SensitiveTopic}) \tag{5}$$

The *content* of policy-2 is the action of removing sensitive topics in the content of statements. To do this it is necessary to retrieve all statements which contains sensitive topics (their message is related to a sensitive topic), second to retrieve all users who make them and remove from the ontology all the assertions that connect the users to those statements. In order to perform this action, we need to define the Content-2 class as equivalent to the class of users who make the set of statements defined by the ActivationCond-2 class:

$$\text{Content-2} \equiv \text{User} \sqcap \exists \text{makeStatement}.\text{ActivationCond-2} \tag{6}$$

The procedure for fulfilling policy-2, by using the sets $\mathscr{X}$ and $\mathscr{Y}$ received by the Policy Enforcement Service, is equivalent to the operation of *removing* from the ontology the assertions that connect the statements in $\mathscr{X}$ to the users in $\mathscr{Y}$.

7. Experiments

In this section, we present the experiments performed to test and evaluate our implementation of the proposed framework. In particular we focus on the following processes: (i) the collection of the data; (ii) their conversion to OWL 2 ontology; (iii) their enrichment with semantic information using natural language processing techniques; (iv) the enforcement of the three formalized obligations on the enriched data.

We used the RestFB software for collecting data from Facebook and Twitter REST API for collecting Tweets from Twitter. We converted those data to OWL ontological propositions, using the OWL-API package and the Algorithm 2 presented in Section 5. Then, we applied natural language processing modules (described in Section 5.2) for annotating the OWL data as private data, presence of sensitive topics and so on. The techniques include: named entity recognition, entity disambiguation, topic detection. They are applied over user properties and textual contents of messages.

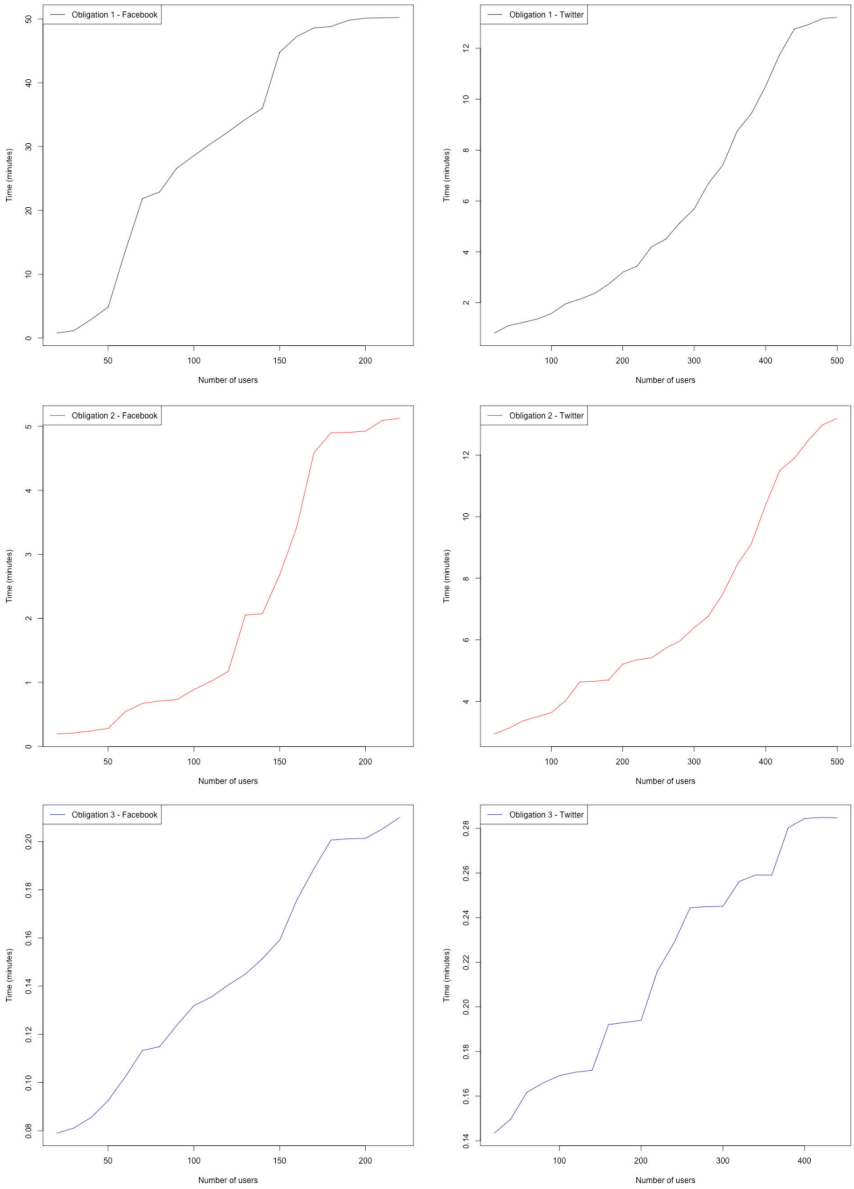

Fig. 3. Response time on the Facebook (left) and Twitter (right) datasets. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/MGS-150233)

We report some statistics about the data acquisition for the Facebook environment only. Table 1 presents the number of seed users, all users and posts/comments. According to the Algorithm 2, with the seed users, the dataset is expanded with: the users who like the posts of the seed users, the users who comment on the posts and so on. After the acquisition, we obtained an expanded set of users, together with a set of posts/comments in the interactions between them. Those data are presented in Table 1.

Once we obtained the data, that is enriched and annotated with semantic information, we enforce the three formalized obligations on the annotated data. With the first two obligations, the private data will be anonymized. While the first obligation enforces on private attributes, the second enforces on private

data inside textual content of messages. Those private data are identified thanks to the NLP mechanisms. With the last enforcement, the messages related to sensitive topics are marked and removed accordingly. The detection whether a message belongs to a topic is done thanks to the topic detection algorithm.

The response time for the enforcement of the three obligations is presented in Fig. 3. Using a PC with Intel(R) Core(TM) 2 Quad CPU Q9650 @ 3.00 Ghz and 4 GB RAM, we report the correlation between response time and data size (in terms of number of users). Each row represents the running time for each obligation and each column corresponds to the Facebook and Twitter datasets. The x-axis represents number of users while the y-axis represents the response time (in terms of minutes).

First, as we can see in both Facebook and Twitter datasets and in all three obligations, the response time reaches a stable level at some point. It means that our application can be applied in reality. Second, the first obligation clearly takes more time than the other ones. The reason is that there are so many private attributes of Facebook/Twitter users, even more than the number of private data entries found within the messages. We can see the third obligation is the fastest. It is because: (i) The number of messages is less than the number of private attributes in the first obligation; (ii) Only a relatively small portion of messages is related to sensitive topics, such as ethnicity, politics, or religion. Nevertheless, the response time is almost stable in each scenario. It also shows the effectiveness of our approach.

8. Conclusion

The study of normative and policy-based systems and their use in different fields of application is a well-known topic of research in the Normative Multiagent Systems community. However, the problem of automatically regulating the process of web data collection, taking into account privacy concerns, have still not been studied in much depth. Given that collecting web data is determining, in particular for social science fields, the goal of this paper has been to formalize privacy policies as an OWL ontology and to enforce them automatically, by benefiting from the properties of the OWL language and its reasoning mechanism.

To our knowledge, this is the first framework of privacy control for web data collection, without the need for reading obligations in natural language, understanding, and finally applying them manually. Our work demonstrates that the data collection can be done in compliant with privacy policies in an automatic way, by exploiting the robustness of Semantic Web technologies for expressing policies formally, for representing the data extracted from social networks and for reasoning on their semantics.

Such approach allows the re-usage of formal policies, the possibility to partially change them without modifying any hard-coded software, the ability to perform data integration between different sources eventually with ontologies and reasoning, and the possibility for automatic agents to read them and regulate their behavior consequently.

In respect to the work that has been done in Normative Multiagent Systems and Web Access Control, the present work improves the state of the art in its model of formal specification of policies. Such model is based on the axiomatization of the policies, while other OWL-based models (e.g. OWL-POLAR [30], KAoS [33]) represent the policies as OWL individuals. That is done in order to better take advantage of the abilities of OWL reasoning for policy enforcement. Moreover, policies in those areas of research are specified usually in order to regulate access to specific pieces of data, or to enforce the execution of specific actions in systems (*external actions*, see Section 6). However, they do not take advantage of the possibilities introduced by the enforcement of *internal actions*, as, e.g. the possibility to anonymize data as a consequence of an obligation to do so.

In our future works, we plan to analyze the application and enforcement of prohibition policies in more depth, focusing the present work on the enforcement of obligations mainly, while specifying both types formally. Such approach would require to opt for a more general view in which policies regulate access to resources as a consequence of a data request, instead of enforcing policies on acquired data only. Access can be regulated in function of users' role, attributes and context of usage, by modeling such information with OWL ontologies.

Acknowledgement

The work described in this paper is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) project nr. C11.0128 within the COST Action IS1004 WEBDATANET.

References

[1] A. Alamri, P. Bertok and J.A. Thom, Authorization control for a semantic data repository through an inference policy engine, *IEEE Trans Dependable Secur Comput* **10**(6) (Nov 2013), 328–340.
[2] J. Bradshaw, A. Uszok, M. Breedy, L. Bunch, T. Eskridge, P. Feltovich, M. Johnson, J. Lott and M. Vignati, The KAoS policy services framework, in: *Proceedings of the Eighth Cyber Security and Information Intelligence Research Workshop (CSIIRW 2013), Oak Ridge, TN: Oak Ridge National Labs* (2013).
[3] L. Costabello, S. Villata and F. Gandon, Context-aware access control for RDF graph stores, in: *ECAI of Frontiers in Artificial Intelligence and Applications*, L.D. Raedt, C. Bessire, D. Dubois, P. Doherty, P. Frasconi, F. Heintz and P.J.F. Lucas, eds, Vol. 242, IOS Press, 2012, pp. 282–287.
[4] N. Criado, E. Argente and V. Botti, Rational strategies for norm compliance in the n-BDI proposal, in: *Coordination, Organizations, Institutions and Norms in Agent Systems VI of LNCS*, M.D. Vos, N. Fornara, J.V. Pitt and G.A. Vouros, eds, Vol. 6541, Springer, 2011, pp. 1–20.
[5] S. Cucerzan, Large-scale named entity disambiguation based on wikipedia data, in: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Association for Computational Linguistics, Prague, Czech Republic, (Jun 2007), 708–716.
[6] K. da Silva Figueiredo, V. Torres da Silva and C. de Oliveira Braga, Modeling norms in multi-agent systems with NormML, in: *Coordination, Organizations, Institutions and Norms in Agent Systems VI of LNCS*, Vol. 6541, Springer, 2010, pp. 39–57.
[7] R. Eynon, J. Fry and R. Schroeder, The ethics of internet research, in: *The SAGE Handbook of Online Research Methods*, N. Fielding, R.M. Lee and G. Blank, eds, SAGE, chapter 2, 2008, pp. 23–41.
[8] N. Fielding, *The SAGE Handbook of Online Research Methods*, SAGE, (Jul 2008).
[9] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough and B. Thuraisingham, ROWLBAC: Representing role based access control in OWL. in: *Proceedings of the SACMAT*, New York, NY, USA, (2008), 73–82. ACM.
[10] N. Fornara, Specifying and monitoring obligations in open multiagent systems using semantic web technology, in: *Semantic Agent Systems: Foundations and Applications of Studies in Computational Intelligence*, Chapter 2, Springer-Verlag **344** (2011), 25–46.
[11] N. Fornara and M. Colombetti, Specifying and enforcing norms in artificial institutions, in: *Declarative Agent Languages and Technologies VI*, Springer Berlin/Heidelberg **5397** (2009), 1–17.
[12] N. Fornara and M. Colombetti, Representation and monitoring of commitments and norms using OWL, *AI Commun* **23**(4) (2010), 341–356.
[13] N. Fornara and C. Tampitsikas, Semantic technologies for open interaction systems, *Artificial Intelligence Review* **39** (2013), 63–79.
[14] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*, Chapman and Hall/CRC, (2009).
[15] M. Horridge and S. Bechhofer, The OWL API: A Java API for OWL Ontologies, *Semant Web* **2**(1) (Jan 2011), 11–21.
[16] M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens and H. Wang, The manchester OWL syntax, in: *OWLED2006 Second Workshop on OWL Experiences and Directions*, Athens, GA, USA, (2006).
[17] V. Kolovski, J. Hendler and B. Parsia, Analyzing web access control policies, in: *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, ACM, New York, NY, USA, (2007), 677–686.
[18] K.L., T. Finin and A. Joshi, A policy language for pervasive systems, in: *In Fourth IEEE International Workshop on Policies for Distributed Systems and Networks* (2003).

[19] A. Masoumzadeh and J. Joshi, Ontology-based access control for social network systems, *IJIPSI* **1**(1) (2011), 59–78.
[20] R. Mihalcea and A. Csomai, Wikify!: Linking documents to encyclopedic knowledge, in: *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management*, ACM, New York, NY, USA, (2007), 233–242.
[21] D. Milne and I.H. Witten. Learning to link with Wikipedia. in: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ACM, New York, NY, USA, (2008), 509–518.
[22] T.V.T. Nguyen, Disambiguation to Wikipedia: A Language and Domain independent approach, in: *Proceedings of the 9th Asia Information Retrieval Societies Conference*, Singapore, (Dec 2013).
[23] T.V.T. Nguyen and A. Moschitti, Structural reranking models for named entity recognition, *Intelligenza Artificiale* **6** (Dec 2012).
[24] T.V.T. Nguyen and M. Poesio, Entity disambiguation and linking over queries using encyclopedic knowledge, in: *Proceedings of the 6th Workshop on Analytics for Noisy Unstructured Text Data (AND), Collocated with COLING*, Mumbai, India, (Dec 2012).
[25] S. Ossowski, *Agreement Technologies of Law, Governance and Technology Series*, chapter Part III Norms, ed., Springer, **8** (2013), 169–249.
[26] P.D. Pedraza, Memorandum of understanding for the implementation of the COST Action IS1004: WEBDATANET, http://w3.cost.eu/fileadmin/domain_files/ISCH/Action_IS1004/mou/IS1004-e.pdf, (Dec 2010). Last accessed on 19 November 2014.
[27] L. Ratinov, D. Roth, D. Downey and M. Anderson, Local and global algorithms for disambiguation to wikipedia, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Portland, Oregon, USA, (Jun 2011), 1375–1384.
[28] O. Sacco and A. Passant, A privacy Preference ontology (PPO) for linked data, in: *Proceedings of the Linked Data on the Web Workshop (LDOW), Colocated with WWW*, Hyderabad, India, (2011).
[29] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, (1986).
[30] M. Sensoy, T.J. Norman, W.W. Vasconcelos and K. Sycara, OWL-POLAR: A framework for semantic policy representation and reasoning, *Web Semantics: Science, Services and Agents on the World Wide Web* **12–13** (Apr 2012), 148–160.
[31] G. Tonti, J.M. Bradshaw, R. Jeffers, R. Montanari, N. Suri and A. Uszok, Semantic web languages for policy representation and reasoning: A comparison of KAoS, rei and ponder, in: *International Semantic Web Conference of LNCS*, Vol. 2870, D. Fensel, K.P. Sycara and J. Mylopoulos, eds, Springer, 2003, pp. 419–437.
[32] G. Tonti, R. Montanari, J. Bradshaw, L. Bunch, R. Jeffers, N. Suri and A. Uszok, Automated generation of enforcement mechanisms for semantically-rich security policies in java-based multi-agent systems, in: *Multi-Agent Security and Survivability, 2004 IEEE First Symposium on* (Aug 2004), 11–20.
[33] A. Uszok, J.M. Bradshaw, J. Lott, M.R. Breedy, L. Bunch, P.J. Feltovich, M. Johnson and H. Jung, New developments in ontology-based policy management: Increasing the practicality and comprehensiveness of KAoS, in: *IEEE International Workshop on Policies for Distributed Systems and Networks*, IEEE Computer Society, (2008), 145–152.

Authors' Bios

Truc-Vien T. Nguyen has obtained her Ph. D. at the University of Trento in Computer Science. Her thesis' topic focuses on the extraction of semantic information from text using state-of-the-art machine learning methods. After the PhD, Truc-Vien worked as a postdoc in the same university, where her work continued to exploit machine learning for other data mining tasks, such as entity disambiguation on multiple languages using web sources. She has more than 20 publications in prestigious conferences and journals in data mining/natural language processing communities. From 2013 Dr. Truc-Vien T. Nguyen is postdoc researcher at the Università della Svizzera italiana, Faculty of Communication Sciences, Lugano, Switzerland where she works in the project "Automatic Web data collection from non-reactive sources by means of normative systems and Semantic Web Technologies" funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) within the COST Action IS1004 WEBDATANET.

Nicoletta Fornara is Senior Researcher and Lecturer at Università della Svizzera italiana, Faculty of Communication Sciences, Lugano, Switzerland. She is principal investigator of a two years SERI-COST

project with title "Automatic Web data collection from non-reactive sources by means of normative systems and Semantic Web Technologies" connected to the COST Action "WEBDATANET" in which she is Member for Switzerland of the Management Committee. From 2007 she is member of the COIN (Coordination, Organization, Institutions and Norms in agent systems) Steering Committee and she was co-chair of COIN@MALLOW 2010 and COIN@AAMAS06 workshops. She has authored numerous journal and international conference papers and some chapters of books in the research area of multi-agent systems, in particular on agent communication languages, artificial institutions, normative systems and agent's environment, and in the area of Semantic Web Technologies. For a complete list of publications visit http://www.people.usi.ch/fornaran/publications.html.

Fabio Marfia is Ph. D. student at Politecnico di Milano since 2011, under the supervision of prof. Marco Colombetti. His master thesis deals with the specification of domain-dependent policies, together with the development of specific functionalities related to policies as policy decision and policy explanation using OWL technology. Fabio Marfia is software developer at the Università della Svizzera italiana, Faculty of Communication Sciences, Lugano, Switzerland from 2014, where he works at the project "Automatic Web data collection from non-reactive sources by means of normative systems and Semantic Web Technologies", funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) within the COST Action IS1004 WEBDATANET.