The 6th International Semantic Web Conference and
the 2nd Asian Semantic Web Conference

ISWC+ASWC 2007

Workshop 11

Privacy Enforcement and Accountability with Semantics (PEAS2007)

Workshop Organizers:

Tim Finin, Lalana Kagal, Daniel Olmedilla

12 Nov. 2007
BEXCO, Busan KOREA

ISWC 2007 Sponsor

Emerald Sponsor

Saltlux

Gold Sponsor

Google

u-MTec

KAIST

VULCAN
A Paul G. Allen Company

Kogb

Silver Sponsor

IBM Research

NeOn

EAST WEB

ASIA-LINK
EUROPEAID
CO-OPERATION OFFICE

ETRI

FRANZ INC.

A
ADUNA

IOS
Press

We would like to express our special thanks to all sponsors

ISWC 2007 Organizing Committee

General Chairs

Riichiro Mizoguchi (Osaka University, Japan)
Guus Schreiber (Free University Amsterdam, Netherlands)

Local Chair

Sung-Kook Han (Wonkwang University, Korea)

Program Chairs

Karl Aberer (EPFL, Switzerland)
Key-Sun Choi (Korea Advanced Institute of Science and Technology)
Natasha Noy (Stanford University, USA)

Workshop Chairs

Harith Alani (University of Southampton, United Kingdom)
Geert-Jan Houben (Vrije Universiteit Brussel, Belgium)

Tutorial Chairs

John Domingue (Knowledge Media Institute, The Open University)
David Martin (SRI, USA)

Semantic Web in Use Chairs

Dean Allemang (TopQuadrant, USA)
Kyung-Il Lee (Saltlux Inc., Korea)
Lyndon Nixon (Free University Berlin, Germany)

Semantic Web Challenge Chairs

Jennifer Golbeck (University of Maryland, USA)
Peter Mika (Yahoo! Research Barcelona, Spain)

Poster & Demos Chairs

Young-Tack, Park (Sonngsil University, Korea)
Mike Dean (BBN, USA)

Doctoral Consortium Chair

Diana Maynard (University of Sheffield, United Kingdom)

Sponsor Chairs

Young-Sik Jeong (Wonkwang University, Korea)
York Sure (University of Karlsruhe, German)

Exhibition Chairs

Myung-Hwan Koo (Korea Telecom, Korea)
Noboru Shimizu (Keio Research Institute, Japan)

Publicity Chair: Masahiro Hori (Kansai University, Japan)

Proceedings Chair: Philippe Cudré-Mauroux (EPFL, Switzerland

Metadata Chairs

Tom Heath (KMi, OpenUniversity, UK)
Knud Möller (DERI, National University of Ireland, Galway)

Contents

Workshop Motivation and Goal

The concept of information sharing has dramatically changed with the new digital era. Handheld devices that could provide highly personal information about the owner (e.g., RFID, GPS) are becoming more pervasive. Our use of the Web also leads to the implicit sharing of information with others through our blogs, websites, social networks, Semantic Desktop sharing, clickstream tracking, as well as through the photographs, documents, and bookmarks we post on sites such as Flickr, Zoomr, and Delicious. Disclosing information to third parties may have unexpected consequences since a receiver of such information might easily use, copy, and redistribute it in ways not intended for by the owner. Users must understand the implications of using such devices or applications and providing information to third parties. Even though users may prevent the direct disclosure of sensitive information by an access control mechanism and the information being leaked may not seem private, sensitive information may revealed by inferences drawn from non-sensitive data and metadata. Examples include identifying a user and providing her sensitive information through a simple search engine query log, and retrieving medical data from sets of anonymized records. Thus along with privacy enforcement, accountability is also important because it may not always be possible to prevent third parties from obtaining sensitive information but accountability helps ensure that this information is used according to certain policies defined by the law or by the owner.

The role of Semantic Web research in privacy and accountability is two-fold. On the one hand, Semantic Web techniques may be used in order to provide advanced privacy and accountability mechanisms. Using formal languages with well-defined semantics in order to represent, reason about, and exchange such information helps to make it non-ambiguously understood by others. Privacy ontologies, sticky policies attached to data, accountability logics, and efforts such as the Creative Commons are some examples. Semantic Web languages can also be used to specify and track provenance of information, which is useful for accountability. Representing information in Semantic Web languages can also prevent sensitive information from being inferred by providing built in semantic models that can be used to recognize some potential inference channels. Another possible way to protect privacy is to disclose an appropriately generalized (or vague) answer to a query. For example, the query "where is John now" might be answered with "in room ITE 329 on the UMBC Campus " or "on the UMBC campus" or "somewhere in Maryland" depending on John's privacy preferences and the identify of the requester. Semantic Web languages provide a natural mechanism for generalization through their subclass structuring. The second role of Semantic Web research in this area is that privacy enforcement and accountability also apply to many emergent Semantic Web research topics. As an example, semantic desktop sharing poses questions about what to share, under which conditions, and how to control the usage of such information in a way that the privacy of the user is not violated. Understanding the new requirements that these scenarios pose is crucial for the short-term research in the area.

This workshop brings together researchers interested in the field in order to discuss and analyze important requirements and open research issues in this context, taking into account both perspectives: how can Semantic Web techniques help and which requirements arise from current Semantic Web research lines.

http://www.l3s.de/~olmedilla/events/2007/PEAS07/

Topics

- Ontologies for privacy
- Techniques for privacy, anonymity, pseudonymity, and unlinkability
- Privacy management & enforcement
- Information hiding and watermarking
- Information provenance
- Inference channels
- Generalization of answers
- Privacy policy specifications and business rules
- Negotiations and incentives for cooperation enforcement
- Accountability
- Privacy and personalization
- Privacy and mobility
- User- and context-awareness in privacy, security and trust
- P3P
- Digital Rights Management
- Creative Commons
- Pervasive technologies (RFID, cellular networks, WiFi) and Semantic Web
- Case studies, prototypes, and experiences
- Desktop search and sharing
- Legal and policy perspective of privacy

Programme Committee

- Elisa Bertino, Purdue University
- Piero Bonatti, University of Naples
- Grit Denker, SRI
- Li Ding, Stanford University
- Sandro Etalle, University of Twente
- Tim Finin, UMBC
- Yolanda Gil, ISI and USC
- Lalana Kagal, MIT
- Wolfgang Nejdl, L3S and University of Hannover
- Daniel Olmedilla, L3S and University of Hannover
- Alexander Pretschner, ETH Zurich
- Filip Perich, Shared Spectrum
- Pierangela Samarati, University of Milano
- Kent Seamons, BYU
- Ralph R. Swick, MIT and W3C
- William Winsborough, GMU
- Daniel Weitzner, MIT and W3C
- Marianne Winslett, University of Illinois at Urbana-Champaign

Beyond Secrecy: New Privacy Protection Strategies for the World Wide Web

Daniel J. Weitzner <djweitzner@csail.mit.edu>
Principal Research Scientist
Decentralized Information Group
MIT Computer Science and Artificial Intelligence Laboratory

In 1967, Alan Westin[1] set in motion the foundations of what most Western democracies now think of as privacy when he published his book, Privacy and Freedom. He defined privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others." His careful collection of sociological, legal, and historical perspectives on privacy came at a time when people worried that human dignity would erode or that governments would tend toward tyranny, becoming tempted to misuse their newfound power over private data. Computer scientists shared these concerns. Following Westin's emphasis on privacy as confidentiality, much of the security and privacy research over the last four decades has concentrated on developing more and more robust access control and confidentiality mechanisms.

Today, despite the fact that technical innovation in cryptography and network security has enabled all manner of confidentiality control over the exposure of identity in information systems, the vast majority of Internet user remain deeply worried about their privacy rights and correctly believe that they are far more exposed today than they might have been a generation earlier. Have we just failed to deploy the proper security technology to protect privacy, are our laws inadequate to meet present day privacy threats, or is have business practices and social conventions simply rendered privacy dead? While there is some truth to each possibility, the central failure to achieve robust privacy in the information age can be traced to an a long-standing mis-identification of privacy with confidentiality and access control.

Privacy protection in an era in which information flows more freely than ever will require increased emphasis on laws that govern how we can use personal data, not just who can collect it or how long they can store it. Much of our current privacy views are based on controlling access to information. We believed that if we could keep information about ourselves secret, prevent governments from accessing emails, and so on, then we would have privacy. In reality, privacy has always been about more than just confidentiality, and looking beyond secrecy as the sine qua non of privacy is especially important. New privacy laws should emphasize usage restrictions to guard against unfair discrimination based on personal information, even if it's publicly available. For instance, a prospective employer might be able to find a video of a job applicant entering an AIDS clinic or a mosque. Although the individual might have already made such facts public, new privacy protections would preclude the employer from making a hiring decision based on that information and attach real penalties for such abuses.

If we can no longer reliably achieve privacy policy goals by merely limiting access to information at one point on the Web, then what systems designs will support compliance with policy rules? Exercising control at one point in a large information space ignores the very real possibility that the same data is either available or inferable from somewhere else. Thus, we have to engineer Policy Aware systems based on design principles suitably robust for Web-scale information environments. Here we can learn from the design principles that enabled the Internet and the Web to function in a globally-coordinated fashion without having to rely on a single point of control. Colleagues at MIT, RPI, Yale and elsewhere are investigating designs for information systems that can track how organizations use personal information to encourage rules compliance and enable what we call information accountability, which pinpoints use that deviates from established rules.[2][3] We should put computing power in the service of greater compliance with privacy rules, rather than simply allowing ever more powerful systems to be agents of intrusion.

Accountable systems must assist users in seeking answers to questions such as: Is this piece of data allowed to be used for a given purpose? Is a string of inferences permissible for use in a given context, depending on the provenance of the data and the applicable rules. Information accountability will emerge from the development of three basic capabilities: policy-aware audit logging, a policy language framework, and accountability reasoning tools. A policy-aware transaction log will initially resemble traditional network and database transaction logs, but also include data provenance, annotations about how the information was used, and what rules are known to be associated with that information. Cryptographic techniques will play an important role in Policy Aware systems, but unlike the current reliance of privacy designs today, cryptography will be more for the purpose of creating immutable audit logs and providing verifiable data provenance information, than for confidentiality or access control.

Access control and security techniques will remain vital to privacy protection -- access control is important for protecting sensitive information and, above all, preserving anonymity. My colleague from UC Berkeley, Deirdre Mulligan, recounts a situation on the Berkeley campus in which a computer vision experiment on campus captured images of a group of female Iranian students engaged in a protest against Iranian human rights violations. Although they were free from harm on the campus, the fact that the researchers recorded the images and made them publicly available on the project's Web site put the students' family members, many of whom were still in Iran, at grave risk. The department took down the images as soon as they realized the danger, but harm could have easily occurred already.

Clearly, the ability to remain anonymous, or at least unnoticed and unrecorded, can be vital to protect individuals against repressive governments. Although US law doesn't recognize a blanket right of anonymity, it does protect this right in specific contexts, especially where it safeguards political participation and freedom of association. Even though no general protection exists for anonymous speech, we have a right keep private our role in the electoral process. Courts will protect the right of anonymous affiliation with political groups, such as the NAACP, against government intrusion. Finally, of course, we don't want our financial records or sensitive health information spilled all over the Web.

Nevertheless, in many cases the data that can do us harm is out there for one reason or another. With usage restrictions established in law and supported by technology, people can be assured that even though their lives are that much more transparent, powerful institutions must still respect boundaries that exist to preserve our individual dignity and assure a healthy civil society.

References

[1] A. Westin, *Privacy and Freedom*, The Bodley Head, 1967.

[2] Weitzner, Abelson, Berners-Lee, Feigenbaum, Hendler, Sussman, Information Accountability, MIT Tech. Report MIT-CSAIL-TR-2007, June 2007.

[3] Hanson, Kagal, Sussman, Berners-Lee, Weitzner, "Data-Purpose Algebra: Modeling Data Usage Policies", IEEE Workshop on Policy for Distributed Systems and Networks 2007, June 2007 (POLICY 2007).

An earlier version of this talk appears in Weitzner, Daniel J., "Beyond Secrecy: New Privacy Protection Strategies for Open Information Spaces," IEEE Internet Computing , vol.11, no.5, pp.96-95, Sept.

Semantic-Driven Enforcement of Rights Delegation Policies via the Combination of Rules and Ontologies *

Yuh-Jong Hu

Emerging Network Technology (ENT) Lab.
Dept. of Computer Science
National Chengchi University,
Taipei, Taiwan, 11605,
hu@cs.nccu.edu.tw

Abstract. We show that the semantic formal model for Open Digital Right Language (ODRL)-based rights delegation policies can be enforced and expressed as a combination of ontologies and rules, e.g., Semantic Web Rule Language (SWRL). Based on ODRL's expressions and data dictionary, a rights delegation ontology is proposed in this study. Furthermore, we express the rights delegation policy as a set of ontology statements, rules, and facts for usage and transfer rights delegation. When verifying ODRL formal semantics, our SWRL approach is superior to the generic restricted First Order Logic (FOL) model because we have an understandable formal semantics of policies for automatic machine processing and a higher expressive power for policy compliance checking. On the other hand, the rights delegation semantics shown as a generic full FOL might have a higher complexity of license verification, which results in a policy compliance checking that is possibly undecidable. A real usage rights delegation scenario for digital content is demonstrated in order to justify the feasibility of our formal semantic model for digital rights delegation. We hope this study will shed some light on future sensitive information usage and delegation rights controlled from a privacy protection perspective.

1 Introduction

The ultimate goal of achieving a distributed Digital Rights Management (DRM) system is content owners can project policies governing their content into remote environments with confidence that those policies will be respected by remote nodes [13]. A node is a trusted system that governs the legal usage of digital works that can be relied on to follow certain rules and enforce its legal rights delegation policy [19]. Aspects of the DRM rights authorization and enforcement problem include formulating delegation policies and a mechanism for "proving"

*This research was partially supported by the Taiwan National Science Council (NSC), under Grant No. NSC 95-2221-E-004-001-MY3.

that a request to access rights complies with relevant policies. A general-purpose Rights Expression Language (REL) is a type of policy delegation language where the focus of the language is on the expression and transference of usage rights or capabilities from one party to another in an interoperable manner. It will be a challenge to design a general-purpose REL for the DRM system that expresses rights delegation policies and controls digital content [13]. Emerging acceptable industry REL are classified into two major camps: Open Digital Right Language (ODRL) and eXtensible rights Markup Language (XrML). Unfortunately, the semantics of both of these RELs are either described in English or as computer algorithms, therefore, they lack machine understandable formal semantics.

There are two core components for a DRM rights delegation policy: an REL language for expressing policies and an evaluator that can make decisions based on such expressions. The policy evaluator must be able to reason correctly concerning all types of policy it may encounter when making a trusted decision to grant rights. Thus, the design of a policy evaluator is going to be influenced by the design of the REL language. A DRM policy evaluator must decide for each requested access whether the policy (or policies) is relevant to the request and whether or not to allow it to occur for a given license. This formulation of a DRM policy evaluation can be regarded as a "compliance checking" decision problem in a trust management system [1]. The license is derived from a legal contract that states the permissible agreements under which digital contents can be legitimately accessed. The languages for writing licenses (or permissible agreements) usually fall into three categories: a human readable natural language, a software readable XML-based language, and a machine understandable language [18].

When we consider digital contents as protected, sensitive, personal information which might be disseminated over the entire Web, then the usage and delegation rights control issues we are facing are just the same as those existing in the DRM system. Disseminated digital content (or information) with associated licenses are encrypted with appropriate security keys. If a node with a service request can decrypt the downloaded information and license, then the node's embedded license evaluator will faithfully interpret the license semantics and enforce the license agreements, including ontology, rules, and facts, to decide whether a request should be granted or not.

2 Research Goal

The goal of this research is to deal with the problem that license agreements written in ODRL REL, are open to interpretation that results in semantic ambiguity. This is because the stated conditions for which resources access legitimate license are written in English. We need an abstract semantic layer that can be overlaid on existing ODRL data models to express their license and service semantics instead of using natural language, such as English.

ODRL is one of the most popular RELs for expressing digital license exchange and sharing, it also has an XML-based markup language. As we know, XML has the capacity of marking up licenses and data for machine processing but does not have the capability of encoding the license semantics. The generic ODRL foundation model consists of three core entities: assets, rights, and parties. We are going to exploit this model by finding out which parts of license semantics can be shown as ontology language and which parts can be shown as rule language.

Therefore, DRM ontology and rights delegation policies will be using machines to ensure their license semantics. Finally, we show that our flexible rights delegation model could explicitly declare and enforce all kinds of rights delegation semantics through existing ODRL expressions and data dictionaries.

2.1 Our Approach

The formal semantics we propose are based on Semantic Web Rule Language (SWRL) [9]. SWRL is a language that combines description logic OWL with logic program rule language, such as RuleML Lite (see http://www.ruleml.org/#Lite.), where a Horn clause rules with the extension to OWL that overcomes many limitations of property chaining [9]. Property chaining features allow us to "transfer rights" from one class of individuals to another via delegation properties other than subClassOf rights inheritance.

In ODRL, possible permission usage rights are display, print, play, and execute. Possible permission transfer rights are usually defined as rights for rights, including sell, lend, give, and lease, etc [10]. Property chaining is a necessary feature for allowing rights delegation policies to delegate rights from one party to another when they belong to different classes. This important feature is not supported by other ontology-based semantic web policy languages, such as KAoS, Rei [20]. However, there are some limitations when using SWRL due to predicates being limited to being OWL classes and properties that only have a maximum parity of two, with no built-in arithmetic predicates or nonmonotonic features [5][9]. Therefore, we use OWL's extended concrete datatypes with unary and binary arithmetic operators in license agreement verification so that the verifier can verify whether prerequisite requirements and constraints in a license are compliant with its rights delegation policy [16].

When verifying ODRL formal semantics, the ontology+rule (SWRL) approach is superior to the generic restricted First Order Logic (FOL) formal semantics model [18]. First, generic restricted FOL-based rights delegation policies cannot be automatically processed by an agent because these FOL-based policies lack a semantic rights markup language. Second, unlike our SWRL (Ontologies+Rules) policies, restricted generic FOL policies do not have a high level of expressive power in their delegation policies [8]. On the other hand, the rights delegation semantics shown as a generic full FOL model might have a higher complexity of license verification, which results in a policy compliance checking that is possibly undecidable. Finally, Descritpion Logic (DL) in SWRL is possibly augmented by unary and binary arithmetic operators to enhance its concrete datatype operation [2].

3 Related Work

DRM and other modern access controls, such as privacy protection, RBAC, etc, are all regarded as $UCON_{ABC}$ models which integrate *Authorization (A), oBligations (B), and Conditions (C)* elements. Usage control is a generalization of access control that covers authorization, obligation, conditions, continuity (ongoing controls), and mutability [17]. In [21], a rule-based policy management system can be deployed in an open and distributed WWW site by creating a "policy aware" infrastructure. This makes the widespread deployment of rules and proofs on the Web to become a reality. However, this server-based access control infrastructure cannot be applied to DRM or other methods of privacy protection for usage and rights delegation control where information might be disseminated over the entire Web. Delegation Logic, a datalog-extended tractable logic-based language with expression of delegation depth and complex principals was proposed to represent policies, credentials, and requests in distributed authorization. However, it did not have a rights markup language to explicitly encode rights delegation ontology for automatic agent processing of its rights delegation semantics [14].

XrML does not have formal semantics [3]. Instead, the XrML specification presents semantics in two ways: as an English description of the language or as an algorithm that determines if rights are permissible from a set of licenses. A formal foundation model for XrML semantics is shown as FOL-based rights expression statements [7]. ODRL is another popular XML-based REL language used to state the conditions under which resources can be legitimately accessed [10]. ODRL does not have formal semantics either. The meaning of the language's syntax is described in English; license agreements written in ODRL are open to interpretation that results in semantic ambiguity. In order to resolve this problem, a formal foundation model for ODRL semantics is shown as a generic restricted FOL but it has less expressive power on rights expression and delegation as our SWRL approach [18]. In [6], they only provide a generic representation of contract information on top of RELs so that the enforcement of access rights can be extracted from ODRL-based digital license contracts. But, machine understandable formal semantics cannot be represented and processed in this study. In [4], an OWL-based ODRL formal semantic model is designed and deployed but it does not have usage and transfer rights delegation service capability. In summary, a formal foundation for ODRL or XrML semantics are shown as either FOL or OWL, but they all lack semantic-driven enforcement of rights delegation policies [4][18].

4 License Agreement for Usage Rights

The central construct of ODRL is a license agreement. A license agreement indicates the policies (rules) under which a principal $Prin_o$ allows another principal $Prin_{u_i}$ to use an asset r presumably owned by $Prin_o$, where $Prin_o$ is an asset owner and $Prin_{u_i}$ is one of n asset users, where $i \in (1, \cdots, n)$. A license agreement refers to a policy set showing any number of prerequisites and policies. A prerequisite is either a constraint, a requirement, or a condition. Constraints are facts that are outside the $Prin_{u_i}$'s influence but are defined by the asset owner $Prin_o$, such as counting or temporal restrictions for digital asset usage rights. Requirements are facts that are within the $Prin_{u_i}$ user's power to meet, such as prepaid fees before using a particular asset. Conditions are constraints that must *not* hold exceptions [18]. If all of the prerequisites are met, then a policy says that the agreement's users may perform the action for the license agreement's assets.

4.1 Rights Delegation Ontology

ODRL does not enforce or mandate any policies for DRM, but provides mechanisms to express such policies. ODRL specifications contain expression language, data dictionary elements, and XML syntax to encode the ODRL expressions and elements [10]. We are going to use these ODRL expression language and data dictionary elements as our rights delegation ontology's entities (see Fig. 1). The source of this ontology conceptualization is based on the ODRL 1.1 specification explicitly defining the ODRL's rights delegation semantics for a license in this ontology [10]. The class and property terms defined in this rights delegation ontology will be considered as antecedents or conclusion(s) in the following usage rights delegation policies to enforce all kinds of real rights delegation inference (see Section 5.2).

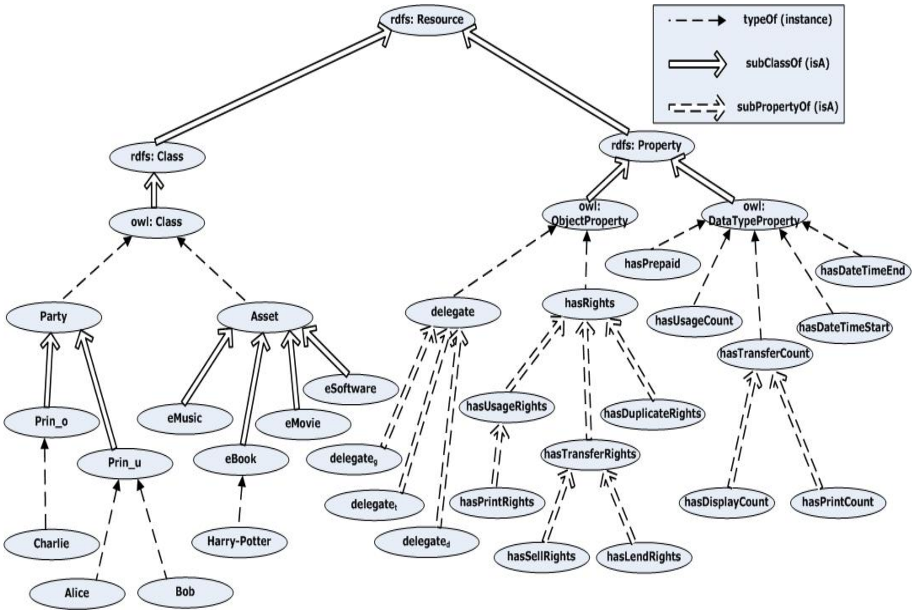


Fig. 1. A rights delegation ontology for an ODRL foundation model based on [10]

4.2 Usage Rights Delegation

We define $hasUsageRights$ as an abstract property describing the generic usage rights for a principal x to use an asset r. The domain class of $hasUsageRights$ property is $Party$, and the range class is Asset (see Fig. 2). The domain class of $delegate$ property is $Prin_o$ and the range class is $Prin_u$, where the delegate does have $subPropertyOf$ $(delegate_g, delegate_t, \cdots)$. The $delegate_g$ represents generic usage rights delegation property and the $delegate_t$ represents rights transfer delegation property. We do not allow a principal x be able to delegate his or her generic rights to another principal y if that principal x only has some usage rights but does not have any permissible transfer rights.

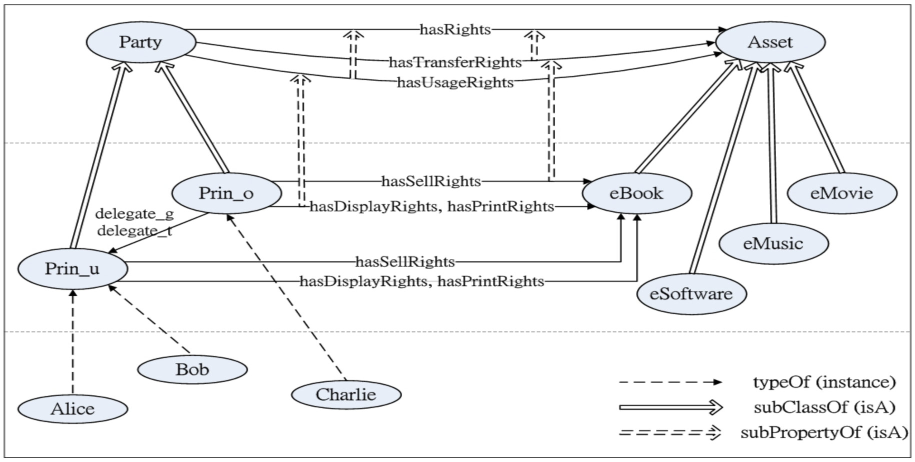


Fig. 2. A rights delegation snapshot based on rights delegation ontology

5 License Agreement for Transfer Rights

The delegation processes for transfer rights license agreements are activated using $delegate_t$ property, where the rights receiver owns the delegated rights but the rights owner might lose the rights temporarily or permanently. This is not true for some rights delegation scenarios where the rights owner and the rights receiver might have their rights concurrently. Thus, we create a rights **duplicate** delegation, indicated as $delegate_d$ from a variety of transfer rights. In this rights duplicate delegation property $delegate_d$, the rights original owner, concurrently has his or her own rights with the rights receiver after the rights duplicate delegation process is completed.

5.1 Prerequisites Expressions

We found that downstream rights receivers are receiving less rights in the rights delegation chain. An original content creator usually specifies his or her usage and transfer rights delegation with a reasonable number of depth d by constraint of $\leq_{\exists d}$ $hasTransferCount$, where d is a constant and is decreased by one for each delegation. Thus, the rights delegation process can be enabled as long as the condition $\geq_1$ $hasTransferCount$ is truth in a delegation policy (see rule ($o4$) in Section 5.2). In summary, we use extended OWL's unary arithmetic operators to express a prerequisite that can be a constraint, a requirement, a condition, or even a delegation depth.

Constraints for prerequisite such as, prepaid conditions, permissible count of upper (or exact) limit of usage rights, permissible delegation depth of transfer rights, and the validity time interval of usage rights, are shown in the following:

- MaxCardinality:$\leq_{\exists u}$ $hasUsageCount_{\exists p}.Asset$
- MaxCardinality: $\leq_{\exists t}$ $hasTransferCount_{\exists p}.Asset$
- Cardinality: $=_{\exists a}$ $hasPrepaid_{\exists p}.Party$
- Validity of time interval $\forall Time \in (t_1, t_2)$:
  $\geq_{\exists t_1}$ $hasDateTime_{\exists p}.Time \wedge \exists \leq_{t_2}$ $hasDateTime_{\exists p}.Time$

Sometimes, the usage rights prerequisite is enforced by a principal who is in charge of a counting action that collects all necessary mutable facts from the downstream rights receivers in the delegation chain. We show conditions as the following, where $\exists =_{\exists u}$ $hasUsageCount_{\exists p}.Asset$, $\exists \geq_{\exists t_1}$ $hasDateTime_{\exists p}.Time$, $\exists \leq_{\exists t_2}$ $hasDateTime_{\exists p}.Time$, and $\exists hasPrepaid_{\exists p}.Party$, indicate which principal p is in charge of mutable constraint parameter computations and policy compliance checking [15]. All of these will be demonstrated in Section 5.3.

5.2 Usage Rights Transfer Delegation

The $hasTransferRights$ is an abstract property describing the transfer rights delegation of usage rights for a principal x for an asset r. The domain class of property $hasTransferRights$ is $Party$ and the range class is Asset. $Prin_o$ might use $delegate_g$ to transfer usage rights only to $Prin_{u_i}$, where $i \in (1, \cdots, n)$, but does not delegate his transfer rights to $Prin_{u_i}$, where transfer rights $\in$ $(hasSell_tRights, \cdots)$. Therefore, each $Prin_{u_i}$ cannot further delegate his usage rights to another $Prin_{u_j}$ (see rule ($o2$)). If we use $delegate_t$ property, then any one of the transfer rights permissions $\in$ $(hasSell_tRights, \cdots)$ and usage rights can be further propagated (see rule ($o4$)). The depth of transfer delegation can be specified in class Asset with cardinality shown as $\exists =_{\exists n}$ $hasTransferCount.eBook$, which indicates that the transfer rights permission for $eBook$ can be propagated with the exact delegation depth of n:

- If party x has both usage and transfer rights for asset r, then he or she is allowed to transfer full (or partial) of both rights to another party but he or she can not keep his or her own rights after delegation:

$hasUsageRights(?x, ?r) \wedge hasTransferRights(?x, ?r)$
$\Longrightarrow hasUsageTransferRights(?x, ?r) \longleftarrow (o1)$

- Party x can only transfer his or her usage rights for asset r to another party y if his or her cumulative depleted usage count $<_{\exists u}$, where u is a constant indicating a count of upper limit of usage rights. Here, party y can have delegated usage rights but cannot have further delegation rights:

  $hasUsageTransferRights(?x, ?r) \wedge delegate_g(?x, ?y) \wedge hasPrepaid(?y, ?a) \wedge$
  $<_{\exists u} hasUsageCount(?r) \Longrightarrow hasUsageRights(?y, ?r) \longleftarrow (o2)$

- If party x has usage rights permission for resource r and the cumulative depleted usage count is $<_{\exists u}$. Furthermore, party x's current local date and time $t \in (t_1, t_2)$, then he or she is permitted to have these particular usage action, such as play, display, or print, etc:

  $hasUsageRights(?x, ?r) \wedge <_{\exists u} hasUsageCount(?r) \wedge \geq_{\exists t_1} hasDateTime(?t)$
  $\wedge \leq_{\exists t_2} hasDateTime(?t) \Longrightarrow Permitted(Usage, ?r) \longleftarrow (o3)$

- Party x can transfer his or her usage and transfer rights for asset r to another party y so party y can have x's both rights to transfer rights forward as long as x is not the final node in a delegation chain:

  $hasUsageTransferRights(?x, ?r) \wedge delegate_t(?x, ?y) \wedge hasPrepaid(?y, ?a) \wedge$
  $\geq_1 hasTransferCount(?r) \Longrightarrow hasUsageTransferRights(?y, ?r) \longleftarrow (o4)$

5.3 A Usage Rights Delegation Scenario

The following license agreement for a usage rights delegation scenario is adopted and modified from [18]. For reasons of space, a detailed discussion of the implications of our complete operational semantics for this scenario is left to the full paper for further study. This might need a speech-act agent communication language to represent message passing ontology, which would then allow our agents to automatically exchange interactive information among themselves as shown in [11]:

- **Natural Language (NL)** denotation of license agreement:

  Content distributor *Charlie c* makes an agreement with two content consumers, *Alice a* and *Bob b*. After each paying five dollars, and then both receiving acknowledgement from *Charlie*, *Alice* and *Bob* are given the usage rights and may each display an *eBook* asset, *Harry Potter and the Deathly Hallows*, up to five times. They may each print it only once. However, the total number of actions, either displays or prints done by *Alice* and *Bob*, may be at most ten. The usage rights validity period is between 2007/05/07/09:00 - 2007/05/10/24:00.

- **Human Readable Abstract Syntax** denotation of license agreement:

```
agreement
between Charlie and {Alice,Bob}
about Harry Potter and the Deathly Hallows
with inSequence[prePay[5.00],attribution[Charlie]]
|==> not[and[Time < 2007/05/07/09:00,Time > 2007/05/10/24:00]]
|==> with count[10] ==>
and[forEachMember[{Alice,Bob};count[5]] ==> display,
     forEachMember[{Alice,Bob};count[1]] ==> print]
```

- **First Order Logic (FOL)** denotation of license agreement:

$$\begin{array}{l}
\forall x((x = Alice \vee x = Bob) \Longrightarrow \\
\exists t_1 \exists t_2 (t_1 < t_2 \wedge Paid(5, t_1) \wedge Attributed(Charlie, t_2))) \Longrightarrow \\
\forall t \wedge hasDateTime(t) \geq 2007/05/07/09:00 \wedge \\
hasDateTime(t) \leq 2007/05/10/24:00 \Longrightarrow \\
count(Alice, id_1) + count(Alice, id_2) + count(Bob, id_1) \\
+\, count(Bob, id_2) < 10 \Longrightarrow \\
(count(Alice, id_1) < 5 \wedge count(Bob, id_1) < 5 \Longrightarrow \mathbf{Permitted}(x, display, ebook)) \\
\wedge\ (count(Alice, id_2) < 1 \wedge count(Bob, id_2) < 1 \Longrightarrow \mathbf{Permitted}(x, print, ebook)))
\end{array}$$

We use the ontologies+rules (SWRL) approach to enforce the semantics of rights delegation policies instead of the above pure FOL-based formula. The following ontology, rules, and facts are a partial view from distributor *Charlie c* based on Fig. 1 and Fig. 2. In the bootstrapping stage, *Charlie c* has all of the usage and transfer (or duplicate) rights for the *eBook* class, including *HarryPotter and the Deathly Hallows*, which are shown as the facts in the following page. Ontology statements ($c1$) - ($c3$) indicate the constraints of associated usage counts shown in the above FOL formula. After consumers *Alice a* and *Bob b* paying five dollars, then we use rules ($c4$) - ($c7$) to derive facts ($c8$) and ($c9$) that become $Alice's\ a$ facts ($a4$) and ($a5$) and derive facts ($c10$) and ($c11$) that become $Bob's\ b$ facts ($b2$) and ($b3$). Rules ($c4$) and ($c5$) are specialized cases for rule ($o1$), while rules ($c6$) and ($c7$) are specialized cases for rule ($o2$), shown in Section 5.2. The mutable facts ($c12$) - ($c14$) indicate a snapshot of current usage, display, and print counts collected from both *Alice a* and *Bob b*; they will be taken into summation by *Charlie c*.

- **SWRL (Ontologies + Rules)** denotation of license agreement:

  - Content distributor $Charlie's\ c$ site:

- Ontology:
  $hasDisplayRights \sqsubseteq hasUsageRights$
  $hasPrintRights \sqsubseteq hasUsageRights$
  $\leq (hasDisplayCount_{\{a,b\}}.eBook,\ hasUsageCount_c.eBook)$
  $\leq (hasPrintCount_{\{a,b\}}.eBook,\ hasUsageCount_c.eBook)$
  $\{Alice, Bob\} \overset{domain}{\longleftarrow} hasUsageRights \overset{range}{\longrightarrow} R_1,$
  where $R_1 =\leq_{10} hasUsageCount_c$
  $\wedge \geq_{2007/05/07/0900} hasDateTime_c.Time$
  $\wedge \leq_{2007/05/10/2400} hasDateTime_c.Time$
  $\exists =_\alpha \exists = sum(\exists \leq_5 hasDisplayCount_i.\{HarryPotter\}),\ i \in \{a, b\},$
  where α: $\exists hasDisplayCount_c.\{HarryPotter\} \longleftarrow (c1)$
  $\exists =_\beta \exists = sum(\exists \leq_1 hasPrintCount_i.\{HarryPotter\}),\ i \in \{a, b\},$
  where β: $\exists hasPrintCount_c.\{HarryPotter\} \longleftarrow (c2)$
  $\exists =_\delta sum(\alpha, \beta),$
  where $\delta : \exists hasUsageCount_c\{HarryPotter\} \longleftarrow (c3)$

- Rules:
  $hasDisplayRights(?x, ?r) \wedge hasSell_dRights(?x, ?r)$
  $\Longrightarrow hasDisplaySell_dRights(?x, ?r) \longleftarrow (c4)$

  $hasPrintRights(?x, ?r) \wedge hasSell_dRights(?x, ?r)$
  $\Longrightarrow hasPrintSell_dRights(?x, ?r) \longleftarrow (c5)$

  $hasDisplaySell_dRights(?x, ?r) \wedge delegate_g(?x, ?y)$
  $\wedge\ hasPrepaid(?y, ?a) \wedge \Longrightarrow hasDisplayRights(?y, ?r) \longleftarrow (c6)$

  $hasPrintSell_dRights(?x, ?r) \wedge delegate_g(?x, ?y)$
  $\wedge\ hasPrepaid(?y, ?a) \Longrightarrow hasPrintRights(?y, ?r) \longleftarrow (c7)$

- Facts:
  $eBook(HarryPotter)$
  $hasDisplayRights(Charlie, HarryPotter)$
  $hasPrintRights(Charlie, HarryPotter)$
  $hasSell_dRights(Charlie, HarryPotter)$
  $hasDisplaySell_dRights(Charlie, HarryPotter)$
  $hasPrintSell_dRights(Charlie, HarryPotter)$
  $\exists =_5 hasPrepaid(Alice)$
  $hasDisplayRights(Alice, HarryPotter) \longleftarrow (c8)$
  $hasPrintRights(Alice, HarryPotter) \longleftarrow (c9)$
  $\exists =_5 hasPrepaid(Bob)$
  $hasDisplayRights(Bob, HarryPotter) \longleftarrow (c10)$
  $hasPrintRights(Bob, HarryPotter) \longleftarrow (c11)$
  $delegate_g(Charlie, Alice)$
  $delegate_g(Charlie, Bob)$
  $\exists =_7 hasUsageCount_c(HarryPotter) \longleftarrow (c12)$
  $\exists =_6 hasDisplayCount_c(HarryPotter) \longleftarrow (c13)$

$\exists =_1 hasPrintCount_c(HarryPotter) \longleftarrow (c14)$

In the bootstrapping stage, all ontology statements, rules, and facts are described as license agreements and will be sent to $Alice\ a$ and $Bob\ b$ from $Charlie's\ c$. Facts $(a4)$ and $(a5)$ and facts $(b2)$ and $(b3)$ were previously inferenced on $Charlie\ c$ site via rule $(c6)$ and $(c7)$, where they were separately sent to $Alice\ a$ and $Bob\ b$. Each time $Alice\ a$ requests to display or print permission for $HarryPotter$, then associated rules $(a1)$ or $(a2)$ will be enforced to check whether conditions on the rule antecedents are all true. In fact, rules $(a1)$ and $(a2)$ are specialized cases of rule $(o3)$ in Section 5.2. For example, if $Alice\ a$ asks permission to print $HarryPotter$, her request will be granted because facts $(a5)$, $(a7)$, and $(a8)$ imply that all of the conditions on rule $(a2)'s$ antecedents are all true. Therefore, the conclusion $Permitted_a(Print, HarryPotter)$ is true. On the other hand, if $Bob\ b$ asks permission to print $HarryPotter$, it will not be granted because mutable fact $(b5)$ implies that $<_1 hasPrintCount_b(HarryPotter)$ is false, so the conclusion $Permitted_b(Print, HarryPotter)$ can not be derived. In our policy framework, we assume that what is not explicitly permitted is forbidden. Therefore, a permission request to print will be denied.

- Content consumer $Alice's\ a$ site:

  ∗ Ontology:
  Similar to content distributor $Charlie's\ c$ site's ontology, except the usage rights constraints are local to $Alice\ a$

  ∗ Rules:
  $hasDisplayRights(?x, ?r) \wedge <_{10} hasUsageCount_c(?r)$
  $\wedge <_5 hasDisplayCount_a(?r) \wedge \geq_{2007/05/07/09:00}\ hasDateTime(?t)$
  $\wedge \leq_{2007/05/10:24:00}\ hasDateTime(?t)$
  $\Longrightarrow Permitted_a(Display, ?r) \longleftarrow (a1)$

  $hasPrintRights(?x, ?r) \wedge <_{10} hasUsageCount_c(?r)$
  $\wedge <_1 hasPrintCount_a(?r)$
  $\wedge \geq_{2007/05/07/09:00}\ hasDateTime(?t)$
  $\wedge \leq_{2007/05/10:24:00}\ hasDateTime(?t)$
  $\Longrightarrow Permitted_a(Print, ?r) \longleftarrow (a2)$

  ∗ Facts:
  $eBook(HarryPotter) \longleftarrow (a3)$
  $hasDisplayRights(Alice, HarryPotter) \longleftarrow (a4)$
  $hasPrintRights(Alice, HarryPotter) \longleftarrow (a5)$

$\exists =_1 \ hasDisplayCount_a(HarryPotter) \longleftarrow (a6)$
$\exists =_0 \ hasPrintCount_a(HarryPotter) \longleftarrow (a7)$
$\exists =_7 \ hasUsageCount_c(HarryPotter) \longleftarrow (a8)$
$hasDateTime_a(2007/05/09/09:00) \longleftarrow (a9)$

- Content consumer $Bob's\ b$ site:

  ∗ Ontology:
  Similar to content distributor $Charlie's\ c$ site's ontology, except the usage rights constraints are local to $Bob\ b$

  ∗ Rules:
  Similar to content consumer $Alice's\ a$ site's rules, except the condition's subscript is b in rules $(a1)$ and $(a2)$

  ∗ Facts:
  $eBook(HarryPotter) \longleftarrow (b1)$
  $hasDisplayRights(Bob, HarryPotter) \longleftarrow (b2)$
  $hasPrintRights(Bob, HarryPotter) \longleftarrow (b3)$
  $\exists =_5 \ hasDisplayCount_b(HarryPotter) \longleftarrow (b4)$
  $\exists =_1 \ hasPrintCount_b(HarryPotter) \longleftarrow (b5)$
  $\exists =_7 \ hasUsageCount_c(HarryPotter) \longleftarrow (b6)$

6 Discussion

In Fig 3, the XML-based rights expression languages (RELs), such as ODRL, XrML, and P3P, are convenient for automatic machine (or agent) processing but do not have formal semantics to represent and enforce access rights permission. Therefore, policies based on these RELs to describe a license agreement (or contract) are usually written in Natural Language to indicate their meaning for the verification of access rights permission. As a result, these natural language policies sometimes are open to interpretation, which result in ambiguity of policy semantics. In order to remove this problem, people use FOL to represent and reason access rights control policies (see Fig 3). As we know, FOL-based policies have a formal and clear syntax and semantics, even these FOL-based policies usually have to limit their expressive power in order to capture those license agreements that are originally written in English. Unfortunately, policies shown as FOL always require policy writers and readers to be logicians. Furthermore, policies indicated as a generic full FOL may feature compliance checking that may be undecidable for their computation time.

To resolve this dilemma, we are going to explore the expressive power of different FOL-based policies representations to decide which conditions allow us to have both decidable and enforceable semantics capability of rights delegation policies. In order to have a decidable and tractable fragment of FOL-based policies to enforce respective compliance checking, we usually restrict policies as datalog Horn rules, where they are negation-free, function-free, and with limited number of parameter parities. Description Logic (DL) is a decidable fragment of

FOL and Logic Program (LP) is closely related to the Horn fragment of FOL. In general, a full FOL is undecidable and intractable even under the datalog restriction. As shown in [5], Description Logic Programs (DLP) is an expressive fragment of FOL and it provides a significant degree of expressiveness and substantially greater power than the RDF-S fragment of DL. Based on DLP, the Semantic Web Rule Language (SWRL) is considerably more powerful than either the OWL DL ontology language or the datalog Horn style rule language alone because SWRL extends OWL with the basic kinds of datalog Horn rule, which states as predicates are limited to being OWL classes and properties with a maximum parity of two, etc [9].

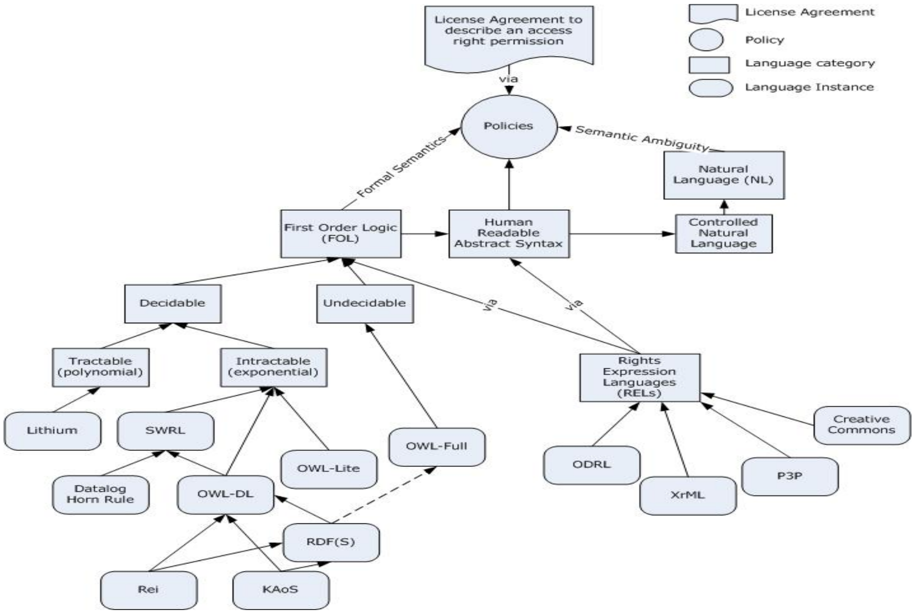


Fig. 3. A license agreement to denote access rights permission from a variety of policy language representations, such as Natural Language, Controlled Natural Language, First Order Logic (FOL), and Rights Expression Languages (RELs), etc

Policies in datalog Horn rules always assume that what is not explicitly permitted is forbidden. In that case, we can not distinguish forbidden access rights from unregulated access rights in a license agreement. Furthermore, we might need function capability in FOL-based policies to support translating English policies to FOL ones. Therefore, a tractable sublanguage, Lithium, with bipolars restriction, e.g., no bipolar literals in the FOL rules, was proposed in [8] to support its representation of denying policies and limited functions in their license agreement policies. Even though the Lithium policies are based on the relaxation of the datalog Horn rules, we still believe that this tractable policy language is only located somewhere in a small subset of FOL language. Therefore, it still lacks a large portion of OWL-DL and datalog Horn rule expressive power to serve both right delegation ontology and usage (or transfer) rights delegation rules, as shown in Section 4 and Section 5.2.

In [20], KAoS and Rei policy languages were shown as originally from DAML → OWL and RDF-S so it is quite trivial that these two policy languages are merely a subset of SWRL. Therefore, the expressive power of KAoS and Rei are less than SWRL because the rights delegation policies cannot be shown as a pure OWL-DL ontology language alone. In [12], Rei was extended to be a policy and delegation framework that includes inter-related resources, policies, policy languages, and meta-policies. However, authorization delegation policies were not explicitly seen in this study.

In this paper, we utilize the power of SWRL combined language to demonstrate the possibility of semantic-driven enforcement of rights delegation policies. A license agreement for a rights delegation policy is a policy set showing any number of prerequisites and relevant policies. A policy set is composed of facts, ontologies, and rules. These license agreements are distributed by distributor *Charlie* to consumers *Alice* and *Bob*. In this delegation scenario, the usage rights are applied to the entire *eBook* class instead of merely to the instance of *HarryPotter*'s *eBook*. In this policy-aware distributed DRM system, each trusted DRM node should faithfully enforce its rights delegation policies via its "compliance checking" inference engine.

There are several mutable facts in each node that express a prerequisite's dynamic status. These mutable facts will be updated and passed between distributor and consumers whenever a usage rights permission is granted and consumed. The mutable facts updating activity will be initiated as an Event-Condition-Action (ECA) reaction rule, where *event* might be triggered by a user's request or a message's arrival. The *condition* is specified in each relevant rights delegation rule and the *action* includes usage rights enforcement and mutable fact updating actions.

7 Conclusions

We have shown that the semantic formal model for an ODRL-based rights delegation policy can be enforced by expressing them as a combination of ontologies and rules. Based on ODRL's expressions and data dictionary, a rights delegation ontology is proposed in this study. Furthermore, we also express the rights delegation policy as a set of rules for usage and transfer (or duplicate) rights delegations. When verifying ODRL formal semantics, our SWRL approach is much more superior to the generic restricted FOL model because of the greater availability of a rights markup language and the higher expressive power of policy compliance checking from our SWRL language. A real usage rights delegation scenario is demonstrated in this paper to justify our formal semantic model.

References

1. Blaze, M., J. Feigenbaum, M. Strauss, Compliance Checking in the PolicyMaker Trust Management System, *Pro. of the Financial Cryptography 198.* LNCS, 1465, 1998, pp. 254-274.
2. Borgida, A., On the relative expressiveness of description logics and predicate logics, *Artificial Intelligence.* 82, 1996, pp. 353-367.
3. ContentGuard Inc. eXtensible rights Markup Language (XrML), Version 2.0.
4. Garcia, R., I. Gallego, and J. Delgado, Formalising ODRL Semantics using Web Ontologies, *2nd International ODRL Workshop.* Lisbon, Portugal, 2005.
5. Grosof, N. B., et al., Description Logic Programs: Combining Logic Programs with Description Logic, *WWW 2003.* Budapest, Hungary, 2003, pp. 48-65.
6. Guth, S., G. Neumann, and M. Strembeck, Experiences with the Enforcement of Access Rights Extracted from ODRL-based Digital Contracts, *DRM'03.* 2003.
7. Halpern, Y. J. and V. Weissman, A Formal Foundation for XrML, *Proc. of 17th IEEE Computer Security Foundations Workshop (CSFW'03).* 2003, pp. 251-263.
8. Halpern, Y. J. and V. Weissman, Using First-Order Logic to Reason about Policies, *Proc. of 17th IEEE Computer Security Foundations Workshop (CSFW'03).* 2003, pp. 187-201.
9. Horrocks, I., et al., OWL rules: A proposal and prototype implementation, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web 3.* 2005, pp. 23-40.
10. Iannella, R., Open Digital Rights Language (ODRL), W3C Note 19. September 2002, `http://www.w3.org/TR/odrl/`.
11. Kagal, L., T. Finin, and A. Joshi, A Policy Based Approach to Security for the Semantic Web, *ISWC 2003.* LNCS 2870, pp. 402-418, 2003.
12. Kagal, L., et al., Self-describing Delegation Networks for the Web, *IEEE Workshop on Policy for Distributed Systems and Networks, Policy 2006.* June 5-7, 2006.
13. LaMacchia, A. B., Key Challenges in DRM: An Industry Perspective, *DRM 2002.* LNCS 2696. 2003, pp. 51-60.
14. Li, N., B. N. Grosof, and J. Feigenbaum, Delegation Logic: A Logic-based Approach to Distributed Authorization, *ACM Trans. Information and System Security.* 6(1), pp. 128-171, 2003.
15. Lutz,C., Description Logics with Concrete Domains - A Survey, *Advances in Modal Logic.* Vol. 4, World Scientific Publishing Co., 2003, pp. 265-296.
16. Pan, J. and I. Horrocks, Web Ontology Reasoning with Datatype Groups, *ISWC 2003.* LNCS 2870, Springer, pp. 47-63.
17. Park, J. and R. T. Sandhu, The $UCON_{ABC}$ Usage Control Model, *ACM Transactions on Information and System Security.* 7(1), 2004, pp. 128-174.
18. Pucella, R. and V. Weissman, A Formal Foundation for ODRL, *Workshop on Issues in the Theory of Security (WITS).* 2004.
19. Stefik, M., Letting Loose the Light: Igniting Commerce in Electronic Publication, *Internet Dreams: Archetypes, Myths, and Metaphors.* MIT Press, 1996.
20. Tonti, G., et al., Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder, *ISWC 2003.* LNCS 2870, pp. 419-437, 2003.
21. Weitzner, D. J., et al., Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web, *Web and Information Security.* Idea Group Inc., 2006.

Access Control for Sharing Semantic Data across Desktops

[*] Juri L. De Coi, Ekaterini Ioannou, Arne Koesling,
Wolfgang Nejdl, Daniel Olmedilla

L3S Research Center/Leibniz Universität Hannover
Appelstr. 9a, 30167 Hanover, Germany
{lastname}@L3S.de

Abstract. Personal Information Management (PIM) systems aim to provide convenient access to all data and metadata on a desktop to the user itself as well as the co-workers. Obviously, sharing desktop data with co-workers raises privacy and access control issues which have to be addressed. In this paper we discuss these issues, and present appropriate solutions. In line with the architecture of current PIM systems [8, 2, 11, 15], our solutions cover all semantic data shared in such a context, i.e. all desktop resources as well as other data structures created by the system, such as metadata in an RDF store and inverted index entries created for efficient textual search. We discuss different kinds of policies to specify protection for desktop data and metadata, and describe our access control system to express and execute these policies efficiently. Additionally, we describe the extension of an existing PIM system, Beagle++, with our approach, as well as our experiments, with convincing results on performance and scalability.

1 Introduction

In recent years, the amount of available digital information has increased considerably not only on the Web but also on personal computers. New innovative Personal Information Management (PIM) systems support users in organizing and managing their calendars, e-mails, address books, and other information on their desktop. PIM systems like Google Desktop [8], Beagle [2], Haystack [11], and Gnowsis [15], define semantic data as all content of the personal information space. Semantic data thus include the actual desktop resources and all additional data structures the PIM system creates, such as extracted metadata, including all machine generated information describing the resources and appropriate data and index structures supporting the functionality of the PIM system. A promising extension of PIM systems is to move from the pure desktop data management system towards the sharing of information among different personal spaces and users [14, 4]. However, these systems are doomed to fail if they do not incorporate mechanisms to deal with privacy issues and access control, specifying and

[*] In alphabetical order

checking when and which semantic data is provided to whom. Appropriate mechanisms must allow users to control the sharing of both the actual resources and the metadata about them, as in many cases revealing the existence of a resource or even parts of the metadata is considered to be sensitive.

In this paper we discuss how to use policy languages [17, 12, 7, 3] to provide users with appropriate functionality to describe access control policies for their shared semantic data. To avoid expensive evaluation at run-time often incurred by such systems, we present an access control system that optimizes run-time execution of these queries, significantly reducing the response time and computer load of the personal computers queried.

The rest of this paper is organized as follows. §2 presents a motivating example and the requirements of an access control for semantic data. §3 presents different kinds of policies and explains how they can be efficiently executed using the access control mechanism we suggest. §4 describes our prototype implementation, and §5 presents the experimental evaluation we performed using this prototype. §6 discusses related work and §7 presents our conclusions.

2 Semantic Data Sharing

Let us consider Alice, who is an employee in a company aware of the great benefits of information sharing among co-workers. In this company, instead of large centralized repositories of information, a PIM system with sharing capabilities[1]. such as BEAGLE^{++} [4] is provided. Each user of the system has a *semantic desktop*, with a set of filters and generators to extract metadata from desktop resources (i.e., emails, publications), an RDF store to maintain this metadata, and an inverted index to allow full-text search. A graphical illustration of semantic desktops and semantic data sharing in our scenario is shown in Fig. 1

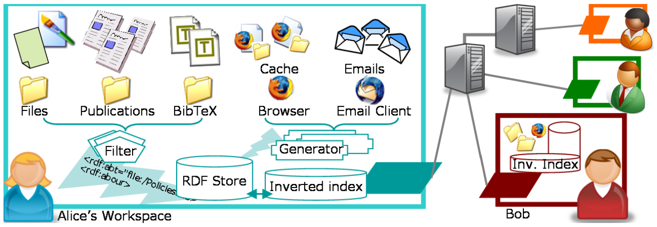


Fig. 1. An illustration of the semantic desktop architecture of the BEAGLE^{++} system and semantic data sharing between different desktops.

Alice has many resources on her desktop but she is not willing to automatically and unconditionally provide access to all her co-workers. Therefore, she creates policies to express the conditions under which she wants to share resources. For

[1] We assume that different desktops are connected with a P2P network such as Edutella [13], which does not require information to be shared among peers

example, for some project-related documents she states that only members of that project are granted full access to the resource, though the metadata about the title and authors are available also to non-members. Alice's co-workers are able to search for information on her desktop by sending queries to system. When Alice's semantic desktop receives these queries, her access control system ensures that all metadata and resources returned to her co-workers conform to the policies specified by Alice.

An access control mechanism for semantic data sharing between semantic desktops has some special requirements. One of the main requirements is typically to assume that everything is private by default, that is, nothing is shared unless explicitly stated otherwise. In our scenario, bad consequences of sharing sensitive data are more harmful than not sharing public information. Another important requirement is that the access control must consider two levels of protection, the metadata describing the resources and the resources themselves. Even if a user receives metadata about our resources —therefore knowing about its existence— that does not imply that the resource itself is publicly available. Also, it is required that normal employees (and not only qualified security administrators) must be able to personalize their policies, even if a default set of policies is provided. Furthermore, since the query execution will be performed at each employee's desktop, the access control mechanism must have a good performance.

3 Fine-Grained Access Control on the Semantic Desktop

This section describes the kind of policies considered in the paper as well as the main challenges addressed. It presents our solution which provides performant and fine-grained access control to information resources at run-time.

3.1 Specifying Policies

Policies specify the conditions that must be satisfied in order to grant access to semantic data. The policies described in our scenario can be classified into the following two main categories (examples are expressed with the PROTUNE policy language [3]):

(A) Resource Policies. These policies specify whether access to an actual resource (e.g., if the resource can be download) is granted or not. The conditions are described using the attributes found in the corresponding metadata. Some examples are listed in the following paragraphs.

Example 1. In our scenario, Alice gives access to any employee marked as co-author of a paper:

allow(access(file(Resource), Requester)) ⟵
  metadata(Resource, author, Requester).

(B) Metadata Policies. These policies state conditions under which different attributes from the metadata describing a specific resource can be disclosed.

Example 2. Another policy from our scenario states that only the subject field of e-mails not sent by her boss Tom are to be shared:

allow(access(metadata(subject, Resource), Requester)) ⟵
metadata(Resource, type, 'e-mail'), metadata('Tom', e-mail, TomAddress),
not metadata(Resource, from, TomAddress).

3.2 Query Processing and Policy Evaluation

When a request for information is received, the access control mechanism must evaluate all desktop policies and decide whether the semantic data to be delivered as search result can be disclosed. Remote requests (see Fig. 2) can be of two types: (a) *resource search* requests asking for metadata of resources relevant to a given query, and (b) *resource download* requests asking for retrieval of an actual resource. Resource search requests correspond to a user searching for resources matching a given query and therefore only return metadata about relevant results. A local inverted index is used in order to identify the resources relevant to the keywords of the query. For each resource, applicable policies have to be evaluated in order to decide whether a metadata field can be disclosed or not. The values of the set of granted fields for each resource are then retrieved from the metadata store and returned to the requester. For the resource download requests, the URI of a resource is given. The applicable policies for that resource are evaluated and the resource is sent back to the requester, if access is granted.

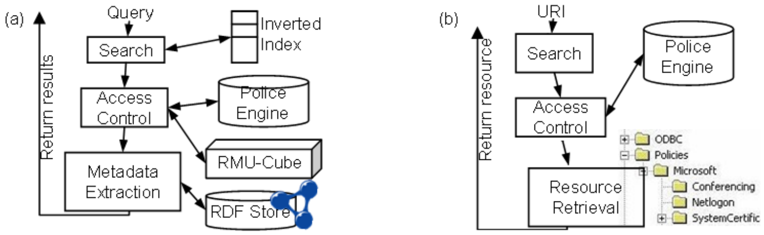


Fig. 2. The execution of (a) Resource Search and (b) Resource Download.

Obviously, for the resource search requests the system needs to evaluate applicable metadata policies whereas for the resource download requests, the system needs to evaluate applicable resource policies. Resource policies are applied only to a single given resource (the resource the remote user requested to download) and therefore the evaluation of the policies for that resource may imply an extra but acceptable performance cost. However, for metadata policies, this situation is quite different. For each relevant result returned by the inverted index, we need to check whether each metadata field can be disclosed or not. In order to make this decision, all applicable (potentially many) policies must be evaluated. Moreover, each one of these policies may imply complex conditions as well as execution of some actions such as queries to the metadata store, e.g. to fetch status of the document. It is clear that metadata policy evaluation only at run-time is too expensive and not feasible for our scenario.

The following sections present some optimizations to dramatically decrease the time required to evaluate which metadata fields are available for each resource to a given requester.

3.3 Optimize Metadata Policy Execution

During pure run-time metadata policy evaluation, all policies must be evaluated for each metadata field of each relevant result returned by the inverted index. This evaluation can be quite time consuming since a policy may involve several actions such as requests to a metadata store. Assuming that the metadata describing the resources are rather static —usually these metadata do not change every minute— we can exploit the fact that also actions performed during the evaluation of the policies will not change much over time. We can therefore improve evaluation costs considerably by pre-compiling the results of the evaluation of the policies for the parameters resource, metadata field being evaluated and requester.

Our solution uses a three dimensional bitmap named RMU-Cube [2] (Fig. 3). The first dimension (vertical in our figure) represents the resources found in our workspace. The second dimension (horizontal) represents the different metadata attributes available for resources. The third dimension (depth) represents the set of users that may act as requesters. This list of users can either be updated manually or possibly automatically by a remote service offered by the company (in order to keep it up-to-date with employees joining or leaving). Each cell of the RMU-Cube represents the result of the evaluation of all policies for a specific resource, metadata attribute and requester. The cells may take two values: access granted (represented by a 1) or access denied (represented by a 0).

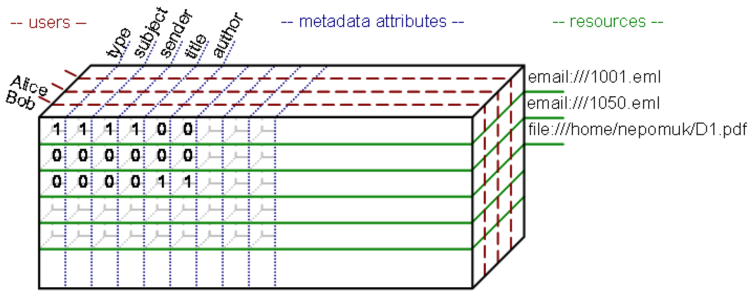


Fig. 3. The RMU-Cube represents the Metadata-Policies specified by Alice.

[2] RMU-Cube stands for Resource-Metadata-User Cube.

Example 3. Consider again Alice's semantic desktop. Among many others, it includes the following four resources: an e-mail *email:///1001.eml* which contains Tom in *cc*, another e-mail *email:///1050.eml* she sent to a colleague (without Tom in any field of the e-mail), and two documents *file:///home/nepomuk/finances.pdf* and *file:///home/nepomuk/D1.pdf* stored in the "/home/nepomuk" directory with status "Confidential" and "Final" respectively. According to these resources and the policies from §3.1 we can build the RMU-Cube depicted in Fig. 3. Metadata attributes that do not apply to a resource (e.g., cc in a normal document or author in an e-mail) are set to "access denied". The rest of the cells are set according to the pre-evaluation performed on the policies. Therefore, the subject of the e-mail *email:///1050.eml* can be shared with anyone, the title and author of *file:///home/nepomuk/D1.pdf* can be shared with anyone and members of the group Nepomuk may access all attributes of the documents *file:///home/nepomuk/finances.pdf* and *file:///home/nepomuk/D1.pdf.* The remaining metadata attributes are set to access denied.

Updating the RMU-Cube. An assumption when building the RMU-Cube is that resources, metadata attributes and users do not change so often that the updating mechanism of the RMU-Cube would overload the system. The removal of a resource, metadata attribute or user can be done quickly, since it only provokes the deletion of the corresponding plane. Additions or modifications are a bit more costly:

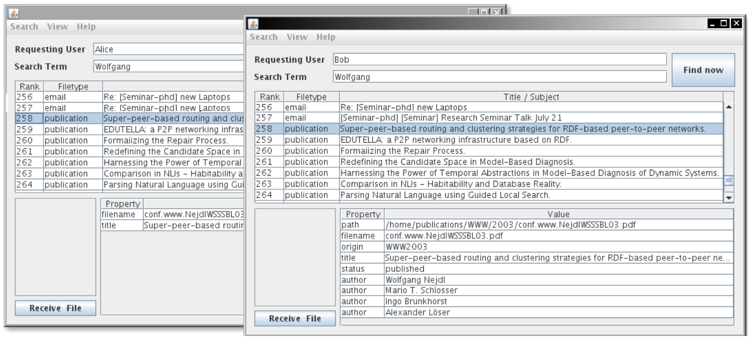
- Addition of a resource requires the creation of a new plane and evaluation of the applicable policies for each of its metadata attributes and potential requesters. Modification of existing resources does not imply an update in the cube unless some of its metadata is changed.
- Addition/modification of a metadata attribute requires the creation or re-evaluation of the corresponding plane according to applicable policies for each resource and potential requester.
- Addition of a user requires the creation of a new plane and evaluation of the applicable policies for each of the resources and its metadata attributes. As with resources, modification of a user (e.g., rename) does not imply an update in the RMU-Cube.

Assuming changes occurring as a result of normal user activity (e.g., editing of documents) and the further optimizations described in the next section, updates in our prototype can be performed without affecting the normal functioning of the desktop computer.

Finally, policies may change as well. If a policy is added, then all cells need to be re-evaluated. In case a policy language allowing only positive authorization policies is used (e.g., Protune), only the cells set to 0 need to be re-evaluated (adding a new policy may only result on more permissions). If a policy is modified or removed, then we need to evaluate all cells of the cube. These updates are costly and therefore should be grouped so they are performed at the same time.

4 Prototype Implementation

We have developed the concepts presented in previous sections using Beagle^{++} as the reference PIM system. We reuse the following components from the architecture of Beagle^{++}: an RDF Store that contains the metadata describing the existing desktop resources (we use *Sesame 2.0*[3]), the desktop ontology providing the schema for the metadata and the inverted index for performing full-text search (we use a relational representation of the inverted index using *MySQL 5.0*[4]).



Requesting User: Alice
Search Term: Wolfgang

Rank	Filetype	
256	email	Re: [Seminar-phd] new Laptops
257	email	Re: [Seminar-phd] new Laptops
258	publication	Super-peer-based routing and clus
259	publication	EDUTELLA: a P2P networking infras
260	publication	Formaiizing the Repair Process.
261	publication	Redefining the Candidate Space in
262	publication	Harnessing the Power of Temporal
263	publication	Comparison in NLIs – Habitability a
264	publication	Parsing Natural Language using Gui

Property	
filename	conf.www.NejdlWSSSBL03
title	Super-peer-based routir

Requesting User: Bob
Search Term: Wolfgang

Rank	Filetype	Title / Subject
256	email	Re: [Seminar-phd] new Laptops
257	email	[Seminar-phd] [Seminar] Research Seminar Talk July 21
258	publication	Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks.
259	publication	EDUTELLA: a P2P networking infrastructure based on RDF.
260	publication	Formaiizing the Repair Process.
261	publication	Redefining the Candidate Space in Model-Based Diagnosis.
262	publication	Harnessing the Power of Temporal Abstractions in Model-Based Diagnosis of Dynamic Systems.
263	publication	Comparison in NLIs – Habitability and Database Reality.
264	publication	Parsing Natural Language using Guided Local Search.

Property	Value
path	/home/publications/WWW/2003/conf.www.NejdlWSSSBL03.pdf
filename	conf.www.NejdlWSSSBL03.pdf
origin	WWW2003
title	Super-peer-based routing and clustering strategies for RDF-based peer-to-peer ne...
status	published
author	Wolfgang Nejdl
author	Mario T. Schlosser
author	Ingo Brunkhorst
author	Alexander Löser

Fig. 4. Searching using the prototype implementation: Alice and Bob search using the same keyword, but Bob is able to access more metadata than Alice.

Integrating our access control mechanisms with the above components requires the following additional components:

Policy Engine: We use the Protune policy engine in order to perform the evaluation of the policies. Its java API allowed us an easy integration with the rest of components. Protune policy engine uses several threads for a more efficient policy evaluation and allows the execution of external actions such as queries to RDF stores, relational databases or LDAP servers.

RMU-Cube: This component is responsible for maintaining and accessing the pre-evaluated information of the policy engine in order to speed up the process of deciding which metadata attributes are accessible for a resource (see §3.3). We represented this RMU-Cube structure in a relational database using *MySQL 5.0*.

User Interface: In order to use the system, we implemented the simple user interface shown in Fig. 4. The interface receives the keywords and the requester name and returns the available metadata of matching resources. If the user wants to access a specific resource, the "Receive File" button allows the user to download it. Access control mechanisms take place on both situations in order to enforce the specified policies.

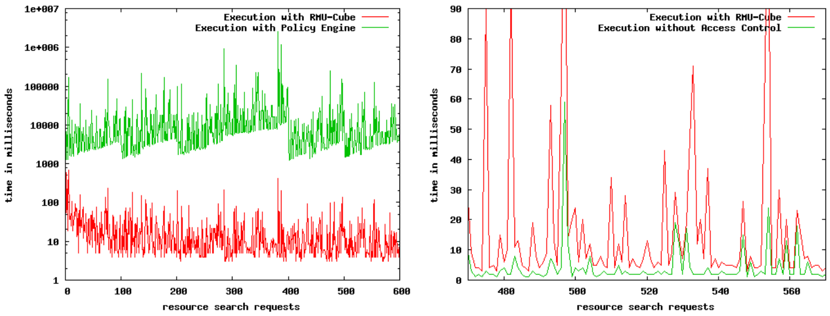
[3] http://www.openrdf.org/

[4] http://www.mysql.com/

5 Experimental Evaluation

We have performed several experiments in order to measure the impact of the concepts described in this paper. We used a large dataset including around 30 directories, 2200 publications from DBLP[5], and 2800 emails from the publicly available ENRON dataset[6]. Our dataset contained more than 5,000 resources and generated a total number of 72,974 triples in the RDF Store and 8,207 number of unique keywords in the inverted index.

In addition to this dataset, we implemented a similar scenario as the one presented in §2. We included a total number of 10 users and 20 policies protecting resources of the dataset. These desktop policies are similar to the ones presented in the examples of §3.1, granting access according to the attribute values of the metadata and the requesters/users defined by Alice[7].



(a) (b)

Fig. 5. (a) Comparison between time required to execute resource search requests using a policy engine and our RMU-Cube, and (b) total time needed for executing search requests without access control mechanisms or with our RMU-Cube.

We performed the experimental evaluation using a Pentium 4 computer with 1.5GB RAM. We executed a total of 600 resource search requests with different keywords which we randomly selected from the data in our inverted index. Fig. 5(a) shows the time needed for the execution of each one of the 600 requests with our RMU-Cube and with the run-time evaluation of the policy engine (we used a logarithmic scale for the time axis). As shown, the use of RMU-Cube dramatically reduces the time required to decide which metadata is allowed for each resource in comparison with the direct use of the police engine. Also, the integration of the RMU-Cube does not strongly impact the search mechanisms, since the difference between an execution without an access control and an execution with the RMU-Cube is only few milliseconds, as shown in Fig. 5(b)[8].

[5] http://www.informatik.uni-trier.de/~ley/db/

[6] http://www.cs.cmu.edu/~enron/

[7] The policies are available at http://www.L3S.de/~ioannou/SDPolicies/

[8] The peaks shown in the graph are produced by temporary overload while accessing the database in which the RMU-Cube is stored. However, average times show that

6 Related Work

In the last years we observed increasing popularity of systems for collaborative work and file sharing. The need for effective search within the increasing amount of information in this context pushed forward further development of search infrastructures for enterprise data management systems [10]. However, the sometimes private nature of such shared information makes it difficult to apply traditional document indexing schemes directly. The problem of applying traditional ranking algorithms to search through access-controlled collections is outlined in [5]. User access levels and access control have to be reflected in the index structures and/or retrieval algorithms as well as in ranking the search results.

In the literature, several solutions addressing the problem of privacy preserving of the data stored on public remote servers, which typically provide a basis for the community platforms, have been proposed. For example cryptographic techniques can enable users to store encrypted text files on a remote server and retrieve them using keyword search [9, 6, 16]. However, these solutions are not suitable for the collaborative multi-user environment. Alternatively, the data shared within a community can be stored locally by the user within an access-controlled collection. In this case efficient retrieval algorithms for search through access-controlled collections need to be provided to enable information sharing within the community. The authors of [1] address the problem of providing privacy-preserving search over distributed access-controlled content. Although this technique enables probabilistic provider selection it does not allow ranking of search results obtained from different document collections. Our semantically enriched community platform should allow providing unified view on the whole information set available to the user.

7 Conclusions and Future Work

Sharing desktop information requires scalable and effective access control mechanisms. In this paper, we have presented an approach that exploits the power of flexible and expressive policies and at the same time enforces them without impacting on the user's computer. Queries can be answered and information may be shared without perceivably increasing the response time of queries or overloading the personal desktop being queried. This approach is based on the pre-evaluation of the policies and its storage in a fast-accessible form (RMU-Cube), allowing for quick decisions for both the desktop resources and the metadata. Our experiments show how the use of an RMU-Cube dramatically reduces the computation and response time of enforcing access control on resources and metadata and how the integration of this mechanisms only provides a slightly higher response time than same queries without access control enforcement. We are currently optimizing our implementation and exploring and evaluating efficient techniques

the addition of the RMU-Cube supposes only some extra milliseconds in the process. Using e.g. an in-memory RMU-Cube representation would avoid such peaks.

for the update of this structure in order to face the evolution of desktop data and metadata. We are also integrating into a desktop agent which crawls the local files, extracts their metadata and index them in order to be shared.

References

1. Mayank Bawa, Roberto J. Bayardo Jr., and Rakesh Agrawal. Privacy-preserving indexing of documents on the network. In *VLDB*, pages 922–933, 2003.
2. Beagle search tool. `http://beagle-project.org/`.
3. Piero A. Bonatti and Daniel Olmedilla. Driving and monitoring provisional trust negotiation with metapolicies. In *6th IEEE POLICY*, pages 14–23, Stockholm, Sweden, June 2005. IEEE Computer Society.
4. Ingo Brunkhorst, Paul Alexandru Chirita, Stefania Costache, Ekaterini Ioannou Julien Gaugaz, Tereza Iofciu, Enrico Minack, Wolfgang Nejdl, and Raluca Paiu. The beagle++ toolbox: Towards an extendable desktop search architecture. In *Semantic Desktop Workshop 2006*, November 2006. Athens, GA, USA.
5. Stefan Büttcher and Charles L. A. Clarke. A security model for full-text file system search in multi-user environments. In *FAST*, 2005.
6. Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *ACNS*, pages 442–455, 2005.
7. Rita Gavriloaie, Wolfgang Nejdl, Daniel Olmedilla, Kent E. Seamons, and Marianne Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *1st European Semantic Web Symposium (ESWS 2004)*, volume 3053, pages 342–356, Heraklion, Crete, Greece, May 2004. Springer.
8. Google desktop. `http://desktop.google.com/`.
9. Hakan Hacigümüs, Balakrishna R. Iyer, Chen Li, and Sharad Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *SIGMOD Conference*, pages 216–227, 2002.
10. David Hawking. Challenges in enterprise search. In *ADC*, pages 15–24, 2004.
11. Haystack project. `http://haystack.lcs.mit.edu/`.
12. Lalana Kagal, Timothy W. Finin, and Anupam Joshi. A policy language for a pervasive computing environment. In *4th IEEE POLICY, 4-6 June 2003, Lake Como, Italy*, pages 63–. IEEE Computer Society, 2003.
13. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. Edutella: a p2p networking infrastructure based on rdf. In *WWW*, pages 604–615, 2002.
14. Nepomuk: The social semantic desktop. `http://nepomuk.semanticdesktop.org/`.
15. Leo Sauermann, Gunnar Aastrand Grimnes, Malte Kiesel, Christiaan Fluit, Heiko Maus, Dominik Heim, Danish Nadeem, Benjamin Horak, and Andreas Dengel. Semantic desktop 2.0: The gnowsis experience. In *International Semantic Web Conference*, pages 887–900, 2006.
16. Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.
17. A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In *POLICY*, page 93, 2003.

Logging in Distributed Workflows

Christoph Ringelstein and Steffen Staab

ISWeb Working Group
University of Koblenz-Landau
Universitätsstraße 1, 56070 Koblenz, Germany
{cringel,staab}@uni-koblenz.de
http://isweb.uni-koblenz.de

Abstract. Business needs are nowadays frequently realized by business workflows spanning multiple organizations. Current infrastructures, such as SOA, support this trend on the implementation side. As a side effect these issues of privacy and data protection arise, because data is shipped across organizational boundaries. At the same time increased awareness about protection of privacy and IPR have lead to comprehensive contractual and legal constructs - including the information of services consumers about the ways their data is handled. We propose to solve such information requests in widely distributed workflow executions by gathering the related information during the execution and attaching it directly to the processed data. Together with the data this information is passed through the workflow and at the end it is returned to the service consumer.

1 Introduction

Applying the paradigm of service-orientation in an organization implies the modeling of business capabilities as independent services. A service-oriented architecture provides standardized interfaces for the communication between services. The standardization enables the loose coupling of services, supports the service provisioning for external service consumers and eases the integration of external services into internal workflows. The resulting flexibility facilitates the combination of services from different organizations into one comprehensive, integrated workflow leading at the organizational level to an agile *virtual organization* [13, 5] that is able to adapt more quickly to new organizational or business needs.

However, the resulting flexibility also shows disadvantages. An integrated workflow may forward confidential data (e.g. personal data or intellectual property) between organizations potentially violating concerns of privacy protection or confidentiality. Under such circumstances of flexible interworking between organizations, the importance of accounting for actions performed on data may become legally and/or contractually a highest priority.

To enforce accountability the privacy laws of some countries, e.g. all countries in the EU (as defined in Directive 95/46/EC), oblige organizations to *notify* about the processing of personal data (which can be described by *privacy policies*) and entitles natural persons to request information about the processing of their personal data (this process is called *information by request*). As a second case, one may consider contractual obligations about data protection between organizations. An organization that uses a web service like Salesforce$^{\mathrm{TM}}$ customer-relationship management software may not want to share the data about its customers with its competitors who use the same service. Hence, a service provider may offer the possibility to retrieve data accounting information and thus to allow for some control of the data processing.

To be able to monitor agreed-upon policies the service consumer (e.g. a natural person or another organization) may request information about the processing of his data. The answer to this request must contain *who* processed the data as well as *why* and *how* the data has been processed. This information constitutes an abstracting log of the workflow execution. The log can be generated in different ways, e.g. by reconstructing or by monitoring the workflow execution. In any case the service provider needs a detailed overview of its workflows. Most frequently such an overview is lacking, even for internal workflows. For several reasons existing logging mechanisms, like the Extended Log File Format [7] or *syslog*[12], are not sufficient to gain a full overview of a workflow that is distributed among multiple organizations. The main reason is that existing logging mechanism are tailored to perform logging within one execution environment. Because the diversity of execution environments and the missing of standardized interfaces for exchanging logs, distributed logs can not automatically be combined into one log. To solve this problem we propose a logging mechanism[1] that is tailored to log all actions on a specific piece of data during its processing in a distributed workflow. To bind the logged information to the corresponding data instances the proposed mechanism attaches the logs directly to the data instances, as metadata. Therefore, according to the ‘sticky policy paradigm’ introduced in [9], we call this approach ‘sticky logging’. The sticky logging mechanism is designed to be used in a service-oriented architecture.

The aim of this paper is to present the actual state of work at the sticky logging mechanism. The mechanism consists of two parts: First, an architecture defining rules how to collect information about the processing of private data in distributed workflows; and second, a formalism specifying how to express the collected information. Parts of the mechanism to achieve accountability, like tamper and fraud resistance, are work in progress and will be presented in follow-up papers.

This paper is organized as follows: In section 2 we analyze general requirements for a mechanisms to collect information about the processing of private data in distributed workflows. Following the requirements we introduce the architecture in section 3 and in section 4 the formalism of the sticky logging mechanism. Then, we give a scenario in section 5 demonstrating the application of the sticky logging mechanism. Before we eventually discuss our approach and conclude in section 7, we compare it with related work in section 6.

2 Requirements Analysis

In this section we present a business case where the realization of an SOA architecture includes a workflow over several organizations and we identify legal requirements.

[1] This mechanism can be used in addition to standard logging mechanisms that are needed to maintain the technical functionality of a workflow environment or service.

From the business case and the legal requirements we derive some issues that arise with regard to data protection. Then, we derive technical requirements for collecting privacy-related information about data processing in a cross-organizational workflow.

2.1 Business Case

The Small-Books Inc. is a book-selling company. Parts of the logistics like storing the books and packaging are done by Small-Books Inc. itself, but for shipping and payment it uses services provided by other organizations. For instance, the shipping of orders is outsourced to a logistics company named Fast-Shipping Inc.

Assume that a customer, Mr. Smith, orders books via the Web site of Small-Books Inc. To place his order he has to insert his address and credit card number. After the order is placed Small-Books Inc. possesses three instances of data related to the order: the list of ordered books, Mr Smith's address and credit card number. As first action Small-Books Inc. uses the payment service of the credit card company. To this purpose, Small-Books Inc. passes the credit card number and the invoice value to the payment service. Then, Small-Books Inc. packages the order and invokes the delivery service of Fast-Shipping Inc. To this end, Small-Books Inc. has to copy the address to pass it as new data instance to Fast-Shipping Inc.

2.2 Legal and Contractual Requirements

In the introduction, we have described why an organization must take responsibility for its way of handling of personal data. In addition it must be able to inform the person concerned about this handling. These implies the following requirements:

Req1 *The person concerned must be able to request or directly access the logs.*
Req2 *The logs must identify the organizations responsible for each action performed on personal data.*
Req3 *An organization must not be able to deny a log entry it has made.*

2.3 Organizational Issues with Accountability

From the business case and above-derived requirements we may derive the following organizational issues.

Loosely-coupled Architectures: At the level of implementation, the use of a service-oriented architecture hampers the generation of an overview. The reason is that services in an SOA are defined and implemented independently of each other. Hence, cross cutting concerns, such as logging, are hard to realize if they are not standardized at the interface level. Also, workflows can be configured in an agile manner making it difficult *a posteriori* to assert which organizations had accessed the data during the execution of the workflow.

Lack of Process Awareness: In order to report on previous handling of data, an organization must be aware of and account for their internal data flows at a very fine-grained level of granularity. Such awareness and accounting at the fine-grained level is rarely available explicitly for cross-organizational workflows, because the increased complexity of the workflows [14].

Autonomy of Organizations: With organizational autonomy transparency decreases, as the autonomous organizations in general do not want their workflows to be controlled or monitored by third parties. At the level of cross-organizational workflow the issue of a lack of process awareness is further aggravated by the organizational responsibility being distributed over autonomous entities.

2.4 Technical Requirements

A technical solution for accountability must address the three organizational issues identified above. To solve these issues we demand the following technical requirements:

Req4 *To avoid ambiguities and to reach standardization, the logs must be formalized using a language with a well-defined semantics.*

Req5 *The used language must be able to express details about the performed actions, their actors, their purposes and their order. These details must have the required level of granularity.*

Req6 *A standardized interface is required to access and share logs between all involved parties (see [14]).*

Req7 *The logging implementation must enable an organization to create logs containing all privacy-related details.*

Req8 *Organizations must be able to hide process internals from third parties. Thus, both implementation and formalization must support security mechanisms.*

Additionally there exist other requirements, like trustworthiness of organisations. Even if those requirements are highly important for logging in general and for accountability in special, they are not considered in this work [2].

3 An Architecture for Distributed Logging

The above-identified issues affect the organizations ability to integrate external services in general and specifically to inform about actions performed on personal data. In the following we propose a logging architecture for distributed workflows to fulfill the above-defined requirements.

As demanded by requirement *Req1*, the person concerned must be able to access the logs about the processing of its data. However, to access the logs it is essential to know where the logs are located and how to access them. In addition, requirement *Req5* demands a detailed overview of the processing of the data. To fulfill these requirements we propose to attach all logs of a single execution of a workflow directly to the processed data, as metadata. The logs are attached by including them into the SOAP [6] messages that also transfer the associated data. Including the logs into the messages is reached by means of a SOAP module[3].

[2] Those requirements will be tackled in later work.

[3] The definition of the module is work in progress.

After the process execution the logs are made accessible to the person concerned by means of a mechanism we call back-propagation (see below). Because the logs are attached to the associated data instances, we call this approach *sticky logging*. The sticky logging mechanism demands specific actions when certain events occur. In the following a short overview of these actions is given:

- **Creation of data instances:** Each data instance gets its own sticky log starting with an entry recording various pieces of information (see section 4) about its creation. In addition, each sticky log gets its own UUID. If the created data instance contains data taken from other data instances, the source instances have to be referenced.
- **Processing and changing of data instances:** Each processing or changing of a data instance has to be logged by means of the formalism introduced in section 4.
- **Copying of data instances:** Each copying of a data instance is associated with the creation of a new instance of the data. To enable the back-propagation of logs and the updating of references (see below) in both sticky logs (the one attached to the original instance of the data and the newly-created one, which is attached to the new instance of the data) a reference to the other sticky log has to be made. A reference contains the UUID of the log and the identifier of the organization that possesses or receives (see passing of data instances) the data. After the data is copied further log entries are only added to the associated sticky log.

  If the newly-created data instance is used for purposes other than processing by the workflow (e.g. storing in a database), an alternative method for the back-propagation (e.g. e-mail, etc.) has to be specified. This method has to be used when the normal reference needed for back-propagation becomes unusable.
- **Merging of data instances:** To merge data instances the instances are processed (see above) with the purpose to create a new instance containing the result of the merge. The creation itself is handled like a normal creation (see above).
- **Passing of data instances:** To pass an instance of data (or parts of it) as parameter of a service call it (or parts of it) has to be copied (see above) and then the copy is passed to the called service. With the copy the newly-created log[4] is transfered, too. As mentioned before the transportation of the log is achieved as part of the SOAP message that is used to call the service and is transferred anyway.

  When a service is called synchronously, the log is back-propagated by means of the SOAP answer after the execution. In contrast, if a service is called asynchronously, this has to be logged explicitly in the log of the original data instance. In addition, because the SOAP answering message can not be used for back-propagation, an alternative method for the back-propagation has to be specified in the newly-created log (see above). In both cases, the organization has to integrate the log into the one of the original data instance after receiving the log.
- **Deleting of data instances:** Deleting a data instance does not cause the deletion of the sticky log. Instead the log has to be back-propagated. The back-propagation is reached by merging the log with the one referred by the reference specified in the log (see copying of data instances). The merging may be reached by means of

[4] Because only the new log is transfered, the provider of the called service gets no insight into internals of the calling organization.

directly accessing the referred log as long as both logs are possessed by the same organization. If the referred log is possessed by another organization, the log of the deleted instance may be returned by means of the SOAP answer. This is feasible as long as the deletion occurs during the execution of a service on behalf of the organization possessing the source instance. In the case that there is no SOAP answer (e.g. because the service has been called asynchronously) the alternate method is used for back-propagation. After merging the logs, all references contained in logs of direct copies of the deleted data instance have to be updated to refer to the merged sticky log.

If the deleted instance is the last instance of a specific piece of data, the log is directly returned to the person or organization that initially created the first instance of this data. This person or organization is responsible to answer requested information to the person concerned.

Beside those actions the sticky logging mechanism makes use of signatures and encryption. The signatures are used to assure that the log is not modified by another organization on its way. For this purpose each logging entity has to sign its log entries by means of a digital signature mechanism. The encryption of logs is possible, because, if the privacy law or the contract demands it, the logs may contain confidential information about the internals of an organization. This information has to be provided to the person concerned, but should not be accessible by other organizations that transfer the log back to the person concerned. We propose to use an encryption mechanism basing on a public-key cryptography algorithm. Such algorithms allow the logging organization to encrypt the logs with the public key of the person concerned or of organizations that are allowed to access the logs.

4 Logging Formalism

To enable a semi-automated analysis of logs we specify the logs by means of a RDF-based [10] semantic formalism. In the following we introduce parts[5] of this formalism.

Instances of Data: Each data may have multiple instances (e.g. through copying). Even if all instances have their own sticky logs, they have to be clearly identifiable. This is because the logs of different instances will be combined when a data instance is deleted (see below). To represent the instance of a data we introduce the **DataInstance**-class that has among others the following properties:

- **hasUUID:** A unique id that clearly identifies this data instance. To assure that the id is unique we use UUIDs [11].
- **isCopyOf:** If this is a copy, this property links to the original instance.
- **hasCopy:** If this data instance has been copied, this property links to the copy.

[5] The complete formalism is accessible at http://isweb.uni-koblenz.de/Research/SOA/StickyLogging

Listing 1 depicts a snippet of the log that is attached to the data instance of Small-Books Inc. containing Mr. Smith's address information[6]. In line 1 to 3 this instance is specified.

Actions on Data Instances: The actions performed on the data instances are represented by the class **Action**. This class has among others the following properties:

- **isOfKind:** The kind of this action.
- **hasPurpose:** The purpose why this action has been performed.
- **performedOnDataInstance:** The data instance the action is performed on.

Line 4 and 7 of Listing 1 depict how Small-Books Inc. logs that an action has been performed on the data instance containing Mr. Smiths address information.

Log Entries: The class **LogEntry** represents one log entry. Whereas, a log entry contains the recording of all information about one single action performed on the processed data. Some of the properties of the *LogEntry*-class are:

- **hasUUID:** A unique id that clearly identifies this log entry.
- **logsAction:** The action logged by this entry.

In Listing 1 lines 8 to 10 depict the beginning of a log entry recorded by the Small-Books Inc.

Kind of Actions: The introduced formalism builds upon the two basic actions that can be performed on data: reading and writing. These actions are subdivided into few sub categories representing different reasons causing the action. We distinguish two reasons read-actions can have: First, reading data to *use* it, e.g. by a service to fulfill its purpose. Second, reading data to *copy* it - including the copying of data to invoke another service. Although copying is a special kind of using data, we define copying as distinguished action. This is because of the specialty that a new instance of the data is created as result of the copy-action. Thus, we introduce the following classes representing these read-actions:

- The class **UseAction** represents all read-actions that are performed on the data, beside read-actions made to copy data.
- The **CopyAction**-class represents the read-action that is made to read the data that is to be copied.

In addition, we distinguish three reasons that cause write-actions: Writing data to *create* a new data instance, writing data to *change* an existing instance, and writing data to *delete* a data instance. The create-action has been distinguished from the change-action, because during the create-action a new data instance is created while the change-action only modifies an existing one. The write-actions are present by the following classes:

- The **CreateAction**-class represents the action that is used to create a data instance.

[6] In the following examples we use *sl:* as namespace for the classes and properties defined as part of the sticky logging formalism.

- All actions that change the data are represented by means of the class **ChangeAction**.
- The class **DeleteAction** represents the action performed to delete a data instance.

Within our business case Small-Books Inc. invokes the shipping service of Fast-Shipping Inc. to ship Mr. Smith's order. Therefore, it creates a new instance of the address data by means of a copy-action (see lines 3 and 6 in Listing 1). In the associated sticky logs Small-Books Inc. connects the two instances by means of the properties *hasCopy* and *isCopyOf*. In parallel Small-Books Inc. creates a new sticky log for the actions that will be performed on the copy of the address data (like Small-Books Inc. has done in lines 1, 2 and 8 to 10 in Listings 1 for the first instance). Then the new instance with its new sticky log is passed as part of the service invocation to Fast-Shipping Inc.

Purpose of Actions: Besides the kind of an action its purpose is of importance, to check if a performed action has been justified. The possible purposes of an action depend on the service and thus on its domain. For instance, actions of services of the delivery domain may have purposes like the delivery of an order, tracing of an order, etc. Thus, to enable a flexible definition of purposes, concepts defined in domain ontologies based on an upper ontology for privacy[7] are used.

Line 7 of the log snipped in Listing 1 shows that Small-Books Inc. has performed the above logged copy-action for delivery purposes (*dofd:DeliverOrder*[8]).

Accountability: Fundamental for achieving accountability is the explicit identification of the entity that performs actions on the data instance. In addition, an organization has to be clearly associated with all log entries it is responsible for and its log entries may not be modified or changed by another entity. Thus, we introduce the **Entity**-class to represent an entity within a sticky log. In addition, the formalism demands the use of mechanisms to sign RDF graphs like the one described in [3]. Among others the class *Entity* has the following properties:

- **hasName:** The name of this entity.
- **hasID:** A legally effective identifier of the entity (e.g. trade register number, international securities identifying number (ISIN), etc.).
- **hasLogged:** The log entries the entity is responsible for.
- **hasPGPCertificate:** A link to the entities PGP certificate.
- **Signature:** The signature that signs all log entries connected with this entity.

The lines 14 to 19 in the example in Listing 1 depict how the Small-Books Inc. identifies itself by instantiating the *Entity*-class and setting its properties.

```
01 :addressDI1 rdf:type    sl:DataInstance
02 :addressDI1 sl:hasUUID  ”a26f4580−39d9−11dc−a3fa −...”
03 :addressDI1 sl:hasCopy  :addressDI2

04 :action1 rdf:type                     sl:action
05 :action1 sl:performedOnDataInstance   :addressDI1
```

[7] Because of the limited space, this ontology will be introduced in detail in a technical report, which is under work.

[8] :dofd is the namespace of the domain ontology for delivery.

```
:action1  sl:isOfKind                    sl:CopyAction
:action1  sl:hasPurpose                  dofd:DeliverOrder

:logEntry1  rdf:type        sl:LogEntry
:logEntry1  sl:hasUUID      "cf2f30c0-39c1-11dc-80ae-..."
:logEntry1  sl:logsAction   :action1

:addressDI2  rdf:type      sl:DataInstance
:addressDI2  sl:hasUUID    "d3020270-3ab8-11dc-a179-..."
:addressDI2  sl:isCopyOf   :dataInstance1

:entity1  rdf:type                sl:Entity
:entity1  sl:hasName              "Small-Books Inc."
:entity1  sl:hasID                "ISIN US0001234567"
:entity1  sl:hasLogged            :logEntry1
:entity1  sl:hasPGPCertificate    "http://sbi.de/cert.asc"
:entity1  sl:Signature            "HrdSDFc..."
```

Listing 1. Example of a Sticky Log.

5 Scenario

In this section we step action-by-action trough our above-introduced business case to demonstrate the functionality of the sticky logging mechanism:

In our business case Small-Books Inc. possesses three data instances: One instance of the list of ordered books, one instance of the address data, and one instance of the credit card information. For each of these instances Small-Books Inc. creates a log. In each of these logs Small-Books Inc. identifies itself as logging entity. For all log entries Small-Books Inc. uses the above-defined semantic formalism (*Req4* and *Req5*).

As first step Small-Books Inc. processes the order list to package the order. This use-action and its purpose (packaging) are logged by Small-Books Inc. After the packaging is finished Small-Books Inc. hands the package over to Fast-Shipping Inc. To this purpose Small-Books Inc. copies the address data and transfers it to Fast-Shipping Inc. The transfered is reached by means of the SOAP message used to call the service. Thus, a standardized way to exchange logs is used (*Req6*).

The copy-action and its purpose (delivery) are logged by Small-Books Inc. In parallel a new log is created by Small-Books Inc. that is transfered to Fast-Shipping Inc. Both organizations need to be identified in the new log. Small-Books Inc. as responsible for the copy action and Fast-Shipping Inc. as receiver of the data instance. All organizations processing the data identify themselves in the logs (*Req2*). Because the delivery service is called asynchronous, a alternative return method is specified in the newly-created log.

As next step Fast-Shipping Inc. uses the address data to deliver the package. This use-action and its purpose (delivery) are logged by Fast-Shipping Inc. into the log attached to its instance of address data. For the logging Fast-Shipping Inc. uses also the above-defined semantic formalism (*Req4* and *Req5*). After the package is successfully delivered Fast-Shipping Inc. deletes its instance of the address data. The delete-action is also logged by Fast-Shipping Inc.

Before returning the log, Fast-Shipping Inc. signs the log entries it has recorded. Then Fast-Shipping Inc. returns the log by means of the specified alternative return method to Small-Books Inc. Small-Books Inc. integrates these log into its own log of the its instance of Mr. Smith's address data. At the end of the processing Small-Books Inc. signs its log entries also. All involved organizations had to sign their log entries (*Req3*). By means of the logs Small-Books Inc. generates an information page that is accessible by Mr. Smith. This Web page contains all information about the processing of Mr. Smith address data (*Req1*). In addition, all actions, their actors, and their purposes can be identified by Mr. Smith (*Req7*).

Finally, Fast-Shipping Inc. did not have insight into internals of Small-Books Inc. The other way Fast-Shipping Inc. could have used the private key of Mr. Smith to encrypt his logs. This way Small-Books Inc. also would have no insight into internals of Fast-Shipping Inc. (*Req8*).

6 Related Work

An example for another work identifying the need for accountability for data usage in distributed environments is presented by Weitzner et al. [15]. The authors find that access restrictions are not sufficient to reliably achieve policy goals, because a certain piece of information is often available at more than one point in the Web. Thus, it can not be guaranteed that a specific agent has no access to a certain piece of information. Therefore, the authors demand transparency of information usage to enable accountability.

To reach accountability, different approaches exist that propose the use of auditing mechanisms: For instance, the authors of [4] propose to use an auditing mechanism to achieve accountability, because the enforcement of policies is difficult in a distributed environment. Therefore, they introduce a framework consisting of a semantic policy language and an associated proof checking system. The policies are used to describe the permissions of agents. The auditing is done based on policies and logged actions, conditions and obligations. In difference to our approach the logs are located at the agent and not at the data. In addition, the logged information is used to audit the actions of the agent, while the purpose of our logging is the auditing of the entire processing of a certain piece of data. A similar approach is presented by Barth et al. [2]. They introduce a logic that can be used to specify and verify privacy and utility goals of business processes. In difference to our formalism their logic is not designed to be used in an inter-organizational environment and the logs are stored at the single agents and not with the data.

One example for an approach to log the communication of data is the Extended Log File Format [7]. The Extended Log File Format is one of the most prominent non-semantic logging formalisms for Web applications. This formalism is tailored to log the technical functionality of Web applications and their communication. Hence, the Extended Log File Format is not sufficient to describe actions performed on data and their purposes.

An approach to enrich existing logs with semantics is presented by Baker and Boakes in [1]. Their approach enables a more common understanding of the logs and thus helps to solve the problem of different taxonomies. However, this approach uses semantics only to enrich existing logs and thus no additional information is gained and the logs have still the restrictions identified above (e.g. missing connection with concrete workflow, etc.).

Karjoth et al. introduce in [9] the sticky policy paradigm. When data is transmitted to an organization via Web form the applicable policies and the users opt-in and opt-out choices are also included into the form. If the data is transferred to another organization, the sticky policies are transferred also. In difference to our work Karjoth et al. attach policies to data. Furthermore their focus is data collected via Web forms, while we consider data transferred at service calls. Another mechanism that attaches privacy-related information to data is the Platform for Privacy Preferences (P3P) [16]. However, the P3P formalism is restricted to policies and allows only the use of few pre-defined categories to describe the kind of data and purpose of its usage.

7 Conclusion and Further Work

This paper presented a logging mechanism for distributed workflows - called sticky logging. As part of the sticky logging mechanism we defined a well-defined, semantic formalism to specify logs. Besides the formalism we specified logging actions trigged by various events during the processing of personal data. In addition, we have described how the logs are shared and accessed by the person concerned. All together the sticky logging mechanism fulfills the requirements as demanded by privacy law and contracts. In addition, the sticky logging mechanism is able to overcome the above-mentioned organizational issues.

The next step of our work is the formal definition of the logging rules. After this we will analyze the integration of the sticky logging in an existing middleware platform. In addition, we are going to extend the sticky logging mechanism to achieve accountability. Therefore mechanisms for tamper resistance, avoidance of fraud, etc. shall be integrated. The complete sticky logging mechanism will be published by means of a technical report, which is under work. In parallel we are working on a prototype that implements the introduced sticky logging mechanism. Once the prototype is finished it will be made available for download.

8 Acknowledgements

This work has been supported by the Federal Ministry for Education and Research in the project SOAinVO - ”Technique Analysis and Risk Management for Service-oriented Architectures in Virtual Organisations” and by the european projects Knowledge Sharing and Reuse across Media (X-Media, FP6-26978) funded by the Information Society Technologies (IST) 6th Framework Programme.

References

1. Baker, M. A. and Boakes, R. J.: Semantic Logging Using the Resource Description Framework, presented in the ”Work-in-Progress Novel Grid Technolgies” track of CCGrid 2005, (2005)
2. Barth, A., Mitchell, J., Datta, A. and Sundaram, S.: Privacy and Utility in Business Processes. Proc 2007 Computer Security Foundations Symposium. IEEE. (2007)
3. Carroll, J. J.: Signing RDF Graphs. In: D. Fensel et al. (Eds.): The SemanticWeb - ISWC 2003, LNCS 2870, pp. 369-384, Springer, Berlin, Heildelberg (2003)
4. Cederquist, J.G. Conn, R. Dekker, M.A.C. Etalle, S. den Hartog, J.I.: An audit logic for accountability, In: Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on. June 2005, pp. 34-43 (2005)
5. Davidow, W., and Malone, M.: The Virtual Corporation. HarperCollins, New York, (1992)
6. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., Frystyk Nielsen, H. F., Karmarkar, A., and Lafon, Y. (Eds.): SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) - W3C Recommendation 27 April 2007. http://www.w3.org/TR/2007/REC-soap12-part1-20070427/ (July 2007) (2007)
7. Hallam-Baker, P. M., Behlendorf, B., Extended Log File Format, http://www.w3.org/TR/WD-logfile-960323.html, (May 2007)(1996)
8. Hanson, C., Kagal, L., Sussman, G., Berners-Lee, T., Weitzner, D.: Data-Purpose Algebra: Modeling Data Usage Policies, IEEE Workshop on Policy for Distributed Systems and Networks 2007. (2007).
9. Karjoth, G., Schunter, M., and Waidner, M.: Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data. In 2nd Workshop on Privacy Enhancing Technologies (PET 2002). Springer, (2002)
10. Klyne, G., and Carroll, J. J. (Eds.): Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, (June 2007) (2004)
11. Leach, P., Mealling, M., and Salz, R.: RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace. Internet Engineering Task Force. http://www.ietf.org/rfc/rfc4122.txt (July 2007) (2005)
12. Lonvick, C.: RFC 3164: The BSD syslog Protocol. Internet Engineering Task Force. http://www.ietf.org/rfc/rfc3164.txt (July 2007) (2001)
13. Nagel, R., and Dove, R.: 21st Century Manufacturing Enterprise Strategy. Lehigh, Pa.: Iacocca Institute of Lehigh University (1991)
14. Ringelstein, C.; Schwagereit, F., and Pähler, D.: Opportunities and Risks for Privacy in Service-oriented Architectures, to appear at the 5th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorporating the 3rd International ODRL Workshop Oct 11 - 13 2007 in Koblenz, Germany. (2007)
15. Weitzner, Abelson, Berners-Lee, Feigenbaum, Hendler, Sussman, Information Accountability, MIT CSAIL Technical Report MIT-CSAIL-TR-2007 (13 June 2007)
16. Wenning, R., Schunter, M. (Eds.), The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, http://www.w3.org/TR/2006/NOTE-P3P11-20061113/, (May 2007) (2006)

Recommendation Privacy Protection in Trust-based Knowledge Sharing Network

Weisen Guo[1,2] and Steven Kraines[2]

[1] Inst. of Systems Eng. of Dalian University of Technology, Dalian 116024, China
guows@dlut.edu.cn
[2] Division of Project Coordination of the University of Tokyo, Tokyo 277-8568, Japan
{gws,sk}@cb.k.u-tokyo.ac.jp

Abstract. Trust can be applied to knowledge sharing on a distributed network of knowledge source agents. Each agent represents a person who trusts some other agents. Based on these trust-relationships, an agent can infer the trustworthiness of an unknown agent by asking trusted agents for recommendations. However, the person represented by an agent may not be willing to share his or her individual opinion about the trustworthiness of a particular agent to agents that do not protect information privacy. A solution for this issue is proposed using three kinds of privacy policies: generosity, caution, and non-cooperation. An agent that adopts the caution policy towards another agent will hide the details of the trust recommendation path. An analysis shows the effect of the privacy policies on the calculated reliabilities of the recommended trust values[3].

1 Introduction

Creation of semantic descriptions based on ontologies could be an effective way to represent knowledge that is better suited for computer-based processing and matching. We have described a four level framework for an agent-based scientific knowledge sharing network of independently distributed knowledge repositories on the Internet [1][2]. The second level prescribes tools for allowing experts to create computer-interpretable semantic representations of their knowledge content. We have developed EKOSS [3][4], the expert knowledge ontology-based semantic search system to implement this level. An agent-based network of distributed knowledge repositories would then be populated by multiple distributed and independent EKOSS systems acting as agents.

In the knowledge network, each agent represents a person sharing or seeking knowledge. When AgentA receives a recommendation for AgentC from AgentB, the recommendation will be the actual opinion of the person represented by AgentB, an inferred trust value calculated from other agent recommendations,

[3] Steven Kraines is the corresponding author. This work is supported by the Japan Science and Technology Agency through the Shippai Chishiki Project, the National Natural Science Foundation of China (70431001) and the Liaoning Province Science and Technology Agency (20061063).

or some combination of the two. In particular, if AgentA knows that AgentB is the final link of a trust chain to AgentC, then AgentA knows that the recommendation is the opinion of the person represented by AgentB because AgentB is not using any other information to infer the recommended trust value. However, the person represented by AgentB may not want his or her opinions to be public.

In our previous work [5], in order to calculate inferred trust values of unknown agents with a high degree of accuracy, all of the trust information along each trust path is returned to the source agent. However, in this case, the source agent will be able to determine how much the target agent is actually trusted by the person represented by the agent giving the trust value of the final link of a trust chain as explained above. This paper introduces a set of privacy policies into the trust model and evaluates the effectiveness of those policies to enable the agents to protect the privacy of the persons that they represent on the trust-based knowledge network.

The rest of the paper is organized as follows. Section 2 discusses a number of trust models in the literature that are related to our study. The privacy policy used by our approach is described in Section 3. In Section 4, several scenarios demonstrating the performance of the trust-recommendation network generated by the system are analyzed. The analysis shows how the effectiveness of the privacy protecting trust inferring algorithm depends on the situation of the trust network. Finally, in Section 5 we conclude the paper with a description of future work.

2 Related Work

Privacy has been a hot topic since as early as the 19th century, when an influential paper “The Right to Privacy” was published. Recently, the primary focus of privacy has shifted from media privacy, territorial privacy, communication privacy, bodily privacy, to information privacy as technologies for information sharing have progressed. The first four aspects of privacy have been well established in most legal frameworks around the world; however, information privacy continues to create many problems today [6].

Goecks and Mynatt [7] noted that privacy is a critical social issue confronting Ubiquitous Computing that requires urgent attention. They proposed that the concepts of trust and reputation are critical to understanding privacy and building systems that enable users to effectively manage privacy. They created a trust network that calculated the reputations of entities in the network. Based on these reputations, users can manage how, when, and where they share their personal information. Their approach offered a new way to protect the privacy of user’s personal information and thereby addressed the problem of privacy protection in Ubiquitous Computing environments. We consider that a user’s trust information for another user is his or her personal information. So, the privacy of user’s trust information for others should be protected as well.

Our previous work introduced trust and recommendation concepts in a web-based knowledge sharing system. We presented the RTI algorithm to infer the trust value for the target agent from the recommendations of other intermediary agents in the trust chain to the target agent. The RTI algorithm uses negative as well as positive recommendations to accurately infer trust values of intermediary agents [5]. Most trust models do not try to evaluate the inferred trust value for the intermediary nodes in the trust chains [8]. However, if an agent, AgentA, is encountered in more than one chain from the source agent to the target agent, then ideally the source agent should give it the same trust value in each chain. The RTI algorithm infers the trust value for each intermediary agent in the trust chains to increase the accuracy of the inferred trust value for the target agent [5]. However, the RTI algorithm assumes that each agent will give all of the trust information that it has to a requesting agent. In this paper, we study the methods for protecting the privacy of trust information of individual agents.

3 Privacy Policy

In order to handle the issues that we presented in Section 1, we introduce a set of privacy policies in the RTI algorithm.

Like humans, when an agent receives a request for information on the trustworthiness of a target agent from a requesting agent, it should have the ability to decide what information it will return to the requesting agent. If the agent does not know or trust the requesting agent well, then it may not give any information to it. Even if the agent does know the requesting agent, if it cannot confirm that the requesting agent will protect its privacy, it may just return the information that it does not mind becoming public, such as an inferred trust recommendation for the target agent, and hide the detailed information of the recommendation chain. Only if the agent can confirm that the requesting agent will protect its privacy, will it return the trust recommendation together with all of the detailed information of the trust chain.

We provide three kinds of privacy policies to handle these three kinds of situations: generosity policy, caution policy, and non-cooperation policy.

When an agent AgentA receives a request for information on the trustworthiness of AgentB from AgentC and AgentA trusts that AgentC will protect its privacy by not giving the information to any unreliable agents, then AgentA will adopt the generosity policy and send to AgentC all of the trust recommendation information that it has that might be related to the trust chains to AgentB. If AgentA cannot confirm that AgentC will protect its privacy, but AgentA still wants to give some information to AgentC, then AgentA will adopt the caution policy and send only inferred trust values for AgentB, hiding the detailed information on the trust chain that led to the inference. If AgentA does not know or trust AgentC at all, then AgentA can adopt the "non-cooperation policy" and not give any recommendation information to AgentC.

Using the privacy policy, a person's privacy can be protected in the following way. As in our previous trust model, when the source agent wants to know the trustworthiness of an unknown target agent, it sends out a request for trust information to the agents that it trusts, specifying a maximum chain length n (step 6 in our previous trust model [5]). If the receiving agent does not have a direct trust value for the unknown agent, it will send out another request for trust information to the agents that it trusts with a maximum chain length n-1. When an agent receives a request for trust information with a chain length of 1, it means that the requesting agent is asking only for direct trust values for the unknown agent, so any value that the receiving agent sends back will be understood to be the actual opinion of the person represented by the receiving agent. As we noted earlier, a person's actual opinions about the trustworthiness of other people is a form of private information. Therefore, an agent asked for trust information for an unknown agent with maximum chain length 1 will only return a trust value if it adopts the generosity policy towards the requesting agent (unless the agent being asked for information is a dishonest agent that is trying to trick the requesting agent with false information).

An agent that is requested for trust recommendations with a maximum length greater than 1 can return a trust value it has for the unknown agent even if it does not adopt the generosity policy. This is because the requesting agent cannot determine whether the recommendation is from the person represented by the agent or an inferred trust value calculated from recommendations of other agents, and so the actual opinions of the person represented by the agent are protected. Furthermore, in order to make full use of the RTI algorithm, an agent A can return the information that it has about an agent B between it and the target agent. However, if agent B is just on link away from the target agent, then the agent receiving the information about agent B from agent A will know the opinions of person represented by agent B. Therefore, agent A should only give this additional information to agents that it trusts highly, i.e. that it can guarantee to agent B to be trustworthy.

In the implementation that we are constructing, each person represented by an agent on the trust network has an interface to set the privacy policy adopted by her agent towards each agent that is known. The agent would initially adopt a default privacy policy, such as the caution policy. Later, the person represented by the agent could change the policy based on her assessment of the trustworthiness of the person represented by the target agent. Because each agent uses different privacy policies for engaging with both known and unknown agents, our modified trust system implements a form of basic privacy protection similar to real human interactions that should provide significantly more accurate trust inference than conventional systems based on statistical analysis of recommendations irrespective of source.

4 The Analysis of RTI algorithm with Privacy Protection

First, we revisit the scenario that we described in the previous paper, reproduced in Fig. 1 [5]. The scenario has a social network composed of ten users each characterized as having high reliability (H), moderate reliability (M), low reliability (L), or as being dishonest (D). Sam is a dishonest service provider. For purposes of simulation, we assume without loss of generality that agents of users with high, moderate, and low reliabilities will give correct recommendations 90%, 80%, and 70% of the time, respectively, and the agent of a dishonest user will give opposite recommendations 90% of the time.

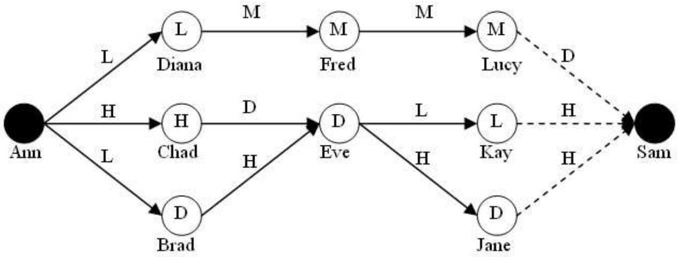


Fig. 1. The first scenario based on the scenario from [5]

The reliability of the service or recommendation trust value of an unknown agent can be calculated based on the recommendation trust values of the agents giving recommendations for the trust value of the unknown agent. We do this by giving each recommendation trust relationship a value between 0 and 1. Specifically, we give a low recommendation trust relationship a value of 0.7, a moderate trust relationship a value of 0.8, and a high trust relationship a value of 0.9. By quantifying the recommendation trust relationships in this way, we can combine recommendation trust values both in series (from the chain rule for Bayesian Networks) and in parallel (from the noisy-OR model for Bayesian Networks).

In this paper we add the privacy policies described in section 3 to the RTI algorithm, and we discuss the effect on the accuracy of the inferred trust values. If everyone adopts the generosity policy, Ann will receive all of the trust relationship information, including the identities and trust values of the target agents of each relationship. Therefore, Ann can calculate the inferred trust value of each inner agent in all trust chains and then use the RTI algorithm to obtain the most accurate inferred trust value for Sam.

If Brad and Chad both adopt the generosity policy towards Ann, then we have seen that the result will be the same as for the RTI algorithm with no privacy policy. If Brad adopts the caution policy towards Ann, and only Chad adopts the generosity policy, Ann will receive the information that Eve is dishonest from Chad and the recommendation that Sam is trustworthy from Brad. However, because Brad does not tell Ann that his recommendation came from Eve, Ann will use his recommendation, which she calculates as being more reliable than Diana's, and she will believe that Sam is highly trustworthy, which is incorrect. If Brad adopts the generosity policy and Chad adopts the caution policy, then Ann will receive all of the recommendation information from Brad. However, because Chad does not have a recommendation for Sam's trustworthiness since he believes that his only source, Eve, is dishonest, Ann will not receive any information from Chad. Because Ann does not have any information leading her not to trust Eve, she will believe the information from Brad, leading again to the incorrect result. If both Chad and Brad adopt the caution policy, then Ann will receive a recommendation that Sam is trustworthy from Brad and no information from Chad, so Ann will accept Brad's recommendation and again incorrectly believe that Sam is highly trustworthy. Therefore, only if both Brad and Chad adopt the generosity policy towards Ann will Ann be able to avoid being tricked by Eve through the application of the RTI algorithm.

In general, the addition of privacy policies to the trust model enables each agent to adopt a different kind of behavior towards each other agent based on the level of trust it has for the other agent. For example, if one agent has reason to believe that another agent asking for information is dishonest, it can choose to adopt the "non-cooperation policy" which in the RTI algorithm means that no recommendation is returned to the requesting agent. If the agent does not know anything about the requesting agent, it might adopt the "caution policy" as a default, giving the requesting agent only the minimum information needed to make the trust network work and hiding the path information of the other agents between it and the target agent. If the agent believes that the requesting agent is trustworthy, perhaps because another trustworthy agent has vouched for it, the agent can adopt the "generosity policy", in which case it returns the recommendation and the path information that it receives from all of the agents between it and the target agent. Then, the requesting agent can interpret all of the information that it receives in terms of the original RTI algorithm and calculate the inferred trust value for the target agent as described in [5].

Now, we consider a slightly more complicated situation where Eve knows more than two agents and where two of the agents she knows give the same trust recommendation for the target agent. The social network shown in Fig. 2 is composed of seven agents representing the Web users Ann, Brad, Chad, Diana, Eve, Kay, and Mary. Mary is a highly trustworthy service provider. There are eight relationships between the seven agents, forming three chains of trust links that connect Ann to Mary.

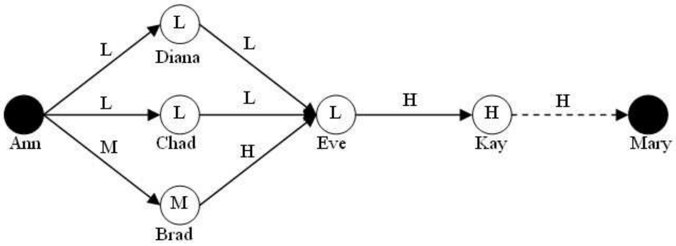


Fig. 2. The second scenario with three parallel trust chains

In this example, because Kay is the final user in all of the trust chains, Kay must adopt the generosity policy to Eve. Furthermore, if Eve adopts either the caution policy or the generosity policy to Diana, Chad and Brad, the final result will be the same. Therefore, we only need to consider the privacy policies adopted by Diana, Chad, and Brad to Ann. In here, we just consider the following two situations to illustrate the impact of the privacy policies to RTI algorithm.

If Brad and Chad adopt the generosity policy and Diana adopts the caution policy, then Ann will receive the single recommendation of "low trust" for Eve from Chad, and the single recommendation of "high trust" for Eve from Brad, but the recommendation from Diana for Mary will have the aggregated reliability value of $0.7 \times 0.9 = 0.63$ with no information about who gave the recommendation. Therefore, Ann will calculate the trust value for Eve to be "high trust" because Ann trusts Brad more than Chad for a single recommendation, i.e. Ann calculates the reliability of the second trust chain using H (0.9) replacing L (0.7) for the trust value of Chad to Eve. The final result for the reliability of the recommendation that Mary is highly trustworthy is $1-(1-0.7\times0.63)\times(1-0.7\times0.9\times0.9)\times(1-0.8\times0.9\times0.9) = 0.91$.

If Brad adopts the generosity policy, and Chad and Diana both adopt the caution policy, then Ann will receive the single recommendation of "high trust" for Eve from Brad. The recommendations for Mary from Chad and Diana will not have any information about who gave the recommendations. Therefore, Ann will assign the trust value for Eve to be "high trust", but she will use the aggregate reliabilities for the recommendations of Mary's trustworthiness from Chad and Diana of $0.7 \times 0.9 = 0.63$. The final result for the reliability of the recommendation that Mary is highly trustworthy is $1-(1-0.7\times0.63)\times(1-0.7\times0.63)\times(1-0.8\times0.9\times0.9) = 0.89$.

The analyses above show the effect of protecting privacy in the trust network. If the agents adopt the caution policy, information from some trust chains will be lost, and the accuracy of the inferred trust value will decrease. On the other hand, if an agent adopts the generosity policy, then it risks having its privacy information exposed. Our implementation of the RTI algorithm with privacy protection supports dynamic propagation of trust and privacy information in two ways. Whenever an agent receives trust recommendations from highly trusted agents for agents that it knows but does not trust, our implementation allows the agent to update the privacy policies accordingly. Alternatively, any time a person represented by an agent confirms that another agent is either trustworthy or dishonest, that person can manually assign an appropriate privacy policy. Furthermore, when an agent changes its trust level for another agent in either of these ways, it will send the new trust information to all of the agents to which it has adopted the generosity policy, resulting in a push style of trust information transfer. This push style information transfer will only occur between highly trusted agents adopting the generosity policy to each other. Each community of agents will adopt its own guidelines for balancing the risks of trusting a particular agent against the benefits of getting useful information from that agent using all of the trustworthy information that is available to it, much the same way that humans interact in society. Our hope is that this will result in the establishment of reliable recommendation networks where a recommendation for a particular agent will be updated quickly among the highly trusted peers of the recommending agent. Because each agent knows that if it is dishonest, its malicious reputation will be rapidly spread through the peer-based connections of the network, we hypothesize that most agents will be motivated to stay honest and friendly. In this way, we propose that the privacy policies in the trust network will result in a dynamic equilibrium where most agents are honest and adopt generosity policy between each other, which forms a robust network of trust recommendation that rapidly exposes dishonest agents, keeping their numbers down. Then, a high accuracy of inferred trust can be maintained while simultaneously protecting privacy.

5 Future Work

We plan to continue our research along several directions. First, we will create a trust network that closely simulates real social networks by exhibiting characteristics such as small world behavior. Based on that, we will conduct simulation studies to analyze how the different trust metrics work. Second, we are investigating methods for integrating the trust-recommendation network with the EKOSS knowledge searching and matching system in order to share different quantities and qualities of knowledge with agents that have different trust values.

References

1. Kraines, S.B., Batres, B., Koyama, M., Wallace, D. R., and Komiyama, H., Internet-based collaboration for Integrated Environmental Assessment in Industrial Ecology - Part 1, Journal of Industrial Ecology 9(3) (2005) 31-50.
2. Kraines, S.B., Batres, R., Kemper, B., Koyama, M., and Wolowski, V. (2006) 'Internet-Based Integrated Environmental Assessment, Part II: Semantic Searching Based on Ontologies and Agent Systems for Knowledge Discovery', Journal of Industrial Ecology, Vol. 10, No. 4, pp.1-24.
3. EKOSS site, (http://www.ekoss.org) World Wide Web.
4. Kraines, S., Guo, W., Kemper, B., and Nakamura, Y., EKOSS: A Knowledge-User Centered Approach to Knowledge Sharing, Discovery, and Integration on the Semantic Web. Proc. of ISWC 06, LNCS 4273/2006, pp. 833-846
5. Guo, W., Kraines, S., Inferring Trust from Recommendations in Web-based Knowledge Sharing Systems. Adv. in Intel. Web, ASC43/2007, pp. 148-153.
6. Langheinrich, M., Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. Ubicomp 2001, LNCS 2201/2001, pp. 273-291.
7. Goecks, J., and Mynatt, E., Enabling Privacy Management in Ubiquitous Computing Environments through Trust and Reputation Systems. Proc. of the 2002 ACM Conference on Computer Supported Cooperative Work, New Orleans, LA, USA.
8. Golbeck, J., Hendler, J., Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks. LNCS, 3257/2004, pp. 116-131.

Privacy Enforcement in Data Analysis Workflows

Yolanda Gil[1], William K. Cheung[2], Varun Ratnakar[1], Kai-kin Chan[2]

[1] Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey CA 90292, United States
{gil, varunr}@isi.edu
[2] Department of Computer Science, Hong Kong Baptist University,
Kowloon Tong, Hong Kong
{william, kkchan}@comp.hkbu.edu.hk

Abstract. Collaborative e-Science projects commonly require data analysis to be performed on distributed data sets which may contain sensitive information. In addition to the credential-based privacy protection, ensuring proper handling of computerized data for disclosure and analysis is particularly essential in e-Science. In this paper, we propose a semantic approach for enforcing it through workflow systems. We define privacy preservation and analysis-relevant terms as ontologies and incorporate them into a proposed policy framework to represent and enforce the policies. We believe that workflow systems with the proposed privacy-awareness incorporated could ease the scientists in setting up privacy polices that suit for different types of collaborative research projects and can help them in safeguarding the privacy of sensitive data throughout the data analysis lifecycle.

Keywords: Workflow generation, scientific workflows, privacy, trust.

1 Introduction

Trust and security were always central to the vision of the Semantic Web [1]. In a recent paper, Weitzner et al. [2] argue for a policy-aware infrastructure for the Web that ensures privacy and other social needs that would encourage people to share information freely. They also propose developing systems that are transparent and accountable [3] regarding their use of sensitive data from individuals and therefore can demonstrate their compliance with existing privacy laws.

The Web has always raised concerns for privacy data. There is concern about the wide availability of yellow pages and other directory information, and the fact that protected or sensitive information may become available over the Web perhaps unintentionally [4]. Of particular concern are record linkage techniques to cross-reference independent data sources and data mining algorithms that detect patterns, associate them with individuals, and reveal private or sensitive information about individuals that may violate basic privacy rights.

1.1 Motivation: Privacy in e-Science

Although privacy has broader interest and applicability, our research arises and focuses in the context of e-Science applications. Many areas in biomedical sciences envision benefit from clinical records (e.g., cabig.nci.nih.gov), phenotype information, and health history. In social and behavioral sciences, widely available on-line information can be integrated and analyzed to reveal significant patterns that emerge in specific communities, influential groups and individuals within a social network, and trends or events of interest. Much of this research is hindered because of the concern of individuals with their privacy and therefore their reluctance to allow the use of their personal data. Yet, many people would choose to give up their privacy for some greater good such as advancing medical research, especially when they are provided with mechanisms to protect the privacy of their data [6].

A variety of mechanisms are being investigated to ensure data privacy including secure data storage, data access control, auditing mechanisms, and securing lines of communication. Also, laws and policies for protecting and enforcing health information privacy will need to be formulated in order to determine how those technologies need to be used to implement the law. These mechanisms are important and necessary to control the access and release of data. However, they will not necessarily support the anticipated sophistication of people's wishes over the *fine-grained control* over the *uses* of their sensitive data, say, for clinical data analysis conducted by some third parties. Furthermore, the control is further complicated by the recent trend that the uses of sensitive data are no longer confined to the institution that collected or owns the data but *highly distributed* (e.g., cabig.nci.nih.gov).

1.2 Privacy Protection in Workflow Systems

In recent years, a variety of workflow systems have been developed to manage complex scientific analysis processes [5]. We see workflows as an artifact that captures, among other things, how data is being transmitted, pre-processed and analyzed, and for what purpose. Of particular concern for us is to enforce privacy protection in workflows by enabling workflow systems with privacy-awareness. Workflow systems can represent detailed models of the individual computations performed in the data, and be extended to express their privacy-related properties. In recent years, a variety of algorithms and approaches for privacy-preserving data analysis are being developed [8], where some transform data into privacy-preserved versions before putting together for subsequent analysis while some compute intermediate analysis results via a distributed and secure protocol. With these kinds of approaches, data sets can be processed and analyzed with well-defined guarantees as well as risks about the preservation of privacy of individuals. Thus, the already complex data analysis processes are now further complicated by the need and at the same time possibility to have data privacy protection integrated. The use of the semantic approach has been demonstrated to be effective in assisting users in creating and validating complex data analysis workflows, e.g., for large-scale earthquake data analysis [10], [11].

1.3 Our Contributions

We take a semantic approach to incorporate privacy awareness into workflow architectures and our implementation in the Wings/Pegasus workflow system [10]. The focus of our work to date has been on privacy policies that need to be addressed when workflows are designed and created. In particular, we show how the workflow systems could be extended to be able to *detect privacy policy violation* and to *provide corrective actions* for revising the workflows before the data analysis process can be safely executed.

The paper begins with ontological representations for privacy-relevant terms in data analysis workflows and illustrate how those ontologies could be used to describe workflow systems that incorporate traditional data analysis algorithms and privacy-preserving algorithms for analyzing sensitive data. To support automatic privacy policy enforcement in data analysis workflows, we propose a particular policy representation which has components describing applicable context, data usage requirement, privacy protection requirement, and corrective actions if the policy is violated. We present initial results on extending a workflow system to include representations of privacy policies that can be enforced by the system. We finalize with a discussion of related work and possible avenues for future research in this area.

2 Ontological Representations of Privacy-Relevant Terms in Data Analysis Workflows

Figure 1 depicts an ontology that contains core workflow concepts typically used to represent workflows and the extensions needed for describing privacy relevant concepts in data analysis workflows (shown in bold face). The classes shown in normal face are adopted from [10] for constructing workflows, including a *file ontology* for representing datasets, a *component ontology to* represent computations that correspond to steps in the workflow, and a *workflow ontology* to represent data-independent workflow templates. Unlike other scientific workflows that are composed of web services, Wings/Pegasus workflows being consider in this work are composed of codes that can be submitted for execution in a resource selected by the workflow system [10].

Privacy Preservation Ontology. This ontology includes a *PrivacyPreservation* class of privacy preservation methods that convert the input into privacy preserved forms. Privacy preservation methods can process on each attribute individually or the data set as a whole. *PrivacyPreservationPerAttribute* contains component types such as *Anonymization* (e.g., masking, substitution) and *PrivacyPreservationPerDataset* contains *Generalization* (e.g., k-anonymity [8]).

Data Analysis Ontology. It provides a separate taxonomy of data analysis methods. We consider here statistical data analysis algorithms that are widely used in many domains. In our ontology, *DataAnalysis* is the root class with subclasses like *Clustering* (e.g., Gaussian mixture model), *Classification*, etc.

Extensions of the file ontology. We extend the ontology to describe data protection up to *per-attribute* level. Some additional properties and classes added include:

- *hasAttribute* whose range captures data attributes described as *Attribute*.
- *hasAuthorizedUse* which refers to the intended use or purpose of the File.
- *Attribute* which models file attributes and has a property *protectedBy* (with sub-properties, e.g. *anonymizedBy*) to indicate the adopted privacy preservation method.
- special types of *File,* e.g., *DataSet* for raw data files and *Clusters* for clustering results which can go with data items (*ClustersWithDataItems)* or just per-cluster statistics (*ClustersWithStatistics*). The latter is needed when data privacy is an issue.

Extensions of workflow template ontology. Some properties added include:

- *hasPurpose* which refers to data analysis purpose.
- *hasOutputQuality* which refers to overall output quality descriptors, e.g. accuracy.

Extensions of component ontology. Some properties and subclasses added include:

- *hasParameterSet* which refers to the set of parameters needed by the component.
- *PPComponentType* which contains privacy preservation methods as its sub-classes, e.g., *Generalizer* (which in turn has sub-classes, e.g. *k-anonymity*)*,* and has a property *hasLevelOfProtection* for describing the level that its output is protected.
- *DAComponentType* which contains data analysis methods as its sub-classes, e.g., *Clustering* (which in turn has sub-classes, e.g. *GMM*), and has *supportPPType* and *supportDataType* to indicate its supporting types of privacy preservation and data.

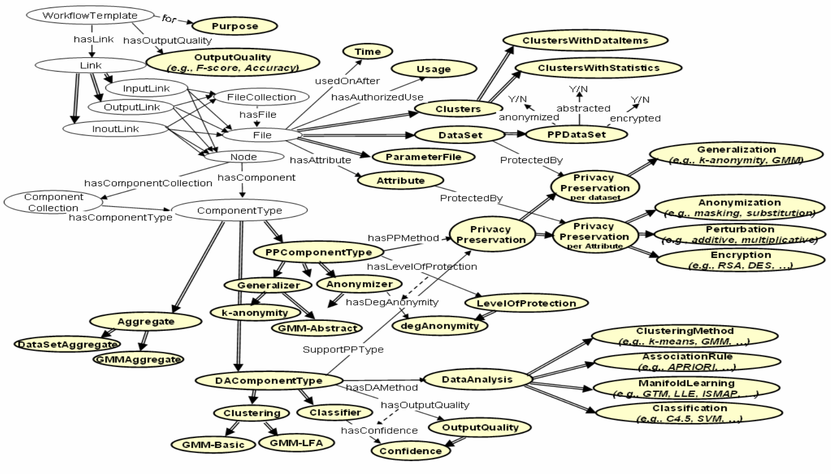


Fig. 1. An ontology for describing privacy aware data analysis workflows.

To illustrate how a domain-specific data analysis workflow can be described, we adopt a hypothetic *clinical data analysis* task. Like many other domains, clinical data can contain patents' personal identification and demographic information as well as sensitive ones including medical measurements, medical treatment, drug dosage, diagnosis, etc. We assume data collected from patient records archived at different clinic to (1) have the personal identification fields anonymized, (2) be generalized into groups based on their demographic information by k-anonymity and (3) be abstracted up to an agreed level of details based on the numerical medical attributes (e.g., by GMM [12]). Clustering is then carried out to identify patterns in different patient groups. Fig. 2 shows a related workflow together with a corresponding domain-specific workflow template created using Wings.

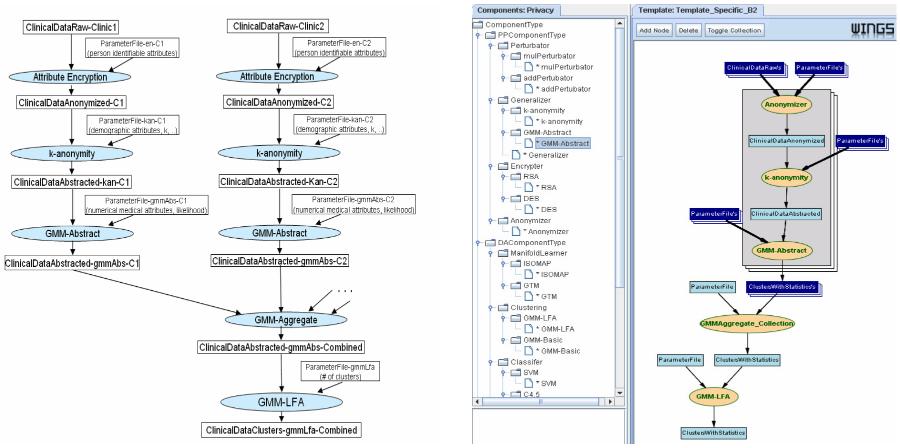


Fig. 2. A clinical data clustering workflow template represented in Wings.

3. A Privacy Policy Representation and Its Enforcement

We represent data privacy policies semantically based on the derived ontologies so as to support automatic policy enforcement in data analysis workflows via reasoning. Note that the privacy awareness being considered here is not conventional credential-based authentication and authorization. Instead we require a policy language that is flexible enough to describe conditions reflecting different relationships among data sets, components, and their privacy-relevant properties. Such a flexibility requirement naturally leads us to the use of rule-based representation. Note that other than expecting the users to specify the rules, rule-based policies carefully created by experts of the respective field can always be adopted.

In our current design, a policy representation contains four parts, namely (1) ***context***, (2) ***usage requirement***, (3) ***protection requirement***, (4) ***corrective action***. Informally, *context* specifies what workflows the policy applies. As we are dealing with data privacy, the context refers to some types of links, data or components where the policy is applicable. *Usage requirements* and *protection requirements* are for detecting policy violation within the context. Finally, *corrective actions* are suggestions for remedy of the policy violation, typically referring to the statements mentioned in the protection requirements. We further characterize requirements to be of *positive* and **negative** types. Positive requirements specify compliance conditions and policy violations occur when the conditions ARE NOT satisfied. Negative requirements specify non-compliance conditions and policy violations occur when the conditions ARE satisfied. As seen in the following, both types of requirements are essential for policy representation.

Context refers to the condition where the underlying policy is relevant. In other words, the policy applies only if this condition is satisfied.

- Example C1: *"Input files of a workflow containing medical images."*
  InLink(?l) ^ hasFile(?l, ?d) ^ hasAttribute(?d, ?a) ^ MedicalImage(?a)

Usage requirement refers to the non-amendable condition under which the use of data is required (+ve) or not allowed (-ve).

- Example UR1 (+ve): *"It is required that the purpose of the workflow should be equal to the authorized usage of the inputs that match the context."*
  for(?w, ?pw) ^ hasAuthorizedUse(?d, ?pw) ^ equal(?pw, ?pd)

Protection requirement refers to the *condition* when the use of data is required (+ve) or not allowed (-ve) with respect to data protection and analysis quality.

- Example PR3 (-ve): *"It is not allowed that the nodes that match the context have inputs with attributes in common."*
  hasAttribute(?d1, ?a1) ^ hasAttribute(?d2, ?a2) ^ equal(?a1, ?a2))

Corrective action refers to the remedies recommended to fix policy violation. For "usage requirement" violation, only a printed message stating the violating policy is expected as no remedy is possible. For "protection requirement" violation, a corresponding recommended action for fixing the violation will also be provided.

Policy Compliance Checking Via Reasoning We create 2 rules for each policy: a context component rule to locate where the policy applies and a requirement component rule to determine if non-compliance conditions occur within the context.

For the policy with a *negative* requirement, its context component rule and requirement component rule can simply be combined by conjunction and applied to a workflow description. Thus, the overall rule becomes [context rule] ^ [requirement rule] -> invalid (?l). Matched results will correspond to the policy violation situations.

For the policy with a *positive* requirement, the overall rule for detection problematic parts can be represented as [context rule] ^ not [requirement] -> invalid (?l). However, the rule becomes not a horn clause and thus cannot be easily represented using SWRL. Thus, instead of applying directly the overall rule, we apply the context rule first to the workflow and the matched results form a set with items of concern. Then, we applied the requirement rule to the set. The newly matched items are removed from the set in context and the remaining ones are the violation situations. This treatment works when the policies are free of conflicts among them. If there are some parts in the workflow with more than one policies applicable, policy conflicts will occur. We are currently investigating algorithms for policy conflict detection and resolution [15].

A Compliance Checking Walkthrough Given the workflow templates discussed in Section 4.1, two particular policy rules expressed in SWRL are considered:

General Policy G1:*"For all the inputs, it is required that the purpose of the workflow should be equal to the authorized usage of the inputs."*

context:	WorkflowTemplate(?w) ^ for(?w, ?l) ^ hasFile(?l, ?d)
usage:	+ve: for(?w, ?pw) ^ hasAuthorizedUse(?d, ?pdl) ^ equal(?pw, ?pd)
protection:	NULL
correction:	prompt [workflow and data purpose mismatch]

Domain Specific Policy S1: *"For data that contain dosage information, it is not allowed that they are not first anonymized before being used for analysis."*

context:	hasLink(?w, ?l) ^ hasFile(?l,?d) ^ hasAttribute(?d, ?a) ^ Dosage(?a)
	hasDestinationNode(?l, ?n) ^ hasComponent(?n, ?c) ^ DAComponent(?c)

usage:	NULL
protection:	-ve: anonymized(?d, ?aVal) ^ equal(?aVal, false)
correction:	prompt [add an anonymization step right after (?d) found at (?l)]

Suppose a researcher creates a simple workflow template that takes directly all the raw clinical datasets and feeds them into a basic GMM clustering component to perform a clinical study. The workflow system would find that policy G1 applies and is respected. However, policy S1 is fired as the aggregate dataset fed to the GMM-basic was found not to be anonymized. Fig. 3 shows the detection of the violation of the policy S1. The workflow in Fig. 2 complies with all these policies. In [13] we describe an interactive scenario where users would be assisted during workflow construction to create workflows that comply with a set of privacy policies.

4 Related Work

To the best of our knowledge, there has not been prior work on extending workflow systems with data privacy awareness. Some policy frameworks like KAoS [14] and Rei [15] have recently been proposed for security and privacy on the Semantic Web. To contrast with KAoS and Rei, our data privacy policies in data analysis workflows need to refer to properties of data, components, etc. In addition, the policies of concern are not credential-based ones as those in KAoS and Rei. Also, the policies we use not only are aimed to detect violations but also to suggest corrective actions in terms of how to fix the causes of violation.

5 Conclusions

In this paper, we motivated the need for a new type of privacy policies that constrain processing on data. We described our initial work on a semantic approach to represent privacy policies relevant to data analysis. We argued the validity of the approach by showing how privacy-preserving data analysis processes can be defined using ontologies, and how the ontologies can be combined with a policy framework to represent the policies. We discussed how those policies can be applied via examples. Future work includes conflict detection algorithms for the proposed policy framework and incorporation of the policy enforcement module in the Wings system.

Acknowledgement

This research was supported in part by the Air Force Office of Scientific Research (AFOSR) through grant FA9550-06-1-0031. The second author was a Visiting Scholar at ISI, USC while this paper was written, and he thanks Hong Kong Baptist University for a sabbatical leave.

References

1. Berners-Lee, T., Hall, W., Hendler, J., O'Hara, K., Shadbolt, N., Weitzner, D. "A Framework for Web Science." Foundations and Trends in Web Science, Vol 1, No 1 (2006)
2. Weitzner, D.J., Hendler, J., Berners-Lee, T., and Connolly, D.: Creating a Policy-Aware Web: Discretionary, Rule-Based Access for the World-Wide Web. In Web and Information Security, E. Ferrari and B. Thuraisingham (Eds), IRM Press (2005)
3. Weitzner, D.J., Abelson, H., Berners-Lee, T, Hanson, C., Hendler, J., Kagal, L., McGuinness, D.L., Sussman, G.J., Waterman, K.K.: Transparent Accountable Data Mining: New Strategies for Privacy Protection. Technical Report, MIT-CSAIL-TR-2006-007, MIT (2006)
4. Sweeney, L.: Finding Lists of People on the Web. ACM Computers and Society, 34(1) (2004)
5. Taylor, I.J., Deelman, E., Gannon, D., and Shields M.S. (eds.). Workflows for e-Science. Springer Verlag (2006)
6. Mandl, K.D., Szolovits, P., and Kohane, I.S.: Public Standards and Patients' Control: How To Keep Electronic Medical Records Accessible But Private". British Medical Journal, Vol. 322, No. 7281 (2001) 283-287
7. Deelman E. and Gil, Y.: Final Report of the NSF Workshop on the Challenges of Scientific Workflows. (http://vtcpc.isi.edu/wiki/images/3/3a/NSFWorkflowFinal.pdf) (2006)
8. Sweeney, L.: k-Anonymity: A Model For Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.
9. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. and Zhu, M.: Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations, 4(2) (2003) 19-26
10. Gil, Y., Ratnakar, V., Deelman, E., Mehta, G. and Kim, J.: Wings for Pegasus: Creating Large-Scale Scientific Representations of Computational Workflows. Proceedings of the 19th Annual Conference on Innovative Applications of Artificial Intelligence (2007)
11. Kim, J., Gil, Y., and Ratnakar, V.: Semantic Metadata Generation for Large Scientific Workflows. In Proceedings of the Fifth International Semantic Web Conference (2006)
12. Cheung, W.K., Zhang, X., Wong, H., Liu, J., Luo, Z. And Tong, F: Service-oriented Distributed Data Mining. IEEE Internet Computing, 10(4) (2006) 44-54.
13. Cheung, W.K., and Gil., Y.: Towards Privacy Aware Data Analysis Workflows for e-Science. Proceedings of 2007 Workshop on Semantic e-Science (SeS2007), held in conjunction with the Twenty-Second Conference of the Association for the Advancement of Artificial Intelligence (2007).
14. Bradshaw J., Uszok, A., Jeffers, R., Suri, N., Hayes, P., Burstein, M., Acquisti, A., Benyo, B., Breedy M., Carvalho, M., Diller, D., Johnson, M., Kulkarni, S., Lott, J., Sierhuis, M. and Van Hoof, R.: Representation and Reasoning For DAML-based Policy and Domain Services in KAoS and Nomads. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. ACM Press, New York (2003) 835-842.
15. Kagal, L. Finin, T., and Joshi, A.: A Policy Language for Pervasive Systems. In Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks, (2003) 63-76

Privacy and Capability Management for the European eIDM Framework

Mario Reyes[1], Ignacio Alamillo[2] and Daniel Chavarri[1]

[1] S21sec Labs, Parque empresarial La Muga, Nº11, Planta 1, Oficinas 1 - 6,
31160 Orcoyen, Spain.
[2] Agència Catalana de Certificació, Passatge de la Concepció, 11
08008 Barcelona, Spain

Abstract. The natural evolution of eGovernment is to go beyond the management of identities and therefore it is necessary to manage people, companies or organizations, and their capabilities to interact with Public Administrations. When developing an application based on an eID management system, this management issue must be tackled within each application (i.e. demonstrate the capability of one person to act, demonstrate the economical reliability, demonstrate his professional status, etc ...) and is normally based on the local jurisdiction. The objective of the present paper is to introduce a distributed system for the privacy-enhanced management of the capabilities associated to a person within the EU framework, independently from the origin and destination EU member state. The core of this system is the intelligence of the Capabilities Resolution Nodes (CRN) to cope with the complexity of the capability resolution and the capability sources discovery in the pan-European scenario. A European Capacity Resolution Network will be able to grow up the interoperability of the digital identities provided and valid in each member state and will answer the question “is this person, identified with this digital identity and who is described by those attributes, allowed to carry out this legal act in this country according to its law?”.

Keywords: privacy-enhanced tools, attribute management, legal roles, ontologies, semantic web, electronic government, identity management, interoperability

1 Introduction

In today’s Europe citizens are free to work and re-locate within the Union. Enterprises trade and carry out business across the Union. When citizens and enterprises do this they frequently have to interact with national public administrations. Member States are currently putting in place eGovernment[1] strategies that will allow such

[1] EGovernment seeks to use information and communications technologies to improve the quality and accessibility of public services. It can reduce costs for businesses and administrations alike, and facilitate transactions between administrators and citizens. It also

interactions to take place electronically. In parallel, they are frequently improving their business processes and the way in which business with citizens and enterprises is carried out. However, *there is a risk that that the development of government e-services may inadvertently result in the erection of barriers* to the continued development of the single market and the associated freedoms of movement. This would happen if citizens and enterprises that need to interact electronically with a national public administration other than their own were unable do so. For enterprises it could mean a relative loss of competitiveness, and for citizens increased costs. For Europe it could mean that the development of the single market and the associated four freedoms is hampered or even blocked.

Full-scale implementation of eGovernment raises difficult issues. These include:

- Safeguarding trust and confidence in on-line interaction with governments,
- Widespread access to on-line services so that no digital divide is created,
- Interoperability for information exchange across organizational and national borders,
  - *organizational* nature, which affect the processes and the *collaboration* between the administrations;
  - *semantic* nature, which is not limited to the interconnectivity of information resources, but also extends to the area where information can be *interpretable* by automatic and consequently *re-usable* forms of software applications that did not take part in the information resources' creation;
  - *technical* nature, which is the most direct form of *interconnection* of applications through diverse technological components; in particular, the development and ubiquity of the Internet technologies, on the base of *standards* and *open specifications* that are universally accepted have allowed for a high degree of technical interoperability.
- Advancing pan-European services that support mobility in the Internal Market and European Citizenship.

In this context, privacy laws impose strict controls on the interchange of personal information, an issue which is specially delicate when the information to interchange is identity information or, in our case, capabilities information, such as authorizations, delegations, powers of attorney and the representation of minors or incapables

1.1 The current scenario for capabilities management

When developing an application (business application, public procurement application ...) based on an eID management system, each application must develop the capability logics (i.e. demonstrate the capability of one person to act, demonstrate the economical reliability, demonstrate his professional status, etc ...), logic which is usually connected to legal theory in a concrete local jurisdiction. The present reality is

helps to make the public sector more open and transparent and governments more understandable and accountable to citizens.

that we negotiate the connection with the information sources locally in personalized scenarios and manage these “attributes” inside that application. Each change in the applied philosophy or in regulations implies the re-development of the application to adapt it to these new environmental conditions, even if the final logic of the application has not changed at all.

Moreover, when we are facing a pan-European or wider scenario, the complexity to build an application intelligent enough to deal with other ID attributes and information sources is enormous (who hosts that info? How can it be provided? How to understand and manage the relevant information?,...). Furthermore there are potential legal issues to be solved: roles and mandates are not homogeneous throughout Europe, privacy laws must be respected in both member states and so forth.

As an example, in a current real e-procurement scenario, if a company want to access a public procurement process in another member state, the representative will be able to identify himself (with current Identity Management technology/infrastructure) but his capability to act as a representative of that company has to be proved also ;… perhaps he will be able to do it locally (in his member state identity), but when trying to solve this for another member state he will be asked for registration of his capability to act as a representative in the destination member ’s state system. …The conclusion is that he will have to go through all the physical procedures in the destination member state to be inscribed as a potential user of the system The normal situation nowadays is that every company must be inscribed in on-line registers (registration that must comply with national laws and thus must be done locally) in every member state (27 times the same procedure).

The actual research challenge should not be aimed towards the integration and deployment of the identity management technologies that are currently in the standardization process, but it should be a step further, focusing on the real-world management of identity management contents (capabilities resolution) and the use of people management contents.

Moreover, it is of paramount importance to consider the privacy issue, as law requires that personal identifiable information must be under control of its owner. Some of the current proposed models of eGovernment initiatives do no consider the citizen as an active actor of the system, but just as an object about which different Public Administrations interchange data: these models present some potential deficiencies to comply with the privacy laws, and as a consequence may not be fully applied to the capabilities resolution domain.

On the contrary, the model we propose do consider the citizen as the actor that controls the capability information that she wants to share with one or more Public Administrations, in her local jurisdictions or along the network.

2 The proposed system

The main objective of the research work is the creation of a distributed system for the management of the capabilities associated to a person (a person is a set of one or more identities) within the EU framework independently from the origin and destination EU member state. This platform will integrate an intelligent system for arbitrating and routing the process flow needed for the capabilities resolution.

The solution is an intelligent system that releases the final application from the complex logic associated with the capability management in a pan-European framework. This simplifies the creation of the final application for businesses and eGovernment applications and at the same time will allow end users (EU citizens) to not only identify themselves in all member states (nowadays this is a fact) but also to be able to act in other member states. Moreover, the system will be a key tool for the citizen to be able to control the attributes and capabilities associated to his set of identities, which is a sound strategy to comply with privacy regulations and to generate user confidence.

The platform will allow any EU citizen in any EU member state to perform private and public procedures, whilst the capabilities resolution will take place in the credentials' origin country if the user has agreed to such a use of his identities. The result will be a real teleprocessing of administrative procedures in the EU framework. Moreover, the system will comply with the legislative framework in the field of privacy of personal data in each EU member state, as the information will not flow through the network without explicit user consent. Each origin member state will resolve the capabilities of a user in the same member state. This approach will follow the EC eGovernment Unit Roadmap design criteria, which state that the pan-European eIDM system must be 'federated in a policy sense'; in other words, this means that administrations mutually trust each other's identification and authentication methods, on the basis that they were considered acceptable by the originating administration.

At this stage of the research, the Catalan Certification Agency is leading the development of a platform for the management of capabilities in Catalonia, called Project PASSI, with the full set of functionality but limited in the scope to the Spanish law. At the moment, the first set of connectors are being developed, to allow a citizen to acquire and share her capabilities registered by Notaries (powers of voluntary representation) with the Catalan Public Administrations adhered to the system, using the interconnection infrastructure offered by the public administrations consortium AOC.

2.1 The proposed architecture

The system proposed does not consist in the network itself (that will follow a federated model and will be based on previous research work) but on the intelligence of the Capabilities Resolution Nodes (CRN) to cope with the complexity of the capability resolution and the capability sources discovery in the pan-European scenario (figure 1).

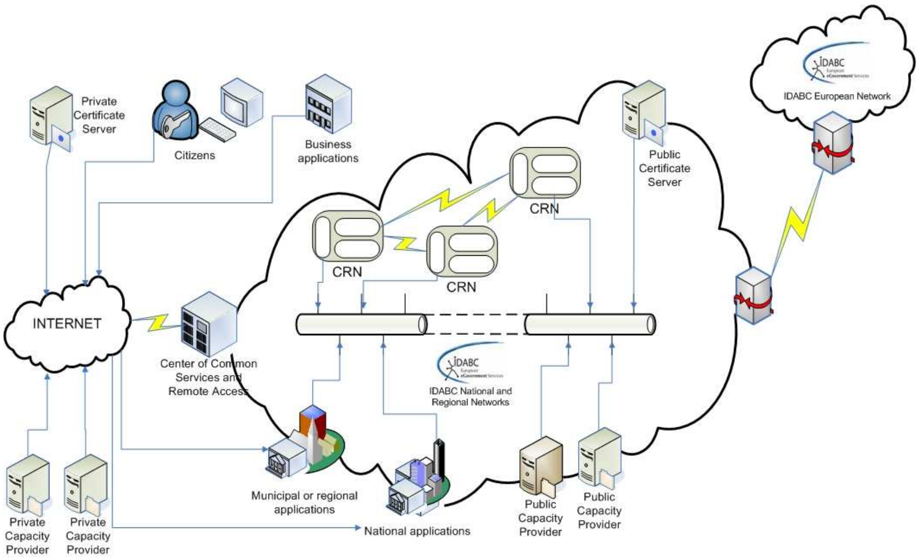

Private Certificate Server
Citizens
Business applications
IDABC European Network
Public Certificate Server
CRN
CRN
CRN
INTERNET
Center of Common Services and Remote Access
IDABC National and Regional Networks
Municipal or regional applications
National applications
Private Capacity Provider
Private Capacity Provider
Public Capacity Provider
Public Capacity Provider

Figure 1. Local Domain: citizen access to the services of the public administration; then, the application of the administration accesses the CRN through the infrastructure of the National and Regional networks.

For this purpose the following modules are developed:

- A **semantic model** to provide the necessary knowledge for the resolution of capabilities. The system should be capable of addressing the appropriate capabilities provider for the resolution of a specific capability.
- An **expert system** that will learn how to resolve the capabilities for a specific purpose, using machine learning technologies and intelligent agents.
- A **conceptual taxonomy** service able to map between roles and procedures in different domains (European, national, regional, local).
- **Interfaces** for the management, administration and communication between the platform providers, both service providers, identity providers and capacity providers. Similarly they are had to include the interfaces necessary to integrate the CRNs in the TESTA network.

2.2 Standards and related work

The Project relies on current Identity Management Technologies, most of which are in the process of being standardized:

- The XACML standard for resources access control, modified in order to include the capability resolution and to allow this resolution to be made in a distributed way.
- The SAML standard as a base to request identity and attribute information related to a given user.
- The Liberty Alliance standard, further analyzed to define the trust and security model for the capabilities federation.
- The federation concept becomes crucial to the concepts of association of identities and pan-European networks of identities. Only in this form is the citizen able to efficiently manage his personal character data.
- The platform is SOA based. Research is needed in this field for the definition and study of the workflows for the presented scenario, establishing a set of recommendations in the web services development phases for the public administrations.

3 Conclusion

The major contribution will not be Identity Management, but the capability and content management associated to an identity in an EU framework.

- **Ontology and semantics** are able to provide knowledge to the building blocks of the distributed intelligent manager. This is the main research block as, on the one hand it is mandatory to represent the semantic models of the member states laws as well as the EU directives, and on the other hand these models must represent the semantic relationship between all the EU legislation.
- The capability resolution intelligent manager will include the **logic required to increase its knowledge** while it continues resolving the assigned tasks (intelligent agents and machine learning). It must be capable of discovering where to direct its consultation to, so that a certain capacity is resolved.
- **Information security** in every area: access, authorization, information flow, personal data protection, citizen rights and audit. The recommendations coming from this project will be valid for the small administration as well as for large corporative administrations; different recommendation levels will be used to address all relevant stakeholders.
- The resulting system is a **privacy enhancing tool** (PET) in which the end user can manage his identities, his information, his personal character data, knowing at any moment where these data are and who has access to them. Furthermore, the provision of explicit access control to identity data by the user. As a consequence, the end user is able to share and to control the use of his attributes and consequently his capabilities.

References

1. Ignacio Alamillo, Xavier Urios: La Gestión de identidades y capacidades por las administraciones públicas. TECNIMAP. Sevilla (2006).
2. Ignacio Alamillo: Beyond identity management: capabilities management as a Public Administration simplification technique.
3. Consultation document for a future policy paper on pan-European Government e-Services. http://ec.europa.eu/enterprise/consultations/government_e-services/
4. IDABC, European eGovernment Services. http://ec.europa.eu/idabc/
5. TESTA: Trans European Services for Telematics between Administrations. http://ec.europa.eu/idabc/en/document/2097/
6. Torsten Priebe, Wolfgang Dobmeier, Nora Kamprath: Supporting Attribute-based Access Control with Ontologies. ARES 2006: 465-472
7. Kamelia Stefanova, Dorina Kabakchieva: User involvement in identity management e-Government architecture Development. Proceedings from workshop on User Involvement in e-Government development projects. September 12, at Interact 2005 in Rome, Italy. http://www.effin.org/egov-workshop_proceedings.html
8. Jena (2002). The jena semantic web toolkit, http://www.hpl.hp.com/semweb/iena-top.html, Hewlett-Packard Company.
9. Jena (2005). Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net/
10. RDQL (2005). Jena RDQL, http://jena.sourceforge.net/RDQL/
11. Protégé (2005). Protégé, Stanford Medical Informatics. 2005.
12. Protégé-API (2006). The Protégé-OWL API - Programmer's Guide, http://protege.stanford.edu/plugins/owl/api/guide.html
13. Liberty Alliance Project. http://www.projectliberty.org/

ISWC+ASWC
2007

The 6th International Semantic Web Conference and
the 2nd Asian Semantic Web Conference

November 11~15 2007
BEXCO, Busan KOREA