3-31-2014

Semantic Privacy Policies for Service Description and Discovery in Service-Oriented Architecture

Diego Z. Garcia
Universidade Federal de Ouro Preto, diego@decea.ufop.br

Miriam A M Capretz
Western University, mcapretz@uwo.ca

M. Beatriz F. Toledo
Universidade Estadual de Campinas, beatriz@ic.unicamp.br

Citation of this paper:

D. Garcia, M. A.M. Capretz, M. B. F. Toledo**, "Semantic Privacy Policies for Service Description and Discovery in Service-Oriented Architecture", Journal of Convergence Information Technology, vol. 9, Number 2, pp. 1-20, March 2014

Semantic Privacy Policies for Service Description and Discovery in Service-Oriented Architecture

[1]Diego Garcia, [2]Miriam A. M. Capretz, [3]M. Beatriz F. Toledo
[1]*Federal University of Ouro Preto, Rua 36, 115, 35931-008, Joao Monlevade, MG, Brazil; diego@decea.ufop.br*
[*2, Corresponding Author] *Department of Electrical and Computer Engineering, Western University, London, Canada; mcapretz@uwo.ca*
[3]*Institute of Computing, University of Campinas, Campinas, Brazil; beatriz@ic.unicamp.br*

Abstract

Privacy preservation in Service-Oriented Architecture (SOA) is an open problem. This paper focuses on the areas of service description and discovery. The problems in these areas are that currently it is not possible to describe how a service provider deals with information received from a service consumer as well as discover a service that satisfies the privacy preferences of a consumer. There is currently no framework which offers a solution that supports a rich description of privacy policies and their integration in the process of service discovery. Thus, the main goal of this paper is to propose a privacy preservation framework for the areas of service description and discovery in SOA. The framework enhances service description and discovery with the specification and intersection of privacy policies using a base and domain-specific privacy ontologies. Moreover, the framework extends SOA to include roles responsible for implementing a privacy registry as well as mediating the interactions between service consumers and providers and the privacy preservation component.

Keywords: *Service-Oriented Architecture; Service Description; Service Discovery; Privacy; Policy.*

1. Introduction

Service-Oriented Architecture (SOA) [1] is a software architecture based on the concept of service, a loosely coupled, abstract and discoverable software component. SOA has been an intense research area because of its potential to facilitate the development and management of software solutions. However, SOA still has open problems [2]. Privacy preservation is one of them.

Privacy [3] can be defined as the right of an individual to have information about them accessed and used in conformity with what is considered acceptable by that individual. The privacy problem in SOA [4] demands solutions that include privacy enhancing mechanisms in the different areas of SOA [5-6]. In basic SOA, service description is restricted to functional characteristics of services. As a consequence, service discovery is based on service functionality.

SOA extensions were proposed in order to include non-functional or Quality of Service (QoS) characteristics in service description. These extensions allow for service discovery that considers not only service functionality but also non-functional characteristics.

However, there is still a lack of an extension for privacy preservation [7-8]. Thus, the privacy problems in service description and discovery are that it is not possible to describe how a provider deals with private information received from a consumer and discover a service that satisfies the privacy preferences of the consumer.

Work that has been done on SOA privacy does not offer a proper solution for the service description and discovery problems. Privacy frameworks proposed in the literature have limitations including limited privacy policy model, privacy vocabulary as well as support for privacy policy specification and intersection as they do not use, for example, ontological concepts for creating policies. Furthermore, existing frameworks have no service discovery integration. Finally, such frameworks do not have proper support for the inclusion of other QoS attributes and for the consideration of domain-specific privacy preservation issues.

This paper addresses the limitations identified in SOA privacy frameworks proposed in the literature. It includes a policy model, which enables the description of privacy practices and preferences of providers and consumers. In the model, policy assertions refer to ontological concepts. Thus, policies are created from concepts defined in privacy ontologies. This information supports the matching between consumer and provider policies. Moreover, the framework includes privacy-aware service discovery, which enables the discovery of services that meet preferences of consumers. The use of policies for service discovery is accomplished by extending SOA with two roles: *privacy* and *mediator*.

Privacy preservation is a problem in several domains. Some privacy issues are common to different domains, but it is important to consider that each domain includes specific issues. Typically, a general privacy regulation [9] deals with common issues and a separate regulation [10] can complement it with domain issues. In order to address this aspect of privacy preservation, the proposed solution follows an approach in which general privacy issues are represented by a base privacy ontology and domain specific issues are captured by ontologies that extend the base ontology.

This work follows an approach that is used in Web service technology to deal with security. In Web service technology, security (Web Services Security – WS-Security [11]) and policy (Web Services Policy – WS-Policy [12]) standards are used together to create security policies for Web services. The privacy policies created in this work can be used in combination with policies for other aspects to improve the non-functional support in SOA. Thus, the proposed framework should be considered as one component of a set of components that would create a comprehensive security framework for SOA.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 gives an overview of the framework. Section 4 describes the privacy policy model that enhances service description. Section 5 describes the SOA extensions that support the use of the policy model for enhancing service discovery. Section 6 presents the implementation and evaluation of the framework. Section 7 concludes the paper.

2. Related work

This section reviews privacy frameworks for Service-Oriented Architecture (SOA) proposed in the literature. Two aspects were considered in the review of the frameworks:

- Policy model: how are privacy policies of service consumers and providers expressed in the framework?
- SOA extension: how is the basic architecture of SOA extended by the framework?

2.1. Policy model

The following questions were considered to review the policy model of the frameworks:

- *Format*: does the policy format defined by the framework allow for flexible specification of privacy policies?

A policy format is a standard structure that has to be followed by privacy policies defined by service consumers and providers. Thus, this first question asks if the framework defines a language that is used to structure policies in a way that they can be processed by computers. Several frameworks [13-17] assume the use of privacy policies by service consumers and providers, but these frameworks do not define a format for the privacy policies. Thus, they do not have a format or the format is not available and consequently the frameworks do not allow for the specification of computer-processable privacy policies. The existing frameworks [18-20] that define a format for privacy policies do not include support for flexibility in the policy format. Thus, these frameworks do not define rules that convert privacy policies to a standard structure and consequently the format is rigid. When these rules are present, consumers and providers can create flexible privacy policies that are converted to a standard structure before being processed. A flexible format includes constructs, for example, alternatives and optional assertions, which allow for richer privacy policy specifications.

- *Vocabulary*: does the privacy vocabulary defined by the framework cover the principles of privacy regulations?

A privacy vocabulary is a set of terms related to privacy and relationships among the terms that are used in the specification of privacy policies by service consumers and providers. Some frameworks [13,16,17] assume the use of a privacy vocabulary together with a format for privacy policies, but these frameworks do not define a privacy vocabulary and do not allow for the specification of interoperable privacy policies. Several frameworks define a privacy vocabulary, but the vocabulary is limited. The privacy vocabulary of some frameworks [14-15] includes the concepts of information and collector only. Other existing frameworks [18-20] define a privacy vocabulary that misses the concepts related to collection means, owner access and use record as well as the categorization of some concepts. Thus, these frameworks do not include terms and relationships that capture the principles defined in privacy preservation regulations and consequently the vocabulary is limited. When the principles of regulations are present, consumers and providers can create comprehensive privacy policies that cover a wide range of requirements and guarantees related to privacy preservation. A comprehensive privacy vocabulary, which includes concepts such as owner access and use record, allows for the specification of policies that can provide a higher level of privacy preservation.

- *Semantics*: does the support for semantics of the framework allow for the specification and intersection of semantic policies?

Meaning can be added to the information in a privacy vocabulary by including support for semantics in the framework. Several frameworks [13,15,16,18,19] do not have a privacy vocabulary enriched with semantic information or the semantics is not available and consequently the frameworks allow for the matching between the privacy policies of a service consumer and provider based on syntax only. The frameworks [14,17,20] that include support for semantics do not allow for the specification and intersection of semantic policies as these frameworks extend service ontologies. Thus, in these frameworks the privacy policy is a part of the service description and consequently the policy is not a separate document. When a privacy ontology is present, consumers and providers can create privacy policies that are easier to maintain as they are likely to change more often than the service descriptions. An ontology-based policy, such as an annotated policy, allows for the reuse of policies and the use of policy intersection for verifying the compatibility of privacy policies.

- *Domain*: does the framework define an approach to deal with domain-specific privacy issues?

Different domains, such as health and learning, have specific privacy issues in addition to the privacy issues that cross multiple domains. Several frameworks [14,15,17,20] do not consider domain-specific privacy preservation issues. Thus, they do not have support for extension and consequently the frameworks do not allow for the specification of privacy policies that include concepts from a given domain. Some existing frameworks [13,16] include placeholders for dealing with domain-specific privacy issues, but these frameworks do not define an approach to the application of the framework to different domains. Thus, these frameworks consider the importance of dealing with domain-specific privacy issues and consequently they are open for extensions. However, they do not define any approach as a part of the framework that drives the extension of the framework with concepts derived from domain-specific issues. The lack of a mechanism to implement the extension of the framework requires the definition of one by the user, which can affect the interoperability of the framework negatively.

2.2. SOA extension

The following questions were considered in order to review the extension to the basic architecture of SOA of the frameworks:

- *Modification*: how does the framework modify the roles and interactions of basic SOA?

Some frameworks [13,14,18] modify basic roles of SOA, whereas other frameworks [15,17,19,20] add new roles to SOA. Between these two design choices, the second choice is better as it facilitates the deployment of the extension to an SOA environment. The new roles are added as services that are used by consumers and providers the same way as they use other services in the environment. The modification of basic roles, including consumer, provider and registry, is hard to deploy as the entities that are active in the environment need to be modified. Interactions related to privacy preservation are needed between the service consumer and provider in some frameworks [13,17,19]. This setting is not a good design choice as in basic SOA the decision on which service to use is done at discovery time and the consumer and provider start interacting after the decision. Thus, privacy-related interactions should involve a third party at publication and discovery times. All existing frameworks require direct

interaction with the components responsible for privacy preservation. This setting is not a good design choice as it affects the scalability of the framework negatively when other non-functional characteristics are dealt with. Thus, direct interaction with the privacy components should be avoided.

- *Discovery*: does the framework integrate privacy policies in the process of service discovery?

No framework that integrates privacy policies in the process of service discovery has been identified in the literature. In the surveyed frameworks [13-20], the service consumer has to perform actions after service discovery in order to receive services that meet the privacy preservation preferences of the consumer; for example, the consumer has to request the policy from the provider as well as forward it to the privacy component for verification or do it itself. Due to the lack of integration, consumers and providers may have to perform additional tasks or the number of interactions needed for a consumer to use a service may increase. The integration of privacy policies in the process of service discovery may lead to modifications to the registry, but they can be avoided. Thus, if the integration can be implemented without modifications to the registry, then it is a better design decision as it keeps compatibility with basic SOA as well as alleviates the burden on service consumers and providers.

- *Quality of Service (QoS)*: does the framework enable the inclusion of other QoS attributes with the separation of the different attributes?

QoS is a set of non-functional characteristics of services such as privacy, security and reliability. Although the framework proposed in this paper has been developed specifically to deal with privacy preservation, it has to be prepared for working with other QoS attributes. The QoS attributes required in different environments and interactions vary. They should be dealt with separately as they are processed differently, for example, they need different matching rules. No framework that supports the inclusion of other QoS attributes with the separation of the different attributes has been identified in the literature. In order to deal with other QoS attributes in the surveyed frameworks [13-20], the service consumer and/or the service provider have to interact with a set of components responsible for the QoS attributes or a single component is responsible for all QoS attributes in the framework. These two settings are not good design decisions. The first one affects the scalability of the framework negatively regarding consumers and providers, which have to interact with an increasing number of components that have to be discovered and bound to. The second design choice affects the performance of the framework negatively as a heavy component, which is responsible for processing all the requested QoS attributes, is included in the framework. In addition, new matching rules have to be added to the component when a new attribute is included in the framework.

3. Privacy preservation framework

The proposed framework addresses the limitations identified in existing frameworks (Section 2). An overview of the framework is shown in Figure 1.

As shown in Figure 1, the framework includes a model for semantic privacy policies and a process of privacy-aware service discovery through an extension to the basic architecture of SOA. The model enables the description of provider privacy practices and consumer preferences in policies. It follows an approach in which privacy preservation issues are represented by a base ontology and domain-specific ontologies. Privacy-aware service discovery enables the discovery of services that meet privacy preferences of consumers. It uses the privacy policy model. At service discovery, policies are intersected to select services from providers whose policies match the consumer's policy. Thus, the framework provides privacy preservation support for the areas of service description and discovery.

The model enhances service description with privacy practices and service request with privacy preferences. The policies complement basic service description and request that include information on service functionality and use. Privacy-aware service discovery integrates privacy-awareness in the processes of service publication and discovery to enable the publication of privacy practices and service discovery that considers privacy preferences. The process of privacy-aware service discovery is accomplished by extending SOA with roles and activities that support the idea of different registry types, including registries for service descriptions and policies.

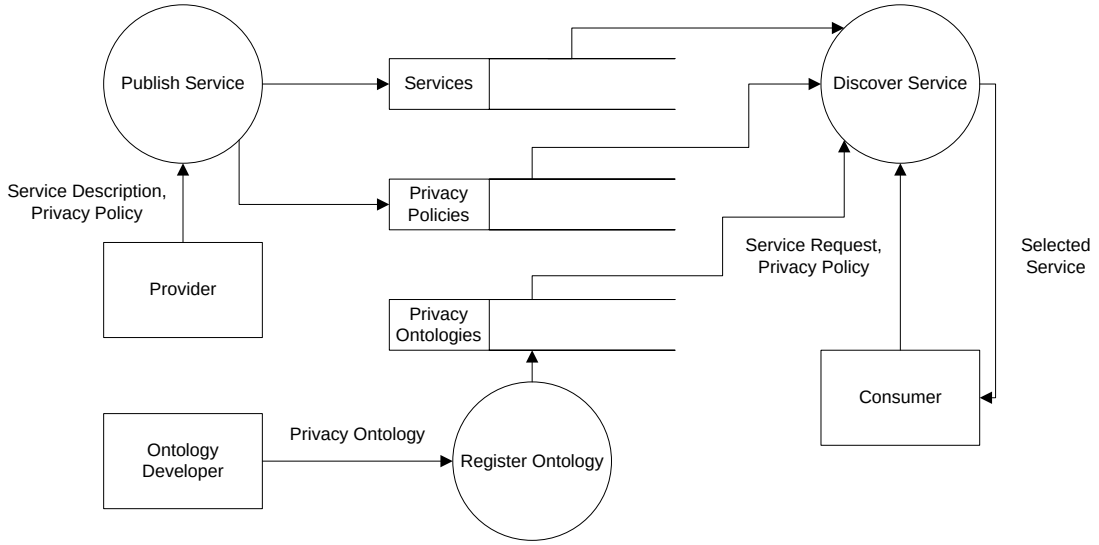

Figure 1. Privacy preservation framework

4. Semantic privacy policies model for service description

The framework includes a policy model based on WS-Policy. WS-Policy [12] is the standard for Web service policies and, thus, its format was used to make the model interoperable. The main difference between the proposed model and WS-Policy is that WS-Policy does not support the use of ontologies, whereas in the proposed framework, ontologies are used to define a privacy vocabulary whose concepts are used to specify policies.

4.1. Policy elements

The policy model includes four elements: component, assertion, alternative and policy. Figure 2 shows a policy example, which is going to be used to illustrate the elements.

```
01  Policy
02    ExactlyOne
03      All
04        Name
05          LegalRetention
06      All
07        Name
08          NoRetention
```

Figure 2. Example of privacy policy

In Figure 2, Line 1 indicates a policy. Line 2 shows that it includes alternatives. The first alternative is defined from Line 3 and the second one from Line 6. Each alternative includes an assertion on the name information piece (Lines 4 and 7). Each assertion includes a component, which defines the retention period (Lines 5 and 8). The elements of the policy model are described as follows:

- Component and Assertion

An assertion deals with a set of information pieces, which is its subject. An assertion includes components and each component restricts one aspect of the handling of the assertion's subject. Figure 3 includes an assertion and a component. The assertion's subject is the name information piece and the component restricts its retention.

```
01    All
02      Name
03        NoRetention
```

Figure 3. Example of component and assertion

Each assertion restricts the handling of a set of information pieces. This way consumers and providers can define assertions for a single information piece or a set with more than one information piece. Thus, by including components to an assertion according to their needs, consumers and providers can express different restrictions to information pieces in different settings and establish different privacy preservation levels based on what each consumer and provider consider as an acceptable practice.

Assertions are expressed using concepts defined in ontologies. These concepts define component types. They create a terminology for expressing policies and indicate general as well as domain-specific privacy semantics. Thus, assertions associated with different services and referring to the same concepts are interpreted similarly. A concept is referred to by an assertion and a component through its qualified name, including the Uniform Resource Identifier (URI) of the ontology that represents the namespace and its local identification. For readability, assertions are expressed using local identifications. In the examples used in this section, the policy components are from the base ontology and some components are used to enrich the examples and would have to be defined in domain ontologies. In Figure 3, the *Name* assertion subject and the *NoRetention* component are defined in a domain and the base ontologies, respectively.

- Alternative

Assertions are grouped in collections called alternatives. An alternative is an ordered assertion collection. It indicates the preferences or practices represented by its assertions and its privacy preservation level depends on the assertions' level. Assertions are processed in the order in which they appear in the alternative. Figure 4 has two alternatives with an assertion each.

```
01  ExactlyOne
02    All
03      Name
04        LegalRetention
05    All
06      Name
07        NoRetention
```

Figure 4. Example of alternative

This element is included in the policy model to offer providers and consumers the possibility to specify alternative settings of privacy practices and preferences. This way the likelihood to successfully intersect policies when discovering services is higher.

- Policy

A policy is created by grouping alternatives. It is an ordered collection of alternatives. A policy with more than one alternative indicates that there are choices of preferences or practices. Alternatives are processed in the order in which they appear in the policy. While processing a policy, the first alternative is checked, then, if needed, the second one and so on. Figure 5 shows a policy with two alternatives.

Policies restrict interactions between consumers and providers. Provider policies specify practices and consumer policies specify preferred practices or preferences. Policies apply to information pieces disclosed by consumers to providers to use their services. Figure 5 can represent a consumer or provider policy. Thus, it can define a consumer's preferences or provider's practices regarding the retention of the name information piece.

```
01  Policy
02    ExactlyOne
03      All
04        Name
05          LegalRetention
06      All
07        Name
08          NoRetention
```

Figure 5. Example of policy

A provider exposes a policy describing conditions under which it performs its activities in the context of a service. A behavior that reflects those conditions is presented by the provider to satisfy the

policy. A consumer can use the policy exposed by the provider to decide whether or not to use the service. It can choose any alternative in the policy, as each one represents valid conditions under which the service can be used. As each alternative represents an alternative set of conditions, the consumer can choose only one for each interaction with the service. A provider supports an assertion if it performs the practice represented by it. An alternative is supported if all of its assertions are supported by it. A provider supports a policy if it supports all the alternatives of the policy. Thus, it must be able to operate under the different conditions represented by the alternatives in a policy so that it can support the policy. According to Figure 5, the provider has to be able to provide the service with legal retention or no retention of name to support the policy. In the case of the consumer, the policy indicates that the consumer accepts services from providers with no retention or legal retention practices.

4.2. Policy format

This section describes the policy format, which defines a standard structure for the specification of policies. Policies follow the format shown in Figure 6.

```
01  Policy Name="" Id=""
02    ExactlyOne
03      All
04        Assertion
```

Figure 6. Policy format.

The items of the policy format are described as follows:

- *Policy*: a policy.
- *Name*: the identity of the policy in the form of an absolute Internationalized Resource Identifier (IRI). The name of a policy is referred to by a service description or request in order to associate them.
- *Id*: the policy's identity in the form of an identifier within its enclosing document. An IRI-reference is composed using the identifier of a policy and the IRI of the enclosing document in order to refer to the policy externally.
- *ExactlyOne*: the collection of all the alternatives of the policy. This item indicates that only one alternative can be selected at a time.
- *All*: an alternative. This item groups the assertions of an alternative and indicates that all assertions are valid when the alternative is selected.
- Assertion: a preference in the case of a consumer policy or a practice in the case of a provider policy.

A policy named http://www.privpol.com/Policy1 in the format is shown in Figure 7. The assertions are illustrative and their definitions are not necessary at this point as the focus is on the description of the format. This example includes two alternatives. The first one states that name and contact information is collected by the provider (Lines 03-04), whereas name information only is collected for the second alternative (Lines 05-06).

```
01  Policy Name="http://www.privpol.com/Policy1"
02    ExactlyOne
03      All
04        Name, Contact
05      All
06        Name
```

Figure 7. Formatted policy

4.3. Policy intersection

Intersection is matching between policies, which identifies compatibility between two policies to verify if their owners can interact with each other. The input of the intersection process is a consumer and provider policy. The output is a policy including a compatible alternative from the provider policy or empty if the policies are incompatible.

Two policies are compatible if at least one consumer alternative is compatible with at least one provider alternative. Two alternatives are compatible if each consumer mandatory assertion is compatible with a provider assertion as well as each provider assertion is compatible with a consumer mandatory assertion. Two assertions are compatible according to matching rules defined by ontologies. The selected provider has to support all practices indicated by the result of the intersection process.

A policy intersection example is shown as follows. Figure 8 and Figure 9 present a consumer and provider policy, respectively. These policies are the intersection input.

```
01  Policy
02    ExactlyOne
03      All
04        Name
05          NoRecipient
06          LegalRetention
07      All
08        Name
09          AnyRecipient
10          NoRetention
```

Figure 8. Consumer policy

Figure 8 includes two alternatives. The first one (Lines 3-6) indicates that *Name* information can be retained as required by law (*LegalRetention*) but the information cannot be disclosed to third parties (*NoRecipient*). The second alternative (Lines 7-10) indicates that *Name* information can be disclosed to any third parties (*AnyRecipient*) but it cannot be retained (*NoRetention*).

```
01  Policy
02    ExactlyOne
03      All
04        Name
05          BusinessRecipient
06          LegalRetention
07      All
08        Name
09          BusinessRecipient
10          NoRetention
```

Figure 9. Compatible provider policy

Figure 9 includes two alternatives. The first one (Lines 3-6) indicates that *Name* is retained as required by law (*LegalRetention*) and disclosed to third-party businesses (*BusinessRecipient*). The second one (Lines 7-10) indicates that *Name* is disclosed to businesses and not retained (*NoRetention*). The first consumer alternative (Figure 8) is not supported by any provider alternative (Figure 9) as it requires no disclosure (*NoRecipient*) and both provider alternatives disclose *Name* (*BusinessRecipient*).

The second consumer alternative is not supported by the first provider alternative as it requires no retention (*NoRetention*) and the first provider alternative retains *Name* (*LegalRetention*). The intersection result (Figure 10) includes the second provider alternative as it supports the second consumer alternative (*NoRetention*).

```
01  Policy
02    ExactlyOne
03      All
04        Name
05          BusinessRecipient
06          NoRetention
```

Figure 10. Policy intersection result

4.4. Base ontology

The semantic approach that supports the model includes a base and domain-specific ontologies. The base ontology includes general privacy concepts. Domain ontologies extend the base one and include

domain-specific privacy concepts. An overview of the base ontology is shown in Figure 11. The base concepts are described under types of information activities to which they relate. Four activity types can be identified in privacy regulations [9,21-23]: initial disclosure, further disclosure, storage and use.

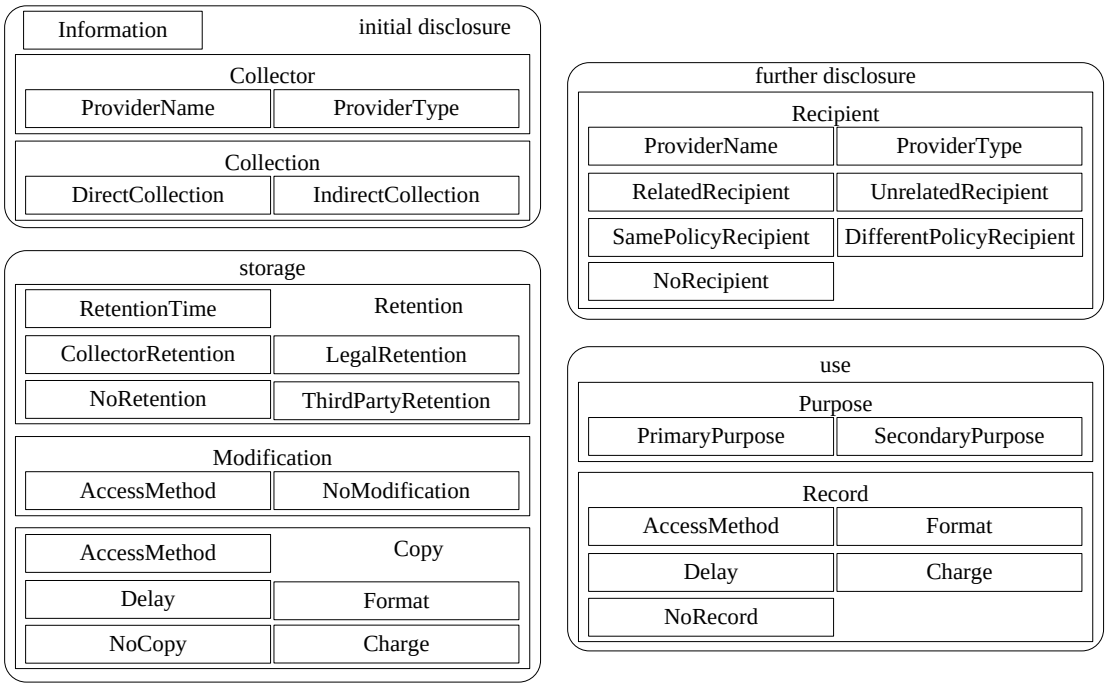

Figure 11. Base ontology

4.4.1. Initial disclosure: In this activity, a consumer discloses information to a provider. It is important to give the consumer the ability to control the disclosure. Firstly, it is necessary to ensure that the consumer is aware of it. It is also important to ensure that it is aware of its implications so that it can balance them and the benefits it is going to get from the disclosure. Three concepts were identified in this activity: *Information*, *Collector* and *Collection*.

- *Information*

This concept represents the type of the information piece to be disclosed by the consumer (in a consumer policy) or collected by the provider (in a provider policy).

- *Collector*

This concept represents the provider that is allowed by the consumer to collect its information (in a consumer policy) and the provider that is going to collect the consumer's information (in a provider policy). *Collector* includes the following concepts:

- *ProviderName*: identifies the providers allowed by the consumer (in a consumer policy) and the one that is going to collect the information (in a provider policy).
- *ProviderType*: indicates the types of the providers allowed by the consumer (in a consumer policy) and the type of the one that is going to collect the information (in a provider policy).

- *Collection*

This concept represents the information collection means, that is, the means the provider employs to collect information, allowed by the consumer (in a consumer policy) and used by the provider (in a provider policy). Types of collection means include:

- *DirectCollection*: indicates that the information can be collected directly (in a consumer policy) and is going to be collected directly (in a provider policy).
- *IndirectCollection*: indicates that the information can be collected indirectly; for example, using information provided by the consumer to obtain publicly-available information (in a consumer policy), and is going to be collected indirectly (in a provider policy).

4.4.2. Further Disclosure: A further disclosure occurs between two providers. In this activity, the provider that collected the information from the consumer shares it with another one. Different indirectness levels can occur, as the third-party provider can share the information received from its collector with another provider. Thus, a provider receives the consumer's information from the provider with which the consumer directly interacted or, in additional indirectness levels, it receives the information from a provider that is not the collector. The *Recipient* concept was identified in this activity.

- *Recipient*

This concept represents the recipient of a further disclosure allowed by the consumer (in a consumer policy) and the third parties that are going to receive from the collector the information disclosed by the consumer (in a provider policy). *Recipient* includes:

- *ProviderName*: identifies the recipients of further disclosures allowed by the consumer (in a consumer policy) and the third parties that are going to be recipients of further disclosures by the provider (in a provider policy).
- *ProviderType*: indicates the types of the recipients allowed by the consumer (in a consumer policy) and the types of the third parties that are going to be recipients of further disclosures by the provider (in a provider policy).
- *RelatedRecipient*: indicates that the recipients must behave on behalf of the collector (in a consumer policy) and are going to do so (in a provider policy).
- *UnrelatedRecipient*: indicates that the recipients can behave on their own behalf (in a consumer policy) and are going to do so (in a provider policy).
- *SamePolicyRecipient*: indicates that the recipients must perform the same practices as the collector regarding the disclosed information (in a consumer policy) and are going to do so (in a provider policy).
- *DifferentPolicyRecipient*: indicates that the recipients can perform different practices from the collector regarding the disclosed information (in a consumer policy) and are going to do so (in a provider policy).
- *NoRecipient*: indicates that no recipient is allowed by the consumer (in a consumer policy) and the collector does not disclose the information to any third party (in a provider policy).

4.4.3. Storage: Two storage types can occur. In the first one, information is stored beyond service completion. The second type refers to information that is stored only for the time period of the transaction. Another dimension that can classify storage is who is going to store it. Information can be stored by the provider with which the consumer interacted or by a third-party provider. Three concepts were identified: *Retention*, *Modification* and *Copy*.

- *Retention*

This concept represents the time period of the information retention and the provider responsible for it. *Retention* includes the following concepts:

- *RetentionTime*: indicates the maximum time period the information can (in a consumer policy) and is going to be retained (in a provider policy).
- *LegalRetention*: indicates that the information can (in a consumer policy) and is going to be retained as required by law (in a provider policy).
- *CollectorRetention*: indicates that the information must (in a consumer policy) and is going to be retained by the collector (in a provider policy).
- *ThirdPartyRetention*: indicates that the information must (in a consumer policy) and is going to be retained by a third party (in a provider policy).
- *NoRetention*: indicates that the information cannot (in a consumer policy) and is not going to be retained beyond service completion (in a provider policy).

- *Modification*

This concept represents the capability of the consumer to request to the provider the modification of the retained information. *Modification* includes the following concepts:

- *AccessMethod*: identifies the means required by the consumer (in a consumer policy) and supported by the provider to request the modification (in a provider policy).
- *NoModification*: indicates that the consumer does not require (in a consumer policy) and the provider does not allow for modification (in a provider policy).

- *Copy*

This concept represents the consumer's capability to request a copy of the retained information to the provider. *Copy* includes the following concepts:

- *AccessMethod*: identifies the means required by the consumer (in a consumer policy) and supported by the provider to request copy (in a provider policy).
- *Format*: identifies the copy format required by the consumer (in a consumer policy) and supported by the provider (in a provider policy).
- *Delay*: identifies the maximum time period the consumer is willing to wait for the receipt of the copy (in a consumer policy) and the delay the provider demands to make it available (in a provider policy).
- *Charge*: identifies the maximum charge the consumer is willing to pay for the receipt of the copy (consumer) and the charge the provider demands to make it available (provider).
- *NoCopy*: indicates that the consumer does not require (in a consumer policy) or the provider does not allow for copy request (in a provider policy).

4.4.4. Use: Two types of use can occur. The first one includes the uses that are necessary for accomplishing the service, while the second one includes secondary uses. Another classification dimension for use is the provider that performs it. Information can be used by the provider with which the consumer directly interacted or third parties to which the collector disclosed it. Two concepts were identified in this activity: *Purpose* and *Record*.

- *Purpose*

This concept represents the purposes for information collection allowed by the consumer (in a consumer policy) and the purposes for which the provider is going to collect the information (in a provider policy). *Purpose* includes the following concepts:

- *PrimaryPurpose*: indicates that the information can (in a consumer policy) and is going to be used for service completion only (in a provider policy).
- *SecondaryPurpose*: indicates that the collected information can (in a consumer policy) and is going to be used for secondary purposes (in a provider policy).

- *Record*

This concept represents the capability of the consumer to request to the provider a record of the use of the collected information. *Record* includes the following concepts:

- *AccessMethod*: identifies the means required by the consumer (in a consumer policy) and supported by the provider to record request (in a provider policy).
- *Format*: identifies the record format required by the consumer (in a consumer policy) and supported by the provider (in a provider policy).
- *Delay*: identifies the maximum time period the consumer is willing to wait for the receipt of the requested record (in a consumer policy) and the delay the provider demands to make it available (in a provider policy).
- *Charge*: identifies the maximum charge the consumer is willing to pay for the receipt of the record (in a consumer policy) and the charge the provider demands to make it available to the consumer (in a provider policy).
- *NoRecord*: indicates that the consumer does not require (in a consumer policy) and the provider does not allow for record request (in a provider policy).

5. Privacy-aware service discovery

The framework includes a process of privacy-aware service discovery that uses the policy model. It allows for consumers to have their preferences considered when looking for services. In order to enable the process, two roles were included in SOA: *mediator* and *privacy*. A publication and discovery space is defined, which includes the *privacy* role, in addition to the basic role of registry. The services in this space are responsible for service publication and discovery. Whereas the registry service is responsible for functional characteristics, the *privacy* service is responsible for privacy characteristics. The second new role, the *mediator*, is added to make the publication and discovery space transparent to the consumers and providers as well as support additional QoS characteristics. As with the service registry, these roles should be played by trusted third parties to ensure that their activities are unbiased. The extended SOA is shown in Figure 12.

The provider uses the extension by sending its policy together with the service description to the *mediator*. In the case of the consumer, the extension is used by sending to the *mediator* its policy together with the service request. The *mediator* can then be added to SOA and interacted with the same way the registry is used, by selecting a service in an Enterprise Service Bus (ESB) and using an Application Programming Interface (API), for example. If consumers and providers do not want to use the privacy feature, then they can still interact similarly to how they do so in traditional SOA. The new roles and their interactions with the basic ones are presented as follows.

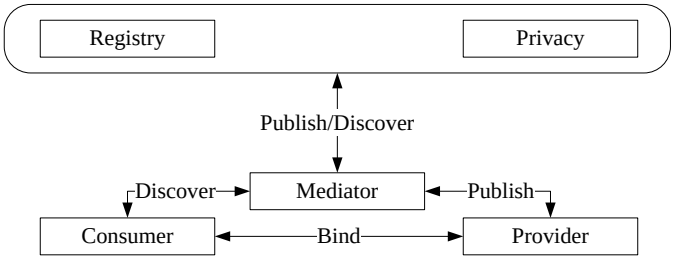

Figure 12. SOA new roles

5.1. Mediator

The *mediator* service is included in SOA to facilitate the interactions between the provider or consumer and the publication and discovery services, including registry and *privacy* services, by making them transparent to consumers and providers. Together with the registry and *privacy*, the *mediator* is responsible for service publication and discovery. It uses them to execute these activities. The *mediator* has a registry of publication and discovery services, which is used to register addresses of registries and privacies. Registry and *privacy* providers are responsible for registering their services in this registry. Based on the message received from the provider or consumer, the *mediator* decides which publication or discovery services are needed to execute the requested activity. It retrieves the addresses of the registry and *privacy* so that it can use them.

The activities of registration and deregistration of publication and discovery services performed by the *mediator* are shown in Figure 13. At publication and discovery service registration/deregistration, the *mediator* receives a registration/deregistration message from the provider including a description. Then, the description is registered/deregistered. Finally, it sends a result message to the provider.

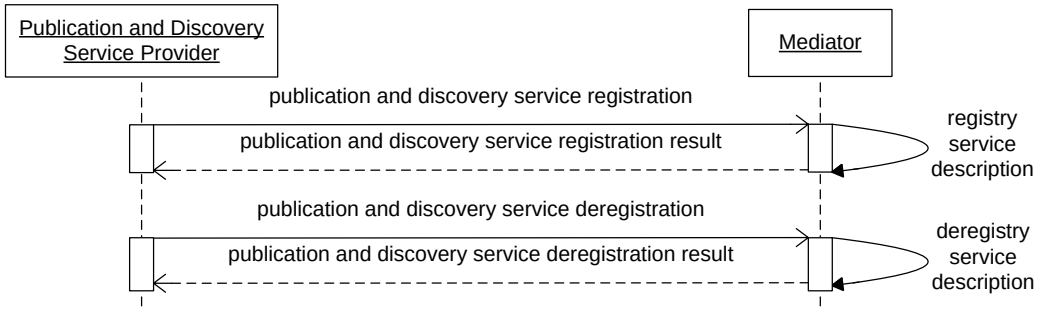

Figure 13. Registration and deregistration of publication and discovery services

The tasks under the responsibility of the *mediator* at service publication and unpublication are shown in Figure 14. At service publication/unpublication, the *mediator* receives a publication/unpublication message from the provider. It sends a service description message to the registry and a privacy policy message to the *privacy* if the publication/unpublication message includes a service description and privacy policy. Then, the *mediator* receives a description and policy result message from the registry and *privacy*. Finally, it sends a final result message to the provider.

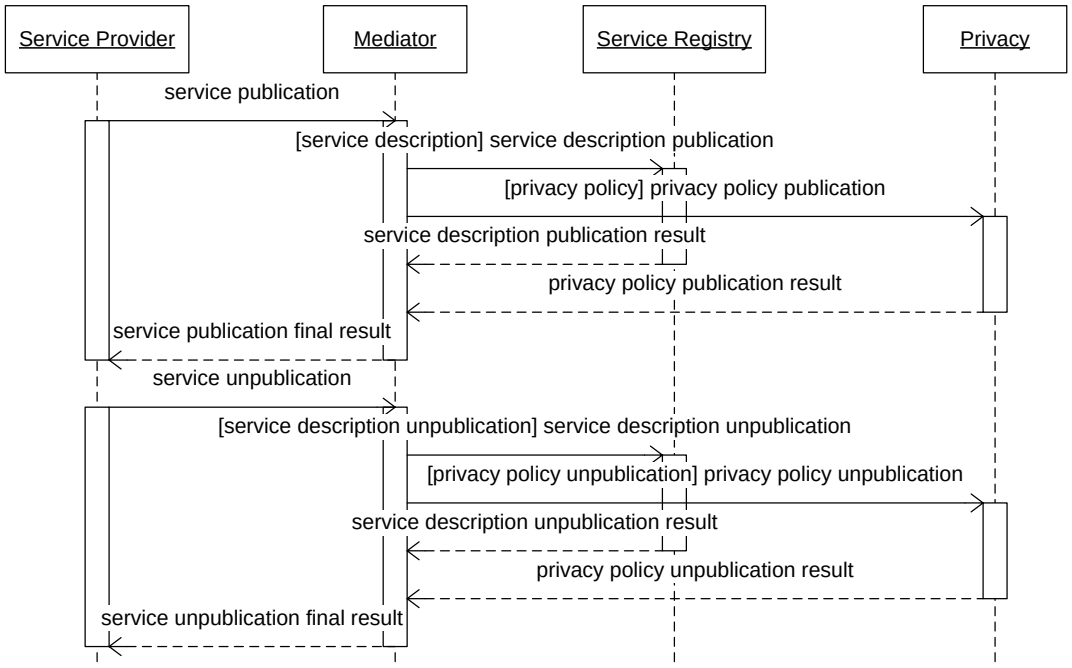

Figure 14. Mediator tasks at service publication and unpublication

The tasks under the responsibility of the *mediator* at service discovery are shown in Figure 15. At service discovery, the *mediator* receives a discovery message from the consumer. It sends a service description and privacy policy message to the registry and *privacy* if the discovery message includes a service request and privacy policy. Then, the *mediator* receives a service description and privacy policy result message from the registry and *privacy*. Finally, it sends a final result message to the consumer.

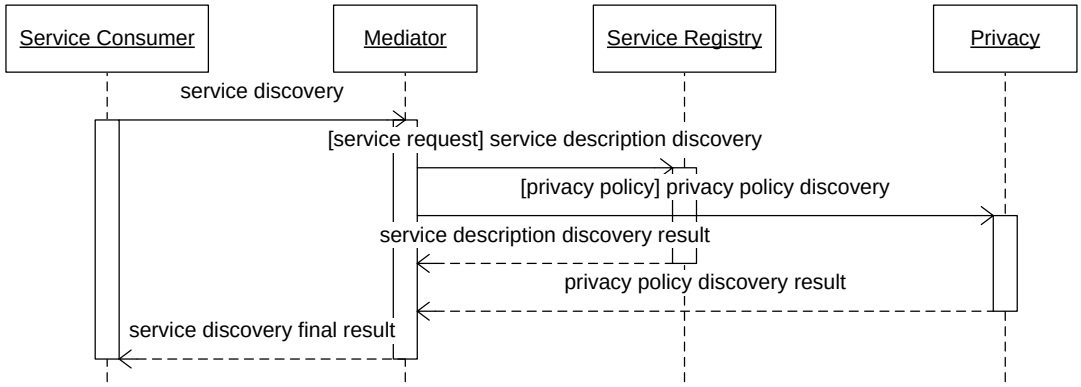

Figure 15. Mediator tasks at service discovery

5.2. Privacy

The *privacy* service is responsible for the publication, unpublication and discovery of policies. It provides these activities to the provider and consumer through the *mediator*. The *privacy* includes a

policy registry, which is used to register provider policies. These policies are retrieved by the *privacy* so that it can intersect them with the consumer policy. The *mediator* is responsible for sending the policies to the *privacy*. The *privacy* also includes an ontology registry, which is used to register the base and domain ontologies and query them to determine compatibility between consumer and provider policies. To verify policy compatibility, the *privacy* retrieves the ontological concepts associated to each assertion in the policies. Then, it checks the relationship between the concepts in the ontologies. Domain representative organizations are responsible for developing domain-specific ontologies and registering them in the *privacy*'s registry.

The activities of registration and deregistration of privacy ontologies, which are defined to apply the framework to specific domains, performed by the *privacy* are shown in Figure 16. At ontology registration/deregistration, the *privacy* receives an ontology message from the ontology developer. Then, it registers/deregisters the ontology. Finally, the *privacy* sends an ontology result message, indicating the outcome of the activity, to the ontology developer.

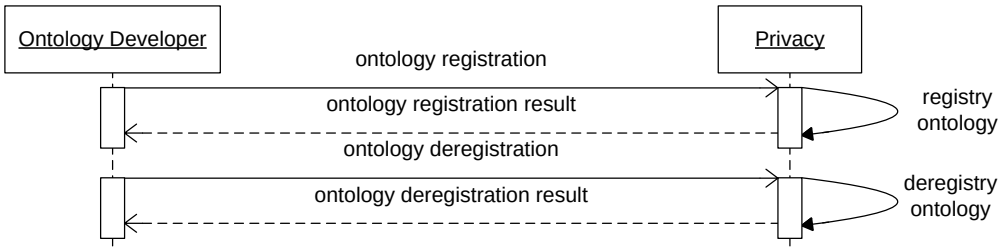

Figure 16. Registration and deregistration of ontologies

The activities of privacy policy publication, unpublication and discovery performed by the *privacy* are shown in Figure 17. At service publication/unpublication/discovery, the *privacy* receives a policy message from the *mediator*. Then, it publishes/unpublishes/discovers the privacy policy. Finally, the *privacy* sends a policy result message, indicating the outcome of the activity, to the *mediator*.

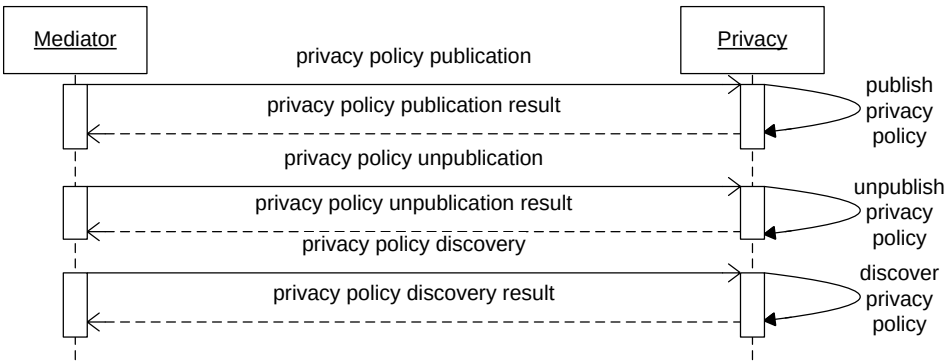

Figure 17. Publication, unpublication, and discovery of policies

6. Implementation and evaluation

In order to evaluate the framework, a prototype was implemented. The goal of the evaluation was to check the effectiveness of the SOA extension and the advantage of using ontologies for comparing

privacy policies. Thus, the emphasis of this implementation and evaluation was on the integration of privacy preservation in service description and discovery through the use of semantic policies.

6.1. Implementation

The prototype was developed using Web service technology. Web services were implemented in Java, including *mediator* and *privacy* services. Other Web services defined an SOA environment and represented providers, consumers and a service registry. The registry's databases for storing service descriptions were created using the Structured Query Language (SQL). Policies were created to demonstrate different cases in the domain scenario that was proposed for the evaluation. They were written using an extended version of the Web Services Policy Framework (WS-Policy), which was created to support the policy model. The base and domain ontologies created for the evaluation were written in the Web Ontology Language (OWL). The *mediator*, *privacy* and registry services were deployed on an application server. The following products were used:

- Sun Java Development Kit Version 1.5: Java support.
- Apache Tomcat Version 4.0: an application server.
- MySQL AB MySQL Version 5.0: a database management system.
- Apache Axis Version 1.3: Web Services Description Language (WSDL) support and a SOAP engine.
- Apache jUDDI Version 0.9: a Universal Description Discovery & Integration (UDDI) registry.
- HP/IBM/SAP UDDI4J Version 2.0: a UDDI Java API.
- Apache WS-Commons/Policy Version 0.9: WS-Policy support.
- Stanford Protégé 4.0: OWL support.

The prototype created an environment formed by a set of Web services (Figure 18). A Web service was used to provide the registry operations through the UDDI API and another Web service implemented the *privacy* service by using the OWL API. These services were encapsulated by a third Web service that implemented the *mediator* service, which provided an interface to the consumers and providers. In this setting, the consumers and providers were represented by services that used the operations provided by the *mediator* to publish and discover services. The policies of the consumers and providers were defined in XML files that were linked to ontologies through Protégé and processed in Java code through the Eclipse Integrated Development Environment (IDE).

6.2. Evaluation

Among the different domains, health care is an example in which privacy preservation is particularly important, as health information is usually regarded as sensitive [24]. Thus, the health care domain [25] was chosen to evaluate the framework's effectiveness. The evaluation involved cases in which the consumers had their policies checked against the providers' policies to verify if providers' practices satisfied consumers' preferences. Thus, the evaluation included the following activities:

- Development of a domain-specific privacy ontology, with the use of a health care privacy regulation to extend the base ontology.
- Creation of a health care scenario, with the inclusion of interactions that could demonstrate the capabilities of the SOA extension.
- Definition of evaluation cases, with the specification of policies by following the created scenario and using the developed health care ontology.

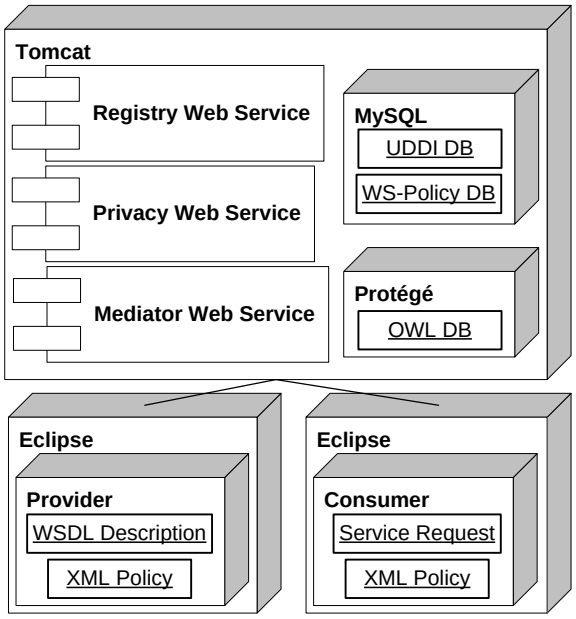

Figure 18. Prototype overview

6.2.1. Health Care Ontology: At the first step to evaluate the framework, in addition to the base ontology, a domain ontology was developed to deal with the issues that are specific to health care. The concepts from the health care ontology were referred to together with the ones from the base ontology to restrict different aspects of information use. The ontology is based on the Personal Health Information Protection Act (PHIPA) [10]. This regulation provides useful definitions for extending the base ontology to create a health care ontology. The definitions extend some aspects captured in the base ontology, including *Information*, *Collector*, *Collection*, *Recipient* and *Purpose*. For example, the concepts related to *Information* are shown in Table 1. The types are divided in two categories: *Personal Health Information* (Concept 01) and *Non Personal Health Information* (Concept 11). *Personal Health Information* is defined by a set of information types (Concepts 02-10).

Table 1: Health Care Ontology – Information

	Information	**Definition**
01	Personal Health Information	Health-related information.
02	Patient Identification	Information that can be used to identify the individual on its own or linked to another piece of information, including the individual's health insurance number.
03	Health	Information that relates to the individual's primary or mental health.
04	Family Health History	Information about the individual's family history that relates to health.
05	Health Care	Information on the health care received by the individual.
06	Health Care Provider Identification	Information that can be used to identify the health care provider responsible for providing health care to the individual.
07	Health Care Payment	Information that relates to the individual's payment for health care as well as the individual's eligibility for health care or for coverage for health care under a health insurance plan.
08	Body Part Donation	Information on the individual's donation of body parts or bodily substances.
09	Substitute Decision-Maker Identification	Information that can be used to identify the individual's substitute decision-maker.
10	Personal Health Information Accompanying Information	Information that belongs to none of the previous categories but is part of a record that contains personal health information.
11	Non Personal Health Information	Non health-related information.

6.2.2. Evaluation Scenario: The second step was the creation of a scenario, which could be used to execute the tests. The scenario was created considering the health care domain so that the ontology developed at the first step could by applied to the evaluation. A constraint for the scenario definition was to include interactions among the different parts, which could be explored in the evaluation cases to demonstrate different capabilities of the SOA extension. Figure 19 shows the scenario, which is based on examples from a PHIPA toolkit [26].

For example, in the scenario, a patient uses services provided by a mental health care service provider. In order to use the services, the patient discloses some of its health information (*Collection*). This interaction is labeled as 1 in Figure 19. In addition to mental health care services, it uses other health care-related services offered by the provider, including primary health care, as well as services unrelated to health care, such as housing and employment services. The mental health care provider employs a holistic approach, that is, it provides primary health care along with mental health care. The primary care services are not provided directly by the provider, but by a third-party health care service provider (Interaction 2). In this case, the mental health care provider, which is a custodian, discloses the health information of the patient to another custodian, the health care provider (*Recipient*).

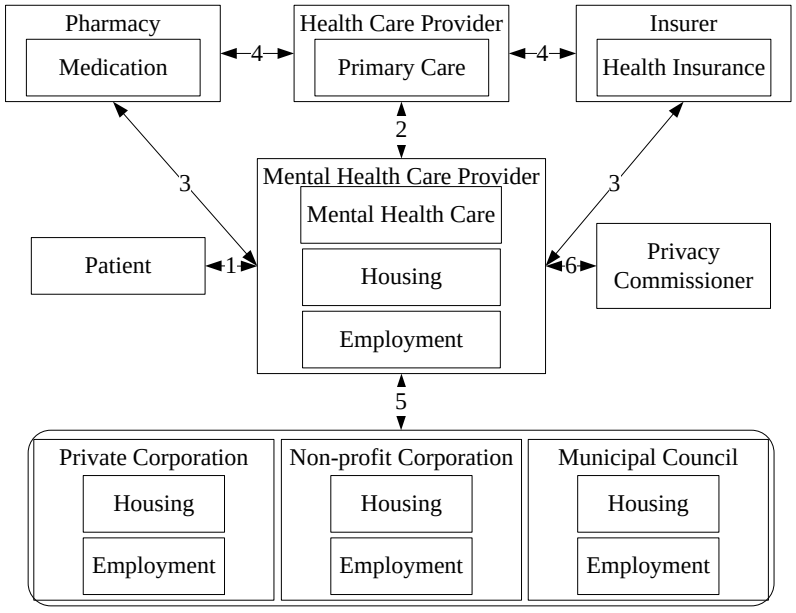

Figure 19. Evaluation scenario

6.2.3. Evaluation Cases: The last step was the evaluation case definition. The main part of the definition was the creation of the provider and consumer policies, which used the base and health care ontologies. These policies were created according to the interactions included in the evaluation scenario. The cases were then executed to demonstrate which of the interactions were possible to happen based on the policies defined for each of the involved parts. For example, a case is described as follows. For readability, the policy format is not shown in the policies.

- Evaluation Case - Health Care Provider

This case considers Interactions 1 and 2 in the scenario. It aims at exemplifying the use of domain-specific knowledge for the verification of compatibility between policies. A mental health care service provider can disclose health information about their patients to a health care service provider for the purpose of primary health care if it is authorized to do so by the original owner of the information. A third party can have the same status as the information owner for that purpose as a substitute decision maker. Thus, that third party would be able to grant the required disclosure authorization to the mental health care provider. In this case, a patient named Patient publishes a policy. In its policy, it states that a third party named ThirdParty is its substitute decision maker for the purpose of health care. Figure 20 shows this statement.

Policy Owner: Patient Information = PersonalHealthInformation Collector.ProviderName = ThirdParty Collector.ProviderType = SubstituteDecisionMaker Recipient Purpose = HealthCareRelated

Figure 20. Patient policy for substitute decision maker

Additionally, a mental health care provider named MentalProvider publishes a policy, which states that it discloses health information collected from its patients to a primary health care provider for the provision of a primary health care service if the patient allows doing so. Figure 21 shows this statement.

Policy Owner: MentalProvider Information = PersonalHealthInformation Collector.ProviderName = MentalProvider Collector.ProviderType = MentalHealthCareProvider Recipient.ProviderType = PrimaryHealthCareProvider Purpose = PrimaryHealthCare

Figure 21. Provider policy for primary health care

Continuing the case, ThirdParty, looking for a mental health care provider that follows a holistic approach for Patient, publishes its policy. It states that health information about the patient can be disclosed by the provider to a health care service provider for purposes related to health care if the patient allows doing so. This statement is shown in Figure 22.

Policy Owner: Patient Information = PersonalHealthInformation Collector.ProviderType = MentalHealthCareProvider Recipient.ProviderType = HealthCareProvider Purpose = HealthCareRelated

Figure 22. Patient policy for mental health care.

In this case, the *mediator* selected the service supplied by MentalProvider for Patient because the *privacy* known that ThirdParty was a substitute decision maker for Patient and it could make decisions on behalf of a patient if authorized to do so.

7. Conclusions

Privacy preservation in SOA still includes open problems. Two of them are that it is not possible to describe how a provider deals with private information received from a consumer as well as discover a service that satisfies the privacy preferences of a consumer in addition to the required service functionality. The framework proposed in this paper provides a novel solution for these problems. It addresses the limitations identified in frameworks presented earlier. It includes a model for semantic privacy policies and support for privacy-aware service discovery. The model enables the description of provider privacy practices and consumer preferences. In the model, policy assertions refer to ontological concepts. Thus, semantic policies are created from concepts defined in privacy ontologies. This information enriches the matching between consumer and provider policies. Policy intersection supports the discovery process that enables the discovery of services that meet consumer preferences.

Future work includes developing tools for policy specification and publication. In the proposed approach, providers have to define a policy for each service they offer, which can be difficult to some providers. As policies usually follow a similar specification, a tool could be provided to facilitate it. For instance, feature modeling could be employed by such tool to manage policy commonalities and help in the specialization of a policy to different services. In the case of consumers, it can be difficult to specify and publish their preferences as it is necessary to understand the ontologies to do so. Again, a tool to guide consumers through the specification and publication of their policies could be used. Policy templates could be created and the tool would support a consumer to configure a template and generate its policy. Such tool could help the consumer to understand the different information activities

and their privacy implications. Moreover, it would be important to have domain representative organizations for consumers and providers defining these templates for each service type in a particular application domain, which would work as default preferences and practices that then could be specialized according to the needs of consumers and providers.

In addition, the proposed approach requires providers to adhere to the practice of specifying policies. Furthermore, the *mediator* and *privacy* roles must have the capability of using policies for service publication and discovery. Thus, regulatory mechanisms are necessary to enforce these behaviors and guarantee that they are unbiased. Another future work is the inclusion of a negotiation protocol in the framework to help providers and consumers reaching an agreement in the case of incompatible policies.

The inclusion of a mechanism to check the correspondence between the policy of a provider and its actual practices is also necessary. This extension can involve mechanisms for policy enforcement and a certification solution with the use of trusted third parties to deal with issues such as providers that do not act according to their policies and obscure the details of their practices in their policies.

Other SOA privacy solutions proposed in the literature have faced difficulties to reach applicability. These difficulties show that several issues should be addressed to guarantee the practical use of the framework, including the issues discussed in this section that have not been currently addressed. Thus, the framework is an important step towards privacy preservation in service description and discovery, but other technical and non-technical solutions must be in place together with it to support its applicability entirely.

References

1. M.N. Huhns, M.P. Singh, "Service-Oriented Computing: Key Concepts and Principles", IEEE Internet Computing, vol. 9, pp. 75-81, 2005.
2. M.P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, "Service-Oriented Computing: A Research Roadmap", International Journal of Cooperative Information Systems, vol. 17, pp. 223-255, 2008.
3. F. Westin, Privacy and Freedom. Atheneum, New York, 1967.
4. P.C.K. Hung (ed.), Security and Privacy Technologies in SOA, http://www.computer.org/portal/web/buildyourcareer/ts020/, 2009.
5. E. Constante, F. Paci, N. Zannone, "Privacy-Aware Web Service Composition and Ranking". In Proceedings of the IEEE International Conference on Web Services, Santa Clara, CA, USA, pp. 131-138, 2013.
6. S. Tbahriti, C. Ghedira, B. Medjahed, M. Mrissa, "Privacy-Enhanced Web Service Composition", IEEE Transactions on Services Computing, vol. PP, pp. 1, 2013.
7. F.F. Wang, N. Griffiths, "Protecting Privacy in Automated Transaction Systems: A Legal and Technological Perspective in the European Union", International Review of Law, Computers, & Technology, vol. 24, pp. 153-162, 2010.
8. A. Malik, S. Dustdar, "Enhanced Sharing and Privacy in Distributed Information Sharing Environments". In Proceedings of the International Conference on Information Assurance and Security, Malacca, Malaysia, pp. 286-291, 2011.
9. Canada, Personal Information Protection and Electronic Documents Act, http://laws.justice.gc.ca/en/P-8.6/FullText.html/, 2000.
10. Ontario, Personal Health Information Protection Act, http://www.e-laws.gov.on.ca/html/statutes/english/elaws_statutes_04p03_e.htm/, 2004.
11. A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker (eds.), Web Services Security: SOAP Message Security 1.1, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf/, 2006.
12. S. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, Ü. Yalçinalp, Web Services Policy 1.5 - Framework, http://www.w3.org/TR/2007/REC-ws-policy-20070904/, 2007.
13. P.C.K. Hung, E. Ferrari, B. Carminati, "Towards standardized Web services privacy technologies". In Proceedings of the IEEE International Conference on Web Services, San Diego, pp. 174-181, 2004.
14. A. Tumer, A. Dogac, I. H. Toroslu, "A Semantic-Based User Privacy Protection Framework for Web Services", Lecture Notes in Computer Science, vol. 3169, pp. 289-305, 2005.
15. M. Al-Nedhami, P.K. Sinha, "A Privacy Framework for Composite Web Services". In Proceedings of the International Workshop on Service-Oriented Engineering and Optimization, Bangalore, 2008, paper no. 2.

16. M. Ali, L. Bussard, U. Pinsdorf, "Obligation Language and Framework to Enable Privacy-Aware SOA", Lecture Notes in Computer Science, vol. 5939, pp. 18-32, 2010.
17. Y. Osawa, S. Imamura, A. Takeda, G. Kitagata, N. Shiratori, K. Hashimoto, "A Proposal for Privacy Management Architecture". In Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet, Seoul, pp. 161-164, 2010.
18. G.O.M. Yee, "A Privacy Controller Approach for Privacy Protection in Web Services". In Proceedings of the ACM Workshop on Secure Web Services, Fairfax, VA, USA, pp. 44-51, 2007.
19. D.S. Allison, Privacy Protection Framework for Service-Oriented Architecture, Master of Engineering Science Thesis (Department of Electrical and Computer Engineering, University of Western Ontario, London, Canada, 2009).
20. H. Meziane, S. Benbernou, "A Dynamic Privacy Model for Web Services", Computer Standards & Interfaces, vol. 32, pp. 288-304, 2010.
21. United States of America, The Privacy Act of 1974, http://www.archives.gov/about/laws/privacy-act-1974.html/, 1974.
22. Organization for Economic Co-operation and Development, OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, http://www.oecd.org/document/18/0,3343,en_2649_34255_1815186_1_1_1_1,00.html/, 1980.
23. European Union, Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML/, 1995.
24. P. Burnap, I. Spasic, W. Gray, J. Hilton, O. Rana, G. Elwyn, "Protecting Patient Privacy in Distributed Collaborative Healthcare Environments by Retaining Access Control of Shared Information". In Proceedings of the International Conference on Collaboration Technologies and Systems, Denver, CO, USA, pp. 490-497, 2012.
25. J.M. Humber, R.F. Almeder (eds.), Privacy and Health Care. Humana Press, Clifton, 2001.
26. Canadian Mental Health Association, Community Mental Health and Addictions Privacy Toolkit, http://www.ontario.cmha.ca/privacytoolkit/docs/privacy_toolkit.pdf/, 2005.