Privacy-Aware Web Service Protocol Replaceability

Nawal Guermouche, Salima Benbernou, Emmanuel Coquery, Mohand-Said Hacid

Privacy-Aware Web Service Protocol Replaceability*

Nawal Guermouche
LORIA, INRIA Lorraine
Campus scientifique
BP 239, 54506 Vandoeuvre-Lès-Nancy, France
guermona@loria.fr

Salima Benbernou, Emmanuel Coquery and Mohand-Said Hacid
Université de Lyon
Université Claude Bernard, Lyon 1
LIRIS CNRS UMR 5205 - UFR d'Informatique
43, boulevard du 11 Novembre 1918, 69622 Villeurbanne cedex - France.
{sbenbern, ecoquery, mshacid}@liris.cnrs.fr

Abstract

Business protocols are becoming a necessary part of Web services description [4]. The work presented in [4] investigates mechanisms for analyzing the compatibility and the substitution (i.e., replaceability) of Web services based on their functional properties. In this paper, we focus on the replaceability analysis.Whether a service can replace another depends not only on their functional properties but also on non functional requirements (e.g., privacy policies). We propose a privacy-aware protocol replaceability approach to extend the work presented in [4] by privacy properties. We introduce a rule-based privacy model and we extend business protocols, leading to what we call private business protocols. Finally, a private replaceability analysis of private business protocols is discussed. We mainly investigate compatibility issues, that is whether one private business protocol can support the same set of conversations with respect to the privacy requirements.

1. Introduction

Web services are becoming one of the main technologies for designing and building complex inter-enterprise business applications. Due to standard interfaces, mainly based on XML technology (exp. WSDL [5], BPEL [2]) used by Web services to integrate heterogeneous and autonomous applications, Web services have changed the way companies manage their business applications. Since the technology of Web services is increasingly being adopted by industrial companies, security and privacy issues in Web services become an important topic of investigation. Some authors already considered access control management [12], mainly the enforcement of access control for Web services based on their conversational behavior. In this paper we consider privacy issues which arise in many situations where private data must be exchanged and handled. For example, to process a client request, business applications can require sensitive data from clients. To gain control over these data, mechanisms that can support disclosure of private data should be designed. *Policies* that are rules specifying conditions under which private data could be collected might provide a way for supporting disclosure of private data. Furthermore, for business processes based on the coordination between several Web services, if a given service becomes overcrowded or damaged, there is an impact on the general process. An alternative might be replacing a non-available service by another service able to provide the same functionalities. This mechanism is called *replaceability*. However, replaceability depends not only on the functional properties but also on non functional properties such as privacy requirements. In this paper, we investigate: (1) an approach for modeling privacy policies and their integration to business protocols [4], and (2) a non functional replaceability analysis of Web service protocols: *privacy aware replaceability*.

The rest of the paper is organized as follows. Section

*This work is partially supported by the French National Research Agency (ANR) - Program "Jeunes chercheurs:Servicemosaic" a part of the international project ServiceMosaic; http://servicemosaic.isima.fr/.

2 presents a motivating example that will be used to illustrate the different steps of the approach. Section 3 describes the rule-based privacy model we propose. Section 4 is devoted to the different scenarios regarding replaceability of privacy policies. Section 5 focuses on the extension of business protocols by the privacy policies model. Related works are presented in Section 6. We conclude in Section 7.

2. Motivating example

We consider a hotel booking service. The requester specifies the desired destination (city). Then, the service suggests a list of hotels. Once the requester has made the choice, she/he is requested to provide a *credit card number* to complete the reservation. Since this data is very sensitive, the requester might ask for a kind of privacy secure usage.

Suppose that due to a sudden collapse, a service cannot work properly. Since, the requester has provided private data, it could be interesting if one can transparently replace the damaged service. Therefore, the candidate service should (1) have the same functionalities as the damaged one and (2) ensure the same privacy level of the collected data. So far, work on replaceability [4] considers only the functional properties, the non functional aspects which are as important as privacy policies are not considered.

3. Privacy model

In this section, we introduce our model of privacy rules as an extension of the categories of rules defined in the platform of privacy preferences P3P [1]. In order to establish an interaction between a client and a provider, the client specifies through rules called *privacy preferences* a way private data can be used by the provider, and the provider specifies through rules called *privacy policies* how private data will be used. To establish a conversation between a client and a provider (the client provides its private data), the preferences of the client must be consistent with the policy of the provider.

Remark 1 *A preference $pref$ is consistent with a policy $plcy$ if the policy $plcy$ is more restrictive than the preference $pref$. We define in Section 4 how we compare the level of restriction of two policies. Since a preference is defined as a policy, we will be able to check the consistency of a preference with a policy.*

Since the service provider can also request another Web service (it becomes a client), it should specify *preferences*. Thus, each Web service owns *preferences* and *policies*. In the following, we present the ingredients that constitute a policy (respectively a preference).

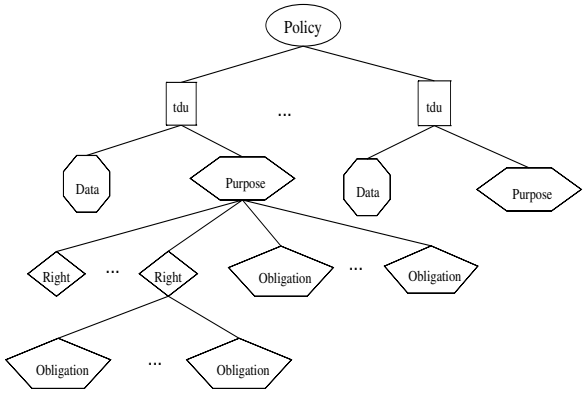

Figure 1. A privacy policy model.

As mentioned previously, *privacy policies* are rules specified by the provider describing how it will handle private data of its client. As depicted in Figure 1, a privacy policy is defined as a finite set of *Terms of Data Usage*, denoted TDU. A $tdu \in TDU$ consists of a *private data* and a *purpose* for which the private data must be collected.

A *purpose* is an action representing the need of a client and executed by a given *entity*. The entities are those that can use the data to fulfill the request (*purpose*) of the client in a given *time frame*.

Furthermore, the provider can require getting a choice to perform or not other actions called *Rights*.

Since the fulfillment of purposes and rights involves the use of the client's private data, the provider must guarantee their security. For this, it must specify actions called *obligations* intented to secure the data.

To summarize, a purpose involves two kinds of actions:

- *Rights*: a right is an action the provider is allowed to do. For each right, we specify the *entities* authorized to perform it, the *delay* during which the entities own the right and the *delay* during which the right must be achieved once activated. Also a right can induce a set of *obligations*.
- *Obligations*: an obligation is the action the provider must achieve after collecting private data to ensure their security. An obligation is specified like a right but it does not involve any action.

Definition 1 *Let us define the following sets:*
U: set of entities, D: set of data, A: set of actions, P: set of purposes, O: set of obligations, R: set of rights, I: set of intervals.

U, D, A are represented as a hierarchy by an ontology used to compare the restriction levels of two policies (section 4).

1. *A privacy policy is a set of terms of data usage tdu where tdu is an element of $D \times P$.*

2. *A purpose p is defined by the tuple $(a, u, \mu, S^R, S^O) \in A \times U \times I \times 2^R \times 2^O$, where a is the action identifying the purpose, u is the entity performing the purpose, μ is an interval in which u must perform the action a, S^R is a set of rights and S^O is a set of obligations associated with the purpose.*

3. *A right r is defined by the tuple $(a', u', \nu', \mu', S^{O'}) \in A \times U \times I \times I \times 2^O$, where a' is the action identifying the right, u' is the entity authorized to perform the action a', ν' is the delay in which the entity u' is authorized to perform the action a', μ' is the delay in which the action a' must be performed once activated.*

4. *An obligation o is defined by the tuple $(a'', u'', \nu'', \mu'') \in A \times U \times I \times I$ such that a'' is the action identifying the obligation, u'' is the entity performing the obligation, ν'' is validity time interval of the action a'', μ'' is the time interval within which the action a'' must be performed once activated.*

As the provider can disclose the data it collects to a third party, we distinguish between two kinds of entities (i) *ours* specifies the entities of the service collecting the private data, and (ii) *others* specifies third party entities for which a service can disclose the collected private data.

Example 1 *Back to the motivating example, the client must provide his credit card number CCN to pay the hotel reservation. The restrictions on CCN are as follows:*

The financial service collects the credit card number CCN to pay the corresponding hotel reservation (p_1) within 20 minutes after getting the CCN. The Bank which is an external entity owns the right to verify the CCN validity (r_1) within 15 minutes after receiving CCN. If the right (r_1) is triggered, the corresponding action must be performed within 5 minutes. The financial service must destroy the CCN (o_1) 10 minutes after the achievement of the purpose. Moreover, the bank must encrypt the CCN (o_2) within 60 minutes following the CCN reception. The deletion (o_1) and the coding (o_2) must be performed immediately after their triggering, specified by an empty interval time [0,0]. Thus, the corresponding policy for CCN can be represented as follows:

$plcy = \{(CCN, p_1)\}$ where
$p_1 = (PayReservation, Ours : FinancialService, \mu : [0, 20\ min], \{r_1\}, \{o_1\})$
$r_1 = (ValidityVerification, Others : Bank, \nu : [0, 15\ min], \mu : [0, 5], \{o_2\})$
$o_1 = (Destruction, Ours : FinancialService, \nu : [t(p_1),\ t(p_1) + 10min], \mu : [0, 0])$ *such that* $t(p_1) \in [0, 20min]$ *is the time interval for fulfilling the purpose.*
$o_2 = (Encrypt, Others : Bank, \nu : [0, 60min], \mu[0, 0])$

3.1. Preference

A client service can send its own private data to a provider, so it should specify through rules called *local privacy preferences* how it wishes the provider to use its private data. Furthermore, the provider, seen as a client, can send the collected private data to a third party. Thus, it should specify through rules called *external preferences* how the third party must use private data. The *local* and *external preferences* constitute the *preferences* of the client.

Similar to a privacy policy, a preference is defined by a finite set of terms of data usage. However, we do not distinguish between the internal (*Ours*) and external (*Others*) entities. In fact, the client ignores whether the entities are considered as a third party (*Others*) or not (*Ours)* for the service collecting the private data.

3.2. Extraction of external preferences from policies

Since a service Q can invoke other services, hence it can disclose the private data of its client to a third party, as depicted in Figure 2.

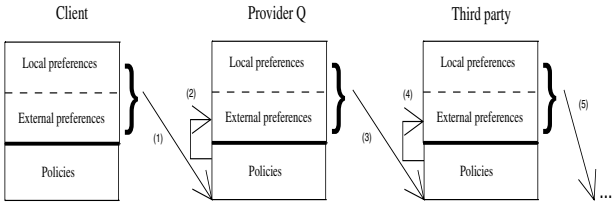

Figure 2. Extraction of the external preferences from policies.

The interaction between the service Q and a third party is based on the preferences of the service Q and the policies of the third party (3). To keep informed a third party of the restrictions of the client through the Web service Q, we add these restrictions in the external preferences of the service Q (2) as depicted in Figure 2. Thus, we propagate the set of tdu of the service Q policy to external preferences (2). Hence, we add each tdu of the policy to the set TDU of preferences when the entity responsible to fulfill a purpose (right, obligation respectively) is an external entity ($Others$).

4. Replaceability analysis of privacy policies

In order to extend the replaceability of Web services defined in [4] by accommodating privacy requirements, we provide a privacy aware replaceability analysis of policies. To compare two policies, we propose to define an order between two policies based on their restriction levels.

4.1. Comparing the restriction levels of two policies

In this section, we give an intuitive idea on how we compare two policies based on their restriction levels. Figure 3 shows parts of available hierarchies: *hierarchy of purposes*, *hierarchy of users* and *hierarchy of data*.

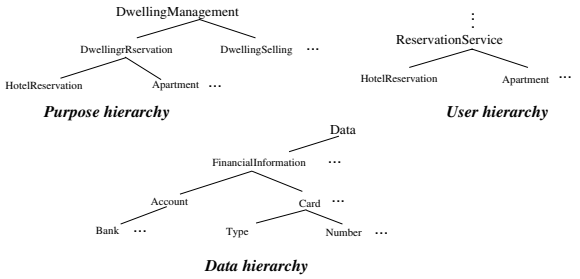

Figure 3. Examples of hierarchy.

Hierarchies are used to compare the data, actions and users related to purposes (rights and obligations respectively) as explained in the following example.

Example 2 *Let us consider the two policies* $plcy_1$ *and* $plcy_2$ *such that:*

$plcy_1$=$\{(FinancialInformation, p_1)\}$ *where:*
p_1= $(ReserveDwelling, Ours : ReservationServices, \mu : [0, 30min], \{r_1\}, \{o_1\})$
r_1= $(ValidityVerification, Others : FinancialPartner, \nu : [0, 15min], \mu : [0, 5], \{o_2\})$
o_1= $(Destruction, Ours : ReservationServices, \nu : [t(p_1), t(p_1) + 10min], \mu : [0, 0])$ *such that* $t(p_1) \in [0, 30min]$ *is the time interval for fulfilling the purpose* p_1.
o_2= $(Encrypt, Others : FinancialPartener, \nu : [0, 60 min], \mu[0, 0])$

$plcy_2$=$\{(NumberCard, p_2)\}$ *where:*
p_2= $(ReserveHotel, Ours : HotelServices, \mu : [0, 20 min], \{r'_1\}, \{o'_1\})$
r'_1= $(ValidityVerification, Others : Bank, \nu : [0, 15min], \mu : [0, 5], \{o'_2\})$
o'_1= $(Destruction, Ours : FinancialServices, \nu : [t(p_2), t(p_2) + 10min], \mu : [0, 0])$ *such that* $t(p_2) \in [0, 20min]$ *is the time interval for fulfilling the purpose* p_2.
o'_2= $(Encrypt, Others : Bank, \nu : [0, 60min], \mu[0, 0])$

For instance, we can observe that the policy $plcy_2$ *is more restrictive than the policy* $plcy_1$ *since the private data considered in the policy* $plcy_1$ *concerns the financial information, and as it is depicted in Figure 3, the financial information gathers the account and card information. Thus the data specified in* $plcy_2$ *(credit card number) is more restrictive (more specific) than the data of* $plcy_1$ *(financial information). Moreover, the purpose specified in the policy* $plcy_1$ *consists in dwelling reservation which gathers hotel reservation and apartment (see figure 3), which is more restrictive than the purpose of the policy* $plcy_2$*, since hotel reservation is a sub-purpose of dwelling reservation. Furthermore, the entity that must achieve the purpose of the policy* $plcy_1$ *is the reservation service which contains the hotel service specified in the policy* $plcy_2$*. In the policy* $plcy_1$ *reservation service must perform the dwelling reservation within* 30 *minutes after the reception of the financial information. The interval specified in the policy* $plcy_2$ *is included in the one specified in the policy* $plcy_1$*, hence the interval of* $plcy_2$ *is more restrictive than the interval of* $plcy_1$*. Similar to what we did with data, purpose and time interval to verify if the policy* $plcy_2$ *is more restrictive than the policy* $plcy_1$*, we do the same with the other elements.*

Next we present how we compare obligations, rights, purposes and terms of data usage based on their restriction levels to perform the privacy-aware replaceabity of policies.

Comparing obligations $\lesssim_O$**.** An obligation $o_2 = (a''_2, u''_2, \nu''_2, \mu''_2)$ is more restrictive than an obligation $o_1 = (a''_1, u''_1, \nu''_1, \mu''_1)$ denoted by $o_1 \lesssim_O o_2$ if and only if $a''_2 \sqsubseteq a''_1$, $u''_2 \sqsubseteq u''_1$, $\nu''_2 \subseteq \nu''_1$, $\mu''_2 \subseteq \mu''_1$. $X \sqsubseteq Y$ *stands for* X *is subsumed by* Y.

As a right involves a set of obligations, the above comparisons of obligations allows the following comparison of rights.

Comparing Rights $\lesssim_{\mathcal{R}}$**.** A right $r_2 = (a'_2, u'_2, \nu'_2, \mu'_2, S_2^{O'})$ is more restrictive than a right $r_1 = (a'_1, u'_1, \nu'_1, \mu'_1, S_1^{O'})$ denoted by $r_1 \lesssim_{\mathcal{R}} r_2$ if and only if $a'_2 \sqsubseteq a'_1$, $u'_2 \sqsubseteq u'_1$, $\nu'_2 \subseteq \nu'_1$, $\mu'_2 \subseteq \mu'_1$, $\forall o'_1 \in S_1^{O'}, \exists o'_2 \in S_2^{O'}$ such that $o'_1 \lesssim_O o'_2$.

As a purpose involves a set of rights and obligations, the above comparison of rights and obligations allows the following comparison of purposes.

Comparing Purposes $\lesssim_{\mathcal{P}}$**.** A purpose $p_2 = (a_2, u_2, \nu_2, S_2^R, S_2^O)$ is more restrictive than a purpose $p_1 = (a_1, u_1, \nu_1, S_1^R, S_1^O)$ denoted by $p_1 \lesssim_{\mathcal{P}} p_2$ if and only if $a_2 \sqsubseteq a_1$, $u_2 \sqsubseteq u_1$, $\nu_2 \subseteq \nu_1$, $\forall r_2 \in S_2^R, \exists r_1 \in S_1^R$ such that $r_1 \lesssim_{\mathcal{R}} r_2$, $\forall o_1 \in S_1^O, \exists o_2 \in S_2^O$ such that $o_1 \lesssim_O o_2$.

As a term of data usage defines a purpose for which the

private data is collected, the above comparison of purposes allows the following comparison of terms of data usage.

Comparing terms of data usage $\lesssim_{TDU}$. A term of data usage $tdu_2 = (d_2, p_2)$ is more restrictive than a term of data usage $tdu_1 = (d_1, p_1)$ denoted by $tdu_1 \lesssim_{TDU} tdu_2$ if and only if $d_2 \sqsubseteq d_1 \wedge p_1 \lesssim_{\mathcal{P}} p_2$.

Since a policy is defined by a finite set of terms of data usage, the above comparison of terms of data usage allows the following comparison of policies.

Comparing policies $\lesssim_{private}$. A policy $plcy_2$ is more restrictive than a policy $plcy_1$ denoted by $plcy_1 \lesssim_{private} plcy_2$ if and only if $\forall tdu_1 = (d_1, p_1) \in plcy_1, \exists tdu_2 (d_2, p_2) \in plcy_2$ such that $tdu_1 \lesssim_{TDU} tdu_2$.

Remark 2 *A preference $pref$ is consistent with a policy $plcy$ if $plcy$ is more restrictive than $pref$ ($pref \lesssim_{private} plcy$). Since a preference is defined as a policy by a finite set of terms of data usage tdu, we can use the above definition for comparing two policies to check the consistency between $pref$ and $plcy$.*

4.2. Privacy replaceability of policies

In this section, we distinguish three classes of privacy-aware replaceability of policies which are: (1) *private replaceability*, (2) *private equivalence*, and (3) *private partial replaceability*.

4.2.1 Private replaceability

A policy $plcy_1$ can be replaced by a policy $plcy_2$, denoted $plcy_1 \preccurlyeq plcy_2$ if all the restrictions supported by $plcy_1$ are also supported by $plcy_2$.

Furthermore, to replace a policy $plcy_1$ by a policy $plcy_2$, we must ensure that the policy $plcy_2$ does not violate the policy $plcy_1$. The violation occurs when for the same data d, the policy $plcy_2$ specifies purposes (respectively rights and obligations) that are not specified in $plcy_1$.

Example 3 *For instance, we see that in the policy $plcy_2$, the credit card number can be used to reserve a car which is not authorized in the policy $plcy_1$. So $plcy_2$ violates the policy $plcy_1$.*

$plcy_1 = \{(Financialnfo, p_1)\}$ *such that:*
$p_1 = (ReserveDwelling, Ours : ReservationServices, \mu : [0, 30min], \{r_1\}, \{o_1\})$
$r_1 = (ValidityVerification, Others : FinancialPartner, \nu : [0, 15min], \mu : [0, 5], \{o_2\})$
$o_1 = (Destruction, Ours : ReservationServices, \nu : [t(p_1), t(p_1) + 10min], \mu : [0, 0])$ *such that* $t(p_1) \in [0, 30min]$ *is the time interval for fulfilling the purpose* p_1.
$o_2 = (Encrypt, Others : FinancialPartner, \nu : [0, 60min], \mu[0, 0])$

$plcy_2 = \{(NumberCard, p'_1), (NumberCard, p'_2)\}$ *such that:*
$p'_1 = (ReserveHotel, Ours : HotelServices, \mu : [0, 20min], \{r'_1\}, \{o'_1\})$
$r'_1 = (ValidityVerification, Others : Bank, \nu : [0, 15min], \mu : [0, 5], o'_2)$
$o'_1 = (Destruction, Ours : HotelServices, \nu : [t(p_2), t(p_2) + 10min], \mu : [0, 0])$ *such that* $t(p_2) \in [0, 20min]$ *is the time when the purpose p_2 was fulfilled.*
$o'_2 = (Encrypt, Others : Bank, \nu : [0, 60min], \mu[0, 0])$

$p'_2 = (CarReservation, Ours : TransportServices, delay : [0, 20min], \{r'_2\}, \{o'_3\})$
$r'_2 = (ValidityVerification, Others : Bank, \nu : [0, 15min], \mu : [0, 5], o'_4)$
$o'_3 = (Destruction, Ours : TransportServices, \nu : [t(p_3), t(p_3) + 10min], \mu : [0, 0])$ *such that* $t(p_3) \in [0, 20min]$*is the time interval for fulfilling the purpose* p_3.
$o'_4 = (Encrypt, Others : Bank, \nu : [0, 60min], \mu[0, 0])$

Definition 2 *(Private replaceability $\preccurlyeq$.) A policy $plcy_1$ can be replaced by a policy $plcy_2$, denoted by $plcy_1 \preccurlyeq plcy_2$, if and only if the following conditions hold:*

- $plcy_1 \lesssim_{private} plcy_2$.
- $\forall tdu_2 = (d_2, p_2) \in plcy_2$, $\nexists tdu_1 = (d_1, p_1) \in plcy_1$ *such that* $d_2 \sqsubseteq d_1 \wedge \neg(p_1 \lesssim_{\mathcal{P}} p_2)$.

In other words, the first condition says that the policy $plcy_2$ must be more restrictive than the policy $plcy_1$ and the second condition says that the policy $plcy_2$ does not violate the policy $plcy_1$.

4.2.2 Private equivalence

Two policies are equivalent if they have the same level of restrictions.

Definition 3 *(Private equivalence $\equiv$.) A policy $plcy_1$ is said equivalent to the policy $plcy_2$ denoted by $plcy_1 \equiv plcy_2$ if and only if $plcy_1 \preccurlyeq plcy_2 \wedge plcy_2 \preccurlyeq plcy_1$.*

4.2.3 Private partial replaceability

When the policy $plcy_2$ does not replace the policy $plcy_1$, but at least one tdu can be replaced, we say that the policy $plcy_2$ can partially replace the policy $plcy_1$.

Definition 4 *(Private partial replaceability.)*
The policy $plcy_2$ partially replaces a policy $plcy_1$ if and only if the following conditions hold:

- $\exists tdu_1 \in plcy_1, \exists tdu_2 \in plcy_2$ *such that* $tdu_1 \lesssim_{TDU} tdu_2$.

- $\exists tdu_1 \in plcy_1, \nexists tdu_2 \in plcy_2$ *such that* $tdu_1 \lesssim_{TDU} tdu_2$.
- $\forall tdu_2 = (d_2, p_2) \in plcy_2, \nexists tdu_1 = (d_1, p_1) \in plcy_1$ *such that* $d_2 \sqsubseteq d_1 \wedge \neg(p_1 \lesssim_{\mathcal{P}} p_2)$.

The first condition says that the policy $plcy_2$ must replace at least one tdu of the policy $plcy_1$ and the second condition says that there is at least one tdu of the policy $plcy_1$ that cannot be replaced by the policy $plcy_2$. The last condition ensures that the policy $plcy_2$ does not violate the policy $plcy_1$.

5. Annotation of business protocols with privacy rules

We present how to integrate our model of privacy rules into business protocols. This will lead to *private business protocols* which enable considering the privacy aspects in the replaceability analysis of Web services. A *business protocol* aims at specifying the external behaviour (the sequences of supported messages) of a Web service. Business protocols are specified as finite state machines, where states correspond to different states of the service and the transitions correspond to exchanged messages (input and output messages). Each transition is labelled with a message name followed by the message polarity indicating whether the message is incoming (plus sign) or outgoing (minus sign) [4].

5.1. Private Business protocols

Our first extension consists in extending the polarity function[1] to consider the different aspects of privacy. Moreover, we extend the definition of transitions and states of business protocols.

5.1.1 Polarity

To specify that the input (respectively the output) message imports (respectively exports) private data (for short, we say private message), we propose to extend the polarity function by defining two variants according to the type of messages:

- *Polarity of incoming messages*: This polarity indicates if the message imports private data of the clients (see Definition 5).
- *Polarity of outgoing messages*: This polarity indicates if the message exports its own private data or those of its clients (see Definition 5).

[1] +: Input message, -: Output message

5.1.2 Transitions

In business protocols, privacy policies must be associated with input private messages. Therefore, we propose to annotate each transition enabling an input private message by the corresponding policies.

5.1.3 States

In business protocols, a state can be a source for a set of transitions enabling an output private message. Hence, the corresponding preferences are associated with a state.

The following definition extends the traditional definition of business protocols by incorporating privacy aspects.

Definition 5 *A private business protocol Q is a tuple $Q = (S, s_0, F, M, PREF, \eth, PLCY, T)$ which consists of the following components:*

- *S is a finite set of states, where $s_0 \in S$ is the initial state.*
- *$F \subseteq S$ is a set of final states. If $F = \emptyset$, then Q is said to be an empty protocol.*
- *M is a finite set of messages. For each message $m \in M$, we define two variants of the function Polarity(Q,m):*
  - *The polarity of input messages has the form of $IPolarity(Q, m) = (+, ClientPData)$ such that:*

$$\textit{ClientPData} = \begin{cases} 1 & \textit{if the message m imports private data of clients.} \\ 0 & \textit{otherwise} \end{cases}$$

  - *The polarity of output messages has the form of $OPolarity(Q, m)$=$(-, ClientPData, ServicePData)$*

$$\textit{ServicePData} = \begin{cases} Y & \textit{if the message m exports private data of the service issuing the message m} \\ N & \textit{otherwise} \end{cases}$$

- *$PREF$ is a finite set of preferences.*
- *$\eth$: $S \rightarrow 2^{PREF}$ assigns a set of preferences to states.*
- *$PLCY$ is a finite set of policies.*
- *$T \subseteq S^2 \times M$ is a finite set of transitions. Each transition $(s, s', m, plcy)$ identifies a source state s, a target state s', a message m, the corresponding policies*

$plcy \subseteq PLCY$ (if m is an input private message) and the corresponding preferences $\eth(s)$ assigned to the state s (if m is an output private message). In this case, we say that the message m is enabled from a state s:

- If m is an input private message, then $plcy \neq \emptyset$.

- If m is an output private message, then $\eth(s) \neq \emptyset$.

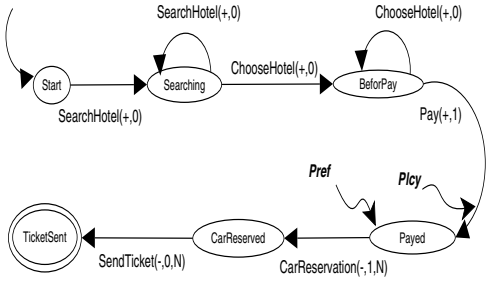

Figure 4. An example of a private business protocol Q.

Example 4 *Figure 4 displays a graphical representation of a private business protocol Q of a Web service that allows the reservation of a hotel, presented in the motivating example. The message $Pay(+,1)$ is an input message $(+)$ which imports a private data (1). Hence, we annotate the corresponding transition by the appropriate policy. Moreover, the message $CarReservation(-,1,N)$ is an output message $(-)$ which exports the private data of clients (1) and does not export the own private data of the service (N). So we annotate the source state of this transition by the corresponding preference:*

$plcy = \{(CCN, p_1)\}$ such that
$p_1 = (PayReservation, Ours : FinancialService, \mu : [0, 20\ min], \{r_1\}, \{o_1\})$
$r_1 = (ValidityVerification, Others : Bank, \nu : [0, 15min], \mu : [0, 5], \{o_2\})$
$o_1 = (Destruction, Ours : FinancialService, \nu : [t(p_1), t(p_1) + 10min], \mu : [0, 0])$ *such that* $t(p_1) \in [0, 20min]$ *is the time interval for fulfilling the purpose.*
$o_2 = (Encrypt, Others : Bank, \nu : [0, 60min], \mu[0, 0])$

$pref = \{(Name, p'_1)\}$ such that
$p'_1 = (CarReservation, u : ReservationAgency, \mu : [0, 20\ min], \{o'_1\})$
$o'_1 = (Destruction, u : ReservationAgency, \nu : [t(p'_1), t(p'_1) + 10min], \mu : [0, 0])$ *such that* $t(p'_1) \in [0, 20min]$ *(i.e., the time when the purpose p'_1 was fulfilled).*

Consider a private data d which is handled by two input messages m_1 and m_2 of a service provider. So, we associate the message m_1 with the corresponding policy $plcy_1$ and the message m_2 with the corresponding policy $plcy_2$. Suppose that the preference of the client (that invokes this service provider) is consistent with the policy $plcy_1$ and is inconsistent with the policy $plcy_2$. In our approach, this inconsistency does not forbid the conversation between the client and the provider unless the invoked message is m_2. However, the traditional approaches consist in checking all the rules (a global policy) related to the data d which prevents the conversation between the client and the provider even if the rule does not deal with the invoked message.

5.2. Simulation of private protocols

Informally, a protocol Q is simulated (replaced) by a protocol Q' if, starting from the initial state, each input (respectively, output) message of Q can be matched with an input (respectively, output) message of Q' [4]. Here, we introduce an extended definition of simulation to accommodate privacy policies.

Definition 6 *Let $Q = (S, s_0, F, M, PREF, \eth, PLCY, T)$ and $Q' = (S', s'_0, F', M', PREF', \eth', PLCY', T')$ be two protocols. A protocol Q' simulates a protocol Q denoted $Q \precsim Q'$ if and only if there exists:*

- *A relation $\Gamma \subseteq S \times S'$ such that the following holds:*
  - *$\forall (s_1, s'_1) \in \Gamma$ and $\forall T(s_1, s_2, m, plcy)$, there exists s'_2 such that $T'(s'_1, s'_2, m', plcy')$, $m \sqsubseteq m'$, $Polarity\ (Q, m) = Polarity(Q', m')$, $(s_2, s'_2) \in \Gamma$, $plcy \preccurlyeq plcy'$.*
  - *$\forall (s, s') \in \Gamma$, if $s \in F$ then $s' \in F'$*

Remark 3 *In order to replace a Web service Q, the first step consists in discovering a set of Web services. Suppose that among all the discovered Web services, there is no service that can fully replace Q according to their private policies. It could be interesting to choose between all of those services, the one likely to have the best neighbouring policies (ensuring the highest level of restrictions with respect to the policies of Q). As the neighbouring replaceability can be partial, it will not be automatic and will require the authorization of the client. The full study of neighbouring replaceability goes beyond the scope of this paper.*

When we replace a Web service Q by a new one Q', Q' will use its own private data, so we do not have to check the local preferences. But Q' may also use the private data of the clients of Q, so obviously we have to check the consistency between the external preferences of Q and Q'. Since the external preferences are extracted from policies already checked in the private replaceability of Q by Q', they are also implicitly checked.

6. Related work

Some authors investigated expressive privacy models (e.g., [6, 7, 10, 9, 13]). The Platform for Privacy Preferences (P3P) is a platform for standardization and specification of privacy rules for Web sites [1]. It enables a Web site user to gain control over his private information. P3P provides mechanisms for Web site owners to express their privacy rules called *policies* in a standard format that a user can programmatically check against his privacy rules (called *preferences*), to decide whether to release his data to the Web site. With respect to our model, to ensure the security of the data, P3P specifies only the saving time interval of private data. However, sometimes the owner needs to specify other actions to ensure the security of the private data which is considered in our model (for example keep the data encrypted). Moreover, P3P does not specify the time interval within which the purpose must be performed as it is specified in our model.

The work presented in [8] deals with the integration of privacy policies in semantic Web services. The authors extended the semantic description of OWL-S. The main advantage of this approach consists in preserving a security of the private data during their submission. However, many aspects such as temporal aspects attached to the data usage are not considered.

In the work presented in [4], the authors present an approach to verify a dynamic substitution (replaceability) of Web services regarding their business protocols based on a protocols intersection operator. This work was extended by temporal constraints [3]. This approach considers only conversations supported by business protocols and their temporal constraints. Nevertheless, Web services are constrained to use private data, which require to consider not only the functional properties, but also the non functional ones. With respect to the work presented in this paper, we have presented an approach that takes into account privacy properties of Web services in the replaceability analysis.

7. Conclusion

In this paper, we have presented an approach dealing with replaceability of Web services by considering privacy policies. We have proposed a rule-based privacy model and extended business protocols. Moreover, we have proposed three classes of replaceability of privacy policies which enable to analyze the replaceability of private business protocols.

As future work, we plan to enhance our private model rules with prohibitions and penalties. Moreover, we plan to extend the proposed model in order to express different temporal aspects. Furthermore, the management of temporal constraints [3, 11] with respect to time intervals of purposes (respectively rights and obligations) is also an interesting issue.

Another direction of research is the investigation of an approach that should be able to allow best neighbouring private replaceability when full replaceability is not feasible. In order to improve the flexibility of the private replaceability, we aim to increase the level of the privacy restrictions.

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Implementing p3p using database technology. *International Conference on Data Engineering*, 2003.

[2] T. Andrews, F. Curbera, H. Dolakia, J. Goland, J. Klein, F. Leymann, K. Lieu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weeravarna. Business process execution language for web serices (version 1.1). 2003.

[3] B. Benatallah, F. Casati, J. Ponge, and F. Toumani. On temporal abstractions of web service protocols. *Proceedings of CAISE Forum 2005*, November 2005.

[4] B. Benatallah, F. Casati, and F. Toumani. Analysis and management of web service protocols. *23rd International Conference on Conceptual Modeling*, November 2004.

[5] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (wsdl) 1.1. http://www.w3.org/tr/wsdl. 2001.

[6] Y. Elovici, B. Shapira, and A. Maschiach. A new privacy model for hiding group interests while accessing the web. *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, October 2002.

[7] S. Fischer-Hûbner and A. Ott. From a formal privacy model to its implementation. *Proceedings of the 21st National Information Systems Security Conference*, October 1998.

[8] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara. Authorization and privacy for semantic web services. *IEEE Intelligent Systems (Special Issue on Semantic Web Services)*, 2004.

[9] G. Karjoth and M. Schunter. A privacy policy model for enterprises. *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 2002.

[10] G. Karjoth, M. Schunter, and M. Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. *Workshop on Privacy Enhancing Technologies*, 2002.

[11] R. Kazhamiakin, P. Pandya, and M. Pistore. Timed modelling and analysis in web service composition. *First International Conference on Availability, Reliability and Security (ARES'06)*, 2006.

[12] M. Macella, M. Ouzzani, F. Paci, and E. Bertino. Access control enforcement for conversation-based web services. *Worl Wide Web*, 2006.

[13] M. C. Mont. Dealing with privacy obligations in enterprises. *HP Labs Technical Report*, 2004.