A Privacy Agreement Model for Web Services [*]

Salima Benbernou [1], Hassina Meziane [1], Yin Hua Li [2], Mohand-Said Hacid [1]
[1]LIRIS, University Lyon1, France
{sbenbern,mshacid}@liris.univ-lyon1.fr, meziane_has@yahoo.fr
[2]CSE, University New South Wales, Sydney, Australia
yinhual@cse.unsw.edu.au

Abstract

Web services among of the applications involving closely the customer's private information. In order to take into account the privacy concerns of the individuals, organizations (e.g Web services) provide privacy policies as promises describing how they will handle personal data of the individual. However, privacy policies do not convince potential individuals to disclose their personal data, do not guarantee the protection of personal information, and do not provide how to handle a possible evolution of the policies. In this paper, we introduce a framework based on an agreement as a solution to these problems. It contains a privacy model defined in the policy level of the agreement. The framework supports in the negotiation level of the agreement a lifecycle management which is an important deal of a dynamic environment that characterizes Web services. A negotiation protocol is proposed that enable ongoing privacy negotiation to be translated into a new privacy agreement.

1 Introduction

Nowadays, the individuals are becoming more and more concerned about the privacy of their personal data [5, 1, 6]. These concerns might lead to a situation where the customers do not trust the web service any more and take their business somewhere else [11]. So, the important enabling factor for a well usage of online services is building customers confidence with service providers when the latter comes to handle their personal data. Privacy policies are used by web services in order to ease the privacy concerns of their clients and to adhere to legislative measures, stating what they would do or not with the personal information of their clients. However, privacy policies alone are not sufficient to convince potential clients to disclose their personal data to the service provider and do not *guarantee* the protection of personal information of data subject. Privacy policies are merely promises and a promise as such sometimes has not legal grounds on which the service provider does not keep its promise. There is a need for something more trustworthy, more formal and more legal than promises -a *privacy agreement*-. Moreover, in the dynamic Web service environment, policies might need to accommodate new business strategies, changes (evolution) to laws and regulations, emerging competitors, and so on. A lifecycle management framework of privacy agreement is needed. It shows how to take into consideration the *dynamic privacy policy evolution* and how to make a consistent update in the privacy agreement induced from the events occurring in the environment, while there are active processes in the service based on the privacy policy being changed.
In this paper we propose a framework for the privacy in the Web services. The privacy policy model is defined as an agreement and supports lifecycle management which is an important deal of a dynamic environment that characterizes Web services based on the state machine, taking into account the flow of the data use in the agreement. In this setting, the features of the framework are:

- The privacy policy and data subject preferences are defined together as one element called *Privacy-agreement*, which represents a contract between two parties, the service customer and the service provider within a validity. We provide abstractions defining the expressiveness required for the privacy model, such as rights and obligations. This part of the agreement is called *policy level*. The private data use flow is presented as a state machine in this level.

- The framework supports lifecycle management of privacy agreement. We defined a set of events that may occur in the dynamic environment, and a set of change actions used to modify privacy agreement. An *agreement-evolution* model is provided in the privacy-agreement. This part of the agreement is called *nego-*

[*] 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/.

tiation level.

- An *agreement-negotiation protocol* is provided to build flexible interactions and conversation between parties when a conflict happens due to the events occurring in the dynamic environment of the Web service.

The remainder of the paper is structured as follows. Section 2 presents a formal model for privacy data in web services. Section 3 proposes an extension of WS-Agreement taking into account the previous model of data privacy as a privacy agreement and the evolution of the privacy policy. Section 4 presents the flow of the data use and the lifecycle of the privacy-agreement. Section 5 discusses the privacy agreement negotiation protocol taking into account evolution in the privacy agreement.

2 Privacy data Model

Based on our previous works [7, 8], informally speaking the abstraction of privacy model is defined in terms of the following requirements:

- *data-right*, is a predefined action on data the data-user is authorized to do if he wishes to.
  We distinguish two types of actions (i) actions used to complete the service activity for the current purpose for which it was provided and are denoted by $Op_{current}$ (ii) actions used by a service to achieve other activities than those for which they are provided, called $Op_{extra-activity}$.
- *data-obligation*, is the expected action to be performed by service provider or third parties (data- users) when handling personal data. This type of obligation is related to the management of personal data in terms of their selection, deletion or transformation.

Formally speaking, we define data-right and data-obligation as follows :

Definition 1 ***(data-right.)*** *A data-right r_d is a tuple (u, d, p, μ_r), with $u \subseteq \mathcal{U}$ and $d \subseteq \mathcal{D}$ and $p \subseteq \mathcal{PO}$ and $\mathcal{R}^d = \{\{r_d^i\}_j \ / \ i > 0 \ j > 0\}$, where $\mathcal{U}$ is the ontology of data users and $\mathcal{D}$ is the ontology of personal data and $\mathcal{PO}$ is the set of authorized operations identifying purposes of the service and μ_r is the period of data retention (the data-right validity), and $\mathcal{R}^d$ is the set of data-rights.*

Example 1 $r_{email}(sp, email, send \quad Offer, [d_s, d_s + 1\ month])$,
specifies that the service provider sp has the right to use $email$ *for sending the available products and their prices during* $1\ month$ *after both sides signed the agreement at* d_s *date.*

Definition 2 ***(data-obligation.)*** *A data-obligation o_d is a tuple (u, d, a_o, μ_o) with $u \subseteq \mathcal{U}$ and $d \subseteq \mathcal{D}$ and $a_o \in \mathcal{A}_o$ and $\mathcal{O}^d = \{\{o_d^i\}_j \ / \ i > 0 \ j > 0\}$, where $\mathcal{U}$ is the ontology of data users and $\mathcal{D}$ is the ontology of personal data and $\mathcal{A}_o$ a set of actions that must be taken by the data user and μ_o is an activated date of the obligation, and $\mathcal{O}^d$ is the set of data-obligations.*

Example 2 *In order to protect a private data, the service provider sp must delete a credit card number ccn for a given data subject at the end of each process of the payment, for instance, at d_{pay}+1 day. This obligation is expressed as :* $o_{ccn}(sp, ccn, delete, [d_{pay}, d_{pay} + 1\ day])$

Definition 3 ***(A privacy Data Model.)*** *A privacy data model $\mathcal{P}^d$ is a couple $< \mathcal{R}^d, \mathcal{O}^d >$, where $\mathcal{R}^d$ is the set of data-rights and $\mathcal{O}^d$ is the set of data-obligations.*

Next we propose an extension of WS-agreement taking into account the privacy contraints and their evolution in the behavior of the service.

3 Extended WS-Agreement structure

WS-Agreement [2, 13] specifies an XML-based language for creating contracts, agreements and guarantees from offers between a service provider and a client. An agreement may involve multiple services and includes fields for the parties, references to prior agreements, service definitions and guarantee terms.

Current WS-Agreement specifications do not support the privacy structure and do not include the possibility to update the agreement at runtime. The proposed extension is reflected in a new component in a WS-Agreement called ***Privacy-agreement***,

3.1 Privacy agreement structure

A privacy-agreement structure is represented in two levels :

(1) *policy level*, it specifies the *Privacy-Data term* including guarantees dealing with privacy-data model defined in section 2.

(2) *negotiation level*, it specifies all possible events that may happen in the service behavior, thus evolving the privacy guarantee terms defined in the policy level. Negotiation terms are all possible actions to take if the guarantee of privacy terms are not respected and a conflict arises. They are used through a negotiation protocol between the service provider and the customer.

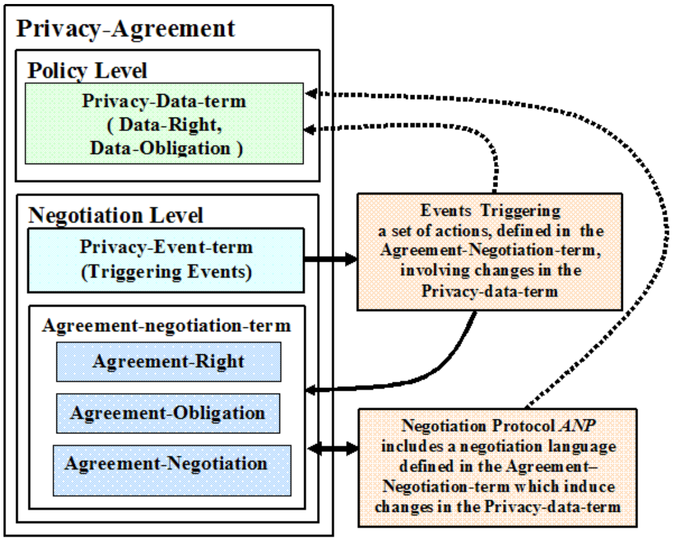
Privacy-Agreement
Policy Level
Privacy-Data-term (Data-Right, Data-Obligation)
Negotiation Level
Privacy-Event-term (Triggering Events)
Agreement-negotiation-term
Agreement-Right
Agreement-Obligation
Agreement-Negotiation
Events Triggering a set of actions, defined in the Agreement-Negotiation-term, involving changes in the Privacy-data-term
Negotiation Protocol ANP includes a negotiation language defined in the Agreement–Negotiation-term which induce changes in the Privacy-data-term

Figure 1. The Privacy Agreement structure

3.1.1 Privacy-Data term

Privacy-data term represents the policy level of privacy-agreement, defined as a set of clauses of the contract between the provider and the customer. The description of the elements defined in privacy-data model in section 2 is embedded in this level.

Definition 4 ***(Data-guarantee)***
A data-guarantee g is a couple (r_d, o_d) with $r_d \in \mathcal{R}^d$ and $o_d \in \mathcal{O}^d$, where $\mathcal{R}^d$ is a set of rights on personal data, and $\mathcal{O}^d$ is a set of obligation on personal data defined in the privacy data model $\mathcal{P}^d$. $\mathcal{G}^d \subseteq \mathcal{R}^d \times 2^{\mathcal{O}^d}$ is a set of guarantees.

Definition 5 ***(Privacy-guarantee term)***
A privacy-guarantee term t_d is a couple (d, g) with $d \in \mathcal{D}$ and $g \in \mathcal{G}^d$, where $\mathcal{D}$ is a set of personal data and $\mathcal{G}^d$ is a set of data guarantees. $\mathcal{T}^d \subseteq \mathcal{D} \times 2^{\mathcal{G}^d}$ is a set of terms t_d.

We also define in this level the validity period of privacy agreement and a set of penalties when the requirements are not fulfilled.

Definition 6 ***(Privacy-agreement validity)***
A privacy agreement validity μ is defined by a tuple (Id_A, d_s, α), with Id_A is an agreement identifier, and d_s is an absolute time indicating when the privacy-agreement was signed, and $\alpha \in [d_s, t]$ is an interval time indicating the validity period of the privacy agreement.

Definition 7 ***(Penalty)***
A penalty $\mathcal{P}$ is a set of applicable punitive actions when guarantees on data are not satisfied, such as inform relevant authorities of the default or cancel the agreement.

Definition 8 ***(Privacy-Data Term)*** *A privacy-data term p_d is defined by a tuple $(\mathcal{T}^d, \mu, \mathcal{P})$ with $\mathcal{T}^d$ a set of guarantee terms, μ the privacy agreement validity, and $\mathcal{P}$ the set of penalties.*

Example 3 *let us assume a privacy agreement identified by PA_1, was signed at the date d_s and its validity period is $[d_s, t]$. The* ***Privacy-Data term*** *p_{ccn} for the credit card number data is:*
$p_{ccn}(t_{ccn}, PA_1, d_s, [d_s, t], penalty)$
where $penalty \in \mathcal{P}$ is an applicable penalty if the obligation "pay a fine" is not satisfied.
The ***privacy-guarantee term*** *t_{ccn} is defined as*
$t_{ccn}(r_{ccn}, o_{ccn}, ccn)$
$r_{ccn}(c, ccn, pay\ invoice, [d_s, d_{pay}])$ (right on ccn).
$o_{ccn}(sp, ccn, delete, [d\_pay, d\_del])$ (obligation on ccn).
This privacy-guarantee term specifies : once the credit card number ccn is used by a company c (third party) to pay the invoice in the time period $[d_s, d\_pay]$, the service provider sp must delete the credit card number at the date $[d\_pay, d\_del]$).

3.1.2 Privacy-Event Term

As an agreement can be carried during the period of validity, it is subject of evolution, because of emerging competitors, changes to laws or regulation, changing the web service business strategies, and so on. All potential events may happen during the agreement validity and are expressed in the Privacy-Event term part of the agreement. They might affect different elements defined in the privacy-data term. We studied and analyzed all possible events that can be occurred in the service behavior and triggering changes on the guarantees of privacy-data term. We denote by $\mathcal{E}$ a set of these events.

In Table 1 is depicted a set of *triggering events*. These events trigger a set of actions dictated by changes denoted by $\mathcal{AC}$. The actions will update the privacy data term.

Definition 9 ***(Event)***
An event type e is a tuple (e_{id}, cat, c_i, t_e) with e_{id} is the event identifier, $cat \in \mathcal{E}$, c_i is an information of the event, t_e denotes the reference time (a date) when the event e_{id} occurs.

Definition 10 ***(Privacy-Event term)***
A privacy-event term p_e is a couple (e, a) with $e \in \mathcal{E}$ and $a \in \mathcal{AC}$, where $\mathcal{E}$ is a set of event types and $\mathcal{AC}$ a set of actions dictated by changes (see table 1). $\mathcal{T}^e \subseteq \mathcal{E} \times 2^{\mathcal{AC}}$ a set of privacy-event term.

3.1.3 Agreement-Negotiation term

An agreement-negotiation term encloses a description of actions fired when an event occurs, including negotiation ac-

Table 1. types of events and example of actions dictated by changes

Events Triggering changes		**Actions dictated by Changes**	
Data-driven	1.add new personal data which becomes necessary at time *t* for a given transaction.	Create Data-Guarantee	1. Add new data-right with new data(with data-user,data retention interval,data usage) . 2. Add new Data-Obligation with new data(with data-user, running obligation date,data usage).
Purpose-driven	1. New purpose associate to data which becomes necessary at time *t* when this data being used or not.	Create Data-Right	1. Add new Data-Right with specific new data use (and add third party if new one).
Data user-driven	1. Add new third party which will help service provider to do particular work. 2. Change third party for any reasons.		1. Add Data-Right with new data user (with data,data retention interval,data use).
Duration-driven	1.Decrease or increase interval data retention during the validity of data retention period or after data retention expiration.	Update Data-Right	1. Update interval of data retention with new time period.
Security-Action-Driven	1. Change security on the data defined in data-obligation to avoid for instance new security threats.	Delete/ Update	1.Delete all data of a given data subject. 2. Delete partially data (e.g.delete only the ccn). 3. Replace data with an updated set of data (e.g. update subject's address).
		Hide/ Unhide	1. hide (encrypt) all data of a subject from any access. 2. hide a part of this data from any access. 3. unhide all data. 4. unhide a part of the data.
		Logs	1. take logs.

tions when a conflict arises.

In order to make the self-containing subsection, we shall introduce the following definitions needed in the agreement-negotiation term.

Definition 11 ***(Agreement-Level)***
The agreement level l is a state in which the agreement is after finishing the data guarantee monitoring by the system handling the agreement.

$$l \in \{unchanged, revised, conflict\}.$$

Remark : Due to the space limitation we can't represent the architecture of the system handling the privacy agreement.

Definition 12 ***(ActionScoope)***
The actionScoope as is an action to be taken regarding the level of the agreement. $as \in \{\mathcal{NA}, \perp, \mathcal{AC}\}$, with $\mathcal{NA}$ is the set of negotiation actions to be taken when a conflict happens in the agreement, then a negotiation protocol is fired, $\perp$ means no action is involved in, and $\mathcal{AC}$ is a set of actions dictated by changes, that is,
$Value(l) =' unchanged'$, then the actionscoope $as = \perp$ and no action is fired, where Value is a function giving the level of the agreement
$Value(l) =' revisited'$, then the actionscoope $as \in \mathcal{AC}$ is fired,
$Value(l) =' conflict'$, then actionscoope $as \in \mathcal{NA}$ is fired.

A set of changes on the terms defined in the privacy-data model are needed. To make an efficient negotiation, based on [10], we need (1) a set of *negotiation actions*, defining possible actions that each party might take on, (2) an *agreement-negotiation protocol*, enabling interaction mechanism between service provider and customer The next section is devoted to the negotiation protocol.
There are three types of actions involved in the negotiation: (1)*Agreement-Right*, it is an action that signing entity will achieve if he wishes to during the negotiation time. (2)*Agreement-Obligation*, it defines a set of duty actions that both service provider and customer must perform when a type of event e happens during the agreement life (3)*Agreement-Negotiation*, defines actions of the negotiation that can be taken by signing parties when conflicts occur between them. Conflict resolution is based on these actions by specifying how the terms of privacy data term can be modified or revised according to the execution circumstances.

Formally speaking, the agreement negotiation language can be defined using the following grammar:

$$\begin{array}{lcl}
Agree-negot-action & \rightarrow & \mathcal{AG}_r(Role, a_{id}, date, validity) \mid \\
 & & \mathcal{AG}_o(Role, a_{id}, date, validity) \mid \\
 & & \mathcal{AG}_n(Role, a_{id}, date, validity) \\
a_{id} & \rightarrow & Action_{Right} \mid Action_{Obligation} \mid \\
 & & Action_{Negotiation} \\
Action_{Right} & \rightarrow & reject \mid accept \\
Action_{Obligation} & \rightarrow & reply \mid notify \\
Action_{Negotiation} & \rightarrow & relate \mid proposal \mid justify \\
Role & \rightarrow & sp \mid cu
\end{array}$$

Definition 13 ***(Agreement-right)***
An agreement-right term $\mathcal{AG}_r$ is a tuple $(Role, a_{id}, d, \nu_r)$ where $Role$ specifies the behavior of entities which can be either service customer cu or provider sp, $a_{id} \in Ac^r$ identifying the type of actions, d denotes the reference time (a date) when the action-right is activated by a Role, and ν_r is a time interval validity of an agreement-right, with $d \in \nu_r$.

Example 4 *Once the service customer receives a privacy agreement proposition from service provider, the customer has the Right to* ***accept*** *or* ***reject*** *the proposition within 2 days after its receipt. This agreement-right is expressed as :* $\mathcal{AG}_r(cu, accept, d_{reply}, [d_{proposal}, d_{proposal} + 2])$ *or* $\mathcal{AG}_r(cu, reject, d_{reply}, [d_{proposal}, d_{proposal} + 2])$

Table 2. Example of Action types in the Agreement negotiation terms

Action	Meaning	Action type
Notify	Service provider **notifies** service customer that Event was happened at time point t_e.	agreement-obligation
Relate	Service provider **relates** which data in the agreement is affected by a change and send it as a report.	agreement-negotiation
Proposal	The provider **proposes** a proposition to the customer that contains revised privacy-agreement.	agreement-negotiation
Reply	Service customers must **reply** by sending an acknowledgment receipt of the proposition.	agreement-obligation
Reject	Service customer **rejects** the proposition.	agreement-right
Justify	Service customer **justifies** the refusal reply by some explanation including additional information about his decision.	agreement-negotiation
Accept	Service customer **accepts** proposition.	agreement-right

Definition 14 ***(Agreement-obligation)***
An agreement-obligation term $\mathcal{AG}_o$ is a tuple $(Role, a_{id}, d, \nu_o)$ with $Role \in \{cu, sp\}$, $a_{id} \in Ac^o$ an obligation action, where Ac^o is the set of these actions, d denotes the reference time (a date) when an action-obligation is activated by a Role, and ν_o is a time interval validity of an agreement-obligation, with $d \in \nu_o$.

Example 5 *Service provider must* ***notify*** *the customer within 5 days after the event happened (at t_e instant time). This agreement-obligation is expressed as :*
$\mathcal{AG}_o(sp, notify, d_{notify}, [t_e, t_e + 5])$

Definition 15 ***Agreement-negotiation***
An agreement-negotiation term $\mathcal{AG}_n$ is a tuple $(Role, a_{id}, d, \nu_n)$ with $Role \in \{cu, sp\}$, $a_{id} \in Ac^n$ is a negotiation action identifier, where Ac^n is the set of these actions, d denotes the reference time (a date) when a negotiation-action is activated by a Role, ν_n is a time interval validity of the negotiation-action, with $d \in \nu_n$.

Example 6 *The service provider sp* ***relates*** *which personal data in the agreement is affected by a change and send it as a report to the customer within 10 days after the occurrence of the event. This agreement negotiation is expressed by :*

$\mathcal{AG}_n(sp, relate, d_{send}, [t_e, t_e + 10])$

Table 2 summarizes and describes briefly the various actions with their types activated by the signing parties.

4 Privacy Agreement use

4.1 Private data use flow

In order to manage privacy data terms, we propose to express the private data use flow as state machine because of its formal semantic, well suited to describe the activation of different clauses of the privacy agreement. It will show *which* and *when* a clause is activated. The state machine will specify the states of each activated clause in the policy level. Figure 2 shows an example of the privacy data term activation for the purchase service provider. We have identified several abstractions in relation to private data flow, *private data use* abstractions and *authorization* abstractions. The first abstractions describe the different states in which the collected private data are used and who use them. The authorization abstractions provide the conditions that must be met for transitions to be fired.

States
we define three types of states:

- The initial state s_i represents the activation of the agreement where the first private data of the customer is collected.

- The intermediary states represent the flow of the collected private data use. By entering a new state, a private data is used (1) to complete the activity of the service for which it was provided (identified in figure 2 by $Op_{current}$), (2) or/and to achieve an extra activity (identified in figure 2 by $Op_{marketing}$), (3) or/and to activate an operation dealing with security when the time retention of the private data is elapsed (e.g. obligations), (4) or a misuse of the data (identified in figure 2 by $Op_{wrong-use}$).

- The final state s_f represents either the failure of the agreement i.e. the agreement is not respected due to the wrong use of the data, or end of the agreement where all the obligations related to the collected private data are finished.

Transitions
Transitions are labeled with conditions which must be met for the transition to be triggered. We have identified three kind of authorization abstractions :
- Activation conditions. We define two types of activation (i) an operation has the authorization to collect a private data to achieve the current aim of the service (ii) an operation dealing with an extra activity of the service has the authorization to be triggered .
- Temporal conditions. The transition is called *timed transition*. We define two types of timed transitions (i) an operation is finished within a time, a transition to another state is fired where the right attached to this finished operation will

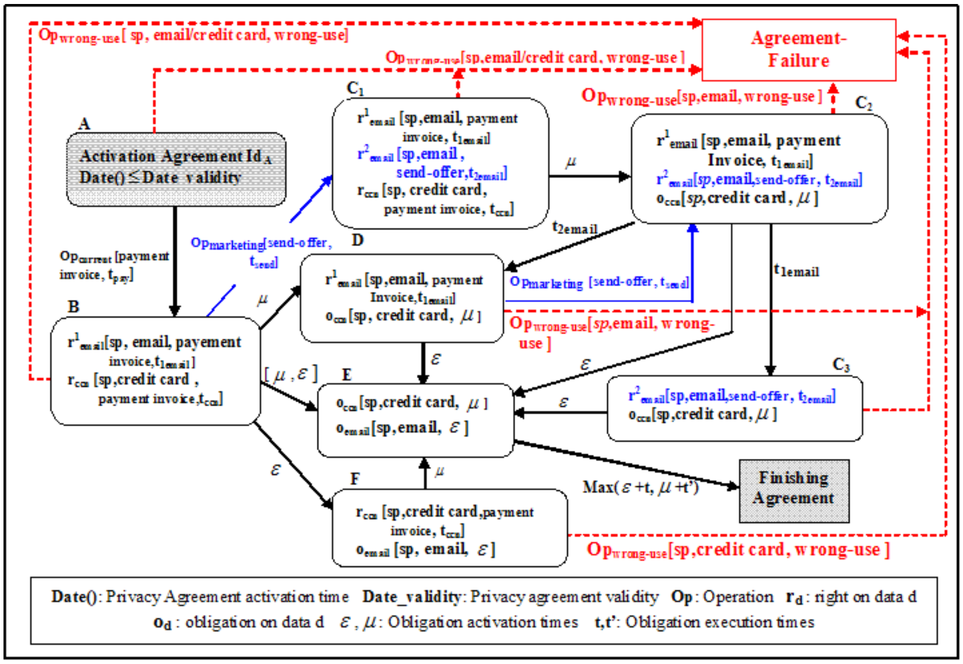
$Op_{wrong\text{-}use}$[sp, email/credit card, wrong-use]
$Op_{wrong\text{-}use}$[sp,email/credit card, wrong-use]
Agreement-Failure
$Op_{wrong\text{-}use}$[sp,email, wrong-use]
A
Activation Agreement Id_A
Date() ≤ Date validity
C_1
r^1_{email} [sp,email, payment invoice, t_{1email}]
r^2_{email} [sp,email , send-offer,t_{2email}]
r_{ccn} [sp, credit card, payment invoice, t_{ccn}]
μ
C_2
r^1_{email} [sp ,email, payment Invoice, t_{1email}]
r^2_{email}[sp,email,send-offer, t_{2email}]
o_{ccn}[sp,credit card, μ]
t_{2email}
t_{1email}
$Op_{current}$ [payment invoice, t_{pay}]
$Op_{marketing}$[send-offer, t_{send}]
D
r^1_{email} [sp,email, payment Invoice,t_{1email}]
o_{ccn} [sp, credit card, μ]
$Op_{marketing}$ [send-offer, t_{send}]
$Op_{wrong\text{-}use}$[sp,email, wrong-use]
B
r^1_{email}[sp, email, payement invoice,t_{1email}]
r_{ccn} [sp,credit card , payment invoice,t_{ccn}]
μ
[μ , ε]
ε
E
o_{ccn} [sp,credit card, μ]
o_{email}[sp,email, ε]
ε
ε
C_3
r^2_{email}[sp,email,send-offer, t_{2email}]
o_{ccn}[sp,credit card, μ]
ε
μ
F
r_{ccn} [sp,credit card,payment invoice, t_{ccn}]
o_{email} [sp, email, ε]
Max(ε +t, μ +t')
Finishing Agreement
$Op_{wrong\text{-}use}$[sp,credit card, wrong-use]
Date(): Privacy Agreement activation time   Date_validity: Privacy agreement validity   Op: Operation r_d: right on data d
o_d : obligation on data d ε , μ: Obligation activation times   t,t': Obligation execution times

Figure 2. Flow of Private data use

not be propagated to the new state (ii) the authorization to keep the private data is finished and the obligation is triggered.

• Misuse Conditions. An unauthorized operation uses the collected data.

The semantic of the state machine is to define all the triggered operations involving private data from the activation of the agreement (initial state) to the end of the agreement (final state).

4.2 Policy level change operations

To update the privacy agreement policy level, it is necessary to define a set of change operations that can be applied to the agreement policy level during the process. We define the set of the operations on the state machine.

• *AddState*: A new right r_n is activated in a new state s_n between two states s_p and s_s. It contains the tuple $(r_n, {r_i}^p, {o_j}^p)$ where ${r_i}^p, {o_j}^p$ represent the rights and the obligations of the previous-state between two states s_p and s_s.

•*RemoveState*: A right r_r and obligations o_r are removed from a state s_r then the previous state of s_r is attached to the successor of s_r defined without r_r.

•*UpdateState*: The elements of rights or obligations are changed in the state.

•*AddTransition*: A new operation Op_n or time conditions t_n are added and fired to a new state.

•*UpdateTransition*: Some elements of the operation or times conditions are updated which induces to update the next state.

•*RemoveTransition*: A transition t_r between two states s_p and s_s is removed (dealing only the activation conditions not temporal conditions), then s_s is removed, and all the states containing rights or obligations induced by firing t_r are removed. In order to maintain a consistency structure, a transition between s_p and the the first good successor of s_s is established.

4.3 Privacy-Agreement lifecycle

An agreement life-cycle is represented by an automaton, as depicted in Figure 3. It includes all states in which the agreement is. When an agreement is created, it does not entail, it is activated (e.g monitored), it remains in a *sleep* state until the service agreement is running, it becomes in an *activated* state. If there is no problem during the running process the agreement will be finished. When an event happens, the agreement is still activated but may be evolved, so it moves to *whipped up* state. The *checked* state is the core state, because the monitoring system is checking the service regarding privacy terms and privacy guarantees within the new data involved by the event. In this state the agreement has three levels (1) *unchanged*, no change is needed in the privacy data term (2) *conflict*, when a guarantee term is not satisfied, the service provider may start negotiation

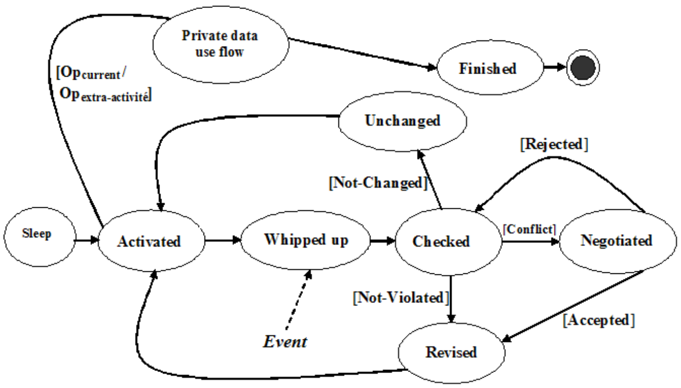

Figure 3. The Privacy Agreement lifecycle

with the costumer until the two parties find an issue. We will define the negotiation protocol later on (3) *revised* the new agreement proposal is accepted and the update should be activated. More details about the agreement level are in section 3.1.3.

5 Agreement Negotiation Protocol

In order to preserve or revise privacy agreement, a web service needs protocols that govern and structure interactions between signing parties. The features of the Agreement Negotiation Protocol $\mathcal{ANP}$ presented here includes a *negotiation language* defined previously, and an *interaction mechanism* that the parties must follow to come to an accord. Such mechanism is based on Rubinstein's Alternating Offers Protocol [12], where two parties A1 and A2 participate in the negotiation process and make offers and counteroffers. In our framework, we modify such model in order to assume that the protocol *is not an alternating offer* model, in the sense that the customer does not make any counter offer to the agreement proposal received from the provider. It is only the provider that makes an offer and waits for the acceptance or refusal of the customer. Also we assume that the players never opt out the negotiation during a time period of the negotiation μ_n that both parties must be defined in the agreement, otherwise the penalties will be fired.

The protocol $\mathcal{ANP}$

During the negotiation session each party uses suitable actions when communicating with each other. The service provider should *notify* service customer when an event $e \in \mathcal{E}$ happens at time point *t* and needs a negotiation in order to activate some actions $ac \in \mathcal{AC}$ updating the privacy agreement data term, then he suggests a privacy agreement proposition to the service customer that contains revised terms in privacy data term (*proposal*). The service customer must *reply* by sending decision about the received agreement privacy proposition. Service costumer has the right to *accept* or *reject* the proposition and in this case he must send some additional information about negative decision (*justify*). Such justification may help the provider to make a new proposal. Finally, the negotiation will end successfully otherwise if the time period of the negotiation is over, then the penalties are fired.

The parties can act in the negotiation only at discrete time point in the set $T = \{0, 1, 2, ...\}$. At each instant *t* ($t \neq 0$) in the negotiation, if the negotiation has not yet terminated, the service customer, whose turn is to respond, may send *accept* or *reject*. If a proposition made by service provider at time instant *t* is accepted by service customer then the negotiation terminates.

We express the bilateral protocol by a state machine (STM), where the states represent the different phases in which the negotiation of the provider (respectively the customer) is in during the interaction with a customer(respectively the provider). Transitions are triggering by messages sent by the customer to the provider or vice versa. Figure 4 shows a graphical representation of a protocol called $\mathcal{P}_1$ that describes the behavior of the negotiation involved by the service provider.

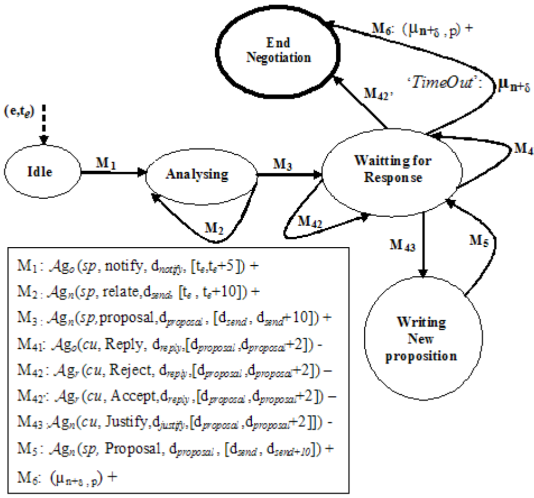
End Negotiation
M6: (μn+δ , p) +
'TimeOut': μn+δ
M42'
(e,te)
Idle
M1
Analysing
M3
Waitting for Response
M41
M2
M42
M43
M5
Writing New proposition
M1 : Ago(sp, notify, dnotify, [te,te+5]) +
M2 : Agn(sp, relate,dsend, [te , te+10]) +
M3 : Agn(sp,proposal,dproposal , [dsend , dsend+10]) +
M41: Ago(cu, Reply, dreply,[dproposal ,dproposal+2]) -
M42 : Agr (cu, Reject, dreply,[dproposal ,dproposal+2]) –
M42': Agr (cu, Accept,dreply ,[dproposal ,dproposal+2]) –
M43 :Agn(cu, Justify,djustify,[dproposal ,dproposal+2]]) -
M5 : Agn(sp, Proposal, dproposal , [dsend , dsend+10]) +
M6: (μn+δ , p) +

Figure 4. ($\mathcal{P}_1$): Provider's Negotiation Protocol

Definition 16 ***(Agreement Negotiation Protocol)*** *Formally an agreement negotiation protocol is a tuple $\mathcal{ANP} = (\mathcal{S}, s_0, \mathcal{F}, \mathcal{M}, \Delta, \mu_n)$, which consists of following elements :*

- *$\mathcal{S}$ is a nonempty set of states*
- *s_0 is the initial state $s_0 \in \mathcal{S}$*
- *$\mathcal{F} \subset \mathcal{S}$ is the set of final states (end or penalties)*
- *$\mathcal{M}$ is a finite set of messages. For each message $m \in \mathcal{M}$, a polarity is defined which will be positive(+) if m is an*

outgoing message in $\mathcal{ANP}$ and negative(-) if m is an incoming message in $\mathcal{ANP}$. In the sequel, we use $m()^+$ (respectively $m()^-$) to denote the outgoing (respectively incoming)

- $\Delta \subseteq \mathcal{S} \times S \times \mathcal{M}$ *a finite set of transitions*
- μ_n *the negotiation time interval over which the penalties are activated. This interval is defined when the agreement is signed*

6 Related Work

In the recent Web services research area, there are increasing demands and discussions about privacy technologies to support different business applications. Relevant works in the area of privacy management are described in [4]. An obligation management model is defined in [9]. However, they are all related to enterprise. A work has been done to deal with policy management, including obligations such as [3]. This paper formalizes the obligations and investigates mechanisms for monitoring obligations. It deals with the access control area. The work in [14] presented an approach for preserving privacy in government web services. The approach is based on digital privacy credentials, data filters, and mobile privacy enforcement agents. Individual privacy contracts are proposed in [11]. The aim of this work is to present the principles and a conceptual view of the management of privacy contracts in relational database systems. An algorithm has been developed to guide the implementation of privacy contracts but this algorithm is not adapted to implement privacy contracts when developing web services applications. Relevant works in the area of privacy negotiation are described in [6, 15]. No evolution of the privacy policy is taking into account.

7 Conclusion

In this paper, we proposed a privacy agreement model, that both service customer and provider might agree before any process is run, and a framework to show how and when the privacy agreement is activated, and a flexible agreement-negotiation protocol enabling negotiation of a bilateral interaction mechanism between the parties. The latter should preserve privacy-agreement and avoid conflicts between the parties when events happen during the running process, leading a change in the web service privacy agreement. The framework supports the life-cycle management of the privacy agreement. A promising area for the future work includes refining the approach and introducing a reasoning mechanism for the temporal aspect about agreement that may change over the time in the agreement negotiation protocol.

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Implementing p3p using database technology. *International Conference on Data Engineering (ICDE'03)*, 00:595, 2003.

[2] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. N. J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification (ws-agreement). In *Technical report*. Grid Resource allocation AgreementProtocol (GRAAP) WG, Sept. 2006.

[3] C. Bettini, S. Jajodia, X. Wang, and D. Wijesekera. Obligation monitoring in policy management. In *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, page 2, Washington, DC, USA, 2002. IEEE Computer Society.

[4] I. Corporation. Enterprise privacy authorization language (epal). In *IBM Research Report*, 2004.

[5] L. Cranor, G. Hogben, M.Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, and M. Schunter. The platform for privacy preference 1.1(P3P 1.1) specification. In *Technical report*. W3C Working Draft, July 2005.

[6] K. El-Khatib. A privacy negotiation protocol for web services. In *Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments Halifax*, Nova Scotia, Canada, Oct. 2003.

[7] Y. Li and S. Benbernou. Representing and reasoning about privacy abstractions. In *6th International Conference on Web Information Systems Engineering,WISE 2005.*, pages 390–403, 2005.

[8] Y. Li, S. Benbernou, H. Paik, and B. Benatallah. Formal consistency verification between bpel process and privacy policy. In *Privacy Security Trust PST'2006.*, pages 212–224, Oct. 2006.

[9] M. Mont. Towards scalable management of privacy obligations in enterprises. In *3rd International Conference on Trust, Privacy Security in Digital Business TrustBus'2006*, pages 1–10, 2006.

[10] A. Ncho and E. Aimeur. Building a multi-agent system for automatic negotiation in web service applications. *Autonomous Agents and Multiagent Systems (AAMAS'04)*, 03:1466–1467, 2004.

[11] H. Oberholzer and M. S. Olivier. Privacy contracts as an extension of privacy policies. *International Conference on Data Engineering Workshops (ICDEW'05)*, 0:1192, 2005.

[12] M. Osborne and A. Rubinstein. *Bargaining and markets*. The Academic Press, 1990.

[13] S. Paurobaly and N. R. Jennings. Protocol engineering for web service conversations. *Engineering Applications of Artificial Intelligence, Special Issue on Agent-oriented Software Development*, 18(2):237–254, 2005.

[14] A. Rezgui, M. Ouzzani, A. Bouguettaya, and B. Medjahed. Preserving privacy in web services. In *WIDM '02, Proceedings of the 4th international workshop on Web information and data management*, pages 56–62, New York, NY, USA, 2002. ACM Press.

[15] G. Yee and L. Korba. Bilateral e-services negotiation under uncertainty. In *International Symposium on Applications and theInternet(SAINT2003), Orlando, Florida, Jan. 27-31*, 2003.