CD-JRA-1.3.3

Benbernou, S., Meziane, H.:

A Dynamic Privacy Model for Web Services

In: ***Computer Standards and Interfaces.*** **(2009)**

A Dynamic Privacy Model for Web Services

Salima Benbernou

Université de Lyon, Université Claude Bernard Lyon 1, LIRIS CNRS UMR 5205, France

Hassina Meziane

Université de Lyon, Université Claude Bernard Lyon 1, LIRIS CNRS UMR 5205, France

Abstract

Nowadays, Web services are being recognized as an emerging platform to quickly develop complex distributed applications. Many services (e.g., mortgage approval, travel agency) require service requestors to disclose some personal data (e.g., credit card number, home address). As the number of inappropriate usage and leakage of personal data is increasing, privacy concerns is becoming one of most important concerns of service requesters, service providers and legislators. 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 the dynamic environment of the policies. In this paper, we introduce a framework based on an agreement as a solution to these problems. We propose a *privacy agreement* model that spells out a set of requirements related to consumer's privacy rights in terms of how service provider must handle privacy information. We define two levels in the agreement (1) policy level (2) negotiation level. A formal privacy model is described in the policy level to provide upon it a reasoning mechanism for the evolution. 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. Hence, the privacy evolution is handled in this level. A negotiation protocol is proposed to enable ongoing privacy negotiation to be translated into a new privacy agreement.

Key words: `WS-agreement, privacy-agreement, evolution, negotiation, policy.`

Email addresses: `sbenbern at liris.univ-lyon1.fr` (Salima Benbernou), `meziane_has at yahoo.fr` (Hassina Meziane).

1 Introduction

Semantic web services is a paradigm aiming at building upon semantic Web and Web services technologies to enable automatic discovery, access, combination, and management of web services [1]. Web services allow software entities to be described, advertised, discovered and accessed through XML based standard language. Over the past years, there has been a widespread increase in the use of semantic web-based services. Because of the increasing popularity of web services, a number of pressing issues should be resolved, especially the issues handling consumers' personal identifiable information (PII). Most of the time, web-based service providers require some personal information or financial information from their consumers. Such information may be used for a number of aims; ranging from regulation access to their on line services (authentication, authorization) to billing (accounting), to service maintenance and so on.
Nowadays, the individuals are becoming more and more concerned about the privacy of their personal data [2–6]. These concerns might lead to a situation where the customers do not trust the web service any more and take their business somewhere else [7]. So, the important enabling factor for a well usage of online services is building customer's 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. The most significant effort currently underway to enable web site users to gain control over their private information, is the Platform for Privacy Preferences (P3P), developed by the World Wide Web Consortium (W3C).

However, privacy policies alone defined in P3P 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*-. However, Current WS-Agreement [8] which is an XML-based language for creating contracts, agreements and guarantees from offers between a service provider and a client is not sufficient to handle the privacy environment, so an extension of WS-agreement is unavoidable. 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 privacy management in Web services. It is part of ServiceMosaic [1] project which is a CASE-toolset model-driven prototype platform for modeling, analyzing, and managing web service models including business protocols, orchestration and adapters. The main goals of the project are (1) to provide the definition of service description framework that distills current standards into a useful and usable service description model, endowed with richer abstractions and a formal semantics (2) the development of a fully-fledged CASE tool for Web service development and lifecycle management that builds on existing Web services infrastructure and augments it with automated and model-driven service management capabilities. The privacy policy model is defined as an agreement and supports the 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* an extension of WS-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 agreement is called *policy level*.
- While *access control* aspect of security and privacy is well understood, it is unclear of how to do *usage control*. Hence, a private data use flow model is described as a state machine in the policy 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 the privacy agreement. An *agreement-evolution* model is provided in the privacy-agreement. This part of the agreement is called *negotiation level*.
- An *agreement-negotiation protocol* is provided to build flexible interactions and conversations between parties when a conflict happens due to the events occurring in the dynamic environment of the Web service.
- We devise an architecture supporting the privacy agreement and some evaluation results are also provided. It is one component of the ServiceMosaic platform.

The remainder of the paper is structured as follows. Section 2 presents a short overview of WS-Agreement, on which our framework is built by abstracting privacy. Section 3 provides a formal model for data privacy in web services. Section 4 proposes an extension of WS-Agreement by taking into account

[1] http://servicemosaic.isima.fr

the previous model of data privacy as a privacy agreement and the evolution part of the privacy. Section 5 presents a model of the private data use flow and a management of the private data use flow is discussed with a set of operations involved for the model in order to take into account the privacy evolution. Section 6 discusses the privacy agreement negotiation protocol in order to take into account evolution in the privacy agreement. We devise an architecture for the management of the privacy agreement in section 7. The evaluation results are also discussed. We discuss a related work in the next Section.

2 WS-Agreement

WS-Agreement [8] specifies an XML-based language for creating contracts, agreements and guarantees from offers between a service provider and a client. In this case, an agreement may involve multiple services and include fields for the parties, references to prior agreements, service definitions and guarantee terms. Here, the service definition is part of the terms of the agreement and is established prior to the agreement creation. In more detail, an agreement is defined as being composed of:

(1) Name identifies the agreement and is used for reference in other agreements
(2) Context includes parties to an agreement, reference to the service provided and to possibly other related or prior agreements.
(3) Service Description Terms provide information to instantiate or identify a service to which the agreement pertains.
(4) Guarantee Terms specify the service levels that the parties are agreeing to and may be used to monitor and enforce the agreement. They consist of : 1) the list of services it applies to, 2) the list of variables representing domain-specific concepts (e.g. response time or bandwidth), 3) optional conditions that have to be met for the guarantee to be enforced, 4) conditions to satisfy the guarantee and 5) one or more business values (e.g. the penalty upon failure to meet the objective, the strength of a commitment by a service provider or the importance and confidence of meeting an objective).

An agreement template follows the above structure. A service provider offers an agreement template describing the service and its guarantees. Negotiation, then, involves a service consumer retrieving the template of agreement for a particular service from the provider and filling in the appropriate fields. The filled template is then sent as an offer to the provider. The provider decides whether to accept or reject the offerer, depending on its resources. Although offers and agreements have mostly the same fields, an offer contains choices

for an agreement from the service customer for the service provision. In an agreement, the choices in an offer are modified by the service provider to finalise the agreement.

2.0.1 ***Weaknesses of WS-Agreement***

There are significant shortcomings [9,10] in WS-Agreement.

Limited Message Types . The first significant weakness lies in the fact that messages in WS-Agreement are limited to two types : offer and agree, according to a template published by a service provider. The WS-Agreement specification is only used at the last stage in a transaction where the parties are closing their interaction with a contract specified as a WS-Agreement.

No Interaction Protocols. WS-Agreement suffers from the lack of an interaction protocol specified between parties. This is the second significant weakness. There is only a two step conversation, an offer followed by an agree. Without an adequate set of speech-acts [10] and specification of how to construct interaction protocols, the usefulness of a WS-Agreement exchange is limited to cases such as buying from catalogues, with take-it or leave-it offers from the seller or buyer. Even if the schema of the WS-Agreement is increasing with various speechacts, there is no concept of how to sequence messages to form a valid conversation.

Lack of Semantics. On the whole, WS-Agreement is a complex specification, with vague and unclear semantics. Significant work is required in clarifying the interfaces before it is successful in enabling web services interactions. Furthermore, the WS-Agreement specification only defines a higher-level template for agreements and offers. There is a need of a language to express the elements in the Service Description Terms and Guarantee Terms. Thus, there is no indication of how to access or provide a service from an agreement.

3 Privacy data Model

There are some guidelines for protecting personal data. For instance, OCED [11] defined eight principles to protect personal data while pursuing free information flow between different organizations which probably access countries. One can cite among these eight principles the collection limitation, data quality, purpose specification, use limitation and security safeguards. Based on this general privacy principles, on some exiting works [12] and on our previous works [13–16], as well as taking into account the characteristics of Web ser-

vices, we model privacy of Web service by identifying and describing relevant abstractions.

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.

Let us illustrate the motivations through the following example dealing with a purchase service where the transactions between the customer and the service is not considered in the paper. Let us assume that the privacy policy of the service provider accepted by the customer is defined as follows: the service has the authorization to collect email address (email) and credit card number (ccn) to complete its activity for the current purpose i.e. the email is used to send invoices and Credit card number for the payment of the invoices. Furthermore, the service provider can also use email address to achieve an extra activity for instance marketing purpose i.e. the email is used to send the available products and their prices.
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

(1) $r^1_{email}(sp, email, send\ Invoice, \mu_{r1email})$,
specifies that the service provider sp has the right to use email for sending invoices during the period $\mu_{r1email}$.

(2) $r^2_{email}(sp, email, send\ Offer, [d_s, d_s + 1\ month])$,
specifies that the service provider sp has also the right to use email for sending the available products and their prices during the period $\mu_{r2email}$

which is 1 month after both sides have signed the agreement at d_s date.

(3) $r_{ccn}(sp, ccn, payment\ Invoice, \mu_{rccn})$,
specifies that the service provider sp has the right to use ccn for the payment of the invoices during the period μ_{rccn}.

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

(1) $o_{ccn}(sp, ccn, crypt, d_{pay} + 1\ day])$,
specifies that the service provider sp must crypt the ccn for a given data subject at the end of each payment process, for instance, at d_{pay}+1 day (μ_{occn}).

(2) $o_{email}(sp, email, hide, \mu_{oemail})$
specifies that the service provider sp must hide the email for a given data subject at μ_{oemail} i.e. when the authorization of email retention time is elapsed.

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 constraints and their evolution in the behavior of the service.

4 Extended WS-Agreement structure

The ideal case is to have web service providers (1) offering and meeting the guarantees related to the services they develop, (2) offering privacy guarantees to protect the sensitive private information of the users. Both service provider and service customer must sign an agreement having some requirements.

4.1 Requirements

Current WS-Agreement specifications do not support the privacy structure and do not include the possibility to update the agreement at runtime. In fact, a guarantee is not fulfilled because of an event occurring in the service behavior and may change the personal data use.
At agreement creation time, the consumer has to be aware of the changes on

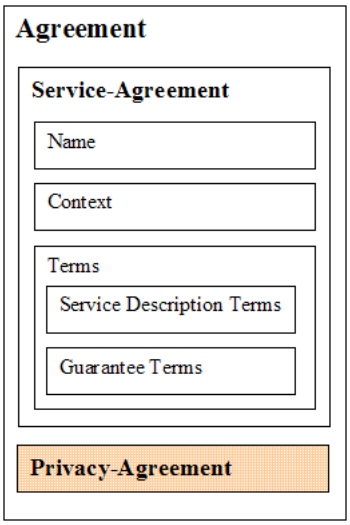

Fig. 1. Extended WS-Agreement structure

the use of its personal data in the service provider behavior. To allow a renegotiation at runtime, it is necessary to add some elements to an agreement specifying how it can be revised according to occurring events. For this purpose, WS-Agreement is extended in its syntax by adding appropriate terms (1) to specify and to express rights and obligations of data privacy (2) to define different types of events that may occur in the behavior of the system, and to define possible negotiation terms of the requirements by means a flexible negotiation protocol of the privacy-agreement when conflicts occur between parties. The protocol will use *negotiation language* based on the negotiation terms. The proposed extension is reflected in a new component in a WS-Agreement called ***Privacy-agreement***. In the rest of the paper, agreement means privacy agreement.

4.2 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 3.
(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 be taken if the guarantee of privacy terms is not respected and a conflict arises. They are used through a negotiation protocol between the service provider and the customer.

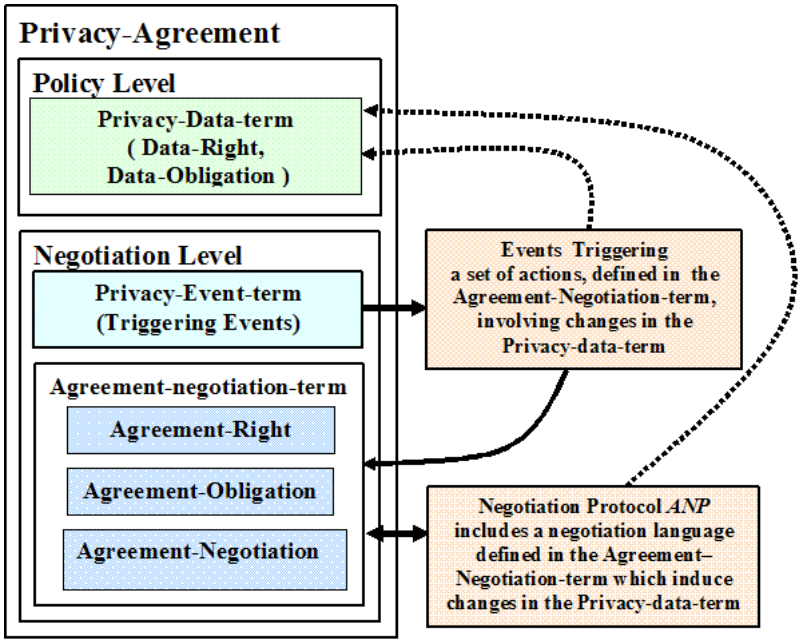

Fig. 2. The Privacy Agreement structure

4.2.1 ***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 3 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 obligations 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 used 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 $\mathcal{P} = \mathcal{P}_{G^d} \cup \mathcal{P}_n$ is a set of applicable punitive actions when guarantees on data ($\mathcal{P}_{G^d}$) are not satisfied or when negotiation process ($\mathcal{P}_n$) terminates without success. (more details about the agreement negotiation are in section 6).

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).*
$o_{ccn}(sp, ccn, crypt, [d\_pay, d\_crypt])$ *(obligation on ccn).*

This privacy-guarantee term says : 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 crypt the credit card number at the date $[d\_pay, d\_crypt]$).

4.2.2 ***Privacy-Event Term***

As an agreement can be carried during the period of validity, it is subject to evolution because of emerging competitors, changes to laws or regulations, 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 occur in the service behavior triggering changes on the guarantees of privacy-data term. We denote by $\mathcal{E}$ the set of these events:

(1) *data-driven*, adding new personal data.
(2) *purpose-driven*, some changes will affect data use on the personal data.
(3) *Duration-driven*, the time retention of personal data may be changed.
(4) *data-user-driven*, a new user will use the personal data.
(5) *security-action-driven*, to avoid new security threats, some new security actions on the personal data are needed.

Table 1 depicts 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.

Table 1
Types of events and examples 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 a 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 a new Data-Right with specific new data use (and add third party if new one).
Data user-driven	1. Add a new third party which will help service provider to do particular work. 2. Change the third party for some 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.

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.*

4.2.3 ***Agreement-Negotiation term***

An agreement-negotiation term encloses a description of actions triggered when an event occurs, including negotiation actions 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.(see the Event handler and Data guarantee controller components in the architecture in section 7.1).

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

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.

In order to preserve bi-laterally binding privacy-agreement, signing parties are willing to interact and negotiate between them when an event occurs, a conflict raises because a set of changes on the terms defined in the privacy-data model is needed. To make an efficient negotiation, based on [17], 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 by means of the previous set of actions. It would define the syntax as well as the semantic of the message exchanged including actions and data. During the negotiation session, each party can use a set of messages when communicating with each other.
The next section is devoted to the negotiation protocol.
Let us define the language of communication which belongs to the dynamic part of the privacy agreement. There are three types of actions involved in the negotiation:

(1) *Agreement-Right*, is an action that the signing entity will achieve if he wishes during the negotiation time.
(2) *Agreement-Obligation*, 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 $d \in \nu_r$.

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 sends 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 explanations including additional information about his decision.	agreement-negotiation
Accept	Service customer **accepts** proposition.	agreement-right

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*
$\mathcal{AG}_r(cu, reject, d_{reply}, [d_{proposal}, d_{proposal} + 2])$

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 sends 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.

5 Privacy Agreement use

In this section we will provide a reasoning mechanism on the privacy agreement. We first define a model to represent the flow of the private data use regarding the clauses in the agreement and how to reason on it by setting up a set of operations. The lifecycle of the agreement will be discussed later on.

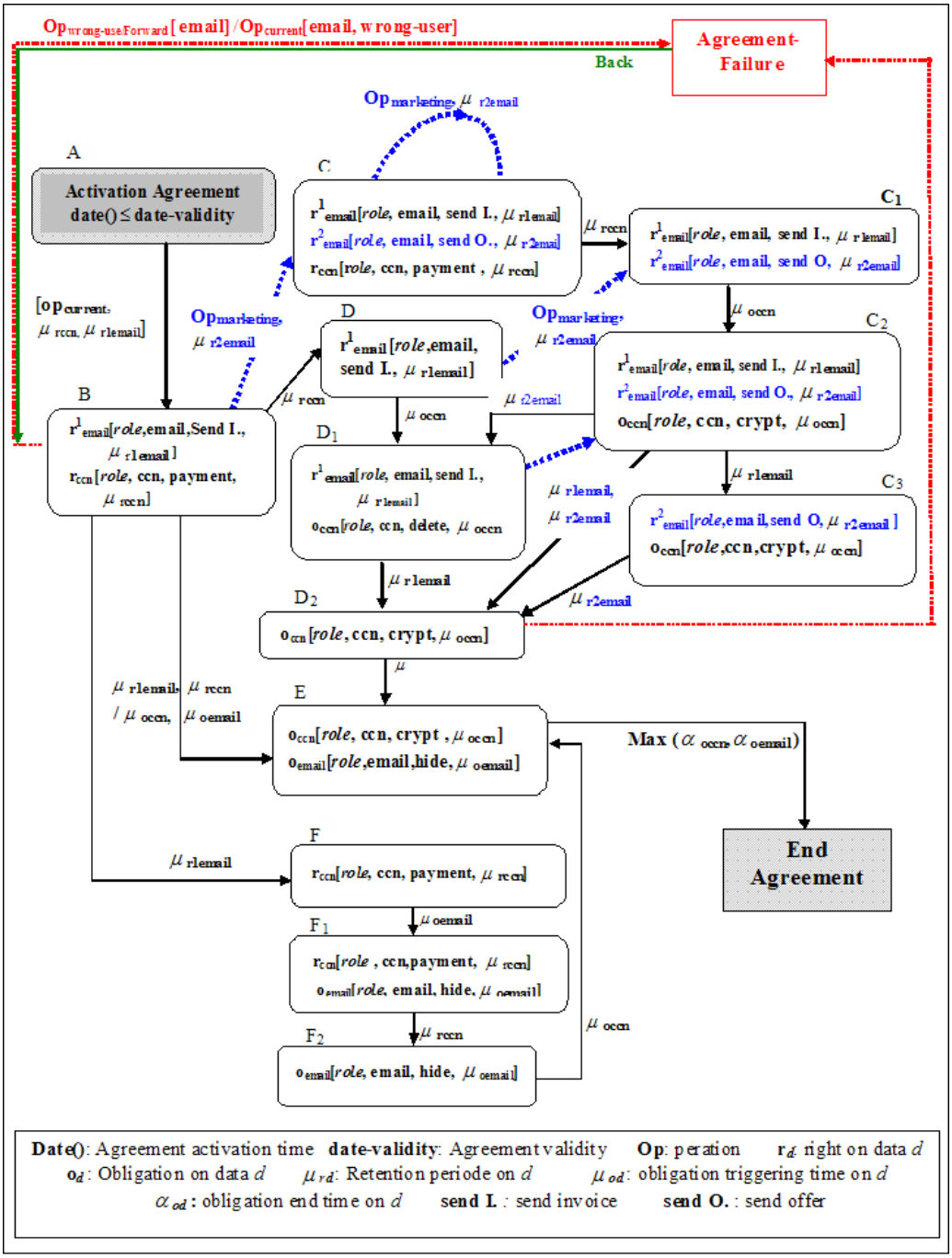

Fig. 3. Flow of Private data use

5.1 Private data use flow

In order to address the issue of privacy data term changes, we propose to express the private data use flow as state machine because of its formal semantic, and because it is well-suited to describing the activation of different clauses of the privacy agreement. It will present *which* and *when* a clause is activated

and specify the states of each activation clause in the policy level.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). Figure 3 shows an example of the privacy data term activation for a purchase service provider.
We have identified several abstractions in relation to private data use flow, *private data use* abstractions and *authorization* abstractions. The first abstractions describe the different states in which the agreement is -which private data is collected and when it is used and for what and who use it- . The authorization abstractions provide the conditions that must be met for transitions to be fired.

Definition 16 *(Private Data Use Flow.) A private data use flow $\mathcal{F}$ is a tuple*

$$\mathcal{F} =\prec \mathcal{S}, \mathcal{T}, \mathcal{C}, \Psi, \rho, \Phi \succ$$

where
• *$\mathcal{S}$ is the set of states including the initial state s_i, the final state s_f and the failed state s_{fail},*
• *$\mathcal{T}$ is the set of transitions of $\mathcal{F}$,*
• *$\mathcal{C}$ is a set of clauses containing a set of rights and obligations $\mathcal{C} \subset \{\mathcal{R}^{d_i} \bigcup \mathcal{O}^{d_j}, d_i, d_j \in \mathcal{D}\}$,*
• *Ψ is the transition assignment function, associating each transition to a source and a target states, $\Psi : \mathcal{T} \rightarrow \mathcal{S} \times \mathcal{S}$*
• *ρ is the operations and the elapsed time assignment function, associating a set of operations and elapsed time from the obligations and the rights to a set of transitions, $\rho : \mathcal{C}.r.op \cup \mathcal{C}.r.\mu_r \cup \mathcal{C}.o.\mu_o \rightarrow \mathcal{T}$.*
• *Φ is the rights and the obligations assignment function, associating a set of rights and obligations to a set of states,$\Phi : \mathcal{C} \rightarrow \sigma(\mathcal{S})$, where sigma denotes the powerset. The same rights and the obligations can be associated to several states which means propagated.*

Definition 17 *(Covered agreement.) Let $\mathcal{PT} = s_i, s_1, s_2, ...s_j$ a path from the initial state s_i to s_j, where $s_i, s_j \in \mathcal{S}$ and $s_k \in \mathcal{S}, \forall k \geq 1$. $\mathcal{PT}$ is called a partially covered agreement iff $\forall s_k \in \mathcal{PT}$ and $s_k \neq s_{fail}$, and $\mathcal{PT}$ is called a covered agreement if $s_j = s_f$.*

Let's define the semantic of privacy data use flow through the following example for the agreement with a set of clauses (rights and obligations).

Example 7 *Let us consider the example of a purchase service introduced in section 3. An agreement has been signed between customer and service setting up a set of clauses with a validity period denoted by validity-date. Those clauses are specified as follows: at the date date() the agreement is activated and the service collects email address (email) and Credit card number (ccn).*

Those private data are used for two types of operations ***(1)*** *to complete the service activity for the current purpose i.e. the email is used to send invoices and Credit card number for the payment of invoices. The operation are expressed by the following rights* $r^1_{email}(role, email, send\ invoice, \mu_{r1email})$ *and* $r_{ccn}(role, ccn, payment\ invoice, \mu_{rccn})$ ***(2)*** *to achieve other activities than those for which they are provided, for instance marketing purpose i.e. the email is used to send the available products and their prices, that clause is expressed by the right* $r^2_{email}(role, email, send\ offer, \mu_{r2email})$*. When the retention times of the private data email and ccn* $(\mu_{r1email}, \mu_{r2email}, \mu_{rccn})$ *are elapsed, the corresponding obligations are triggered,* $O_{email}(role, email, hide, \mu_{oemail})$ *and* $O_{ccn}(role, ccn, crypt, \mu_{occn})$*. Those obligations specifying the role must hide (respectively crypt) as soon as the activation date* μ_{oemail} *(respectively* μ_{occn}*) is reached.*

In what follow, we will comment the state machine, and for the sake of clarity, we omit some details about it.

States

States in the model represent the status of agreement terms, it can be *activated*, *finished* (the privacy terms are respected) or *violated*. By entering a new state, the monitoring system can observe which resources are captured in the private data use flow. These resources are identified by: the operations of the services using the private data and the role i.e. people or the process using it and the temporal resources i.e. the time of the operation activation, operation end or operation violation. Instead of assigning all these resources to a state, we use the rights and the obligations abstractions defined earlier. Rights and obligations can be cumulative, which means previously activated rights and obligations can not be deactivated (e.g. finished) when entering a new state. This is due to the fact that the rights and the obligations depend closely on time. While a right r_1 is still active, a new right r_2 can be activated and then the agreement policy is entering in a new state with r_2 and keeping r_1.

We define four types of states:

- The initial state s_i represents the activation of the agreement where the first private data of the customer is collected. In Figure 3, s_i is defined by A.
- The intermediary states represent the flow of the collected private data use. By entering a new state, a private data is used :
  - to complete the activity of the service for which it was provided, identified in Figure 3 by $Op_{current}$. In the state B, the current operations are *SendInvoice* and *payment*.
  - and/or to achieve an extra activity as depicted in Figure 3 by $Op_{marketing}$. The right r^2_{email} is activated in the state C as soon as the marketing oper-

ation is triggered. The same operation can be activated as many times as the data time retention μ_{remail} is valid. It is represented by a *loop* in the state C. The privacy agreement remains in the same state.

- and the data use is finished (the right). For instance, the agreement will be in the state C_1 since the retention time of private data *ccn* is elapsed μ_{rccn}.
- and/or to activate an operation dealing with the security (e.g. obligations) when the retention time of the private data defined as a fixed time in the right is elapsed and the time for triggering the obligations starts. For instance, such case is depicted in Figure 3 in the state C_2, where o_{ccn} is activated when the usage time of the date μ_{rccn} is elapsed and the obligation time starts defined in the transition by μ_{occn}.

- The *virtual* state labeled *Agreement Failure* will be reached when a private data is used to achieve the operation misuse and/or role misuse,for instance, the fist misuse is identified by $Op_{wrong-use/Forward}[email]$ between state B and Agreement Failure state. We call this state as a virtual state because it is considered only like a flag of misuses.
- The final state s_f represents the end of the agreement where for all collected private data the security operations (obligations) are finished.

The details of the agreement are provided in Appendix A. A part of the privacy agreement use is depicted in Figure 3. For instance, in the state C_1 three clauses of privacy agreement policy level are triggered (1) the *current* operation for two private data (r^1_{email}, r_{ccn}) which is payment invoice, is still activated by the service provider to achieve the service aim. The rights are cumulated from the previous state because the retention times of the rights r^1_{email} and r_{ccn} associated to the private data are not elapsed (2) the *send-offer* operation (r^2_{email}) is activated by entering C_1 for marketing purpose of the service (not to complete the service), it is an extra activity of the service.
In the state C_2 three clauses of the privacy agreement policy level are triggered (1) the current operation (r^1_{email}) is still activated and then cumulated from the previous state C_1 (2) the extra activity in r^2_{email} is still activated and then cumulated in the new state from C_1 (3) the action of security is triggered (o_{ccn}) because the time of data retention is elapsed (μ_{rccn}).
In the state E two clauses are triggered (1) the obligation o_{ccn} is still activated and cumulated from the previous state D_2 (2) the obligation o_{email} is activated because the time μ_{oemail} to activate is reached.

Transitions

Transitions are labeled with conditions which must be met for the transition to be triggered. The authorization abstractions are assigned to transitions. We have identified three kinds of authorization abstractions (transition con-

ditions):

- Activation conditions. We define two types of activation (i) an operation has the authorization to collect private data to achieve the current aim of the service, for instance, $op_{current}$ condition on the transition from the state A to the state B, an operation dealing with an extra activity of the service has the authorization to be triggered. For instance, the operation $op_{marketing}$ from the state B to the state C.
- Temporal conditions. The transition is called *timed transition*. We define tree types of timed transitions (1) an operation is finished within a time, a transition to another state is fired where the right dealing with this finished operation is then removed from the previous state in the new state. For instance from the state C to state C_1 the transition is labeled μ_{rccn}, which means the ccn use is over (2) the authorization to keep the private data is finished and the obligation is triggered. For instance from the state C_1 to C_2, the transition is labeled μ_{occn}, the operation of security must be fired (3) *Obligation end time* α_o, the obligation is over, for instance from the state E to the end-agreement state, we calculate the maximum of the two end times α_{oemail} and α_{occn}.In our case, it is the best way to finish the activation of the agreement.
- Misuse conditions. It exists a set of shapes of misuse such as an unauthorized operation on the collected data or an unauthorized time, for instance the retention time of the right is elapsed and a right is still activated etc. In This paper we will not discuss this set of misuses.

5.2 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. To be well constructed, these operations also called primitives of the state machine need to satisfy and ensure some properties.
To define these properties and primitives, let's denote by $\mathcal{F}^p$ the previous state machine and $\mathcal{F}^n$ the new one after the update.

Lemma 1 *(1) The partially covered agreement needs to be still satisfied after inserting a state. We can not map a right r_s to the state s_s while the associated obligation on data is finished (property $\mathcal{P}^1(r_s)$).*
$\mathcal{F}^n.\Phi = \mathcal{F}^p.\Phi \cup \{r_s \rightarrow s_s\} \Leftrightarrow \nexists s_s \in \mathcal{PT} \mid [\mathcal{F}^p, s_s] \models o_s$ *where* $(r_s, o_s) \in \mathcal{G}^d$ *and* $\mathcal{PT}$ *is partially covered.*

(2) If $\mathcal{PT}$ is a covered agreement, then it is not possible to map a right r_s to a new state s_s (i.e. add any states). We may start a new agreement with a validity date.
$if\ \exists\ s_f \in \mathcal{PT} \Rightarrow \nexists s_s \in \mathcal{S} \mid \mathcal{F}^n.\Phi = \mathcal{F}^p.\Phi \cup \{r_s \rightarrow s_s\}$

Lemma 2 *The partially covered agreement needs to be still satisfied after inserting a transition.The insertion of a transition is correct if the outgoing state of the transition is not the fail state s_f (property $\mathcal{P}^2(t)$.*
$\mathcal{F}^n.\Psi = \mathcal{F}^p.\Psi \cup \{t \rightarrow (s_p, s_s)\} \Leftrightarrow s_s \in \mathcal{PT},\ s_s \neq s_{fail},\ \mathcal{PT}$ *is partially covered.*

•**AddTransition**(t, s_p, s_s, at): A new operation Op_n associated to a right r_n or time conditions t_n associated to the right r_n or obligations o_n are fired and a new state is added. A time condition *timeout* is fired when a right and obligation are removed from the policy level (see RemoveAddState primitives). The property $\mathcal{P}^2$ must be satisfied.

$$[[AddTransition(t, s_p, s_s, at)]] = \begin{cases} s_s, s_p \in \mathcal{F}^P.S \text{ and } t \notin \mathcal{F}^P.\mathcal{T} \\ \mathcal{F}^n.\mathcal{T} = \mathcal{F}^p.\mathcal{T} \cup \{t\} \\ \models \mathcal{P}^2(t) \\ \mathcal{F}^n.\Psi = \mathcal{F}^p.\Psi \cup \{t \rightarrow (s_p, s_s)\} \\ \mathcal{F}^n.\rho = \mathcal{F}^p.\rho \cup \{\{at \rightarrow t\}\} where \\ at \in \{r.op, o.\mu_o, r.\mu_r, timeout\} \end{cases}$$

• **AddState**(s_s, s_p, t): A new right r_s is activated in a new state s_s which is the successor of s_p. It contains the tuple (r_s, r^p, o^p) where r^p, o^p represent the rights and the obligations of the previous-state s_p and are propagated to the state s_s . A transition t is added from the state s_p to the state s_s. The property $\mathcal{P}^1$ must be satisfied. The semantic of the operation:

$$[[AddState(s_s, s_p, t)]] = \begin{cases} s_s \notin \mathcal{F}^P.S \text{ and } t \notin \mathcal{F}^P.\mathcal{T} \\ \models \mathcal{P}^1(r_s) \\ \mathcal{F}^n.\mathcal{S} = \mathcal{F}^p.\mathcal{S} \cup \{s_s\} \\ \mathcal{F}^n.\mathcal{C} = \mathcal{F}^p.\mathcal{C} \cup \{r_s\} \\ \mathcal{F}^n.\Phi = \mathcal{F}^p.\Phi \cup \{r_s \rightarrow s_s\} \cup \{r_p \rightarrow s_s\} \cup \{o_p \rightarrow s_s\} \\ AddTransition(t, s_p, s_s, at) \end{cases}$$

•**RemoveAddState**(r_r, o_r, s_p): A right r_r and obligations o_r are removed from the policy level. If the right r_r and obligations o_r are activated in the state s_p then a new state s_s is created in which r_r or o_r are removed (it depends when the change happened) and the rights r_p and the obligations o_p mapped to the state s_p are propagated to the new state s_s. A new transition t is added and

the $t_{timeout}$ is assigned to it.

$$[[RemoveAddState(r_s, o_s, s_p)]] = \begin{cases} s_s \notin \mathcal{F}^P.S \text{ and } t \notin \mathcal{F}^P.\mathcal{T} \\ \mathcal{F}^n.\mathcal{S} = \mathcal{F}^p.\mathcal{S} \cup \{s_s\} \\ \mathcal{F}^n.\mathcal{C} = \mathcal{F}^p.\mathcal{C} - \{r_r \wedge o_r\} \\ \mathcal{F}^n.\Phi = \mathcal{F}^p.\Phi \cup \{r_p \rightarrow s_s\} \cup \{o_p \rightarrow s_s\} \\ AddTransition(t, s_p, s_s, timeout) \end{cases}$$

•**UpdateState**(s_c, at): If the attributes of a right r_c(the current right in the current state s_c) or obligation o_c are changed while they occurred in the state s_c, so, the state is updated within the new parameters at.

$$[[UpdateSate(s_c, at)]] = \begin{cases} s_c \in \mathcal{F}^P.S \\ \mathcal{F}^n.\mathcal{C} = \mathcal{F}^p.\mathcal{C} - \{r_c and/or\ o_c\} \cup \{r_c(at)\ and/or\ o_c(at)\} \\ \mathcal{F}^n.\Phi = \mathcal{F}^p.\Phi - \{(r_c\ and/or\ o_c)\ \rightarrow s_c\} \cup \\ \{(r_c and/or\ o_c)(at) \rightarrow s_c\} \end{cases}$$

where $r_c(at)$ means the attribute at associated to the right r_c.

•**UpdateTransition**(t, at): If some attributes of the operation associated to a right or a time conditions are updated while the transition is occurring, so the transition is updated.

$$[[UpdateTransition(t, at)]] = \begin{cases} t \in \mathcal{F}^P.\mathcal{T} \\ \mathcal{F}^n.\Psi = \mathcal{F}^p.\Psi - \{(r_c.op\ and/or\ o_c)\ \rightarrow s_c\} \cup \\ \{(r_c and/or\ o_c)(at) \rightarrow s_c\} \end{cases}$$

5.3 Privacy-Agreement lifecycle

An agreement life-cycle is represented by an automaton, as depicted in Figure 4. It includes all states in which the agreement is. When an agreement is created, it is not deduced, it is activated and 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 evolve, so it moves to *whipped up* state. The *checked* state is the core state

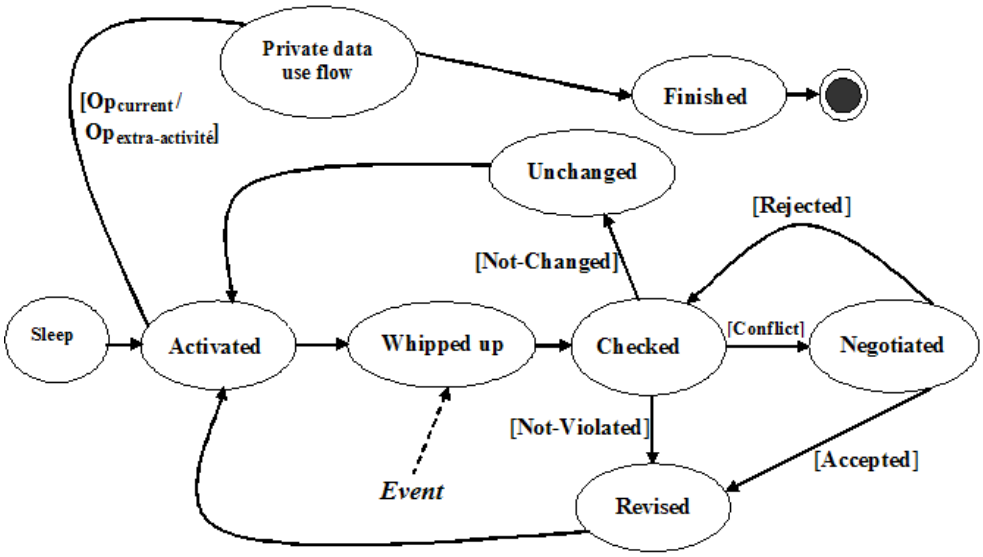

Fig. 4. The Privacy Agreement lifecycle

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 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 4.2.3.

The semantic of each state is defined in Table 3.

$[[sleep]]$	The agreement is created and not used e.g. not monitored
$[[activated]]$	The service involving the agreement is running then the agreement is activated
$[[whipped\ up]]$	During the running service an event occurs subject to change the agreement
$[[checked]]^{[Not-violated]}$	The agreement is checked if no conflict exists
$[[checked]]^{[Conflict]}$	The agreement is checked when a conflict exists then a negotiation is started
$[[checked]]^{[Not-changed]}$	The checking implies no changes in the agreement
$[[negotiated]]^{[Accepted]}$	The agreement is negotiated and accepted by the two parties
$[[negotiated]]^{[Rejected]}$	The negotiation failed and starts again until an agreement is defined
$[[revised]]$	The agreement is revised and is running again with new updates
$[[unchanged]]$	After the event being occurred, the agreement remains unchanged
$[[finished]]$	The agreement is terminated
$[[private\ data\ use\ flow]]$	Clauses of the agreement are activated

Table 3
The semantic of the states

6 Agreement Negotiation Protocol

In order to preserve or revise a 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 include a *negotiation language* defined previously, and an *interaction mechanism* that the parties must follow to come to an accord. Such a mechanism is based on Rubinstein's Alternating Offers Protocol [18], where two parties A1 and A2 participate in the negotiation process and make offers and counter-offers. In our framework, we modify such a 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 define in the agreement, otherwise the penalties will be fired.

The protocol $\mathcal{ANP}$

During the negotiation session each party uses suitable actions when com-

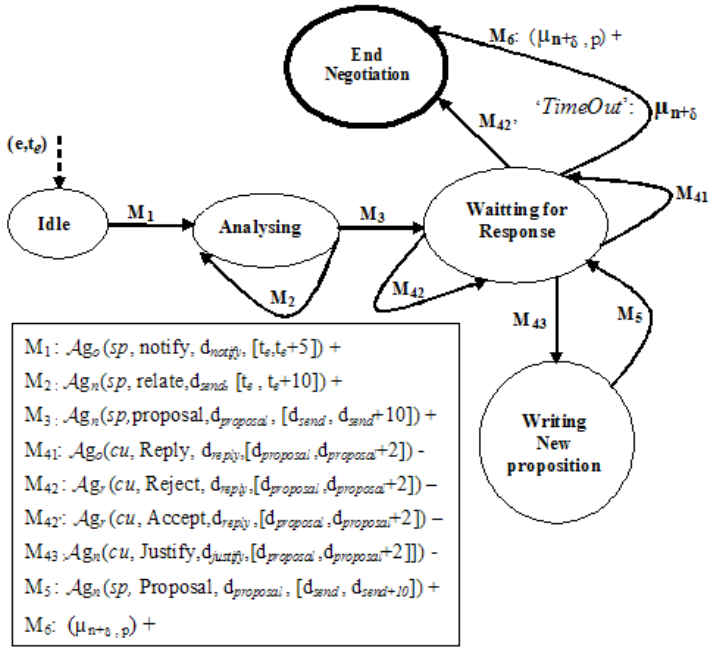

Fig. 5. ($\mathcal{P}_1$): Provider's Negotiation Protocol

municating with each other. The service provider should *notify* the 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 its decision about the received agreement pri-

vacy proposition. The service costumer has the right to *accept* or *reject* the proposition and in this case he must send some additional information about a 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 a 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 the service provider at time instant t is accepted by the 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 the customer (respectively the provider). Transitions are triggering by messages sent by the customer to the provider or vice versa. Figure 5 shows a graphical representation of a protocol called $\mathcal{P}_1$ that describes the behavior of the negotiation involved by the service provider. Each transition is labeled with a message using the agreement negotiation language followed by a polarity, that is, whether the message is incoming (minus sign) or outgoing (plus sign). For instance, the negotiation is initially in the **Idle** state, and the provider starts the negotiation when an event e happens at time t_e and some conflicts with the agreement data term, by sending a *notify* message, upon which the provider moves to the **analyzing** state (transition $\mathcal{AG}_o(sp, Notify, d_{notify}, [t_e, t_e + 5]^+)$. In this state the provider *relates* the event regarding privacy agreement and sends it to the customer and remains in this state until he'll provide a proposition to the customer (transition $\mathcal{AG}_n(sp, Proposal, d_{proposal}, [d_{send}, d_{send} + 5]^+)$, upon which the provider moves to the **waiting** states. If the interval time of the negotiation is over, then the negotiation will end and penalties will be fired. The states depicted in Figure 5 are describing the provider behavior when a negotiation starts between him and the customer.
We can define formally the state machine as follows:

Definition 18 ***(Agreement Negotiation Protocol)***
Formally an agreement negotiation protocol is a tuple $\mathcal{ANP} = (\mathcal{S}, s_0, \mathcal{F}, \mathcal{M}, \Delta, \mu_n, \mathcal{P}_n)$, *which consists of the following elements :*

- $\mathcal{S}$ *is a non-empty 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* $m \in \mathcal{M}$. *In the sequel for the sake of the clarity, we use* $m()^+$ *(respectively* $m()^-$*) to denote the outgoing (respectively incoming) messages.*
- $\Delta \subseteq \mathcal{S} \times S \times \mathcal{M}$ *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*
- $\mathcal{P}_n$ *is the set of the penalties activated when the negotiation time interval*

is over and the parties have not find an issue.

6.1 Negotiation strategy

We discuss here a possible strategy used in $\mathcal{ANP}$ based on the one proposed in the game theory presented in [9][18]. There are some elements in the agreement that the negotiation is looking for so that the parties can find an issue during the negotiation.
We denote by $\Sigma = \{\sigma_1, \sigma_2...\}$ all possible proposals that can be reached at a discrete time t = (0,1,2...). The possible agreement is referenced by a couple (σ, t).
A weight is assigned to each proposal (for both provider and customer) regarding the best element in the agreement for both of them, using a weight function $\mathcal{W} : (\Sigma \times T) \rightarrow \mathcal{R}$, $\mathcal{W}(\sigma, t) = w$ with $w \in \mathcal{R}$ and $t < \mu_n$. The strategy for each role is to maximize the weight w, in other words, to find the *best possible solution for the agreement.*

Definition 19 ***(The Best Proposed Agreement)***
Let us assume the set of all possible proposals that the customer can accept is denoted by $Solution(Role, t)$ *(in this case Role=cu), with* $t < \mu_n$. (σ, t) *is defined as the best proposed agreement iff* $(\sigma, t) \in Solution(Role, t) \neq \oslash$ *where* $Role \in \{cu, sp\}$ *and satisfies* $\mathcal{W}(\sigma, t) = max_{\sigma \in Solution(Role,t)} \mathcal{W}_{Role}(\sigma)$.

If S_{sp} the strategy developed by the provider and S_{cu} developed by the customer the couple (S_{sp}, S_{cu}) is called the *equilibrium solution* as defined in game theory if none of them changes the strategy using new weights. We present hereafter an algorithm, called privacy term negotiation, describing the negotiation between parties regarding terms affected by change.

7 Architecture and Implementation

In order to support the privacy agreement negotiation model described in this paper, we devise an architecture of a system providing an explicit management of the privacy agreement which is one component of the ServiceMosaic platform.

Algorithm 1 Privacy Term Negotiation

Require: t_e, μ_n {t_e time when event e happens, μ_n is the time period of negotiation}

Ensure: Revised terms.

1: $\mathcal{AG}_o(sp, notify, d_{notify}, [t_e, t_e + x])$ {service provider sp notifies service customer cu when an event e happens}
2: $\mathcal{AG}_n(sp, relate, d_{send}, [t_e, t_e + x])$ {service provider sp relates which personal data in the agreement is affected by a change }
3: **while** not μ_n and decision=reject **do**
4: $\mathcal{AG}_n(sp, proposal, d_{proposal}, [d_{send}, d_{send} + x])$ {sp suggests a proposition to cu}
5: $\sigma_i = proposal$
6: $t_i = d_{proposal}$
7: $W(\sigma_i, t_i) = w_{spi}$ { service provider assignes a weight to the proposal at t_i}
8: $\mathcal{AG}_o(cu, reply, d_{reply}, [d_{proposal}, d_{proposal} + x])$
9: decision=reply
10: $t'_i = d_{reply}$
11: **if** decision=reject **then**
12: $\mathcal{AG}_n(cu, justify, d_{justify}, [d_{proposal}, d_{proposal} + x])$
13: $\mathcal{W}(\sigma_i, t'_i) = w_{cui}$ {service customer assignes a weight to the proposal at t'_i}
14: **else**
15: End negotiation
16: **end if**
17: **end while**
18: **if** μ_n **then**
19: $\mathcal{W}(\sigma, t) = max_{\sigma \in Solution(Role,t)} \mathcal{W}_{Role}(\sigma)$ {(σ, t) is the best proposition}
20: **end if**
21: **if** no consent **then**
22: $p \rightarrow role$ {the action of penalizing $role \subset \{cu, sp\}$ by applying penalty $p \subset \mathcal{P}_n$}
23: **end if**

In the architecture of the privacy agreement management system is implemented the negotiation model between signing parties to manage the behavior of services when possible events may happen and also their interactions. Developers can visually edit privacy agreement model and generate a negotiation as depicted in Figure 7. We focused on providing tools to support the negotiation as well as the detection and analysis of relevant events in the dynamic environment of web services. An overview of our architecture is shown in Figure 6 which consists of following components:

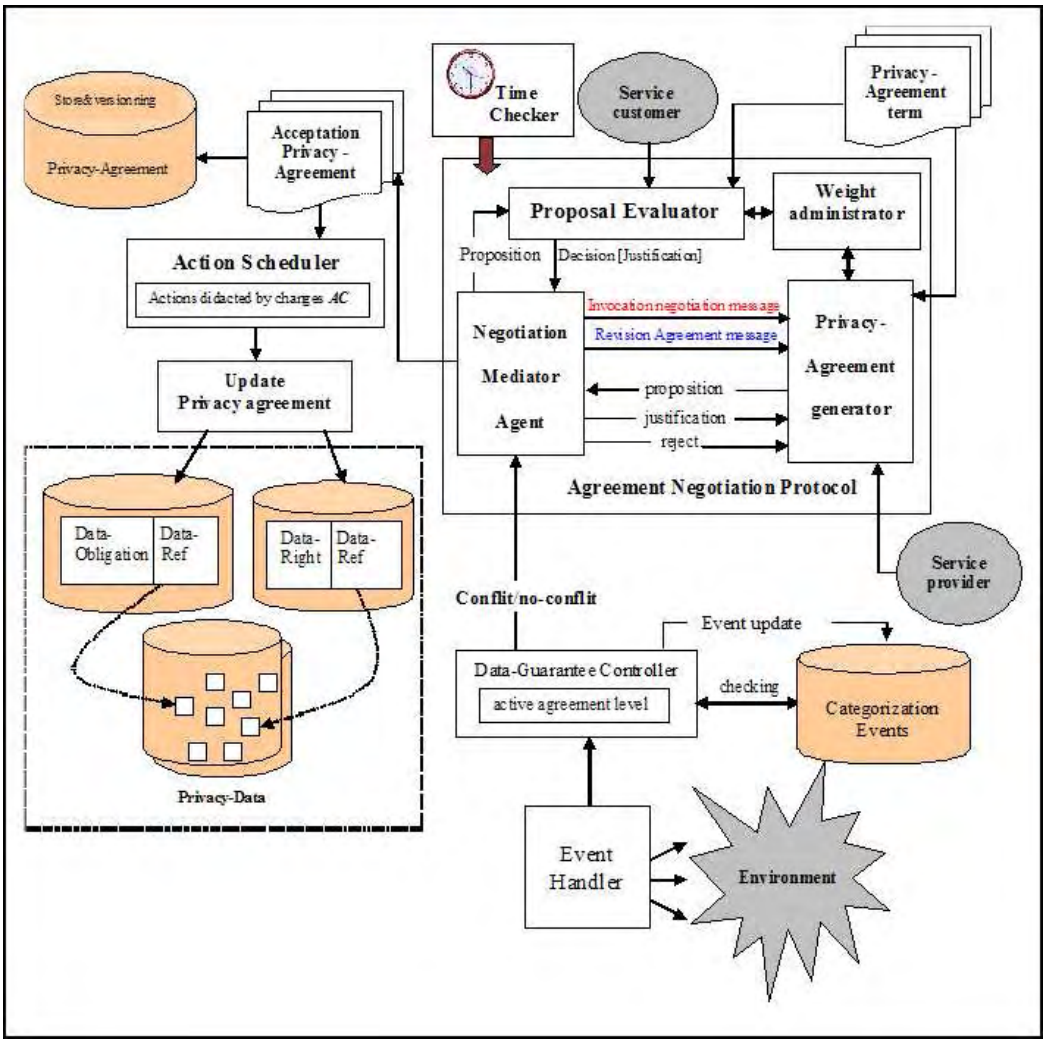

Fig. 6. Architecture of privacy agreement Management System

- **Event Handler** monitors and detects relevant events in the environment which may affect the privacy data term. The detection of events can happen via instrumented application/services. They can also be directly generated by service provider.

- **Data guarantee controller** analyzes the events coming from the event handler by means of the categorization event module. If the event is not known then a negotiation is activated. However, if the event is known the data controller identifies the category of the event. In fact, the events are classified in two categories: (1) $Conflicting\ events: CEvent = \{Data-driven, purpose-driven, data-user\ driven, duration-driven\}$ (2) *No Conflicting events:* $NCEvent = \{Security-action\ driven\}$. In case of the conflicting events new negotiation instances should be created.
- **Negotiation Mediator Agent** receives message from the Data guarantee controller and forwards it to the Privacy Agreement generator.These messages might contain the invocation for a negotiation when conflicting events happened or on contrary a revision agreement message in case of no conflicting events . This component plays also the role of the mediator between Privacy-Agreement generator and Proposal Evaluator.
- **Privacy-Agreement Generator**, it is an editing interface which assists the service provider to generate a proposition.
- **Proposal Evaluator**, in collaboration with the service customer, it evaluates the proposal regarding the customer preferences and generates an appropriate response based on this evaluation.
- **Weight Administrator** has the responsibility (i) to assign the weight to each proposal by summing separately the weights affected by the provider and customer for each term revised or proposed in the proposal (ii) to define the equilibrium solution or select the best proposed agreement by calculating for each party the maximum of the weights affected to the proposition.
- **Acceptation Privacy-Agreement** is the result of the negotiation or revision processes and contains a set of terms that will be modified or added in the privacy agreement term.The Acceptation Privacy-Agreement is stored as XML documents.
- **Action Scheduler** generates a set of actions $\mathcal{AC}$ in the table from documents sent by the Acceptation Privacy-Agreement module and specifies which data-obligations and data-rights are concerned by these change actions.
- **Update Privacy agreement** executes all the actions defined in the action table on an appropriate data-right and data-obligation.

7.2 Implementation and Evaluation

Given the above specifications for privacy agreement and interactions between signing parties, both service provider and costumer need an infrastructure to manage, propose and evaluate the proposition. In this section, we discuss the implementation of a framework which should help service provider to manage privacy agreement and generate a proposition to the customer. In the case of the consumer, the framework should allow him to decide whether to accept or

reject and justify incoming propositions.

The management of the privacy agreement and the experiments has been implemented in java and run using Sun's JDK 1.6 on an AMD Athlon-XP 2.0 GHz with 512 MB memory. We provide different graphic user interfaces to assist the service provider to generate the propositions during the negotiation process that allow the service client to evaluate this proposition as depicted in Figure 7.

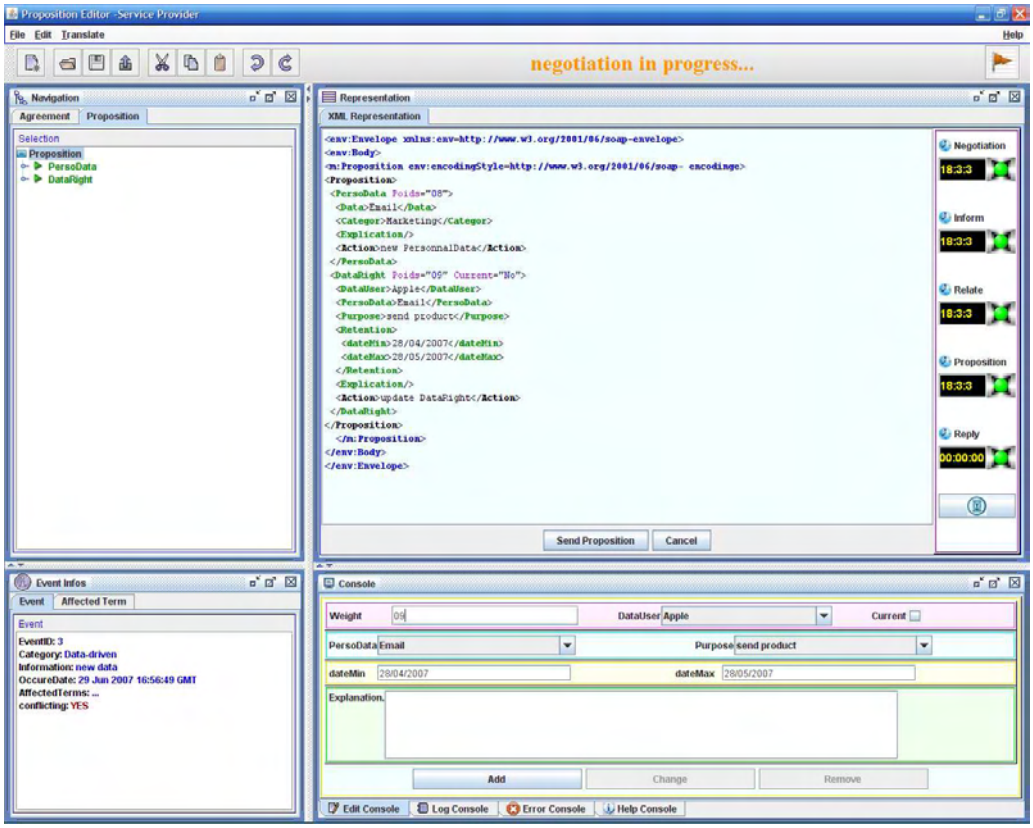

Fig. 7. Privacy-Agreement Generator interface

Scaling well the consequence and influence of the event are important to maintain the validity of privacy Agreement. We must note that each event generates one instance of negotiation. To evaluate the impact of each event in the negotiation, we measure the time of the negotiation and a number of the proposition proposed by the service provider to the customer. Each party assigns a weight to the proposition regarding the emphasis degree of the private data. These two measurements express the persuasion degree to convince the service customer to agree with the changes in privacy agreement.
The graph depicted in Figure 8 shows the performance of the proposition acceptance by the customer. The graph in the left shows a high level of acceptance of the changes, while the graph on the right shows low level of acceptance of different propositions.
The graph in Figure 9 shows for each event the time taken in the negotia-

tion and the number of the proposition proposed by the service provider to persuade the customer to make the revision. The vertical axis of the graph (Figure 9) expresses both the negotiation time and the number of the propositions. As we can see from the graph, the increasing number of the proposition causes a linear increase in the time taken in the negotiation instance.

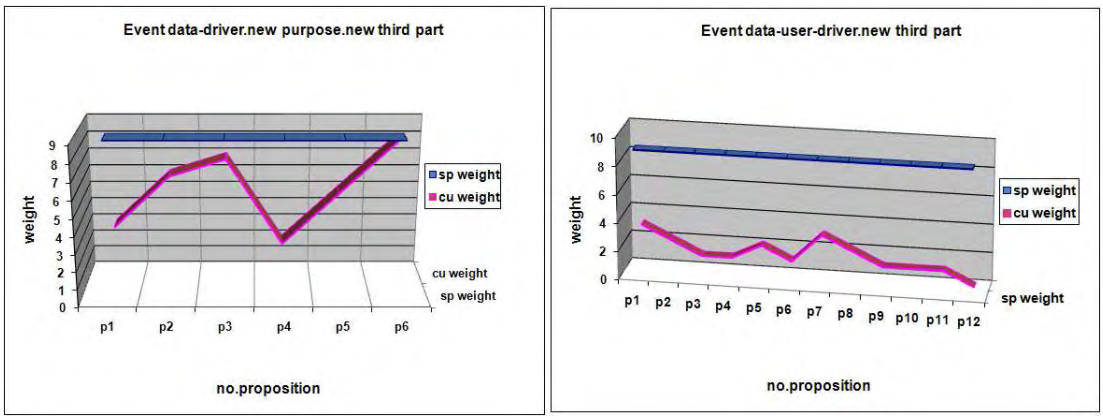

Fig. 8. The evaluation of the acceptance degree of the propositions by the customer.The graph in the left shows a high level of acceptance of the changes. On the right side we can observe a low level of acceptance of different propositions.

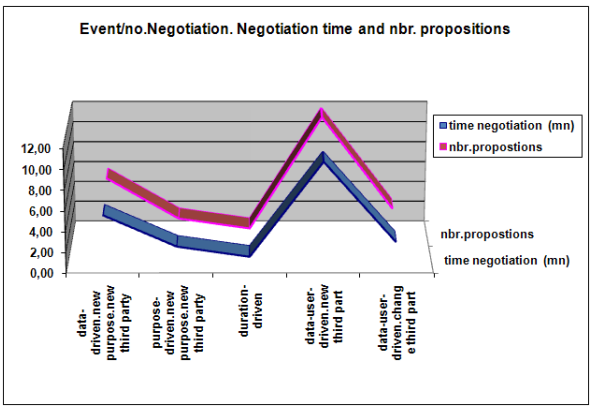

Fig. 9. The evaluation of the negotiation time and the number of the propositions proposed by the provider for each event.

8 Related Work

Our work is related to efforts in providing privacy abstractions for web services. 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 [19,5]. An Enterprise Privacy Architecture is introduced in [19]. The semantics of the E-P3P language for enterprise privacy policies is described in [20]. An obligation management model and the parametric obligation policies and a

scalable obligation management system are defined in [21–23].However, they are all related to enterprise.
A work has been done to deal with policy management such as [24–26]. In [24], the authors formalize the obligations and investigate mechanisms for monitoring obligations. In [25] the authors showed how different aspects of data protection can be handled by an extension of access control models. In [26], they propose an approach using free variable tableaux for detecting conflicts resulting from the combination of various kinds of authorization and constraint policies used in Web Services environments. In [27], the authors describe a formalism called a 'privacy system' that adapts access control matrices to the context of privacy. They have developed an architecture based on DRM that can carry out the negotiations to establish the rights in a privacy system. These works deal with the access control area.
There are a few number of research works related to Web services privacy policies. The work in [28] presents 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 [7]. 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 negotiation protocol in web services are described in [29,17,30]. The specification for a domain-independent,symmetrical, two-party negotiation protocol to reach binding agreements between services based on the principles of contract law is presented in [31]. These works don't take into account the privacy in the negotiation.
The works related to the privacy negotiation are proposed in [32,33]. In [32], an approach for bilateral negotiation between an e-service provider and an e-service consumer in the presence of uncertainty is presented. The approach can be applied to any type of negotiation, including buyer/seller negotiation. The type of uncertainty discussed is uncertainty of what offers and counter-offers to make, at a particular point in the negotiation. In [33], presented a privacy negotiation protocol that can be used to negotiate a binding privacy policy between two entities. It also presented an architecture that used the protocol and identified an extension of the P3P specification in order to express a policy template with all possible alternatives for each term in the policy. No evolution of the privacy policy has been taken into account.

There are numerous framework on privacy which investigated the privacy in another point of view than the one discussed in Web services area, such as [34–41]. In [34], a new method is presented for collaborative filtering which protects the privacy of individual data. The method is based on a probabilistic factor analysis model which handles missing data without requiring default

values for them. In [35], the requirements for secrecy in user-adaptive systems are discussed, to guarantee the anonymity of both users and user modeling servers. Giving users the option to conceal their identities seems a viable way to alleviate users privacy concerns whilst preserving the benefits of personalized interaction. Users trust in anonymity can moreover be expected to lead to more extensive and frank interactions, thus to more data about the user, and hence to better personalization. In [36],the authors evaluate the usability of online privacy policies as well as the practice of posting them. They analyze 64 current privacy policies, their accessibility, writing, content and evolution over time. They examine how well these policies meet user's needs and how they can be improved. They determine that significant changes need to be made to current practice to meet regulatory and usability requirements. The research in [37] examines the privacy comfort levels of participants if others can view traces of their web browsing activity. During a week-long field study, participants used an electronic diary daily to annotate each web page visited with a privacy level. Content categories were used by participants to theoretically specify their privacy comfort for each category and by researchers to partition participant's actual browsing. The study in [38] examines how Internet businesses can motivate consumers to disclose their personal information. It identifies seven types of benefits that Internet businesses can provide when soliciting personal information from consumers.It presents an instrument with which Internet businesses can find out the preferences of consumers for the seven types of benefits and associates personality factors with preferences for the various types of benefits which give Internet businesses some hints about the types of benefits that their consumers might prefer. the work in [39] discusses a pervasive bug in web application software. It presents a number of direct and indirect measurement techniques that can effectively exploit real-world leaks of private information, including a new cross-site timing method that can reveal private user state. In [41], the authors investigated the feasibility of achieving a balance between users privacy and search quality. An algorithm is provided to the user for collecting, summarizing, and organizing their personal information into a hierarchical user profile. Through this profile, users control what portion of their private information is exposed to the server by adjusting the minDetail threshold.

It is worth mentioning that several ongoing efforts in the area of privacy recognize the need for high level-specification of privacy. However, in the field of the web services, up to date no privacy abstractions has been spelled out and no privacy evolution have been discussed.

Our work attempts to fill the aforementioned gaps in the state of the art by providing a privacy model for web services and a framework handling the privacy lifecycle management of Web services.

9 Conclusion

The main contribution of the work presented in this paper is a framework that leverages the privacy and contract representation technologies and established modeling notation (state machine-based formalism) to provide a high level support for describing the private data use flow as well as the evolution of privacy in Web services.
We proposed a formal model for privacy called privacy agreement which is an extension of WS-Agreement specifications, that both service customer and provider might agree before any process is running. We argue that endowing privacy in service with abstractions have benefits and is useful in several situations to automate a non functional activity in the web service. We have emphasized a lifecycle of privacy which is an important issue to date which has not been addressed . The proposed framework with such abstractions shows the automation process of the privacy lifecycle activities that can be greatly enhanced.
Based on a formalization of the private data use flow model, we have presented privacy policy evolution primitives and an agreement negotiation protocol that allow to evolve the privacy agreement in a new one. In fact, a flexible and a game theory based agreement-negotiation protocol enabling negotiation of a bilateral interaction mechanism between the parties is provided. The latter should preserve privacy-agreement and avoid conflicts between the parties when events happen during the running process, leading to a change in the web service privacy agreement.
Finally, we point out that the framework presented in this paper is only one component of a Broader CASE tool in ServiceMosaic platform, partially implemented, that manages the entire service development lifecycle.
Future work around our framework will be considering the extension of the proposed approach by the refining and introducing a reasoning mechanism for the temporal aspect about agreement that may change over the time in the agreement negotiation protocol. We will also investigate the expansion of the framework by handling the composition of the services regarding the privacy-agreement.

References

[1] B.Benattallah, F.Casati, F.Toumani, Analysis and management of web srevices protocols, in: ER'04: International Conference on Conceptual Modeling, Springer Berlin Heidelberg, Shanghai, China, 2004, pp. 524–541.

[2] L. Cranor, M. Langheinrich, M.Marchiori, A p3p preference exchange language 1.0 (appel1.0), in: Technical report, W3C Working Draft, 2002.

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

[4] P. C. K. Hung, E. Ferrari, B. Carminati, Towards standardized web services privacy technologies, International Conference on Web Services (ICWS'04) 00 (2004) 174.

[5] I. Corporation, Enterprise privacy authorization language (epal), in: IBM Research Report, 2004.
URL www.zurich.ibm.com/security/enterprise-privacy/epal

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

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

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

[9] S. Paurobally, V. Tamma, M. Wooldridge, Cooperation and agreement between semantic web services, W3C Workshop on Frameworks for Semantics in Web Services. Innsbruck, Austria.

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

[11] OECD, Oecd guidelines on the protection of privacy and transborder flows of personal data, in: Information Security and Privacy, 1980.

[12] A. Antn, J. B. Earp, D. Bolchini, Q. He, C. Jensen, W. Stufflebeam, The lack of clarity in financial privacy policies and the need for standardization, IEEE Security and Privacy 2 (2) (2004) 36–45.

[13] Y. Li, S. Benbernou, Representing and reasoning about privacy abstractions., in: 6th International Conference on Web Information Systems Engineering,WISE 2005., 2005, pp. 390–403.

[14] Y. Li, S. Benbernou, H. Paik, B. Benatallah, Formal consistency verification between bpel process and privacy policy, in: Privacy Security Trust PST'2006., 2006, pp. 212–224.

[15] N. Guermouche, S. Benbernou, C. Coquery, M. Hacid, Privacy-aware web service protocol replaceability, IEEE International Conference on Web Services ICWS'07.
URL http://liris.cnrs.fr/membres/?idn=sbenbern&onglet=publis

[16] S. Benbernou, H. Meziane, Y. Li, M. Hacid, A privacy agreement model for web services, IEEE International Conference on Service Computing SCC'07.

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

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

[19] G. Karjoth, M. Schunter, M. Waidner, Privacy-enabled services for enterprises, dexa 00 (2002) 483.

[20] P. Ashley, S. Hada, G. Karjoth, M. Schunter, E-p3p privacy policies and privacy authorization., in: WPES '02, Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, ACM Press, New York, NY, USA, 2002, pp. 103–109.

[21] M. C. Mont, A system to handle privacy obligations in enterprises, in: HP Labs Technical Report, HPL-2005-180, 2005.

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

[23] M. C. Mont, F. Beato, On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises, policy (2007) 51–55.

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

[25] M. Hilty, D. Basin, A. Pretschner, On obligations, in: S. de Capitani di Vimercati, P. Syverson, D. Gollmann (Eds.), 10th European Symposium on Research in Computer Security (ESORICS 2005), Vol. 3679, Springer-Verlag, 2005, pp. 98–117.

[26] H. Kamoda, M. Yamaoka, S. Matsuda, K. Broda, M. Sloman, Policy conflict analysis using free variable tableaux for access control in web services environments, in: Policy Management for the Web, A WWW2005 Workshop 14th International World Wide Web Conference, ACM Press, Chiba, Japan, 2005, pp. 5–12.

[27] A. C. Gunter, M. J. May, S. G. Stubblebine, A formal privacy system and its application to location based services., in: Privacy Enhancing Technologies, 2004, pp. 256–282.

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

[29] L. Ardissono, A. Goy, G. Petrone, Enabling conversations with web services, in: AAMAS '03, Proceedings of the second international joint conference on Autonomous agents and multiagent systems, ACM Press, New York, NY, USA, 2003, pp. 819–826.

[30] H. Skogsrud, B. Benatallah, F. Casati, Trust-serv: Model-driven lifecycle management of trust negotiation policies for web services, in: Proc. 13th World Wide Web Conf., 2004.
URL `citeseer.ist.psu.edu/skogsrud04trustserv.html`

[31] M. Parkin, D. Kuo, J. Brooke, A framework and negotiation protocol for service contracts, IEEE International Conference on Service Computing SCC'06 (2006) 253–256.

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

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

[34] J.Canny, Collaborative filtering with privacy via factor analysis, in: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, New York, NY, USA, 2002, pp. 238–245.

[35] A. Kobsa, J.SchreckKobsa, Privacy through pseudonymity in user-adaptive systems, ACM Transactions on Internet Technology (TOIT) 3 (2) (2003) 149–183.

[36] C. Jensen, C. Potts, Privacy policies as decision-making tools: an evaluation of online privacy notices, in: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, New York, NY, USA, 2004, pp. 471–478.

[37] K. Hawkey, K. M. Inkpen, Examining the content and privacy of web browsing incidental information, in: WWW '06: Proceedings of the 15th international conference on World Wide Web, ACM Press, New York, NY, USA, 2006, pp. 123–132.

[38] K. Hui, B. C. Y. Tan, C. Goh, Online information disclosure: Motivators and measurements, ACM Trans. Inter. Tech. 6 (4) (2006) 415–441.

[39] A. Bortz, D. Boneh, Exposing private information by timing web applications, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM Press, New York, NY, USA, 2007, pp. 621–628.

[40] R. Kumar, J. Novak, B. Pang, A. Tomkins, On anonymizing query logs via token-based hashing, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM Press, New York, NY, USA, 2007, pp. 629–638.

[41] Y. Xu, K. Wang, B. Zhang, Z. Chen, Privacy-enhancing personalized web search, in: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM Press, New York, NY, USA, 2007, pp. 591–600.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube)

Appendix A. An Example of Privacy Agreement

$Term_1$: Privacy agreement was signed at an absolute time $date_s$.

$Term_2$: The period of privacy agreement's validity is specified by $val \in [date_s, t_2]$.

$Term_3$: All information (e.g. purchase order, payment, notifications,.) are to be sent electronically.

$Term_4$: This Privacy agreement is governed by Country law and the parties hereby agree to submit to the jurisdiction of the Courts of the Country with respect to this privacy agreement.

$Term_5$**: Personal data collected**

$Term_{51}$: Contact information : Email and contact address.

$Term_{52}$: Financial information: Credit card number.

$Term_6$**: Duration**

$Term_{61}$: Retention period of Email is equal to val.

$Term_{62}$: Contact address is kept 3 months after the $date_s$.

$Term_{63}$: Credit card number is held until the end of each process of payment.

$Term_7$**: Right on data**

$Term_{71}$: Service provider use Email to send product available, price, periods of promotion sales and invoices.

$Term_{72}$: Service provider use contact address to deliver product.

$Term_{73}$: Service provider use Credit card number for the payment of invoices.

$Term_{74}$: Company C share Contact address.

$Term_{75}$: Company C use Contact address to deliver product.

$Term_8$**: Obligation on data**

$Term_{81}$: After the contact address has been used by Company X for a certain period of time $[d_1, d_2]$ where $d_1 \geq date_s$ and $d_2 \leq t_2$ (Privacy agreement validity), service provider must replace contact Address with an update data subject's address at absolute time d_{update}.

$Term_{82}$: Service provider must Crypt credit card number at the end of each process of payment.

$Term_9$**: Events**

$Term_{91}$: Add new data which becomes necessary at time t.

$Term_{92}$: New purpose associated to data which become necessary at time t.

$Term_{93}$: Increase and decrease the data retention during period data retention validity or after data retention expiration.

$Term_{94}$: Add new third party which will help the service provider to do particular work.

$Term_{95}$: Change third party for some reasons.

$Term_{96}$: Change security on the data to avoid for instance new security threats.

$Term_{10}$**: Agreement Right**

$Term_{101}$: Service customer has the right to reject privacy agreement proposition made by the service provider.

$Term_{102}$: Service customer has the right to accept privacy agreement proposition made by the service provider.

$Term_{103}$: Either the provider or the consumer can end negotiation if they decide to give up the negotiation process.

$Term_{11}$**: Agreement Obligation**

$Term_{111}$: the service provider must notify the service customer that a given event happened at instant time t_e where $date_s \leq te \leq t2$ ($[date_s, t_2]$ Privacy agreement validity), But he shall notify it within five days after the occurrence of this event.

$Term_{112}$: Service customer must reply by sending his decision of the received privacy agreement proposition within two days after the receipt of the proposition .

$Term_{12}$**: Agreement Negotiation**

$Term_{121}$: If parties fall in conflict, they must negotiate to arrive to a Privacy Agreement revised.

$Term_{122}$: Service provider relates which personal data in the agreement is affected by a change and sends it as a report within 10 days after the occurrence of the event.

$Term_{123}$: Service provider proposes a privacy agreement proposition to service costumer within 10 days after sending a detailed report.

$Term_{124}$: If service customer rejects the proposition, he justifies the refusal reply by including additional information about his decision within 2 days after the receipt of the proposition.

$Term_{125}$: If service customer accepts the proposition, he should notify his decision to service provider within 2 days after the receipt of the proposition.

$Term_{13}$**: Penalties**

$Term_{131}$: If one party violates some terms defined in privacy agreement, the relevant authorities will be informed of the default or the agreement will be terminated by force.

Table .1
Realistic Example of a Privacy-Agreement between signing parties

1. **Privacy-data-term**

$R^{email} = \left\{r^1_{email}, r^2_{email}\right\}$

$R^{credit\ card} = \{r_{ccn}\}$

$R^{contact\ address} = \{r_{contact\ address}\}$

$r^1_{email}\,(sp, email, send\ offer, \mu_{remail})$ [Email is used for marketing purpose (e.g. send promotions)].

$r^2_{email}\,(sp, email, send\ invoice, \mu_{remail})$ (1)
$r_{ccn}(sp, credit\ card, payment\ invoice, \mu_{rccn})$ (2)
$r_{contact\ address}(C, contact\ address, deliver\ product, \mu_{rcontact@})$ (3)

[Company C use Contact address to deliver product]

(1,2,3) are used to complete the activity of the service for which it was provided.

$O^{contact\ address} = \{o_{contact\ address}\} \quad \rightarrow o_{contact\ address}(sp, contact\ address, update, \mu_{ocontact@})$

$O^{credit\ card} = \{o_{ccn}\} \quad \rightarrow o_{ccn}(sp, credit\ card, Crypt, \mu_{occn})$

2. **Events triggering actions involving changes in Privacy-data-terms**

$e(e_{id1}, data - driven, c_{eid1}, t_{eid1})$
$e(e_{id2}, purpose - driven, c_{eid2}, t_{eid2})$
$e(e_{id3}, duration - driven, c_{eid3}, t_{eid3})$
$e(e_{id4}, data\ user - driven, c_{eid4}, t_{eid4})$
$e(e_{id5}, securite\ action - driven, c_{eid5}, t_{eid5})$

3. **Agreement-Negotiation-action**

$\mathcal{AG}_r(cu, accept, d_{reply}, [d_{proposal}, d_{proposal} + 2])$ (1)

$\mathcal{AG}_r(cu, reject, d_{reply}, [d_{proposal}, d_{proposal} + 2])$ (2) $\rightarrow (1, 2)$ **Agreement-right**

$\mathcal{AG}_o(cu, reply, d_{reply}, [d_{proposal}, [d_{proposal} + 2])$ (3)

$\mathcal{AG}_o(sp, notify, d_{notify}, [t_e, t_e + 5])$ (4) $\rightarrow (3, 4)$ **Agreement-obligation**

$\mathcal{AG}_n(cu, justify, d_{justify}, [d_{proposal}, d_{proposal} + 2])$ (5)

$\mathcal{AG}_n(sp, relate, d_{send}, [t_e, t_e + 10])$ (6)

$\mathcal{AG}_n(sp, proposal, d_{proposal}, [d_{send}, d_{send} + 10])$ (7) $\rightarrow (5, 6, 7)$ **Agreement-negotiation**

Table .2
Rights and Obligations of Privacy-Agreement signatories