P3P Privacy Enhancing Agent

A Writing Project

Presented to

The Faculty of the Department of

Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

By

Hsu-Hui K. Lee

December, 2006

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Mark Stamp

Dr. Chris Pollett

Dr. Richard Low

APPROVED FOR THE UNIVERSITY

Acknowledgements

I would like to thank Dr. Mark Stamp for his guidance, patience, insights and encouragement without which my project would not have been possible. I would also like to express my thanks to Dr. Richard Low and Dr. Chris Pollett for agreeing to be the committee members to review and certify my project.

I would also like to thank all my friends who have supported me immensely during this project.

Finally, I would like to thanks my family members and most importantly my wife Terri for their support throughout my graduate studies. Without their cares, patience and encouragement, I would not be able to complete this important task of my life.

Abstract

Protecting personal privacy information is an inherently difficult problem. Privacy Enhancing Agent is collection of software agents that help web users to protect their private information by collecting web site P3P [1] information and exchanging knowledge of web site privacy practices with other agents. Based on the information collected, the software agents begin the decision-making and negotiation process. A suggestion or warning is presented to the user if any discrepancy is detected [12]. In this paper, we propose our design for a P3P Privacy Enhancing Agent (PEA).

Keywords: Privacy, Personal Identifiable Information (PII), Platform for Privacy Preferences Project (P3P), Software Agent, Ontology.

Table of Contents

1. Introduction

In this paper, we present a Privacy Enhancing Agent (PEA) , which is a software agent system developed using the Java Agent DEvelopment Framework (see Section 1.5) to address privacy protection issues related to the growing marketplace on the Web. The purpose of PEA is to enhance personal privacy information protection by learning from user behavior, gathering company privacy policies and examining the track record of privacy policy practices histories obtained from other PEA agents. To aid users in Web privacy information transactions, PEA is deployed on the client machine. It analyzes Web site P3P policy and gathers any related privacy practice information from other agents within the global agent community as shown in Figure 1-1.

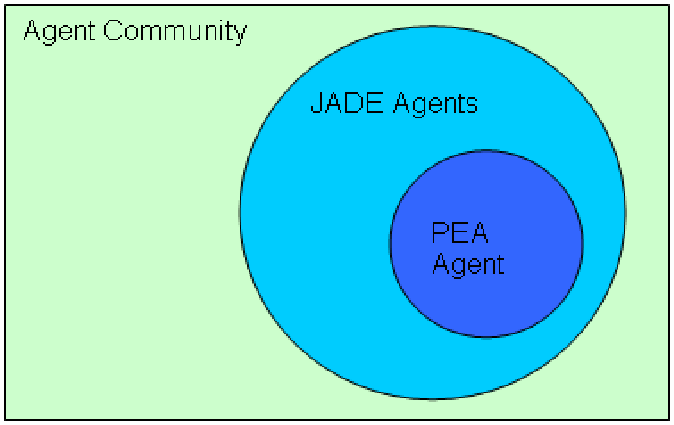


Figure 1-1: Agent Environment

PEA takes into account the following factors with regard to a company's P3P privacy policies:

- Predefined personal privacy policy preferences.
- User Web transaction history of a Web site and its privacy policies.

- Relevant privacy policy information and decision from other PEA agents.
- PEA agent's suggestions and recommendations on Web site privacy policy practices.

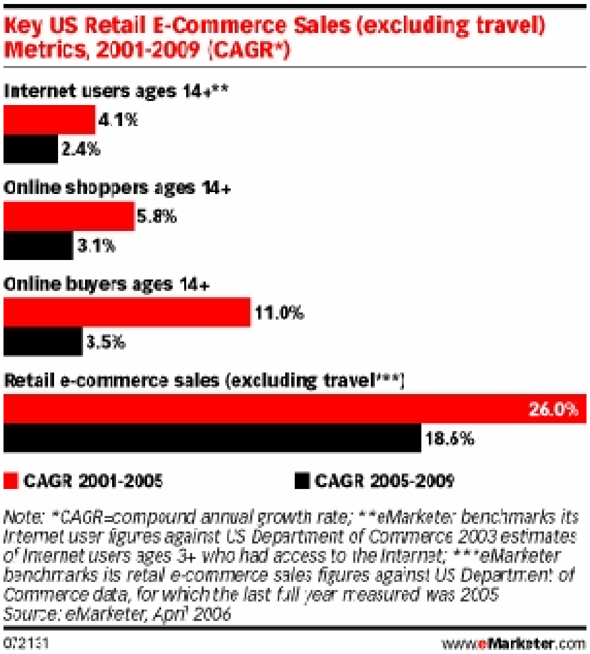
The purpose of PEA is to aid both users and collectors in managing online personal identifiable information (PII) in a reasonably secure and transparent manner. At a high level, PEA performs the following functions:

1. Automatically retrieve P3P privacy policies from Web sites.
2. Automatically check privacy policies against predefined user privacy preferences.
3. Assist users in the decision making process by analyzing privacy polices, user preferences, past transaction history and "knowledge" gained from the experiences of other users.

PEA is able to take appropriate action to assist users and to enhance the user experience by protecting privacy in a transparent way, which should increase user confidence that PII is being handled properly. PEA should increase consumer confidence in online transactions, at least with respect to PII issues [16].

1.1 Privacy and Security on World Wide Web

The Web has become a major marketplace where people buy and sell goods and information. According to a recent report, "More than one billion people in the world have access to the Internet, with a quarter of them with



Key US Retail E-Commerce Sales (excluding travel) Metrics, 2001-2009 (CAGR*)

Note: *CAGR=compound annual growth rate; **eMarketer benchmarks its Internet user figures against US Department of Commerce 2003 estimates of Internet users ages 3+ who had access to the Internet; ***eMarketer benchmarks its retail e-commerce sales figures against US Department of Commerce data, for which the last full year measured was 2005
Source: eMarketer, April 2006

eMarketer, “Internet Sales of Big Ticket Items on the Rise”. 18 May 2006 <http://www.ebusinessnews.info/?action=read&article=27>

broadband or high-speed connections” [3]. However, despite the continued rapid growth of the Internet, e-commerce growth actually appears to be leveling off. From a recent report on U.S. retail e-commerce sales growth, “eMarketer estimates that retail e-commerce sales will increase an average 18.6% per year between 2005 and 2009 — that's strong growth, but still a downturn from the 26% annual rate seen between 2001 and 2005” [4]. Many factors might contribute to the downturn in the sales growth rate. Among them, a big negative factor is consumer concerns over the collection and use of their personal identity information, and security and privacy on the Web in general.

1.2 Personal Identifiable Information and Privacy

In the early 1990’s, the information exchange on the Web was uni-directional. The information received from the Web is often anonymous which posed no obvious threat to privacy information. With the emergence of online services and e-commerce, many Web sites keep track of visitors and collect the information for various reasons. The Web information flow is therefore transformed from a uni-directional to a bi-directional process. This transformation has raised many concerns of how to safeguard personal information being accessed and redistributed from one Web server to another Web server [20].

Most Web merchants today provide some degree of privacy protection, especially when the transaction involves releasing personal information over the Internet. There are still many issues to privacy information protection, such as

- Privacy information policies are generally created by lawyers and are therefore written in a language that can often be lengthy and intimidating for consumers. This discourages people from reading such policies.
- Web sites could abuse user's trust and disregard their stated policy, for example, by selling personal information.
- Once a user provides his or her personal identifiable information to a Web site, there is no way for the user to know how the information is stored and treated.
- There are no standard for privacy policies for the Web.

Such threats might deter users from online shopping and accessing online service on the Web. To address these issues, many standards have been developed. The World-Wide-Web Consortium started the P3P project (see Section 1.4) to try to standardize Web privacy policy over personal privacy information (PII) [14].

1.2.1 What is Personal Identifiable Information (PII)

In the context of information security and privacy, Personal Identifiable Information (PII) refers to any information that identifies or can be used to identify, contact or locate the person to whom such information pertains, or from which other personal identifiable information can be easily derived. PII includes names, addresses, phone numbers, fax numbers, e-mail addresses, financial profiles, social security numbers and credit card information. In terms of P3P definition, PII consists of 1) physical contact or location information, 2) online contact or location information, 3) government issued identifier, or 4) information about an individual's finances.

1.3 Platform for Privacy Preferences (P3P)

The P3P project was proposed by the World-Wide-Web Consortium to facilitate Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by software agents. The software can better informed users of a Web site privacy practices based on user transaction history and personal privacy preference. The main purpose of P3P is to standardize Web privacy policies and aid Web users to make better decision regarding the release of user’s personal information to a Web site [18].

1.4 Java Agent Development Framework (JADE)

JADE [5] is a software framework implemented in Java which provides the development environment for distributed multi-agent applications based on the peer-to-peer communication architecture. In JADE, the environment can evolve with agents. Also the communication between agents is completely symmetric which means that an agent can either be a communication initiator or responder.

As described in the JADE White Paper [7], JADE was developed based on the following principles:

- *Interoperability* – JADE is compliant with the FIPA (The Foundation of Intelligent Physical Agent) specifications [8]. As a consequence, JADE agents can interoperate with other agents, provided that the other agents comply with the same standard.
- *Uniformity and portability* – JADE provides a homogeneous set of APIs that are independent from the underlying network and Java version. More precisely, the JADE run-time provides the same APIs as the J2EE, J2SE and J2ME environment. In theory, application developers could decide the Java run-time environment at deploy-time.
- *Easy to use* – The complexity of the middleware is hidden behind a simple and intuitive set of APIs.
- Pay-as-you-go philosophy – Programmers do not need to use all the features provided by the middleware. Features that are not used do not require programmers to know anything about them. In addition, these unused features do not add any computational overhead.

The features of JADE can greatly benefit the development of the Privacy Enhancing Agent (PEA) discussed in this paper by reducing development time and effort. We can easily enable PEA agents to migrate from one client to another client which allows agents to share knowledge regarding Web site privacy policy and we can also control information sharing by setting different sharing rules depending on the profile of the requestor.

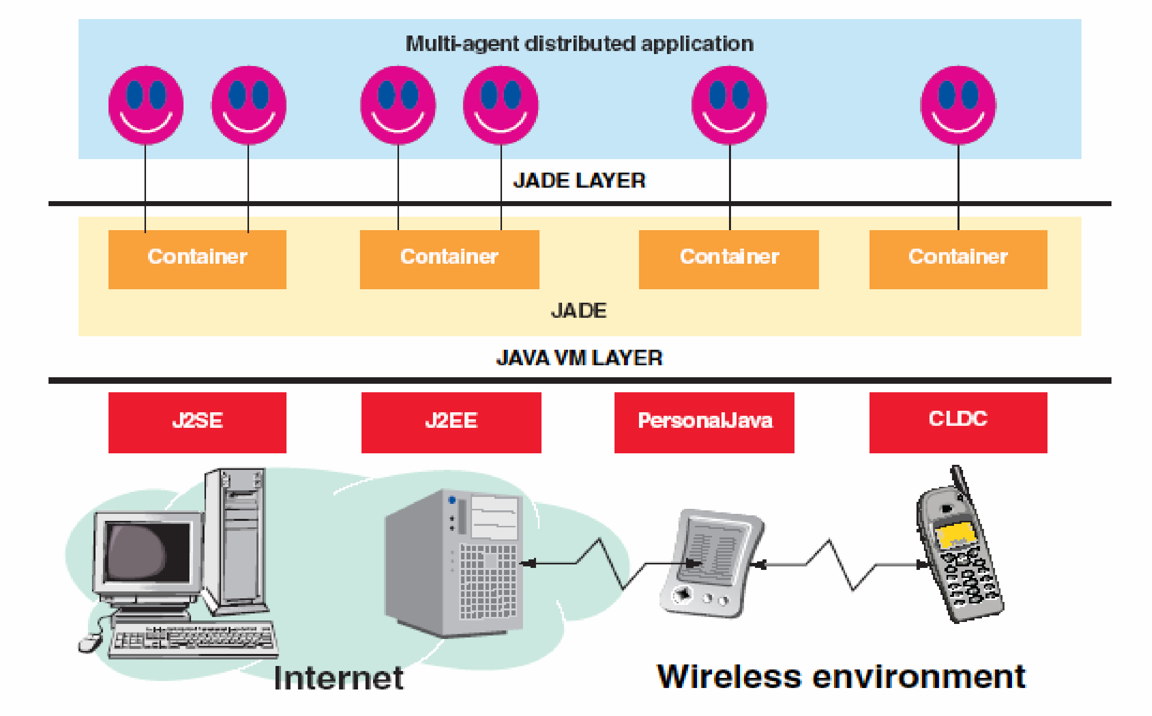


Figure 1-2: JADE Architecture

Tilab, JADE White Paper. "The JADE architecture". Sep 2003.

<http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf>

JADE also provides application-defined content language and ontology support. Our PEA agent system can take advantage of the predefined generic content languages and ontologies classes by describing the Web privacy policies and preferences. These policies and preferences are extended from the P3P policy schema using JADE ontology classes.

1.5 Scope of Paper

The paper illustrates how PEA agents can enhancing privacy protecting on the Web using P3P and JADE. Other aspects of PEA such as architecture design, transaction management, information negotiation, policy XML schema and ontologies, and implementation will be discuss in detail through the following chapters.

Chapter 2, Design Overview, describes PEA concepts and architecture. The responsibility of each module of PEA is briefly described. Chapter 3, Privacy Information Management and Transaction, provides the detail of how PEA manages P3P transaction using a generic example. A function model is presented to illustrate the transaction flow. Chapter 4, Privacy Information Negotiation, describes the algorithm which PEA used to verify and validate if a P3P policy is valid and doesn't violate user's privacy preference. The definitions which define the concept of negotiation are introduced in this chapter. Chapter 5, Privacy Policy XML Schematic and Ontologies, provides the detail XML schema and the corresponding ontology concepts of P3P policy and PEA privacy preference. With the XML and ontology concepts, PEA agents communicate and exchange knowledge. Chapter 6, Implementation of PEA, describes the detail implementation of PEA. Snapshots of UML class diagrams are used to demonstrate the fundamental class hierarchy of PEA. Follow by UML sequence diagrams to explain how PEA agents interact to one another to complete various tasks. The process of converting XML schemas to ontology concepts is covered with the detail of ontological elements. Chapter 7, Conclusion provides the summary of this paper and recap. Chapter 8, Future Work, describes the possibility of future works.

2. Design Overview

PEA has access to a user's personal privacy preferences. Armed with this information and the user's history, PEA monitors the user's Web transactions and responds according to the user's preference if an online transaction involves privacy information exchange. Based on a user's privacy preferences and the P3P policy of a given Web site, PEA performs the following functions:

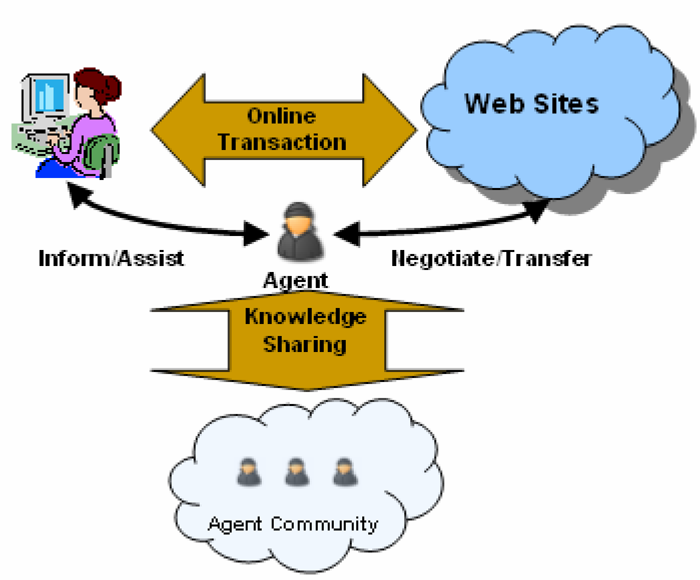


Figure 2-1 PEA Concept

- PEA informs the user when it detects potential privacy violations or if it is unable to understand a privacy policy. Under such circumstances, PEA will suspend the transaction and request user intervention.
- If the user has a positive transaction history with a Web site that requests PII, and the privacy policies of the Web site comply with the user's privacy preferences, PEA will authorize the transaction on the user's behalf, without user intervention. In this case, PEA automatically negotiates the policies, transfers the information and logs the transaction activity to a database for future reference. All of these actions are transparent to the user.
- In cases where PEA cannot complete the transaction on the user's behalf,

it can still assist the user by suggesting possible decisions base on user preferences, past transaction history and knowledge from other PEA agents, if available. Of course, the user can choose to either follow the suggestion or not. In any case, PEA will attempt to complete as much of the transaction as possible by negotiating policies based on the user's decision. The user can ultimately decide whether to complete the transaction or abort, based on these negotiated results. PEA will log the resulting information for future reference.

2.1 Basic Architecture

The architecture of PEA includes the following major components: the JADE framework, the PEA agent, a P3P knowledge module, a negotiation module, a database module, a GUI module and a mobile module. The PEA agent monitors Web traffic between a user's Web browser and the Web as illustrated in Figure 2.2. All Web interactions are filtered by the PEA agent. Once the PEA agent detects a privacy information transaction, the agent notifies the P3P agent. The P3P agent then retrieves the P3P privacy policy file from the requesting Web site and interprets the policy with its P3P ontologies and vocabularies. After the P3P module has done its job, the negotiation module will begin the negotiating process by analyzing the P3P information, based on knowledge of the user's privacy preferences.

During the decision-making process, the policy agent can request additional information from other PEA agents via a mobile agent. A mobile agent’s responsibility is to migrate to another machine on the network and retrieve information based on the requested information. In this way, the PEA agent can discover information known to other agents on other systems within the same JADE framework pertaining to a specific Web site. Before accepting such information, the PEA agent checks that the mobile agent passes a security profile check, as discussed below. Whenever user intervention or notification is required, a GUI agent will gather the information and pop up the appropriate type of GUI component [12]. The role of the database module is to record transaction activities, P3P policy files and user decisions for future references. The following sections will explain these components in detail.

2.1.1 P3P Module

Along with the negotiation module, the P3P module is one of the two main components in the PEA system. Its responsibility is to know about P3P and to translate a P3P policy XML file into a form that is understandable to an agent which is based on the JADE ontology API. All local agents and remote agents in the PEA system communicate based on the P3P “language”.

After the translation, the P3P agent will update the database P3P file and tables, and wake up the negotiation agent for the negotiation phase of the process. If there are any obvious problems, such as a P3P file that has expired or a P3P XML schema error, the P3P agent will inform the GUI agent to notify the user accordingly and the process will abort.

2.1.2 Negotiation Module

The negotiation module is another major component in our PEA system. The negotiation module takes over after the P3P module has successfully retrieved the P3P policy file and translated it into a policy. The negotiation agent matches the P3P policy with the user's privacy preferences. If the policy is compatible with the user's preferences, the negotiation module informs the P3P agent to grant the transaction and complete the process. In this scenario, the whole process is completed automatically and it is transparent to the user. On the other hand, if the policy does not comply, the negotiation agent will first attempt to analyze the policy based on the transaction activity history with respect to the specific Web site under consideration. If this analysis is not conclusive, the negotiation agent will check with the JADE directory facilitator agent (which is analogous to a telephone operator) on available agents and services. If there is any agent available and capable of performing the requested services, the negotiation module will send a request to mobile agents to go to the remote agents and retrieve information relevant to the specific Web site. After the information is gathered and consolidated, the negotiation agent will send the data to the GUI agent to present to the user. The user can then make a decision to either abort or complete the transaction, based on all available information. Of course, the information, activity logs and final results will be stored in the database for future reference.

2.1.3 Database Module

The database module is the archive which stores the P3P policy file and URL of visited Web sites, the user privacy preference and transaction history. Since both P3P policy files and PEA privacy preference files are in XML format, native XML databases are good candidates for the module. Currently, there are two popular open source XML database, one is by Apache Xindice [9], and the other is eXist [10]. Both are freely available. For personal usage, an XML database might not be necessary since the amount of data can be easily stored in plaintext files. Thus, the type of database depends on the amount of data and performance requirements. In the PEA system, a database agent is responsible for data storage and retrieval. Depending on the type of database, different database agent can be implemented and deployed at run time to accommodate different types of databases.

2.1.4 Mobile Module

The mobile agent acts as a liaison for PEA agents. Its sole purpose is to migrate to another container within the same JADE framework to share available information and knowledge relevant to a specific Web site. Before the mobile agent can access the knowledge base of other PEA agents, it needs to have adequate permission. This is somewhat analogous to entering a foreign country, where a valid passport and visa are required. Once the mobile agent has completed its task of gathering information, it clones itself and the clone reports the location where the requestor resides with the shared information. The mobile agent self-destructs after job completion.

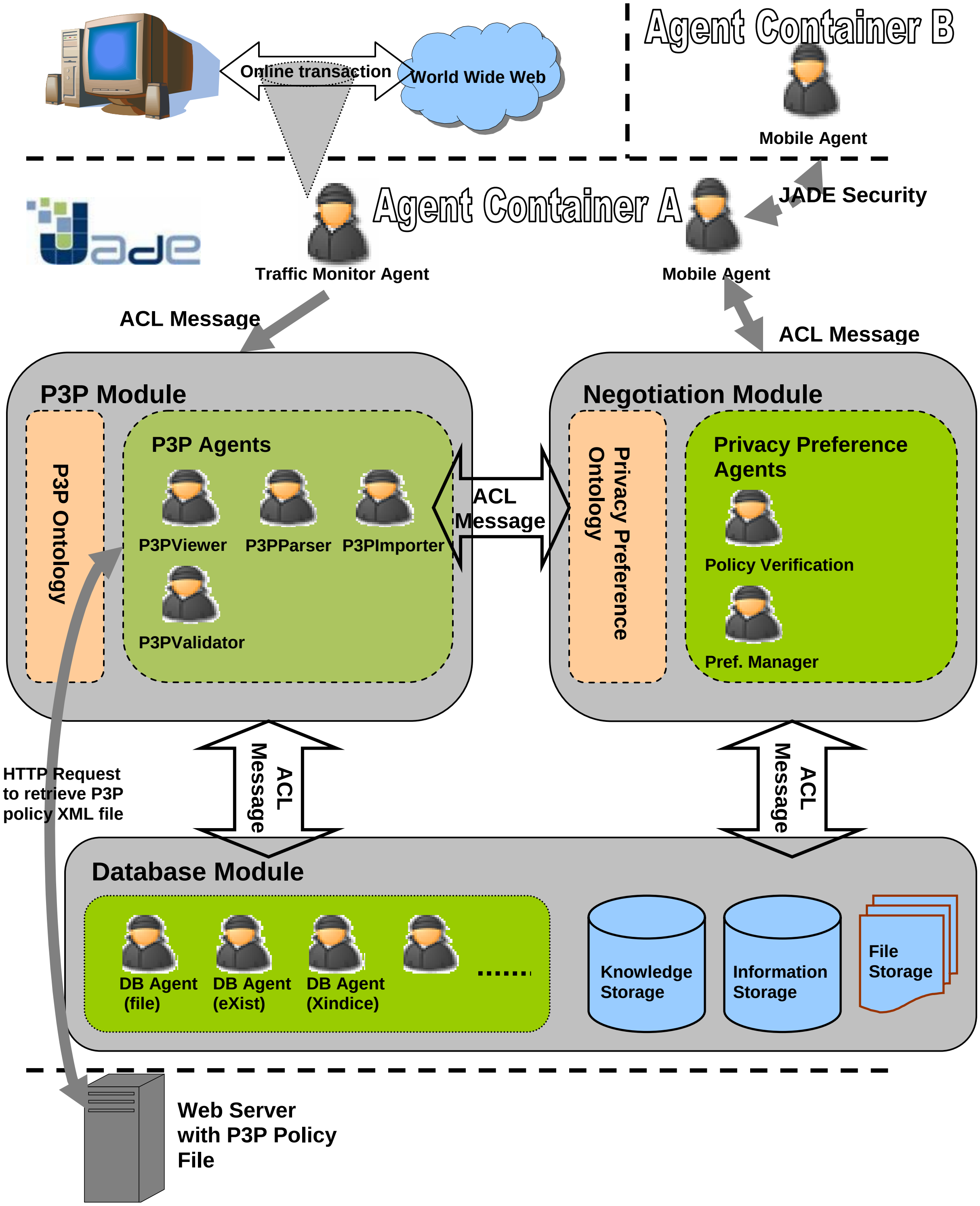


Figure 2-2 PEA Architecture Overview

2.1.5 GUI Module

The GUI agent is, in effect, the presentation layer of the PEA system. Any agent in the system can send requests to the GUI agent. The GIU agent acts as the bridge between the user and PEA.

3. Privacy Information Management and Transaction

3.1 Information Management

PEA manages a user's online privacy information. It accomplishes this feat by supporting the HTTP and P3P protocols. In the next few sections we describe how PEA handles P3P-related Web transactions and how it manages user's personal privacy information.

3.2 P3P Transaction

In this section, we describe the P3P transaction flow when a user requests a Web page and the PEA agent transparently handles the P3P proposal. First, the client request is intercept by the PEA agent. Then the Web server sends a reference of P3P policy along with the response from client request. The PEA agent checks the policy against the user’s privacy preferences and the user’s transaction history. If PEA is able to validate and verify the policy seamlessly without user intervention, it does so. Otherwise the evaluation engine must evaluate the proposal and resolve the conflicts, if possible [17].

The functional model in Figure 3-1 describes the flow of information between the stages of the transaction and illustrates the sequence of events and tasks that occur in the PEA system. Before going into the details of these events and tasks, we explain the overall flow using a simple transaction example.

3.2.1 A Simple Example

The following steps occur in a transaction involving the PEA agent (each task mentioned here is described in more detail in Section 3.2.2):

1. The transaction begins with the user requesting a Web page. The initial HTTP request is intercepted and checked for type by the PEA agent. Since this is an initial request, the type of the message is a “request.” The result is forwarded to the designated Web server. This is task 1, below.
2. The Web server responds the initial request with P3P preference information embedded within the HTTP response header.
3. Task 1 intercepts the HTTP response and invokes task 2.
4. Task 2 finds the P3P preference from the response.
5. Task 3 then extracts the P3P policy and delivers this information to the P3P agent for validation and parsing.
6. The P3P agent sends the successfully parsed P3P policy data to the negotiation agent.
7. The negotiation agent matches the policy against the user’s privacy preference rules. It references P3P transaction histories and consults with other agent via a mobile agent, if necessary.

8. If the negotiation agent successfully matches the Web site’s P3P policy with the user’s preference rules, then it invokes task 4 to forward the response to the Web browser.

The transaction occurs transparently between the user (via the initial Web page request) and the Web server response. The trust engine, as shown in Figure 3-1, acts on the user’s behalf and grants access rights to a particular set of privacy information based on the user’s privacy preference rules, privacy information transaction histories, and with the help of other agents, if necessary.

3.2.2 Generic Functional Model

The outline in the previous section shows how the PEA agent handles Web transactions on a user's behalf under normal circumstances, that is, assuming no errors occur and no user intervention is required. During such a transaction, the user is only involved in sending the initial Web page request. From the user’s perspective, this is the same as any other Web interaction.

We now take this one step further and describe the functional model (see Figure 3-1) in detail. This description covers all possible scenarios for a P3P transaction in PEA.

In Figure 3-1, a solid arrow denotes the flow of data from one unit (such as a task, Web server, Web browser, etc.) to another. A dotted arrow denotes a boolean control flow between units. The blue ovals represent PEA tasks.

Now we describe each individual task in the PEA functional model.

Task 1 - Intercept HTTP Stream

If an HTTP stream is detected, PEA must first determine if the HTTP stream is a request or response stream. Only response streams from a Web server are of interest. If a server response stream is found, contact task 2 to look for possible P3P information.

Task 2 - Check P3P Preference

Task 2 checks the response stream obtained from task 1 for P3P information. It searches the HTTP message for a P3P preference file or P3P policy. If P3P information is found, task 3 is invoked; otherwise, task 2 simply forwards the response stream to the Web browser and no further processing by PEA is required.

Task 3 – Extract P3P Policy Information

If task 2 finds P3P information, task 3 will try to locate the appropriate P3P policy files, based on the preference information retrieved by task 2. Once a policy file is located, it is passed to the P3P agent, which validates and parses the information.

Task 4 – Forward Server Response

If no P3P information can be retrieved, or the trust engine cannot resolve the P3P policy, or the user has explicitly accepted the policy, the original server response stream is forward to the Web browser unmodified.

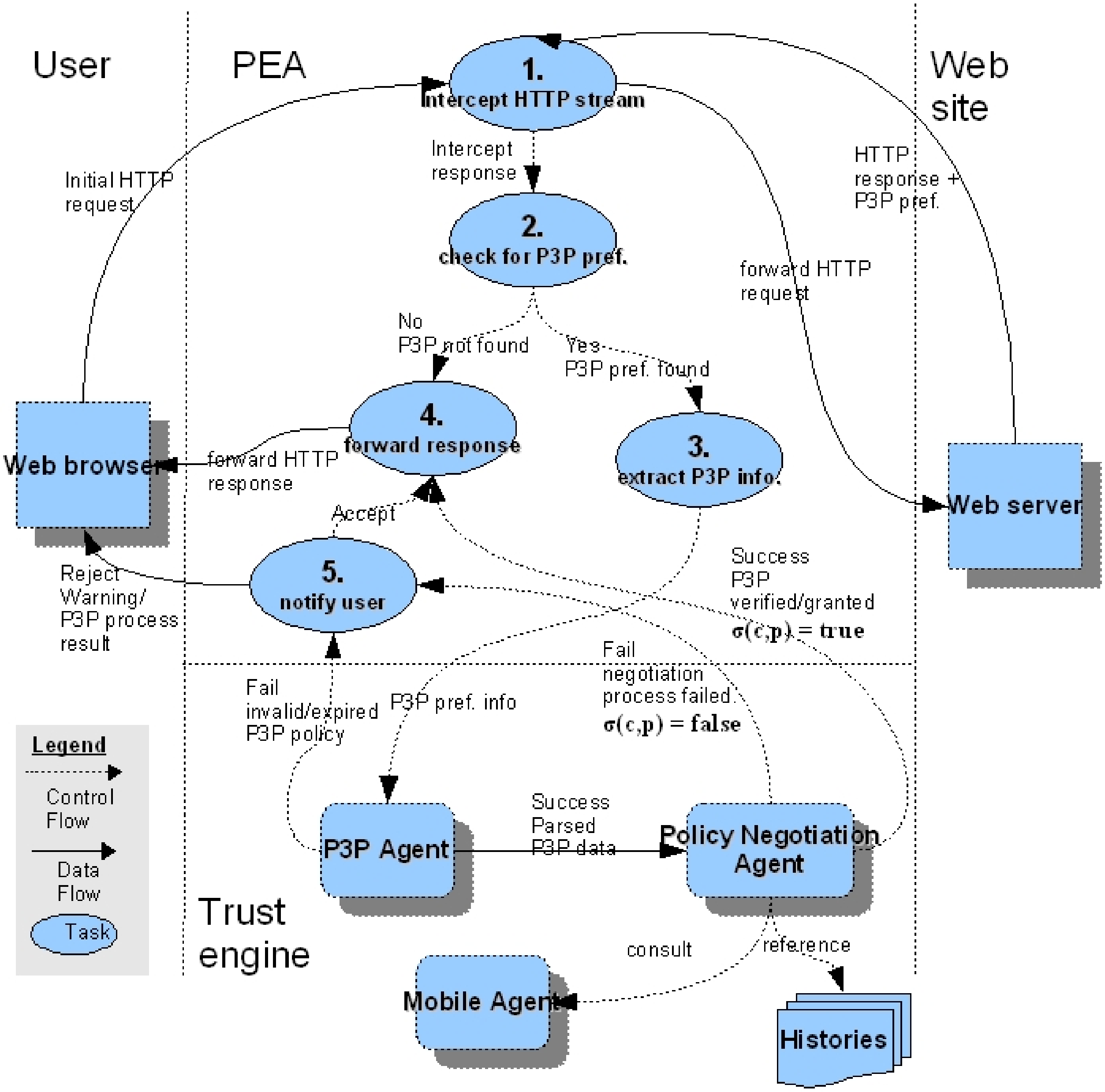


Figure 3-1: Functional Model

Task 5 – Notify User

If the PEA trust engine failed to validate the P3P information or the negotiation is inconclusive, the user is notified with warning information regarding the P3P policy. The user is provided the negotiation result for reference and the user can choose to either accept or reject the P3P policy.

4. Privacy Information Negotiation

An important function of PEA is its ability to negotiate based on a P3P privacy policy and the user's specified personal privacy rules. PEA can extract a privacy policy from a P3P preference file and automatically negotiate the preference using the user's privacy rules to try to reach an agreement that is satisfactory to the user. In this section, we will cover the ideas behind this negotiating process in PEA.

4.1 The Concept of Information Negotiation

In PEA, negotiations are based on the idea of defining constraints on information sets. The constraints are used to specify the conditions under which personal information can be accessed. A P3P privacy preference policy contains a request to gain access to a particular set of information. PEA checks the request and verifies whether the request satisfies all of the user-defined constraints. If the constraints are satisfied, access to the information is granted. Otherwise, the request is rejected or a negotiation process can try to find an alternative solution based on additional historical information or user input. The goal is to try to reach an agreement that is satisfactory to the user without jeopardizing his or her privacy information and, in the process, to be as transparent as possible [19].

4.1.1 Basic Terminology (Information, Rules and Facts)

Now we introduce the basic terms, information, rules and facts used by PEA in the negotiating process. These terms will be used to define algorithms and additional concepts related to negotiating sets of information. First, we need to define personal information more precisely.

Definition 1: Personal information consists of a set

$\mathbf{I} = \{d_1,d_2,d_3,...,d_n\}$

where **I** is a finite set of personal information and each d_i, for $1 \leq i \leq n$, is a data element. All data elements of a user's personal information are contained in **I**.

Definition 2: A *rule* r is used to define the specific circumstances under which a set of information can be accessed. A rule is of the form

$r = (D_r, C_r)$ where $D_r \subseteq I$ and $C_r = \{c_1,c_2,.....,c_n\}$

where r is a pair (D_r,C_r) and D_r denotes a set of personal information from **I**, and C_r denotes a set of constraints on D_r. Each constraint $\mathbf{c_i}$, for $1 \leq i \leq n$, represents a particular condition which must be met in order to satisfy rule r.

Definition 3: *Facts* f are used to describe a request for a set of data and the conditions under which the request is made. Then

$f = (D_f, V_f)$ where $D_f \neq \emptyset$ and $V_f = \{p_1,p_2,.....,p_n\}$

where a fact f is defined by a pair (D_f, V_f) where D_f denotes the requested data and V_f denotes the conditions under which the data D_f is requested. The conditions specified by V_f contain name-value pairs of the form $p_i = (n_i,v_i)$, for $1 \leq i \leq n$, where n_i denotes the name of an item and v_i denotes the value of the item.

Definition 4: The *collection of facts* relevant to a particular request is denoted

$$F = \{f_1,f_2,...,f_n\}$$

where each f_i, for $1 \leq i \leq n$, is a fact, as defined in Definition 3.

Definition 5: A *rule set* is denoted

$$R = \{r_1,r_2,...,r_n\}$$

where each r_i, for $1 \leq i \leq n$, is a rule.

4.1.2 Tree Representation of a Ruleset

According to Definition 2, a rule is a pair that defines constraints on a set of data. To represent a rule set, as per Definition 5, we will use a tree structure, which will be very useful when we evaluate rules and match facts with rules.

We begin with a simple rule set, say, R_x. Suppose the access to three different sets of information is controlled by three rules of the form

$$r_1 = (D_{r1},\{c_1,c_2\})$$
$$r_2 = (D_{r2},\{c_1,c_3\})$$
$$r_3 = (D_{r3},\{c_4,c_5\})$$
$$R_x = \{ r_1 , r_2 , r_3 \}$$

Note that both r_1 and r_2 contain the same constraint c_1. The rule set tree for R_x is given in Figure 4-1.

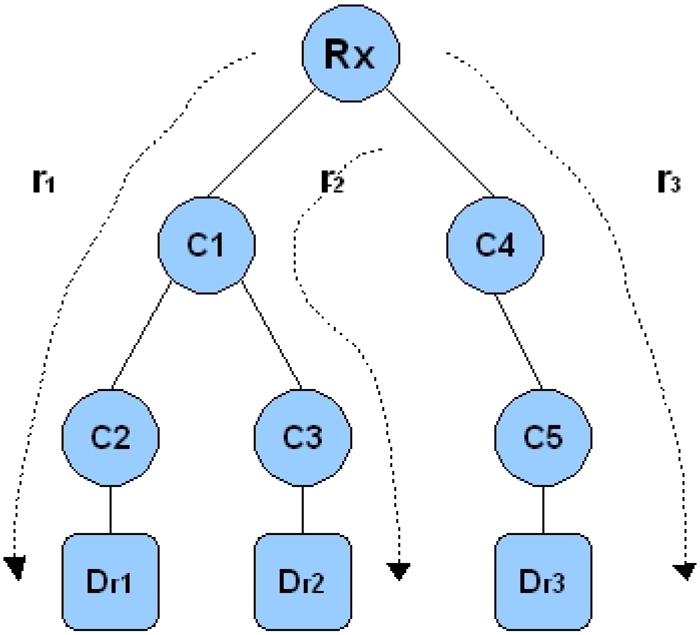


Figure 4-1: Ruleset Tree

Each rule is represented by a path from the root node R_x to the leaf node which represents the data element D_{ri}, for i = 1,2,3. With this tree representation, we can easy see the constraints that lie between the root node of the rule set and the data elements, which are the leaf nodes.

4.1.3 Rule Evaluation

Rule evaluation is the process of matching facts extracted from requested information against rules. Access will not be granted to requested information unless the facts appropriately match the rules. Of course, the rules have been defined to guard the user's PII data. For a fact to match a rule, the fact must first satisfy all the constraints (see Definition 6, below) and second the requested information in the fact must be a subset of the information specified in the rule (see Definition 7, below). In other words, facts match a rule when all the constraints of the rule are satisfied and the requested data specified in the facts are covered by the rule (see Definition 2, above).

Definition 6: Let σ be the Boolean function for the constraint p $\in$ V_f (see Definition 3) matches a specific constraint set c $\in$ C_r (see Definition 2) of rule r. Then

$$\sigma(c,p) = \begin{cases} \text{true if p satisfy c} \\ \text{false otherwise} \end{cases}$$

Definition 7: The given facts f = ($D_{f,}$ V_f) match the rule r = (D_r, C_r), if

(i) $\sigma(c_i,p_j)$ = true if and only if for all $c_i \in C_r$ there exists $p_j \in V_f$

(ii) $D_f \subseteq D_r$ if and only if for all $d_k \in D_f$ there exists $d_k \in D_r$.

4.1.4 Sample Rule Evaluation

Now we describe the rule evaluation process in a simple scenario. Suppose John and Mary have two bank accounts, denoted as saving and checking, and the following rules apply to access of these accounts:

r_1 = ({Saving Account}, {(Owner=Mary),(Action=Deposit),(Action=Transfer)})

r_2 = ({Checking Account},{(Owner=John),{(Owner=Mary),(Action=Withdraw)})

The following snapshot of the privacy preference rule XML definition corresponds to the rules above (see Section 5.3 for detail information of the Privacy Preference XML schema):

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="Bank_A_Saving">
      <DATA ref="#user.bankaccount.saving_account"/>
      <RULE mandatory="yes">
         <ACCESS><Deposit/><Transfer/></ACCESS>
         <RECIPIENT><Mary/></RECIPIENT>
      </RULE>
   </PREFERENCE>
   <PREFERENCE name="Bank_A_Checking">
      <DATA ref="#user.bankaccount.checking_account"/>
      <RULE mandatory="yes">
         <ACCESS><Withdraw/></ACCESS>
         <RECIPIENT><Mary/><John/></RECIPIENT>
      </RULE>
   </PREFERENCE>
</PREFERENCESET>
```

These rules specify that only John can access the saving account, since it is an individual account owned by John. As for the checking account, both John and Mary can access it since they both are the owner and it is a joint account.

Now suppose that Mary wants to withdraw from the saving account. The facts of her request are

f_1 = ({Saving Account},{(Owner=Mary),(Action=Withdraw)})

The facts of her request do not match r_1 or r_2. Therefore, her request cannot be granted. Suppose Mary then issue another request with the facts

f_2 = ({Checking Account},{(Owner=Mary),(Action=Withdraw)})

This withdrawal request is granted, because the facts of the request match r_2, as indicated by the shaded area in Figure 4-2.

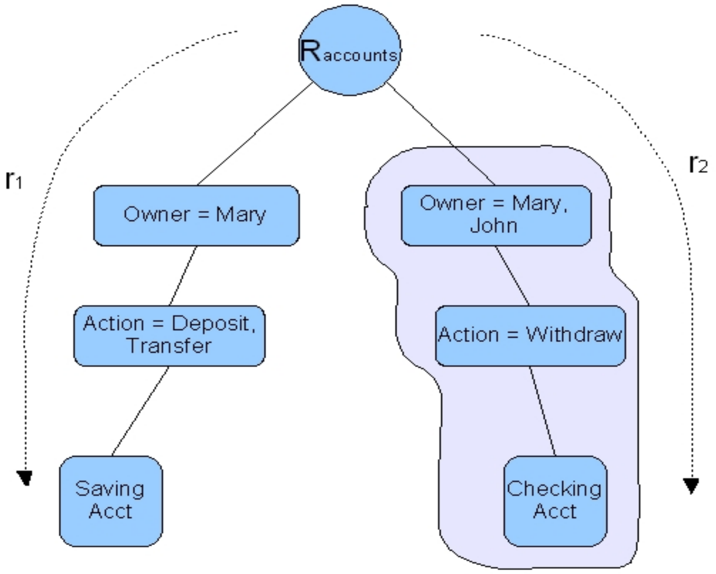


Figure 4-2: Facts f2 matches rule r2

5. Privacy Policy XML Schematic and Ontologies

5.1 P3P Policy Schematic

A P3P policy consists of the following five elements: Policy element, Entity element, Access Element, Disputes Element and Statement Element. In this section, we will briefly describe these five elements in general terms. For complete XML schematic information, please refer to Appendix B: P3P Policy Schematic.

- **Policy Element** - a policy element contains a single policy the name of which must be unique throughout the Web site policies. The Policy

Element is the container for Entity, Access, Disputes and Statement elements.

- **Entity Element** - gives a precise description of the legal entity making the representation of the privacy practices.
- **Access Element** - indicates whether the site provides access to various kind of information.
- **Disputes Element** - describes dispute resolution procedures that should be followed for disputes regarding privacy practices, or in case of protocol violation. Each disputes element should contain one or more remedies that specify the possible remedies in case a policy breach occurs.
- **Statement Element** – is a statement that concerns the data practices as applied to Data Elements. A Statement Element is a container that groups together a <PURPOSE>, a <RECIPIENT>, a <RETENTION>, a <DATA-GROUP>, and optionally a <CONSEQUENCE>.

5.2 P3P Ontology Model

The P3P ontology translates from the P3P XML policy according to the P3P 1.0 specification. Base on the definition of P3P 1.0, the PEA system creates an ontological model as shown in Figure 5-1, which defines the relationship between concepts.

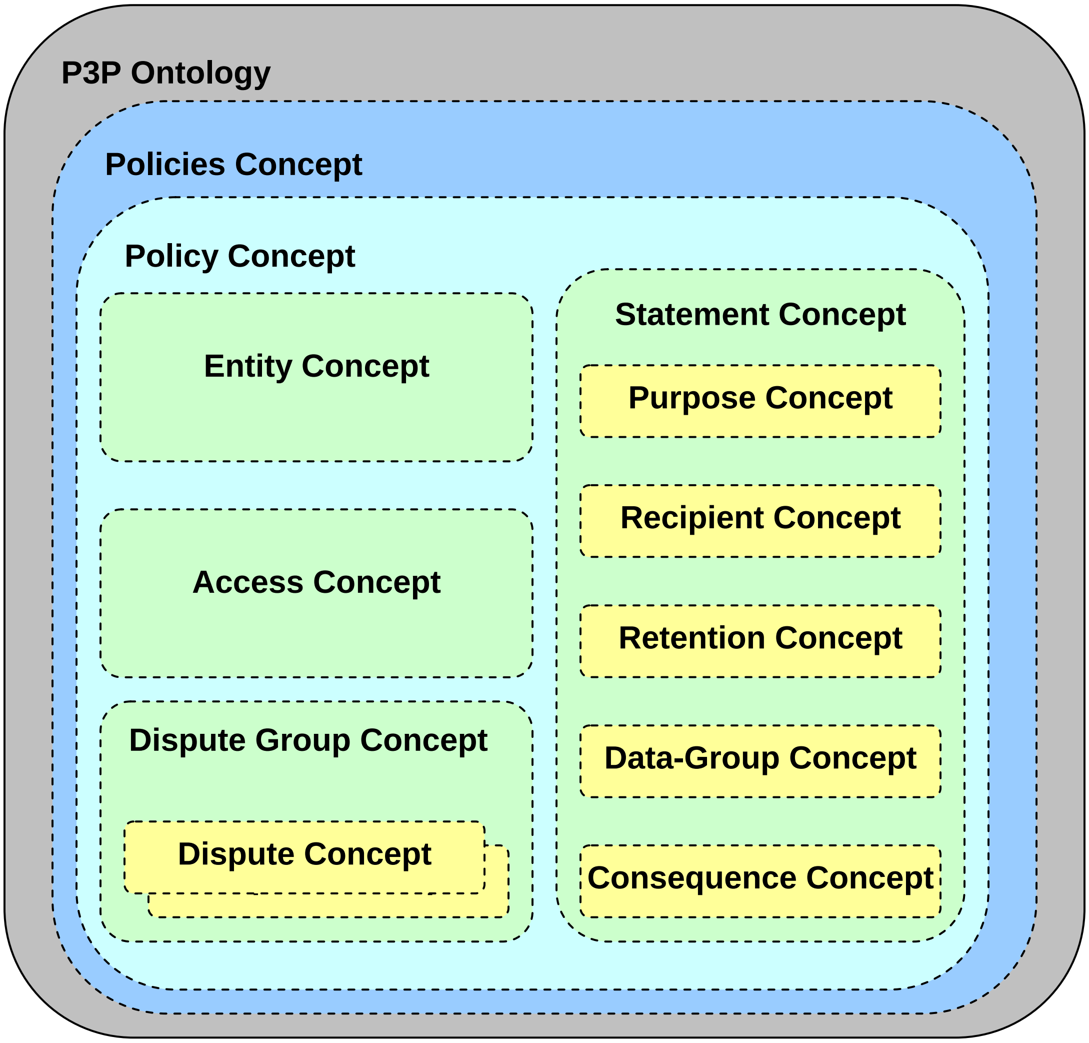


Figure 5-1: P3P Ontology Model

The ontology model is based on the ontology framework of the JADE framework. The detailed implementation of the ontology model is covered in Section 6.1 Content Language and Ontologies, below.

5.3 PEA Privacy Preference Schematic

The PEA privacy preference defines how a piece of data can be accessed based on rule and constraint definitions. The rules and constraints consist of a set of tokens defined by the P3P specification. For example, in P3P

```
                          <ACCESS>
                             <noident/>
                          </ACCESS>
```

means that the site does not collect PII. The same information in a PEA privacy preference means that the data within the preference definition can be accessed if and only if the merchant does not collect or store the information. The schema of PEA Privacy Preference inherits from P3P 1.0 XML schema. It reuses the Access, Recipient, Purpose and Retention elements for rule and access constraint definition. It also adds a history element to record the transaction history of the specific privacy preference.

Next, we will describe the PEA Privacy Preference XML schema in detail. Here, we exclude the Access, Recipient, Purpose and Retention elements, which we borrow from the P3P XML schema. For a detailed definition of Access element, refer to B.3 Access Element; for a detailed definition of Recipient, Purpose and Retention elements, refer to B.5 Statement Element.

- o **PREFERENCESET element** – is the container element which groups a collection of preferences.

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="default">
      ……………
   </PREFERENCE>
</PREFERENCESET>
```

  - ▪ **ownerid**: unique owner identifier

- o **PREFERENCE element** – describes the privacy preference for an entity whose entity name is the value of the attribute. The preference element consists of Data, Rule and History elements, which define the single rule that is applied to the enclosed data and the transaction history of this privacy preference.

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="default">
      <DATA ref="#user.home-info.telecom.telephone"/>
      <RULE mandatory="yes">
         <ACCESS><noident/></ACCESS>
         <RECIPIENT><same/></RECIPIENT>
         <PURPOSE><current/><individual-analysis/></PURPOSE>
         <RETENTION><stated-purpose/></RETENTION>
      </RULE>
      <HISTORY>
         ………
      </HISTORY>
   </PREFERENCE>
</PREFERENCESET>
```

- **name**: This is the entity/merchant name that the privacy preference is specifically applied to. This attribute is an optional attribute. If omitted or if the value is “default”, the preference is a general privacy preference.

o **DATA element** – defines the type of privacy data that the Rule element applies to. A Preference element can contain multiple DATA elements and the same RULE is applied to all included data.

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="deaful">
      <DATA ref="#user.home-info.telecom.telephone"/>
      <DATA ref="#user.bdate"/>
      <RULE mandatory="yes">
        ………
      </RULE>
      <HISTORY>
         ………
      </HISTORY>
   </PREFERENCE>
</PREFERENCESET>
```

- **ref** (required): This is a URI reference to the corresponding data schema. The data schema is defined in the P3P 1.0 specification. The PEA privacy preference follows the same data schema as P3P 1.0

        specification with respect to matching the type of data collected by merchants’ P3P policies. The URI is the base URI of the data group. The names of data elements and sets are case-sensitive. For detail information of the base data schema, please refer to Appendix D: P3P Base Data Schema.

- **RULE element** – groups data collection purpose, data recipient, information access type and retention policies together. The rule element defines the privacy rule for the PII which is defined in the DATA element within the same preference node.

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="deafult">
      <DATA ref="#user.home-info.telecom.telephone"/>
      <RULE mandatory="yes">
         <ACCESS><noident/></ACCESS>
         <RECIPIENT><same/></RECIPIENT>
         <PURPOSE><current/><individual-analysis/></PURPOSE>
         <RETENTION><stated-purpose/></RETENTION>
      </RULE>
      <HISTORY>
         ………
      </HISTORY>
   </PREFERENCE>
</PREFERENCESET>
```

    - **mandatory** (required): If this is ‘yes’, the rule is a mandatory rule. That is, all elements within the rule node need to be 100% satisfied in order to be granted the access right. If this flag is ‘no’, the user will be given the option to either accept or deny the access based on the information provided by PEA agents. The default value is ‘no’.

- **HISTORY element** – records the transaction history with regarding to the specific privacy preference.

```
<PREFERENCESET ownerId="John Doe">
   <PREFERENCE name="deafult">
      <ACCESS><noident/></ACCESS>
      <RECIPIENT><same/></RECIPIENT>
      <STATEMENT>
         ………
      </STATEMENT>
      <HISTORY createTime="1163708276072">
         <lastAccess></lastAccess>
         <lastViolation></lastViolation>
         <accessed></accessed>
         <violated></violated>
         <remoteAccess><remoteAccess>
         <remoteUserName></remoteUserName>
      </HISTORY>
   </PREFERENCE>
</PREFERENCESET>
```

- **createTime** (required)**:** The timestamp when the preference was created, in milliseconds.
- **lastAccess:** The time stamp when the last access occurred, in milliseconds.
- **lastViolation:** The time stamp when the last privacy policy violation occurred, in milliseconds.
- **accessed:** Total number of times that this privacy preference has been accessed.
- **violated:** Total number of time that this privacy rule has been violated. This number can be useful when the privacy preference is for a specific merchant whose name is the value of the 'name' attribute of PREFERENCE element.
- **remoteAccess:** Total number of times the privacy preference has been referenced remotely by other PEA agents. Remote access is not granted to "default" privacy preference, but instead to merchant specific privacy preferences.

- **remoteUserName:** The owner's name of the mobile agent which last accessed the privacy preference. This element is only available for merchant-specific privacy preferences.

5.4 PEA Privacy Preference Ontology Model

The PEA privacy preference ontology translates PEA privacy preferences into a language that a PEA agent can understand and communicate to local and remote agents. Based on the definition of the PEA privacy preference schema, the PEA system creates an ontological model as shown in Figure 5-2. The purpose of this model is to define the relationship between PEA privacy ontology concepts.

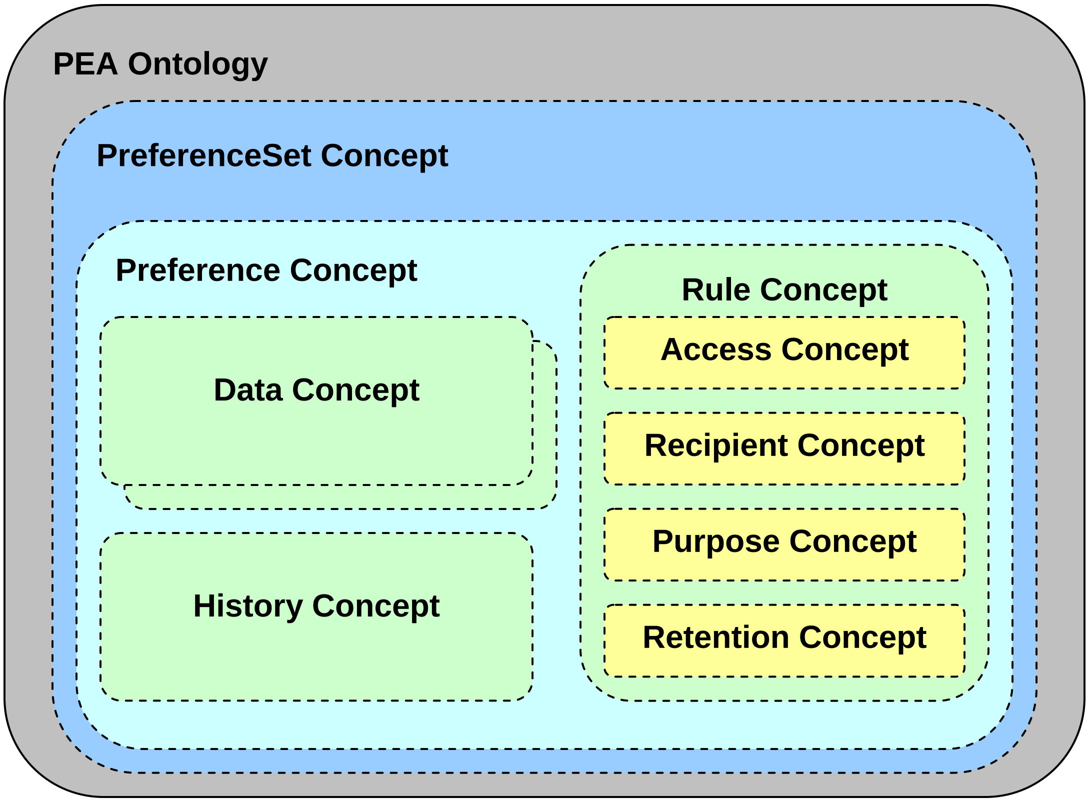


Figure 5-2: PEA Ontology Model

The PEA ontology model is defined based on the ontology features of the JADE framework. The detail implementation details of the ontology model are covered in Section 6.1 Content Language and Ontologies.

6. Implementation of PEA

In this chapter, we will describe how PEA agents handle their tasks and the core behind the PEA system. We first will go through the implementation of ontologies in detail as defined in the previous chapters. Then, the interaction sequence between PEA agents will be covered to demonstrate the role of each agent in enhancing privacy protection.

6.1 Content Language and Ontologies

PEA agent communication is based on the content language codec and ontologies with the support of JADE framework. As shown in Figure 6-1, the content exchange between agents is performed by JADE automatically [13]. For example, agent A needs to send a piece of information to agent B.

1. Agent A needs to convert the internal representation of information $\boldsymbol{I}$ into ACL content expression representation and agent B needs to perform the opposite conversion.
2. Agent B needs to perform a number of semantic checks to verify that $\boldsymbol{I}$ is a valid piece of information according to the ontology rules which both agent A and B should both know and agree.

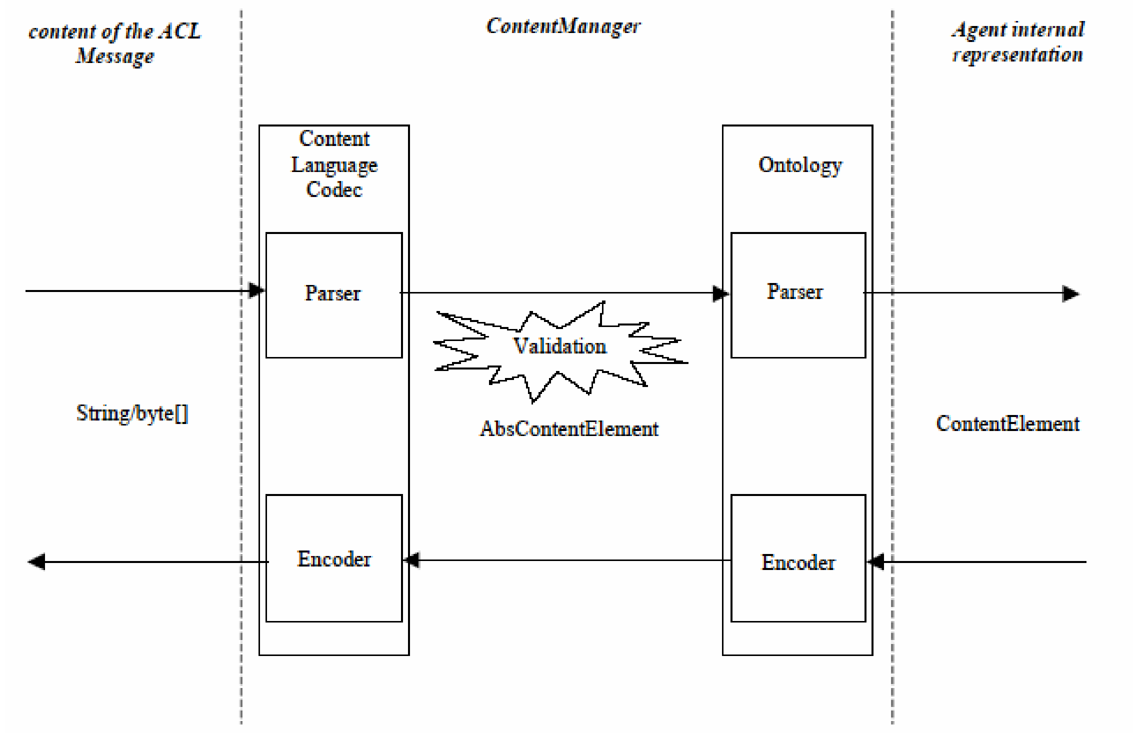


Figure 6-1: conversion pipeline of Agent communication

Tilab, "JADE Tutorial – Application-Defined Content Language and Ontologies". Nov 2004.

< http://jade.tilab.com/doc/CLOntoSupport.pdf>

For more information regarding JADE content language codec and ontologies, please refer to JADE document on creating and using applications-specific ontologies [6].

6.1.1 How to create an Ontology object

P3P policy and PEA privacy preference are both defined in XML format. In order to translate the XML files into the ontologies which can be understood by PEA agents, Apache XMLBeans [11] is used to ease the access of XML information by binding it to Java Types. The java types of P3P and PEA ontology concept objects are compiled and generated through XML schema to represent the schema types, which enable accessing XML document at all levels without having to resort to parsing through the XML documents to retrieve desire data and eliminate the need for multiple tools to accomplish the goal. Both XML schemas for P3P and PEA privacy preference are available at Appendix A: Platform for Privacy Preference (P3P) XML Schema and Appendix C: PEA Privacy Preference XML Schema.

After the XML files are complied into java-bean style objects, PEA maps each concept object into corresponding java-beans. The concept objects are then enclosed into ontology objects by invoking addConecpt(concept) function of the ontology class(es). As shown in Figure 6-2, the steps of XML data to ontology object conversion.

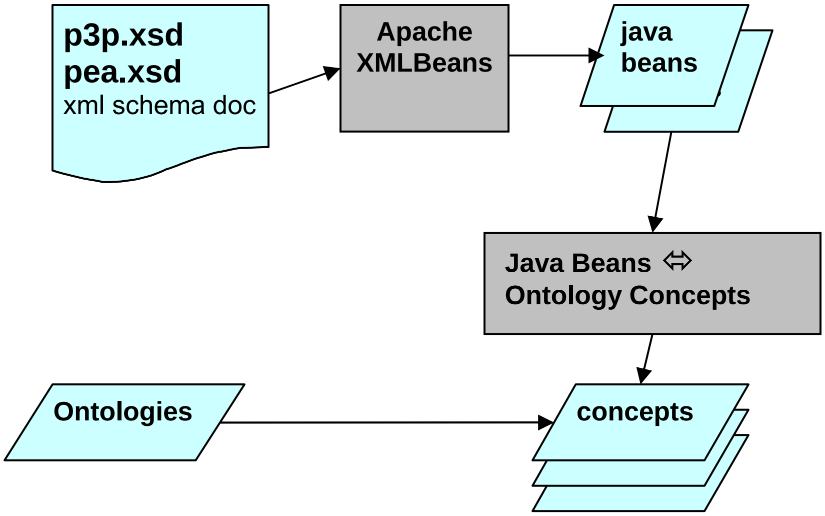


Figure 6-2: The conversion from XML schema to Ontology objects

6.1.2 Snapshot of Ontology Class Diagram

Within PEA system, we are dealing with considerable large ontologies which are obtained by extending ontologies from P3P and PEA privacy preference ontology. To keep track of each ontology, one or more vocabularies need to be mapped. Or the ontologies can become unmanageable as the ontology structure grows. To solve the issue, we use a simple design as presented in Figure 6-3. All vocabulary constants are defined in the vocabulary interfaces. With this pattern, each vocabulary constant can be accessed as if it were defined in the parent ontology, PEAOntology.

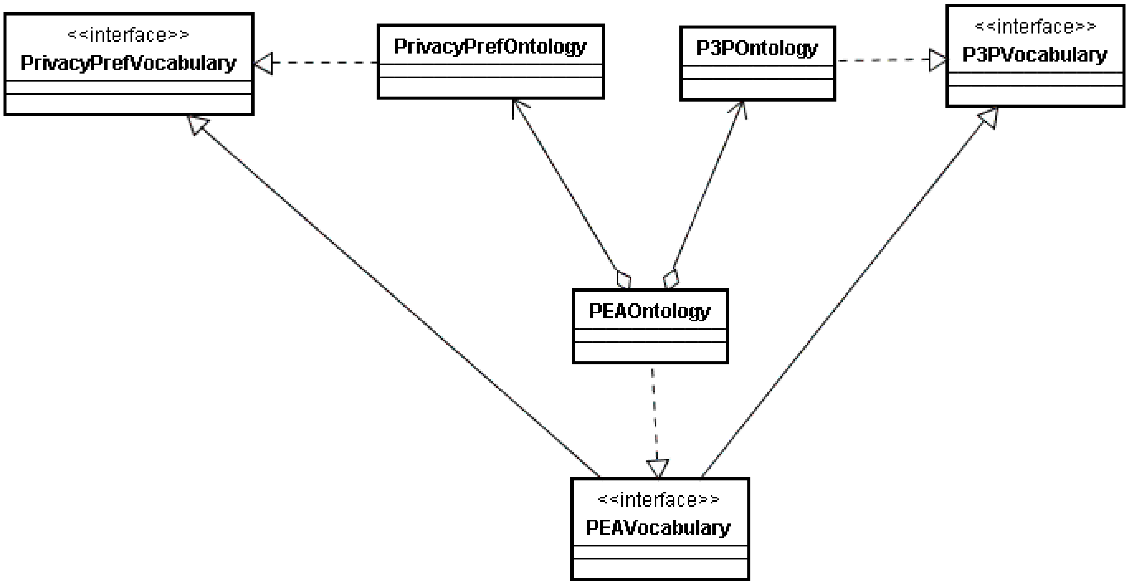


Figure 6-3: Basic Ontology-Vocabulary Class Diagram

P3PVocabulary is composed with several other vocabularies. The purpose of doing the division is to better group the vocabularies into specific categories. As shown in Figure 6-4, P3PVocabulary groups of the following vocabularies.

- **P3PPolicyVocabulary:** all vocabularies related to the P3P policy are defined in this vocabulary interface.

- **P3PAccessVocabulary:** all vocabularies related to privacy information access policy are defined in this vocabulary interface.
- **P3PEntityVocabulary:** merchant entity information vocabularies are defined in this interface.
- **P3PDisputeVocabulary:** dispute policy vocabularies are defined in this interface.

This simplifies the program design by allowing ontologies to extend from one single interface. If an ontology needs to use a specific vocabulary rather than a general vocabulary, it can simply extend from any vocabulary interface.

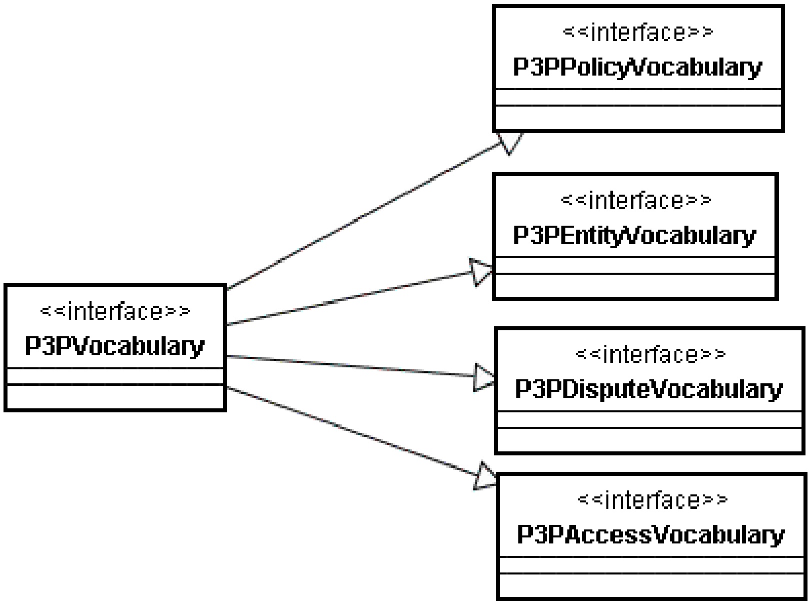


Figure 6-4: P3P Vocabulary Grouping

6.1.3 Ontological Elements

This section defines the slot for all ontological elements in PEA privacy preference and P3P ontology.

P3P Policies:

Slot Name	Type	Values classes	Cardinality
fileName	String		1
url	String		1
xmlString	String		1

P3P Policy:

Slot Name	Type	Values classes	Cardinality
opturi	String		1
discuri	String		1
name	String		1
lang	String		1
entity	Class	Entity	1
disputeGroup	Class	DisputeGroup	1
access	Class	Access	1
statement	Class	Statement	1

P3P Entity:

Slot Name	Type	Values classes	Cardinality
name	String		1
postalAddress	String		1
postalCity	String		1
postalStateProv	String		1
postalCode	String		1
postalCountry	String		1
email	String		1
telephoneIntCode	String		1
telephoneLocCode	String		1
telephneNumber	String		1

P3P Access:

Slot Name	Type	Values classes	Cardinality
access	Class	Access	1

P3P Dispute Group:

Slot Name	Type	Values	Cardinality

		classes	
dispute	Class	Dispute	1:*

P3P Dispute:

Slot Name	Type	Values classes	Cardinality
resolutionType	String		1
service	String		1
verification	String		1
shortDesc	String		1
remedies	Class	Remedy	0:*

P3P Statement:

Slot Name	Type	Values classes	Cardinality
consequence	String		1
non-identifiable	Boolean		0:1
purpose	Class	Purpose	1:*
recipient	Class	Recipient	1:*
retention	Class	Retention	1
dataGroup	Class	DataGroup	1

PEA Preference Set:

Slot Name	Type	Values classes	Cardinality
preference	Class	Preference	1:*
ownerID	String		1

PEA Preference:

Slot Name	Type	Values classes	Cardinality
data	Class	Data	1:*
rule	Class	Rule	1
history	Class	History	1
name	String		1

PEA History:

Slot Name	Type	Values classes	Cardinality

lastAccess	DateTime		1
lastViolation	DateTime		1
accessed	Integer		1
violated	Integer		1
createTime	DateTime		1
remoteAccess	Integer		1
remoteUserName	String		1

PEA Rule:

Slot Name	Type	Values classes	Cardinality
purpose	Class	Purpose	1:*
access	Class	Access	1
recipient	Class	Recipient	1:*
retention	Class	Retention	1
mandatory	Boolean		0:1

Remedy:

Slot Name	Type	Values classes	Cardinality
remedy	String		1

Purpose:

Slot Name	Type	Values classes	Cardinality
purpose	String		1

Purpose:

Slot Name	Type	Values classes	Cardinality
purpose	String		1

Access:

Slot Name	Type	Values classes	Cardinality
access	String		1

Retention:

Slot Name	Type	Values	Cardinality

		classes	
retention	String		1

DataGroup:

Slot Name	**Type**	**Values classes**	**Cardinality**
data	Data		1:*

Data:

Slot Name	**Type**	**Values classes**	**Cardinality**
ref	String		1
optional	Boolean		0:1

Problem:

Slot Name	**Type**	**Values classes**	**Cardinality**
num	int		1
msg	String		1

6.2 Agent Interaction

PEA is built on an agent oriented framework, JADE. Different task is handled by a different agent or group of agents. Within the system, agents know each other by their registered name and service. One central DF (Directory Facilitator) agent maintains a list of available services and provider agent information which helps other agents to locate available agent information for a specific service from the DF agent. In the following sections, we will discuss how agents interact with each other to accomplish a given task.

6.2.1 P3P Policy Detection and Process

As shown in Figure 6-5, policy recovery and validation starts with the traffic monitor agent detecting online transaction carrying P3P policy information. The traffic monitor agent retrieves the URL/URI information of the P3P file and sends the information to the P3P importer agent. The importer agent uses the information to retrieve the physical file and hands over the file to the validation agent. The validation agent then confirms whether the P3P policy XML file complies with P3P XML schema. Once the policy file is validated, the validation agent sends a request to directory facilitator agent asking for parsing service agent information. After locating a parser agent, the file is handed over to

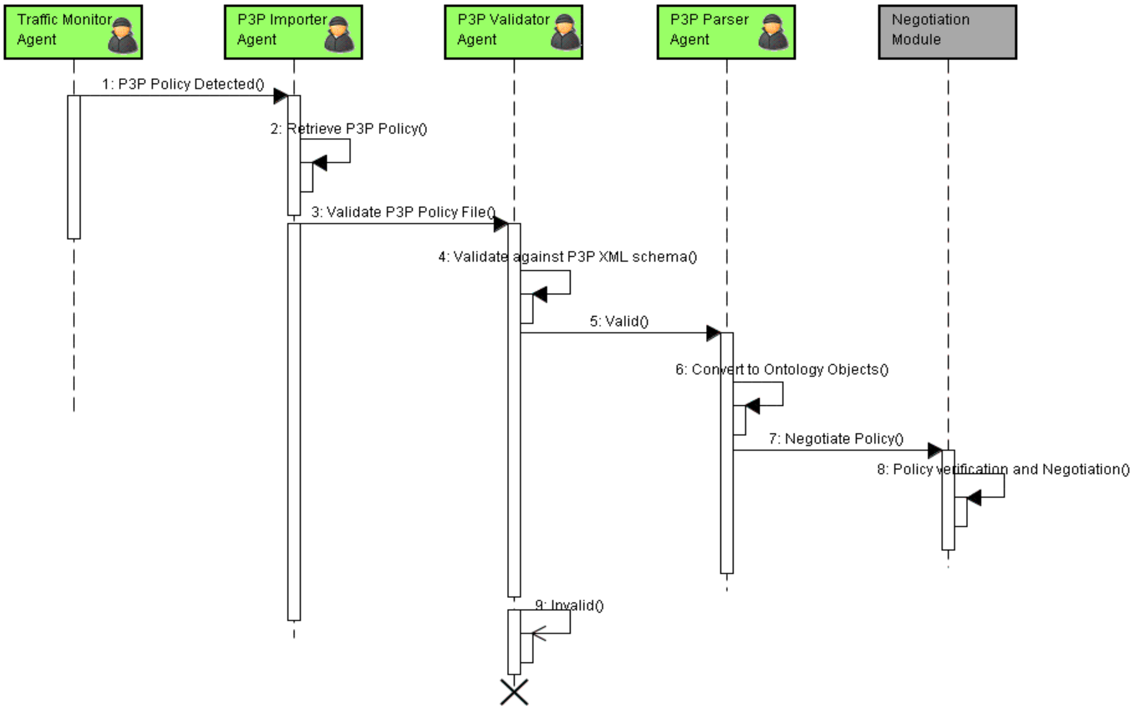


Figure 6-5: Sequence Diagram of Policy Detection and Process

the parser agent to be converted to P3P Ontology objects (see Section 5.2). The ontology objects will then be sent to negotiation module to be verified.

6.2.2 Policy Verification and Negotiation

Policy verification and negotiation process is to verify a P3P policy received from the P3P module against personal privacy preferences. During the privacy preference retrieving process, a default preference can be used only if there is no matching privacy preference for the Web site. In such case, besides notifying user and requesting for user intervention, an assist request will be sent to other agent containers and one or more mobile agents will migrate from other PEA agent containers along with the privacy policy knowledge of the Web site if available. The gathered information will then be supplied to the user via GUI agent to aid the user in decision making. There will be no mobile agent involvement, if a Web site specific privacy preference is violated. A warning will be sent to the user right away via the GUI agent request for user intervention. Figure 6-6 below illustrates the agent interaction sequence of the process.

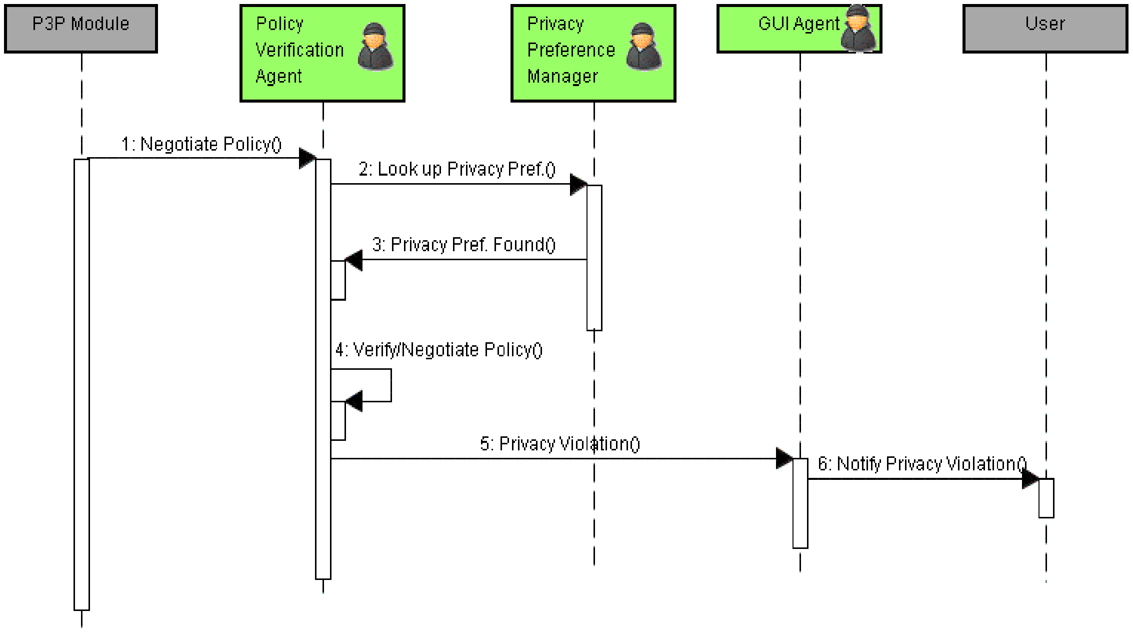

P3P Module
Policy Verification Agent
Privacy Preference Manager
GUI Agent
User
1: Negotiate Policy()
2: Look up Privacy Pref.()
3: Privacy Pref. Found()
4: Verify/Negotiate Policy()
5: Privacy Violation()
6: Notify Privacy Violation()

Figure 6-6: Sequence Diagram of Policy Verification and Negotiation

6.2.3 Knowledge Sharing

Collective knowledge is how PEA comes up with a suggestion when there is a privacy violation detected. To achieve this, PEA relies on the mobile ability of JADE agents. However, as of JADE version 3.4, JADE only supports intra-platform mobility. An agent can only move within the same platform from container to container [7].

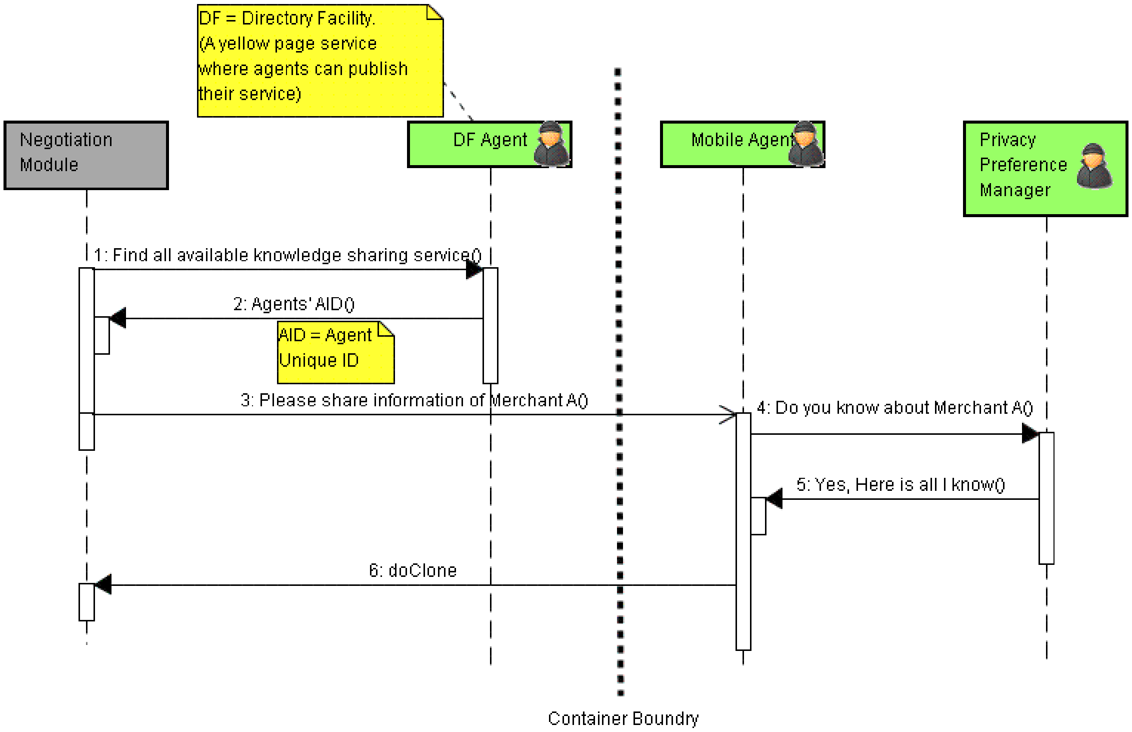


Figure 6-7: Sequence Diagram of Knowledge Sharing

The process of knowledge sharing is shown in Figure 6-7. When the negotiation module doesn't know about a specific Web site and its P3P policy violates the default personal privacy preference setting, it will request relevant information from other PEA container(s). The first step is to consult with the DF (Directory Facility) agent, where all agents publish their services, to locate all mobile agents which can provide knowledge sharing service. Once all the agents are located, a request is sent to all the agents with information of the P3P policy. The mobile agents then ask their privacy preference managers for any relevant information. If a match is found, the mobile agent creates a clone. The clone will migrate to the requestor location with any sharable information. Once the knowledge sharing process is completed, the clone self-destructs without returning back to its owner.

7. Conclusion

With the rapid growth of the Internet, the use of personal information in online transaction is of ever increasing concern. With the obvious benefits of modern information technology come the important responsibilities to protect user privacy rights. The Platform for Privacy Preferences Project (P3P), a project maintained by the World Wide Web Consortium (W3C), is a protocol suite designed to standardize Website privacy policy formats so that the policies can be retrieved and intercepted easily by software user agents. A P3P user agent can inform the user of the privacy practices of a Web site and automate the decision-making process on the user's behalf, based on privacy practices and preferences.

In this paper, we have described a Privacy Enhancing Agent (PEA), whose purpose is to reduce the burden on Web users with respect to privacy issues, while satisfying the P3P requirements. In the vast majority of cases, with PEA there is no need for the user to read the privacy policies of a Web site during a Web transaction, since PEA will automatically retrieve, evaluate and respond to privacy policies in the context of the user’s privacy preferences. PEA maintains a privacy-related transaction history, which improves its decision-making ability. PEA agents can also share knowledge with each other across the network, which further strengthens an agent’s analytical potential.

By using PEA, users will be made aware of possible privacy threats, without being overwhelmed by routine privacy interactions. In effect, PEA greatly reduces the potential for information overload on the Web by representing the user during online transactions. Although PEA cannot prevent a Web merchant from violating its own privacy policies, it can assist users by providing information with regarding to the stated privacy practices of any P3P compliant Web site.

We believe that PEA can be a valuable asset to users. PEA will enable a user to manage, negotiate and analyze personal privacy information on the World Wide Web and, in this way, PEA can make the Web safer and more secure with regard to the management of personal private information.

8. Future Work

There are several possibilities for future work regarding the PEA, such as the limitation of agents’ communication and security issues. In addition, PEA can improve its negotiation capability with server-side participation. With respect to Implementation, the process of converting from XML schema to PEA related ontology objects is tedious and time consuming.

PEA agents' communication is limited to agent containers within one JADE agent platform because of JADE framework limitation. JADE development team is currently working on cross-framework communication feature. Once the feature is available, PEA system would be able to extend its ability over the boundary of single JADE framework, which would further broaden the privacy agent community. The larger the agent community grows the better and more accurate the knowledge of Web site privacy practice could be.

PEA is lacking of the capability to protect user from internet phisher. A role-based security system needs to be in place to allow only users with adequate security clearance to be able to access sensitive personal information. It will be beneficial to integrate some role-based access control language such as SAML and XACML into PEA system to further protect user's personal information from potential harmful PEA agents.

To enhance PEA negotiation capability to achieve 2-way negotiation (browser and Web server), PEA needs to be able to exchange messages with the Web page. P3P is working on Preference Exchange Language (APPEL 1.0) which can be used by P3P user agents to make automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled Web sites. To do so, the Web sites needs to be capable of interacting with the agents. This requires a substantial amount of effort to build up the environment and since most of the Web servers do not support P3P currently, let alone supporting APPEL.

The effort of converting P3P and PEA XML schemas to corresponding ontology objects is non-trivial. It does not take much effort to deploy a new PEA agent which understands the changes of XML schemas. But the conversion process involves compiling new schema file with Apache XMLBeans and some coding effort to map the newly generated beans to JADE concept objects. For PEA to become more adaptable to schema changes, developing a process or a tool to automate the XSD to JADE ontology concept conversion could be beneficial for the future of PEA.

Appendix A: Platform for Privacy Preference (P3P) XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<schema
  xmlns='http://www.w3.org/2001/XMLSchema'
  xmlns:p3p='http://www.w3.org/2002/01/P3Pv1'
  targetNamespace='http://www.w3.org/2002/01/P3Pv1'
  elementFormDefault='qualified'>

<!-- enabling xml:lang attribute -->
 <import namespace='http://www.w3.org/XML/1998/namespace'
    schemaLocation='http://www.w3.org/2001/xml.xsd' />

<!-- Basic P3P Data Type -->
 <simpleType name='yes_no'>
  <restriction base='string'>
   <enumeration value='yes'/>
   <enumeration value='no'/>
  </restriction>
 </simpleType>


<!-- *********** Policy Reference *********** -->
<!-- ************** META ************** -->
 <element name='META'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:POLICY-REFERENCES'/>
    <element ref='p3p:POLICIES' minOccurs='0'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

<!-- ******* POLICY-REFERENCES ******** -->
 <element name='POLICY-REFERENCES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXPIRY' minOccurs='0'/>
    <element ref='p3p:POLICY-REF' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:HINT' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
  </sequence>
  </complexType>
 </element>

 <element name='POLICY-REF'>
  <complexType>
   <sequence>
    <element name='INCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element name='EXCLUDE'
```

```
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element name='COOKIE-INCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='p3p:cookie-
element'/>
    <element name='COOKIE-EXCLUDE'
             minOccurs='0' maxOccurs='unbounded' type='p3p:cookie-
element'/>
    <element name='METHOD'
             minOccurs='0' maxOccurs='unbounded' type='anyURI'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute name='about' type='anyURI' use='required'/>
  </complexType>
 </element>

 <complexType name='cookie-element'>
  <attribute name='name' type='string' use='optional'/>
  <attribute name='value' type='string' use='optional'/>
  <attribute name='domain' type='string' use='optional'/>
  <attribute name='path' type='string' use='optional'/>
 </complexType>

<!-- ************* HINT ************* -->
 <element name='HINT'>
  <complexType>
   <attribute name='scope' type='string' use='required'/>
   <attribute name='path' type='string' use='required'/>
  </complexType>
 </element>

<!-- ************ POLICIES ************ -->
 <element name='POLICIES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXPIRY' minOccurs='0'/>
    <element ref='p3p:DATASCHEMA' minOccurs='0'/>
    <element ref='p3p:POLICY' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>


<!-- ************* EXPIRY ************* -->
 <element name='EXPIRY'>
  <complexType>
   <attribute name='max-age' type='nonNegativeInteger' use='optional'/>
   <attribute name='date' type='string' use='optional'/>
  </complexType>
 </element>

<!-- **************** Policy **************** -->
<!-- ************* POLICY ************* -->
 <element name='POLICY'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:TEST' minOccurs='0'/>
    <element ref='p3p:ENTITY'/>
    <element ref='p3p:ACCESS'/>
    <element ref='p3p:DISPUTES-GROUP' minOccurs='0'/>
    <element ref='p3p:STATEMENT' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute name='discuri' type='anyURI' use='required'/>
   <attribute name='opturi' type='anyURI' use='optional'/>
   <attribute name='name' type='ID' use='required'/>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

<!-- ************* TEST ************* -->
 <element name='TEST'>
  <complexType/>
 </element>

<!-- ************* ENTITY ************* -->
 <element name='ENTITY'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element name='DATA-GROUP'>
     <complexType>
      <sequence>
       <element name='DATA' type='p3p:data-in-entity'
maxOccurs='unbounded'/>
      </sequence>
     </complexType>
    </element>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='data-in-entity' mixed='true'>
  <attribute name='ref' type='anyURI' use='required'/>
 </complexType>

<!-- ************* ACCESS ************* -->
 <element name='ACCESS'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice>
     <element name='nonident' type='p3p:access-value'/>
     <element name='ident-contact' type='p3p:access-value'/>
     <element name='other-ident' type='p3p:access-value'/>
     <element name='contact-and-other' type='p3p:access-value'/>
     <element name='all' type='p3p:access-value'/>
     <element name='none' type='p3p:access-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='access-value'/>

<!-- ************ DISPUTES ************ -->
 <element name='DISPUTES-GROUP'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element ref='p3p:DISPUTES' maxOccurs='unbounded'/>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <element name='DISPUTES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice minOccurs='0'>
     <sequence>
      <element ref='p3p:LONG-DESCRIPTION'/>
      <element ref='p3p:IMG' minOccurs='0'/>
      <element ref='p3p:REMEDIES' minOccurs='0'/>
      <element ref='p3p:EXTENSION' minOccurs='0'
maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element ref='p3p:IMG'/>
      <element ref='p3p:REMEDIES' minOccurs='0'/>
      <element ref='p3p:EXTENSION' minOccurs='0'
maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element ref='p3p:REMEDIES'/>
      <element ref='p3p:EXTENSION' minOccurs='0'
maxOccurs='unbounded'/>
     </sequence>
    </choice>
   </sequence>
   <attribute name='resolution-type' use='required'>
    <simpleType>
     <restriction base='string'>
      <enumeration value='service'/>
      <enumeration value='independent'/>
      <enumeration value='court'/>
      <enumeration value='law'/>
     </restriction>
    </simpleType>
   </attribute>
   <attribute name='service' type='anyURI' use='required'/>
   <attribute name='verification' type='string' use='optional'/>
   <attribute name='short-description' type='string' use='optional'/>
  </complexType>
 </element>

<!-- ******** LONG-DESCRIPTION ******** -->
```

```
 <element name='LONG-DESCRIPTION'>
  <simpleType>
   <restriction base='string'/>
  </simpleType>
 </element>

<!-- ************** IMG *************** -->
 <element name='IMG'>
  <complexType>
   <attribute name='src' type='anyURI' use='required'/>
   <attribute name='width' type='nonNegativeInteger' use='optional'/>
   <attribute name='height' type='nonNegativeInteger' use='optional'/>
   <attribute name='alt' type='string' use='required'/>
  </complexType>
 </element>

<!-- ************ REMEDIES ************ -->
 <element name='REMEDIES'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='correct' type='p3p:remedies-value'/>
     <element name='money' type='p3p:remedies-value'/>
     <element name='law' type='p3p:remedies-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='remedies-value'/>

<!-- *********** STATEMENT ************ -->
 <element name='STATEMENT'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <element name='CONSEQUENCE' minOccurs='0' type='string'/>
    <choice>
     <sequence>
      <element ref='p3p:PURPOSE'/>
      <element ref='p3p:RECIPIENT'/>
      <element ref='p3p:RETENTION'/>
      <element name='DATA-GROUP' type='p3p:data-group-type'
maxOccurs='unbounded'/>
     </sequence>
     <sequence>
      <element name='NON-IDENTIFIABLE'/>
      <element ref='p3p:PURPOSE' minOccurs='0'/>
      <element ref='p3p:RECIPIENT' minOccurs='0'/>
      <element ref='p3p:RETENTION' minOccurs='0'/>
      <element name='DATA-GROUP' type='p3p:data-group-type'
minOccurs='0' maxOccurs='unbounded'/>
     </sequence>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

<!-- ************ PURPOSE ************* -->
 <element name='PURPOSE'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='current' type='p3p:purpose-value'/>
     <element name='admin' type='p3p:purpose-value'/>
     <element name='develop' type='p3p:purpose-value'/>
     <element name='tailoring' type='p3p:purpose-value'/>
     <element name='pseudo-analysis' type='p3p:purpose-value'/>
     <element name='pseudo-decision' type='p3p:purpose-value'/>
     <element name='individual-analysis' type='p3p:purpose-value'/>
     <element name='individual-decision' type='p3p:purpose-value'/>
     <element name='contact' type='p3p:purpose-value'/>
     <element name='historical' type='p3p:purpose-value'/>
     <element name='telemarketing' type='p3p:purpose-value'/>
     <element name='other-purpose'>
      <complexType mixed='true'>
       <attribute name='required' use='optional' type='p3p:required-
value'/>
      </complexType>
     </element>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <simpleType name='required-value'>
  <restriction base='string'>
   <enumeration value='always'/>
   <enumeration value='opt-in'/>
   <enumeration value='opt-out'/>
  </restriction>
 </simpleType>

 <complexType name='purpose-value'>
  <attribute name='required' use='optional' type='p3p:required-value'
default='always' />
 </complexType>

<!-- *********** RECIPIENT ************ -->
 <element name='RECIPIENT'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice maxOccurs='unbounded'>
     <element name='ours'>
      <complexType>
       <sequence>
        <element ref='p3p:recipient-description' minOccurs='0'
maxOccurs='unbounded'/>
```

```
       </sequence>
      </complexType>
     </element>
     <element name='same' type='p3p:recipient-value'/>
     <element name='other-recipient' type='p3p:recipient-value'/>
     <element name='delivery' type='p3p:recipient-value'/>
     <element name='public' type='p3p:recipient-value'/>
     <element name='unrelated' type='p3p:recipient-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='recipient-value'>
  <sequence>
   <element ref='p3p:recipient-description' minOccurs='0'
maxOccurs='unbounded'/>
  </sequence>
  <attribute name='required' use='optional' type='p3p:required-value'/>
 </complexType>

 <element name='recipient-description'>
  <complexType mixed='true'/>
 </element>

<!-- *********** RETENTION ************ -->
 <element name='RETENTION'>
  <complexType>
   <sequence>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
    <choice>
     <element name='no-retention' type='p3p:retention-value'/>
     <element name='stated-purpose' type='p3p:retention-value'/>
     <element name='legal-requirement' type='p3p:retention-value'/>
     <element name='indefinitely' type='p3p:retention-value'/>
     <element name='business-practices' type='p3p:retention-value'/>
    </choice>
    <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
  </complexType>
 </element>

 <complexType name='retention-value'/>

<!-- ************** DATA ************** -->
 <complexType name='data-group-type'>
  <sequence>
   <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
   <element name='DATA' type='p3p:data-in-statement'
maxOccurs='unbounded'/>
   <element ref='p3p:EXTENSION' minOccurs='0' maxOccurs='unbounded'/>
  </sequence>
  <attribute name='base' type='anyURI'
             use='optional' default='http://www.w3.org/TR/P3P/base'/>
 </complexType>
 <complexType name='data-in-statement' mixed='true'>
  <sequence minOccurs='0' maxOccurs='unbounded'>
   <element ref='p3p:CATEGORIES'/>
  </sequence>
  <attribute name='ref' type='anyURI' use='required'/>
  <attribute name='optional' use='optional' default='no'
type='p3p:yes_no'/>
 </complexType>

<!-- ************** Data Schema ************* -->
<!-- *********** DATASCHEMA *********** -->
 <element name='DATASCHEMA'>
  <complexType>
   <choice minOccurs='0' maxOccurs='unbounded'>
    <element ref='p3p:DATA-DEF'/>
    <element ref='p3p:DATA-STRUCT'/>
    <element ref='p3p:EXTENSION'/>
   </choice>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
 </element>

 <element name='DATA-DEF' type='p3p:data-def'/>
 <element name='DATA-STRUCT' type='p3p:data-def'/>

 <complexType name='data-def'>
  <sequence>
   <element ref='p3p:CATEGORIES' minOccurs='0'/>
   <element ref='p3p:LONG-DESCRIPTION' minOccurs='0'/>
  </sequence>
  <attribute name='name' type='ID' use='required'/>
  <attribute name='structref' type='anyURI' use='optional'/>
  <attribute name='short-description' type='string' use='optional'/>
 </complexType>

<!-- *********** CATEGORIES *********** -->
 <element name='CATEGORIES'>
  <complexType>
   <choice maxOccurs='unbounded'>
    <element name='physical' type='p3p:categories-value'/>
    <element name='online' type='p3p:categories-value'/>
    <element name='uniqueid' type='p3p:categories-value'/>
    <element name='purchase' type='p3p:categories-value'/>
    <element name='financial' type='p3p:categories-value'/>
    <element name='computer' type='p3p:categories-value'/>
    <element name='navigation' type='p3p:categories-value'/>
    <element name='interactive' type='p3p:categories-value'/>
    <element name='demographic' type='p3p:categories-value'/>
    <element name='content' type='p3p:categories-value'/>
    <element name='state' type='p3p:categories-value'/>
    <element name='political' type='p3p:categories-value'/>
    <element name='health' type='p3p:categories-value'/>
    <element name='preference' type='p3p:categories-value'/>
    <element name='location' type='p3p:categories-value'/>
    <element name='government' type='p3p:categories-value'/>
    <element name='other-category' type='string'/>
   </choice>
  </complexType>
 </element>

 <complexType name='categories-value'/>

<!-- *********** EXTENSION ************ -->
 <element name='EXTENSION'>
  <complexType mixed='true'>
   <choice minOccurs='0' maxOccurs='unbounded'>
    <any minOccurs='0' maxOccurs='unbounded' processContents='skip'/>
   </choice>
   <attribute name='optional' use='optional' default='yes'
type='p3p:yes_no'/>
  </complexType>
 </element>

</schema>
```

Appendix B: P3P Policy Schematic

B.1Policy Element

```
<POLICIES xmlns="http://www.w3.org/2002/01/p3pv1">
   <POLICY name="default"
      discuri="http://www.mycompnay.com/privacy/policy.html"
      opturi="http://www.mycompany.com/privacy/preference.hmtl"
    xml:lang="en">
   </POLICY>
</POLICIES>
```

- **<POLICIES>:** Contains reference to P3P XML schema. POLICY tag is the root tag for all policies of the site which are distinguished by the policy name attribute. The policies can also be separated into different files.

- **<POLICY>:** Contains a single policy with name which must be unique through out the Web site policies.
  - **name** - unique policy identifier
  - **discuri** - uniform resource identifier (path) to the natural language policy.
  - **opturi** - path to where user can deny or request the data be used for a specific purpose.
  - **xml:lang** – language used in the XML policy. This only applies to the pure text, not the tag name or predefined attribute value.

B.2 Entity Element

```
<POLICIES xmlns="http://www.w3.org/2002/01/p3pv1">
   <POLICY name="default"
```

```
    discuri="http://www.mycompnay.com/privacy/policy.html"
    opturi="http://www.mycompany.com/privacy/preference.hmtl"
    xml:lang="en">
      <ENTITY>
         <DATA-GROUP>
            <DATA ref="#business.name">Company Name</DATA>
            <DATA ref="#business.contact-info.postal.street">1880 Main
               st.</DATA>
            <DATA ref="#business.contact-info.postal.city">San
Jose</DATA>
            <DATA ref="#business.contact-
info.postal.stateprov">CA</DATA>
            <DATA ref="#business.contact-
info.postal.postalcode">12345</DATA>
            <DATA ref="#business.contact-
info.postal.country">USA</DATA>
            <DATA ref="#business.contact-
info.online.email">help@mycompany.com
               </DATA>
          .......
         </DATA-GROUP>
      </ENTITY>
   </POLICY>
</POLICIES>
```

- **<ENTITY>:** contains information about the company which collects the information. It must contains legal entity name and one or more contact information among postal address, telephone number, e-mail address and/or URI.
- **<DATA-GROUP>:** see <DATA-GROUP> below
- **<DATA>:** see <DATA> below

B.3 Access Element

```
<POLICIES xmlns="http://www.w3.org/2002/01/p3pv1">
   <POLICY name="default"
      discuri="http://www.mycompnay.com/privacy/policy.html"
      opturi="http://www.mycompany.com/privacy/preference.hmtl"
xml:lang="en">
      <ENTITY>
         <DATA-GROUP>
            <DATA ref="#business.name">Company Name</DATA>
          .......
         </DATA-GROUP>
      </ENTITY>
      <ACCESS>
         <nonident />
      </ACCESS>
   </POLICY>
</POLICIES>
```

- **<ACCESS>:** The access element indicates whether the site provide access to various kind of information.
  - **<nonident/>** - the site does not collect identified data.

- **<all/>** - access is given to all identified data.
- **<contact-and-other>** - access is given to identified online and physical contact information as well as to certain other identified data.
- **<ident-contact/>** - access is given to identified online and physical contact information.
- **<other-ident>** - access to other identified data is given.
- **<none/>** - no access is provided.

B.4 Disputes Element

```
<POLICIES xmlns="http://www.w3.org/2002/01/p3pv1">
   <POLICY name="default"
      discuri="http://www.mycompnay.com/privacy/policy.html"
      opturi="http://www.mycompany.com/privacy/preference.hmtl"
xml:lang="en">
      <ENTITY>
         <DATA-GROUP>
            <DATA ref="#business.name">Company Name</DATA>
          .......
         </DATA-GROUP>
      </ENTITY>
      <ACCESS>
         <nonident />
      </ACCESS>
      <DISPUTES-GROUP>
         <DISPUTES resolution-type="service"
            service="http://customerservice.mycompany.com"
            short-description="Customer Service">
            <REMEDIES>
               <correct />
            </REMEDIES>
         </DISPUTES>
         <DISPUTES resolution-type="independent"
            service="http://www.independentprivacyorg.com"
            short-description="Independent Privacy Organization">
            <LONG-DESCRIPTION>If you are not satisfy with the response
from
               mycompany.com, please contact Independent Privacy
Organization
               at http://www.independentprivacyorg.com. </LONG-
DESCRIPTION>
            <REMEDIES>
               <correct />
            </REMEDIES>
         </DISPUTES>
      </DISPUTES-GROUP>
   </POLICY>
</POLICIES>
```

- **<DISPUTES-GROUP>:** A company privacy policy should contain a DISPUTES-GROUP element which contains one or more DISPUTES element.

- **<DISPUTES>:** The tag contains information regarding where the user should turn to if there is a disagreement about the privacy policy and also how the company handle disputes.
    - **resolution-type** (mandatory)
        - **service** – one may complain to the company's customer service representative for disputes regarding the use of collected data. Customer service contact information must also be provided.
        - **independent** – one may complain to an independent organization foe the disputes. The contact information to the organization must also be included.
        - **court** – one may file a legal complaint against the company.
        - **law** – disputes regarding the privacy policy shall be resolve in accordance with the legal reference in the description.
    - **service** (mandatory) – URI of the company customer service page or independent organization, or information retaining the relevant legal references.
    - **verification** – URI or document that can be sure for verification purpose.
    - **short-description** – a short description (< 255 characters) of the company dispute department, third party organization, applicable legal documents.
- **<LONG_DESCRIPTION>:** contains long human readable description related to the dispute.
- **<REMEDIES>:** remedies if an error occurred.
    - **<correct/>** - errors or wrongful actions arising in connection with the privacy policy will be remedied by the service.
    - **<money/>** - if the service provider violates its privacy policy, it will pay the individual an amount specified in the human-readable privacy policy or the amount of damages.
    - **<law/>** - remedies for the breaches of the policy statement will be determined base on the law references in the human-readable description.

B.5 Statement Element

```
  <POLICIES xmlns="http://www.w3.org/2002/01/p3pv1">
   <POLICY name="default"
      discuri="http://www.mycompnay.com/privacy/policy.html"
      opturi="http://www.mycompany.com/privacy/preference.hmtl"
xml:lang="en">
      <ENTITY>
         .......
      </ENTITY>
      <ACCESS>
         <nonident />
      </ACCESS>
      <DISPUTES-GROUP>
         .......
      </DISPUTES-GROUP>
      <!-- use of #dynamic.miscdata-->
      <STATEMENT>
         <PURPOSE><admin/><develop/></PURPOSE>
         <RECIPIENT><ours/></RECIPIENT>
         <RETENTION><stated-purpose/></RETENTION>
         <DATA-GROUP>
            <DATA ref="#dynamic.miscdata" optional="no">
               <CATEGORIES>
                  <demographic/>
                  <navigation/>
                  <state/>
                  <uniqueid/>
                  <computer/>
               </CATEGORIES>
            </DATA>
         </DATA-GROUP>
      </STATEMENT>
      <!--use of use of elements from base data schema-->
      <STATEMENT>
         <CONSEQUENCE> We record some information in order to serve
your
            request and to secure and improve our Web site.
         </CONSEQUENCE>
         <PURPOSE><admin/><develop/></PURPOSE>
         <RECIPIENT><ours/></RECIPIENT>
         <RETENTION><stated-purpose/></RETENTION>
         <DATA-GROUP>
            <DATA ref="#dynamic.clickstream"/>
            <DATA ref="#dynamic.http"/>
         </DATA-GROUP>
      </STATEMENT>

      <!--non-identifiable data-->
      <STATEMENT>
         <NON-IDENTIFIABLE/>
         <DATA-GROUP>
            <DATA ref="#user.gender"/>
         </DATA-GROUP>
      </STATEMENT>

      <STATEMENT>
         <PURPOSE><current/><individual-analysis required="opt-
in"/></PURPOSE>
         <RECIPIENT><ours/></RECIPIENT>
         <RETENTION><stated-purpose/></RETENTION>
         <DATA-GROUP>
            <DATA ref="#user.home-info.telecom.telephone"
optional="no"/>
         </DATA-GROUP>
      </STATEMENT>
      <STATEMENT>
         <PURPOSE><current/><pseudo-analysis/></PURPOSE>
         <RECIPIENT>
            <unrelated>We share information about your gender with our
               advertisers. The information is not in itself personally
               identifiable, unless the advertiser can identify the
user
               through the IP address or through the use of cookie file.
            </unrelated>
         </RECIPIENT>
         <RETENTION><stated-purpose/></RETENTION>
         <DATA-GROUP>
            <DATA ref="#user.gender"/>
         </DATA-GROUP>
      </STATEMENT>
   </POLICY>
</POLICIES>
```

- **<STATEMENT>:** Defines data practices as applied to data elements enclosed. The STATEMENT element is a container that groups together a PURPOSE element, a RECIPIENT element, a RETENTION element, a DATA-GROUP element, and optionally a CONSEQUENCE element and one or more extensions. All the data referenced by the DATA-GROUP is handled according to the disclosures made in the other elements in the statement.
- **<CONSEQUENCE>:** Contains human-readable information about what the Web site is about to do to the data in the statement, why and for how long. This tag is optional.
- **<NON-IDENTIFIABLE>:** A statement can be declared non-identifiable if either that there is no data collected under this STATEMENT, or that all of the data referenced by that STATEMENT will be anonymized upon collection. If the NON-IDENTIFIABLE element is present in any STATEMENT elements in a policy, then a human readable explanation of how the data is anonymized MUST be included or linked to at the **discuri**. If the non-identifiable tag is used, the purpose, recipient and retention tags are optional.
- **<PURPOSE>:** Purposes for data processing relevant to the Web site and must contain one or more of the following:
  - **<current/> - Completion and Support of Activity For Which Data Was Provided:** Information may be used by the service provider to complete the activity for which it was provided, or a recurring activity.
  - **<admin/>** - Web Site and System Administration: Information may be used for technical support of the Web site and its computer system.
  - **<develop/> - Research and Development:** Information may be

used to enhance, evaluate, or otherwise review the site, service, product or market.

- **<tailoring/> -** One-time Tailoring**:** Information may be used to tailor or modify content or design of the Web site where information is used only for a single visit to the site and not used for any kind of future customization.
- **<pseudo-analysis> -** Pseudonymous Analysis**:** Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *for purpose of research, analysis and reporting*, but it will not be used to attempt to identify specific individuals.
- **<pseudo-decision/> -** Pseudonymous Decision**:** Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *to make a decision that directly affects that individual*, but it will not be used to attempt to identify specific individuals.
- **<individual-analysis/>** - **Individual Analysis**: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *for the purpose of research, analysis and reporting*.
- **<individual-decision/> - Individual Decision**: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *to make a decision that directly affects that individual*. For example, an online store suggests items a visitor may wish to purchase based on items he has purchased during previous visits to the Web site.
- **<contact/>** - **Contacting Visitors for Marketing of Services or Products**: Information may be used to contact the individual, through a communications channel other than voice telephone, for the promotion of a product or service. This includes notifying visitors about updates to the Web site. This does not include a direct reply to a question or comment or customer service for a single transaction -- in those cases, **<current/>** would be used. In addition, this does not include marketing via customized Web content or banner advertisements embedded in sites the user is visiting -- these cases would be covered by the **<tailoring/>**, **<pseudo-analysis/>** and **<pseudo-decision/>**, or **<individual-analysis/>** and **<individual-decision/>**purposes.

    - **<historical/>** - **Historical Preservation**: Information may be archived or stored for the purpose of preserving social history as governed by an existing law or policy. This law or policy MUST be referenced in the `<DISPUTES>` element and MUST include a specific definition of the type of qualified researcher who can access the information, where this information will be stored and specifically how this collection advances the preservation of history.
    - **<telemarketing/>** - **Contacting Visitors for Marketing of Services or Products Via Telephone**: Information may be used to contact the individual via a voice telephone call for promotion of a product or service. This does not include a direct reply to a question or comment or customer service for a single transaction -- in those cases, **<current/>** would be used.
    - **<other-purpose>** ***string*** **</other-purpose>** - **Other Uses**: Information may be used in other ways not captured by the above definitions. (A human readable explanation MUST be provided in these instances.)

If you would like to provide the user with the possibility to opt in or opt out of the data collection, you can add a ***required*** attribute (with the exception of **<current/>**), which has the following values:

    - **always** : The purpose is always required; users cannot opt-in or opt-out of this use of their data. This is the default when no `required` attribute is present.
    - **opt-in** : Data may be used for this purpose only when the user affirmatively requests this use. For example, when a user asks to be added to a mailing list. An affirmative request requires users to take some action specifically to make the request. In addition, for any purpose that users may affirmatively request, there must also be a way for them to change their minds later and decline which must be specified at the *opturi*.
    - **opt-out** : Data may be used for this purpose unless the user requests that it not be used in this way. When this value is selected, the service must provide clear instructions to users on how to opt-out of this purpose at the *opturi*. Services should also provide these instructions or a pointer to these instructions at the point of data collection.

- **<RECIPIENT>:** The legal entity, or domain beyond the service provider and its agent where data may be distributed. The element must contain one or more of the following:
    - **<ours>** - **Ourselves and/or entities acting as our agents or entities for whom we are acting as an agent**: An agent in this instance is defined as a third party that processes data only on

behalf of the service provider for the completion of the stated purposes.

- **<delivery> - Delivery services possibly following different practices**: Legal entities *performing delivery services* that may use data for purposes other than completion of the stated purpose. This should also be used for delivery services whose data practices are unknown.
- **<same> - Legal entities following our practices**: Legal entities who use the data on their own behalf under equable practices.
- **<other-recipient> - Legal entities following different practices**: Legal entities that are constrained by and accountable to the original service provider, but may use the data in a way not specified in the service provider's practices.
- **<unrelated> - Unrelated third parties**: Legal entities whose data usage practices are not known by the original service provider.
- **<public> - Public forum**: Public forum such as bulletin boards, public directories, or commercial CD-ROM directories.

Each of the above tags can optionally contains the following:

- one or more `recipient-description` tags, containing a description of the recipient.
- with the exception of `<ours/>`, a **required** attribute: this attribute is defined exactly as the analogous attribute in the `PURPOSE` tag, indicating whether opt-in/opt-out of sharing is available. The default value is “always”

- **<RETENTION>:** Indicates the kind of retention policy that applies to the data referenced in that statement. The element must contain one of the following.
  - **<no-retention/> - Information is not retained** for more than a brief period of time necessary to make use of it during the course of a single online interaction. Information MUST be destroyed following this interaction and MUST NOT be logged, archived, or otherwise stored. This type of retention policy would apply, for example, to services that keep no Web server logs, set cookies only for use during a single session, or collect information to perform a search but do not keep logs of searches performed.
  - **<stated-purpose/>** - **For the stated purpose:** Information is retained to meet the stated purpose. This requires information to be discarded at the earliest time possible. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.
  - **<legal-requirement/> - As required by law or liability under**

applicable law: Information is retained to meet a stated purpose, but the retention period is longer because of a legal requirement or liability. For example, a law may allow consumers to dispute transactions for a certain time period; therefore a business may for liability reasons decide to maintain records of transactions, or a law may affirmatively require a certain business to maintain records for auditing or other soundness purposes. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

  - **<business-practices/>** - **Determined by service provider's business practice:** Information is retained under a service provider's stated business practices. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.
  - **<indefinitely/> - Indefinitely:** Information is retained for an indeterminate period of time. The absence of a retention policy would be reflected under this option. Where the recipient is a public forum, this is the appropriate retention policy.

Each `STATEMENT` element that does not include a `NON-IDENTIFIABLE` element MUST contain at least one `DATA-GROUP` element that contains one or more `DATA` elements. `DATA` elements are used to describe the type of data that a site collects.

- **<DATA-GROUP>:** Group the data to be transferred or inferred.
  - **base** – base URI for URI references in ref attribute of DATA element. When omitted, the default value is the URI of the P3P base data schema (http://www.w3.org/TR/P3P/base.) When leave as empty string (“”) , the base is local document.
- **<DATA>:** Describe the data to be transferred or inferred.
  - **ref** (mandatory) – URI reference of the corresponding data schema. If the URI part is missing, the default URI is the base URI of the data group. The **names of data elements and sets are *case-sensitive*.**
  - **optional** - indicates whether or not the site requires visitors to submit this data element to access a resource or complete a transaction; "no" indicates that the data element is not optional (it is required), while "yes" indicates that the data element is optional. *The default is "no."*

Appendix C: PEA Privacy Preference XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<schema
  xmlns='http://www.w3.org/2001/XMLSchema'
  xmlns:pea='http://localhost/PEA'
  targetNamespace='http://localhost/PEA'
  elementFormDefault='qualified'>

<!-- enabling xml:lang attribute -->
 <import namespace='http://www.w3.org/XML/1998/namespace'
    schemaLocation='http://www.w3.org/2001/xml.xsd' />

<!-- Root container -->
<element name='ROOT'>
  <complexType>
   <sequence>
    <element ref='pea:PREFERENCESET' minOccurs='0'
maxOccurs='unbounded'/>
   </sequence>
   <attribute ref='xml:lang' use='optional'/>
  </complexType>
</element>

<!-- ************ PREFERENCESET ************ -->
 <element name='PREFERENCESET'>
  <complexType>
   <sequence>
    <element ref='pea:PREFERENCE' minOccurs='0' maxOccurs='unbounded'/>
   </sequence>
   <attribute name='ownerId' type='ID' use='required'/>
  </complexType>
 </element>

 <element name='PREFERENCE'>
  <complexType>
   <sequence>
    <element ref='pea:DATA' minOccurs='1' maxOccurs='unbounded'/>
    <element ref='pea:RULE'/>
    <element ref='pea:HISTORY'/>
   </sequence>
   <attribute name='name' type='ID' use='required'/>
  </complexType>
 </element>

<!-- ************* RULE ************* -->
<element name='RULE'>
  <complexType>
   <sequence>
    <element ref='pea:PURPOSE'/>
    <element ref='pea:ACCESS'/>
    <element ref='pea:RECIPIENT'/>
    <element ref='pea:RETENTION'/>
   </sequence>
   <attribute name='mandatory' use='optional' default='no'
type='pea:yes_no'/>
```

```
  </complexType>
 </element>

<!-- ************* ACCESS ************* -->
 <element name='ACCESS'>
  <complexType>
   <sequence>
     <choice>
     <element name='nonident' type='pea:access-value'/>
     <element name='ident-contact' type='pea:access-value'/>
     <element name='other-ident' type='pea:access-value'/>
     <element name='contact-and-other' type='pea:access-value'/>
     <element name='all' type='pea:access-value'/>
     <element name='none' type='pea:access-value'/>
    </choice>
   </sequence>
  </complexType>
 </element>

 <complexType name='access-value'/>


 <!-- ************ PURPOSE ************* -->
 <element name='PURPOSE'>
  <complexType>
   <sequence>
    <choice maxOccurs='unbounded'>
     <element name='current' type='pea:purpose-value'/>
     <element name='admin' type='pea:purpose-value'/>
     <element name='develop' type='pea:purpose-value'/>
     <element name='tailoring' type='pea:purpose-value'/>
     <element name='pseudo-analysis' type='pea:purpose-value'/>
     <element name='pseudo-decision' type='pea:purpose-value'/>
     <element name='individual-analysis' type='pea:purpose-value'/>
     <element name='individual-decision' type='pea:purpose-value'/>
     <element name='contact' type='pea:purpose-value'/>
     <element name='historical' type='pea:purpose-value'/>
     <element name='telemarketing' type='pea:purpose-value'/>
     <element name='other-purpose'>
      <complexType mixed='true'>
       <attribute name='required' use='optional' type='pea:required-
value'/>
      </complexType>
     </element>
    </choice>
   </sequence>
  </complexType>
 </element>

 <simpleType name='required-value'>
  <restriction base='string'>
   <enumeration value='always'/>
   <enumeration value='opt-in'/>
   <enumeration value='opt-out'/>
  </restriction>
 </simpleType>
 <complexType name='purpose-value'>
  <attribute name='required' use='optional' type='pea:required-value'
default='always' />
 </complexType>

 <!-- *********** RECIPIENT ************ -->
 <element name='RECIPIENT'>
  <complexType>
   <sequence>
    <choice maxOccurs='unbounded'>
     <element name='ours'>
      <complexType>
       <sequence>
        <element ref='pea:recipient-description' minOccurs='0'
maxOccurs='unbounded'/>
       </sequence>
      </complexType>
     </element>
     <element name='same' type='pea:recipient-value'/>
     <element name='other-recipient' type='pea:recipient-value'/>
     <element name='delivery' type='pea:recipient-value'/>
     <element name='public' type='pea:recipient-value'/>
     <element name='unrelated' type='pea:recipient-value'/>
    </choice>
   </sequence>
  </complexType>
 </element>

 <complexType name='recipient-value'>
  <sequence>
   <element ref='pea:recipient-description' minOccurs='0'
maxOccurs='unbounded'/>
  </sequence>
  <attribute name='required' use='optional' type='pea:required-value'/>
 </complexType>

 <element name='recipient-description'>
  <complexType mixed='true'/>
 </element>

 <!-- *********** RETENTION ************ -->
 <element name='RETENTION'>
  <complexType>
   <sequence>
    <choice>
     <element name='no-retention' type='pea:retention-value'/>
     <element name='stated-purpose' type='pea:retention-value'/>
     <element name='legal-requirement' type='pea:retention-value'/>
     <element name='indefinitely' type='pea:retention-value'/>
     <element name='business-practices' type='pea:retention-value'/>
    </choice>
   </sequence>
  </complexType>
 </element>

 <complexType name='retention-value'/>
 <!-- *********** HISTORY *********** -->
 <element name='HISTORY'>
   <complexType>
    <sequence>
      <choice>
       <element name='lastAccess' type='dateTime'/>
       <element name='lastViolation' type='dateTime'/>
       <element name='accessed' type='integer'/>
       <element name='violated' type='integer'/>
       <element name='remoteAccess' type='integer'/>
       <element name='remoteUserName' type='string'/>
      </choice>
    </sequence>
    <attribute name='createTime' type='dateTime'/>
   </complexType>
 </element>

<!-- ************** DATA ************** -->
<element name='DATA'>
 <complexType>
  <attribute name='ref' type='anyURI' use='required'/>
  <attribute name='optional' use='optional' default='no'
type='pea:yes_no'/>
 </complexType>
</element>

<!-- Basic P3P Data Type -->
 <simpleType name='yes_no'>
  <restriction base='string'>
   <enumeration value='yes'/>
   <enumeration value='no'/>
  </restriction>
 </simpleType>
</schema>
```

Appendix D: P3P Base Data Schema

All P3P-compliant user agent implementations MUST be aware of the data elements in the P3P base data schema. The P3P base data schema includes the definition of the basic data structures, and four data element sets: **user**, **thirdparty**, **business** and **dynamic**. The user, thirdparty and business sets include elements that users and/or businesses might provide values for, while the dynamic set includes elements that are dynamically generated in the course of a user's browsing session. User agents may support a variety of mechanisms that allow users to provide values for the elements in the user set and store them in a data repository, including mechanisms that support multiple personae. Users may choose not to provide values for these data elements.

The formal XML definition of the P3P base data schema is given in Appendix 3. In the following sections, the base data elements and sets are explained one by one. In the future there will be in all likelihood *demand for the creation of other* *data sets and elements.* Obvious applications include catalogue, payment, and agent/system attribute schemas (an extensive set of system elements is provided for example in http://www.w3.org/TR/NOTE-agent-attributes.)

Each table below specifies a **set**, the elements within the set, the category associated with the element, its structure, and the display name shown to users. More than one category may be associated with a fixed data element. However, each base data element is assigned to only one category whenever possible. It is recommended that data schema designers do the same.

D.1 User Data

The **user** data set includes general information about the user.

user	**Category**	**Structure**	**Short display name**
name	Physical Contact Information, Demographic and Socioeconomic Data	personname	User's Name
bdate	Demographic and Socioeconomic Data	date	User's Birth Date
login	Unique Identifiers	login	User's Login Information
cert	Unique Identifiers	certificate	User's Identity Certificate
gender	Demographic and Socioeconomic Data	*unstructured*	User's Gender (Male or Female)
employer	Demographic and Socioeconomic Data	*unstructured*	User's Employer
department	Demographic and Socioeconomic Data	*unstructured*	Department or Division of Organization where User is Employed
jobtitle	Demographic and Socioeconomic Data	*unstructured*	User's Job Title
home-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Home Contact Information
business-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	User's Business Contact Information

Note, that this data set includes elements that are actually sets of data themselves. These sets are defined in the Data Structures subsection of this document. The short display name for an individual element contained within a data set is defined as the concatenation of the short display names that have been defined for the set and the element, separated by a separator appropriate for the language/script in question, e.g. a comma for English. For example, the short display name for user.home-info.postal.postalcode could be "User's Home Contact Information, Postal Address Information, Postal code". User agent implementations may prefer to develop their own short display names rather than using the concatenated names when displaying information for the user.

D.2 Third Party Data

The **thirdparty** data set allows users and businesses to provide values for a related third party. This can be useful whenever third party information needs to be exchanged, for example when ordering a present online that should be sent to another person, or when providing information about one's spouse or business partner. Such information could be stored in a user repository alongside the user data set. User agents may offer to store multiple such thirdparty data sets and allow users to select the appropriate values from a list when necessary.

The thirdparty data set is identical with the user data set. See section 5.6.1 User Data for details.

D.3 Business Data

The **business** data set features a subset of user data relevant for describing legal entities. In P3P1.0, this data set is primarily used for declaring the policy entity, although it should also be applicable to business-to-business interactions.

business	**Category**	**Structure**	**Short display name**
name	Demographic and Socioeconomic Data	*unstructured*	Organization Name
department	Demographic and Socioeconomic Data	*unstructured*	Department or Division of Organization
cert	Unique Identifiers	certificate	Organization Identity Certificate
contact-info	Physical Contact Information, Online Contact Information, Demographic and Socioeconomic Data	contact	Contact Information for the Organization

D.4 Dynamic Data

In some cases, there is a need to specify data elements that do not have fixed values that a user might type in or store in a repository. In the P3P base data schema, all such elements are grouped under the dynamic data set. Sites may refer to the types of data they collect using the dynamic data set only, rather than enumerating all of the specific data elements.

dynamic	**Category**	**Structure**	**Short display name**
clickstream	Navigation and Click-stream Data, Computer Information	loginfo	Click-stream Information
http	Navigation and Click-stream Data, Computer Information	httpinfo	HTTP Protocol Information
clientevents	Navigation and Click-stream Data	*unstructured*	User's Interaction with a Resource
cookies	*(variable-category)*	*unstructured*	Use of HTTP Cookies
miscdata	*(variable-category)*	*unstructured*	Miscellaneous Non-base Data Schema Information
searchtext	Interactive Data	*unstructured*	Search Terms
interactionrecord	Interactive Data	*unstructured*	Server Stores the Transaction History

These elements are often implicit in navigation or Web interactions. They should be used with categories to describe the type of information collected through these methods. A brief description of each element follows.

clickstream

The clickstream element is expected to apply to practically all Web sites. It represents the combination of information typically found in Web server access logs: the IP address or hostname of the user's computer, the URI of the resource requested, the time the request was made, the HTTP method used in the request, the size of the response, and the HTTP status code in the response. Web sites that collect standard server access logs as well as sites which do URI path analysis can use this data element to describe how that data will be used. Web sites that collect only some of the data elements listed for the clickstream element MAY choose to list those specific elements rather than the entire dynamic.clickstream element. This allows sites with more limited data-collection practices to accurately present those practices to their visitors.

http

The http element contains additional information contained in the HTTP protocol. See the definition of the httpinfo structure for descriptions of specific elements. Sites MAY use the dynamic.http field as a shorthand to cover all the elements in the httpinfo structure if they wish, or they MAY reference the specific elements in the httpinfo structure.

clientevents

The clientevents element represents data about how the user interacts with their Web browser while interacting with a resource. For example, an application may wish to collect information about whether the user moved their mouse over a certain image on a page, or whether the user ever brought up the help window in a Java applet. This kind of information is represented by the dynamic.clientevents data element. Much of this interaction record is represented by the events and data defined by the Document Object Model (DOM) Level 2 Events [DOM2-Events]. The clientevents data element also covers any other data regarding the user's interaction with their browser while the browser is displaying a resource. The exception is events which are covered by other elements in the base data schema. For example, requesting a page by clicking on a link is part of the user's interaction with their browser while viewing a page, but merely collecting the URL the user has clicked on does not require declaring this data element; clickstream covers that event. However, the DOM event DOMFocusIn (representing the user moving their mouse over an object on a page) is not covered by any other existing element, so if a site is collecting the occurrence of this event, then it needs to state that it collects the dynamic.clientevents element. Items covered by this data element are typically collected by client-side scripting languages, such as JavaScript, or by client-side applets, such as ActiveX or Java applets. Note that while the previous discussion has been in terms of a user viewing a resource, this data element also applies to Web applications which do not display resources visually - for example, audio-based Web browsers.

cookies

The cookies element should be used whenever HTTP cookies are set or retrieved by a site. Please note that cookies is a *variable data element* and requires the explicit declaration of usage categories in a policy.

miscdata

The miscdata element references information collected by the service that the service does not reference using a specific data element. Categories have to be used to better describe these data: sites MUST reference a separate miscdata element in their policies for each category of miscellaneous data they collect.

searchtext

The searchtext element references a specific type of solicitation used for searching and indexing sites. For example, if the only fields on a search engine page are search fields, the site only needs to disclose that data element.

interactionrecord

The interactionrecord element should be used if the server is keeping track of the interaction it has with the user (i.e. information other than clickstream data, for example account transactions, etc

Bibliography

[1] World Wide Web Consortium (W3C), Platform for Privacy Preferences Project – P3P, W3C, Cambridge, MA, http://www.w3.org/p3p

[2] World Wide Web Consortium (W3C) (2005), *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification*, http://www.w3.org/TR/2005/WD-P3P11-20050701

[3] Yahoo News (2006-05-18), *One billion people have Internet access*, http://news.yahoo.com/s/afp/20060518/tc_afp/afplifestyleitinternet_060518163500

[4] eMarketer (2006 June), *US Retail E-Commerce*, http://www.emarketer.com/Report.aspx?ecom_us_jun06

[5] JADE (Java Agent Development Framework), http://Worldjade.tilab.com/

[6] JADE Online Document, *Creating and using applications-specific ontologies*, http://jade.tilab.com/doc/CLOntoSupport.pdf

[7] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa: *JADE White* (2003), http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf

[8] FIPA (The Foundation of Intelligent Physical Agent), http://www.fipa.org/

[9] Apache Xindice, http://xml.apache.org/xindice/

[10] eXist, open source native XML database, http://exist.sourceforge.net/

[11] Apache XMLBeans, http://xmlbeans.apache.org/

[12] Mark S. Ackerman, Lorrie Cranor. *Privacy Critics: UI Components to Safeguard Users' Privacy*, ACM Press, May 1999

[13] Giovanni Caire, David Cabanillas. *Application-Defined Content Languages and Ontologies*, TILab S.p.A., Nov. 2004.

[14] Lorrie Faith Cranor. *Web Privacy with P3P*, O'Reilly & Associates, Sep. 2002

[15] Helena Lindskog, Stefan Lindskog. *Web Site Privacy with P3P*, Wiley Publishing, 2003

[16] Lorrie Faith Cranor. *Agents of Choice: Tools That Facilitate Notice and Choice Web Site Data Practices*, AT&T Labs-Research.

[17] Wei Xu, R. Sekar, I.V. Ramakrishnan, V.N. Venkatakrishnan. *An Approach for Realizing Privacy-Preserving Web-Based Services*. ACM Press, May 2005.

[18] Karen Coyle. *P3P:Pretty Poor Privacy? A Social Analysis of the Platform for Privacy Preferences*. June 1999, http:/www.kcoyle.net/p3p.html

[19] Saket Kaushik, Duminda Wijesekera, Paul Ammann. *Policy-Based Dissemination of Patial Web-Ontologies*. SWS'05, ACM Press. Nov 11 2005

[20] Marco Casassa Mont. *Dealing with Privacy Obligation in Enterprises*. HP Laboratories Bristol. June 30, 204

Glossary

FIPA The Foundation for Intelligent Physical Agents (FIPA) produces computer software standards for heterogeneous and interacting agents and agent-based systems

JADE Java Development Framework is an agent-oriented java development framework developed by Tilab.

P3P The Platform for Privacy Preferences is a protocol designed to give users more control of their personal information when browsing Internet Websites. P3P was developed by the World Wide Web Consortium (W3C) and was officially recommended on April 16, 2002.

PEA Privacy Enhancing Agent

PII Personal Identifiable Information. Information that can be traced back to a specific individual user, e.g., name, postal address, e-mail address, telephone number, or Social Security number.

SAML Security Assertions Markup Language which defines how identity and access information is exchanged and lets organizations convey security information to one another without having to change their own internal security architectures.

UML Unified Modeling Language (UML) is an Object Management Group (OMG) standard for modeling software artifacts.

W3C The World Wide Web Consortium, the governing body for web standards. (http://www.w3.org/)

XACML Extensible Access Control Markup Language (XACML) is an open standard XML-based language designed to express security policies and access rights to information for Web services, digital rights management (DRM), and enterprise security applications.