Suggestions for Long term changes to P3P

For "Long Term Future of P3P" workshop, Kiel, Germany, 18-20 June 2003.
Giles Hogben, Joint Research Centre
Contact:giles.hogben@jrc.it cc to: marc.wilikens@jrc.it

Abstract

This paper discusses problem areas in P3P 1.0 and proposes possible solutions, focusing on longer term changes, which could form part of P3P 2.0. Key issues discussed are Compact Policies, APPEL, Consent Mechanisms, The Base Data Schema, P3P in Enterprise environments and P3P in Identity Management Systems.

1. Introduction

This paper presents a list of changes suggested in P3P 2.0 based on a detailed implementation and study of P3P 1.0 in the JRC's P3P Proxy [1]. Key issues discussed are Compact Policies, APPEL, Consent Mechanisms, The Base Data Schema, P3P in Enterprise environments and P3P in Identity Management Systems. The P3P working group recently issued a set of draft backwards compatibility guidelines for P3P 1.1. which state that "P3P 1.1 policies and policy reference files are fully compliant with the P3P 1.0 XML schema". Therefore any suggested changes which involve additional attributes or other changes to the XML schema are included in the scope of P3P 2.0, as well as issues orthogonal to the schema definition. A translation of the "recommended Extensions" elements from 1.1 into 2.0 Schema elements is however assumed.

2. Against Compact Policies

Problem:

Compact Policies should not be part of the P3P 2.0 spec for the following reasons.

1. They undermine the clarity of meaning established by the specification with respect to full P3P policies. The P3P 1.0 specification requires that "a site MUST honor a compact policy for a given URI in any case (even when the full policy referenced in the policy reference file for that URI does not correspond … to the compact policy itself)." The specification also states that "Compact Policies are summarized P3P policies that provide hints to user agents to enable the user agent to make quick, synchronous decisions about applying policy". Therefore at the same time as being binding in as far as any P3P statement *is* binding, they are also expected to provide a summary of the full policy. Some policies may be mapped directly onto a compact policy, but as compact policies rely on a handful of tokens to summarize a full policy and do not allow for granularity on the level of data types, they necessarily corrupt the meaning of many policies.
2. They were introduced for reasons of performance. Our studies have shown however that with efficient caching, parsing and evaluation performance is not a significant issue in the evaluation of full policies against preference sets.
3. In practice, compact policies have been used to replace full policies. As they provide a much smaller semantic space, they therefore degrade the value of P3P in these cases. Writers of Compact Policies are restricted to a relatively much smaller set of possible statements and therefore necessarily must be less accurate in their description. As the ability to be accurate about descriptions is already something that raises concern in corporate legal departments, this is not something to be encouraged.

Solution:

Remove Compact Policies from the P3P specification in P3P 2.0 and provide guidelines on caching and matching algorithms.

3. Need for a Preference Language

Problem:

It is unlikely that ordinary users will be willing to configure P3P agents. Work needs to be done in creating consistent and creative interfaces, which will give users control over privacy preferences. However, the matching of specific details in P3P policies will always be a matter better handled by data protection professionals than by end-users. In order to prevent such preferences being simply "baked" into browsers by browser manufacturers, a preference exchange language needs to be developed which can allow third parties to plug in recommended preference sets. This would also have the advantage of making preferences portable between different applications. APPEL has not been adopted as a preference exchange language by any major implementers because of a number of problems they have expressed.

1. Constructing the logic of matching patterns is too complex and ambiguous (see point 2.). APPEL provides a very idiosynchratic way of expressing logical connectives. With six logical connectives, this language is very difficult to write by hand, impossible to produce a simple interface for and incorporates a high degree of redundancy. For example the same match can often be expressed with several different connectives.

2. As stated above, the language allows for logically inconsistent preference sets. For example, the following rule looks for any information which is not the user's IP address or user agent string and blocks resources which ask for it.

```
<appel:RULE behavior="block">
<p3p:POLICY>
        <p3p:STATEMENT>
                <p3p:DATA-GROUP appel:connective="non-and">
                        <p3p:DATA ref="#dynamic.clickstream.clientip.fullip"/>
                        <p3p:DATA ref="#dynamic.http.useragent"/>
                </p3p:DATA-GROUP>
        </p3p:STATEMENT>
</p3p:POLICY>
</appel:RULE>
```

This RULE will cause a block behavior for the following web site policy (only relevant parts quoted),

```
<POLICY>
        <STATEMENT>
                <DATA-GROUP appel:connective="and">
                        <DATA ref="#dynamic.clickstream.clientip.fullip"/>
                        <DATA ref="#dynamic.http.useragent"/>
                </DATA-GROUP>
        </STATEMENT>
</POLICY>
```

but not for this one

```
<POLICY>
        <STATEMENT>
                <DATA-GROUP>
                        <DATA ref="#dynamic.clickstream.clientip.fullip"/>
                </DATA-GROUP>
        </STATEMENT>
        <STATEMENT>
                <DATA-GROUP>
                        <DATA ref="#dynamic.http.useragent"/>
                </DATA-GROUP>
        </STATEMENT>
</POLICY>
```

Note the presence of the "non-and" connective, which means - "only if not all sub-elements in the rule are present in the sub-elements of the matched element in the policy". This is true for the first policy snippet but not the second, which given that they have the same meaning is clearly unacceptable.

3. Developing a fast enough matching algorithm for a matching protocol which is unique to APPEL is too costly.

Solution:

Because buy-in from browser vendors is so crucial, discussions should be held with them so that future preference language specifications will take into account their requirements. However, a first pass at a solution would include the use of a standard query language for the condition matching part of APPEL. Instead of using APPEL's somewhat quirky connective system and recursive matching algorithm the rule condition could be specified by an XPATH [2] query (or by the time it becomes relevant, Xpath 2.0[3]). These query languages are designed to match arbitrary node sets with high efficiency. They have the advantage that developers are familiar with them and efficient algorithms exist to execute the queries. As it has become very clear that an XML preference language is unlikely to be written by anyone other than developers or ordinary users using a GUI, this is clearly the best approach. The specification of the GUI to be put on top of this could incorporate research from the user-agent translations group and work on ontologies suitable for end-users, currently being undertaken by JRC [4].

For example, a rule in this format, which would solve the above ambiguity problem would be:

```
<appel:RULE behavior="block" prompt="yes" promptmsg="Resource will use your
home info beyond current purpose ">
<appel:MATCHQUERY query=
"//DATA[not(substring(@ref,' dynamic.clickstream.clientip.fullip') or substring(@ref,'
dynamic.http.useragent'))]"
querylangauge="XPATH">
</appel:RULE>
```

It should be noted that the recent issue of the XPATH 2.0 [3] specification, which provides an even more powerful matching language, makes this an even more compelling solution.

4. Security Vocabulary

Problem:

The European directive specifies that adequate security measures should be taken to protect data (95/46/EC Article 17). However there is no means within P3P to express the level of security around personal data. The reasons for this are clear: state-of-the-art security measures are constantly changing. Furthermore, it is very difficult to define security measures in any meaningful way. It might for example be stated that a database is password protected, but password might in reality be "abc".

Solution:

There are several candidate schemas already in existence for classifying and describing security measures. It may be that these are able to provide some solution to this problem if incorporated within the P3P taxonomy. However, as mentioned above, any security taxonomy will either be too general to be useful, or would be out of date within a short space of time.

An additional solution, which may solve this problem, is therefore to provide the opportunity for third party security seals within policies. P3P already provides a placeholder for data protection seals within the DISPUTES element. However these do not relate to security measures, only to data practices. The specific provision of a security seal placeholder would allow for a validation by a third party which would not constrain expressiveness to a security taxonomy based around a changing and meaningless set of parameters. Instead, it would provide proof of a flexible and intelligent audit carried out by a reputable organization. It may also include a datestamp as an indication of the "freshness" of the seal. The expense of providing a meaningful seal may be a problem in itself. One solution to this would be to provide an alternative free text field for organizations without adequate resources to describe their security pratices.

Finally, we suggest that the incorporation of a framework for machine understandable audit trails (see section 8.) may also provide some solution to this problem, as it would provide the possibility for rapid and accurate assessment of security policies.

5. Consent Issues

Problem:

The EU's Article 29 working group has stated. "Internet users must have a real possibility of objecting … on-line by clicking a box"[5]. Any collection of personal data must have a specific opt-in mechanism - in other words, consent must be explicitly expressed.

Although P3P is able to check what a P3P policy states about consent, using the opt-in, opt-out attributes in the policy, it is not able to check that there is actually a mechanism in place for expressing consent. More specifically, the following could be provided;

- An integration with Xforms [6] to extract the semantics of consent boxes and validate claims of opt-in mechanisms. It would have to be investigated whether the mere presence of a check-box is sufficient to constitute an opt-in mechanism. It might be argued however, that the semantics of marking a check-box as an opt-in mechanism would carry some legal weight.

- Methods for expressing (possibly signed) consent. Although the requirements of the EU directives do not stipulate this, the specific requirement that users must have the option of explicit objection to data collection effectively requires that businesses can prove, in certain cases that consent was given. If some way of expressing signed consent were built into P3P, it would be a considerable aid to both parties and especially to businesses wishing to protect themselves against the consequences of disputes.

It may be argued that it is not within the remit of P3P to deal with the issue of consent, and that this should be addressed perhaps by the XFORMS group. However, consent for using personal data is an issue, which relates specifically to data privacy, and is independent of whether that data is transmitted through forms, or through for example, http headers. Therefore P3P is the ideal specification to include a mechanism for expressing consent.

Solution:

Here we outline a sketch of how such mechanisms might work. Full details would be a matter for the specification group.

Checking for an opt-in/out mechanism

a. There could be a specific attribute published within a namespace approved by the P3P specification, but mentioned within the Xforms specification (alongside other proposed attributes such as the policy reference declaration), which expresses in a machine readable way the fact a check box or other formfield is for expressing consent.s

E.g. <xform:checkbox ref="YesIDo" P3P:consentfield="yes">

This would have the important advantage of providing a standard syntax useable by all form systems for expressing consent.

Requesting signed consent

We considered the possibility of a mechanism for expressing consent by including, within a policy, an element specifying the name and various other specifications for a hidden form field to be added to a POST operation, containing a signed statement, as specified in the element.

However, this mechanism has several disadvantages:

It is attached to the form, and not the processing application. There is therefore not an authoritative relationship to the application which processes the data. For example many web shops use third parties to process their forms. These third party processors might find it difficult to control expressions of consent if they were managed by third parties'.
It is not ideal for the client to have to add POST fields to a form, there considerable opportunity for ambiguity. Also it is generally harder for a third party software vendor to alter the operation of an application (e.g. API) to do this than for example to alter http headers sent.

We therefore suggest that a mechanism could be provided for requesting and providing consent using http headers, which would also provide the option of asking for a signed consent.

In this case, an element would be added to the P3P policy similar to the following:

```
<DATA ref="user.home-info">
<CONSENTREQUEST method="httpheader" headername="consent1">
<DATAREQUIERED certificate="X.509" algotrithmtype="RSA"
minkeylength="128">I agree that my data in this form will be published on the
internet.
</DATAREQUIRED>
</CONSENTREQUEST>
<DATA/>
```

CONSENTREQUEST could be inserted within a DATA element to state that the collection of this type of data requires the consent of the user, and how this consent should be sent.

"method" - specifies that the consent should be specified using an HTTP header specified by the attribute "headername" - specifies the name of the header which should contain the signature data. The DATA element contains the statement which is required to be signed to express consent. In its attributes, it contains various requirements to allow for flexibility in the requirements for signature types.

Structure of message.

To be of any use, consent messages need to be stored in a structured way in the "back office" of the service provider. The most important requirement for the "back office" is that the message can be linked to the data which it provides consent in the case of a dispute. This requirement however needs to be set against the possible loss of privacy involved should the message be linked with a unique identifier.

Because of this latter consideration, it should be left up to the service provider to link the consent message with a unique identifier binding it to the information, such that the possible privacy losses contained in such an identifier are appropriate to the situation. For example if the subject is willing for their entire information to be retained indefinitely, then a hash of the all or part of the information may be used. However, if they are not, then this is not appropriate, because such a hash could later be used to perform data mining operations on sensitive information. In this case, a hash of some form of session id might be more appropriate. Another solution is a key escrow system, which could be used to unlock the identifiers by a legal authority requiring the proof of consent. This is overkill for most situations, but in situations where a mandate is being given for something very important, which cannot be done in person, it could be very useful. In either situation, the date of the consent may be taken from the http request headers.

One possibility for structuring of the messages themselves however is to express them according to an OWL [7] semantic model, such as that discussed in section 7.. For example, RDF statements could be constructed to formally express statements such as

"I am a data subject and I agree that the data objects transferred in this request may be transferred to third parties." (ontological terms underlined)

If such a consent statement were expressed using RDF statements it would carry more legal weight through this unambiguous and transparent semantics and would make management of different consent statements easier by making them easily processable by software agents.

6. Improvements to Data Schema.

Problem:

Considerable work has now been carried out in improving the P3P base data schema. An XML schema version and XSLT transforms to ensure backward-compatibility are now available [8]. This work has shown however that changes are required to the Data Schema, which are not possible with the strict requirement of backward-compatibility, which requires the possibility of a 1-1 transformation from the P3P 1.0 schema. The following problems however still exist with the XML schema version of the data schema.

1. Categorization of personally identifiable information: there is no way to simply specify whether a data type is personally identifiable, which is perhaps the most important category.
2. Clarified semantics: currently the schema creates 2 orthogonal systems of categorization - that is the categories and the data elements. These should be amalgamated for the sake of semantic consistency and simplicity of processing and expression. This would also overcome the redundancy which presently exists in the XML schema, whereby multivalent category attributes must be declared several times for what is essentially the same semantic. As XML schema strictly does not have a semantics, we would suggest eventually that the schema is expressed using OWL[7].
3.There are a number of errors in small points of detail. For example, the following category description from the base data schema is the closest we can get to a descriptions of the http header information.
http="Navigation and Click-stream Data, Computer Information"
However, http header information cannot be described by any of the terms in the sentence "Navigation and Click-stream Data, Computer Information ".

Solution:

The categories and ontology of the P3P data schema should be revisited and altered in the light of requirements gathered from Policy writers by actual studies of users writing policies. Such studies should not be prejudiced by technologists and should allow for the possibility of a greatly simplified schema. Tools for creating and validating customized schemas could also be provided. The possibility of creating a formal OWL(or similar) ontology of data types should be investigated.

7. Detailed study of ontology and useability studies.

Problem:

End-user studies carried out on the current P3P taxonomy are based on testing a number of alternatives which are pre-determined by technologists and getting users to choose the best ones or suggest modifications. This method is of course simple and cheap to implement, but it restricts the representation of privacy concepts to users to a set predetermined by technologists. Furthermore, alignment with legal principles has been thus far an informal process.

Solution:

Key concepts used in P3P 2.0, particularly in recommended user agent translations, should be based on situational testing which does not prejudge outcomes. That is, users should be tested on which words they prefer to use based on their experience of a situation, rather than be being presented with a choice between alternative predetermined descriptions. Such methods are outlined in [9]. This may result in the selection of entirely different user conceptual models, which may be based on metaphors rather than literal expression. Such metaphors should be aligned using formal ontological processes with legal documentation and expertise. The possibility of expressing P3P using formal ontological language such as OWL [7] should also be investigated as it would bring increased flexibility and expressivity.

8. P3P in the enterprise and audit trails

Problem:

P3P has sometimes been presented as an aid to the enforcement of data protection principles. However in its present form, it can only provide statements of companies' intentions about their data practices. This has no necessary connection with their actual practices so P3P has no enforcement power. In other words, P3P cannot guarantee that the promise matches the practice. There is also an inverse problem that companies may abide by the law to the letter, and yet not publish a p3p policy. Therefore what is needed is a way of applying P3P as a means of establishing and thereby enforcing actual practices.

Solution:

One solution is to use a language based on P3P within a system for automated audit trails. This solution can be compared to the solution adopted by restaurants, who wish to make clients trust their hygiene practices. They put the kitchen in full view of their customers. In the same way, given a sufficiently standardized system, perhaps based on P3P, servers could record their data processing events and security related events in such a way that authorized auditing agents could assess them in a measurable way against the regulatory standards and of perhaps additional standards of trust seals. The full details of such a system lie within the domain of developing an enterprise privacy language. However, a scenario is presented below which helps to view this set of extensions in a concrete way, and from this, extract some requirements for P3P 2.0.

Audit Trail Scenario:
1.User U1 submits their email address to company x1. This event is logged as: *"Data submission event, data type emailaddress:stored in database D, linked to unique ID"*
2. Query of database D1 by software agent x3.
At this point, the data can take one of 2 paths, which must be clearly distinguished:
A.Data viewed by a system user, U2 with certain legal responsibilities and perhaps risks, outlined in his profile, P1.

3.Profile P1 is of U2 who has been allowed to view the data.
4.P1 may contain information such as links to signed NDA's, commitments made by M, a trust profile etc...
5. Audit trail records that fields x4,x5,x6 for subject x7 are displayed to U2.

B. Data passed to another application.

6.Agent profile Pa contains
7. A set of commitments entered into by that agent as described in a P3P policy.
8. A pointer to how to find the audit trail left by that agent (anonymized versions may even be publicly available).

This is represented graphically in figure 1.

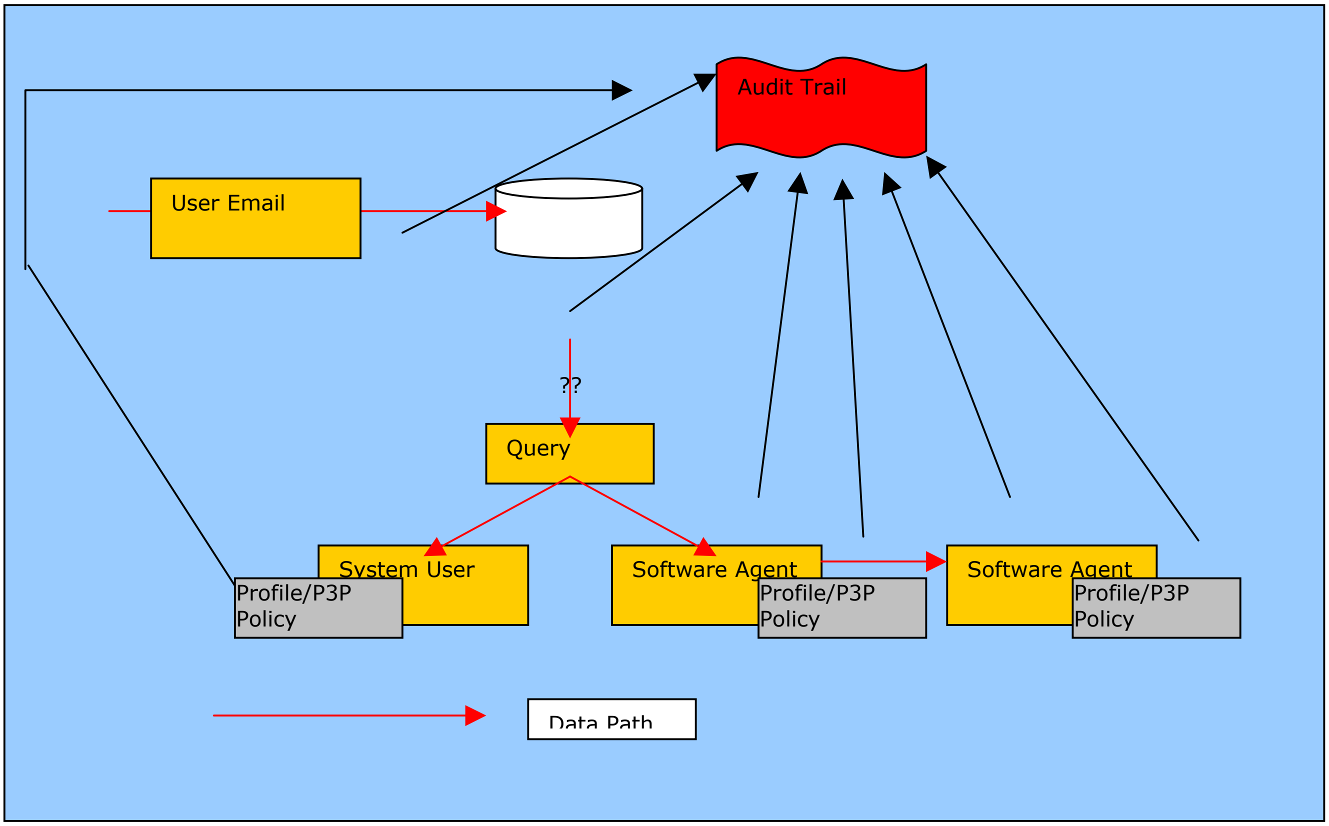

Figure 1.

An important feature of such a system is that any agent system A1 passing information to another agent system A2 must have a way of knowing whether A2 is also committed to recording audit trail information and where and under what circumstances this information could be accessed. Otherwise knowledge of data processing practices is effectively meaningless. In order to track real privacy practices rather than privacy promises such as contained in P3P policies, the information may not be passed to any systems which do not have a specified audit trail system.

What this requires of P3P policies for 3rd party data recipients is

1. A placeholder for attributes of audit trail giving
-commitments
-access conditions and locations.
-seals

2.Improved recipient taxonomy to allow expression of privilege profiles as in 2.A.

3.A taxonomy for creating audit trail logs (for example was data passed in encrypted form or not, was it placed in a secure environment.)

4. If competing systems exist, then there must be a way of distinguishing between them. That is, a system must be able to understand the meaning of
"if you release information I1, it will be passed from an environment which uses audit trail system AT1 to a system which uses audit trail system AT2"

9. P3P and Identity Management.

Problem:

P3P 1.0 makes no recommendations on how to link privacy policies to specific data transfer events, and how to make decisions around such events. The W3C APPEL note, which is discussed in section 3 above, makes recommendations on how to make such decisions. However, these recommendations are limited to only three basic behaviors. What is needed to make P3P into a powerful tool within e-business, is the ability to release data selectively based on privacy policies and the agent's level of trust in them.

To look at a specific example, the mobile device community has expressed interest in linking P3P with the CC/PP (Client Capabilities, Preferences Profile)[10]. In this case, it would be extremely powerful if a P3P enabled agent + Rule-base were able to reveal only selected device capabilities, basing a decision on the privacy policy and a set of capabilities, which the service might need to know. Most client applications would benefit from such a capability, if it were made easy to use and robust. When filling in forms, users generally reveal only what is necessary and if the users do not trust the entity with the information which it claims to require, they will not go ahead with the data transfer at all.

It should be mentioned that the ability to selectively release data is strongly connected with identity management, and therefore any developments in this area should be linked into research in this area.

Solution:

As this is an area where extensive further research is required, rather than describing a detailed solution, we will just outline the technical requirements of such a system, and briefly suggest their likely solution.

Technical requirements:

1. An ontology expressive enough to capture the various data types which might make up a composite identity (selective release of personally identifiable information). This has already been discussed in section 7 above.

2. Ways of linking that ontology to requests to databases from applications such as enterprise applications, Xforms and CC/PP based applications.

3. A rule language and user-interface expressive enough to allow selective release of information. This would most likely involve the definition of identities, in other words groups of information types, using a visual representation of a PII ontology and their linking to certain patterns recognized in policies. The identities would effectively become super-classes within the ontology.

4. A clear specification of what kind of information is being requested, which elements are optional, and which is required. Without this, the engine cannot decide what information to release in a particular case. As it stands, it would be very difficult for P3P to perform this function alone, because P3P policies are necessarily generalized between different resources and semantically they do not give any information about what is required on a particular page.

The underlying semantic structure of P3P policies is:

1. "whatever the resource this policy is applied to, if you give us information x, we will do y" (a hypothetical policy)

and NOT

2. "please send us information x for resource y" (an information request)

What is needed in this new scenario is to have both the above semantics. That is, if the second semantic, 2. (the information request) is provided by (e.g.) the Xform and linked in a granular way with P3P policies, this can provide enough information for an agent to make a decision. For example a particular XForm might be able to express the semantic

3. "I require your email - this email address will be processed according to P3P policy Policy1, which can be found by means x."

Policy1 will then express the semantic.

"Any Email addresses received by this application will be given to 3rd parties for marketing purposes."

On the part of P3P, it would simply require the capability to associate P3P policies to a more granular level than that of the resource. This should already be possible within P3P 1.1. In particular, in the case of Xforms, it requires P3P policies to be associated with individual form fields. If a more general specification can allow the association of policies with more diverse entities, this opens up the way for the application of P3P in other similar settings such as CC/PP, irc (chat) etc...

References

[1] JRC proxy full implementation of P3P http://p3p.jrc.it
[2] Xpath 1.0 W3C specification, see http://www.w3.org/TR/xpath
[3] Xpath 2.0 W3C specification, see http://www.w3.org/TR/2002/WD-xpath20-20020816/
[4] See JRC PRONTO project http://pronto.jrc.it
[5] Art 29 - Data Protection Working party: Recommendation 2/2001 on certain minimum requirements for collecting personal data on-line in the European Union; Opinion on P3P, 16 June 1998.
[6] Xforms W3C specification http://www.w3.org/MarkUp/Forms/
[7] Web Ontology Language, W3C specification http://www.w3.org/2001/sw/WebOnt/
[8] See http://www.w3.org/P3P/2003/03-xml-data-schema.html
[9] See White paper on the use of ontologies in PETs, Joint Research Center of the European Commission http://pronto.jrc.it/files/bestpractice.doc
[10] CC/PP W3C specification, see http://www.w3.org/Mobile/CCPP/