Profiles for the Situated Web

Lalitha Suryanarayana
SBC Technology Resources
9505 Arboretum Blvd.
Austin, TX 78759, USA
+1 512 372 5417
lalitha@tri.sbc.com

Johan Hjelm
Ericsson Research
Torshamnsgatan 23, Kista
SE-16480 Stockholm, Sweden
+46-8-585 347 99
johan.hjelm@era.ericsson.se

ABSTRACT

The World Wide Web is evolving into a medium that will soon make it possible for conceiving and implementing situation-aware services. A situation-aware or situated web application is one that renders the user with an experience (content, interaction and presentation) that is so tailored to his/ her current situation. This requires the facts and opinions regarding the context to be communicated to the server by means of a profile, which is then applied against the description of the application objects at the server in order to generate the required experience. This paper discusses a profiles view of the situated web architecture and analyzes the key technologies and capabilities that enable them. We conclude that trusted frameworks wherein rich vocabularies describing users and their context, applications and documents, along with rules for processing them, are critical elements of such architectures.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/ Hypermedia---*architectures*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services---*web based services.* I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods---representations, semantic networks. C.2.0 [Computer Communication Networks]: General---*data communications*.

General Terms: Design, Standardization, Languages.

Keywords

Situated-aware applications, web architecture, profiles, vocabulary, XML, CC/PP

1. INTRODUCTION

In less than a decade, the World Wide Web has evolved from a document hosting and rendering medium to one that is compelling and particularly suitable for interactive services that can be tailored to meet specific users needs. With the help of web technologies currently available, it is now becoming possible to conceive of application architectures that can inherently support varying degrees of personalization depending upon the amount of information obtained from the end user. On the one hand are architectures that enable a "one-size fits all" approach towards web based services. These require minimal data to be gathered about the end user environment in order to render applications.

On the other end of the spectrum are architectures that provide highly relevant, context-aware experiences that are very much targeted towards generating greater degrees of end user satisfaction. Such systems are facilitated by application and processing of knowledge that represents a highly granular description of the user's context. The latter signifies what is commonly termed in the computing industry as a "situated", "contextualized" or "adaptive" service.

The notion of context-based computing has been around for some time [1], [2], [3] and [4]. Early works in this area have defined context to be any information that can be used to characterize the situation of an entity. The situation is typically characterized by its physical and social environment[1]. It covers such aspects as facts pertaining to the physical and computation components of the interaction with the system, history of past interactions, personal information, etc. and opinions expressed as preferences and interests with regards to the social aspects of the environment. The latter are relatively static descriptions while the former encompasses dynamic attributes such as time, temperature and position relative to the user.

The emergence of mobile devices has made context-aware or situation-aware web applications particularly interesting. We identify a situation-aware service to be one where the access to the web application has moved from the traditional desktop and into the world - to the location at which people are trying to accomplish their tasks. We specifically distinguish a situated service from one that is customized or personalized in the amount of information about the environment that is communicated by the user and used by the web server to generate very targeted content. See Table 1. Content that is customized or specialized typically involves generating a specialized presentation and format that takes into account facts regarding the delivery context of the user [5]. Generation of personalized content involves additional processing. In addition to facts about the delivery context, personalization takes into consideration opinions about the user's preferences, tastes, likes and dislikes regarding the content or application itself. Such data may be determined by mining previous browsing behaviors or by explicitly requesting it from the user – a priori. Situated services are even broader in addressing the scope of awareness regarding the user's immediate environment. With up to the minute information about the relative presence, absolute location and other dynamic data, the web application can intelligently generate an experience that is more relevant and germane than is possible with personalization or

[1] Technically speaking, situation is a subset of the context. The latter also includes, in addition, linguistic variables such as sociolects the discussion of which is beyond the scope of this paper.

WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

customization. In the long run, such applications benefit both, the user as well as the provider.

The focus of discussion in this paper is on the role of profiles within the World Wide Web architecture in the context of situated services. First we introduce the types of profiles necessary to enable the situated web. We then analyze relevant technologies, citing examples, and capturing some of the associated issues and challenges with regards to the communication and processing of profiles. However specific discussions regarding mechanisms for gathering, inputting, updating and storage of profile data are considered outside the scope of this paper.

Table 1. Dimensions and Implications of Situated Applications

Criteria	Customized app	Personalized app	Situated app
Information about the user	Capabilities about the delivery context including user preferences regarding presentation	+ personal data regarding habits, behaviors, tastes and preferences	+data regarding the immediate environment such as time, temperature, position
Nature of information	Static	Static	Dynamic
Granularity of tailoring	Coarse	Medium	User-specific
Granularity in practice	Everyone with the same device features and access mechanism will likely receive the same presentation	Varies based on web site implementa-tions	It is highly unlikely for two users to have the exact same profiles
Personali-zation mechanism	Documents typically modified by templates	Database logic; or XML components combined and transformed, using templates	Database logic; or system based on rules and facts. The role of application and user profiles
Benefits to user	Optimized user experience in terms of look and feel, interaction	Personalized to individual preferences	Highly relevant experience – reduces information overload
Benefits to information provider	Information (content) usable	User retention, e.g. in e-commerce	Same as for personalization

2. SITUATION-AWARE APPLICATIONS – A PROFILES VIEW

Meta data description about a system, component, application, object or environment can be assimilated and modeled into a structured collection or set termed as a profile vocabulary. Like in the database and AI worlds, the taxonomy or organization of these elements is referred to as the schema or ontology. The elements generally pertain to a specific application and the vocabulary identifies the constraints on the elements of objects within the application. In recent years, XML technology is being commonly used for encoding the schema and profile. The use of XML namespaces [6] makes it possible to create application profile instances that incorporate elements from multiple vocabularies that have been developed and standardized by different communities. RDF [7], [8] as an application of XML provides a coherent structure for the expression of semantics and uses XML syntax for transport. Complementary XML technologies such as Xpath [9] and Xpointer [10] can be effectively leveraged to parse, traverse and query profiles.

In this paper, we review the technologies associated with enabling an end to end profiles based situated application aware architecture. We identify the following components as core elements of such architecture:

- A detailed description about the environment in which the service is to be rendered or delivered. Referred to as the user profile, the capabilities of the context may, at any given instance, be distributed across the web.
- A means for communicating or transporting the distributed user profile information to the origin server across the web. Since some of the characteristics of the user profile for situated services are dynamic in nature, the values must be sought in real-time and be contextual.
- Trust mechanisms to ensure security, integrity and privacy of the profile data.
- Meta data assertions regarding content, media, and applications residing on the origin server. This is termed as the content profile or application profile.
- Tools and logic at the server end with regards to processing the information received and generating a user experience[2] that is specifically targeted for the context that was received.

The rest of the paper discusses each of these in detail.

2.1 User Profile

As mentioned earlier, the user's environment is the context in which the web content or application is rendered. It is characterized by facts and opinions about the user and his/her immediate situation [11] as identified in Table 2, and is a collective set of attributes gathered from different sources on the web. The distributed nature of profile information makes it difficult for one entity to store and maintain the user profile information, necessitating a standardized framework for asserting and assembling profiles.

2.1.1 User Profile Frameworks and Vocabularies

The situated web architecture involves distributed profile management and thus requires dynamic composition of the user profile in real-time from multiple segments, which may be distributed across the web [13]. While capabilities such as HTTP Queries or Post, web services (using well known registries) and XML queries enable retrieval of individual profile segments, we still need to have standardized ways of making assertions that will assist automated composition and subsequent processing of the profile. The issue is further complicated when different communities undertake efforts to specify profile vocabularies for vertical components of the profile.

[2] In this discussion the term user experience is used to imply the content object and data itself, as well as the interaction and navigation functions that are associated with the content, and the final presentation look and feel.

Table 2. Key Components of User Profile

Attributes	Description and examples	Source
Personal facts and preferences	Personal information such as address, payment information, demographics (gender, age). Persona attributes in terms of social context and preferences, habits, interests, tastes.	Explicitly from the user. Implicitly from tracking mining user behaviors from historic logs
Capabilities of the access mechanism and delivery context	Physical characteristics of the hardware and software computing environment (screensize, markups supported), capabilities of the transport mechanism	Hardware and software vendors, network providers, user
Situation data (also known as mobility profile in the wireless environment)	Peripheral awareness – dynamic data regarding user's position in absolute or relative terms, event information including time of day, weather, temperature, presence etc.	Sensors in the device, network, environment; third party providers; information databases such as SKiCalendar
Permission preferences	Opinions and policies regarding who can access what information and when, privacy attributes, etc. Typically also includes information regarding the user's relation with the information provider[3]	User

Composite Capabilities Preferences Profile (CC/PP) specifies an XML and RDF based framework to address this need [14]. It provides a machine understandable and interoperable basis for meta data descriptions of profile information using XML namespaces that allow for association of device capabilities and terminal preferences from multiple vocabularies. The use of Defaults attribute allows multiple profile instances to share common device parameters (Figure 1).

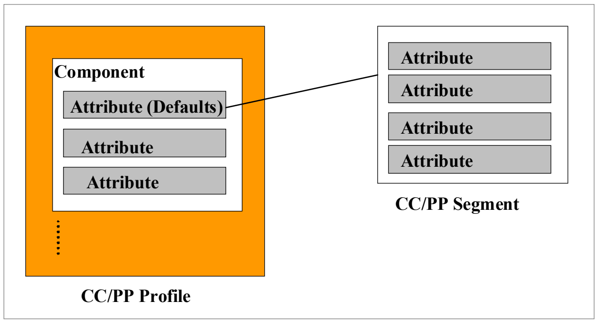

Figure 1. Structure of a CC/PP Document

Consider a hypothetical example, two users accessing the web each from his/her Super Device terminal, model 4216, which does not support color and comes with a 1M memory. However one of the users has subsequently upgraded the memory to 2M. Except for this memory upgrade on one of the terminals, the two device profiles will have similar hardware characteristics that are retrievable from the vendor site (SuperDev.com). Identifying these capabilities as Defaults and externally referencing them in the profile allows both users to assert the same hardware capabilities for their terminals. Additionally, CC/PP allows the user to also assert the upgrade in memory by specifying the attribute as an override parameter with the new value (2M). We illustrate this in Figure 2 using example attributes from the WAP User Agent Profile vocabulary [15].

Device capabilities Profile communicated by the Client (verified with RDF Validation Service available at http://www.w3.org/RDF/Validator/)

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
xmlns:prf="http://www.wapforum.org/profiles/UAPR
OF/ccppschema20010330#">

<rdf:Description
rdf:about="http://FooInfo.com/myDevProfile">
  <ccpp:component>
    <rdf:Description
rdf:about="http://FooInfo.com/myDevProfile#Super
HrdwrCpblty">
    <rdf:type
rdf:resource="http://www.wapforum.org/profiles/U
APROF/ccppschema20010330#HardwarePlatform" />
     <ccpp:Defaults
rdf:resource="http://www.SuperDev.com/4216" />
        <prf:Memory>2M</prf:Memory>
</rdf:Description>
</ccpp:component>

<ccpp:component>
<rdf:Description
rdf:about="http://FooInfo.com/myDevProfile#Super
BrsrCpblty">
<rdf:type
rdf:resource="http://www.wapforum.org/profiles/U
APROF/ccppschema20010330#BrowserUA" />
<prf:BrowserName>somehtmlbrowser</prf:BrowserNam
e>
        <prf:HTMLVersion>3.2</prf:HTMLVersion>
   </rdf:Description>
  </ccpp:component>
</rdf:Description>
</rdf:RDF>
```

Profile segment for default hardware capabilities available at http://www.SuperDev.com/4216 (verified with RDF Validation Service available at http://www.w3.org/RDF/Validator/)

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:prf="http://www.wapforum.org/profiles/UAPR
OF/ccppschema20010330#">
<rdf:Description
rdf:about="http://www.SuperDev.com/4216">
<rdf:type
rdf:resource="http://www.wapforum.org/profiles/U
APROF/ccppschema20010330#HardwarePlatform" />
<prf:Model>SuperDev4216</prf:Model>
<prf:Memory>1M</prf:Memory>
<prf:ColorCapable>No</prf:ColorCapable>
</rdf:Description>
</rdf:RDF>
```

Figure 2. Sample CC/PP Device Profile using the WAP UAProf Vocabulary

[3] Per discussions in the 3GPP OSA activity [12].

CC/PP is extensible and generic enough to be leveraged for assertion of user profile information beyond simply device capabilities [16], [17]. In previous work [18], [19], CC/PP has been used to describe personal information such as demographic and preferences data. We provide an example in Figure 3 using a simplified version of the P3P Base vocabulary [20].

Profile segment for User Demographics (verified with RDF Validation Service available at http://www.w3.org/RDF/Validator/)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
       xmlns:ccpp="http://www.w3.org/2000/07/04-
ccpp#"
       xmlns:usrdmg="http://www.w3.org/2002/01/p3
prdfv1#">
<rdf:Description
rdf:about="http://FooInfo.com/MyDmgrphicProfile">
<ccpp:component>
        <rdf:Description
rdf:about="http://FooInfo.com/MyDmgrphicProfile#my
self">
        <rdf:type
rdf:resource="http://www.w3.org/2002/01/p3prdfv1#u
ser" />
       <usrdmg:user.Gender>Female</usrdmg:user.Ge
nder>
       <usrdmg:user.name>Foo
Bar</usrdmg:user.name>
       <usrdmg:user.home-info.postal.street>Cool
Street </usrdmg:user.home-info.postal.street>
       <usrdmg:user.home-info.postal.city>Cool
Town </usrdmg:user.home-info.postal.city>
       <usrdmg:user.home-
info.postal.stateprov>Cool Province
       </usrdmg:user.home-info.postal.stateprov>
<usrdmg:user.jobtitle>supermom</usrdmg:user.jobtit
le>
    </rdf:Description>
</ccpp:component>
</rdf:Description>
</rdf:RDF>
```

Figure 3. Sample CC/PP Demographic Profile using Attributes from the P3P Base Vocabulary

The IETF CONNEG is another framework and vocabulary that describes device capabilities and other media features for applications such as fax [21], [22]. Using MIME types rather than XML, it defines syntax for relational style expressions. The addition of new media types involves a formal process of registration with IANA, making it relatively difficult to extend. Creating URN namespaces for the IANA registries would allow media tags from the CONNEG vocabulary to be directly used in CC/PP profiles, thus enabling interoperability of CC/PP with CONNEG.

Non-CC/PP based prominent XML vocabularies include CPExchange [23] and P3P[4] [20]. CPExchange provides an information model for business data and associated privacy preferences meta data. The business data schema and vocabulary define a base set of profile elements that describe people and organizations that have a business relationship with a particular company. Parts of this vocabulary could be effectively used in asserting user profile information. The main intent of the Platform for Privacy Preferences (P3P) schema and vocabulary is for the purposes of asserting privacy preferences with regards to collection of personally identifiable information by web sites. The schema is also being considered for associating privacy preferences with information entered by users through web based forms[5].

Similarly there have been efforts to define vocabulary for calendar data such as from iCalendar which can be used to assert events based situation parameters [25], [26] in a machine understandable format. Another major development has been the Conneg vocabulary that defines feature tags for describing capabilities and preferences of display, print and fax systems [27].

2.2 Application Profile

Application or document profiles provide a means for describing the characteristics of the content or resources that are to be rendered to the user agent. Like user profiles, an application profile comprises of one or more profile segments, each of which can be associated with assertions regarding a specific resource or object that is to be rendered, or a collection thereof. Thus the ontology structure of the profile would reveal a profile segment that applies to a whole set of similar documents, another segment that is specific to a particular subset of documents, and yet another segment that is applicable only to one particular document. In Figure 4, the three profile segments would need to be combined in order to generate a document profile for object C.

A profile segment includes assertions regarding one or more characteristic of the object or application, such as the associated media types, encoding methods, policies over the content with regards to validity, cacheability, digital rights, etc. Information about the author, title and attribution and preferences, policies or opinions regarding manipulation of the content by third parties [28], and transformation rules thereof, could also be included as semantic hints. Situation-aware applications would additionally require semantic assertions about the user navigation, interaction logic and associated data model for the purposes of temporal and positional relevance. Consider for example an interaction logic (implemented as JSP bean or Javascript, etc.) such as one necessary for collecting user's credit card details. Different variants of this logic would be executed in different situations. The decision (rules) regarding which variant to invoke would dynamically depend upon the facts presented in the user's profile vis a vis the logic meta data. Such granularity of characterization ensures high quality mass customization of the content or business logic according to the situation of the user, and enables a scalable solution for application authoring and management at the server. Figure 4 illustrates the use of the Dublin Core element set [29] to describe attributes of an HTML document.

As with user profiles, application profiles are best expressed as XML or RDF schemas since they inherently enable extensibility and interoperability. The use of common technology helps

[4] The W3C P3P Working Group has recently published an RDF representation of the P3P abstract information model. See http://www.w3.org/TR/p3p-rdfschema/ for details. The interoperability of RDF and XML schemas is discussed in Section 5 of this paper.

[5] XForms are an XML forms technology that can be used to provide a vehicle by which a user can submit personal information in a user profile to a web server in the form of an HTTP Post [24].

automate the process of parsing and correlating the two profiles, which together form the tenets of the semantic web. As we shall see in Section 4, there has to be a direct correlation between the vocabulary of user profiles with that of application profiles in order that rules based determination regarding the relevance of a content object in a given context can be carried out. Such common vocabulary and framework are necessary to enable communication and exchange of profiles between heterogeneous systems and applications.

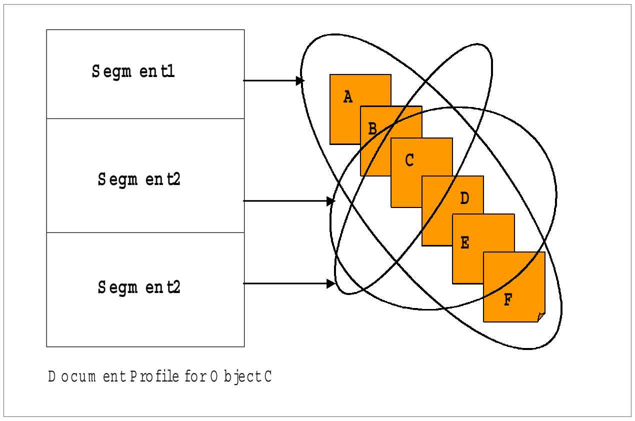

Figure 4. Composition of a Document Profile

Early work in this area suggests the use of external annotations that can be attached to raw resources or intermediate or final versions thereof. [30] has proposed an RDF based framework that includes a vocabulary for transcoding and a syntax to annotate already published HTML documents. The World Wide Web Consortium's HTML Working Group also has developed preliminary requirements such as for XHTML in this regard [31]. However no formal vocabularies have been standardized to date. Most of the implementations are internal to vendor systems and as such there is no measurement of the extent of correlation of these proprietary profiles to that of user profiles.

2.3 Transporting Profiles

Clearly, the profiles must be processed together in order for the user to receive an experience that is situation aware. This implies that either the user profile must be communicated to server or the application profile, to the user agent (or device) or both, the application as well as the user profiles to a third party processing server[6]. The HTTP 1.1 protocol [32] has specified user side and server side content negotiation as viable options for generating customized presentation. As such it is conceivable to have implementations that enable user agent side content negotiation in cases where the user agent has sufficient processing capabilities and there are no limitations anticipated on the network in order to download multiple variants of the presentation to the terminal. This method ensures privacy of the user data, but obviously leaves the content providers guessing about the suitable media types that should be sent to the client, and may even result in the user receiving unusable experience with the content. Since most situated services involve mobility[7], the processing of the profiles is constrained to occur at either the origin server or at a trusted third party intermediary or server. As we shall see later, it is also possible to distribute the content adaptation processing to multiple servers, each specializing in specific type or types of content transformation.

An Application profile segment (verified with RDF Validation Service available at http://www.w3.org/RDF/Validator/)

```
        <?xml version="1.0"?>
        <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:app="http://superdocs.com/apps/schemas-1.0#"
xmlns:ccpp="http://www.w3.org/2000/0704-ccpp#"
xmlns:dc="http://purl.org/dc/elements/1.0/">

<rdf:Description
rdf:about="http://www.superpark.com/pub/cooltown/k
idsplay.html/">
        <dc:title> The CoolTown Park for Kids
</dc:title>
        <dc:creator> CoolAuth </dc:creator>
        <dc:subject> nature; kids; parks; play
</dc:subject>
        <dc:date> 2001-11-06 </dc:date>
        <dc:format>
           <rdf:Bag>
                    <rdf:li>text/html</rdf:li>
                    <rdf:li>text/xml</rdf:li>
           </rdf:Bag>
        </dc:format>
        <dc:language> en </dc:language>
        <dc:description> This document describes
how the park is a favorite place for children of
all ages that enjoy playing amidst nature.
</dc:description>
        <app:clientCapability>
           <ccpp:Defaults>
             <rdf:resource>
                 <rdf:Bag>
     <rdf:li>http://www.SuperDev.com/4216</rdf:li>
     <rdf:li>http://www.CoolDev.com/6304</rdf:li>
     <rdf:li>http://www.FictDev.com/9912</rdf:li>
                 </rdf:Bag>
            </rdf:resource>
          </ccpp:Defaults>
        </app:clientCapability>
</rdf:Description>
</rdf:RDF>
```

Figure 5. Sample application profile in RDF using attributes from the Dublin Core and CCPP Schemas

We therefore limit the rest of this discussion to the transport of user profile information to a web server. Indeed we note that there has already been much work to date in this area. The issue of trust associated with the transport and use of the user profile are discussed towards the end of the paper.

[6] A viable scenario in which both user and application profiles are transported would be a web service that specializes at processing and correlating the two profiles.

[7] Obviously location or position is a critical parameter in describing a situation.

2.3.1 HTTP Header mechanisms

Rudimentary information regarding user agent and device capabilities can be conveyed by using the query part of a URI in the HTTP request. It is also possible to use specific HTTP 1.1 headers (see Table 3) [13] to communicate the information. Unfortunately the format of these headers is non-standardized, and subject to vendor implementations, making server side content negotiation unreliable and error prone. Furthermore, the limitation on size restricts the amount of profile information that can be conveyed via these headers, making them unsuitable for the situated web**.**

HTTP 1.1 also allows communities to specify and implement new headers. While this can be used to convey situation information, most of the time there is considerable effort involved in evangelizing the existence and benefits of such headers, which could potentially hinder widespread adoption. One exception is the Cookie header, which is being used by most personalization-based web sites today for the purposes of retrieving user profile data that is proprietary to the provider. Cookies appear to aid the security of profile data in transit, but there are privacy issues for the user to consider, obviously not making them very suitable for conveying the context since such information is also required to be dynamic. While insufficient by themselves, cookies can be augmented with HTTP headers and such, to communicate a richer set of profile attributes to the server.

Another way to incorporate headers is to use the HTTP Extension mechanism [33], which has been proven by CC/PP as a means of conveying the capabilities information from terminals such as WAP devices [34] and intermediate proxies. But the lack of unanimous support for HTTP-Exchange protocol in the vendor community implies the unlikelihood of this method to be a defacto means of extending HTTP headers. The WAP Forum has since preferred to specify simple user agent profile headers in wireless profiled HTTP (W-HTTP)[8] to accomplish this [15].

Table 3: HTTP1.1 Headers used for Capabilities Negotiation

HTTP1.1 header	Description
User Agent	Identifies the browser making the request. The format is not specified.
Accept	Identifies the MIME (media) types acceptable to the browser
Accept-Charset	Lists the character sets supported by the browser
Accept-Encoding	Identifies how content delivered to the browser may be packaged (encoded)
Accept-Language	Lists the natural languages accepted by the browser and user

2.3.2 SOAP

A promising way to transport an XML encoded user profile is by encapsulating it into Simple Object Access Protocol (SOAP) message binding over HTTP. SOAP is a general-purpose peer-to-peer messaging protocol for exchanging structured and typed information within a distributed and decentralized environment such as the web [35] making it suitable for communication of CC/PP information. It requires the various situated web entities (viz; clients, servers, proxies, intermediaries, etc.) to be SOAP-aware. The RDF profile can be conveyed to the origin server as an optional SOAP header block or a mandatory body block within a SOAP envelope included in the HTTP payload. The processing requirements for profiles sent via SOAP are similar to those when communicated directly over HTTP. For example, encapsulating the profile within SOAP header blocks enables SOAP intermediaries[9] to process or modify the CC/PP information just as they would if CC/PP were implemented directly over HTTP. It also becomes possible to address portions of the profile to specific value-added entities. Alternately, SOAP can also be used for retrieving profile segments (such as default hardware capabilities of a terminal) that are cached at web repositories (e.g. vendor web site – SuperDev.com in our example above) and advertised as web services via centralized UDDI registries. Leveraging security capabilities of HTTP along with those of XML[10] makes the case for SOAP-based profile communication, a very compelling one indeed! However much work remains to be done in terms of formalizing SOAP as a standard transport mechanism for CC/PP.

2.3.3 Other

The Location Interoperability Forum has specified APIs for transport of position information via the body of an HTTP Post message (HTTP Post) by encapsulating it in an XML document [37]. Although a point solution, it can be used in conjunction with profile frameworks transported over HTTP, but would require the origin server to implement multiple (possibly incompatible) means of gathering the situation profile data.

3. PROCESSING MODELS

A user's experience with the situated web depends, to a large extent, upon the components rendered by the application server: the digital data objects themselves (be it images, text or scripts) the appropriate navigational logic that combines and interweaves those objects, and the presentational aspects (such as markup, style sheets) associated with the ultimate look and feel of the application. These in turn depend upon a number of implementation factors - the software and hardware architecture associated with the application, the nature of tools and technologies used, and the overall content development set up. Thus server side situation specific authoring and processing can take the form of content generation, transformation, adaptation, selection or a combination of these. Truly a broad area with significant prior work, and several proprietary implementations are already in existence [38], [39], [40], [41], [42]! Some of these have specialized exclusively for targeting thin clients. We limit our discussion here to server side processing models that leverage profile mechanisms using XML technologies.

The separation of content from structure, navigation and style, combined with application metadata identifying immutability, replaceability and validity, helps the user profile to be associated with the various elements of the content generation and adaptation process [28]. The user profile, as we have already seen, consists of multiple components, some of which would likely be distributed

[8] termed x-wap-profile and x-wap-profile-diff.

[9] Also identified as SOAP processing nodes along a message path between the SOAP sender and ultimate receiver. Note, however, that SOAP intermediaries are not the same as HTTP intermediaries.

[10] Such as HTTPS/SSL, IPSec and the use of XML digital signatures to sign SOAP messages [36].

across profile repositories and caches on the web. Depending upon the application, it would obviously be necessary to compose and parse the profile facts or portions thereof, in order to extract the situation information along with presentation and personalization requirements. Operating on a vocabulary for rules, a rules interpreter would then apply a set of service logic specific rules (a policy which can also be referred to as a rules profile instance) against the information, drawing inferences by correlating it with the profile associated with the content (aka the application profile). In doing so, suitable application profiles would also need to be dynamically composed (or reconciled) and parsed as appropriate. Several steps must be executed. The content objects relevant to the data model should be identified in the context of a suitable navigation logic, the appropriate variant of each, chosen or generated, then assembled together, marked up and styled to ensure a presentation consistent with the situation.

Clearly, it is impossible for the rules inference engine to carry out this task if there is not a tight correlation between the application meta data descriptions and the user situation description. Furthermore, a standardized markup language such as RulesML [43] to express rules and policies regarding the service logic would ensure the deployment of standard rules engine in a semantic web environment. If application communities were to define vocabulary for common used rules, the burden of the content author and provider would ease significantly due to the ability to plug and play rules from multiple vocabularies, ensuring a fairly consistent and uniform user experience regardless of the application interacted with. The work has barely begun in the industry but we are optimistic about the influence of the semantic web in enabling this processing paradigm.

Profile segment Describing Weather Data (verified with RDF Validation Service available at http://www.w3.org/RDF/Validator/)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
  xmlns:exp="http://www.weatherinfo.com/RDF/exp-
schema#">
<rdf:Description>
    <exp:Date>Today (Mon) </exp:Date>
    <exp:Area>CoolTown </exp:Area>
<exp:Temperature>88 farenheit</exp:Temperature>
<exp:Situation>Sunny</exp:Situation>
</rdf:Description>
</rdf:RDF>
```

Figure 6. User Profile Fragment for current Weather in CoolTown based on RDF schema in [17]

We illustrate with an example, drawing upon the profiles (Figures 2, 3 and 4) illustrated in Section 2. Foo Bar is a mom living in CoolTown with fragments of the situation also described in Figure 6 below. She is interested in finding a park for her child. A set of rules along the lines of "if the user *is* a mom *and is* querying for park information, *and* the Temperature *is* between 80 and 94 farenheit, *then* read Home-info *and* compare to all documents whose title *is* Cooltown park *and* subject *is* play, *then*, if the browser *is* HTML *and* one of the documents *is* described in HTML, then if colorCapable *is* ….." [18]. This set of rules in pseudo-logic appears to be similar to those used in database operations today. However it is possible to perform much more complex operations if the rules are encoded in a formal language such as Prolog and supported by a robust inference engine. The rules and policies can be linked together suitably to generate the desired output.

Another technology that is critical in enabling content transformation is XSLT [44], [45]. Not quite a scripting language, it operates in batch mode with the operations irreversible. Using Xpath for addressing makes it possible for processing portions of the XML document tree in lieu of the user context. Multiple transformations can be applied on an XML object, making it suitable for situation-aware applications. However much consideration is necessary with regards to the processing logic since the order in which the transformations are applied impacts the end result. The rules in the above example can be expressed in XSLT, by establishing the scope of a set of variables using assertions in the profiles, and then applying these to deduce the output for presentation to an HTML browser.

4. VOCABULARY DEVELOPMENT

It should thus be clear that user profiles and application profiles together play a critical role in enabling situation-aware applications on the web. It is also obvious that the development of suitable profile vocabularies is vital to this end and XML and RDF provide an interoperable infrastructure for such semantic web implementations. XML and associated technologies allow content providing communities to share the authorship of resource vocabularies and meta data schema development, thus easing the adoption of profiles on the world wide web. The Unified Modeling Language (UML) [46] offers a standardized graphical modeling tool that can be effectively used by the diverse stakeholders to design profile vocabularies. In doing so, the community can initially clarify the terms, relationships and constraints of the elements without worrying about syntax and other implementation issues for mapping into XML and RDF. [47] describes one such mapping of RDF-Schema into a subset of the UML framework.

Profiles based architecture for situated services require the suitable vocabularies for both, rules as well as facts. Vocabularies defining facts have been defined in a number of areas, most notably the capabilities profile for WAP terminals (UAProf) and the Dublin Core element set pertaining to electronic documents and resources. There haven't been as many vocabularies for stating rules (or policies) as there have been for facts. The P3P rules language, APPEL, is expected to identify user policies for how different elements of privacy information should be shared [48].

Vocabularies describe a shared understanding of an information space. Our experience with designing and evolving the WAP User Agent Profile vocabulary suggests that vocabulary development is as much a social process as it is a technical one. Designers intending to develop vocabulary ontologies for their applications must be cognizant of the existence of other relevant vocabularies. These vocabularies could likely be leveraged or extended to provide the functionality that is intended, as opposed to designing a brand new one that includes some of the attributes that are already incorporated into an existing one. The extensions can take the form of assertions in a new namespace (vocabulary) that can be suitably integrated with an existing vocabulary in the context of a profile. Complementary RDF and XML schemas that are linked together would benefit from defining the vocabulary in both, RDF as well as XML, and then leveraging the rich semantic support in RDF with the structural, cardinality and data-typing constraints offered by XML schemas [49]. RDF's capability of multiple inheritance also allows new attributes to specialize from, in some cases, multiple vocabularies, inheriting semantics and constraints from other sources[11].

We see the immediate need for the content development and management communities to come together to define standard vocabularies suitable for document and application profiles that can be consistently correlated with the user profiles. Similarly, there is an apparent need for the development of vocabularies for formal rules-based methodologies that can operate on the profiles. While formidable tasks by themselves we believe that these efforts are crucial to providing enhanced user experiences that are conceivable with contextual awareness.

5. THE ROLE OF TRUST

The very nature of the data about the user that is being communicated to the situation-aware application provider necessitates the formation of a trusted relationship between the user and the provider. Such a relationship can be ad hoc, limited to one transaction or could be based on contractual agreements that span multiple interactions. The trust hinges upon bi-directional communication – the provider disclosing intentions with regards to the use of the profile information, and the user asserting preferences, policies and authorization regarding the access, use, storage, disposal and sharing of personal profile data[12]. Furthermore, the store and forward protocols underpinning the Internet architecture are a cause for concern about the security and integrity of the profile as it is transported over the web. The matter is even more complicated due to the participation of network and edge intermediaries. Thus far we have not encountered any one totally comprehensive solution that addresses all the needs for the situated web. However, there have been numerous industry efforts to solve portions of the broad security-privacy-trust maze that might be applied to personalization and contextualization. We discuss some of the prominent ones here.

5.1 Privacy

One can argue whether or not or which if any, attributes of the user profile are to be considered private. Clearly portions of the personal information are considered personally identifiable data (PII). Others may not be so, but the likelihood of them also being PII when combined with other attributes in the profile is high. Thus privacy-enhancing technologies become a critical element of the situated web. The platform for privacy preferences (P3P) [20] is a user empowerment technique that provides an XML based solution to enable web sites to assert their privacy policies. As a result, the user can make informed choices regarding disclosure of personally identifiable information at that site. If the site's policy were unacceptable, the user would likely not share critical profile information in which case the user experience of a situation-aware service would be sub-optimal. P3P does not provide a mechanism for policy enforcement although privacy seal organizations provide just that function. The onus of establishing and maintaining a trustful relationship with the user thus lies on the provider. In practice, users seem to allow for the tradeoff between loss of privacy and gain of convenience achieved by personalization. We would anticipate similar behavior for situation-aware services as well. Regardless, it is likely that P3P policies would have to accommodate data elements from multiple data schemas or vocabularies.

5.2 Security

The reliability, security, confidentiality and integrity of the profile information are all also critical elements of the trust framework. No part of the profile should be compromised, adulterated or maliciously modified in anyway during transit. Eavesdroppers and sniffers should be prevented from gaining access to the user profile. This also applies to the situation-aware content generated by the server that is transported back to the user in response to the request. Depending upon the configuration, complementary approaches could be implemented together at multiple layers so as to minimize impact of potential maliciousness. For example, XML signatures [47] and XML encryption [48] are candidates for ensuring integrity and confidentiality of portions of or the whole digital content including XML. Other application layer options include S/MIME and PGP. Lower layer solutions include the use of TLS, HTTP and IPSec over TCP/IP. Note that some of the end to end solutions may prevent intermediaries from participating in the situated services food chain.

6. ROLE OF INTERMEDIARIES AND PROXIES

Thus far, we have mostly presumed a simplified architecture of the situated web. With the user on one end and the application provider on the other, we have ignored the role of intermediate network and edge elements like caching proxies, firewalls, gateways and such in the generation transport or processing of profiles. However these elements abound on the web and cannot be ignored in terms of the impacts on the architecture. In fact there have been several efforts in the industry to standardize potential functions and capabilities that can be enabled by these systems, and business models have emerged in support of these features[13]. Most of the focus has naturally been in the area of customization of the presentation and personalization of the content prior to rendering to the end user device. Such intermediaries, specializing in broad or niche transformation, must be able to assert attributes pertaining to their capabilities. Proxy profiles thus contribute to both, user as well as application profiles. As part of the user profiles, intermediaries assert facts about their capabilities, limitations and preferences, and transport them along with the HTTP request[14]. When applied in conjunction with application profiles, they describe the embedded rules and policies with regards to content adaptation.

The development of the CC/PP framework and associated transport protocols together offer a simple means for proxies to communicate capabilities data to the origin server. In parsing the composite profile (that may contain assertions from multiple intermediaries), the challenge is one of determining the order of processing of the various profile segments, which could potentially impact how the rules of content generation or

[11] XML schemas do not have any concept of inheritance but allow for Types to be extended or restricted into subTypes.

[12] Such as with third parties, in the case of niche web services

[13] A transcoding proxy and the WAP method proxy are good examples.

[14] Typically, the existing headers are either modified or additional ones introduced to convey this information.

adaptation are applied, and thus also the end user experience. Standardization efforts have also been recently initiated in the IETF to develop a framework for personalization of content and services (FPCS) within the bounds of an open pluggable edge services (OPES) framework. The intent is for caching edge and intermediary systems to be the vehicle for content personalization [52].

While the role of intermediaries for personalization and customization of content in the path of the request or outside are fairly clear, the extent to which they will really provide value-added situation awareness is unclear, particularly so in a web services framework. Trust relationship with providers and users would be a determining factor for the long term of these systems in the network or on the edge. Clearly, much remains to be seen how these systems will evolve in the semantic web environment.

7. SUMMARY

We have provided a profiles view of the web architecture in which situation-aware applications can be conceived and implemented. We believe that such applications are in the interest of both, the user as well as the service provider, and will be a prominent part of the semantic web. The technology underpinnings of such architecture with regards to profile frameworks, transport and intermediaries have been analyzed in detail. We conclude that the development of suitable vocabularies that effectively correlate user profiles to application profiles upon which rules can be applied and inferences made, and the implementation of adequate trust mechanisms are critical to the future of the situated web.

8. ACKNOWLEDGMENTS

The authors wish to acknowledge the contribution of Javier Arellano and thank SBC Technology Resources and Ericsson Research for resources provided to participate in W3C standards activities.

9. REFERENCES

[1] Theodorakis, Contextualization: An Abstraction Mechanism for Information Modeling, Doctoral dissertation, University of Crete, 06/2001.

[2] Schilit, A system architecture for Context-aware computing, Doctoral thesis, Columbia University, 06/1995.

[3] MacIntyre, Context-aware Personal Augmented Reality, Workshop on research directions in situated computing, ACM conference on Human factors in computing, CHI 2000, http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pospapers.pdf

[4] Morse, The what, who, where, when and why of context awareness, CHI 2000 Workshop, 04/2000, http://mcs.open.ac.uk/drm48/chi2000.htm

[5] Hacioglu, K., Ward, W. Dialog-Context Dependent Language Modeling Using N-Grams and Stochastic Context-Free Grammars", to appear in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2001), Salt Lake City, May 2001, http://communicator.colorado.edu/papers/kadri-icassp-2001.pdf

[6] Bray, et al., Namespaces in XML, W3C Recommendation, 01/1999 available at http://www.w3.org/TR/1999/REC-xml-names-19990114/

[7] Lassila, O., Swick, R. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 02/1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[8] Hjelm, J. Building the Semantic Web with RDF, John Wiley and Son, 2001.

[9] Clark, DeRose, XML Path Language 1.0, W3C Recommendation, 11/1999, http://www.w3.org/TR/xpath

[10] DeRose et al. XML Pointer Language Version 1.0, W3C Recommendation, 09/2001, http://www.w3.org/TR/xptr/

[11] Sadeh, A semantic web environment for context aware mobile services, Workshop on research directions in situated computing, ACM conference on Human factors in computing, CHI 2000, http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pospapers.pdf

[12] The 3GPP Generic User Profile, Work Item Description TSGS#13(01) 0548, http://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_13/Docs/PDF/SP-010548.pdf

[13] Singhal, et. al. WAP: Writing Applications for Mobile Devices, Addison Wesley, 10/2000.

[14] WAG User Agent Profile Specifications, “WAP 2.0 Version, 10/2001, http://www1.wapforum.org/tech/documents/WAP-248-UAProf-20011020-a.pdf

[15] Wireless Application Protocol User Agent Profile Specification, 05/2001, http://www1.wapforum.org/tech/documents/WAP-248-UAProf-20010530-p.pdf

[16] Klyne, et. al. CC/PP: Structure and Vocabularies, W3C Working Draft, 09/2001, http://www.w3.org/TR/CCPP-struct-vocab/

[17] Suryanarayana, L., Hjelm, J., CC/PP for Content Negotiation and Contextualization, Lecture Notes in Computer Science, Springer Verlag, 02/2001

[18] Hjelm, et al. If The Sun Is Shining, Show Me The Way To The Beach: Profile-Based Interactive Information Service Models, YRP Telecom Summit Conf. 2001 (IEEE Comsoc Japan Chapter, Yokosuka Research Park)

[19] Hjelm, J., Nilsson, M. Position-Dependent Services Using Metadata Profile Matching, INET 2000.

[20] Cranor, et al. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Working Draft 09/2001, www.w3.org/TR/P3P

[21] Klyne, G., Massinter, L. Identifying Composite Media Features, IETF RFC 2938, http://www.ietf.org/rfc/rfc2938.txt?number=2938

[22] Klyne, G. A syntax for describing media feature sets, IETF RFC 2533, 03/1999 http://www.ietf.org/rfc/rfc2533.txt?number=2533

[23] Bohrer, Customer Profile exchange specification, Version 1.0, 10/2000, http://www.cpexchange.org

[24] Dubinko, et al. XForms 1.0, W3C Working draft, 08/2001, http://www.w3.org/TR/xforms/

[25] Miller, Brickley, RDF calendar notes, 02/2001 http://www.ilrt.bris.ac.uk/discovery/2001/02/calendar/

[26] Fitzpatrick, et al., SkiCal – An extension of iCalendar, IETF Internet Draft, 07/2001, http://www.ietf.org/internet-drafts/draft-many-ical-ski-04.txt

[27] Massinter, et al. Media Features for Display, Print and Fax, RFC 2534, IETF, http://www.ietf.org/rfc/rfc2534.txt?number=2534

[28] Orman, Data integrity for mildly active content, 08/2001, http://www.ietf-opes.org/ietf51%20files/OPES%2051IETF%20files/opes-data-integrityPaper.pdf

[29] Dublin Core Metadata Initiative: http://dublincore.org/

[30] Hori, et al., Annotation of Web Content for Transcoding, W3C Note, 07/99, http://www.w3.org/TR/annot

[31] Raggett, D., et al. XHTML Document Profile Requirements, W3C Working Draft, 09/1999, http://www.w3.org/TR/xhtml-prof-req

[32] R. Fielding et. al. Hypertext Transfer Protocol -- HTTP/1.1, IETF RFC 2068, January 1997. http://www.w3.org/Protocols/rfc2068/rfc2068

[33] Nielsen, et al. HTTP Extension Framework, IETF Internet Draft, 03/1999, http://www.w3.org/Protocols/HTTP/ietf-http-ext/draft-frystyk-http-extensions-03.txt

[34] Ohto, H., Hjelm, J. CC/PP Exchange Protocol, W3C Note, 06/1999, http://www.w3.org/TR/NOTE-CCPPexchange.

[35] Gugdin, et al. SOAP version 1.2 Part I, W3C Working Draft, 10/2001, http://www.w3.org/TR/soap12-part1/

[36] Brown, et al. SOAP Security Extensions: Digital Signatures, W3C Note, 06/2001 available at http://www.w3.org/TR/SOAP-dsig/

[37] Johan Hjelm, Creating Location Services for the Wireless Web, Professional Developer's Guide, John Wiley and Son, February 2002.

[38] Adaptive Media transcoding proxy, http://www.adaptivemedia.net/htmMain.htm

[39] Avant Go, http://avantgo.com/frontdoor/index.html

[40] IBM Web Application Server – Websphere product suite, information available at http://www-4.ibm.com/software/webservers/appserv/advanced_v35.html

[41] Microsoft's .NET framework, http://www.microsoft.com

[42] Java Technology, http://java.sun.com/

[43] Rule Markup Language home page, http://www.dfki.uni-kl.de/ruleml/

[44] Clark, XSLT Transformations, Version 1.0, W3C Recommendation, 11/1999, http://www.w3.org/TR/xslt

[45] Hjelm, J., Stark, P. XSLT: The ultimate guide to transforming web data, John Wiley and Sons, New York 2001.

[46] Carlson, Modeling XML vocabularies with UML, Part I and II, http://www.xml.com/pub/a/2001/09/19/uml.html

[47] Chang, W. A Discussion of the relationship between RDF schema and UML, W3C Note, 08/98, http://www.w3.org/TR/NOTE-rdf-uml/

[48] Langheinrich, et al. A P3P Preference Exchange Language (APPEL) 1.0, W3C Working Draft, 02/2001, http://www.w3.org/TR/P3P-preferences

[49] Hunter, Lagoze, Combining RDF and XML schemas to enhance interoperability between metadata application profiles, WWW10*,* 05/ 2001, Hong Kong.

[50] Eastlake, et al. XML – Signature Syntax and Processing, W3C Proposed Recommendation, 08/2001 http://www.w3.org/Signature/

[51] Reagle, XML Encryption Requirements, W3C Working Draft, 10/2001, http://www.w3.org/Encryption/2001/

[52] Barbir, et al. A Framework for Personalization of Content and Services, IETF Internet Draft, 06/2001 www.ietf.org/draft-barbir-opes-pcs-draft1-01.txt