Modeling of Context Information
for Pervasive Computing Applications

Albert Held
DaimlerChrysler AG, Research & Technology
D-89081 Ulm, Germany

and

Sven Buchholz and Alexander Schill
Department of Computer Science, Dresden University of Technology
D-01062 Dresden, Germany

ABSTRACT

In a dynamic heterogeneous environment, such as Pervasive Computing, context-aware adaptation is a key concept to meet the varying requirements of different clients. In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. Therefore, a common representation format for the context information is required. In this paper we examine possible representation formats for context information. We discuss requirements for such a representation format and how those requirements are met by existing approaches.

Based on our findings about the flaws of existing approaches we propose a novel representation format that is comprehensive and thoroughly structured to meet all the requirements: *Comprehensive Structured Context Profiles* (*CSCP*). CSCP is based on the Resource Description Framework (RDF) and overcomes the shortcomings of the Composite Capability/Preference Profiles language (CC/PP) regarding structuring. Furthermore it extends the mechanisms to express user preferences.

Keywords: WWW, Pervasive Computing, Context Profiles, hierarchical caching, content adaptation, adaptation paths.

1. INTRODUCTION

In the upcoming world of Pervasive Computing users are accessing the World Wide Web by a huge variety of mobile devices featuring heterogeneous capabilities (regarding display, data input, computing capacity, etc.). Those devices are attached to the Internet by various communication systems offering different functionality and quality of service (bandwidth, delay, etc.).

The key to meet the demands in this heterogeneous environment is the adaptation of the contents in order to provide the optimal presentation according to the capabilities of the devices and the characteristics of the network connectivity. However, the question of the optimal presentation cannot be answered objectively. It rather depends on the preferences of the particular user. Accordingly, content adaptation must take into account context information about the device, the network connection, and the user.

In order to enable context-aware adaptation, context information must be gathered and eventually presented to the application performing the adaptation. Therefore, a common representation format for the context information is required. In this paper we examine possible representation formats for context information. We discuss requirements for such a representation format and how those requirements are met by existing approaches. Thereupon, we introduce a comprehensive, structured representation for all flavors of context information that meets the identified requirements.

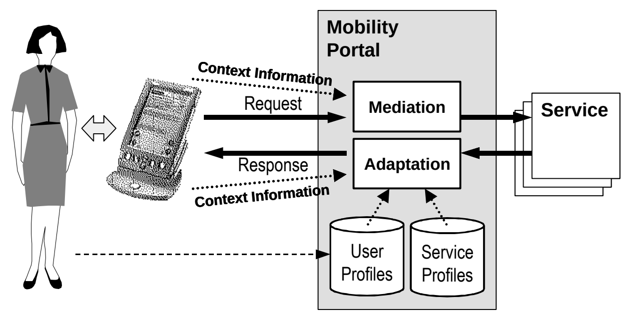


Figure 1: Sample system scenario

Sample System Scenario

In the following we will introduce the reference system scenario we have taken as the basis in our project. We have conceived a *Mobility Portal* to perform context-aware content adaptation (cf. fig. 1). The *Mobility Portal* is the central access point to the portfolio of information services for mobile users. By evaluating context information and managing the user profiles, it allows for personalized, context-aware service mediation and content adaptation.

The concepts for the representation and handling of context information introduced in the following are not restricted to this *Mobility Portal* system scenario. Rather, they are applicable in general for context-aware applications.

Relevant Context Information

As mentioned before, relevant context information includes the capabilities of the mobile devices, the characteristics of the network connectivity, and user specific information.

With respect to the device, relevant context information includes basic hardware features such as CPU power and memory as well as the user interface. The user interface is characterized by the available input (e.g. keyboard, character recognition, voice, or pointing device) and output devices (display, audio) and the particular specifications of those devices, such as display size/resolution, color capability, stereo capability, etc. With respect to the software platform of the device we consider a detailed specification of the supported media content types to be sufficient. Certainly, a simple enumeration of MIME ([1]) content type names does not suffice. Rather, we suggest to supplement the MIME content type names by content type specific characteristics, such as the supported version (e.g. HTML 4.0) or restrictions that apply (e.g. no support for HTML frames).

The network connection is characterized by the bandwidth, delay, and bit error rate (BER). The network connection must be continuously monitored in order to have fresh values for particular characteristics. However, as fresh measurements will not always be available (e.g. immediately after connection setup before starting the communication), we allow for specifying expected values as well as upper and lower bounds. Therewith, the context-aware application may estimate the available network connection. Besides, we provide for indicating the uplink and the downlink characteristics separately to allow for asymmetric connections.

User specific information primarily consists of user preferences. They comprise preferences with respect to the service selection (e.g. cost preferences) and the appearance of the service (e.g. font size, omitting of particular multimedia contents, and preference of frames). User preferences may vary depending on the device capabilities and other context conditions. Therefore, user profiles should provide for means to express conditions applying to the preference attributes. Not all user preferences will always be satisfiable. Therefore, the user should be empowered to specify preference priorities. A maximum priority means that the service shall not be provided if the preference cannot be satisfied. Differentiated priorities furthermore allow resolving conflicting preferences and service capabilities.

Besides user preferences, additional user specific information is relevant for the *Mobility Portal*, e.g. user master data, authentication information, subscriber information, etc. Correspondingly, this information may be included into the user profile.

Generally, user specific information may apply to all applications or to a specific application only. Accordingly, we distinguish generic and application specific user information.

2. REQUIREMENTS TO A COMPREHENSIVE CONTEXT REPRESENTATION

Context information is gathered, stored, and interpreted at different parts of the system. However, the service mediation and content adaptation mechanisms require all relevant information to be available at the *Mobility Portal*, interpretable by it, and unambiguously assigned to the client’s current application-level session (e.g. WML or HTML session). A representation of the context information should be applicable throughout the whole process of gathering, transferring, storing, and interpreting of context information. Therefore, there are a couple of requirements concerning the representation format. A context profile representation should be:

- **structured:** Context profiles may represent a huge number of different context information. A structured representation provides for means to filter relevant information effectively and for natural structuring. Furthermore, it eases unambiguous attribute naming as attribute names can be interpreted context-sensitively.
- **interchangeable:** Context-profiles must be interchangeable among the different components of the system (mobile device, portal). This requires a serializable representation. Besides transfer of the whole profile, a mechanism to transfer a subtree of the profile or a single attribute is required. By this means, a profile does not need to be completely retransferred after the change of a single attribute (e.g. a change in the network delay).
- **composable/decomposable:** By allowing for profile decomposition and composition profiles can be stored and maintained in a distributed way. For instance, a default device profile may be stored at the device vendor’s web site whereas the deviation of a particular device from the defaults is stored at the device itself. By this means, only the deviation from the defaults must be transferred via the wireless link.
- **uniform:** A uniform representation of all flavors of context profiles (device and network profiles, user profiles, possibly additional context data) eases the interpretation during the process of service mediation and content adaptation in the portal.
- **extensible:** No set of attributes that can be identified today will be sufficient for all future applications. Therefore, a profile representation format should provide for future extensions.

- **standardized:** Context profiles are to be exchanged among different entities of the system, e.g.: the mobile device, the Mobility Portal, or the device vendor's web site. Those entities do not typically belong to the same administrative domain. Therefore, there is a strong need for a standardized representation of the context information.

3. PREVIOUS WORK

The approach to utilize profile information for making applications more context-aware is not new. Therefore, several approaches for hardware and user profiles have been proposed before. In this section, existing approaches for hardware and user profile representation will be introduced and their limitations will be discussed.

Composite Capability/Preference Profiles (CC/PP)

Composite Capability/Preference Profiles (CC/PP) ([2]) is the W3C's proposal for a profile representation language. CC/PP is a framework based on the Resource Description Framework (RDF; [3]), an XML based meta data description framework. CC/PP is intended to express both device capabilities and user preferences.

The CC/PP specification defines a basic structure for profiles. A profile is basically constructed as a strict two-level-hierarchy: each profile having a number of components, and each component having a number of attributes. The particular components and attributes are not defined by the CC/PP specification. The definition of a specific vocabulary is up to other standardization bodies. Vocabularies should be defined using RDF Schema ([4]). This flexible mechanism ensures CC/PP extendibility.

A sample vocabulary for describing WAP device profiles is defined by WAG UAProf ([5]) by the WAP Forum. This vocabulary, however, is pretty much tailored to the needs of WAP devices. It does not scale with more complex devices, such as notebook computers. Moreover, there are no means to express the characteristics of the current network connection. Besides, the structure of WAG UAProf is rather tangled. And also, device capabilities and user preferences are interlaced complicating the maintenance of the profiles.

Nevertheless, even the CC/PP framework itself has some disadvantages, such as the strict two-level-hierarchy. The device structure usually is not strictly two-level. That is why the natural structure has to be mapped into two levels. Furthermore, CC/PP requires attribute names to be unambiguous even if they are used in different components. On the other hand, the XML interchange format and a referencing mechanism for external resources excellently meet the interchangeability and composability/decomposability requirement, respectively. A summarization of how CC/PP meets the requirements is illustrated in table 1. Notably, most of the CC/PP features that meet the requirements are inherited from RDF. Accordingly, we propose to build a profile representation based on RDF.

Table 1: CC/PP evaluation

Requirement	*Rating*	*Comment*
structured	–	strict two-level hierarchy, unambiguous attribute naming
interchangeable	✔	by XML serialization
decomposable	✔	by resource references
uniform	✔	
extensible	(✔)	(structural restrictions apply)
standardized	✔	W3C standard

Table 2: IETF Media Feature Sets evaluation

Requirement	*Rating*	*Comment*
structured	–	attribute/value pairs only
interchangeable	✔	
decomposable	–	no means to reference external Media Feature Sets
uniform	✔	
extensible	(✔)	(no formal, machine readable extension mechanism)
standardized	✔	Internet standard

IETF Media Feature Sets

The IETF Content Negotiation (CONNEG) Working Group's Media Feature Sets ([6]) have been developed to allow for protocol-independent content negotiation. It specifies device capabilities and user preferences by unstructured attribute/value pairs. As opposed to CC/PP, it rather specifies the features of the supported and preferred content representation than simply the device capabilities.

Complex capabilities and preferences are expressed by Boolean expressions of attribute/value pairs. An interesting feature of Media Feature Sets is the possibility to assign so-called quality values to capabilities and preferences descriptions. By this means, preferences priorities (as proposed in section 1) may be expressed.

Table 2 summarizes how IETF Media Feature Sets meet the requirements identified in section 2.

Non-standard approaches

Besides the standardized approaches of the W3C and IETF, there are even product specific approaches for capabilities and preferences profiles. Many products (e.g. *IBM WebSphere*, [7]) use simple user preferences profiles in terms of unstructured attribute/value pairs to allow for some degree of personalization.

A more elaborated approach is Microsofts *.NET My Services* (AKA Hailstorm, [8]). *.NET My Services* offers a structured repository of user specific information that can be queried remotely using SOAP. However, the features of *.NET My Services* have not been published in detail by the time of writing.

4. COMPREHENSIVE STRUCTURED CONTEXT PROFILES

In the previous section, we have illustrated that existing approaches for the representation of context information do not meet the requirements identified in section 2. Existing solutions particularly lack sufficient structuring for complex context profiles. Hence, we propose a representation format that is thoroughly structured and comprehensive to allow for all flavors of context information: *Comprehensive Structured Context Profiles* (*CSCP*).

Usage Principles

CSCP expresses context information by means of session profiles (cf. fig. 3). A session profile is attached to the session context of a client's session at the Mobility Portal. It describes all relevant context information of the session. This includes: the device profile, the network profile, the user profile, and possibly other context information, such as environmental information (e.g. location, sensor information, etc.).

The session profile is initially assembled at the client and transferred to the portal during session establishment (cf. fig. 2). However, this does not mean that all context

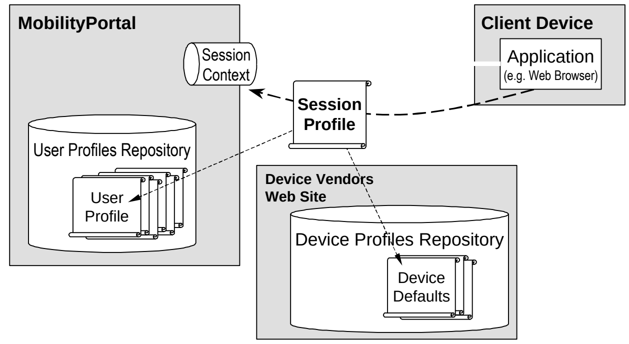


Figure 2: Usage scenario

information must be gathered at the client and transferred via the wireless link. The session profile may rather contain references to external resources, such as the client profile stored at the portal or the device defaults that can be retrieved from the device vendor's web site.

After session establishment, the client does not need to re-send a session profile during the lifetime of the session but simply updates the existing one by a differential profile.

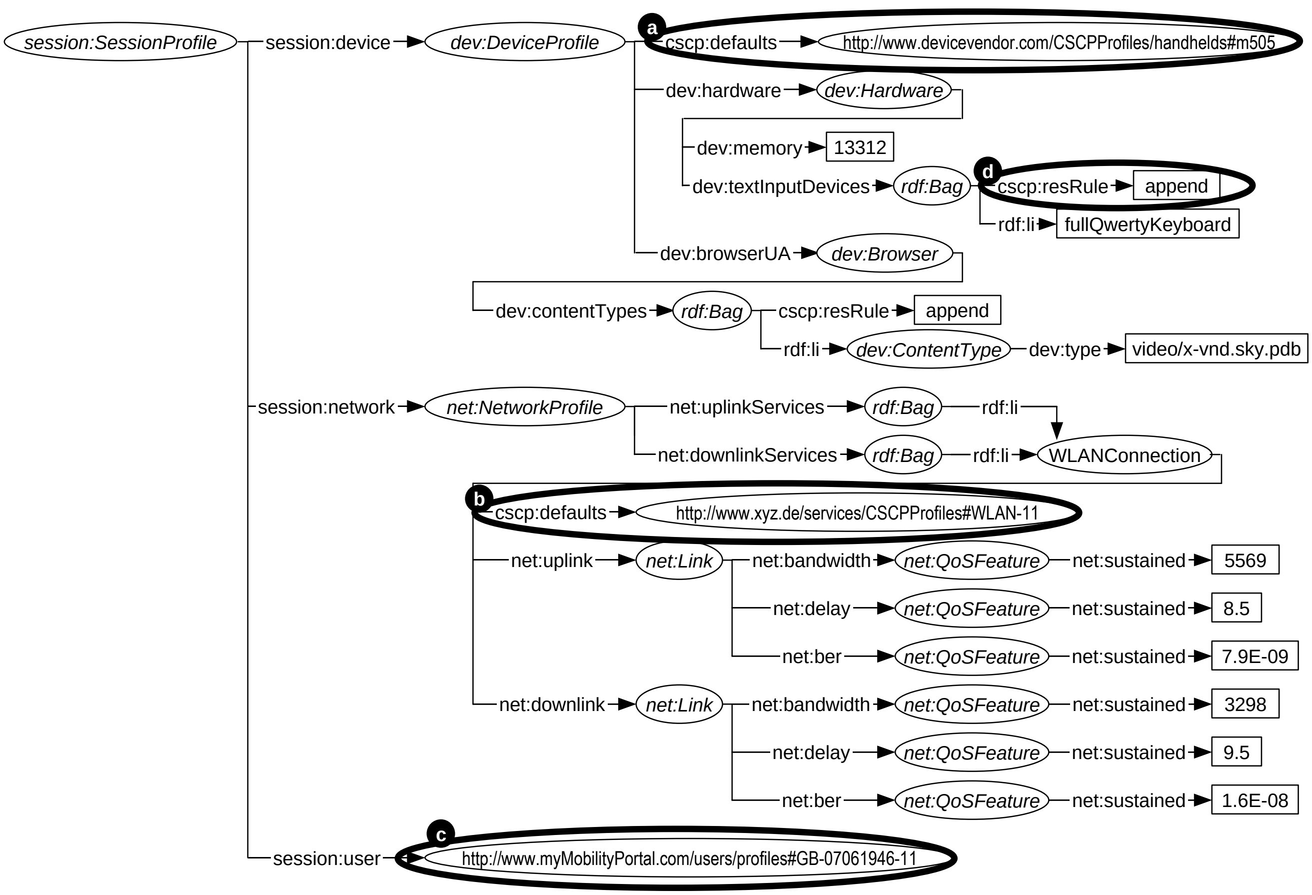


Figure 3: Sample session profile (RDF graph notation; cf. [3])

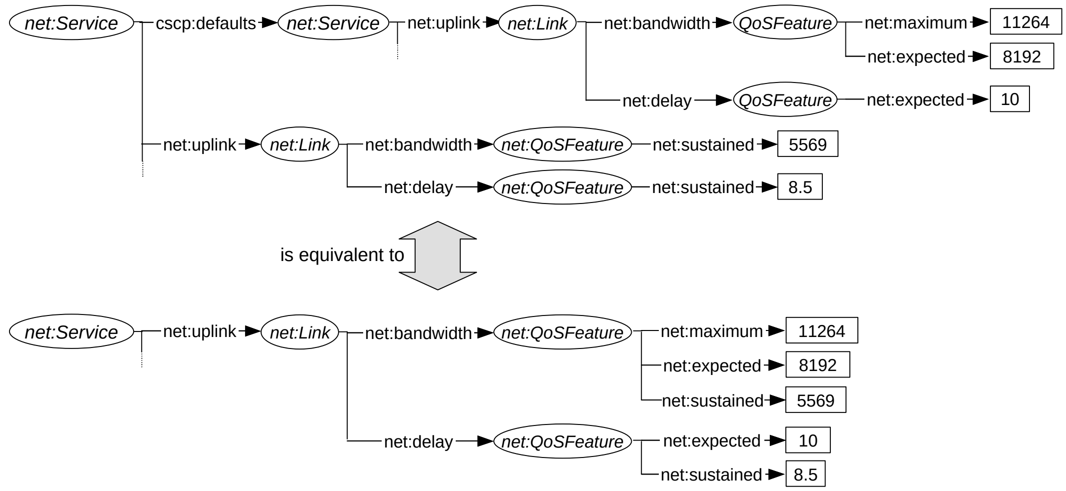


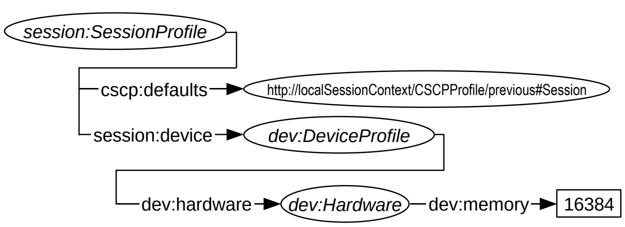
Figure 4: Merging of a profile subtree with its corresponding default subtree (RDF graph notation; cf. [3])

The CSCP Language

Just as CC/PP, CSCP is an RDF ([3]) based meta language. As a descendant of RDF, CSCP inherits the interchangeability, decomposability, and extensibility of RDF. CSCP interchangeability is based on the XML serialization syntax of RDF (cf. fig. 5b).

CSCP overcomes the shortcomings of CC/PP regarding structuring and extends the mechanisms to express user preferences.

Structuring: Unlike CC/PP, the CSCP language does not define any fixed hierarchy. It rather supports the full flexibility of RDF to express natural structures of profile information. Attribute names are interpreted context-sensitively according to their position in the profile structure. Hence, unambiguous attribute naming across the whole profile (as necessary with CC/PP) is not required.



a) RDF graph notation (cf. [3])

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:cscp  = "http://example.org/CSCPProfileSyntax#"
   xmlns = "http://example.org/SessionProfileSyntax#"
   xmlns:dev = "http://example.org/DeviceProfileSyntax#"
   xmlns:net = "http://example.org/NetworkProfileSyntax#">
   <SessionProfile rdf:ID="Session">
      <cscp:defaults rdf:resource=
         "http://localSessionContext/CSCPProfile/previous#Session"/>
      <device><dev:DeviceProfile>
         <dev:hardware><dev:Hardware>
            <dev:memory>9216</dev:memory>
         </dev:Hardware></dev:hardware>
      </dev:DeviceProfile></device>
   </SessionProfile>
</rdf:RDF>
```

b) XML serialization (cf. [3])

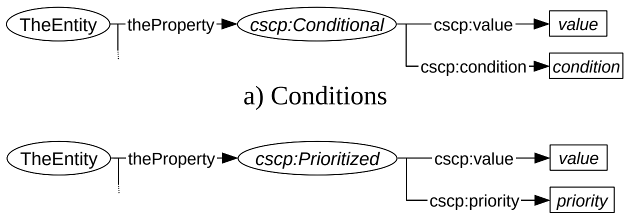
Figure 5: Profile updates

Decomposability: CSCP supports decomposability by means of external references and defaults. External references are used to extract subprofiles to separate CSCP documents. For instance, the user profile may be extracted from the session profile and stored in a separate CSCP document at the Mobility Portal (cf. fig. 3 (c)). For the purpose of referencing external subprofiles CSCP uses the URL-based RDF resource references.

The defaults mechanism is used to extract default properties from a CSCP description. For instance, device defaults or network service defaults can be extracted from the session profile and stored at the device vendor's or network service provider's web site (cf. fig. 3 (a)+(b)). For this purpose CSCP comes with a defaults mechanism similar to the one of CC/PP. Unlike CC/PP, which allows for overriding of default attribute values only, the CSCP defaults mechanism allows for merging of profile subtrees with their corresponding default subtrees (cf. fig. 4) in addition to the overriding semantics[1]. By this means, the defaults mechanism supports the structuring requirement.

Besides, the defaults mechanism is utilized to propagate profile updates. Profile changes are expressed in a differential profile. The differential profile refers to the previous profile as its default settings and simply overrides attribute values that have changed (cf. fig. 5). By this means, a client does not need to re-send a session profile during the lifetime of a session.

[1] Whether to use the merging or overriding semantics is indicated by the *cscp:resRule* attribute. For RDF containers (such as *rdf:Bag*) CSCP defines further semantics for dealing with default values, such as appending, intersection, or difference (cf. fig. 3 (d)).



a) Conditions

b) Priorities

Figure 6: CSCP conditions and priorities
(RDF graph notation; cf. [3])

Extended mechanisms for expressing user preferences: Furthermore, CSCP provides features to attach conditions and priorities to attributes (cf. fig. 6). This extends the means to express user preferences. By assigning conditions to user preference attributes, varying user preferences depending on the device capabilities or other context conditions may be specified. By means of multiple conditional RDF statements about an attribute, if-then-elsif-else expressions may be formulated. Differentiated priorities of attributes allow resolving conflicts between preferences and service capabilities.

CSCP Vocabularies

Besides the CSCP language, we have defined a CSCP vocabulary to express session profiles comprising a device profile, a network profile, and a user profile. These are distinguished using XML namespaces (cf. fig. 3, *session:*, *dev:*, *net:*). These vocabularies meet the requirements in our sample application scenario. However, the CSCP vocabulary is easily extensible for future applications by RDF Schema [4].

5. CONCLUSIONS

In this paper we examined representation formats for context information that are necessary to allow context-aware adaptation. We discussed relevant context information and identified the requirements to a comprehensive context representation. Nevertheless, existing approaches for the representation of context information fail to meet all of those requirements.

Based on our findings about the flaws of existing approaches we proposed a novel format: Comprehensive Structured Context Profiles (CSCP). CSCP is based on RDF and overcomes the shortcomings of the CC/PP language regarding structuring. Furthermore it extends the mechanisms to express user preferences.

CSCP was designed to fit a system scenario with a Mobility Portal performing context-aware adaptation. However, it is flexible and open to be applicable in general for context-aware applications.

6. REFERENCES

[1] N. Freed, N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, *RFC 2045*, Nov 1996. (URL: http://www.faqs.org/rfcs/rfc2045.html)

[2] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies”, *W3C Working Draft*, Mar 15, 2001. (URL: http://www.w3.org/TR/2001/WD-CCPP-struct-vocab-20010315/)

[3] O. Lassila, R. Swick, “Resource Description Framework (RDF) Model and Syntax Specification”, *W3C Recommendation*, Feb 22, 1999. (URL: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/)

[4] D. Brickley, R. Guha, “Resource Description Framework (RDF) Schema Specification 1.0”, *W3C Candidate Recommendation*, Mar 27, 2000. (URL: http://www.w3.org/TR/rdf-schema/)

[5] “WAG UAProf, Proposed Version 30-May-2001”, Wireless Application Protocol Forum Ltd. (URL: http://www1.wapforum.org/tech/documents/SPEC-UAProf-19991110.pdf)

[6] G. Klyne, “A Syntax for Describing Media Feature Sets”, *RFC 2533*, Mar 1999. (URL: http://www.faqs.org/rfcs/rfc2533.html)

[7] “4.4.2: Keeping user profiles”, in “IBM WebSphere Application Server Version 3.5 Advanced Edition”, *IBM WebSphere InfoCenter*, IBM Corporation. (URL: http://www-4.ibm.com/software/webservers/appserv/doc/v35/ ae/infocenter/was/040402.html)

[8] “Building User-Centric Experiences: An Introduction to .NET My Services”, *White Paper*, Microsoft Corp. (URL: http://www.microsoft.com/net/hailstorm.asp)