Context for Simplicity: A Basis for Context-aware Systems Based on the 3GPP Generic User Profile

Enrico Rukzio, George N. Prezerakos, Giovanni Cortese, Eleftherios Koutsoloukas, Sofia Kapellaki

***Abstract*—**The paper focuses on the area of context modeling with respect to the specification of context-aware systems supporting ubiquitous applications. The proposed approach, followed within the SIMPLICITY IST project, uses a high-level system ontology to derive context models for system components which consequently are mapped to the system's physical entities. For the definition of user and device-related context models in particular, the paper suggests a standard-based process consisting of an analysis phase using the Common Information Model (CIM) methodology followed by an implementation phase that defines 3GPP based components. The benefits of this approach are further depicted by preliminary examples of XML grammars defining profiles and components, component instances, coupled with descriptions of respective ubiquitous applications

***Keywords*—** 3GPP, Context, Context-awareness, Context Model, Information Model, User Model, XML.

I. Introduction

THE evolution towards systems beyond 3G led to a dramatic increase in the demand for user centric, personalized, context-aware, ubiquitous applications via a variety of terminal devices regardless of the underlying infrastructure. These requirements comprised the stimulus and basis for the SIMLICITY IST project [1], which is firmly rooted on the principles of context-aware applications and ubiquitous computing.

The project introduces the Simplicity Device (SD) that targets the nomadic user. It is a lightweight and wearable personalization device, which is compliant with many types of terminals. The SD, when plugged to a terminal, becomes the "virtual" identity of the end-user, providing personalization information which is utilized by system components that process user behavior related data (the Simplicity user context). Context information maintained by the SD, in combination with the terminal device, can be exploited by applications to implement adaptive, context-aware services.

Obviously, Simplicity requires a coherent Simplicity Context Model to be defined. The process of the Simplicity Context Modeling is performed in several stages. Initially, key system components and the relations between them are identified; this process yields the Simplicity Information Model, which is, in essence, a high-level ontology of our system. Finally, the features mandated by each context model have to be mapped to physical network entities residing within the Simplicity system.

A pragmatic approach is followed that aims at reusing existing methodologies and recommendations from international standards organizations. The Simplicity ontology was developed using the Common Information Model (CIM) methodology of the Distributed Management Task Force (DTMF) [2]. Simplicity context modeling followed the methodology proposed by the 3GPP Generic User Profile [3] for the development of user profiles, and was based on the W3C Composite Capabilities/Preferences Profile (CC/PP) [9] framework for modeling device related profiles. XML Schemas were produced by applying the 3GPP Data Description Method (DDM) [4].

II. Context Modeling – Related work

Several proposals for context information modeling exist, all dealing, not surprisingly, with people, locations, devices, services and their static and dynamic relationships. We summarize below key characteristics from these models (see also [5] for a longer survey).

People: Information about people includes static and dynamic information describing people currently in the environment. Personal data typically includes identity, characteristics, user abilities, preferences, and owned resources. Dynamic context for mobile users is acquired through sensors located in the user device or in the environment, and includes position and other data [6]. Other application-specific data may also be relevant for context-aware services, e.g. user skills and interests [7].

Places and Locations are typically arranged in containment hierarchies. Locations carry context information such as data

This work was supported in part by the EU project "Simplicity" IST-2004-507558.

E. Rukzio is with the Ludwig-Maximilians-University (LMU) of Munich, Geschwister-Scholl-Platz 1, 80539 Munich Germany (e-mail: Enrico.Rukzio@ifi.lmu.de).

G. N. Prezerakos, Eleftherios Koutsoloukas and Sofia Kapellaki are with the National Technical University of Athens, School of Electrical & Computer Engineering, 9 Heroon Polytechniou Str., 157 73 Athens, Greece (e-mail: {prezerak, lefterisk, sofiak}@telecom.ntua.gr). G. N. Prezerakos is also with the Technological Education Institute (TEI) of Piraeus, Dpt. of Electronic Computer Systems, 250 Thivon Av. & Petrou Ralli, 122 44 Athens, Greece.

G. Cortese is with the Telecom Italia Learning Services, V.le Parco de' Medici 37 00148 Rome Italy (e-mail: g.cortese@computer.org).

on temperature, humidity, brightness, and sound levels, acquired by *sensors* in the environment. Given that entities such as people, devices, and objects live and move in physical space, locations can also be seen as 'containers' for other entities. Geographical location info is conveyed either by geometric or by symbolic models. Existing systems often use mixed models [8].

Devices may be mobile devices carried by users, or devices installed in the environment. Related to terminals and devices, the Composite Capabilities/Preferences Profile (CC/PP) framework aims to standardize the specification of user agent profiles [9].

Services: Technically, a service can be anything ranging from a web service, a web application, or a functionality of a networked appliance accessed through a peer-to-peer protocol. A lot of literature, both from the agent (DAML-S) and Web Services community exists on service modeling, where the emphasis is on providing descriptions which help in matching services to user requests.

III. Reference Architecture

The Simplicity Architecture is comprised of a brokerage framework consisting of a number of Simplicity Brokers (SB), that host the different core Simplicity Services (SS); brokers orchestrate the user's interaction with 3rd party services and applications.

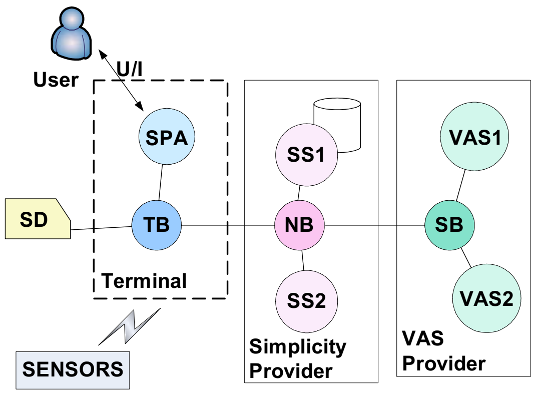

Figure 1: Simplicity architecture

Figure 1 illustrates the core entities in the Simplicity architecture.

The Terminal Broker (TB) is present at each Simplicity enabled user terminal. The TB interacts with its peers residing at the network side, the Network Brokers (NB), which are managed by the Simplicity Operator or Value Added Service (VAS) Providers. Besides the TB, a Simplicity enabled terminal also hosts a Simplicity Personal Assistant (SPA), an entity which directly interacts with the user through personalized UIs. Finally, the Simplicity Device (SD) is the innovative Simplicity component that represents the user's identity and stores user preferences and profiles.

Each entity contributes to context generation based on its role in the architecture. The SD, as storage of user profiles and credentials, provides this information to the TB for the initial user context generation. The SPA, by directly interacting with the user, contributes information about the user activities, such as the information filled in web forms. The TB is the entity that interacts directly with the terminal equipment, and is therefore responsible for the device profile, which includes the hardware, software and networking features of the terminal. Also, the TB collects context information from sensors in the user environment. Finally, the NBs contribute user information stored in the network side, such as subscribed services, service usage history etc.

The information contributed from the entities above is used for *context generation* by a dedicated entity, the Context Management Subsystem (CMS), which is attached to the Network Broker (NB). Additionally, the Context Management Subsystem is responsible for distributing context information to other entities, and for updating the context, according to user actions and changes in the networking and physical environment.

IV. Simplicity Information Model

During the analysis phase of our project we have developed an information model, which describes the entities of our system, their properties and relationships.

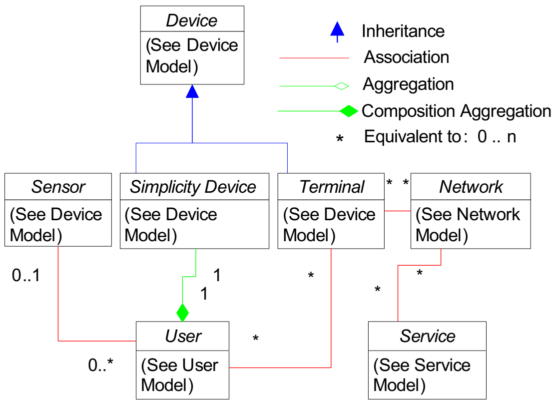

Figure 2: Core model of the information model

We used the methodology, syntax and semantic of the Common Information Model (CIM) standard of the Distributed Management Task Force (DMTF) [2] which is based on UML.

A subset of this information model provides the basis for our context model. It consists of one core model (see Figure 2) which is refined in the four sub-models user, device, service and network.

V. Simplicity Context Model

After designing the information model, we decided to use the architecture, conventions and syntax of the emerging standard 3GPP Generic User Profile [3], currently under development, to create a grammar framework which describes the syntax of our context model. 3GPP is an open framework; it makes no assumptions about the content of the profile, limiting itself to the definition of its structure. Our user-focused context model was respectively named *Generic User Profile*.

A. Data Description Method of 3GPP

The 3GPP Generic User Profile is a collection of user related data, which can be used for diverse adaptations in different applications. The Data Description Method (DDM) [4] of 3GPP does not define an ontology with a specific set of elements, attributes and relationships; it is rather a set of common rules to specify profiles and data components.

Every profile consists of different profile components which might represent e.g., general user information, authorized and subscribed service information, public land / mobile network specific user information, terminal related data or service specific information of the user [10]. Furthermore the DDM describes in detail how profiles and profile components are specified based on XML Schema.

The DDM describes, currently based on different examples, the syntax of the different XML Schemas. They describe for instance how XML namespaces, annotations, and data type definitions have to be specified. Our implementation of the context model cannot conform entirely to the 3GPP Generic User Profile because the standard is still under development and some documents have the state of a draft.

B. XML Schemas for Profiles and Components

Based on our ontology and the DDM, we have implemented XML Schema grammars for the different profiles and profile components. Figure 3 provides a graphical overview of the XML Schema defining the syntax of the generic user profile.

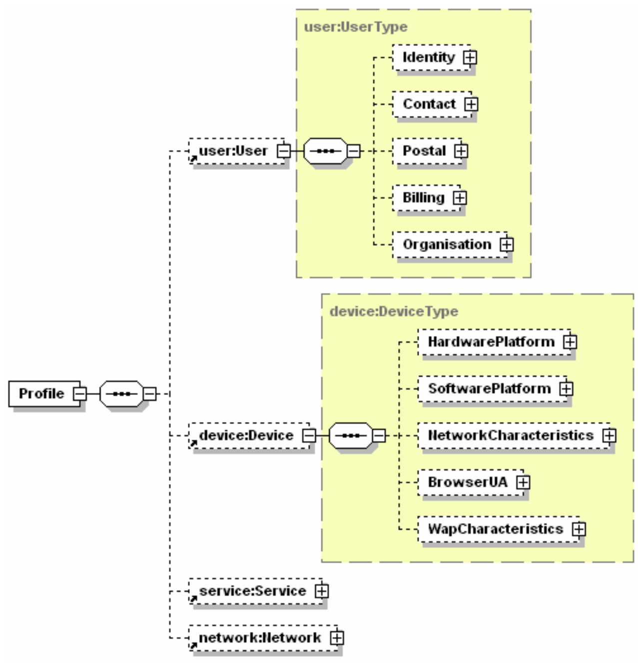

Figure 3: Profiles, Components and Elements

In Figure 4 the different profile components are integrated in the XML Schema for the profile through corresponding namespaces and the import of the XML Schema definitions of different profile components.

The main element *Profile* defines that a profile might consist of a *Device, User, Service* and *Network* component.

```
<schema targetNamespace="http://www.ist-simplicity.org/GUP/" …
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:user="http://www.ist-simplicity.org/GUP/User"
xmlns:device="http://www.ist-simplicity.org/GUP/Device"
xmlns:network="http://www.ist-simplicity.org/GUP/Network"
xmlns:service="http://www.ist-simplicity.org/GUP/Service">
<import namespace="http://www.ist-simplicity.org/GUP/Device"  … />
<import namespace="http://www.ist-simplicity.org/GUP/User"  … />
<import namespace="http://www.ist-simplicity.org/GUP/Network"  … />
<import namespace="http://www.ist-simplicity.org/GUP/Service" …"/>
<element name="Profile">
 <complexType>
  <sequence>
  <element ref="user:User" minOccurs="0"/>
  <element ref="device:Device" minOccurs="0"/>
  <element ref="service:Service" minOccurs="0"/>
  <element ref="network:Network" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
</schema>
```

Figure 4: XML Schema for profile description

Below, we present, by example, the grammars for the components *Device* and *User* to illustrate the simplicity and strength of component definitions with DDM.

C. User Component

The grammar for the user component is based on the W3C Working Draft *Client Side Automated Form Entry* [11] which defines data elements which are grouped into *identity information, contact information, postal information, postal information (billing)* and *organizational information*. We merged this structure with related information from our ontology and developed a corresponding XML Schema describing the syntax of the user component.

```
<schema … targetNamespace="http://www.ist-simplicity.org/GUP/User"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:user="http://www.ist-
simplicity.org/GUP/User">
 <annotation>
  <documentation xml:lang="en">
   Profile Component: User
   Parents: P GUP
   Version: 1.0
   Histroy: 1.0 created on 2004-25-07
  </documentation>
 </annotation>
 <element name="User" type="user:UserType"/>
……
</schema>
```

Figure 5: XML Schema description for user component

Figure 5 focuses on how we used the different namespaces and how profile components are annotated according to the DDM.

D. Device Component

The XML Schema for the device component is based on a UAProf Schema [12] which is provided by the WAP Forum. With this component it is possible to describe the different hardware elements (e.g. mobile devices, presentation devices, terminals, etc.) that are involved. We translated the RDF document defining UAProf into a XML Schema for integration into our XML-based context model.

```
<schema … targetNamespace="http://www.ist-simplicity.org/GUP/Device"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:device="http://www.ist-
simplicity.org/GUP/Device">
 ……
 <element name="Device" type="device:DeviceType"/>
 <complexType name="DeviceType">
  <sequence>
```

```
    <element name="HardwarePlatform" …/>
    <element name="SoftwarePlatform" …/>
    <element name="NetworkCharacteristics" …/>
    <element name="BrowserUA" …/>
    <element name="WapCharacteristics" …/>
  </sequence>
</complexType>
.......
<complexType name="SoftwarePlatformType">
  <annotation>
    <documentation xml:lang="en">The SoftwarePlatform component
    contains properties of the device's application environment, operating
    system, and installed software.</documentation>
  </annotation>
  <all>
    <element name="AcceptDownloadableSoftware"
    type="boolean" minOccurs="0"/>
  </all>   .......
</complexType>
.......
</schema>
```

Figure 6: XML Schema description for device component

As presented in Figure 6, the structure of the device component is very similar to the structure of the user component. According to the UAProf Schema, substructures exist for describing hardware, software, network, browser and WAP characteristics. The documentation of the UAProf Schema was also integrated in our XML Schema for the device component as it can be seen in the definition of the *SoftwarePlatformType*.

With the transformation possibilities of XML, it is easily possible to transform a UAProf document in a device component. Consequently, existing UAProf documents can be used by our context model.

E. *Instance Documents Representing Context*

A concrete profile has to conform to the different grammars which were presented earlier. Some aspects of the DDM might be changed in the future by 3GPP. In Figure 7 for instance the different elements of the components are qualified by a namespace as defined in DDM which does not support the readability of this document.

```
<Profile …>
<user:User>
<user:Identity>
<user:FirstName>Enrico</user:FirstName>
<user:LastName>Rukzio</user:LastName>
</user:Identity>
<user:Contact>
<user:Email>Enrico.Rukio@ifi.lmu.de</user:Email>
<user:Homepage>http://www.mimuc.de</user:Homepage>
<user:TelephoneWork>+49 89 2180-4656</user:TelephoneWork>
</user:Contact>
</user:User>
<device:Device>
<device:HardwarePlatform>
<device:ScreenSize>101x80</device:ScreenSize>
<device:Model>T68R1</device:Model>
<device:ImageCapable>true</device:ImageCapable>
<device:Keyboard>PhoneKeypad</device:Keyboard>
<device:Vendor>Ericsson Mobile </device:Vendor>
</device:HardwarePlatform>
...
</device:Device>
.....
</Profile>
```

Figure 7: Profile Instance

VI. Conclusion

The paper illustrates the feasibility of a standards based approach to the creation of context models for system components supporting ubiquitous applications. Experiences from performing context modeling with CIM are very convincing with respect to the maturity of the standard; especially when it comes to providing standardized syntax and templates for the specification of high-level information models. Moreover, models designed with CIM can be used effectively in conjunction with 3GPP specifications for the creation of profiles describing context information. The use of XML-grammars ensures that the component profile descriptions can be easily transformed to equivalent descriptions mandated by 3GPP standards when such descriptions become available. The profiles examined within the paper are basically static ones (user and device) but it should be noted that this is one of the very first approaches to exploit 3GPP principles for the purposes of context modeling. Based on the preliminary state of the current XML documents for defining profiles and profile components, we are developing first prototypes which will lead to further development of these documents. Future work will also involve context modeling of more dynamic information elements such as user location.

As a proof of concept for the Simplicity vision, the project will implement a demonstrator comprised of a ubiquitous media streaming, a medical monitoring and a tour-guide application. Every application, even though possessing diverse characteristics, will exploit the same context model. The open design and flexibility of the Simplicity context model provides seamless integration of novel context-aware applications with only minor modifications.

References

[1] Simplicity Project Web Site. http://www.ist-simplicity.org
[2] Distributed Management Task Force. Common Information Model. http://www.dmtf.org/standards/cim/
[3] 3rd Generation Partnership Project. Generic User Profile – architecture. *Technical specification of Technical Specification Group Services and System Aspects*, Version 6.3.0. 2004.
[4] 3rd Generation Partnership Project. Data Description Method (DDM) - 3GPP Generic User Profile (GUP). *Technical specification of Technical Specification Group Terminals*, Version 6.1.0. 2004.
[5] Cortese G., Davide F., Lunghi M. Context Awareness for Pervasive Computing Environments. *Ambient Intelligence Book*, IOS Press (to be published)
[6] Schmidt A., Beigl M. and Gellersen Hans-W. There is More to Context than Location. *Computers & Graphics Journal*, Elsevier, Volume 23, No.6, December 1999, pp 893-902.
[7] Chen H., Perich F., Finin T., Joshi A. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. *International Conference on Mobile and Ubiquitous Systems: Networking and Services*. Boston, August 2004.
[8] Domnitcheva S. Location Modeling: State of the Art and Challenges. *Workshop on Location Modeling for Ubiquitous Computing.* Atlanta, Georgia, September 30, 2001.
[9] Composite Capabilities / Preferences Profile, http://www.w3.org/Mobile/CCPP/
[10] 3rd Generation Partnership Project. Service requirements for the 3GPP Generic User Profile (GUP). *Technical specification of Technical Specification Group Services and System Aspects*, Version 6.3.0. 2004.
[11] Hallam-Baker P. Client Side Automated Form Entry. W3C Working Draft, WD-form-filling-960416.
[12] WAP Forum. Wireless Application Group, User Agent Profile Specification, WAP Forum Approved Specification WAP-174, 1999.