Proposal for a User Modeling Markup Language (UserML)

Dominik Heckmann
European Post-Graduate College
"Language Technology and Cognitive Systems", Saarland University
(heckmann@dfki.de, www.dfki.de/~heckmann)

Abstract

A unified markup language for User Modeling should be a key issue in the research community. Standardization is of great importance but single applications also call for very specific solutions. With UserML, we try to contribute some aspects and try to feed into future standardization efforts. One purpose of the UserML project is to develop a platform for the communication about partial user models in an ubiquitous computing environment, where all different kinds of systems work together to satisfy the user's needs. Jon Orwant claims in his Doppelgänger project [1] *that we need a protocol for encoding information about users, that any given user modeling system should be able to benefit from others and that user models should follow you around.*

Keywords

Markup Language, Distributed User Modeling, Ubiquitous Computing

1 Introduction

The main idea of UserML is to enable communication about partial user models via the Internet. Thus one task of UserML is the representation of *partial user models*. Our approach is to start with a user model taxonomy, (see figure 1). We use this taxonomy to define parts of the XML document structure, and to have a starting point to define a *user model ontology* in OIL [3] or DAML+OIL later on. But it is not clear to us at this stage, if we should use the taxonomy entries to define the XML-tags or if we should use them only as reference names.

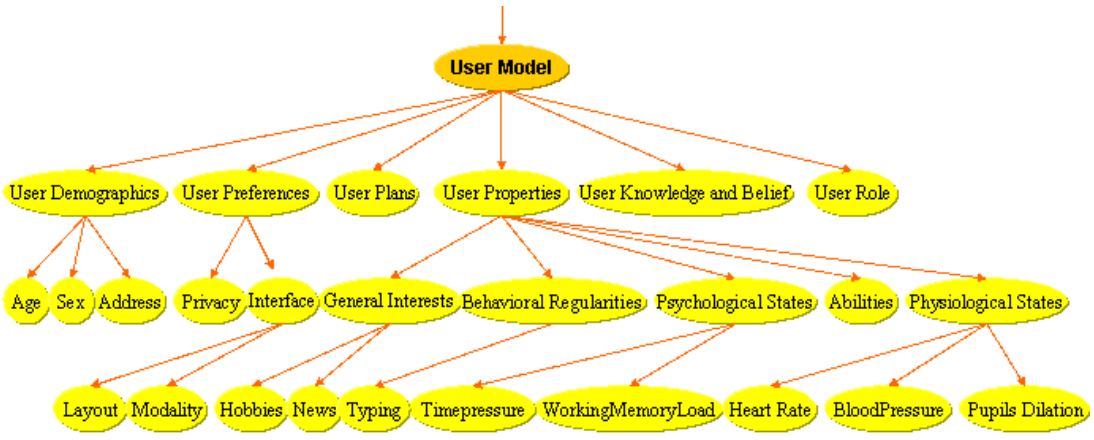

Figure 1: A partial User Model Taxonomy to start with

Using XML as knowledge representation language has the advantage that it can be used directly in the Internet environment. One disadvantage of XML is that the nested structures of the

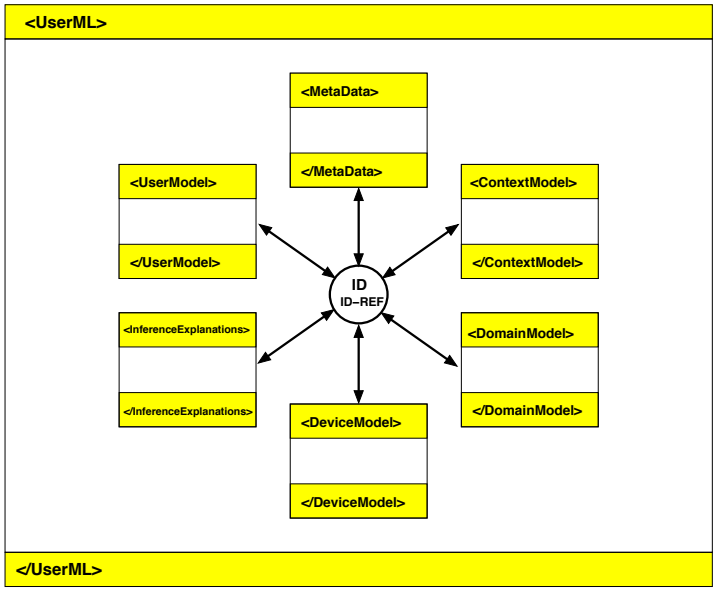

Figure 2: UserML Containers, connected via IDs and ID-REFs

XML-tags only represent a *tree*, while often the structure of a *graph* is needed. For UserML we have chosen to use a modularized approach, (see figure 2), in which several categories will be connected via identifiers (IDs) and references to identifiers (ID-Refs). With this method, the tree structure of XML can be extended to represent graphs. Figure 2 introduces the terms MetaData, UserModel, InferenceExplanations, ContextModel, DomainModel, DeviceModel. These terms are not fixed, they only represent possible containers to store connected information. The main focus of our research will lay on the UserModel and the InferenceExplanations. The idea behind them is the following: in order to explain inferred user model entries, the environment of the user modeling process, like devices, context (see i.e. [2], [5], [6]) or domain knowledge need to be represented or at least referred to.
We analysed several markup languages like EML (Emotional Markup Language), EML (Educational Markup Language), HumanML (Human Markup Language), VHML (Virtual Human Markup Language), UIML (User Interface Markup Language) and CPEX (Customer Profile Exchange Language). They are more or less related to the defined requirements of a user modeling markup language but not fully suitable to be used straightforwardly for our purpose.

2 An example XML document

We start bottom up with an example XML document of an user-adaptive airport navigation system, which is currently under development in the integrating scenario of the *German Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB378, in the projects EM5 (READY) and EM4 (REAL).*

2.1 The situation in the airport example

How can we represent for example such an information? *A system at an airport detects that a person is currently under high time pressure because he has a flight ticket for a flight, which*

boarding time will probably close in 10 minutes and the user still has to navigate to the gate.

A <UserModel> document and its DTD

```
<UserModel>
    <entry id="231">
        <name>Time Pressure</name>
        <category>userproperty.timepressure</category>
        <value>high</value>
        <range-ref>discrete.5.low-high</range-ref>
    </entry>
    <entry id="224">
        <name>Walking Speed</name>
        <category>userproperty.walkingspeed</category>
        <value>fast</value>
        <range-ref>discrete.5.slow-fast</range-ref>
    </entry>
    <entry id="122">
        <name>Current Location</name>
        <category>usercontext.location</category>
        <value>X35</value>
        <range-ref>airport.location</range-ref>
    </entry>
</UserModel>
```

```
<!ELEMENT  UserModel (entry)*>
<!ELEMENT  entry (name, category?, value?, range-ref?)>
<!ATTLIST  entry id ID>
<!ELEMENT  name (#PCDATA)>
<!ELEMENT  category (#PCDATA)>
<!ELEMENT  value (#PCDATA)>
<!ELEMENT  range-ref (#PCDATA)>
```

An <InferenceExplanations> document and its DTD

```
<InferenceExplanations>
    <inference id="002">
        <inferred>
            <userplan>goto.airport-location-X34</userplan>
            <userproperty>timpressure.high</userproperty>
       </inferred>
        <inferred-from>
            <evidence>user.has-flight-ticket</evidence>
            <evidence>user.airport-location-X20</evidence>
            <evidence>boarding-time.flight</evidence>
        </inferred-from>
        <inferred-by>
            <device>334</device>
        </inferred-by>
    </inference>
</InferenceExplanations>
```

```
<!ELEMENT  InferenceExplanations (inference)*>
<!ELEMENT  inference (inferred, inferred-from?, inferred-by?)>
<!ATTLIST  inference id ID>
<!ELEMENT  inferred (category, value, ...)>
<!ELEMENT  inferred-from (evidence)*>
<!ELEMENT  evidence (#PCDATA)>
<!ELEMENT  inferred-by (device)*>
<!ELEMENT  device (#PCDATA)>
```

FIGURE 3: A MOBILE USER MODELING EDITING TOOL

A <MetaData> document

```
<MetaData>
   <meta name="author" content="system234" />
   <meta name="date" content="23/04/2002" />
   <meta name="time" content="14:00" />
</MetaData>
```

The <meta> element describes single pieces of meta data about the current UserML document. This may include date, time or information about the author. The `name` attribute specifies an identifier for the meta-information, the `content` attribute gives its content.

3 Tools for UserML

If a system sets up a model about a user, the person should have the possibility to inspect this model in a human readable format and to edit this data. A private (not necessarily mobile) device seems to be a good choice to serve as an editing tool for user models of different user-adaptive systems, or for general privacy preferences, since it could offer security and privacy as well as user identification.
Currently such an Editing tool is implemented while the user data can be selected by a user model menu (see Figure 3), or by selecting an adaptive system. The current implementation is database driven and needs a web browser.

4 Ubiquitous Computing will influence User Modeling

Ubiquitous computing [8] is a paradigm shift where technology becomes virtually invisible in our lives. Instead of having a mobile machine or a desk-top, the technology will be embedded in our environment. Xerox describes it like this: *imagine a world with hundreds of wireless computing devices of different sizes in the same room. In order to bring this type of computing out into the environment, among the things we need to rethink are user interfaces, displays, operating systems, networks, and wireless communications.* All these ideas will have some impact on user modeling and should be taken into account in UserML. Within ubiquitous computing the traditional architecture of a user modeling system (see figure 4) will decompose into parts. Some entities sense data and others do inferences or store information, while all of them use the ubiquitous network for communication.
Acquisition, processing, and storage of personal data ubiquitously requires an intensified consideration of user demands to security, privacy, and anonymity (see, e.g., [4]). Access control and

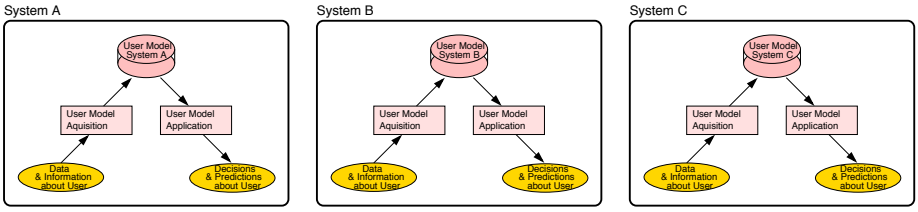

FIGURE 4: STRUCTURE OF THREE INDEPENDENT USER MODELING SYSTEMS

cryptographic techniques enable secure communication, free from interception and modification. We have to take them into account while constructing the UserML ontology.

Conclusion

We try to develop a markup language for user modeling. At the moment, we focus on the following three aspects: representation of partial user models, representation of inference explanations and the environment, representation and realization of privacy issues. This work is under progress.

References

[1] Jon Orwant: Heterogeneous Learning in the Doppelgänger User Modeling System. User Modeling and User-Adapted Interaction **4** (1995) 107-130

[2] Anthony Jameson: "Modeling Both the Context and the User", Personal Technologies, **5** (2001)

[3] Dieter Fensel et al: "OIL in a nutshell" In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference), R. Dieng et al. (eds.) (2000)

[4] Jörg Schreck: "Security and Privacy Issues in User Modeling" User Modeling Conference, Doctoral Consortium (1997) 453-454

[5] Fritz Hohl et al.: "A Context System for a Mobile Service Platform" In H.Schmeck, T. Unger, and L. Wolf (Eds): ARCS 2002, LNCS 2299, Springer Berlin Heidelberg (2002) 21-33

[6] Florian Michahelles et al.: "Detecting Context in Distributed Sensor Networks by Using Smart Context-Aware Packets" In H.Schmeck, T. Unger, and L. Wolf (Eds): ARCS 2002, LNCS 2299, Springer Berlin Heidelberg (2002) 33-47

[7] Martin Raubal: "Ontology and epistemology for agent-based wayfinding simulation" In Int. J. Geographical Information Science 15(7) (2001) 653-665

[8] Marc Weiser: "The Computer for the 21st Century". Scientific American, 265(3) (1991) 94-104