Ubiquitous User Modeling

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

vorgelegt von
Dominikus Heckmann

Saarbrücken
15. November 2005

Dekan:
Prof. Dr. Jörg Eschmeier
Vorsitzender des Prüfungsausschusses:
Prof. Dr. Thorsten Herfet
Berichterstatter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Jon Oberlander (University of Edinburgh)
Promovierter akademischer Mitarbeiter der Fakultät:
Dr. Jörg Baus
Tag des Kolloquiums:
07.12.2005

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Dominikus Heckmann, Saarbrücken, 15. November 2005

Danksagung

Die vorliegende Arbeit entstand im *Internationalen Graduiertenkolleg Sprachtechnologie und Kognitive Systeme* an der Universität des Saarlandes und der Universität von Edinburgh, im Umfeld der Projekte READY und REAL, des von der Deutschen Forschungsgemeinschaft geförderten Sonderforschungsbereichs 378 *Ressourcenadaptive kognitive Prozesse*, der Nachwuchsforschergruppe FLUIDUM sowie des Projektes SPECTER am Deutschen Forschungszentrum für Künstliche Intelligenz.

Meinen ersten und ganz herzlichen Dank möchte ich Herrn Professor Wolfgang Wahlster aussprechen. Ich danke ihm für die Vergabe dieses interessanten Promotionsthemas, für seine zahlreichen Anregungen und das Interesse, mit der er mich und meine Arbeit begleitet hat. Mein zweiter ganz besonderer Dank gilt Professor Jon Oberlander für die sehr gute Betreuung in Edinburgh, die Übernahme der Zweitkorrektur und seine Besuche in Saarbrücken. Professor Anthony Jameson, Professor Antonio Krüger und Jörg Baus danke ich für die hervorragende Fachbetreuung in *User Modeling* und *Ubiquitous Computing*, insbesondere für die Vielzahl von Vorschlägen und Ideen. Doris Borchers, Claudia Verburg, Magdalena Mitova, Sabine Schulte im Walde, Professor Mathew Crocker, Professor Marc Steedman, Marie-Luise Fackler, Sabine Burgard, Andrea Placzková, Karima Bikry sowie der Infrastrukturgruppe danke ich für die Organisation, Verwaltung und Betreuung meiner Anliegen am Lehrstuhl, im Graduiertenkolleg, in Edinburgh und am DFKI. Allen Kolleginnen und Kollegen, die es mir ermöglicht haben, die Arbeit in dieser Form zu realisieren, möchte ich freundschaftlich danken - Sinéad Higgins, Rachel Reed, Dorothée Heckmann, Eric Schwarzkopf, Boris Brandherm und Rainer Wasinger ganz besonders für das Korrekturlesen. Für inspirierende wissenschaftliche Diskussionen, Forschungsbesuche oder besonderes Forschungsmaterial danke ich Henry Thompson, Professor Chris Mellish, Professor Alfred Kobsa, Josef Fink, Professor Paul de Bra, Professor Lora Aroyo, Professor Gert Jan, Vadim Chepegim sowie Professor Helmut Prendinger, Christian Kray, Thorsten Bohnenberger, Andreas Eberhart, Alexander Kröner, Patrick Gebhardt, Adam Pease, Vania Dimitrova und Francesca Carmagnola.

Von ganzem Herzen danke ich meinen Eltern, Geschwistern und Freunden, die mich während meines gesamten Studiums und meiner Promotion in jeder Hinsicht unterstützt und gefördert haben.

Acknowledgments

This work has been supported by the International Post-Graduate College *Language Technology and Cognitive Systems* at Saarland University and the University of Edinburgh. This research has also been supported by the German Science Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes (SFB 378) Project EM 4 REAL and Project EM 5 READY, as well as by the German Ministry of Education and Research (BMBF) under grant 524-40001-01 IW C03 within the project SPECTER at the German Research Center for Artificial Intelligence (DFKI).

Kurzzusammenfassung

Immer häufiger auftretende Interaktionen im täglichen Leben zwischen Menschen und vernetzten oder mobilen IT-Systemen bieten insbesondere für die Benutzermodellierung eine große Chance, durch ständige Evaluation des Benutzerverhaltens verbesserte Adaptionsleistungen zu erzielen.

Die vorliegende Arbeit entwickelt ein komplettes Rahmensystem, um dieses neu definierte Konzept der *ubiquitären Benutzermodellierung* zu realisieren. Die erarbeiteten Werkzeuge umfassen Methoden zum einheitlichen Austausch und zur semantischen Integration von partiellen Benutzermodellen. Sie berücksichtigen aber auch die erhöhten Anforderungen an die Privatsphäre, sowie das Recht der Menschen auf Introspektion und Kontrolle über die erhobenen Daten. Auf syntaktischer Ebene werden die situationsbeschreibenden Aussagen sowie die Austauschsprache `UserML` entworfen. Auf semantischer Ebene werden die allgemeine Benutzermodell-Ontologie GUMO und die UBISWELT-Ontologie entwickelt.

Ein mehrstufiger Konfliktlösungsmechanismus, der das Problem sich widersprechender Aussagen bearbeitet, wird zusammen mit einem webbasierten Benutzermodell-Service implementiert, sodass die Praxistauglichkeit und die Skalierbarkeit dieses Ansatzes an mehreren Beispielen gezeigt werden kann.

Short Abstract

More and more interactions take place between humans and mobile or connected IT-systems in daily life. This offers a great opportunity, especially to user modeling, to reach better adaptation with ongoing evaluation of user behavior.

This work develops a complete framework to realize the newly defined concept of *ubiquitous user modeling*. The developed tools cover methods for the uniform exchange and the semantic integration of partial user models. They also account for the extended needs for privacy and the right of every human for introspection and control of their collected data. The SITUATIONALSTATEMENTS and the exchange language USERML have been developed on the syntactical level, while the general user model ontology GUMO and the UBISWORLD ontology have been developed on the semantical level.

A multilevel conflict resolution method, which handles the problem of contradictory statements, has been implemented together with a web-based user model service, such that the road capability and the scalability can be proven with this approach.

Zusammenfassung

Ein Hauptanliegen der Forschung zur Mensch-Maschine-Interaktion ist die flexible Anpassung der IT-Systeme an die Interessen, das Wissen und das Verhalten der Benutzer. Auch die aktuelle, kontextbezogene Situation der Menschen spielt dabei eine immer wichtigere Rolle. Insbesondere durch den zunehmenden Einsatz mobiler und vernetzter IT-Systeme finden Mensch-Maschine-Interaktionen nun ständig und überall im täglichen Leben statt. Als zentrale Methode für benutzeradaptive Anwendungen hat sich die Benutzermodellierung bewährt, die jedoch die Technisierung des unmittelbaren Benutzerumfeldes sowie die häufigen Wechsel der benutzten IT-Systeme noch nicht ausreichend berücksichtigt.

Das Ziel dieser Arbeit war die Konzeption, Entwicklung und Anwendung von Werkzeugen zur *ubiquitären (allgegenwärtigen) Benutzermodellierung*, die einerseits die neuen Möglichkeiten des verändert-instrumentierten Benutzerumfeldes miteinbeziehen, andererseits aber auch die gestiegenen Anforderungen an Transparenz, Privatsphäre und Introspektion berücksichtigen. Die Kernhypothese dieser Arbeit besagt, dass eine permanente Evaluation der Benutzerinteraktionen mit unterschiedlichen benutzeradaptiven Systemen die Benutzermodelle verbessert und dadurch bessere Adaptionsleistungen möglich werden. Zunächst werden in dieser Arbeit die grundlegenden Konzepte und Definitionen aus den Gebieten Benutzermodellierung, Adaptive Systeme und Ubiquitous Computing vorgestellt. Darauf basierend wird ein Modell für *situierte Interaktion in instrumentierten Umgebungen* entworfen. Anschliessend wird eine Übersicht über eingesetzte Techniken des Semantic Web präsentiert und ein Überblick über verwandte Arbeiten aus den Gebieten der allgemeinen Benutzermodellierungssysteme, der Metadatenapplikationen und der externen Ontologien gegeben. Die vorgestellten und verglichenen Systeme (Doppelgänger, UM Toolkit, Personis, DeepMap), Applikationen (Dublin Core, Customer Profile Exchange, HumanML, CC/PP) und Ontologien (Cyc, SUMO/MILO, WordNet, FrameNet) führen zu den diskutierten Design-Entscheidungen des hier entworfenen neuen Ansatzes.

Die Motivation und Herausforderung, komplexe Situationen in einer einheitlichen Datenstruktur zu beschreiben, wird am Beispiel eines Reisenden an einem Flughafen erläutert. Der gezeigte Lösungsansatz führt allgemeine situationsbeschreibende Aussagen ein, definiert die mehrschichtige Semantik und bietet eine Vielzahl von syntaktischen Variationen an. Dabei wird beschrieben, wie diese sogenannten SITUATIONALSTATEMENTS in den neu entwickelten `UserML`-Berichten und `UserML`-Speicher zusammengefasst werden. Zusätzlich wird das URI-Konzept zu ganzen Ausdrücken von global-eindeutigen Referenzen erweitert. Als wesentlicher Beitrag dieser Arbeit werden folgende zwei neu entworfene und detailliert beschriebene Ontologien gesehen: (a) zum einen die allgemeine Benutzermodell-Ontologie GUMO, die eine Vielzahl von Dimensionen definiert, klassifiziert und mit Zusatzinformationen anreichert, (b) sowie die UBISWELT-Ontologie, die eine spezialisierte Simulationsumgebung für Ubiquitous Computing bildet. Ein Schwerpunkt liegt hier bei den räumlichen Konzepten. Beide Ontologien werden in die allgemeine Ontologie-Familie SUMO/MILO eingebettet und decken somit ein umfangreiches semantisch-definiertes Vokabular ab. Ein weiterer wesentlicher Beitrag dieser Arbeit wird in dem neu entwickelten Anfragemechanismus gesehen, der neben der syntaktischen Ebene auch semantische Funktionen integriert. Dabei werden sowohl die Vorteile der Massenspeicherung von situationsbeschreibenden Aussagen in Datenbanken als auch die Vorteile der ontologischen Inferenzkomponenten erläutert. Die Realisierung paralleler Anfragen auf verteilte Benutzermodelle wird beschrieben, wobei der integrierte Privacy-Algorithmus nur ausgewählte personenbezogene Daten zum Austausch frei gibt. Darüberhinaus wird ein umfassender Ansatz zur Auflösung sich widersprechender situationsbeschreibender Aussagen vorgestellt. Es werden zunächst die wichtigsten Konfliktarten kategorisiert, verschiedene Möglichkeiten aufgezeigt, diese zu erkennen und schliesslich beschrieben, wie sie durch wählbare Konfliktauflösungsstrategien aufzulösen sind. Darauf basierend wird die Integration von verteilten, partiellen Benutzermodellen anhand von zwei alternativen Inferenzmechanismen analysiert. Der syntaktisch-semantische Zusammenhang zwischen `UserML` und GUMO wird anschließend behandelt, wobei die direkte Einbettung der situationsbeschreibenden Aussagen in die Ontologie als mehrstellige Relationen realisiert wird. Ausserdem wird ein Datenbankschema präsentiert, das eine relationale Modellierung der Ontologie ermöglicht. Am Ende dieses Abschnittes wird noch der Einfluss mobiler Geräte auf die Benutzermodellierung analysiert und ein Ansatz zur Skalierbarkeit der verteilten Benutzermodelle aufgezeigt.

Im letzten Teil dieser Arbeit wird die Gesamtarchitektur eines Benutzermodellierungsservices entworfen, der sowohl in der realen Welt wie auch im Internet Anwendungen findet. Alle Komponenten werden prototypisch realisiert und an mehreren Beispielen aus den Bereichen der personalisierten Museumsführer, der Fußgängernavigationssysteme und der intelligenten Einkaufsassistenten getestet. Die Funktionsweise speziell entwickelter Werkzeuge, wie zum Beispiel des Ontologie-Editors, des Benutzermodell-Editors sowie des UBISWELT-Browsers, werden anhand von Bildschirmabzügen erläutert. Die einheitliche Interpretation und Verwaltung von verteilten, heterogenen Benutzermodellen wird in dieser Arbeit erstmalig durch die Verwendung von Techniken des Semantic Web realisiert. Mit diesem praxistauglichen und skalierbaren Ansatz wird gezeigt, dass permanente Benutzermodellierung mit einer Vielzahl unterschiedlicher IT-Systeme ermöglicht werden kann. Zum Abschluss werden die in dieser Arbeit erzielten Ergebnisse zusammengefasst und in die drei Forschungsbereiche Benutzermodellierung, Ubiquitous Computing und Semantic Web eingeordnet. Der Ausblick zeigt stichpunktartig, welche zukünftigen Forschungsthemen sich daraus ergeben können.

As we walk, our locomotion reveals our destinations.
As we talk, our speech reveals our intentions.
As we gesture, our motions reveal our thoughts.

As we read, our gaze reveals our focus of attention.
As we type, our keystrokes reveal our intentions.
As we surf the web, our clicks reveal our interests.

Jon Orwant - DOPPELGÄNGER PROJECT - [Orwant, 1995]

Contents

Part I

Introduction

1 MOTIVATION & OUTLINE

1.1 Motivation

Nowadays, most citizens interact with a variety of IT systems at work and home. They use cash machines and car navigation systems. They program their video-tape recorders and their central heating systems. They download photos from their digital cameras and music files for their iPods. They browse and search the internet or play computer games on their mobile phones. However, most of them are unaware of the variety and the amount of IT systems that they use every day.

Concurrently, more and more of these IT systems are getting interconnected via the internet. Prominent visions even predict the complete integration of all these systems into a federation of communicating IT systems that could be described as so-called "intelligent environments." Furthermore, there is a tendency pointing towards an "always-on" paradigm, where each citizen is constantly connected to the internet.

If we now manage to unify all user related assumptions (that are currently applied by these systems individually) into one consistent model, then we could expect several improvements in relation to the existing situation (without information sharing and model integration). The motivating key hypothesis[1] is the following: *ongoing evaluation of user behavior with systems that share their user models will improve the coverage, the level of detail, and the reliability of the integrated user models and thus allow better functions of adaptation.*

The identified improvements are:

- *increased coverage:* more aspects will be covered by the aggregated user model, because of the variety of the contributing systems with user modeling components.
- *increased level of detail:* if the same aspects are modeled by several related systems, the integration of these user models will lead to an increased level of detail.
- *increased reliability:* much of the evidence that a system can obtain about the user behavior is unreliable and uncertain; however, with the means of exchanging and inte-

[1]This key hypothesis will be refined and formally defined in section 1.3

grating user models, more evidence will be available for all systems, and thus the level of uncertainty will decrease and the level of reliability increase.

Another initial motivation for this work lies in Jon Orwant's DOPPELGÄNGER PROJECT, see e.g. [Orwant, 1995], where he phrases the three claims 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*.

The identified key challenge for my doctoral research is to enable a variety of user-adaptive systems to share their user models. Figure 1.1 shows a characteristic example to illustrate the conceptual advantage of ubiquitous user modeling.

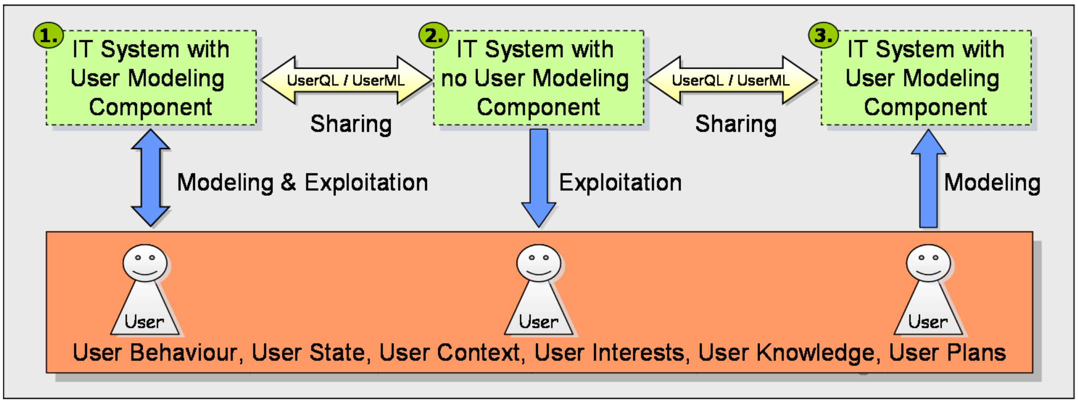


Figure 1.1: *An abstract view of ubiquitous user modeling with the focus on modeling, sharing and exploitation of user related data*

The second IT system in the middle of this diagram has no user model acquisition component, which means that it could hitherto not provide user adaptation that is based on most recent information about the user. However, the introduced ubiquitous user modeling enables the two neighboring IT systems 1 and 3 to share their user models. Thus, IT system 2 is able to exploit the shared user models and demonstrate an adequate, user-adaptive interaction. In contrast, IT system 3, which has no explicit user model exploitation component, is able to contribute assumptions to the shared user models. A central question that arises is: *How can systems and environments share their user models?*

1.2 Application Scenario

For a concrete example, let's consider what we call the *airport scenario*, which is defined as follows: You are planning a flight at your office with a web service that is already user-adaptive and creates the first user model. Now, you drive to the airport and arrive at an information kiosk, which already knows your goals and your preferences from the web service. The system's reactions to your needs are adapted to your long-term user model as well as to the actual level of noise at the current location. You are navigated through the airport with a hand-held device that considers current information about the airport shops and recommends products to you. The motivating question that arises is: *How does such an integrated scenario influence user modeling techniques?* If the airport information kiosk realizes that you are under time pressure, it could forward this information to the surrounding adaptive systems like the pedestrian navigation guide. The detection of being under time pressure could, for example, be realized by analyzing your typing behavior and your speaking style. The navigation guide could then optimize the suggested path, taking time constraints into consideration, rather than entertainment preferences. Figure 1.2 shows one traveler in four different situations.

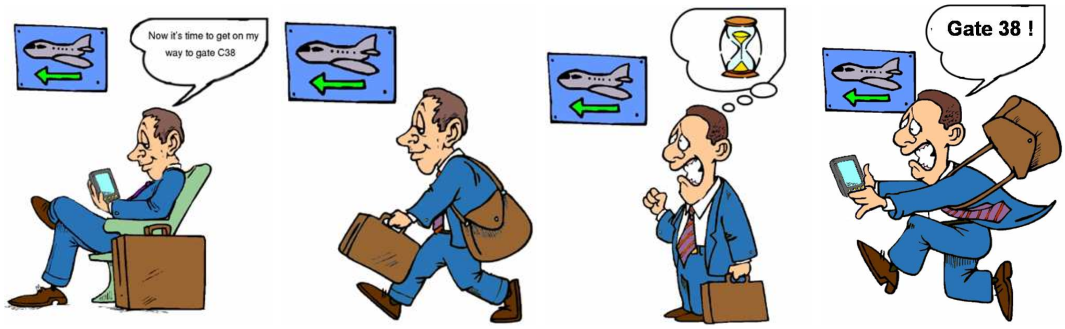


Figure 1.2: *Various traveler situations: in the first one (from left to right) he seems to be relaxed, waiting and chatting with his interaction device. In the second one, he is walking and carrying two pieces of luggage. In the third situation the traveler is not moving, but realizing that he is under time pressure. In the fourth situation the traveler is running, using his hand-held device and talking louder, uttering only "Gate 38."*

The task to guide someone quickly through a large airport[2] is user-dependent but also situation-dependent, i.e. unpredictable waiting times at check-in or hand luggage checkpoints have to be taken into consideration, as well as information about the user's capability to use stairs, or whether he is carrying heavy luggage. This airport scenario, that was introduced in the *Collaborative Research Center on Resource-Adaptive Cognitive Processes*[3] at Saarland University, pointed out that there is a need to do research on ongoing user modeling with a variety of systems, at different locations and situations and that communication between independent user-adaptive systems is important.

[2]The complex task of planning optimal routes with multiple goals and resource limitations as described in the airport scenario has been realized in a parallel PhD thesis by Thorsten Bohnenberger with hidden Markov models, see [Bohnenberger, 2004].

[3]The work was embedded in the so-called "airport scenario" of the Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378, Project READY, see e.g. [Bohnenberger et al., 2002]

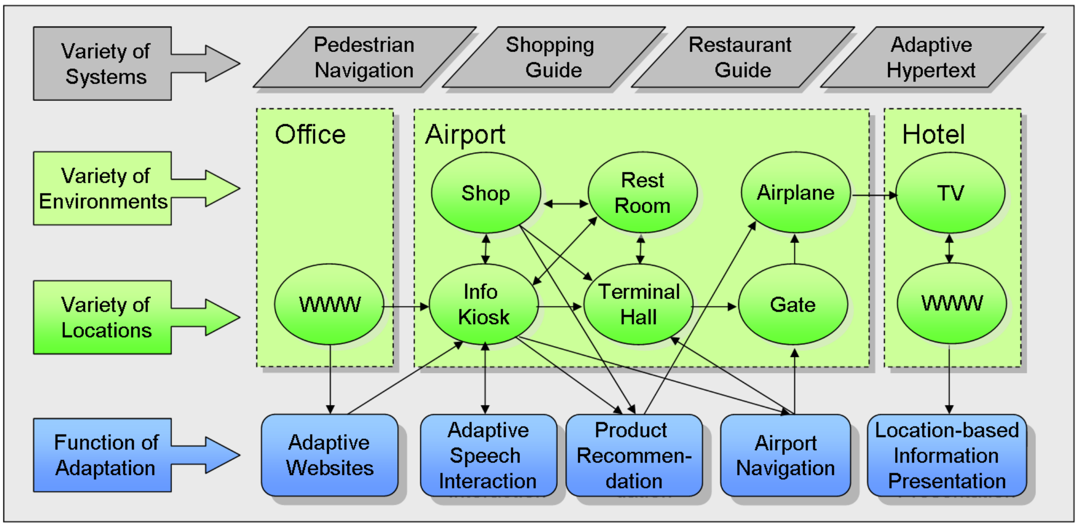


***Figure 1.3**: The airport scenario as a test bed for ubiquitous user modeling: the central research scenario is the setting of a large airport (e.g. Frankfurt Airport), where travelers navigate, for example, through a shopping mall with restaurants and from the departure halls to the boarding gates, using mobile, wearable or ubiquitous technologies. A user plans a flight at his office with an adaptive web service and buys a ticket. Later, at the airport he speaks to an information kiosk to receive airport information, he visits some shops and buys some things. He navigates through the airport to the gate and enters the airplane. At the destination, he interacts with the local hotel reservation system and receives location-based information. The arrows "→" in this diagram indicate the information flow.*

1.3 What is Ubiquitous User Modeling?

Ubiquitous user modeling means that the user's behavior is constantly tracked at any time, at any location and in any interaction context. Furthermore, the various user models are shared, merged and integrated on demand. The following definition defines this new concept:

> **Definition 1.1 (Ubiquitous User Modeling)** *Ubiquitous user modeling describes ongoing modeling and exploitation of user behavior with a variety of systems that share their user models.*

These shared user models can either be used for mutual or for individual adaptation goals. Ubiquitous user modeling can be differentiated from *generic user modeling*[4] by the three additional concepts: *ongoing modeling*, *ongoing sharing* and *ongoing exploitation*. It covers and integrates the two following aspects:

1. *User modeling for situated interaction in ubiquitous computing environments.*
   The shift in human-computer interaction from *desktop computing* to mobile, real-world

[4]see section 2.1 for an introduction to generic user modeling, based on [Wahlster and Kobsa, 1989] and [Kobsa, 2001a]

interaction in augmented and instrumented environments, as introduced in the research area of ubiquitous computing, highly influences the needs and possibilities for future decentralized user-adaptive systems. A detailed concept of *situated interaction* can be found in section 2.2.4.

2. *Ongoing user modeling with a variety of systems and applications.*
   The idea is to enable (so far) isolated user modeling applications to exchange partial user models with each other. While *adaptive hypermedia* systems on the world wide web are already provided with the means to communicate, wireless networks technically allow one to integrate any kind of user-adaptive systems with another; however, the challenge is the semantic integration of the distributed heterogeneous partial user models to enable long term user modeling.

Both subfields perfectly complement one another and promise interesting results; however, ubiquitous user modeling also implies new challenges of *scalability*, *scrutability* and *privacy*. The new issues of *decentralization*, *communication* and *integration* have to be addressed. A complete conceptual overview of our suggested approach for ubiquitous user modeling can be found in figure 7.1 on page 141.

"Ubiquitous user modeling" is a new, compound term that has been coined by my supervisor Wolfgang Wahlster while defining the title of this thesis and identifying the relevant research topics. The research ideas behind ubiquitous user modeling have been introduced for the first time in [Heckmann, 2001]. The corresponding acronym `U2M` has been derived via `UUM` and $\mathtt{U}^2\mathtt{M}$ and can be read as "*U-squared-M*". It lead to the internet domain `http://www.u2m.org`, where all research activities of *ubiquitous user modeling* are presented.

1.4 Orientation and Integration into Related Research Areas

How can we link and integrate ubiquitous user modeling into existing, related research areas?

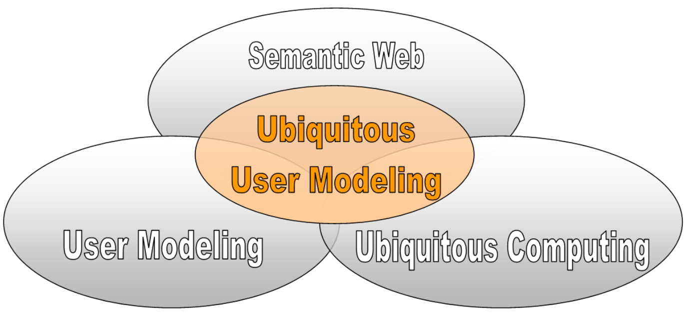


Figure 1.4: *The intersection and interrelation of the three main research areas user modeling, ubiquitous computing and semantic web leads to "ubiquitous user modeling."*

Ubiquitous user modeling can be interlaced with three prominent research areas:

- *user modeling*, which manages knowledge sources containing explicit assumptions on all aspects of the user that may be relevant for a system's interaction behavior
- *ubiquitous computing*, which integrates computation into the environment, rather than having computers which are distinct objects
- *semantic web*, which intends to create a universal medium for information exchange by giving meaning to the web, in a manner understandable by machines

In fact, we consider ubiquitous user modeling as describing the intersection of these three research areas. After the investigation of these three related research areas, the main task of this thesis was to find effective methods for joining user modeling with ubiquitous computing, user modeling with the semantic web approach, ubiquitous computing with the semantic web approach and finally, the combination of all three research areas into the new discipline of *ubiquitous user modeling*. Figure 1.4 illustrates this view.

1.5 Main Research Questions

The presented research in this thesis focuses on the issue of formalizing and exchanging user model knowledge in the era of semantic web and ubiquitous computing. Special focus is set on the following research questions, that are listed below. A short summary of the results, or of our approach in general, is stated after each question.

1. *How can we conceptualize the complex process of situated interaction for ubiquitous user modeling?*

   We investigate situated interaction and define a clear model for situated human-computer interaction in ubiquitous computing environments that integrates all relevant elements.

2. *How can a user model exchange language be designed that is especially well-suited for the communication between different user modeling applications ?*

   We develop the RDF-based user model exchange language `UserML`, which builds on the new concept of SITUATIONALSTATEMENTS. The innovation is that the semantics in this approach is consequently defined in external semantic web ontologies.

3. *How can partial user models be communicated via the information structures within intelligent environments?*

   We implement centralized and decentralized, as well as global and local web service networks with XML messaging that allow wireless and network-connected communication via HTTP protocols. The implemented `u2m.org` user model service has been used by several ubiquitous applications to test the communication under real conditions.

4. *How can knowledge about user model dimensions be well organized for a semantic web ontology?*

We investigate existing user modeling systems and deduce the general user model ontology GUMO with specialized, domain specific extensions. It is represented in the semantic web languages DAML+OIL and OWL. Additionally, we implement an online ontology editor and tree browser that enable the distributed extension and refinement of this ontology.

5. *How can the changing physical and virtual environment around the user be represented uniformly?*

   We build an extended blocks world with the name UBISWORLD on top of our new human-environment interaction model for ubiquitous computing. The corresponding UBISONTOLOGY defines a large T-Box and several A-Boxes to represent the world. Furthermore, we design a state of the art hybrid location model to represent symbolic and geometric spatial references.

6. *How can entities like locations and objects be identified uniquely and efficiently in distributed real-world applications with multi-user and multi-systems?*

   Since more powerful, globally unique identifiers were needed for ubiquitous user modeling, we investigate related concepts and develop the unique naming concepts `UbisIdentifier` and `UbisExpression` as extensions to the URI naming concept.

7. *How can users inspect and control their distributed user models?*

   We realize a user model editor as a web browser application for mobile and large-screen devices with which the user can inspect, change, delete and control his/her personal data within the distributed user models.

8. *How can the huge amount of (sensor) data be handled technically by the server?*

   Our approach allows the smooth integration of ontologies and distributed databases by offering `UserML` models in RDF and SQL. We define a set of hierarchical databases that balances the load dynamically according to days, weeks, and months or according to other predefined groups.

9. *How can situation retrieval and conflict resolution be managed in such a distributed approach?*

   We develop a smart situation retrieval mechanism with a multi-level conflict resolution strategy that resolves requests according to given preferences on meta information within the queries. The conflict resolution algorithms can operate on several distributed user models in parallel.

10. *How can the integration of instantiated partial user models be realized within ubiquitous user modeling?*

    We investigate integration methods and present a user model integration method that reverts back to the conflict resolution strategy for situational repositories. Partial user models need not to be free of conflicts, but they are resolved for each individual request at runtime.

11. *How does a reasonable architecture for decentralized user modeling look?*

    We introduce an architecture of a multi-fold decentralized user model service that can be distributed within intelligent environments. Furthermore, it operates on spatially distributed partial user models and the semantic layer can even be distributed throughout the whole net of systems.

1.6 Organization of the Thesis

The thesis is organized into five main parts with several chapters each. Figure 1.5 shows the relations between these parts and chapters and indicates the suggested path for reading.

In *Part I: Introduction*, the first chapter motivates the presented research issues and outlines our approach. Chapter two covers the needed background and basic concepts, namely the introduction to user modeling, privacy and context-awareness, the introduction to situated interaction and ubiquitous computing, and finally the introduction to semantic web and web services. Chapter three discusses related work, describes the state of the art and lists our goals and design decisions for the developed new contribution.

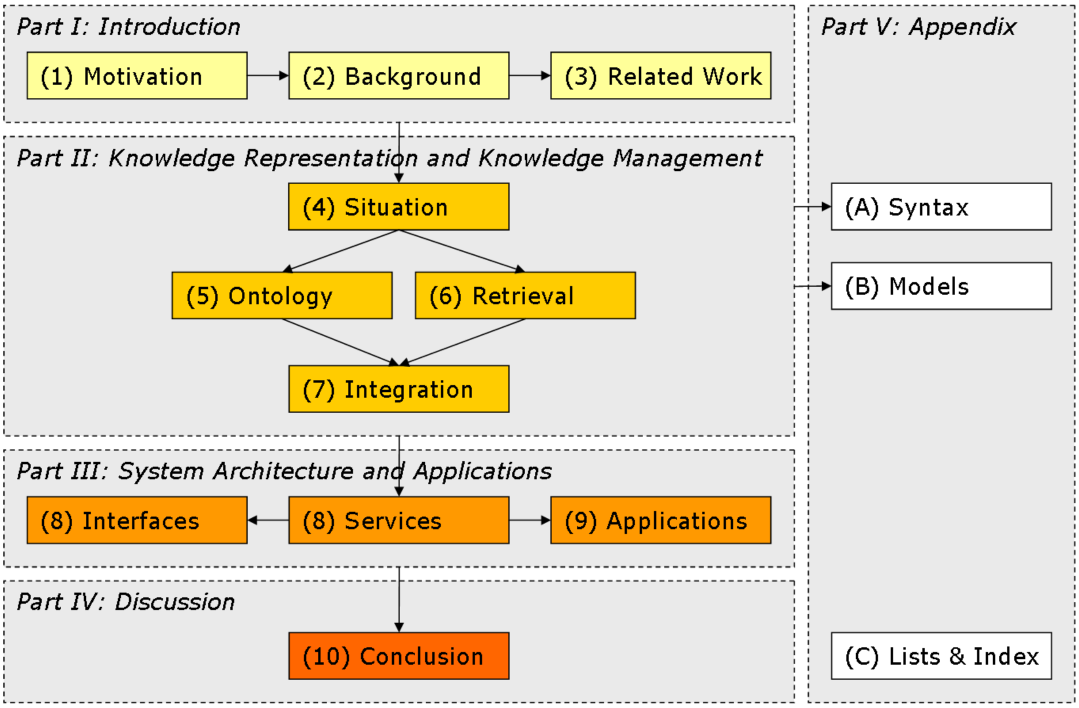


***Figure 1.5**: Structural organization of this thesis. The arrows between the chapters and parts indicate the suggested path for reading*

Part II: Knowledge Representation and Knowledge Management starts with the chapter about so-called SITUATIONALSTATEMENTS and SITUATIONREPORTS, where the syntactical issues of structured meta data are analyzed and the user model markup language `UserML` is defined. Chapter five arranges all user model dimensions into the general user model ontology GUMO and introduces UBISWORLD as a blocks world for ubiquitous computing. Chapter six defines SITUATIONALQUERIES and `UserQL` that enable powerful retrieval mechanisms. Chapter seven covers conflict resolution and our solution to the issue of user model integration.

Part III: System Architecture and Applications presents the implemented USERMODELSERVICE architecture with its various possibilities of adding and retrieving statements. Chapter eight also shows user interfaces and tools that were developed especially for the ubiquitous user modeling approach. Chapter nine tests the approach by demonstrating ubiquitous applications that use or integrate the newly developed tools successfully.

Part IV: Discussion points out the scientific contributions, concludes the work and discusses further research opportunities.

Part V: Appendix contains additional material like syntax definitions, ontology listings and lists of acronyms, figures, tables as well as the index.

2 Background & Basic Concepts

Information is a tool designed by human beings to make sense of a reality assumed to be both chaotic and orderly as stated by [Dervin, 1999]. We start this chapter with thoughts about *Information Design*, since information imposes order on a chaotic reality like situated real-world interaction. The following four concepts of information have been posted in history:

- Information describes an ordered reality that varies across time and space.
- Information describes an ordered reality that varies from culture to culture.
- Information describes an ordered reality that varies from person to person.
- Information describes an ordered reality but can be "found" only by those with the proper observing skills and technologies.

These fundamental basic concepts of information already tend towards user-adaptation and location awareness. Historically, information was conceptualized as a natural description of natural reality. According to [Dervin, 1999] this is still the dominant conceptualization assumed in the design of information systems nowadays. The next step is to integrate and extend the concept of information in the *Continuum of Understanding* from data to wisdom, see [Shedroff, 1999]. Figure 2.1 visualizes the process of understanding into the arrows from *data* (covering: creation, gathering and discovery) to *information* (covering: presentation and organization), *knowledge* (covering: conversation, storytelling and integration) and finally *wisdom* (covering: interpretation, evaluation and retrospection).

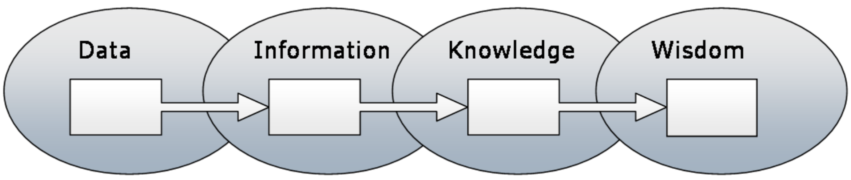


***Figure 2.1**: The simplified continuum of understanding, according to [Shedroff, 1999]*

These four meta levels of understanding play a major implicit role throughout our whole approach and should be borne in mind. The next pre-discussed point in this section about basic concepts is *Clarity in Information Design*. According to [Shedroff, 1999], the most important goal of effective communication is clarity. However, clarity is not the same as simplicity. If the message is about a complex relationship this complexity can be made clear through effective organization and need not to be reduced to simplicity! And according to [Larkin and Simon, 1987] the most clear representation of information - either diagrammatic or textual - depends on the intended tasks and operations on this information. This argument has especially been recognized in the design of SITUATIONALSTATEMENTS that decompose complex situations into the clear structure of *reports*, *statements*, *attributes* and *elements*.

2.1 Introduction to User Modeling and Context-Awareness

Adaptive technologies are those in which the behavior of the user/reader/consumer/actor changes the experience. According to [Jacobson, 1999] this appearance of intelligence alludes to a much larger question about intelligence, life, and how these are defined. It is sufficient to realize that making certain kinds of choices to change behavior based on the actions of others (whether instinctive or algorithmic) can create the appearance of a sophisticated system or process and imply a kind of independent intelligence. Two of these adaptive technologies are *user modeling* and *context-awareness*. Such systems must be able to observe the user's behavior and changes in the context and make generalizations and predictions about the user and the context based on their observations. The advantages of an integrated treatment of user modeling and context-awareness has for example been discussed in [Jameson, 2001b] and in [Heckmann, 2003b]. The information about the user is usually collected in a so-called *user model* and administrated by a *user modeling system*, see [Wahlster and Kobsa, 1989]. They define (in the context of a dialog system) the following two fundamental concepts:

> **Definition 2.1 (User model)** *A user model is a knowledge source in a system which contains explicit assumptions on all aspects of the user that may be relevant to the behavior of the system. These assumptions must be separable by the system from the rest of the system's knowledge.*

> **Definition 2.2 (User modeling component)** *A user modeling component is that part of a system whose function is to incrementally construct a user model; to store, update and delete entries; to maintain the consistency of the model; and to supply other components of the system with assumptions about the user.*

All systems that perform an adaptation to the individual user in some nontrivial way are defined in [Jameson, 2003] as *user-adaptive systems*. Several generic user modeling systems with the aim to facilitate the provision of user modeling services to application systems have been described in [Kobsa, 2001a]. However in most of these systems, user modeling functionality is an internal part of the user-adaptive application. In [Brusilovsky, 1996] and [Brusilovsky et al., 1998] methods and techniques for *adaptive hypertext and hypermedia* have been introduced or analyzed, while in [Wu, 2002] a reference architecture for such applications has been developed. Research on the requirements, design, and evaluation of user modeling servers is presented in [Fink, 2004]. The context of building commercial software is also discussed there, while in e-commerce the more generic term *personalization* is preferred, which denotes user-adaptive system features and user modeling issues at the same time. Several generic user modeling systems are discussed in more detail in section 3.1.

2.1.1 Conceptual View of Context-Aware User Modeling

A conceptual view of the theory of user modeling with integrated context-awareness is presented in figure 2.2. This concept influenced the one for distributed user modeling as defined

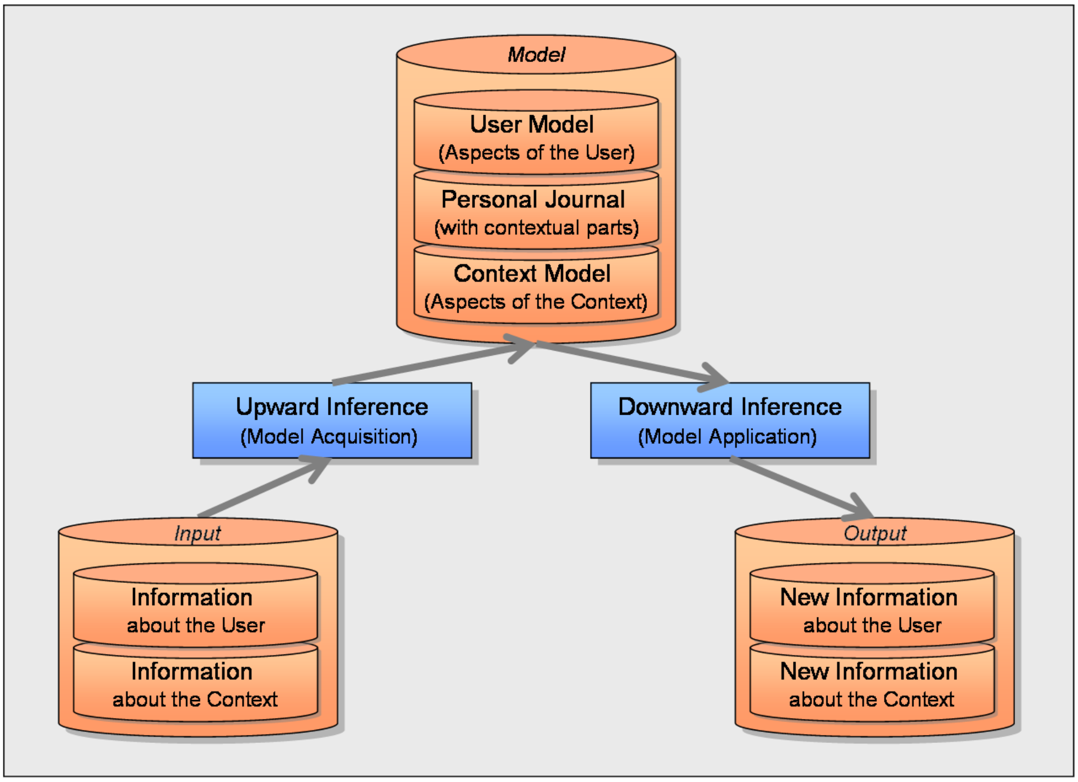


***Figure 2.2**: Extended processing schema within context-aware user-adaptive systems, derived from [Jameson, 2001a], [Jameson, 2001b] and [Kleinbauer et al., 2003]*

in chapter 7. Input data concerning the user and input data (or just information) concerning the context is processed in an upward inference step (like machine learning techniques or instantiation of Bayesian networks). This *model acquisition* adds inferred aspects about the user to the *user model* or to the *personal journal*. The personal journal has been introduced in the SPECTER[1] project, see e.g. [Kröner, 2005]. It combines user data with contextual parts and provides the user with a diary-like view of the collected information. The downward inference step or so-called *model application* calculates new hypotheses about the user or the

[1] SPECTER homepage: `http://www.dfki.de/specter/`

context, for example by applying rules or evaluating Bayesian networks. According to the various examples in [Jameson, 2001c], this basic theory proved very helpful for describing existing user-adaptive systems.

2.1.2 Requirements for User-Adaptive Systems

Prominent functions for user-adaptive systems are *help the user to find information*, *tailor information to the user*, *product recommendation*, *interface adaptation* (e.g. [Kröner, 2001]), *giving help*, *support learning*, *dialog adaptation* and *support collaboration*. A classification and discussion about the benefits and limitations of such systems can be found in [Jameson, 2001c], where also the requirements for user-adaptive systems are described by the following questions:

- What function is to be served by the adaptation?
- What properties and dimensions of the users (and context) should be modeled?
- What type of input data about the user (and context) should be processed?
- How should the system make the necessary inferences and decisions about the user?
- How should the adaptation be evaluated?

Two interesting new questions that arise in *ubiquitous user modeling* are concerning the need for communication of partial user models and the consistency for user model integration:

- How should user model information be exchanged?
- How can distributed user model information be merged, synchronized and fused?

In [Fink, 2004] requirements for user model servers and user modeling are analyzed and the following topics are identified for the servers: *multi-user synchronization*, *transaction management*, *query and manipulation language*, *persistency*, *integrity* and *access control*. Furthermore the following user modeling requirements are identified: 1) *offered functionality* such as acquisition of user-related information, user modeling and user-related adaptations, 2) *data acquisition* including user data, usage data and acquisition methods (such as acquisition rules, statistics, case-based reasoning, decision trees, neural networks, stereotype reasoning and group model reasoning), 3) *representation methods* (like attribute-value pairs, graph-based representations or production rules), 4) *extensibility and flexibility*, 5) *integration of external user and usage information*, 6) *privacy* and 7) *inspectability* of user model contents.

User-related factors like the abilities and properties of individual people, the intention as well as the current goal and the user's interests and preferences influence the interaction with a system. The emotional state can also play an important role, as argued in [Picard, 1997]. All user-related factors are categorized and analyzed in detail in chapter 5.2, where the general user model ontology GUMO is defined. In addition to these user-related factors, contextual factors that are determined by the environment have an impact on various reasoning processes. The current weather conditions affect outdoor activity, for example if it is raining hard. The acoustic state of the environment, e.g. whether there is a lot of noise is a further factor. Additionally, the computational resources that are available to a system have a great impact on all reasoning processes, as they impose hard constraints on the size of problems that can be handled in a timely fashion, see [Blocher, 1999]. Some of these factors are called resources. They are discussed in the following subsection.

2.1.3 Resource-Adaptive Systems

The term *resource* is an abstract and general concept that is defined in the WordNet dictionary, see section 3.3.3, as *a source of aid or support that may be drawn upon when needed.* In the research area of human-computer interaction, this concept is defined as the *available means to solve a task*, see [Jameson and Buchholz, 1998]. Two main types of relevant resources[2] are distinguished: *cognitive resources* and *technical resources*, as pointed out in [Kray, 2003], which we will briefly discuss below. *Resource-adaptivity*[3] overlaps with *user-adaptivity* and *context-awareness* since the human's cognitive resources fall into the user model, while the system's technical resources can be seen as part of the context model, see figure 2.4 on page 23.

> **Definition 2.3 (Resource-aware systems)** `Resource-aware systems` *are systems that are aware of what resources are available to them at any given time.*
>
> **Definition 2.4 (Resource-adaptive systems)** `Resource-adaptive systems` *are systems that are aware of what resources are available to them at any given time, and that employ a predefined adaptation strategy to perform well when faced with such a situation of varying resource availability.*

It is interesting to consider the reaction mechanisms to changes in the current resource situation: one can think of different strategies if certain resources are not available. A practical example of the reaction of resource-awareness is the power management strategy of notebooks: "If the current power supplier is the battery and the battery is low, then the display's brightness is reduced in order to save energy." More sophisticated approaches of the provided service are imaginable. [Wahlster and Tack, 1997] distinguish between three related types of resource-aware processes:

- resource-adapted processes,
- resource-adaptive processes and

[2]Not to be confused with the term *resource* in the semantic web where the "Resource Definition Framework (RDF)" and the "Unified Resource Identifier (URI)" are described, see section 2.3.

[3]This research has partially been supported by the German Science Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378.

- resource-adapting processes.

Resource-adapted processes have been adapted to resource restrictions that are not only previously known but also static. *Resource-adaptive processes* employ a predefined adaptation strategy to perform well when faced with such a situation of varying resource availability. *Resource-adapting processes* address this issue by dynamically generating adaptation strategies or by switching between several ones.

The behavior of resource-adapted processes is deterministic so that the quality of results is directly determined by the input. However, this kind of resource adaptation does not cope well with varying or new resource restrictions. Resource-adaptive processes result in a less deterministic output quality, which is mainly determined by what resources are available during computation. The main drawback of this approach consists in its inflexibility on the strategic level which is solved by the resource-adapting processes. They may also allow for adaptation on the meta-level by analyzing past resource restrictions and evaluating the success of various adaptation strategies.

Cognitive resources are all types of resources that influence the cognitive processes a human performs. Since there is an ongoing discussion about how human cognition works and how it is structured, there is no single model of cognition. Consequently, it makes sense to focus on resource restrictions that have been documented as affecting cognition. In our approach we take the standpoint of [Kray, 2003] and assume that cognitive resources are a subclass of user-related factors that impact cognition for any task the user is performing, unlike for example familiarity with the environment, which only influences some tasks and that are not intrinsically tied to the users body functions like age or physical condition. However, consider for example the emotional state of a person. While emotions are not necessarily required for all tasks a human is performing, it has been argued that they are nevertheless involved in most if not all cognitive processes, see [Picard, 1997]. Nevertheless, `cognitive load` and `time pressure` are examples of cognitive resources that influence a person's ability to reason, judge and perceive, see e.g. [Lindmark and Heckmann, 2000]. Cognitive resources are defined as part of the GUMO ontology in chapter 5.2.

Technical resources are seen as part of the context model. By analogy to human beings, artificial systems hold certain resources that impact their performance in various ways. This concept does not only include factors that are directly related to the software and hardware constituting the system but also some factors that are influenced by the current context. One important technical resource is the computational power that is available to solve a given task. In many cases, this resource directly determines whether a task can be performed at all and how long the process will take. The same applies to the available memory as well as to the bandwidth that can be used in the communication between different components. Even though most personal computers have nowadays fast processors and large memories, in ubiquitous computing we deal with small embedded or mobile devices, where these technical resources still play a major role. Furthermore, the means by which a system interacts with the user and perceives its environment play an important role. This includes not only the `screen size`, `screen resolution` and `screen colors` but also the means for generating `audio output` for example. Similarly, on the input side, technical resources include `keyboards` and `pointing devices`. Additionally, the means by which a system can perceive its environment are considered as technical resources. These include sensors such as `microphones`, `cameras`, or positioning devices like `GPS`. All these technical resources are defined within the UBISONTOLOGY in chapter 5.3.

2.2 Introduction to Ubiquitous and Situated Interaction

Ubiquitous computing leads to
a pleasant and effective "place" to get things done. [Weiser, 1991]

2.2.1 Motivation for Ubiquitous Computing

Mark Weiser's classification of a ubiquitous computing system is based on two fundamental attributes: namely *ubiquity* and *transparency*, see [Weiser, 1993]. Ubiquity denotes that the interaction with the system is available wherever the user needs it. Transparency denotes that the system is non-intrusive and is integrated into the everyday environment.

In [Maffioletti, 2001] the requirements for a ubiquitous computing infrastructure are analyzed and the general idea of a middleware that provides the basic functionality for modeling interactive environment applications is presented. They define:

Definition 2.5 (Ubiquitous Computing System) *A ubiquitous computing system consists of a heterogeneous set of computing devices, a set of supported tasks, and some infrastructure on which the devices rely on in order to carry out their tasks.*

In [Salber et al., 1999b], Abowd identified according to Weiser's classification, two dimensions that provide a clear boundary for ubiquitous computing systems and expresses the relationship with other emerging research areas such as *mobile computing*, *augmented reality* and *wearable computing*, namely *user mobility* and *interface transparency*. User mobility reflects the *degree of freedom* that the user has to move about, when interacting with the system. Interface transparency applies to the system's interface and reflects the conscious effort and attention that the system requires of the user, either for operating it or for perceiving its output. The spatial arrangement in the diagram 2.3 indicates that ubiquitous computing tries to maximize the user mobility and the interface transparency, while in contrast, desktop computing offers no user mobility and mostly no transparency to the end user.

Current research in ubiquitous computing leads toward the development of interactive environments that enable the mobility of both users and computing devices. The vision of ubiquitous computing relies according to [Coen et al., 1999] on the presence of so-called *intelligent environments* enriched by computers embedded in everyday objects like blackboards, tables, chairs and enriched by sensors able to catch information from the context. According to [Maffioletti, 2001] these environments represent the spatial boundaries of applications integrated in our everyday context, they represent the physical space where the applications are placed and executed. It is distinguished between a *service dimension* that

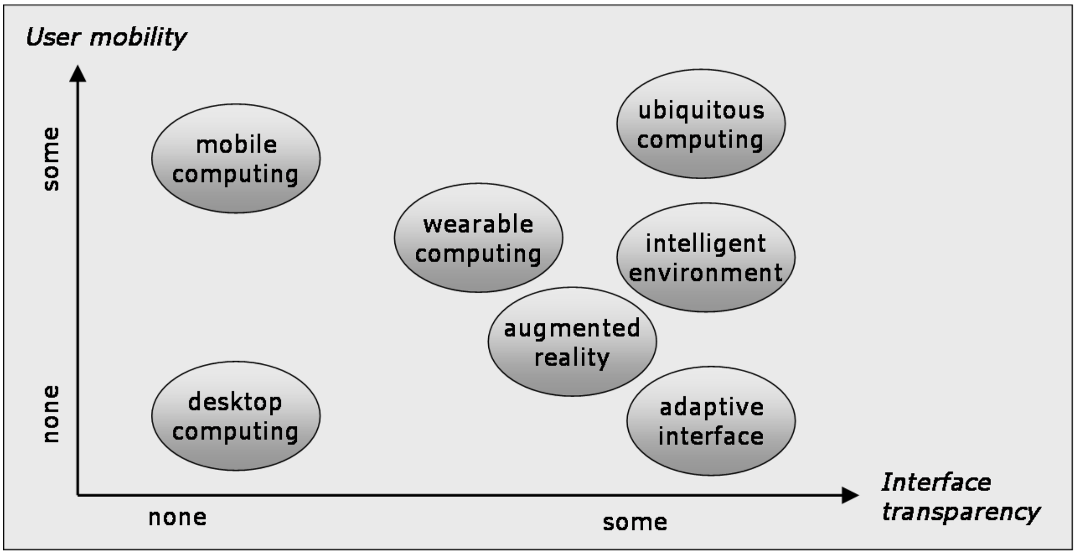


Figure 2.3: *User mobility and interface transparency dimensions that characterize research streams relevant to ubiquitous computing, according to [Maffioletti, 2001]*

represents the number of services that is available in the system and a *device dimension* that represents the number of devices incorporated in the environment. Intelligent environments have a large number of hardware and software components that need to cooperate. They tend to be highly dynamic and require reconfiguration and resource management on the fly as their components and inhabitants change. Further inspiring statements from [Weiser, 1991]:

- Ubiquitous computing is fundamentally characterized by the connection of things in the world with computation.
- The main idea of ubiquitous computing: integrate computing into objects of daily life but hide its existence if possible.
- Things in the world can be actively supported by integrating computing devices or adding additional identification badges or labels. Things can be connected into Intelligent Environments via e.g. wireless lan.
- The real power of the concept comes not from any one of these devices; it emerges from the interaction of all of them. The hundreds of processors and displays are not a "user interface" like a mouse and windows, just a pleasant and effective "place" to get things done.

2.2.2 Aspects of Mobile Computing

Since the interaction with intelligent environments supposes that the user can move around, results from the research area of mobile computing can be integrated into the situated interaction within ubiquitous computing. See for example [Aslan and Krüger, 2004] for a complex wearable mobile interaction platform. The following fundamental differences between the interaction with mobile devices and the interaction with desktop systems are listed in [Malaka, 2003]:

- Device related aspects in mobile scenarios are
  - smaller displays
  - keyboard input is slower
  - mouse-interaction is not predominant
  - voice control is more practical
  - less standardized and high variability in their input and output capabilities
- Human related aspects in mobile scenarios are
  - the focus of attention needs not to lie on the device
  - the user does other things concurrently like walking, driving or watching
  - many stimuli from the environment most likely distract the user
  - there are more time-constraints
  - influence on the user's security

Among other techniques for computer-mediated interaction with the environment, we focus in the following subsection on the concept of situation and situated interactions. Flexible and symmetric multimodal human-machine interaction in mobile environments are for example discussed in [Wahlster, 2003].

2.2.3 Situation Semantics

An inherent characteristic of situations in our everyday life is according to [Wrightson, 2005] that they are not completely known, and two people can talk about the same situation, each contributing observations new to the other.

The *Situation Semantics* that has been developed by Jon Barwise and John Perry, see [Barwise, 1981] and [Barwise and Perry, 1983], offers a framework to conceptualize everyday situations. First, the actual world can be thought of as consisting of situations. Second, situations consist of objects having properties and standing in relationships. A situation naturally carries with it a time and place. It is assumed that any actual situation is far too rich in detail to be captured by any finite process which leads to the concept of *situation-types*. Thus any perception of a situation, any believe about a situation, or any natural language description of a situation designates situation-types, see [Evans, 1981]. In the abstract model, situation-types are partial functions that characterize various types of situations. Totally understanding a situation would presume, that one is able to derive a situation-type which includes all the objects, properties, and relationships that exit in this situation.

An interesting concept in situation semantics is the so-called *infon*. An infon[4] is a formalization of a single piece of information. The information that *Peter is under time pressure at time t_i* and *Peter is not happy at time t_i* is expressed by the two following infon notations:

[4]Extended infons lead to the concept of SITUATIONALSTATEMENTS that are defined in section 4.2

$$<< Peter, timepressure, t_i, 1 >>$$

$$<< Peter, happy, t_i, 0 >>$$

The “1” or “0” at the end is the so-called *polarity* of the infon, where 1 is positive and 0 is negative. However, these are distinct concepts form the truth values “true” and “false”, since an infon is not an assertion. Facts about the world are represented as polarized infons in situation semantics. Infons are seen as semantic objects and not as syntactic representations, thus the denotation plays a minor role.

An inherent characteristic of exchanging situations is according to [Wrightson, 2005] that different people will deduce different knowledge from the same new infons, since they will apply different constraints linking the new infons to other situations.

2.2.4 What is Situated Interaction?

The fundamental data structure in our approach is the one of SITUATIONALSTATEMENTS, see chapter 4. Here, in this prefacing subsection the following issue is investigated: *What is situated interaction and how can it be conceptualized?* We start our discussion with the following definition of *Intelligent User Interfaces (IUI)* by [Maybury and Wahlster, 1998][5]:

> **Definition 2.6 (Intelligent User Interfaces)** *Intelligent user interfaces are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction.*

To realize these properties, intelligent user interfaces employ explicit user models, discourse models, domain models, task models and available media models for the interaction. They include systems that automatically process input like language, gesture, or graphics and that render multimodal output. According to [Bohnenberger, 2004], some intelligent user interfaces use for example decision-theoretic planning to perform automatic content selection, media allocation, media realization and layout. There are systems that can automatically design graphics from structured data or coordinate the layout of multimedia content in time and space. Intelligent user interfaces often come as anthropomorphic agents, such as assistants or mediators.

Situated Interaction belongs to the research area of *Human-Computer Interaction (HCI)*. The extended concept of *implicit Human-Computer Interaction (iHCI)* is defined in [Schmidt, 2003]:

> **Definition 2.7 (implicit Human-Computer Interaction)** *implicit human-computer interaction is defined as the interaction of a human with the environment and with artefacts which is aimed to accomplish a goal. Within this process the system acquires implicit inputs from the user and may present implicit output to the user.*

[5]The naturalness of human-machine interaction however leads through the transparency within ubiquitous computing to a situation where the “machine” completely disappears. Thus one might rather talk about “human-environment” interaction instead of “human-machine” interaction, see [Wasinger and Wahlster, 2005].

Implicit Input are actions and behavior of humans, which are done to achieve a goal and are not primarily regarded as interaction with a computer, but captured, recognized and interpreted by a computer system as input. *Implicit Output* of a computer is not directly related to an explicit input and is seamlessly integrated with the environment and the task of the user.

As last concept, before we present a diagrammatic definition for situated interaction, we discuss the *situation concept*. In [Kray, 2003] it is pointed out that throughout the different research communities and disciplines, there are various definitions of what exactly is contained in the *context model* [McCarthy and Buvac, 1998], the *user model* [Dey and Abowd, 1999], and the *situation model* [Jameson, 2001b]. Therefore, it is necessary to clarify how those terms will be used in our approach. A situation consists of two parts on the one hand, there are user-related factors, which are intrinsically tied to a specific user, the abilities, goals, personal traits, etc. These factors are captured in a user model [Wahlster and Kobsa, 1989]. On the other hand, the user perceives and acts in a certain environment, which also has distinctive properties. In contrast to user-related factors, these factors are independent of an individual user as they equally affect all users in this location and they are determined by the environment. They define the context model. Even though, we distinguish the system in the partitioning of the world from the context, the system's factors are either modeled with the context or as resources, since the *resource model* from the area of resource-adaptive computing contains factors from the user as well as the system.

Together, these three models form the situation model as shown in figure 2.4. The term *situational factors* can be used as denotation for all these factors. Apart from the models within the situation concept, the system could also manage a task model, a discourse model and domain models as defined in [Maybury and Wahlster, 1998].

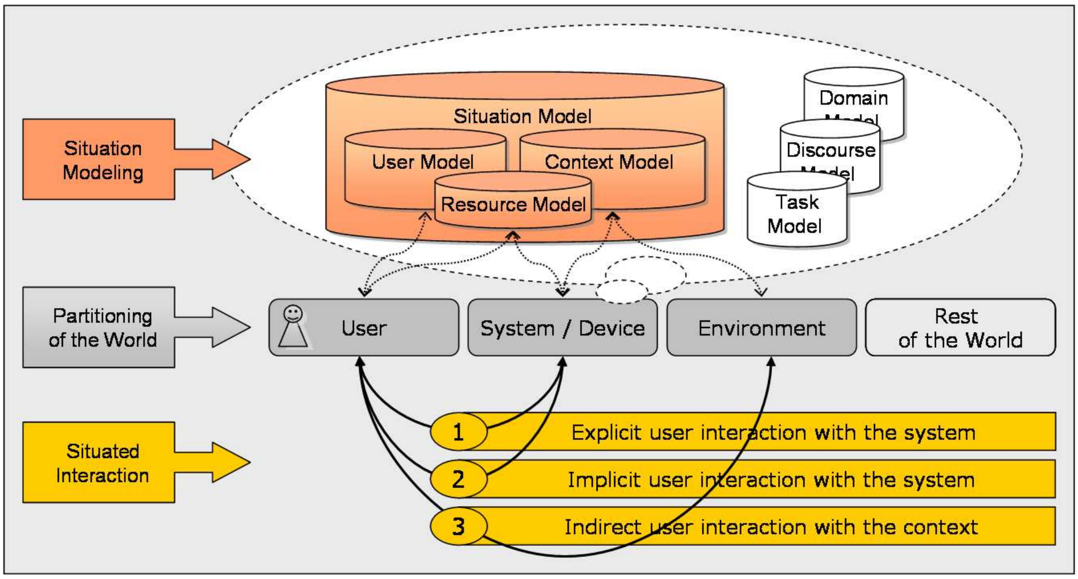


***Figure 2.4**: Situated interaction and the system's situation model for mobile computing*

Figure 2.4 shows a conceptual overview of situated interaction for mobile computing, where the world is logically partitioned into the four parts: *user*, *system/mobile device*, *environment* and *rest of the world*. It is interesting that in addition to the traditional explicit (1) human-computer interaction the implicit (2) user interaction with the system and the indirect (3) user interaction with the context are also modeled. The dimension of explicit and task related interaction versus implicit or indirect interaction is orthogonal to the dimension of mobility like *still* in human desktop computer interaction and *moving* in human mobile device interaction. This conceptual model is further described in [Heckmann, 2005b].

Figure 2.5 incorporates the application of the situation model and adds adaptation to the conceptual model of situated interaction: user-adaptivity, context-awareness and resource-adaptivity. Situated interaction within mobile computing leads to *Situation-Aware Human-Computer Interaction*.

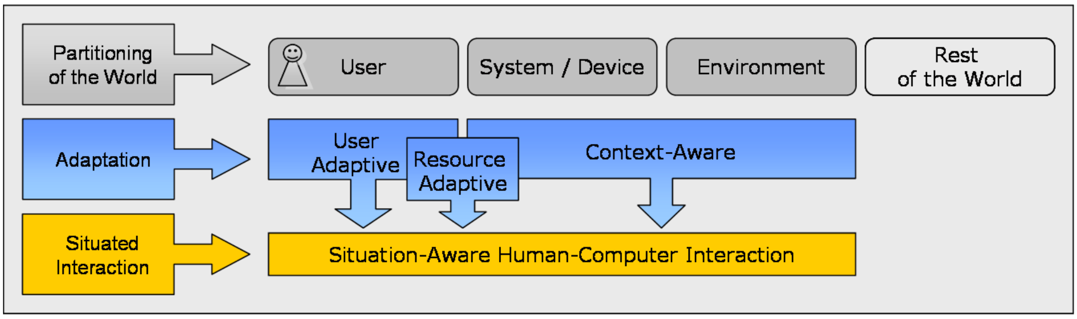


***Figure 2.5**: Situated interaction within mobile computing incorporates user-adaptivity, resource-adaptivity and context-awareness*

However, the final definition of *situated interaction* depends heavily on the underlying interaction model and of course on the parties that take part in the interaction. In the *desktop computing metaphor*, a user interacts locally with a computer. In the *mobile computing metaphor*, a mostly moving human interacts with a small portable device and has to cope with various contextual distractions. In the *ubiquitous computing metaphor*, a person or a group interacts with an invisibly instrumented environment, where the borders between system, device and context are gone. Furthermore the so-called *ubiquitous-awareness* extends the notion of situation-awareness by additionally taking into account communication with neighboring systems and the use of situations in the spatial and temporal history. In ubiquitous computing there is apart from the *user intended task* like "shopping for a meal", the *system's intended task* like "helping with the recipe" also the *environment intended task* like for example "optimize the food shop in general". Situated interaction within ubiquitous computing leads to *Situation-Aware Human-Environment Interaction*.

New in the ubiquitous computing paradigm is that there exist a different view to the partitioning of the world. Compare the first row of figure 2.6 and figure 2.5. The user interacts with the *intelligent environment* as a whole, which covers the context and the various systems, rather than interacting with the systems within their context. This model suits best Albert Einstein's aphorism about the environment: *The environment is everything*

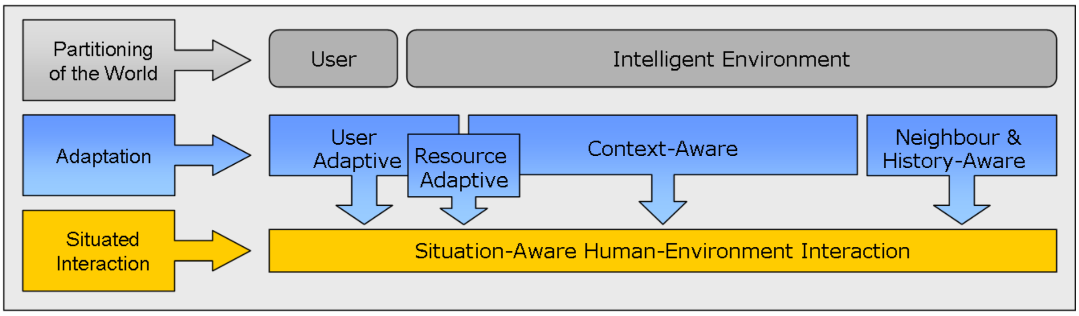


***Figure 2.6**: Situated interaction within ubiquitous computing also incorporates neighbor and history awareness*

that isn't me. Even parts of the users' cloths become - by *wearable computing* - part of the intelligent instrumented environment. Furthermore, the topic of multi-user modeling and interaction, see e.g. [Vassileva et al., 2003], versus individual human interaction will become more prominent in ubiquitous computing. Since multi-user interaction is not yet integrated into this conceptual interaction model, we conclude this subsection with the answer to the question *What is situated interaction?* with the following preliminary definition:

> **Definition 2.8 (Situated Interaction)** `Situated interaction` *denotes the explicit, implicit and indirect interaction between a human and an intelligent environment (or systems and their interaction devices), that take situation-awareness into account.*

> **Definition 2.9 (Situation-Awareness)** `Situation-awareness` *denotes the combination of user-adaptivity, resource-adaptivity, context-awareness, neighbor-awareness and history-awareness.*

Neighbor-awareness and *history-awareness* are informal denotations for systems that take their neighboring systems with their abilities and services into account, as well as their own interaction history. A formal definition of these two concepts is regarded as future work.

2.2.5 Real and Virtual versus Original and Reference

Closely related to *ubiquitous computing* (UC) are the research area of *augmented reality* (AR), *mixed reality* (MR) and *ambient intelligence* (AmI), see [Wikipedia, 2005]. Although the technologies differ, they are united in a common philosophy: the primacy of the physical world and the construction of appropriate tools that enhance our daily activities, see [Wellner et al., 1993]. A further late-braking research area in human-computer interaction is *virtual reality* (VR). Virtual reality allows us to use our whole bodies and a rich variety of virtual objects to interact with the computer. It attempts to replace the physical world with a computer-generated one, using devices such as head-mounted display goggles and data gloves. However it cuts us off and excludes us from the (real) world in which we live, work and play. In [Mackay, 1998] the real and virtual worlds are linked by the new paradigm of augmented reality which is defined in [Azuma, 1997] as an environment that includes both virtual reality and real-world elements.

The underlying question in this short subsection is the concrete problem of finding unique identification for objects, concepts and events in real worlds, augmented worlds, ubiquitous worlds (intelligent environments) and virtual worlds, see [Mattern, 2002b]. In chapter 5 we will introduce the blocks world "UbisWorld" in order to model and simulate ubiquitous computing applications. Since we need references to real worlds and virtual worlds at the same time, a clarifying model is needed. Several projects introduce virtual counterparts of real world objects. According to [Mattern, 2002a], many objects will have their own internet home page, or even better, their own internet portals. An interesting, recent example of mixed reality is the so-called "product associated displays" that has been described in [Spassova et al., 2005].

In *embedded computing* and *ambient intelligence*, small processors and sensors will allow us to enrich ordinary objects (like chairs, tables or flowers) with information processing capabilities. See [Aarts and Marzano, 2003] and [Aarts et al., 2003] for an overview of ambient intelligence. Furthermore, it is already sufficient to uniquely identify everyday objects and map them to their virtual counterparts which do the information processing part for the original object. The identification is nowadays mostly done with bar codes and bar code readers or RFID tags and RFID tag readers. Once we reached this insight that ordinary objects (or their references) will become smart, it is obvious, that smart objects will cooperate with each other and with the users. The concepts behind the following terms need to be clarified: *real, virtual, original, referential, physical, augmented, digital, copy*. For example, a real-physical user will be mapped in UbisWorld to a virtual-digital user. A real-physical device will be mapped to a virtual-physical device. However, the other way round is also possible: if you receive a digital email, it could also be modeled with a virtual letter-icon in the referential world. So far we have seen `physical->digital` mapping and a `digital->digital` one. The following example presents a rare `digital->physical` mapping in a ubiquitous computing application. It is interesting, since it changes the point of view: imagine a system, that for each email, that arrives in your email folder a real physical marble[6] falls into a box. Thus a real-world physical object, the marble, references the original digital-world email. A possible application is now that the physical sorting of the marbles controls the sorting of the emails. Such interfaces are called *tangible interfaces*, see [Ullmer and Ishii, 2001]. A `physical->physical` mapping is for example given, where soldiers, tanks and hills in the real battle field are modeled by small physical pieces on a strategy board. Figure 2.7 presents the first attempt to clarify this complex situation.

This model is not complete, since it neglects the *real* versus *virtual* dimension, on purpose. "What is real?" and "What is virtual?" in augmented reality and virtual reality, where ev-

[6]The `Marble Answering Machine` by Durrell Bishop: Incoming voice messages are physically instantiated as marbles. The user can grasp the message (marble) and drop it into an indentation in the machine to play the message. The user can also place the marble onto an augmented telephone, thus dialing the caller automatically.

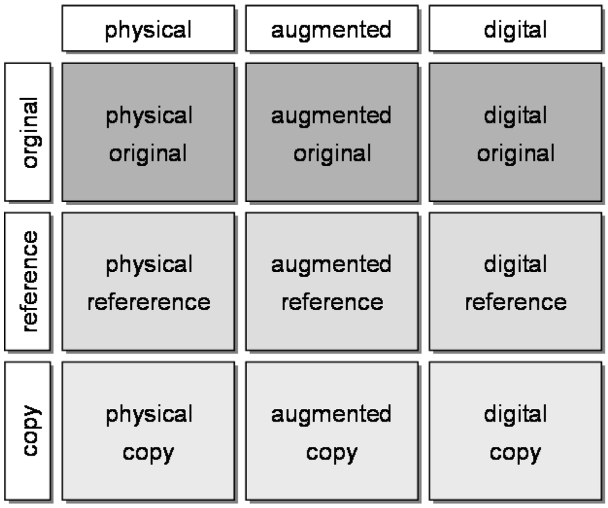


Figure 2.7: *Original-reference-copy versus physical-augmented-digital*

eryday objects speak with each other? In most cases "real" collapses with "physical" and "virtual" collapses with "digital" but not always, as pointed out above. We also neglected the possibly reflexive mental images. However, the suggested model allows to categorize and distinguish between different instantiations of objects in order to provide a powerful modeling language for UbisWorld.

2.3 Introduction to the Semantic Web

The initial idea behind the *semantic web* is to annotate documents in the world wide web with semantic information by the use of ontologies, see [Fensel et al., 2003]. Current search engines are limited to a syntactical analysis of the content of their indexed pages. Since it is hard for a machine to understand the meaning of graphical and textual information found on a website, languages like RDF have been designed to allow the author to declare the page contents as resources. Statements can be included in order to describe the semantics and relationships between resources using an ontology. By doing so, the web may be used in the sense of a very large distributed knowledge base. This section heavily relies on documents of the W3Consortium[7]. Especially the documents [Manola and Miller, 2004] and [McGuinness and van Harmelen, 2003], were taken into account. The goal in mind is the development of a machine-processable and human readable language, see [Studer et al., 2003], for representing and exchanging knowledge about a specialized domain. Ontologies as well as the semantic web form a fundamental basis in our approach. For example in [Stahl and Heckmann, 2004a] we show how to use the semantic web technology for ubiquitous hybrid location modeling.

[7]W3 Consortium homepage: `http://www.w3.org`

2.3.1 The Resource Concept and Uniform Resource Identifiers

The existing web architecture provides an elegant facility to identify any resource. A *resource* is defined in the glossary [Candela, 2003] as *a network data object or service that can be identified by a URI. Resources may be available in multiple representations (e.g. multiple languages, data formats, size, resolutions) or vary in other ways.* Originally, a resource was introduced as a piece of information that could be linked across the web. In HTML, a web page can be linked across the web by so-called *Uniform Resource Locators (URLs)*. URLs are used by browsers to retrieve the web pages itself. However, a generalized *resource* can be anything we want to talk about, including for example human beings, flowers, countries, also web pages or even abstract concepts that don't physically exist, like the concept of `friendship` or the concept of `time pressure`. A generalized resource does not need the feature to be retrievable in the Web. Thus, *resource* can be seen as a synonym for *entity*. In the terminology of ontologies, a resource can be a *class* like `animal` or `vehicle`, a *property* like `has-color`, but also an *individual* like `Peter`. The problem of *how to name and identify resources uniquely?* is described below. In order to clarify the mismatching concepts of the term resource[8], the following two definitions are used throughout this thesis:

Definition 2.10 (Resource) *A `resource` can be anything that has identity.*

Definition 2.11 (Web Resource) *A `web resource` can be anything that can be retrieved from the web.*

Uniform Resource Identifiers (URI) provide a simple and extensible means for identifying a resource. According to [Manola and Miller, 2004], they have primarily been developed in the W3C approach for naming or identifying resources on the web. An example for a URI is `http://www.w3.org/Protocols/rfc2616/rfc2616.html`. It is an identifier for the protocol document *Hypertext Transfer Protocol – HTTP/1.1.* URIs have been known by many names: *WWW addresses*, *Universal Document Identifiers*, *Universal Resource Identifiers* , and finally the combination of *Uniform Resource Locators (URL) and Uniform Resource Names (URN).* In [Berners-Lee et al., 1998] the syntax of URIs is defined.

Definition 2.12 (Identifier) *An `identifier` is an object that can act as a reference to something that has identity.*

Definition 2.13 (Uniform Resource Identifier) *A `uniform resource identifier` is an identifier, where the object is a string with the restricted syntax, defined by [Berners-Lee et al., 1998].*

In other words uniform resource identifiers are simply formatted strings which identify a resource. All URIs share the property that different persons or organizations can independently create them, and use them to identify things which reduces at the same time the possibility of confusion with a similar-looking identifier that might be used by someone else on the web. Some URI schemes depend on centralized systems, others are completely decentralized. This means that one doesn't need special authority or permission to create a URI for something.

[8]The term *resource* in the semantic web meaning and not as in resource-awareness of section 2.1.3.

2.3.2 RDF Resource Description Framework

The *resource description framework* is a language for representing information about resources in the World Wide Web. It is particularly intended for representing meta data about web resources, such as the title, author and modification date of a web page, copyright and licensing information about a web document, or the availability schedule for some shared resource. However, by generalizing the concept of a *web resource* as defined above, RDF can also be used to represent information about things that can be identified on the web, even when they cannot be directly retrieved on the web. Since the approach of situational statements is based on RDF, a profound investigation of the resource description framework has been undertaken. A detailed introduction can be found in the so-called “RDF-Primer” document, see [Manola and Miller, 2004].

In [Stuckenschmidt and van Harmelen, 2005] it is pointed out that, if we wanted to describe information on the meta-level and to define its meaning, we have to look for further approaches than XML and the RDF standard has been proposed as a data model for representing meta data. See [Brickley and Guha, 1999], [Lassila and Swick, 1999] for early papers on RDF. Nonetheless, this abstract data model finds a concrete syntax representation in XML. The corresponding schema language to define the vocabularies is called RDF Schema, or RDFS, see e.g. [Manola and Miller, 2004]. In RDFS one can define which properties apply to which kinds of objects and what value ranges hold. Furthermore one can describe the relationships between objects. In section 4.1.3, a model of the basic RDF triple is presented and extended to the model of situational statements.

2.3.3 OWL Web Ontology Language

The prominent web ontology language OWL has more facilities for expressing semantics compared to XML, RDF, and RDFS , see [McGuinness and van Harmelen, 2003] for an overview. Furthermore, it has a greater machine interpretability, according to [Stuckenschmidt and van Harmelen, 2005]. OWL adds more vocabulary for describing properties and classes and among others, relations between classes (e.g. disjointness), cardinality (e.g. “exactly one”), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. Figure 7.6 on page 146 shows a diagrammatic model of the OWL vocabulary. OWL can especially be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. OWL is a revision of the DAML+OIL web ontology language, in which the first user model ontology was defined, see [Heckmann, 2003d]. [Chen et al., 2003] and [Chen and Kotz, 2002] combine semantic web and OWL with ubiquitous computing. [Eberhart, 2003] defines an ontology based infrastructure for intelligent applications. The general user model ontology GUMO and the UbisWorld ontology are defined as OWL applications, see chapter 5.

2.3.4 Web Services and Semantic Web Services

The concept of *web services*[9] provides a new solution to solve the integration problem among heterogeneous application systems. According to [Kim et al., 2004] the web services

[9] W3C Web Services homepage: `http://www.w3.org/2002/ws/`

concept can be seen as a kind of standardized software technology that can integrate and share various systems. This web services concept has an advantage of flexibility by perfectly defining standard specifications for mutually sharable data among distributed systems. So the web services provide the advantage that they can transparently access any web servers in any place with any device and at any time. The web services architecture combines three essential roles: *service provider*, *service registry* and *service requestor*, see figure 2.8.

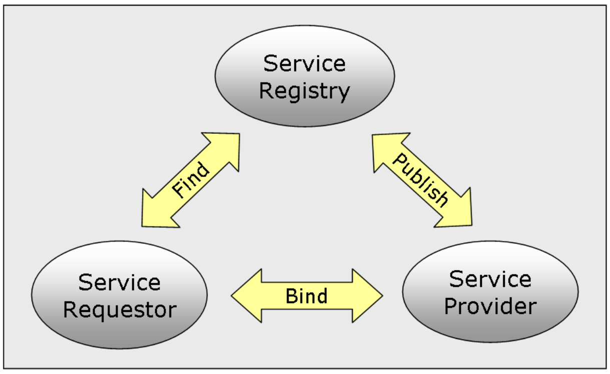


Figure 2.8: *Publish, find and bind: the three basic processes in the web service framework*

The service provider publishes the availability of their resources using WSDL (Web Services Description Language) that defines the usage of web services and is used in order to describe the interface name, argument and return value of programs. The service registry is acting as a blackboard of services using UDDI (Universal Description Discovery and Integration), with the purpose of building a distributed global registry that could be accessed through the web environment. The web services standard architecture is composed of XML, UDDI, WSDL and SOAP (Simple Object Access Protocol), a protocol that enables users to mutually communicate their services under distributed environment with the well established XML. The developed `u2m.org` user model service (as described in chapter 8.1) operates as service provider.

The idea behind *Semantic web services*[10] is that the semantic web should also enable greater access to services on the web. In [Martin et al., 2004] it is argued that semantic web services are developing the means by which services can be given richer semantic specifications. This richer semantics can enable fuller, more flexible automation of service provision and use, and support the construction of more powerful tools and methodologies. The semantic web should enable users and computer systems to discover, invoke, compose, and monitor all web resources that offer particular services. The DARPA Agent Markup Language program (DAML)[11] is about to develop and refine a service ontology that is called OWL-S. This OWL-based web service ontology is described in its release 1.1 at [OWL-S Coalition, 2005].

UDDI is widely used by businesses to register their presence on the Web by specifying

[10] Semantic Web Services Interest Group: `http://www.w3.org/2002/ws/swsig`
[11] DAML Semantic Web Services homepage: `http://www.daml.org/services/owl-s/`

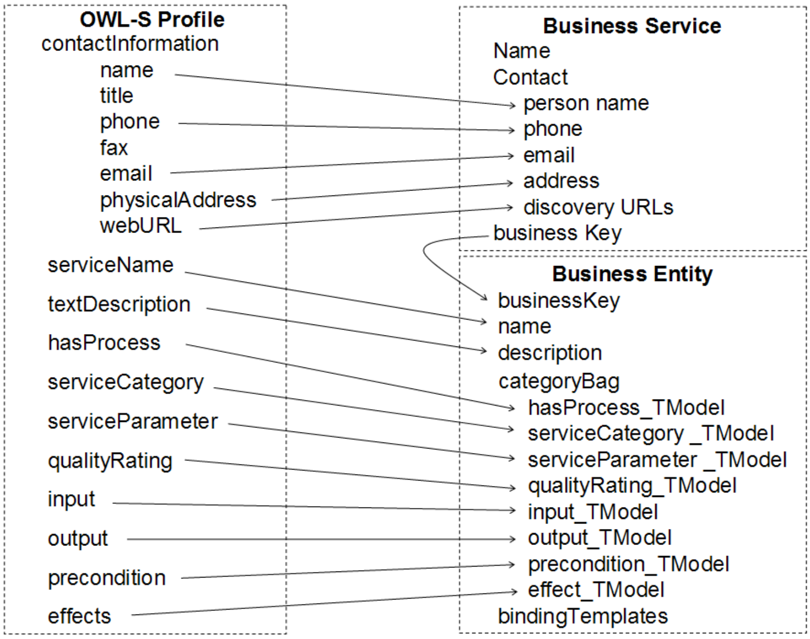


Figure 2.9*: The OWL-S to UDDI mapping according to [Martin et al., 2004]*

their points of contact both in terms of the ports used by the service to process requests and in terms of the physical contacts with people that can answer questions about the service. Figure 2.9 shows the mapping from OWL-S to UDDI and thus integrates both approaches.

Figure 2.10 shows the top level of the semantic web service ontology: a `Resource` provides a `Service`. This service presents the `ServiceProfile`, that defines what the services does. This service is described by the `ServiceModel`, which defines how the service works. And finally the service supports the `ServiceGrounding`, that shows how to access it.

2.3.5 Semantic Grid

Another late-braking, interesting research issue is the so-called *Semantic-Grid*[12]. It can be seen as the application of semantic web technologies to Grid computing[13], see for example [Foster et al., 2001] for an overview. The semantic grid can be defined as *an extension of the current Grid in which information and services are given well-defined meaning, better enabling computers and people to work in cooperation*, (Wikipedia). The vision points towards a generically useable infrastructure, which is comprised of easily deployed components with flexible collaborations and computations on a global scale. The key to this is seen in an infrastructure where all resources, including services, are adequately described in a

[12] Semanitc Grid homepage: `http://www.semanticgrid.org/`

[13] Grid Computing uses the resources of many separate computers connected by a network (usually the internet) to solve large-scale computation problems (Wikipedia).

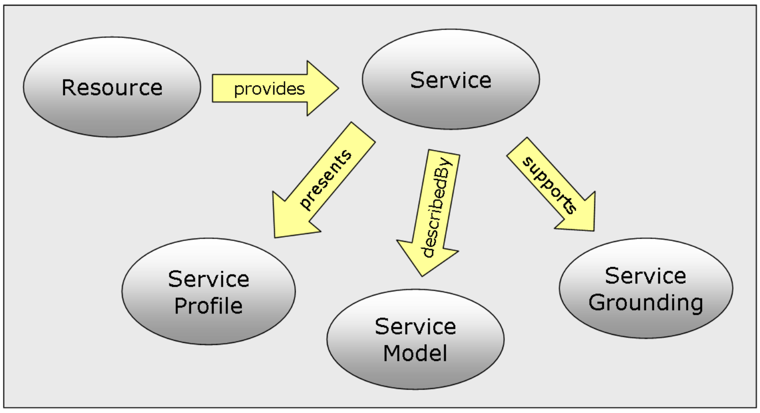


Figure 2.10: *The top level ontology of Semantic Web Services in OWL-S*

form that is machine-processable, to say, the goal is semantic interoperability. According to [Zhuge, 2005], the semantic grid is an internet centered interconnection environment that can effectively organize, share, cluster, fuse, and manage globally distributed resources based on the interconnection semantics.

The semantic grid approach promises to provide effective middleware technology for ubiquitous user modeling. However, because of its early state of development, the investigation and the integration are postponed to future work.

3 RELATED WORK & STATE OF THE ART

3.1 Generic User Modeling Systems

According to [Kobsa, 2001a], a user modeling system is called a *generic user modeling system* if it is independent of the specific user adaptive application with respect to its architecture and the user model contents. It is called a *user modeling server* if it operates as a server of information. In this section four different generic user modeling systems are described. However user modeling (shell) systems that are tightly interlaced with the application system with its functionality exclusively geared to the demands of one application are not discussed here. An overview of such systems can be found in [Kobsa, 2001c], [Fink and Kobsa, 2000] or [Jameson, 2001c].

3.1.1 Doppelgänger

DOPPELGÄNGER is a generic user modeling system that gathers data about users, performs inferences upon the data and makes the resulting information available to applications, see [Orwant, 1994]. The main implemented application in the DOPPELGÄNGER project was to create newspapers that were customized on the basis of the user's `news preferences` and the user's `news interests`, see [Orwant, 1996]. The computations take place at spatially distributed locations and make use of portable user models that are carried by the users. The focus is set on heterogeneous learning techniques that were developed for an application-independent, sensor-independent environment. The user models were stored in a centralized database in LISP-like lists, either on fixed hard disks or on PCMCIA cards, to have the possibility to remove them for privacy reasons physically from the the server. Communication between the user and the server occurs by the *"pleasant path of e-mail"*. An important and motivating statement in [Orwant, 1995] is the following one:

> *Our computers support many different applications, each of which does one thing well ... each application can personalize its behavior for users, but when these applications can make use of a common database of information about the user, and communicate with one another about the user, their ability to personalize themselves increases dramatically.*

Even though this statement does not completely suggest the whole idea of *Ubiquitous User Modeling* (according to the definition in section 1.3) it already motivates the communication between different user-adaptive applications (within one computer). Furthermore, DOPPELGÄNGER already entered the world of ubiquitous computing since it tracked the user's location on the basis of *active badges* and *smart chairs*: an infrared sensor at this smart chair notified the user's workstation, when he was sitting in front of it. The centralization of DOPPELGÄNGER's architecture was good for constructing a common store of personal information between applications, but didn't scale well to modeling several people. In [Orwant, 1995] it is already concluded, that there is a need for distributed servers, the so-called DOPPELGÄNGERS.

3.1.2 UM Toolkit

The UM toolkit is an early user modeling server, see [Kay, 1995]. UM's representation of user models was strongly influenced by the goal of making the user model itself accessible. To achieve this, it includes explanations and justifications for elements of the user model and the modeling process. It uses the *accretion* representation and architecture which is a simple but very flexible approach to represent user models. It consists of so-called *components* each of which models an arbitrary aspect of the user. It distinguishes `knowledge`, `beliefs`, `preferences` and other attributes of the user including their `personal attributes` like `names`, `date of birth` and arbitrary aspects like their `current location`. The accretion representation makes no built-in assumptions for any of these types. The essential idea according to [Kay et al., 2003] is that each piece of evidence about a component is simply added to a list, thus over time, it *accretes*[1]. Sources are only permitted to contribute to the model if they are authorized to do so[2]. The generic user model simply holds the uninterpreted collection of evidence for each component. It is only at runtime that the application uses a resolver to interpret the evidence available to it and conclude the value of that component. UM's models and tools are also intended to be used by a range of systems that act as *consumers* for the user model and the representation was flexible enough to be useful to different UM-consumers. To summarize, the major technical goal of the UM work has been the creation of a toolkit, composed of simple elements that can be combined to do interesting user modeling tasks.

3.1.3 Personis

The goal of the PERSONIS project, that is based on the *UM toolkit*, is to explore ways to support powerful and flexible user modeling and - at the same time - to design it to be able to support user scrutiny and control, see [Kay et al., 2002]. It is novel in its design being explicitly focussed on user control and scrutability. Figure 3.1 shows the high level architecture of several Personis-based adaptive hypertext systems. A Personis server can support the reuse of user models over a series of adaptive hypertext systems. The so-called *views* are the conceptual high-level elements that are shared between the server and each application.

[1]The *accretion* representation idea influenced the design of SITUATIONREPORTS.

[2]We extend this position and allow every source to contribute, but store metadata about the `creator`.

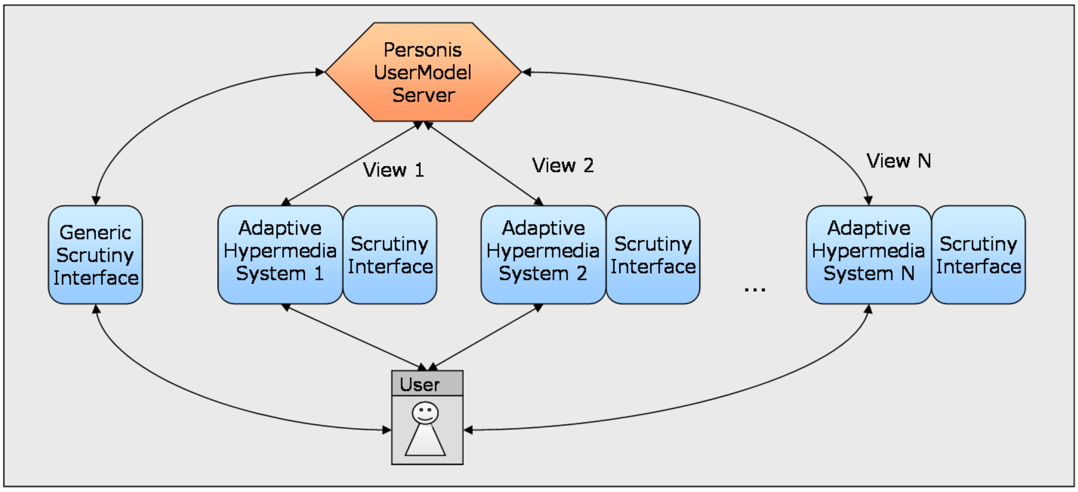


Figure 3.1: *Personis server architecture according to [Kay et al., 2002]*

The name stems from the aspect, that user model consumers might need just a few components of the user model, and a database view is applied to the whole model. The underlying representation of the user model collects *evidence* for each *component* of the user model[3]. The user model information in Personis is held together with the access control information as a relational object database. Several user interfaces have been implemented. The Personis *resolvers* follow the tradition of the UM toolkit and interpret the evidences at runtime. Furthermore, in [Kay, 1995] it has been demonstrated that many users can and do scrutinize their user models.

3.1.4 Deep Map User Modeling System

In [Fink and Kobsa, 2002] the user modeling system of DEEPMAP is described as the state of the art user modeling system for personalized city tours. This work was carried out in the context of the Deep Map project [Malaka and Zipf, 2000] of the European Laboratory in Heidelberg, Germany. The DEEPMAP user modeling server is aimed to provide information in a personalized manner in the travel and tourism domain. Especially the users' `interests` and `preferences` are taken into account. The all-in-one system offers services to personalized systems with regard to the analysis of user actions, the representation of the assumptions about the user, as well as the inference of additional assumptions, based on domain knowledge and characteristics of similar users. The system is open and can be accessed by clients that need personalization services via different interfaces. That is how the DEEPMAP user modeling system stands in tradition with [Fink and Kobsa, 2000]'s generic

[3]We use and extend this idea in our conflict resolution part, where we also collect *evidence* for each extended *component*, or in our words: we also collect SITUATIONALSTATEMENTS of *user model dimensions* with their semantic entailments P^*, see section 6.3.

architecture for the user modeling shell system BGP-MS[4] which proposes an object-oriented approach to realize a network-oriented user modeling shell system and which allows a distributed use of a central user model. In [Fink, 2004] the pros and cons of *directories* and *databases* are evaluated while LDAP[5]-based systems are recommended as a basis for user model servers. Such directories are specialized database management systems that maintain information about relevant characteristics of users, devices and services on a network. One advantage[6] are for example predefined user-related information types. A second advantage[7] of LDAP directories it that they can manage information that is dispersed across a network of several servers.

The DEEP MAP user modeling system provides three user modeling components: The *User Learning Component*, which learns user interests and preferences from usage data, and updates individual user models. The *Mentor Learning Component* that predicts missing values in individual user models from models of similar users, and the *Domain Inference Component* that infers interests and preferences in individual user models by applying domain inferences to assumptions that were explicitly provided by users or the two other components. The user modeling server is designed on top of an existing LDAP server as an application that is built of loosely cooperating autonomous components. Figure 3.2 shows the general user modeling server according to [Fink, 2004]. In the center, the *Directory Component* consists of the three sub-systems *Communication*, that is responsible for managing communication with external and internal clients, *Representation*, that is in charge of managing directory content and *Scheduler*, that has to mediate between the different sub-systems and components of the user modeling server. On the right, several *User Modeling Components* are shown that perform dedicated user modeling tasks like collaborative filtering. The components can be flexibly distributed across a network of computers[8]. The directory component communicates with the user model components, the clients and the models via *CORBA* and *LDAP*. On the left hand side of figure 3.2, several models are shown that constitute the representational basis of the DEEPMAP user modeling service, namely the *User Model*, *Usage Model*, *System Model*, and *Service Model*. Above, a few user model clients or consumers are presented.

DEEPMAP is part of a family of long-term research projects. One central aim of the project [Kray, 2003] was the provision of personalized tour recommendations for the city of Heidelberg that cater to an individual user's interests and preferences. The WEBGUIDE sub-project identified geographical points of interest and computed tours that connect these points via presumably interesting routes, based on geographical information, information about selected means of transport, the user model, and by the user specified tour restrictions like distance and duration. Finally such tour recommendations were presented to the user.

[4]BGP-MS was influence by the SFB 314 project XTRA (A Natural-Language Access System to Expert Systems), Department of Computer Science, Saarland University, see for example [Wahlster and Kobsa, 1986], [Kobsa, 1990], and [Allgayer et al., 1989]

[5]LDAP = Lightweight Directory Access Protocol

[6]We have chosen a different approach, since we consequently separate the syntactical representation from the semantics: we define the general user model ontology GUMO in a semantic web language.

[7]We have also chosen a different approach here, since we support the emerging web-based services and standards that perfectly solve the distributed network problem.

[8]We adopt this idea for our approach of ubiquitous user modeling.

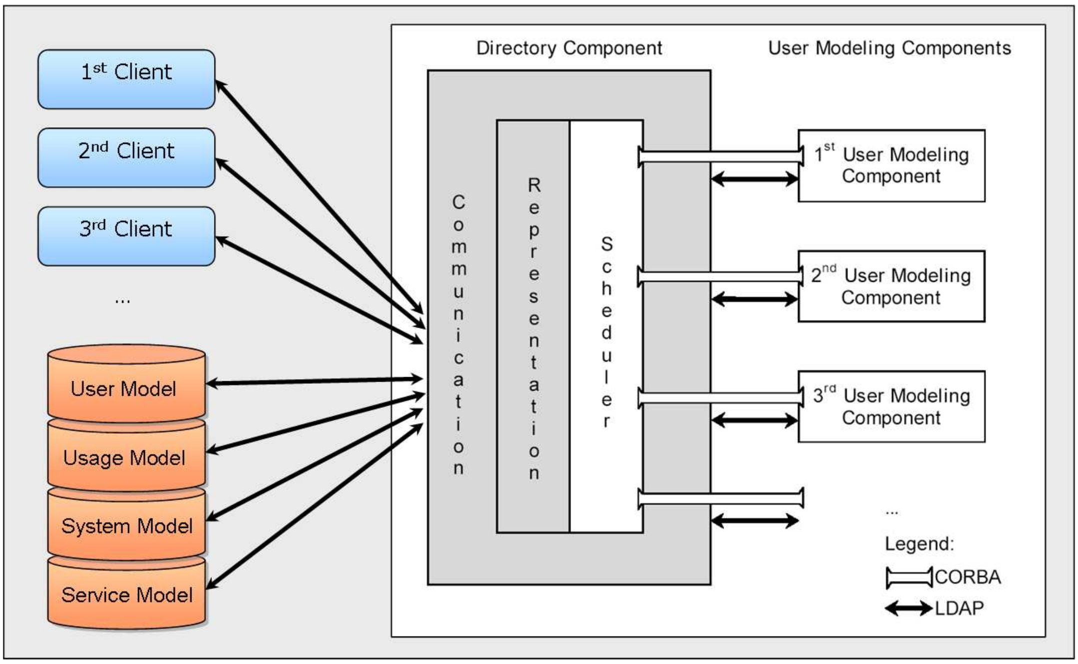


Figure 3.2: *User modeling server according to [Fink, 2004]*

3.2 Metadata Applications

In this section we present the *Dublin Core Metadata Initiative*, the *Customer Profile Exchange*, the *Human Markup Language* and the *Composite Capabilities/Preference Profile*. The discussion and comparison will follow in section 3.4.

3.2.1 Dublin Core Metadata

The Dublin Core[9], see [Manola and Miller, 2004], is a set of metadata elements for describing documents. The term *metadata* refers to data used to identify, describe or locate information resources. While structured metadata processed by computers is relatively new, the basic concept of metadata has been used for many years in helping manage and use large collections of information. The element set was originally developed at the Metadata Workshop in Dublin, Ohio in 1995. The Dublin Core has subsequently been modified on further workshops. The goal of the Dublin Core is to provide a minimal set of descriptive elements that facilitate the automated indexing and description of document-like network objects, in a manner similar to a library card catalog. In addition, the Dublin Core is meant to be sufficiently simple to be understood and used by the wide range of others and casual publishers who contribute information to the internet. Dublin Core elements are nowadays widely used in documenting internet resources. The Dublin Core Metadata Element Set contains definitions for the following properties:

[9]Dublin Core homepage: `http://dublincore.org/documents/dces/`

Title A name given to the resource

Creator An entity primarily responsible for making the content of the resource

Subject The topic of the content of the resource

Description An account of the content of the resource

Publisher An entity responsible for making the resource available

Contributor An entity responsible for making contributions to the content of the resource

Date A date associated with an event in the life cycle of the resource

Type The nature or genre of the content of the resource

Format The physical or digital manifestation of the resource

Identifier An unambiguous reference to the resource within a given context

Source A reference to a resource from which the present resource is derived

Language A language of the intellectual content of the resource

Relation A reference to a related resource

Coverage The extent or scope of the content of the resource

Rights Information about rights held in and over the resource

A detailed description can be found at `http://dublincore.org/documents/dces/`. The Dublin Core elements influenced the development of the `UserML/RDF` application. A descriptive Dublin Core example[10] is given in figure 3.3:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc ="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://dublincore.org/">
    <dc:title>Dublin Core Metadata Initiative - Home Page</dc:title>
    <dc:description>The Dublin Core Metadata Web Site</dc:description>
    <dc:date>2001-01-16</dc:date>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
    <dc:contributor>The Dublin Core Metadata Initiative</dc:contributor>
  </rdf:Description>
</rdf:RDF>
```

***Figure 3.3**: A Dublin Core example in RDF.*

The resource `dublincore.org` is described within the `rdf:Description` element, which is itself wrapped into an `rdf:RDF` element where the `dc` namespace is defined.

[10]Dublin Core example: `http://dublincore.org/documents/2002/07/31/dcmes-xml/`

3.2.2 Customer Profile Exchange

The CPExchange[11] specification derives from *Customer Relationship Management* (CRM) and is intended to support enterprise-wide architectures in which many applications use and update information relating to a customer. This specification differs according to [Bohrer and Holland, 2000] from enterprise frameworks such as the *Lightweight Directory Access Protocol* (LDAP) in the following ways:

- It provides a comprehensive view of the customer, not just as a user of a particular application, but also as an entity that interacts with multiple facets of an enterprise.
- It allows views of the customers activities over time, providing a cumulative historical record of events that enhances the enterprises understanding of the customer.
- It provides a granular privacy and authorization model that is optimized for aggregated and interchanged information as opposed to traditional monolithic storage models.
- It is designed to promote generation and collection of customer profile information. Useful query and reporting for enterprise customer relationship management is also supported.

Using CPExchange to describe profile information for people will lead to inter-operability between all applications in an enterprise that touch the same person (or organization), according to [Bohrer and Holland, 2000]. The specification is further outlined as:

The CPExchange specification can be used to carry profile information about people or organizations that have other relationships to an enterprise, such as a supplier or employee. The term customer[12] *can also refer to a person or organization regardless of their particular relationship to the enterprise. The ability to exchange profile information throughout and between enterprises leverages the collective knowledge of a customer to provide service, support, and new products that are tailored to a customers needs and wishes. It allows both customers and enterprises to be more efficient in their contacts with each other, whether by customization of self-service web sites or by personalization of more direct call center or e-mail contacts. Within an enterprise, no single application need use nor store all the information in a customer profile. Each application can aggregate the information through exchange of messages carrying profile data in the standard representation defined in this specification. Applications that support the CPExchange data model should be able to receive an XML-based CPExchange customer profile, map the elements into appropriate internal data representation and generate an XML-based CPExchange customer profile from their internal data representation.*

A generated customer profile does not need to match any CPExchange customer profile exactly that has previously been taken in. However, it is intended that a given application consistently map a given data item to the same CPExchange element.

[11] CPExchange homepage: `http://www.cpexchange.org/`

[12] We share the same goal in our approach. However, we do not restrict ourself to the commercial domain, where the *user* is mainly treated as *customer* or *employee*.

3.2.3 HumanML, the Human Markup Language

The Human Markup Language[13], with its primary abbreviation `HumanML`, is according to [Brooks and Cagle, 2002] an attempt to codify the characteristics that define human physical description, emotion, action, and culture through the mechanisms of XML and other appropriate schemas. The specification further describes that

> *`HumanML` is intended to provide a basic framework for a number of endeavors, including the creation of standardized profiling systems for various applications, building a framework for describing emotional state and response of both people and humanoid animations, laying the foundation for the interpretation of gestures for both person-to-person and person-to-computer interpretations. Personalization in the `HumanML` context is defined as Preferential Personalization, in which specific `emotional states`, `aesthetic preferences` and `gestural actions` are used to shape the interfaces between people and computer systems, and Habitual Personalization, in which `habits`, `interests`, and `activities` of people define both the interfaces used for computer interaction and the content presented to them. However, as people increasingly migrate between platforms - from multiple desktop environments to PDAs and web kiosk, to cell phones and embedded intelligent toasters - the difficulty of maintaining personalization profiles has to be solved in a distributed manner.*

The non-profit international consortium OASIS, that creates interoperable industry specifications based on XML, formed a technical committee to develop HumanML in 2001. They intend to develop HumanML as a specification for conveying human characteristics to contextually convey cultural, social, kinesics, and psychological intent within communications. Figure 3.4 shows for example the specification for `address` as a `complexType` in XML.

```
<xs:complexType name="Address">
    <xs:annotation>
        <xs:documentation>
            Address Type
            An address in a named address system, such as
            street, city, state, etc....
        </xs:documentation>
        <xs:appinfo>NONE</xs:appinfo>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="postal" type="xs:string"/>
        <xs:element name="residential" type="xs:string" minOccurs="0"/>
        <xs:element name="email" type="xs:string" minOccurs="0"/>
        <xs:element name="previous" type="xs:string" minOccurs="0"/>
        <xs:element name="current" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attributeGroup ref="humlIdentifierAtts"/>
</xs:complexType>
```

***Figure 3.4**: The HumanML specification for address, according to [Brooks, 2002]*

A discussion will follow in section 3.4 and section 4.1.

[13] Human Markup homepage: `http://www.humanmarkup.org/` (However, seems to be orphaned)

3.2.4 The CC/PP Framework

The W3C's *Composite Capabilities/Preference Profile* (CC/PP)[14] specification addresses the problem of describing device capabilities of mobile devices with highly divergent input, output and network connectivity capabilities as well as user preferences. The motivation for this specification can be found in the variety of new mobile devices for browsing the web. The CC/PP framework which has been developed since 1999 is supported by companies like IBM, Ericsson, Nokia and SAP AG. According to [Manola and Miller, 2004], the CC/PP framework defines the relatively simple structure of a two level hierarchy of components and attribute/value pairs. A so-called component may be used to capture a part of a delivery context, while each component may contain one or more attributes. For example a component that encodes user preferences may contain an attribute to specify whether or not the user prefers audio output versus visual output. The basic idea behind the CC/PP framework is content adaptation and thus adaptive hyper media.

The *client attribute names* defined below may be used to identify some common features associated with client devices that print or display visual information, such as text or graphics.

DeviceIdentifier A URI that serves as an identifier of the client device or user agent type

Type A MIME content type that can be accepted and presented by a client

Schema A URI that identifies a schema that is recognized by the client

CharWidth For a text display device, the width of the character display

CharHeight For a text display device, the number of lines of text that can be displayed

Charset For a text handling device, a character encoding that can be processed

Pix-x For an image display device, the number of horizontal pixels that can be displayed

Pix-y For an image display device, the number of vertical pixels that can be displayed

Color For text and image display devices, an indication of the color capabilities

One is encouraged to use this vocabulary[15] rather than define new terms. The relevant discussion will follow in the subsection 3.4.1.

3.3 External Ontologies and Knowledge Bases

The question that arises before any new ontology is about to be introduced is: *Do we need this ontology at all, since there already exist so many other ontologies?* In this section several existing ontologies and knowledge bases are presented; however, the reasons for the introduction of the UBISWORLD ONTOLOGY are also discussed.

[14] CC/PP homepage: `http://www.w3c.org/Mobile/CCPP/`
[15] CC/PP attribute vocabulary: `http://www.w3.org/TR/CCPP-struct-vocab/`

3.3.1 SUMO/MILO Ontologies

The SUMO ontology (Suggested Upper Merged Ontology)[16], see for example [Niles and Pease, 2001] or [Pease et al., 2002], was created as part of the *IEEE Standard Upper Ontology Working Group*. The goal of this working group is to develop a standard upper ontology that will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing. SUMO has been translated into various representation formats, but the language of development is a variant of *Knowledge Interchange Format* KIF, which is a version of the first-order predicate calculus, see [Genesereth, 1991]. An upper ontology is limited to concepts that are meta, generic, abstract or philosophical, and hence are general enough to address at a high level a broad range of domain areas. Concepts specific to particular domains are not included in an upper ontology, but such an ontology does provide a structure upon which ontologies for specific domains can be constructed. Figure 3.5 shows the top level concepts in SUMO.

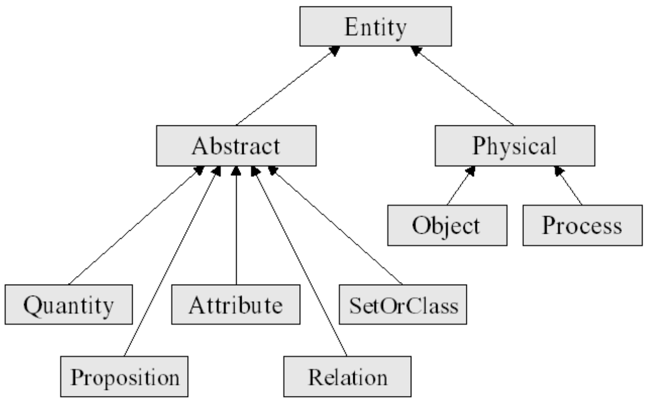


***Figure 3.5**: Top-level concepts in SUMO according to [Sevcenko, 2003]*

It uses the term `entity` as root which is well established in database research. The main distinction is drawn between `abstract` and `physical` entities. Physical entities are further distinguished as `objects` and `processes`, etc.

According to [Sevcenko, 2003], SUMO is a collection of approximately 1000 well-defined and well-documented concepts, interconnected into semantic network and accompanied by a number of axioms. The axioms mostly reflect common-sense notions that are generally recognized among the concepts. The concepts range from very general ones, such as `Quantity` as shown in figure 3.5, to very specific, such as `Emotional State` as shown (together with its superclasses) in figure 3.6(a) . Subclasses of a class are usually

[16] SUMO homepage: `http://ontology.teknowledge.com`

mutually exclusive, i.e. they do not share common instances. For example, nothing can be both an abstract entity and a physical entity, neither both an object entity and a process entity, which is explicitly specified in SUMO. However, some classes can have multiple superclasses. For example, a `Human` is both an `Organism` (which is itself a subclass of `Organic Object`) and a subclass of `Agent` and a `Cognitive Agent` which is defined as an entity with the ability to reason as shown in figure 3.6(b).

(a) (b)

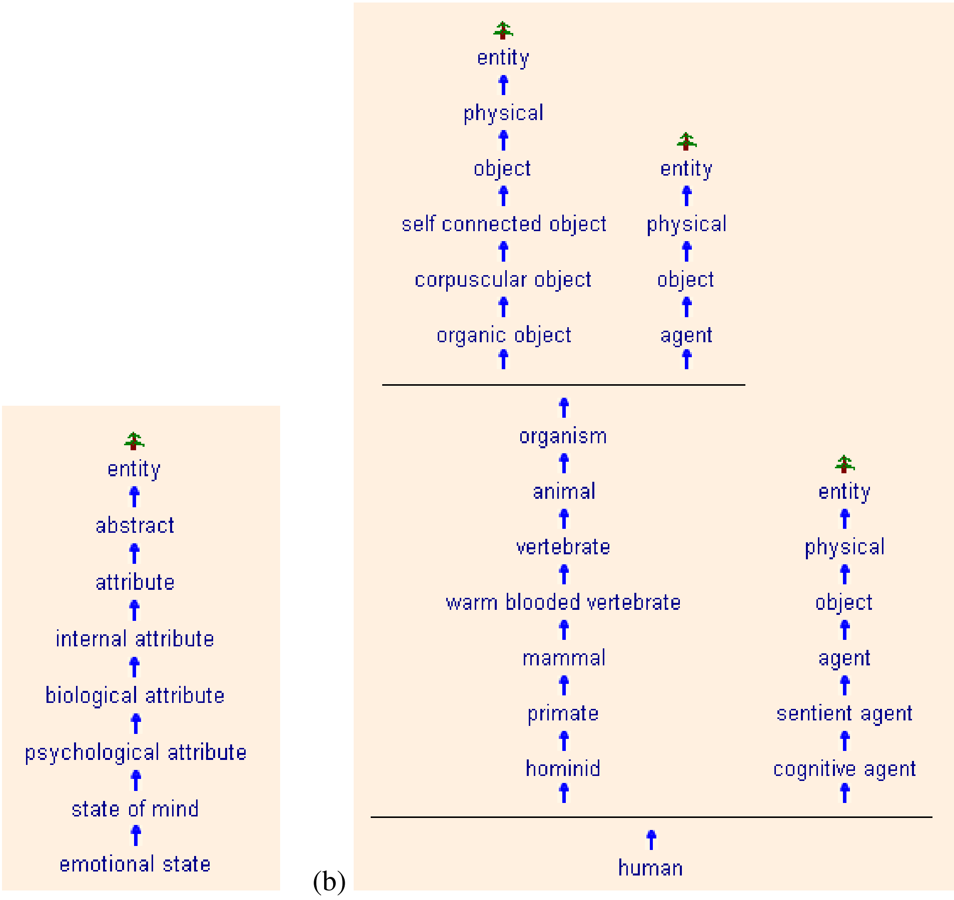


***Figure 3.6**: (a) Superclasses of `Emotional State` as shown in the SUMO browser and (b) multiple inherited superclasses of the `Human` concept.*

There exist a specialized SUMO Browser[17], an online tool for exposing the content of SUMO to the user, which is similar to the UBISONTOLOGYBROWSER which is described in section 8.2.3. The SUMO Browser also shows the connection between SUMO and the WordNet framework which is described in the next section. Multiple inheritance is realized by tree copying. SUMO is intended as a domain-independent knowledge base for designing

[17]SUMO Browser homepage: `http://virtual.cvut.cz/kifb/en/`

domain ontologies and is therefore very interesting for the design, integration and extension of UBISWORLD ONTOLOGY.

One of the drawbacks of SUMO, according to [Sevcenko, 2003], is its relatively low coverage that does not allow its employment for open-domain applications. However, the *Suggested Upper Merged Ontology*, together with all its domain ontologies like the *Mid-Level Ontology* (MILO) and ontologies of Communications, Countries and Regions, Economy, Finance, Geography, etc. form the largest formal public ontology in existence with approximately 20,000 terms today[18]. MILO is an ontology that has been developed as a bridge between the abstract content of the SUMO and the rich detail of the various domain ontologies. It is still provisional and under development. The combination of ontological concepts and natural language words has partially been established by connecting SUMO to the *WordNet* lexicon.

3.3.2 Cyc Knowledge Base

The Cyc knowledge base[19] is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life, see [Cycorp, 2004]. This *everyday life* character makes the Cyc knowledge base very interesting for the UbisWorld approach. However, the medium of representation is the formal language *CycL*. The knowledge base consists of terms and assertions which relate those terms. These assertions include both simple ground assertions and rules. The Cyc knowledge base can be seen as an ontology, even though it was originally not represented in an ontology language. The Cyc knowledge base is divided into many *microtheories*, each of which is essentially a bundle of assertions that share a common set of assumptions; some microtheories are focused on a particular domain of knowledge, a particular level of detail, a particular interval in time, etc. These microtheories could be compared with subontologies. The most general ones are `Things, Space, Time, Physical Objects, Agents, Living Things, Organization, Human Beings, Human Activities` plus domain-specific knowledge like `Bio-Warfare, Terrorism, Military Tactics, Command & Control`[20].

The microtheory mechanism allows Cyc to independently maintain assertions that could be contradictory, and enhances the performance of the Cyc system by focusing the inferencing process. The Cyc knowledge base currently contains more than two hundred thousand terms and several dozen hand-entered assertions about/involving each term. New assertions are continually added to the knowledge base by human knowledge engineers, while Cyc is already able to add assertions to the knowledge base by itself as a product of the inferencing process. The Cyc system also includes (like UbisWorld) a variety of interface tools that permit the user to browse, edit, and extend the knowledge base, to pose queries to the inference

[18] September 2005 at `http://www.ontologyportal.org/`
[19] Cyc homepage: `http://www.cyc.com`
[20] This military orientation originates in the DARPA influence

engine, and to interact with the natural-language and database integration modules.

However, the top-level hierarchy, which is shown in figure 3.7, is said to be too convoluted, see chapter 3.5.2 in [Lönneker, 2003] for a discussion. The main reason for the

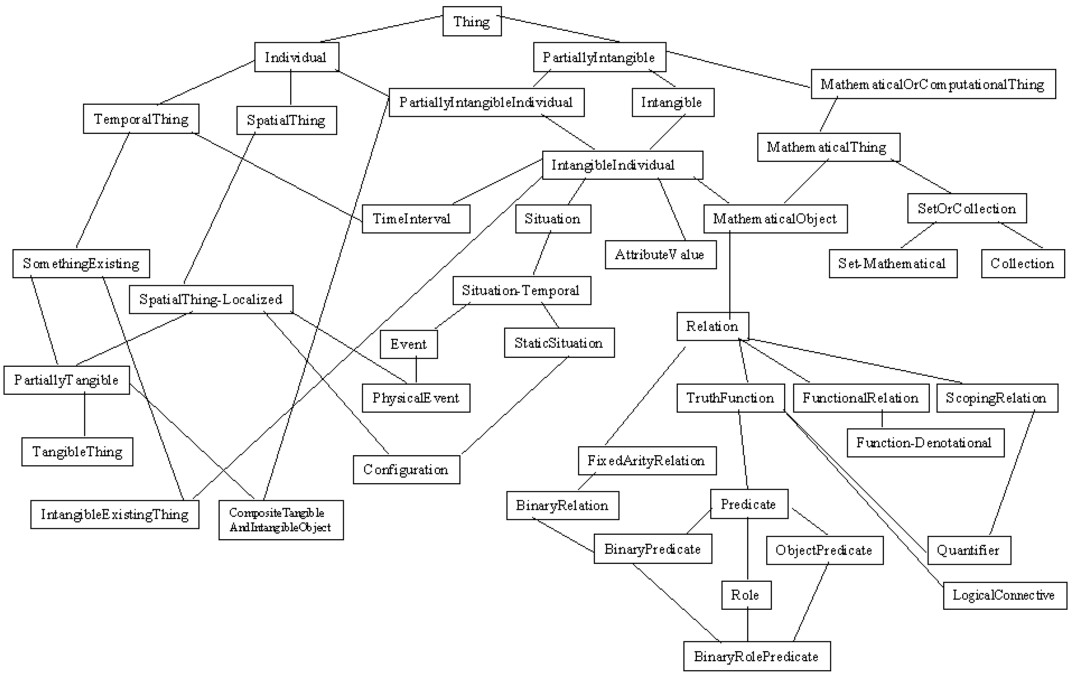


***Figure 3.7**: Top-level concepts in Cyc according to [Cycorp, 2004]*

decision against using the Cyc knowledge base as the main ontology structure for UbisWorld was that it is a commercial system[21]. Only a small part with approximately 3000 concepts is freely available in the OpenCyc[22] project.

3.3.3 WordNet Lexicon

An ontology is similar to a dictionary or glossary, but with greater detail and structure that enables computers to process its content. An ontology consists of a set of concepts, relations, and axioms that formalize a field of interest. However, WordNet[23] is a (freely available) online *lexicon* that has been developed for automated processing of natural language[24]. Technically, WordNet is an electronic thesaurus, defining a large set of word meanings, interlinked with semantic pointers. A word form can have many meanings which is called *polysemy* and many word forms can refer to the same meaning, which is called *synonymy*.

[21] Cycorp was incorporated in 1994 and has 60 full-time employees (in 2005). Its core technology is Cyc, which consists of a very large, commercial, context-faceted knowledge base of common sense knowledge

[22] See `www.opencyc.org` or `http://www.daml.org/ontologies/132`

[23] WordNet homepage: `http://www.cogsci.princeton.edu/~wn/`

[24] Linguists at the Princeton University have created WordNet as a result of their psycholinguistic research

Each word meaning entry which is also called *synonym set* and abbreviated as *synset*, is accompanied with a short informal definition which is called *gloss* and a list of word forms that can represent the synset in spoken or written language. See [Miller et al., 1993] for an introduction to WordNet and [Sevcenko, 2003] or chapter 3.5 in [Lönneker, 2003] for summaries. Other kinds of term relations that form networks of terms are according to [Stuckenschmidt and van Harmelen, 2005] for example *hypernyms*, terms with a broader meaning, *hyponyms*, terms with a narrower meaning, *holonyms*, terms that describe a whole the term is part of, and finally *mereonyms*, terms describing parts of the term.

WordNet is fully available in RDF/RDFS[25]; there exist databases of nouns, verbs, adjectives and adverbs, which originates in the separation of synsets for different part of speech. Semantic relations between synsets are different for different part of speech. For example, for nouns a chief relation between synsets is an is-a relation which is called *hypernymy* and *hyponymy*. At the first sight it seems that synsets in WordNet built up a large semantic network, as introduced in the knowledge representation in artificial intelligence. However, semantic relations in WordNet are sometimes very vague and non-logical, and cannot be used for logical inference. According to [Reed and Lenat, 2002], large portions of WordNet have been integrated into the Cyc knowledge base, and according to [Sevcenko, 2003], the WordNet conceptualizations have also been mapped to the SUMO ontology. Furthermore, apart from the Princeton WordNet approach, there exists a EuroWordNet[26] approach with multilingual databases with *wordnets* for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian) and a Global WordNet Association [27] which is a free, public and non-commercial organization that provides a platform for discussing, sharing and connecting wordnets for all languages in the world.

3.3.4 FrameNet Lexical Resource

The Berkeley *FrameNet*[28] project is creating an online lexical resource for English, based on frame semantics and supported by corpus evidence. See [Fillmore, 1985] for an introduction. The aim is to document the range of semantic and syntactic combinatory possibilities of each word in each of its senses, through manual annotation of example sentences and automatic capture and organization of the annotation results. The FrameNet database is in a platform-independent format, and can be displayed and queried via the web and other interfaces. A *frame* is an intuitive construct that makes it possible to formalize the links between semantics and syntax in the results of lexical analysis. Semantic frames are schematic representations of situations involving various participants, props, and other conceptual roles, each of which is a frame element (FE). The semantic arguments of a predicating word correspond to the frame elements of the frame(s) associated with that word. Each frame identifies a set of

[25]WordNet in RDF/RDFS available at: `http://www.semanticweb.org/library/`

[26]EuroWordNet homepage: `http://www.illc.uva.nl/EuroWordNet/`

[27]Global WordNet Association homepage: `http://www.globalwordnet.org/`

[28]FrameNet homepage: `http://www.icsi.berkeley.edu/~framenet/`

frame elements, which are frame-specific semantic roles like participants, props, phases of a state of affairs. Figure 3.8 shows an example frame of a `Criminal Process`[29] together with its subframes.

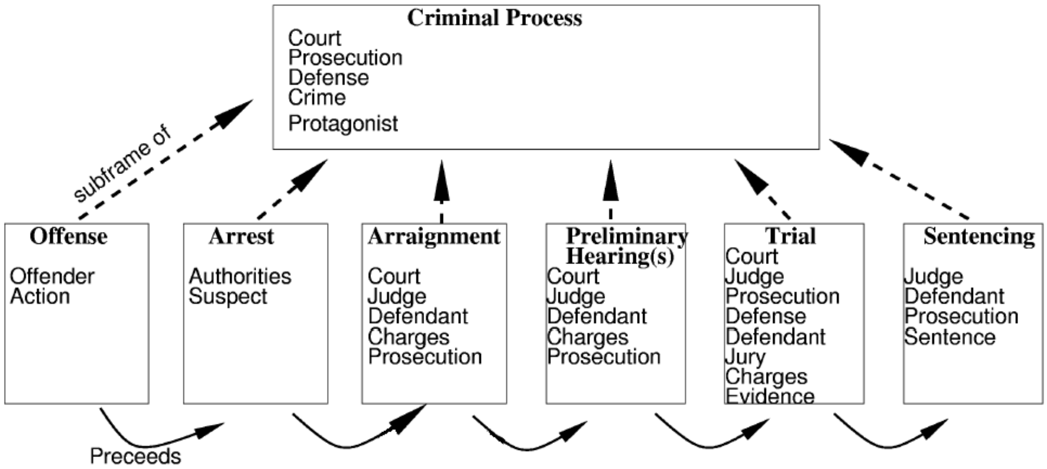


Figure 3.8: *The* `Criminal Process` *frame and some of its subframes according to [Narayanan et al., 2003]*

In [Narayanan et al., 2003] an encoding of FrameNet I data in the semantic web language DAML+OIL is presented. At FrameNet II's webpage it is claimed, that the database currently contains descriptions of more than 8,600 lexical units based on more than 135,000 annotated sentences. The semantic information is available for a wide range of natural language processing applications, including question answering, machine translation, and information extraction. The FrameNet database differs from existing lexical resources in the specificity of the frames and semantic roles it defines, the information it provides about relations between frames, and the degree of detail with which it describes, for each lexical item it covers, the possible syntactic realizations of semantic roles. Statements that are annotated with frame elements[30] are prepared to be expressed, as stated in section 4.1.

The use of frame semantic annotations is explored for a large German lexicon in the SALSA[31] project. The manually annotated FrameNet frame structures are created with a specialized annotation tool, see [Erk et al., 2003a] and [Erk et al., 2003b].

[29]Preview: if a criminal process had to be described with SITUATIONALSTATEMENTS in UbisWorld, this frame could be used to model the semantic roles.

[30]Preview: the statement's `range=framenet` and the statement's `object=`*the whole FrameNet expression*

[31]SALSA = The Saarbrücken Lexical Semantics Acquisition Project

3.4 Design Decisions and Goals

In this section we briefly discuss the design decisions and goals that result from the analysis of the related generic user modeling systems, the related metadata applications and the related external ontologies and knowledge bases that were presented in this chapter.

3.4.1 Discussion about U2M and Related Systems

We see our newly developed U2M user model service in continuation of the DOPPELGÄNGER project, the UM toolkit, the PERSONIS project and the DEEPMAP user modeling system. Ideas, techniques and user model dimensions from all of these four systems have been incorporated into our approach. However, even more changes and extensions have been undertaken in order to come up with our ubiquitous user modeling framework.

- The early generic user modeling system DOPPELGÄNGER was highly motivating and influential for our approach. Even though the used technology is nowadays mostly outdated, the ideas stayed the same. Most of Orwant's visions are now realized in our presented new framework.
- The UM toolkit's accretion representation is used for SITUATIONREPORTS, however much more metadata is stored along with the evidences.
- The idea of the *resolvers* in the PERSONIS system is applied but extended from purely syntactical matching to ontology-driven semantical conflict resolution.
- The DEEPMAP user modeling system is a state of the art user modeling system for the tourist domain. However, it is not realized as a web service and with semantic web ontologies. We did not follow the LDAP path, since we put the same functionality into action with the more prominent and more general tools from the semantic web world.
- The CPExchange specification influenced the development of SituationML and UserML, but the degree of freedom was not sufficient since no ontology mapping is provided.
- The idea behind HumanML is related to the idea of the user model ontology as introduced in section 5.2. However, the technology is based on XML only and not on the semantic web. Furthermore, the approach is very broad, since it should be valid for humans and humanoid animations. However the main reason not to use HumanML was, that it was still under development and not available.
- The CC/PP vocabulary is used to represent dimensions of resource-adaptive systems. However, the vocabulary for the user preferences is not used, since we integrate the general user model ontology GUMO.

To summarize the discussion about related systems, not only have the earlier generic user modeling systems been taken into account, but so too have the metadata applications. Some of them are integrated directly, some are recalled on demand and some of them are only influences on our new framework.

3.4.2 Discussion about UbisWorld and Related Ontologies

The following seven arguments discuss the relation between external ontologies or knowledge bases as described in the previous section and UBISONTOLOGY plus the general user model ontology GUMO:

- The Cyc knowledge base is not freely available and not easy to handle.
- The SUMO/MILO ontologies are also still under construction and cover only approximately 25% of the concepts that are needed in the special domain of ubiquitous user modeling. However, several concepts are used as resources. Most recently a cooperation to integrate SUMO and GUMO has started.
- The UBISONTOLOGY is intended to contain - apart from the classes and concepts - also their instances. In SUMO and MILO these instances of interest are not expected and had to be defined externally in order to develop a blocks world.
- The top-level elements in UbisWorld are defined according to the identified needs of ubiquitous user modeling, while other ontologies take a different design focus.
- UBISONTOLOGY is a specialized domain ontology that is new and had to be defined in one block rather than by combining several incomplete parts from already existing ontologies that were developed under different intentions.
- The WordNet lexicon is integrated into the SUMO/MILO approach as well as the UbisWorld approach.
- Even though the huge semantic information within FrameNet II is not used directly in the UBISONTOLOGY, the concept of frame elements has influenced the design of the situation model. Furthermore complete FrameNet frames can directly be represented within SITUATIONALSTATEMENTS.

To conclude the analysis of external ontologies and knowledge bases, we claim that the UBISONTOLOGY can currently not be replaced by any other set of ontologies. Especially from the point of view of the integrating research of ubiquitous computing, user modeling and semantic web, its development was necessary. However, it should be used together with other existing ontologies, especially with SUMO/MILO and WordNet.

3.4.3 Some Design Decisions and Goals

This subsection presents some of the design decisions and goals that stem from the state of the art in related work. Further design decisions are presented within succeeding chapters, close to their defined terminology. It is important . . .

- . . . *to support inconsistencies:* in [Kono et al., 1994] it is pointed out that users themselves may have inconsistent believes and thus a system that always tries to construct a consistent user model imposes constraints that are inappropriate. In [Kay et al., 2003] it is argued, that especially in ubiquitous computing the unreliability of information from sensors leads to inconsistencies in the user models.

- ... *to support adaptive conflict resolution:* in order to resolve these inconsistencies, conflict resolution at runtime will take place. Now, we ask for *adaptive* conflict resolution such that each user and system can influence the selection of conflict resolution strategies.
- ... *to support modularity:* clear separation between storage, service, inference, representation and semantic is envisaged. A motivation for a modular approach to user modeling is presented in [Torre, 2000].
- ... *to support clarity: clarity in information design* according to [Shedroff, 1999] is a major goal for the knowledge representation tasks. We try to minimize complexity while maximizing the semantic outcome.
- ... *to support decentralization:* the model of communication in intelligent environments calls for decentralization. [Fink, 2004] recommends carrying out further research on decentralized user modeling. One technical implication could for example be, that we won't have permission to delete information in some servers.
- ... *to support scrutability:* the user should be provided with means to examine the user model. In [Kay et al., 2003] scrutability is seen as a foundation for user control over personalization.
- ... *to support communication:* different user-adaptive systems need to communicate with each other. This is the basic assumption and design decision in our approach.
- ... *to support situated interaction:* user modeling in intelligent environments has to support situated interaction that stems form the idea of ubiquitous computing.
- ... *to support external ontologies:* since user modeling for situated interaction can reach all parts of the world, and by that all elements of the representational description, only an open approach that includes external ontologies promises to prove scaleable.

Part II

Knowledge Representation and Knowledge Management

4 SITUATIONAL STATEMENTS & EXTENDED RESOURCE IDENTIFIERS

The need for the integration of ideas from the three research areas "User Modeling", "Ubiquitous Computing" and "Semantic Web" is inherent in the definition of the basic concept of SITUATIONALSTATEMENTS, that is introduced in this chapter. A situational statement carries information about user model entries, context data or low-level sensor data. It forms the central, uniform structure for representing any situational information. The first part of this chapter presents a new solution to the issue of decomposing descriptions of complex situations into structured, uniform sentences, so-called SITUATIONALSTATEMENTS. In the second part of this chapter, the concept of SITUATIONALSTATEMENTS is realized by the XML application `UserML`, which serves as an exchange language for ubiquitous user-adaptive systems.

4.1 How to Represent Information about Complex Situations?

The question that arises from the point of view of knowledge representation is: *How can all these different types of contextual information be represented in a handy, uniform way?* On the way to find a solution, the story behind the picture in figure 4.1 is extended, analyzed and annotated with semantical meta information.

4.1.1 Situational Descriptions in the Airport Scenario

Let us assume that an inference system deduces that *the passenger Peter is now most probably under high time pressure because he is near the duty-free shop of the airport, while boarding of his flight closes in a few minutes. Additionally, his walking speed sensors report fast walking. According to his privacy settings, this information is only freely available to preselected people and systems.*

The same description could be annotated as: "An inference system (`creator`) deduces that Peter (`subject`) is now (`start`) most probably (`confidence`) under high (`object`)[1] time pressure (`predicate`), because (`evidence`) he is near the duty-free

[1]The naming of the `object` and `predicate` is inherited from the naming of the original RDF triple, while

Figure 4.1: *This image depicts a typical complex situation of a passenger at an airport. He is interacting with a mobile device, probably a pedestrian navigation system, carrying some luggage over his shoulder while hurrying towards a gate. This image produces the impression of time pressure and slight despair. The bridge between such an extreme situation and human-computer interaction research is the fundamental need for adaptation in the communication between this passenger and his instrumented environment.*

shop (`position`) of the airport (`location`), while boarding of his flight closes in a few minutes (`evidence`). Additionally, his walking speed sensors (`creator`$_2$) report fast walking (`predicate`$_2$`, object`$_2$). According to his (`owner`) privacy settings (`privacy`), this information is only freely available to preselected people and systems (`access`)."

The first insight from this annotation example is that one can separate the whole description of this airport scenario into smaller, sentence-like units, that is to say into a sentence about Peter's time pressure, a sentence about Peter's walking speed, and probably a sentence about the boarding time of Peter's flight. However, these intuitive semantic roles do not lead straight forward to a clear structure, if one wants to omit the complexity of natural language.

4.1.2 Remarks on Design Decisions for the User Model Language

The following three different approaches from knowledge representation research influenced the design decisions for the model of SITUATIONALSTATEMENTS, which is defined in the next section:

- the resource description framework RDF, with subject-predicate-object triples, reification, collections and all the theory of semantic web in support, as well as other semantic web languages like OIL, DAML or OWL, see chapter 2.3 for an introduction.

the first one carries the *value* and the second one the *attribute*.

- the FrameNet approach, see section 3.3.4 for a short introduction, with the annotations playing the role of frame elements, allows the links between semantics and syntax in the results of lexical analysis to be formalized. Semantic frames are schematic representations of situations involving various participants, props, and other conceptual roles.

- extended markup language applications concerning users or humans in general, like `HumanML`[2], the Human Markup Language, `VHML`[3], the Virtual Human Markup Language, `EML`[4], the Emotional Markup Language or `CPExchange`[5], the Customer Profile Exchange

The analysis of related markup languages and approaches led to an early decision to create a new, specialized user modeling (markup) language, with the acronym `UserML`[6]. One purpose of the `UserML` project is to develop a platform for communication about partial user models in a ubiquitous computing environment, where all different kinds of systems work together to satisfy the user's needs. Standardization is of great importance but individual applications also call for very specific solutions.

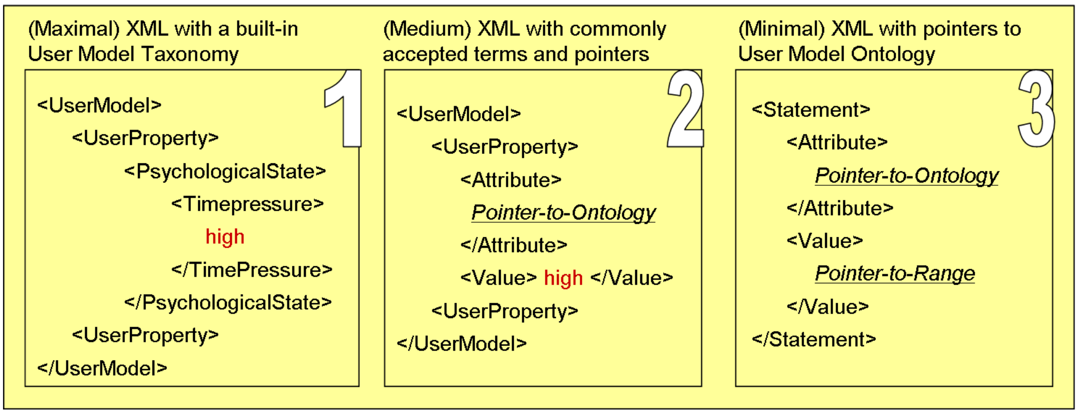


***Figure 4.2**: Three design drafts for a user model representation language, the first one with the built-in user model taxonomy, the second one with a commonly accepted, minimal taxonomy and the third one only with pointers to a user model ontology*

Three different ways of forming user model information into the XML tree structure are illustrated in figure 4.2. All three have advantages and disadvantages. The first approach transforms the whole complexity of the domain knowledge into the tree structure. The advantage is that no external taxonomy or ontology is needed, but the main disadvantage is the acceptance in the research community of any fixed definition and the maintainability for

[2] HumanML homepage: `http://www.humanmarkup.org/`
[3] Virtual Human Markup Language homepage: `http://www.vhml.org/`
[4] Emotional Markup Language is part of the VHML project:
`http://www.interface.computing.edu.au/documents/VHML/2001/`
[5] Customer Profile Exchange homepage: `http://www.cpexchange.org/`
[6] see [Heckmann, 2002a] for a discussion

extensions if the user model dimensions are coded into the DTD or XML-Schema definition. Another important disadvantage is which point of view should be chosen and with which focus? Some possibilities for instance:

- **the user seen as human-being**: human-being related properties like emotions, cognition, or physical descriptions could form the main focus

- **the use of user-adaptive systems**: putting the user-adaptive systems in the main focus causes the problem that demand-orientation and system-dependency could lead to too system-specific structures

- **user interaction centered**: the user interacts with his/her environment including computing systems, context and the whole situation, resulting in the problem of modeling the environment with the whole interaction process

The idea for the second approach in figure 4.2 of combining the minimal accepted common user model taxonomy with pointers to external ontologies was developed in a discussion with Henry Thompson[7], see [Carolis et al., 2003] for a similar approach. Surprisingly, not even a minimal intersection of top-level terms could be found in the user modeling research community[8]. For instance the concept of a *usage model* is widely used, see [Kobsa, 2001a], while on the other hand, the concept of a *personal journal*, see [Kleinbauer et al., 2003], has also been developed. The attempt to find a common, merged taxonomy for the three research areas of user modeling, context-awareness and resource-adaptivity seemed impossible.

The third approach in figure 4.2, however, showed a solution to the problems of the previous two. Thus, the best solution to the information design task seemed to be the complete separation of the domain of user modeling (as well as the domain of context-awareness, see figure 4.3 for a simplified taxonomy for the airport scenario), and the structure of the representation language. The user model domain knowledge will be modeled in a user model ontology and the structure in an extended resource-description-like framework.

The advantage of this approach is that if the research community subgroups can not find a commonly accepted user model ontology, they can all use their own ones, but still use the same framework. Of course, the third design draft of figure 4.2 is not powerful enough to cover all the demands of modern user modeling tasks, since it only offers the simple structure of attribute-value pairs. However, it points towards the resource description framework and further to an extension of the resource description framework, which is described in the following section. Another important advantage of this approach is that the property information about the object of interest is equally represented in user-adaptive systems, context-awareness and resource-adaptive computing, as described in [Heckmann, 2003b].

[7]Discussion with Henry Thompson on the design for a user modeling markup language at the University of Edinburgh, 2002

[8]User Modeling Research Community homepage: `http://www.um.org`

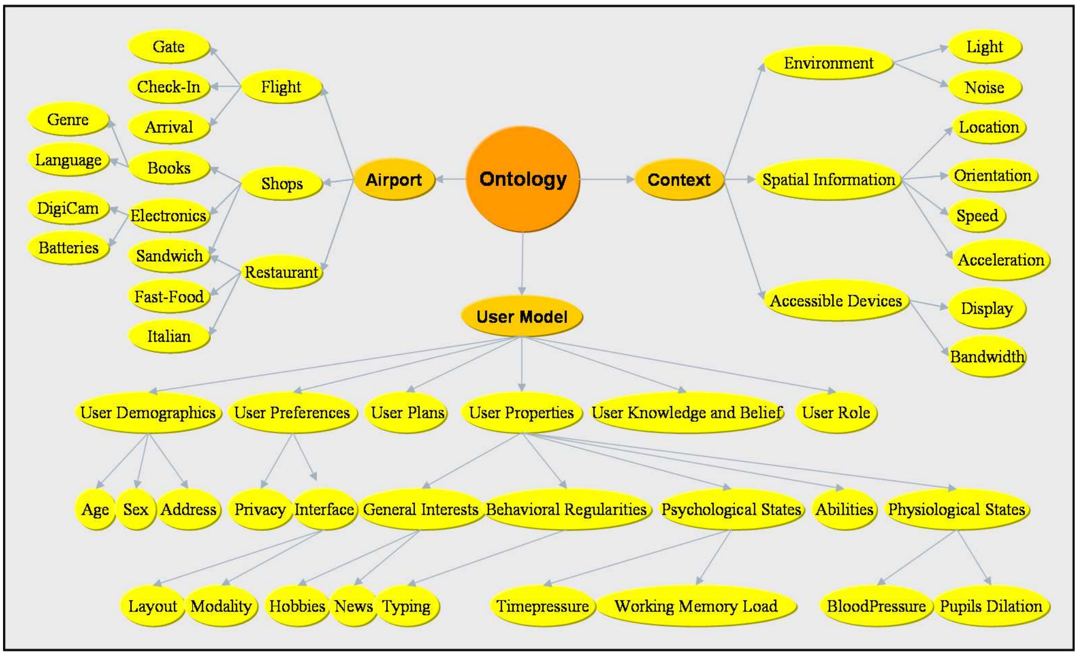


Figure 4.3: *A simplified, partial taxonomy for the airport scenario, which is divided into three sub parts, namely the airport related domain concepts on the left, the context related concepts on the right, and the user model related concepts on the bottom*

4.1.3 From RDF to Extended Triples

Together with the reification mechanism, the resource description framework RDF[9] forms a powerful tool for knowledge representation. RDF is based on triples of `subject`, `predicate` and `object` as shown in figure 4.4: a resource, the subject, is linked to another resource, the object, denoting the value through an arc labeled with a third resource, the predicate. It is said that the `subject` has a property `predicate`, valued by `object`.

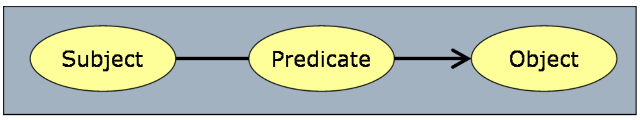


Figure 4.4: *The basic RDF triple with subject, predicate and object*

With this basic mechanism, one could already represent user model information like: "the `subject`=*Peter* has the `predicate`=*CognitiveLoad* with the `object`=*high*". However, if one needs to add more information to this description, for example a temporal aspect like *"the time when Peter was under high cognitive load"*, RDF offers either the

[9]Resource Description Framework homepage: `http://www.w3c.org/rdf`

reification mechanism with which the basic triple is transformed into one new resource, that can be used as a whole in other statements, or the construct of collections. Setting up a specialized structure within RDF was introduced in [Heckmann, 2002a] as a third possibility into the concept of *Extended Triples*. See figure 4.5 for a diagrammatic view to this extended-triple RDF-application.

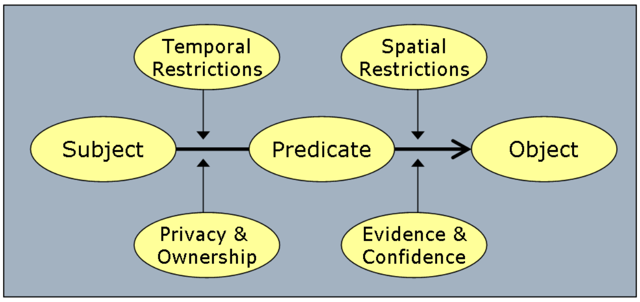


Figure 4.5: *The extended RDF triple with subject, predicate and object, as well as temporal restrictions, spatial restrictions, privacy and ownership, evidence and confidence*

RDF is extended with predefined semantical slots, like *owner* or *confidence* instead of handling triples with reification. The meta level information of a statement like *"who is responsible for this piece of information?"*, or *"what is the confidence value for it?"*, is combined with the actual content of the statement like *"the cognitive load of this person is high"*. Furthermore the basic information content is enriched with temporal and spatial constraints like *"this property is valid between now and tomorrow"*.

The so-called *Extended Triple* is in fact a 7-tuple, with temporal- and spatial restrictions as well as meta-data about ownership, privacy, evidence and confidence. The idea is to allow more powerful, but still structured, statements about situations. The new arrow type in figure 4.5 has the meaning of *adding information* to the basic triples. Nevertheless, the graph of the extended triples did not prove to be flexible enough for further extensions and more importantly, from an implementation point of view, did not prove to be easily transformable into the relational database system. Therefore, the idea of extending basic RDF finally led to the structure of SITUATIONALSTATEMENTS, which is defined in the following section.

4.2 The Model of SITUATIONALSTATEMENTS

SITUATIONALSTATEMENTS represent partial descriptions of situations like user model entries, context information or low-level sensor data. They form the main data structure in the given situation concept. What is new and special about SITUATIONALSTATEMENTS is the extensive layered approach shown in the onion model of figure 4.6. The information is organized in a predefined hierarchy of meta levels wrapped around the mainpart information.

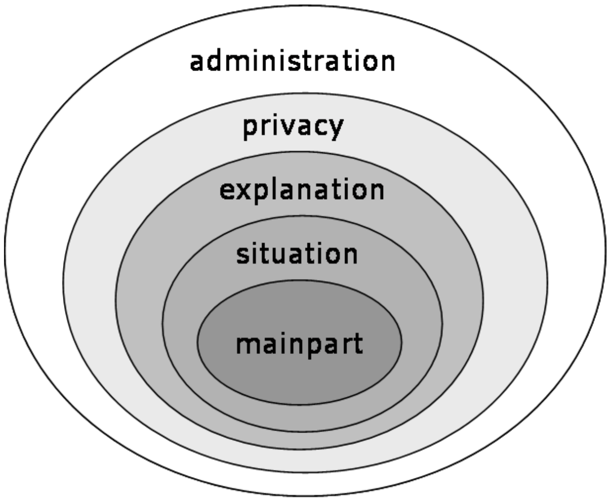
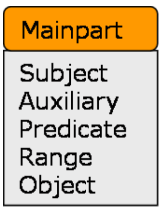


***Figure 4.6**: Onion model to represent the five layers of situational statements*

These layers of meta level information within SITUATIONALSTATEMENTS can be seen as a collection of slots or attributes that are arranged in five boxes, namely the `mainpart` box, the `situation` box, the `explanation` box, the `privacy` box and the `administration` box. These boxes have an organizing and structuring functionality. Their meaning is explained in detail in the following subsections.

4.2.1 The Mainpart Box



The leading content information of the described partial situation can be expressed by the attribute-value pairs of the mainpart box. The three attributes `subject`, `predicate` and `object` are necessary attributes for each SITUATIONALSTATEMENT, while `auxiliary` and `range` are optional attributes. The mainpart attributes are directly derived from the theory of RDF triples with subject-predicate-object and form a superset of them, see section 4.4 for a discussion. That means that every RDF triple can be expressed by a SITUATIONALSTATEMENT. On the other hand, all SITUATIONALSTATEMENTS can be translated back into RDF documents with reification, see section 4.4.3 for a short introduction. The different syntactic specifications are shown in the appendix section A.1.

A basic example of information that could be found in a simple user model is: `subject=`*"Peter"*; `predicate=`*"time pressure"*; `object=`*"high"*; with the intended meaning that *Peter has high time pressure*.

A second example that could be expressed by a context-aware system is: `subject=`*"mobile phone"*; `predicate=`*"battery power"*; `object=`*"low"*; with the intended meaning that *the actual battery power of the mobile phone is low*.

The original RDF predicate is split into `auxiliary`, `predicate` and `range` in order to be more flexible in expressing properties. A new surplus in expressivity according to RDF is the auxiliary construction. It was introduced as a unifying element with respect to the special needs of knowledge representation with pointers to external ontologies. To demonstrate the necessity of this auxiliary construction the following example is given: the term *teaching* is defined and described in a semantic web ontology, while a user-adaptive system might want to represent the three statements:

- *Peter is currently teaching*
- *Peter likes teaching very much*
- *Peter knows a lot about teaching*

With the means of SITUATIONALSTATEMENTS, the user-adaptive system would define `subject=`*"Peter"*, `predicate=`*"teaching"* and the corresponding `auxiliary` as *"interest"* or *"knowledge"*. A predefined list of auxiliaries can be found in section 5.2.

The `range` attribute offers a new degree of freedom to the ontology definition: it decouples the definition for the predicate from possible range scales. The following examples illustrate this argument:

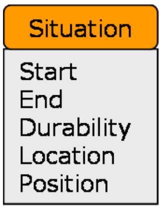
- *Peter's time pressure is low (within a scale of low-medium-high)*
- *Peter's time pressure is 0.6 (within a numeric scale between 0 and 2)*
- *Peter's time pressure is 30% (within 0% - 100%)*

In terms of SITUATIONALSTATEMENTS, one would define `subject=`*"Peter"*, `predicate=`*"time pressure"* and `range=`*"low-medium-high"* or `range=`*"numeric-0-2"*, or `range=`*"percentage"* respectively, while these ranges are defined in a special part of the ontology. A mapping between different ranges, which is also defined in the ontology, is especially interesting for conflict resolution in chapter 6 and user model integration in chapter 7.

A more elaborate approach with uncertain knowledge, probability distributions and reasoning under uncertainty could also be integrated in SITUATIONALSTATEMENTS. For example, the *Dempster-Shafer Theory*, see [Dempster, 1968] and [Shafer, 1979] with measures of belief, based on evidences, or Bayesian probability distributions, see for example [Jensen, 2001], could be interesting for later extensions. However, at this point the focus is set on representational issues that stem from RDF. The *Discourse Representation Theory*, see [Kamp and Reyle, 1990], is used in natural language processing, to dynamically interpret so called *discourses* (linguistic units composed of several sentences like conversations, arguments or speeches). This theory provides a logical framework that could have been interesting to model SITUATIONALSTATEMENTS. However, since no connection is given towards the Semantic Web, this theory was not further recognized.

To summarize, the mainpart box extends the RDF triple with the attributes `auxiliary` and `range` in order to simplify and strengthen the definition of the general user model ontology GUMO.

4.2.2 The Situation Box



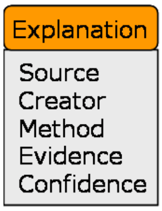
Information in the situation box is responsible for the temporal and spatial embedding of the whole statement in the real physical world. Since this situational aspect is especially important and dominant in ubiquitous computing, it gave its name to the whole concept of SITUATIONALSTATEMENTS. All attributes in this box are optional. The attribute `start` denotes the point in time when the statement becomes valid. The default value is the date-time when this statement was given. The attribute `end` denotes the point in time when the validity of the statement ends. This ending could be in the past, in the present or in the future. With this open approach one can handle the issue of the history in user modeling and context-awareness. Earlier statements are still available, they are not overridden or replaced by newer values. Instead the query answer mechanism returns - by default - the current value by temporal inference. Furthermore, the attribute `durability` carries the qualitative time span of how long the statement is expected to be valid (minutes, hours, days, years). In most cases when user model dimensions or context dimensions are measured, one has a rough idea about the expected durability, for instance, emotional states change normally within hours, however personality traits won't change within months. Since this qualitative time span is dependent on every user model dimension, a definition mechanism is prepared within the general user model ontology GUMO. Some examples of rough `durability` classifications are:

- physiologicalState.heartbeat - can change within seconds
- mentalState.timePressure - can change within minutes
- emotionalState.happiness - can change within a few minutes
- characteristics.inventive - can change within months
- personality.introvert - can change within years
- demographics.birthplace - can't change at all (under normal circumstances)

The idea is that if no new actual value is available after a while, one can still work with old values combined with probably reduced confidence values. The suggested terms for durability are not hard coded into the SITUATIONALSTATEMENT approach, but a predefined setting for possible values can be found in appendix B.2.3 on page 236. In addition to the duration time interval, together with a given requesting time, the complex temporal inference process can calculate the status of the statement like *not expired*, *just expired* or *expired*, see [Schmidt, 2005] for a discussion on controlled aging of user context.

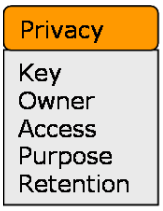
The attribute `location` defines the qualitative spatial location, where this statement takes place, in relation to the specialized spatial model for ubiquitous computing that is presented in the UBISWORLD ONTOLOGY section 5.3.2 from page 105 onwards, while the attribute `position` allows to specify the exact quantitative coordinates.

4.2.3 The Explanation Box



The collection of explanatory meta attributes is partly required by the user's right for explanation and partly by the conflict resolution mechanism for inferred user modeling data, see chapter 6.3. The `source` attribute contains the statement's origin, that is to say where the statement was stored, or from which sensor measurement it came from. With the help of this attribute, double entries can be detected in the user model integration process, see chapter 7.1. The user's right to ask for explanation is handled with the next three attributes: the `creator` names the system or person that is responsible for the creation of this statement. The `method` attribute leads to the answer "how" the mainpart was inferred, deduced or measured. Its attribute's value can either be a pointer to an ontology or free text. The list of evidence that supports the statement can be stored in the `evidence` attribute, while the `confidence` provides a place to store the creator's expected level of confidence in the statement. All attributes in the explanation box are optional.

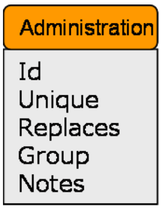
4.2.4 The Privacy Box



The privacy meta attributes are very important for the acceptance of user modeling data, since these attributes allow controlled distribution of sensitive data. The `key` attribute which forms an encrypted security key can be attached to the statement and lock it if the key can not be resolved. The `owner` attribute refers to the person or system that is in charge of managing the distribution and editing of the three following attributes: `access`, `purpose`, `retention`. The first one can contain the class of users or systems that are allowed to read the statement. The second one can put restrictions on the intention for which the statement may be used. And finally the third one controls how long the statement may be kept or used. See chapter 8.4.2 for a detailed description of privacy handling. The possible values for the `access` attribute are *public*, *friends* and *private*, for the `purpose` attribute: *commercial*, *research* and *minimal* and for the `retention` attribute: *long*, *middle* and *short*. According to these settings, the privacy handler will grant or deny access and delete the appropriate statements for privacy reasons.

New in this privacy approach is that the owner's intended privacy settings go with the statement itself on the information journey when it is exchanged into a distributed ubiquitous network. In other current user model services, once the information is given out to some client, the privacy information is not attached any more, and this client could spread this statement under new privacy settings.

4.2.5 The Administration Box



The administration box is the outermost layer in the layered onion model. It is meant to fasten organizational issues in huge sets of situational statements. The `id` attribute contains a locally unique identifier for referencing the statement and is mostly used as a primary key in a database. The `unique` attribute however forms a globally unique identifier for referencing the statement throughout the whole internet. It could either be a URI[10] or a GUID[11]. Since in our approach we cannot guarantee that the creator of the statement has rights or the means to delete a statement after creation, the `replaces` attribute makes it possible to state which statement should be replaced by a newer one. Especially sensors that constantly emit more up-to-date data receive the identification numbers of their last situational statements from the user model server as a return value and therefore replace older information without deleting it. Figure 4.7 on page 63 shows a rough classification of the statements into groups like *UserModel*,

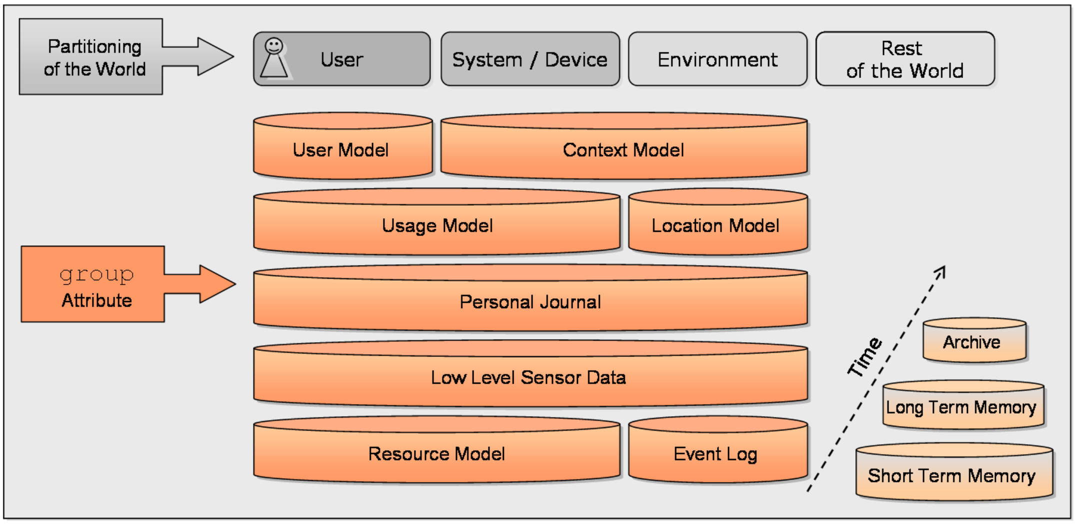


***Figure 4.7**: Possible values for the `group` attribute (within the administration box) in relation to the conceptual partition of the world and a temporal dimension*

UsageModel, *ContextModel* or *LowLevelSensorData* that can be expressed by the `group` attribute. However, this grouping is only of minor importance in our approach, since the statements are conceptually and physically stored at the same place. It is used within the smart situation retrieval process for efficiency reasons. In related systems, for example in the SPECTER system[12], a clear definition of these memory models plays a more important role, since each group is stored and treated independently, see e.g. [Kröner, 2005]. And finally,

[10]The W3C URL/URI homepage: `http://www.w3.org/Addressing/`
[11]A web service for the generation of *Globally Unique Identifiers* can for example be found at: `http://www.hoskinson.net/WebServices/GuidGenerator.asmx`
[12]SPECTER homepage: `http://www.dfki.de/specter`

the `notes` attribute has no predefined semantics and can contain additional unstructured meta information about the statement.

To summarize, the information represented in a SITUATIONALSTATEMENT is arranged in a layered fashion of five times five attributes. The mainpart attribute group contains the actual content of the statement, the situation group defines necessary temporal and spatial side conditions, the explanation group covers the creator and the creation-method as well as the evidence and confidence. The privacy group offers the possibility of expressing privacy meta data, while the attributes of the administration group cover the identification, replacement and grouping of the statements. A complete listing of these attributes, together with short descriptions, can be found in table 4.1.

Definition 4.1 (Situational Statement) *A* SITUATIONALSTATEMENT *is a collection of attributes as defined in table 4.1. The term* STATEMENT *without the prefix serves as an abbreviation.*

4.3 The Model of SITUATIONREPORTS

Individual SITUATIONALSTATEMENTS are only able to carry the information of one single entity or - more accurately - of one statement. The question of arranging several statements into a new data structure is now interesting.

4.3.1 From Situational Descriptions to SITUATIONREPORTS

The informal situational descriptions, as presented in section 4.1.1, can now be formalized with the help of SITUATIONALSTATEMENTS and an additional data structure. The easiest data structure for collecting several statements into one unit is the *bag*, which is a *set* with possible multiple elements. Figure 4.8 depicts a general SITUATIONREPORT schema, which is defined as an unordered collection of SITUATIONALSTATEMENTS.

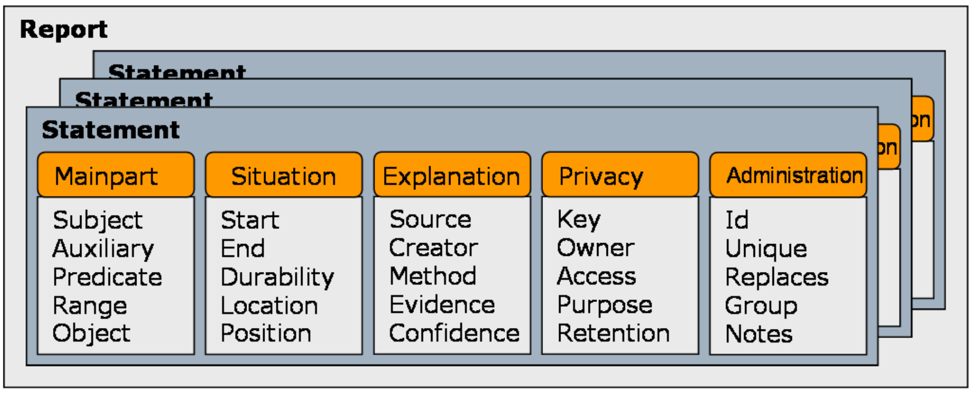


Figure 4.8*: A* SITUATIONREPORT *is defined as a bag of* SITUATIONALSTATEMENTS

Mainpart	*the basic, rdf-based five-tuple of situational statement attributes*
`subject`	the main entity the statement is about (mostly the user)
`auxiliary`	the auxiliary part of the predicate like "hasProperty" or "hasInterest"
`predicate`	the dimension or category of the user like "Age" or "Swimming"
`range`	the range of the object attribute, defaults are possible
`object`	the value for the subject-auxiliary-predicate triple of the statement

Situation	*temporal and spatial constraints, restricting the mainpart*
`start`	the point of time when this statement was given
`end`	the point of time when this statement is no longer valid
`durability`	the qualitative time span of how long the statement is valid
`location`	the qualitative spatial location where this statement takes place
`position`	the quantitative spatial location, the exact coordinates

Explanation	*a collection of explanatory meta attributes*
`source`	the origin where the statement is stored
`creator`	the person or system that is responsible for the creation of this statement
`method`	the manner by which the statement was created
`evidence`	a list of evidence that supports the statement
`confidence`	a number that displays the creator's expected truth of the statement

Privacy	*a collection of privacy meta attributes, controlling the distribution*
`key`	optional encrypted security key that can be attached to the statement
`owner`	the person or system that may manage the distribution
`access`	the class of users or systems that are allowed to use the statement
`purpose`	the qualitative purpose for which the statement may be used
`retention`	the qualitative time of how long the statement may be kept or used

Administration	*a collection of administrative meta attributes*
`id`	a locally unique identifier for referencing the statement in the database
`unique`	a globally unique identifier for referencing the statement
`replaces`	a unique identifier of another statement that has to be replaced
`group`	rough classification of the statement like "UserModel" or "SensorData"
`notes`	an additional attribute with free semantics, can serve as a variable

Table 4.1: *The intended meaning of the twenty-five predefined attributes of Situational-Statements, containing main content information, as well as situational, explanatory, access and administrative meta-information, in order to describe complex situations. This selection forms a minimal set for the special needs according to the integration of user modeling, ubiquitous computing and semantic web.*

Definition 4.2 (Situation Report) *A* SITUATIONREPORT *is an unordered collection of* SITUATIONALSTATEMENTS*, that belong to the same situational description. The term* REPORT *is used as an abbreviation.*

4.3.2 A SITUATIONREPORT in the Airport Scenario

The partial situation model of the airport scenario of section 4.1.1 on page 53 could be represented by the three SITUATIONALSTATEMENTS of figure 4.9. However, the attribute values are not correctly presented. They are only indicated and should be understood as simplified resources. The definition of these values and their datatypes can partly be found in chapter 5.

Situational Statement / Box	Situational Statement / Box	Situational Statement / Box
Mainpart	**Mainpart**	**Mainpart**
Subject = Peter	Subject = Peter	Subject = Flight LH225
Auxiliary = hasProperty	Auxiliary = hasProperty	Auxiliary = hasPlan
Predicate = timePressure	Predicate = walkingSpeed	Predicate = boarding
Range = low-medium-high	Range = slow-medium-fast	Range = time
Object = high	Object = fast	Object = in 10 minutes
Situation	**Situation**	**Situation**
Start = 2003-04-16T19:15	Start = 2003-04-16T19:14	Start = 2003-04-16T19:14
End = ?	End = ?	End = ?
Durability = few minutes	Durability = few minutes	Durability = few minutes
Location = airport.dutyfree	Location = airport.dutyfree	Location = airport.gate23
Position = x34-y22-z15	Position = x32.y23.z15	Position = ?
Explanation	**Explanation**	**Explanation**
Source = peter.repository	Source = sensor.repository	Source = airport.repository
Creator = airport.inference	Creator = sensor.PW	Creator = airport.inference
Method = deduction13	Method = Bayes	Method = deduction13
Evidence = id2, id3	Evidence = LowLevelData	Evidence = fight-system
Confidence = most-probably	Confidence = 0.8	Confidence = 0.6
Privacy	**Privacy**	**Privacy**
Key = ?	Key = ?	Key = ********
Owner = Peter	Owner = Peter	Owner = Airport
Access = friends-only	Access = friends-only	Access = public
Purpose = research	Purpose = research	Purpose = commercial
Retention = 1 day	Retention = 1 week	Retention = 1 month
Administration	**Administration**	**Administration**
id = 1	id = 2	id = 3
unique = u2m.org#154123	unique = u2m.org#154124	unique = u2m.org#154125
replaces = ?	replaces = u2m.org#154109	replaces = u2m.org#152148
group = UserModel	group = UserModel	group = ContextModel
notes = ;-(	notes = ;-|	notes = ;-)

***Figure 4.9**: SITUATIONREPORT with three SITUATIONALSTATEMENTS from the airport scenario, with meta data plus the three mainparts: (1) Peter is most probably under high time pressure, (2) Peter is walking fast, and (3) Boarding of flight LH225 will close in 10 minutes*

4.3.3 Aggregating Statements in SITUATIONALREPOSITORIES

"Belonging to the same situational description" distinguishes the SITUATIONREPORTS from the most general collection of statements, namely the SITUATIONALREPOSITORY, that are meant to save statements that do not necessarily belong to the same situation. SITUATIONALREPOSITORIES serve as containers for any kind of SITUATIONALSTATEMENTS. An important additional property is that repositories must be identifiable in order to state the origin of a statement after it has been exchanged.

> **Definition 4.3 (Situational Repository)** *A* SITUATIONALREPOSITORY *is a uniquely identified collection of* SITUATIONALSTATEMENTS. *The term* REPOSITORY *is used as abbreviation.*

However, the intended semantics for the attributes of SITUATIONALSTATEMENTS, together with their under-specified data types, leave some scope for variations within concrete instantiations of this conceptual model in user-adaptive services. Such a concrete, further specified instance of this rather general model of SITUATIONREPORTS has been defined and implemented along with `SituationML` in section 4.4.

4.3.4 Remarks about the Closed World Assumption

Intelligent behavior depends on the knowledge that is known about the environment. Much of this knowledge is descriptive and can be expressed for example in declarative situational statements. For the semantic interpretation of statements it is important to know the standpoint, regarding the frame axiom and the closed word assumption.

A theory (or knowledge base) is called *complete* if either every ground atom (here, the auxiliary-predicate-range triple) or its negation is in the theory. If the statement *"It is raining at location X"* is true, the negation of this sentence *"It is not raining at location X"* must be false. The question *"Is it raining at location Y?"* while no weather information is given for location Y, leads to the issue of incomplete knowledge and the question whether one should assume that it is not raining there because it was not explicitly stated. The *closed world assumption* is a straight forward convention to complete a theory in the way that the negation of a statement is assumed if the statement cannot logically be deduced from the theory.

For the work with SITUATIONREPORTS we do not apply the closed world assumption, since we allow contradicting statements at the same time, that could be resolved by conflict resolution as shown in section 6.3. We do not intend to model a complete blocks world, but we intend to model parts of the real world, see section 5.3. Or more precise, we intend to model situations in the sense of [Barwise and Perry, 1983], that correspond to the limited parts of reality we perceive, reason about and live in.

4.4 The Syntax of `SituationML` and `UserML`

The overall idea of integrating the three research areas of "User Modeling", "Ubiquitous Computing" and "Semantic Web" is especially inherent in the definitions in this section.

The general model of SITUATIONALSTATEMENTS, as proposed in the previous section, is transformed into several XML applications, as well as RDF applications and OWL applications. These syntactical variations are defined in parallel, since special circumstances need specialized tools, i.e. the same information needs to be read by human beings, it needs to be prepared for inference systems and integrated into semantic web ontologies, it needs to fulfil database constraints for storage, and at the same time the transfer volume has to be optimized for mobile interaction settings. Because of these manyfold demands, and since the SITUATIONALSTATEMENTS form the fundamental background of the whole u2m.org-research, an investigation of the information design possibilities has been undertaken.

The XML application family `SituationML` includes a concrete realization of the abstract model of SITUATIONREPORTS and SITUATIONALSTATEMENTS. `SituationML` can be read as *Situation Markup Language*. However, since SITUATIONALSTATEMENTS also cover the conceptual purpose of user models and context models, the terms `UserML` and `ContextML` are also used as abbreviations. The acronym `UserML` stands simultaneously for *User Model Markup Language*, as well as *User Model Exchange Language*. Analogically, the acronym `ContextML` stands for *Context Markup Language*, as well as *Context Exchange Language*.

4.4.1 SITUATIONALSTATEMENTS within `SituationML`

The following list in figure 4.10 presents the default representation for SITUATIONALSTATEMENTS within `SituationML`. All information is put into XML-elements, while the corresponding attribute groups like *explanation* or *privacy* are omitted for simplification reasons, since a flat tree structure is the easiest for prototyping. No attributes are used. Empty elements can be omitted, which leads to short statements, if no meta data are needed. The root element is named `statement`.

The variables from `a1` to `a25` do not necessarily carry the information in simple structures. Complex data types and ranges are defined in section 4.5. At this point, the focus lies on the pure design decision of the top level XML structure.

The following section presents three alternative syntax variations that all have their own advantages and disadvantages. The document type descriptions (the DTDs) of `SituationML` specifications are listed in appendix A.2, the XML Schema is listed together with its schema tree in the appendix A.3.

4.4.2 Syntax Variations on `SituationML/XML`

In the first syntax variation on the left hand side of figure 4.11, the attribute-groups introduce a new level in the XML tree. It is a one-to-one equivalent realization of the SITUATIONALSTATEMENTS model as defined in chapter 4.4. The two following syntactic variations represent the SITUATIONALSTATEMENT attributes as XML attributes. This has the advantage of compactness, but the disadvantage of restricted expressivity. The Mix version is a mixture

```
<statement>
     <subject>    a1  </subject>
     <auxiliary>  a2  </auxiliary>
     <predicate>  a3  </predicate>
     <range>      a4  </range>
     <object>     a5  </object>

     <start>      a6  </start>
     <end>        a7  </end>
     <durability> a8  </durability>
     <location>   a9  </location>
     <position>   a10 </position>

     <source>     a11 </source>
     <creator>    a12 </creator>
     <method>     a13 </method>
     <evidence>   a14 </evidence>
     <confidence> a15 </confidence>

     <key>        a16 </key>
     <owner>      a17 </owner>
     <access>     a18 </access>
     <purpose>    a19 </purpose>
     <retention>  a20 </retention>

     <id>         a21 </id>
     <unique>     a22 </unique>
     <replaces>   a23 </replaces>
     <group>      a24 </group>
     <notes>      a25 </notes>
</statemtent>
```

***Figure 4.10**: `SituationML`/XML is the default representation*

between Max and Min and it turned out that statements in this version are especially good for being read by humans with ordinary browsers.

The minimal syntactic variation, on the right hand side in figure 4.11, only uses XML attributes and thus has the disadvantage of restricted expressivity within the variable. However, it is the premium choice if bandwith for mobile systems is a problem and every single character counts.

The tools that were developed for handling `SituationML` can handle and transform all these variations. Thus, the individual application can choose which representation suits best. Of course, all XML applications can be used with namespaces. The recommended namespace is `st:` or `s:` and should be read as “statement”. It is mapped to `http://www.u2m.org/2003/02/situation/`. An example is given in appendix A.1.2 on page 206.

Situational Statement / XML (Max)

```
<statement>
    <mainpart>
        <subject> a1 </subject>
        <auxiliary> a2 </auxiliary>
        <predicate> a3 </predicate>
        <range> a4 </range>
        <object> a5 </object>
    </mainpart>
    <situation>
        <start> a6 </start>
        <end> a7 </end>
        <durability> a8 </durability>
        <location> a9 </location>
        <position> a10 </position>
    </situation>
    <explanation>
        <source> a11 </source>
        <creator> a12 </creator>
        <method> a13 </method>
        <evidence> a14 </evidence>
        <confidence> a15 </confidence>
    </explanation>
    <privacy>
        <key> a16</key>
        <owner> a17 </owner>
        <access> a18 </access>
        <purpose> a19 </purpose>
        <retention> a20 </retention>
    </privacy>
    <administration>
        <id> a21 </id>
        <unique> a22 </unique>
        <replaces> a23 </replaces>
        <group> a24 </group>
        <notes> a25 </notes>
    </administration>
</statemtent>
```

Situational Statement / XML (Mix)

```
<statement>
    <mainpart
            subject = "a1"
            auxiliary ="a2"
            predicate = "a3"
            range = "a4"
            object = "a5" />
    <situation
            start = "a6"
            end = "a7"
            durability = "a8"
            location = "a9"
            position = "a10" />
    <explanation
            source = "a11"
            creator = "a12"
            method = "a13"
            evidence = "a14"
            confidence = "a15" />
    <privacy
            key = "a16"
            owner = "a17"
            access = "a18"
            purpose = "a19"
            retention = "a20" />
    <administration
            id = "a21"
            unique = "a22"
            replaces = "a23"
            group = "a24"
            notes = "a25" />
</statement>
```

Situational Statement / XML (Min)

```
<statement
        subject = "a1"
        auxiliary = "a2"
        predicate = "a3"
        range = "a4"
        object = "a5"
        start = "a6"
        end = "a7"
        durability = "a8"
        location = "a9"
        position = "a10"
        source = "a11"
        creator = "a12"
        method = "a13"
        evidence = "a14"
        confidence = "a15"
        key = "a16"
        owner = "a17"
        access = "a18"
        purpose = "a19"
        retention = "a20"
        id = "a21"
        unique = "a22"
        replaces = "a23"
        group = "a24"
        notes = "a25"
/>
```

Figure 4.11*: Three syntax variations for* `SituationML`

4.4.3 `SituationML/RDF` and `SituationML/OWL`

Since most of the attribute values will be used like resources in RDF, the connection between `SituationML` and RDF is analyzed in this subsection. Because of its rich vocabulary RDF allows several possibilities of modeling the same information. Interesting for statements is the reification method, since the meta tags enrich the main sentence directly. Figure 4.12 shows `SituationalStatement/RDF_Graph(Reification)`, while the reified mainpart is simplified and does not incorporate the auxiliary-predicate-range concept yet.

The translation of this RDF_Graph model into RDF_XML is not presented in the appendix but it could easily be constructed. For our purposes the reification concept is not necessarily needed in its original RDF format if we move the focus from the mainpart to the whole statement itself, where the mainpart attributes are also used as pointers to the central identification point of the statement as shown in figure 4.13.

Representing SITUATIONALSTATEMENTS in this simple graph format leads to the

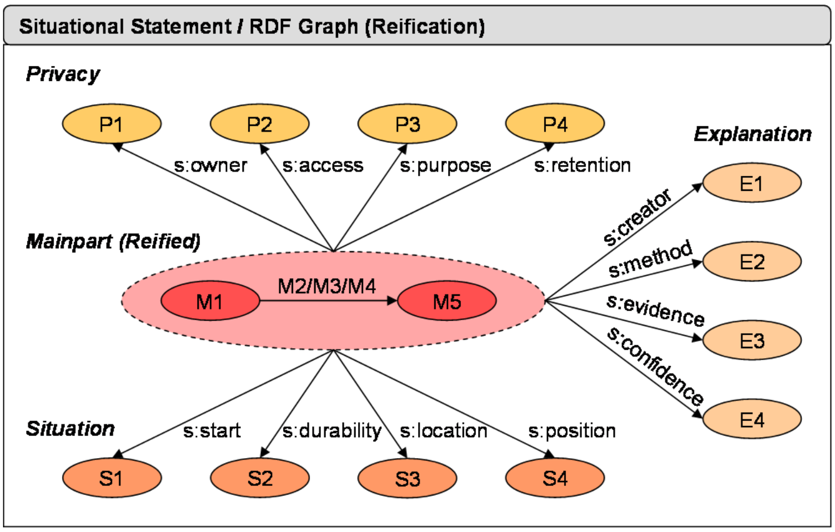


***Figure 4.12**: A* SITUATIONALSTATEMENT *schema in RDF graph notation in the variation with reification. The reified mainpart is enriched with some situation attributes, explanation attributes and privacy attributes. Each oval represents one resource. The attributes are realized as predefined RDF predicates, while their values can be defined as RDF objects*

RDF_XML document as presented in figure 4.14. The similarity to the basic XML representation as shown in 4.10 on page 69 is remarkable. Because of the external definition of the semantic roles of the attributes and the implicit layered approach, no further XML structure is needed. With the help of SituationML/RDF_XML, the statements can straightforwardly be integrated into RDF ontologies for inferences. However, since most modern ontologies are represented either in the semantic web languages DAML+OIL or OWL, we have to allow `SituationML` to be integrated into these ontologies.

The major advantage of using the statement's identification as the RDF's subject and preferring the RDF graph without reification is that is is possible to choose the data type for every attribute, that is to say to choose between *URI reference* and *literal*, for the `st:subject`, the `st:predicate` and the `st:object`. As described in [Lassila and Swick, 1999], an RDF triple contains three components, namely the subject (1), which is a URI reference (or a blank node), the predicate (2), which is a URI reference, or the object (3), which is a URI reference, a literal (or a blank node). This means that only the RDF objects can contain literals as values. This hidden RDF feature turned out to be a disadvantage for the modeling purposes in the U2M approach. In the following section we will introduce extended resource identifiers that need to be represented as literals. With the chosen RDF structure of figure 4.14, we can handle this need. A list of real-life examples can be found online at the UbisWorld USERMODELSERVICE[13].

The content of SITUATIONALSTATEMENTS may be represented in any suitable language.

[13] `http://www.u2m.org/UbisWorld/UserModelServer.php?format=UserRDF`

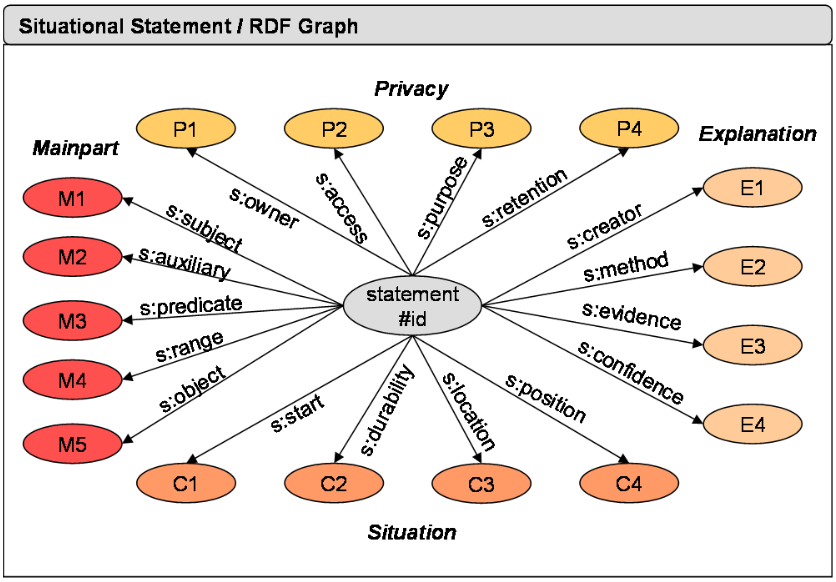


***Figure 4.13**: A* SITUATIONALSTATEMENT *schema in proper RDF/Graph without reification. The meta level attributes are arranged around the statement's identification*

However, RDF seems to be an ideal candidate for knowledge representation if inference purposes are also important. The newly developed SituationML/RDF application is closely related to the *Dublin Core Metadata Initiative* (as described in section 3.2.1 on page 37), and the *Composite Capabilities/Preference Profile Specification* (as described in section 3.2.4 on page 41). SituationML/RDF is like Dublin Core about meta data, but the focus is different. It is not documents in a network that should be annotated but partial situations. Some Dublin Core elements are so general that they can also be used to describe situations. However, since the intended meaning and usage could be different, the attributes are introduced independently. Furthermore they are structured into groups and added with algorithmic entailments.

4.4.4 Discussion: Why did we introduce SituationML?

The question that arises is: why did we introduce `SituationML` to represent situational information rather than using description logics, first-order logics or situation semantics? Almost every knowledge representation language is able to represent our intended situational information. However, in [Larkin and Simon, 1987] it is argued, that even when two representations are informationally equivalent, which means that they carry the same information, their computational efficiency can vary enormously. The efficiency depends on the information-processing operators that act on them, which in turn depends on the tasks that should be performed with the information. SITUATIONALSTATEMENTS identify the following three main tasks: 1) *human readability*, 2) *situated retrieval* and 3) *inferences on meta-*

```
<rdf:RDF
      xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax#
      xmlns:st=http://www.u2m.org/2003/situation#
      xml:base=http://www.u2m.org/statements>
<rdf:Description rdf:ID=a22>
             <st:subject> a1 </st:subject>
             <st:auxiliary> a2 </st:auxiliary>
             <st:predicate> a3 </st:predicate>
             <st:range> a4 </st:range>
             <st:object> a5 </st:object>
             <st:start> a6 </st:start>
             <st:end> a7 </st:end>
             <st:durability> a8 </st:durability>
             <st:location> a9 </st:location>
             <st:position> a10 </st:position>
             <st:source> a11 </st:source>
             <st:creator> a12 </st:creator>
             <st:method> a13 </st:method>
             <st:evidence> a14 </st:evidence>
             <st:confidence> a15 </st:confidence>
             <st:key> a16 </st:key>
             <st:owner> a17 </st:owner>
             <st:access> a18 </st:access>
             <st:purpose> a19 </st:purpose>
             <st:retention> a20 </st:retention>
             <st:id> a21 </st:id>
             <st:unique> a22 </st:unique>
             <st:replaces> a23 </st:replaces>
             <st:group> a24 </st:group>
             <st:notes> a25 </st:notes>
</rdf:Description>
</rdf:RDF>
```

Figure 4.14: *`SituationML/RDF_XML` defines each statement as an `rdf:Description` The variables a1 to a25 serve as placeholders for the actual content*

level information. Representations in logics are more *sentential* like the propositions in a text than *diagrammatic*. However, `SituationML` is more *diagrammatic* since it preserves with its layered approach explicitly the relation between the information about the meta data and the mainpart of the statements, which are important for the efficiency in the main tasks number 2 and 3. In [Larkin and Simon, 1987] it is also argued that human abilities to recognize information are highly sensitive to the exact form and representation in which the information is presented to the senses. Ease of recognition may be strongly affected by what information is explicit in a representation, and what is only implicit. We argue, that it is an advantage for the human readability if the meta-level information is represented together with the mainpart content itself and not distributed in sentential form over several propositions. Furthermore the representation of the semantic web resource locators within `SituationML` (the pointers to the ontology) has been simplified as demonstrated in the following section about the extended resource identification.

`SituationML` shares several ideas with the theory of situation semantics. The introduced concept of SITUATIONALSTATEMENTS forms a direct extension to the concept of infons, see section 2.2.3. However, the original framework of situation semantics is not used directly in our approach, since roles like ownership and privacy for example are not defined there. Furthermore, there exists no XML version or semantic web representation of situation semantics, which was one of our main design constraints.

The following section deals with the remaining question of the possible values and their representation of the variables from `a1` to `a25` as introduced in figure 4.10 in section 4.4 on page 69, or seen in figure 4.14.

4.5 Extended Resource Identification

This section looks at the issue of resource identification in RDF and extends the concepts of URIs especially for `SituationML`. *Mary* or *Peter* could for example be the `subject` of a user model statement. For an isolated, experimental blocks-world system `<subject>`Mary`</subject>` would be sufficient to represent the person Mary as pure literal. However, the question that arises is: *how can entities be referenced uniquely and efficiently in distributed real-world applications?* This question is especially important for the research task of UBIQUITOUS USER MODELING FOR SITUATED INTERACTION since globally unique identifiers are needed. Furthermore, these globally unique identifiers should still be human readable and secondly, short versions of these references should exist for efficiency reasons.

There are two approaches to introduce globally unique identifiers. The first one is the so-called GUID approach where more than 99,999% uniqueness is granted through the generation of a large random number that is calculated on the basis of the actual time in milliseconds together with a directed encryption method. The second one is the approach of uniform resource identifiers (URI) that was introduced in the internet, especially in the semantic web, see [Berners-Lee et al., 1998], [Lassila and Swick, 1999], [Champin, 2001] or section 2.3.1 for a detailed introduction. Both approaches to globally unique identifiers are used in `SituationML`: the statements receive GUIDs according to their time of appearance, while the ontological classes, their instances and the entities in the real-world are mapped to extended versions of URI references.

Finally, in the last subsection, the combination of unique identifiers into structured expressions is introduced. This so-called *UbisExpression* allows for statements about groups like *Mary and Peter*, for ontological entailments like class-extensions, as well as for abbreviations of long URIs. The interesting point about these new UbisExpressions is that they fulfill the requirements of proper RDF syntax because of our design decisions in chapter 4.4 where all `SituationML` attributes can be filled with RDF literals.

4.5.1 Extended URI References with Namespaces

`SituationML` is - like RDF - about describing resources. It can identify resources with qualified uniform resource identifiers. “Qualified URIs” are URIs with the slight difference that they can have an optional fragment identifier: a text added to the URI with a “#” between them. Resources in this sense can be considered as being the basic entities of a graph or conceptual mappings to entities. `SituationML` considers every qualified URI (with or without fragment identifier) as a full resource by itself. Since every qualified URI is unique, one could consider resources as linking unique identifiers to the referred objects and concepts. Figure 4.15 shows an example of two qualified URIs. The idea behind the qualification is that they allow parts of documents to be referenced as resources. The document `UserModelOntology` for example defines, among many others, the concept of *Cognitive Load*, (see chapter 5 for the ontology), and now this concept can be used as a resource, even if it is not a document by itself.

```
(1) http://www.u2m.org/UbisWorld#Peter
(2) http://www.u2m.org/UserModelOntology#CognitiveLoad
```

***Figure 4.15**: Two qualified uniform resource identifiers, (1) for a person in the context of UbisWorld with the name Peter, and (2) for the user modeling concept of cognitive load*

Although resources are not in the general focus of attention, they need to be analyzed in detail in order to keep the representation simple. Qualified URIs can be generally abbreviated by XML namespaces. However, this namespace abbreviation is only possible in the RDF tags but not within the RDF content elements. Thus the two qualified uniform resource identifiers with namespaces in figure 4.16 cannot be used in `SituationML/RDF`.

```
(1) u2m:UbisWorld#Peter
(2) u2m:UserModelOntology#CognitiveLoad
```

***Figure 4.16**: Two qualified uniform resource identifiers with the namespace `u2m:`*

4.5.2 UbisIdentifier (Ubid)

For the sake of clarity, one would prefer to use non-expanded notation like the ontology prefixes `Gumo:` and `UbisWorld:`, instead of `http://www.u2m.org/Gumo#` and `http://www.u2m.org/UbisWorld#` respectively, as shown in figure 4.17.

```
(1) UbisWorld:Peter
(2) Gumo:CognitiveLoad
```

***Figure 4.17**: Two resource identifiers with the prefixes UbisWorld: and Gumo:*

These resource identifiers are pure literals and no longer URIs. They are not yet standardized but they are well formed for human readers (in comparison to figure 4.15), they contain the same information as the full URI references and could thus be preprocessed into ordinary URIs. The idea of namespaces could be used within RDF content elements with standardized preprocessing.

A second problem that occurs in real-world naming is the fact that many entities have the same name. Several people could have the same first name or family name. Thus, information for disambiguation has to be added to the identifier, independently from the namespace. In large emailing systems, a counter is often added as a suffix to the identifying name. A second example is, if an ontology defines the two concepts of the animal *mouse*, and the computing device *mouse* in one document, one has to distinguish between these two meanings. In the WordNet[14] research approach, see section 3.3.3 for a description, disambiguation is also realized by adding a counter to the semantical concept, but only to those concept names that show different meanings or semantical readings.

Bearing in mind the need for human readability and machine-processable names, we introduced the following new concept: We add a six digit number[15] to every concept and entity name, together with a separating dot between the identifier's name and the identifier's number. With this syntactical convention, we can mostly omit the namespaces but it is still possible to distinguish between simple literals in parallel, and cover the problem of disambiguation-numbering in one step. Figure 4.18 shows two, so-called *UbisIdentifiers*.

```
(1) Peter.210004
(2) CognitiveLoad.340050
```

Figure 4.18: *Two UbisIdentifiers with a separating dot and exactly six digits*

These identifiers can omit the ontology denoting namespace because they refer per default to the *UbisWorldOntology*, which is defined in chapter 5. External ontologies can of course also be used, however mostly not with UbisIdentifiers, but with ordinary uniform resource identifiers. If external ontologies also use the same naming convention as UbisIdentifiers, together with the same ranges of numbers, then the concept of namespaces had to be applied. UbisIdentifiers can be seen as abbreviated references to ordinary URIs. After the preprocessing step, `Peter.210004` will be mapped to `http://www.u2m.org/UbisWorld#Peter`, or if the number is needed for disambiguation, to `http://www.u2m.org/UbisWorld#Peter.210004`.

Figure 4.19 shows seven different possibilities to denote a reference to the person Peter. All seven versions[16] have their advantages and disadvantages and will be used for different purposes.

[14] WordNet homepage: `http://www.cogsci.princeton.edu/~wn/`

[15] Restricting UbisIdentifiers to six digits allows the ontology to carry up to 1 million concepts, which seemed to be sufficient for prototyping aspects. A debatable advantage of six-digit-numbers is that they can be read out and remembered easily as two three-digit-numbers: for example 204109 could be read as: *two-hundred-four, one-hundred-nine*

[16] The second reference will be defined in the EBNF grammar in figure 4.20

```
(1) Peter
(2) ubis(Peter)
(3) Peter.210004
(4) UbisWorld:Peter
(5) http://www.u2m.org/UbisWorld#Peter
(6) http://www.u2m.org/UbisWorld#Peter.210004
(7) map(http://www.anydomain.org/anyOntology#Pepe)
```

Figure 4.19: *Seven Possible identifiers to refer to the entity "Peter": (1) purely literal, not pointing to an ontology, (2) abbreviated UbisIdentifier without the UbisId part, possibly pointing to several entities, (3) unique UbisIdentifier, (4) namespace-abbreviated URI, (5) UbisWorld URI, (6) UbisWorld URI with disambiguating UbisId, (7) mapping from an external ontology*

4.5.3 UbisExpression (Ubex)

Now, the combination and grouping of identifiers is analyzed. Instead of representing only one resource reference per attribute, we allow a whole list of resource identifiers. If *Mary and Peter play together*, for example, both should form the subject of one statement `<subject> Mary.210005 | Peter.210004 </subject>`, rather than using two statements. This contributes to the actual research issue of group modeling additionally to single user modeling. Alternatively, one could introduce and identify the group of Mary and Peter as one referenceable unit, or as done in RDF, introduce an unspecified "blank node" that connects Mary and Peter to one group without introducing a new concept. However, it turned out that the introduction of the list structure on the level of attribute values and the level of identifiers forms an expressive element of syntactic sugar. Instead of using commas ',' to indicate a new list element, vertical lines '|' are used as separators.

Once we have implemented preprocessor functionality into the basic syntactic structure of SITUATIONALSTATEMENTS and introduced the list structure, we can go further and bring - for example - ontological inferences down to the level of statements. Instead of naming all instances (A-Box) like *Mary* and *Peter* that correspond to a concept (T-Box) like *student*, we could wish to write `<subject> extension(` *concept* `) </subject>` and simulate "for-all statements". This small, but fundamental step integrates the semantic web ontology with the syntactical representation level, not only by referencing with pure URIs like RDF does, but with the ontological extension functionality. Other ontology functions like `<subject> ancestors(` *concept* `) </subject>`, have been implemented.

A third request for the design of UbisExpressions came from the research area of ontology merging. Since the `SituationML` approach allows any ontology to be referred to, we have to deal with the fact that several ontologies are used simultaneously by different systems, which leads to same concepts with different names on the one hand and same instances or individuals with different names on the other hand. At this syntactical level of identifier-preprocessing, a mapping function fits perfectly into the evaluation of UbisExpressions: `<subject> map(` *Identifier* `) </subject>`

Figure 4.20 shows the grammar that defines UbisExpressions in extended Backus Naur format in the version of [Scowen, 1998]. Nonterminals are not marked but terminals are written between two apostrophes. Underspecified nonterminals are written between quotation marks.

```
UbisExpression  -->  UbisExpression '|' UbisExpression
                 |   UbisFunction
                 |   UbisElement

UbisFunction    -->  'extension(' UbisIdentifier ')'
                 |   'ancestors(' UbisIdentifier ')'
                 |   'ubis(' UbisName ')'
                 |   'map(' UbisElement ')

UbisList        -->  UbisList '|' UbisList
                 |   UbisElement

UbisElement     -->  UbisIdentifier
                 |   "RDF URI reference without '|'"
                 |   "RDF Literal without '|'"

UbisIdentifier  -->  UbisName '.' UbisId
                 |   UbisId

UbisName        -->  "String not starting with a digit and without '|'"

UbisId          -->  "Number with exactly 6 digits"
```

***Figure 4.20**: The EBNF grammar of UbisExpression that defines the vocabulary and strings that can be interpreted or used as values for statement attributes*

The two new concepts that were not mentioned so far in this section are `UbisElement` and the `ubis(` *UbisName* `)` concept. The first one states that apart from UbisIdentifiers, ordinary RDF URI references and RDF literals without the special character | are also UbisExpressions. Thus `SituationML/RDF` is a superset of RDF "modulo |". However, since statements with the list constructor | can be transformed into several statements without lists, every `SituationML/RDF` document can be translated into an ordinary RDF document. The second concept states that the transformation of a UbisName - without the UbisId - into one UbisIdentifier or into a list of UbisIdentifiers is offered as another abbreviation service. This functionality is important in order to distinguish between abbreviated UbisIdenfiers and general RDF literals.

To summarize, *UbisExpression* is an extension of RDF node elements that makes it possible to abbreviate URIs into smaller and more readable UbisIdentifiers, which represent an entity within the UbisWorld ontology. Technically speaking, a preprocessor maps all kinds of abbreviations into lists of URIs and literals. Furthermore, by this extension the gap between ontological reasoning and pure knowledge representation has been filled with the help of *UbisFunctions*. Thus the variables from `a1` to `a25` as introduced in figure 4.10 in section 4.4 can carry complex preprocessable - but still ordinary - RDF literals or URIs as values.

To conclude the section about the syntax analysis of SITUATIONALSTATEMENTS and `SituationML`, an example statement with UbisIdentifiers is presented in figure 4.21.

```
<rdf:Description rdf:ID="statement1128">
  <st:subject>FinePixA202.230060</st:subject>
  <st:auxiliary>HasProperty.600100</st:auxiliary>
  <st:predicate>Price.820035</st:predicate>
  <st:range>Text.640002</st:range>
  <st:object>199 Euro</st:object>
  <st:start>2004-01-01T00:00:00</st:start>
  <st:durability>Month.520050</st:durability>
  <st:creator>Anonym.210020</st:creator>
  <st:method>ManualChangeWithEditor.910040</st:method>
  <st:confidence>0.99</st:confidence>
  <st:access>Public.640121</st:access>
  <st:purpose>Commercial.640131|Research.640132</st:purpose>
  <st:retention>Month.640142</st:retention>
</rdf:Description>
```

***Figure 4.21**: Example statement in `SituationML/RDF` with UbisIdentifiers and literals with the intended meaning that the camera FinePix A202 has the actual price of 199 Euro.*

Most attributes in this example statement carry UbisIdentifiers in their unabbreviated version as values, while the `<st:object>` and the `<st:start>` carry literals as values. The `<st:purpose>` attribute contains a list of two values as a UbisExpression.

4.5.4 UbisList (Ubli)

A UbisEpression without any UbisFunction is called a UbisElementList, or short UbisList. Every UbisExpression can be mapped to a UbisList that consists of a |-separated (read: pipe-separated) list of UbisIdentifiers, URI references or RDF literals respectively. Figure 4.20 contains the corresponding part of the EBNF grammar.

Compared to RDF node values, UbisList has the advantage of abbreviating blank nodes with multi child nodes and it allows abbreviated identifiers in addition to the uniform resource identifiers without namespace abbreviation.

5 USER MODEL ONTOLOGY & UBISWORLD

Ontologies provide a shared and common understanding of a domain. As pointed out in [Fensel, 2001], they can be communicated between people and widely spread application systems. Since ontologies have been developed and investigated in artificial intelligence to facilitate knowledge sharing and reuse, they should form the central point of interest for the intended goal task of exchanging user models. The three fundamental properties of information according to [Castel, 2002] are:

- it is *functional*, which means that the information has its purpose
- it is *artificial*, that is to say it is man-made or machine-made
- it is *designed*, which means that it is created through specific choices

and because of these fundamental properties ontologies differ in their appearance. Furthermore, specialized tasks call for specialized ontologies. This chapter contains three sections. The first one develops the OWL application `UserOL` for the uniform management of user models in a semantic web. The design choices in this approach are described there. The second one describes the so-called general user model ontology GUMO and the third one presents UbisWorld and UBISONTOLOGY. The integration of ontologies and the statements is presented in section 7.1.3.

5.1 Preparatory Work

In this section we argue, why we have chosen the web ontology language OWL for the ontology application `UserOL`. We present three concept definitions, namely the user model class *Physiological State*, the user model dimension *Happiness* and the user model auxiliary *has Knowledge*. Finally, we describe some elements of the OWL application `UserOL`.

5.1.1 Choosing OWL as Ontology Language

XML is designed to serve for weakly structured data as an interchange format by defining the underlying data model in a schema and using annotations from the schema. The goal is to relate parts of the information to the schema specification. However, XML is purely syntactic and structural in nature. In [Stuckenschmidt and van Harmelen, 2005] it is pointed out that, if we wanted to describe information on the meta-level and to define its meaning, we have to look for further approaches and the RDF standard has been proposed as a data model for representing meta data. See 2.3.2 for an introduction to RDF. RDF is domain independent and no assumptions about a particular domain, especially the domain of user models, are given. The corresponding schema language to define the vocabularies is called RDF Schema, or RDFS, see e.g. [Manola and Miller, 2004]. In RDFS one can define which properties apply to which kinds of objects and what value ranges hold. Furthermore one can describe the relationships between objects.

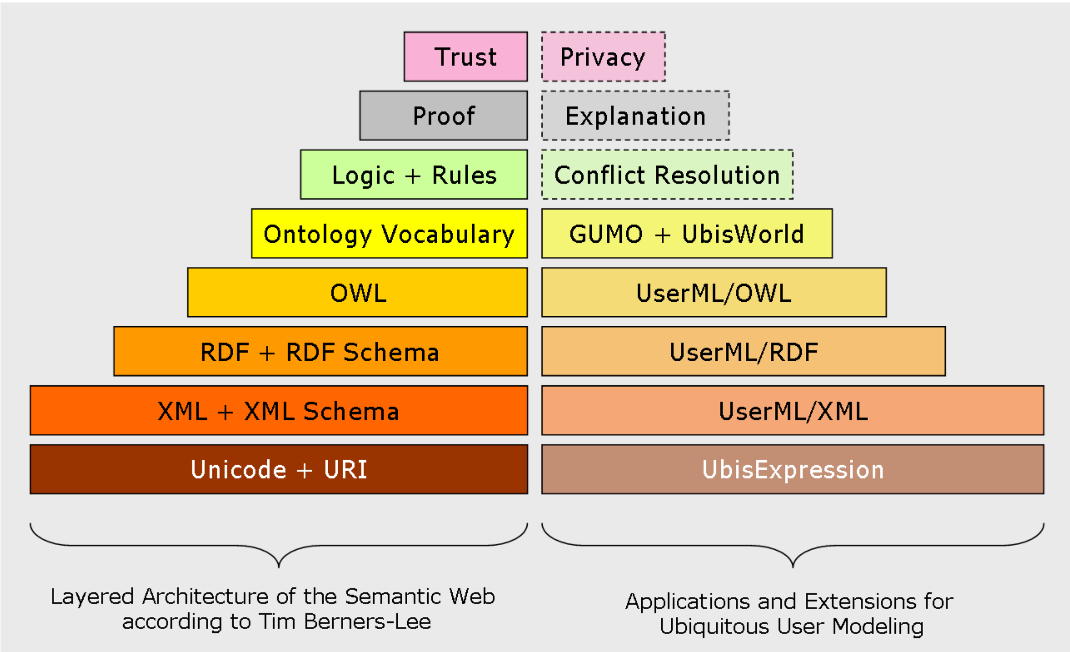


Figure 5.1: *Confrontation of the semantic web layers according to [Fensel et al., 2003] or [Berners-Lee and Fischetti, 1999] with the applications and extensions for ubiquitous user modeling*

The web ontology language OWL however has more facilities for expressing semantics, see section 2.3.3, and [Stuckenschmidt and van Harmelen, 2005], than XML, RDF, and RDFS. OWL adds more vocabulary for describing properties and classes and among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL can especially be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. OWL is a revision of the DAML+OIL web ontology language, in which the first user model ontology was defined, see [Heckmann, 2003d] for an online listing of this ontology. Figure 5.1 shows the layered architecture of the semantic web according to [Berners-Lee and Fischetti, 1999] (with OWL playing a central role), accompanied by the newly defined applications and extensions as described throughout this thesis.

To summarize, OWL is our choice for further representation of user model terms and their interrelationships into an ontology, while at the same time RDFS, RDF and XML are integrated into the OWL web ontology language and thus used implicitly. The general user model ontology GUMO is defined as the OWL application `UserOL`.

5.1.2 Example Definitions in `UserOL`

Figure 5.2 presents as a first example the concept of the user model dimension class *Physiological State* which is realized as a `owl:Class`. A class defines a group of individuals that belong together because they share some properties. Classes can be organized in a specialization hierarchy using `subClassOf`. There is a built-in most general class named `Thing` that is the class of all individuals and a superclass of all OWL classes. The *Physiological State* is defined as a subclass of *Basic User Dimensions*.

```
<owl:Class rdf:ID="PhysiologicalState.700016">
    <rdfs:label> Physiological State </rdfs:label>
    <u2m:identifier> 700016 </u2m:identifier>
    <u2m:lexicon>the state of the body or bodily functions</u2m:lexicon>
    <u2m:website rdf:resource="&UserOL;concept=700016" />
    <rdfs:subClassOf rdf:resource="#BasicUserDimensions.700002" />
</owl:Class>
```

***Figure 5.2**: The OWL class definition of "Physiological State"*

Every new concept has a unique `rdf:ID`, that can be resolved into a complete URI, as discussed in section 4.5. The identification number `u2m:identifier` in this case is 700016, it has been chosen arbitrarily but seen under its namespace, it is unique. It has the advantage of freeing the textual part in the `rdf:ID` from the need of being semantically unique, as argued in the last section about UbisExpressions. Apart from solving the problem of conceptual ambiguity, this number facilitates the work within relational databases, which is important from the implementation point of view. Figure 5.2 also defines the lexical entry `u2m:lexicon` of the concept of *Physiological State* as "the state of the body or bodily functions", while this textual definition could also be realized through a link to an external lexicon. The attribute `u2m:website` points towards a web site that presents this ontology concept to a human reader. The abbreviation `&UserOL;` is a shortcut for the complete URL to the general user model ontology GUMO.

Figure 5.3 defines the user model dimension *Happiness* as an `rdf:Description`. It contains a `rdfs:label`, a `u2m:identifier` and a `u2m:website` attribute. Additionally it provides a default value of the average expiry `u2m:expiry`. It carries the qualitative

```
<rdf:Description rdf:ID="Happiness.800616">
    <rdfs:label> Happiness </rdfs:label>
    <u2m:identifier> 800616 </u2m:identifier>
    <u2m:expiry> minutes.520050 </u2m:expiry>
    <u2m:privacy> medium.640032 </u2m:privacy>
    <u2m:image rdf:resource="http://u2m.org/UbisWorld/img/happiness.gif" />
    <u2m:website rdf:resource="&UserOL;concept=800616" />
    <rdf:type rdf:resource="#EmotionalState.700014" />
    <rdf:type rdf:resource="#FiveBasicEmotions.700015" />
</rdf:Description>
```

***Figure 5.3**: UserOL definition of "Happiness"*

time span of how long the statement is expected to be valid (like minutes, hours, days, years). In most cases when user model dimensions or context dimensions are measured, one has a rough idea about the expected durability, for instance, emotional states change normally within hours, however personality traits won't change within months. Since this qualitative time span is dependent from every user model dimension, a definition mechanism is prepared within the `UserOL`. Some examples of rough durability-classifications, without any attempt of proven correctness, are:

- physiologicalState.heartbeat - can change within seconds
- mentalState.timePressure - can change within minutes
- emotionalState.happiness - can change within tens of minutes
- characteristics.inventive - can change within months
- personality.introvert - can change within years
- demographics.birthplace - can't normally change at all

Another important point that is shown in the definition of *happiness* in figure 5.3 is the capacity in OWL for multiple-inheritance. In detail, happiness is defined as `rdf:type` of the class *Emotional State* as well as `rdf:type` of the class *Five Basic Emotions*. Thus OWL allows to construct complex, graph like hierarchies of user model concepts, which is especially important for ontology integration.

Figure 5.4 defines the auxiliary *has Knowledge* as `rdfs:subPropertyOf` of the resource *user model auxiliary* with the `rdf:domain` *#Person*, which is not part of the user model ontology itself, but which is part of the general UbisWorld Ontology, see section 5.3 or [Stahl and Heckmann, 2004b].

5.1.3 The U2M Namespace Elements

The acronym `u2m` stands for *ubiquitous user modeling* and forms a collection of standards, that are online available at `http://www.u2m.org/`. The new vocabulary for the `UserOL` language consists of

```
<rdf:Property rdf:about="hasKnowledge.600120">
    <rdfs:label> has Knowledge </rdfs:label>
    <u2m:identifier> 600120 </u2m:identifier>
    <u2m:expiry> year.520020 </u2m:expiry>
    <u2m:privacy> high.640033 </u2m:privacy>
    <u2m:website rdf:resource="&UserOL;concept=600120" />
    <rdfs:domain rdf:resource="#Person.110003" />
    <rdfs:subPropertyOf rdf:resource="#UserModelAuxiliary.600020" />
</rdf:Property>
```

Figure 5.4*: UserOL Property hasKnowledge as example for general auxiliaries*

- `u2m:identifier`, which is a special identification number
- `u2m:expiry`, which presets the expected expiration time
- `u2m:privacy`, which describes the predefined general privacy level
- `u2m:lexicon`, which contains a lexical description of the concept
- `u2m:website`, which points towards a virtual representative of the concept
- `u2m:image`, which points towards an image, graph or diagram of the concept

This vocabulary is used to enrich the modeling of the general user model ontology GUMO and the UBISWORLD ONTOLOGY. Both are introduced in the two following sections. In order to support the distributed construction and refinement of the top level user model ontology and the UbisWord ontology, we developed a specialized online ontology editor, see section 8.2.2, that helps with introducing new concepts, adding their definitions and transforming the information automatically into the required semantic web ontology language[1].

5.2 GUMO - the General User Model Ontology

A commonly accepted top level ontology for user models could be of great importance for the user modeling research community as argued in [Heckmann et al., 2005a]. This ontology should be represented in a modern semantic web language like OWL or DAML+OIL, see [Fensel et al., 2003] for example, and thus be available via internet for all user-adaptive systems at the same time. The major advantage would be the simplification for exchanging user model data between different user-adaptive systems. The current problem of syntactical and structural differences between existing user modeling systems could be overcome with a commonly accepted taxonomy, specialized for user modeling tasks. Note, that we are talking about a user model ontology rather than a user model*ing* ontology[2], which would include, the inference techniques, or knowledge about the research area in general. We are analyzing the user's dimensions that are modeled within user-adaptive systems like

[1] Currently supported web ontology languages are: Instance-OIL, DAML+OIL, RDF and OWL

[2] A recent approach towards a "User Modeling Meta-ontology" has been started by [Yudelson et al., 2005]

the *user's heart beat*, the *user's age*, the *user's current position*, the *user's birthplace* or the *user's ability to swim*. Furthermore, the modeling of the users‘ interests like *reading poems*, or *playing adventure games* or *drinking expensive French Bordeaux wine* is analyzed.

The main conceptual idea in the approach of SITUATIONALSTATEMENTS that influences the construction of the intended general user model ontology GUMO is to divide the descriptions of user model dimensions into the three parts `auxiliary`, `predicate` and `range` as shown in figure 5.5 and discussed in section 4.2.1 on page 59.

$$\texttt{subject}\left\{\ \textit{user model dimension}\ \right\}\texttt{object}$$
$$\Updownarrow$$
$$\texttt{subject}\left\{\begin{array}{c}\texttt{auxiliary}\\ \texttt{predicate}\\ \texttt{range}\end{array}\right\}\texttt{object}$$

***Figure 5.5**: User model dimensions within* SITUATIONALSTATEMENTS

For example if one wants to say *something about the user's interest in football*, one could divide this so-called *user model dimension* into the auxiliary part "has interest", the category part "football" and the range part "low-medium-high" as shown in the list below.

$$\texttt{subject}\left\{\begin{array}{rcl}\texttt{auxiliary} & = & \text{has interest}\\ \texttt{predicate} & = & \text{football}\\ \texttt{range} & = & \text{low-medium-high}\end{array}\right\}\texttt{object}$$

If a system wants to express something like *the user's knowledge about Beethoven's Symphonies*, one could divide this into the triple "has knowledge", "Beethoven's Symphonies" and "poor-average-good-excellent":

$$\texttt{subject}\left\{\begin{array}{rcl}\texttt{auxiliary} & = & \text{has knowledge}\\ \texttt{predicate} & = & \text{Beethoven's symphonies}\\ \texttt{range} & = & \text{poor-average-good-excellent}\end{array}\right\}\texttt{object}$$

As a third example, information about *the user's hair-color* would lead to "has property", "hair-color" and "red-brown-blonde-white-black":

$$\texttt{subject}\left\{\begin{array}{rcl}\texttt{auxiliary} & = & \text{has property}\\ \texttt{predicate} & = & \text{hair-colour}\\ \texttt{range} & = & \text{hair-colour-range}\end{array}\right\}\texttt{object}$$

The implication for the general user model ontology GUMO of these examples above is the clear separation in the modeling between user model auxiliaries, predicate classes and special ranges. What leads to a tricky problem is that actually everything can be a predicate if the auxiliary is "interest" or "knowledge".

5.2.1 User Model Auxiliaries

First of all, the group of auxiliaries has to be identified. Table 5.1 shows a list of several, identified, important user model auxiliaries together with their attached UbisIds.

***Table 5.1**: List of User Model Auxiliaries*

Group	Name	Id
UserModelAuxiliary	has Property	600100
UserModelAuxiliary	has Interest	600110
UserModelAuxiliary	has Believe	600120
UserModelAuxiliary	has Knowledge	600130
UserModelAuxiliary	has Preference	600140
UserModelAuxiliary	has Regularity	600150
UserModelAuxiliary	has Plan	600160
UserModelAuxiliary	has Goal	600170
UserModelAuxiliary	has Done	600180
UserModelAuxiliary	has Location	600190

This listing is not intended to be complete but it is a start with which most of the important user model statements can be realized. The auxiliary `has Property` leads to the more user centric predicates that are described in the next section with the name *Basic User Dimensions*. The auxiliaries `has Interest` and `has Preference` lead mainly to the *User Model Interest Categories* like music categories or film genres which are introduced in section 5.14. The auxiliaries `has Regularity` and `has Done` lead to the so-called *Usage Data* as defined in [Kobsa, 2001a] and the *Low Level Sensor Data*. The auxiliary `has Location` leads to a spatial ontology as described in section 5.3.2 on page 105.

However, as stated above, it turned out that actually any concept in the whole world can be needed to express user model data. The crucial design decision is to leave this part open for existing other ontologies like the general SUMO/MILO ontologies or the Cyc ontology as described in section 3.3 about external ontologies and knowledge bases. This in turn leads to a modular approach which forms a key feature rather than a disadvantage. Furthermore a so-called UBISWORLD ONTOLOGY has been developed to fill the gaps of missing concepts in external ontologies and to enable fast prototyping, see section 5.3 or [Stahl and Heckmann, 2004b]. Nevertheless, the attempt of defining a commonly accepted, specialized top level ontology for the user modeling research is presented in the following section.

5.2.2 Basic User Model Dimensions

An overview of modeled properties in user-adaptive systems is presented in [Jameson, 2001a] and [Jameson, 2001c] together with the analysis about the *breadth of implications*, *directness of decision-relevance* and *ease of assessment*. Some user model dimensions can typically be observed by the system directly, some user model dimensions may require additional acquisition and inference steps, again others are entered into the system by self-reports by the users. In [Kobsa et al., 2001] an analysis about the *input data for personalized systems* is presented which discusses the different kinds of data that user-adaptive systems may need to consider. The classifying terms *user data* and *usage data* are introduced. However, it is also shown that there are potential overlaps between these two categories. The user data is further distinguished into *demographic data*, *user knowledge*, *user skills and capabilities*, *user interests and preferences* and *user goals and plans*.

The categorization in the general user model ontology GUMO approach, as shown in figure 5.6, is based on Jameson's and Kobsa's work; however, it is different, in that the modeling of the dimensions can now be split into `auxiliaries`, `predicates` and `ranges` which leads to a more powerful interplay between the SITUATIONALSTATEMENTS and the ontology, as argued in the sections above. Furthermore, no top level user model ontology for the semantic web has been proposed so far, apart from [Heckmann, 2003d]. Figure 5.6 classifies several groups of user dimensions that can be identified. The complete list of the modeled basic user dimensions together with their UbisIds are presented in the appendix table B.2 on page 223.

Figure 5.6*: Groups of basic user dimensions as shown in the Ubis Ontology Browser*

Since there is no consensus in the nature or meaning of *affect*, existing theories and models of personality, mood, emotion and facial expressions are discussed individually in the following subsections without classifying them under the term "affect" in the ontology.

5.2.2.1 Contact Information and Demographics

Contact information is available to most commercial systems, since the user has to register their record data. Some of them are mostly compulsory to be entered while others are optional. Even these basic bits of user information like `given name`, `postal code` or `country` as shown in figure 5.7(a) are good for user-adaptation. There exist commercial databases that map the `postal code` together with the `street` and sometimes even the `house number` to the assumed wealth and living standard of the inhabitants. Knowing the `city` of living makes it possible to offer regional awareness. The `country` refers to the official spoken languages. Even the assumption, that old-fashioned `given names` could refer to elderly people is used in some commercial applications.

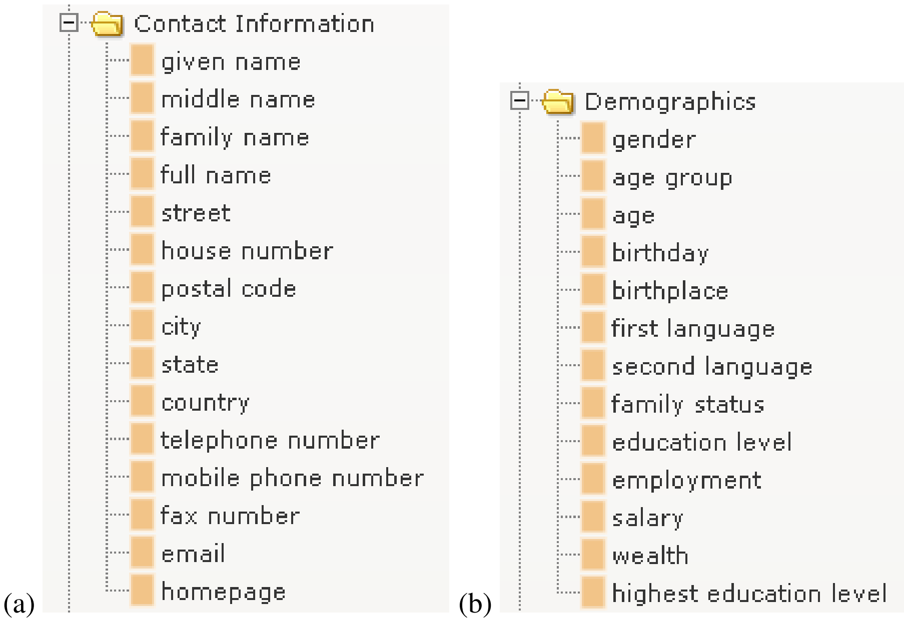


Figure 5.7: *Contact information and demographics as shown in the Ubis Ontology Browser*

User demographics like `gender` and `age` as shown in figure 5.7(b) are very important user model dimensions that are for example inferred from speaking behavior, see [Müller et al., 2001]. `Second languages`, `family status`, `highest education level`, `employment` and `salary` also belong to the demographics group. Today's personalized websites often operate on the basis of these data and according to [Kobsa et al., 2001] the value can be high when combined with high-quality statistical data, such as about the purchase behavior of different customer groups.

5.2.2.2 Personality and Characteristics

`Personality` and `Characteristics` describe permanent or very slow changing patterns that are associated with an individual. They might be difficult to detect by a user adaptive system but once they are known they can be permanently used. There exist several models of personality in the psychological literature. Not all of them can be considered for the basic user model ontology. However, since the ontology should not stick to one model only, several models have been analyzed and are discussed in the following. This section is based on [Klesen, 2002], [Oberlander, 2004], [Picard, 1997] and [André et al., 2000].

A predominant model for personality is according to [Klesen, 2002] the *Five-*

Factor Model or *OCEAN Model*, see [McCrea and John, 1992] for an introduction. It is a descriptive model with the five factors `Extraversion`, `Agreeableness`, `Conscienciousness`, `Neuroticism` and `Openness`. They can for example be derived from a factor analysis of a large number of reports on personality relevant adjectives. The *Extraversion-Energy-Enthusiasm* factor refers to the number of relationships with which one is comfortable. High extraverts tend to be more physically and verbally active and to be more friendly and outgoing around others. Low extraverts (or Introverts) tend to be more independent, reserved, planning ahead, steady and more comfortable with being alone. The *Agreeableness-Altruism-Affection* factor refers to the number of sources from which one takes one's norms for right behavior. High agreeableness describes a person who obeys many norms. Low agreeableness describes a person who rather follows the inner voice only. The *Conscientiousness-Control-Constraint* factor refers to the number of goals on which one is focused. High conscientiousness refers to a person who focuses on fewer goals and exhibits the self-discipline associated with such focus. The *Neuroticism-NegativeAffectivity-Nervousness* factor refers to the number and strength of stimuli required to elicit negative affectivity in a person. Low neurotic persons are bothered by fewer stimuli in their environment. Furthermore, the stimuli must be very strong in order to bother them. The fifth factor, the *Openness-Originality-OpenMindedness* factor, refers to the number of interests to which one is attracted and the depth to which those interests are pursued. High openness refers to a person with many interests and probably less depth within each interest.

The second major trait theory of personality is the *three factor model* or *PEN model*[3] of [Eysenck and Eysenck, 1991] as mentioned in [Oberlander and Gill, 2004]. It has a descriptive and causal aspect. The three factors are *Psychoticism*, *Extraversion* and *Neuroticism*. Interesting for affective computing with the use of bio sensors is, that the PEN model is biologically based, see `http://www.personalityresearch.org/`

- Extraversion for example is based on cortical arousal. Arousal can be measured by `skin conductance`, `brain waves`, or `sweating`. While theoretically introverts are chronically overaroused, theoretically extraverts are chronically underaroused and bored.
- Neuroticism is based on activation thresholds in the sympathetic nervous system or visceral brain[4]. Activation can be measured by `heart rate`, `blood pressure`, `cold hands`, `sweating`, and `muscular tension`. These causal indications are modeled as *low level sensor data* in the general user model ontology GUMO. Neurotic people, who have a low activation threshold, experience negative affect already in the face of minor stressors which means that they are easily upset. Emotionally stable people, who have a high activation threshold, experience negative affect only in the face of very major stressors which means that they stay calm under pressure.
- Psychoticism is associated not only with the liability to have a psychotic episode, but also with aggression. The biological basis for Psychoticism can be found in increased `testosterone levels`.

[3] PEN model homepage: `http://www.personalityresearch.org/pen.html`

[4] Visceral brain is the part of the brain that is responsible for the fight-or-flight response in the face of danger

In [Matthews et al., 2003], chapter 7, several reasons for linking personality traits to neural systems are given. The hypothesis that personality is an expression of individual differences in brain function and the challenge to develop neuropsychological theories of personality traits is discussed there. The theory of *Personality Types* is the third theory that has been recognized for the development of the personality dimension in the general user model ontology GUMO. The MyersBriggs type inventory as shown in figure 5.8(a) contains four dimensions: `Extravert versus Introvert`, `Sensor versus Intuiter`, `Thinker versus Feeler` and `Judger versus Perceiver`. The possible combinations of these four dimensions form sixteen different personality types if no continuous values are used. Every user is said to have a natural preference that falls into one category or the other in each of these four dimension. In [Klesen, 2002], the *Inventor Type* is defined by "extraverted-intuitive-thinking-perceiving" and the *Mastermind Type* is defined as "introverted-intuitive-thinking-judging".

Figure 5.8(a) shows that the user's personality dimensions appear several times within the general user model ontology GUMO, namely within their personality models but also standing along. This offers maximal possibilities to express statements about personality. The range for the personality dimension like `Extravert` could be *low-medium-high*, the continuous interval [0,1] or any other range that suits. In the case of bipolar dimensions as in `Introvert versus Extravert` the discrete set {-3, -2, -1, 0, 1, 2, 3} could be used for example. Interesting symptomatic indicators for the personality traits extraversion and neuroticism are for example analyzed for the natural language domain in [Pennebaker and King, 1999], [Dewaele and Furnham, 2000] and [Gil and Oberlander, 2002].

Figure 5.8(b) shows a list of `Characteristics` like `talkative`, `assertive`, `dominant`, `quiet`, `reserved` and `shy`. These long-term user properties are interrelated with the personality dimensions in such a way that causal dependencies can be identified in both directions. The *Interpersonal Theory*[5] for example deals with people's characteristic interaction patterns, which vary along the dimensions of dominance and friendliness. The Interpersonal theory comprises three strands of leading ideas: the principle of complementarity, the principle of vector length, and the principle of circumplex structure. Another interesting theory is the *Balance Theory* by [Heider, 1958] that combines cognitive elements into cognitive systems to explain interpersonal relations. [Schmitt, 2005] extends this theory and develops a dynamic attitude model that simulates changes in interpersonal relationships as a result of communication processes.

Figure 5.9 shows the so-called *circumplex* of the interpersonal theory. All three principles can be expressed with the combination of SITUATIONALSTATEMENTS and the general user model ontology GUMO via the `predicate` and `range` attributes.

According to [Jameson, 2001a] inferring personal characteristics on the basis of indirect evidence is in general difficult. However, the necessary information is often available or easily supplied by the user himself, for example in self-reports.

[5]Interpersonal Theory: `http://www.personalityresearch.org/interpersonal.html`

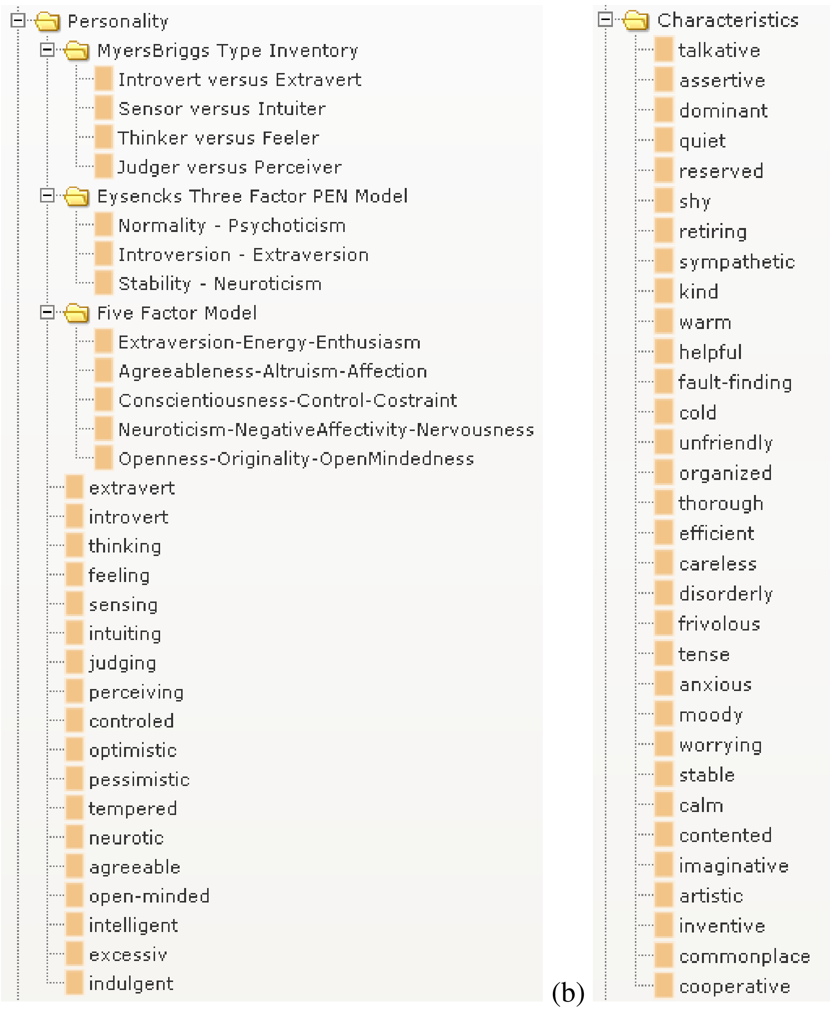
(a)


***Figure 5.8**: (a) Personality and (b) Characteristics in the general user model ontology* GUMO

5.2.2.3 Mood, Emotion and Facial Expression

`Mood` and `emotion` differ from personality and characteristics in respect to the user model dimensions in terms of duration. Emotions tend to be closely associated with a specific event or object and have a short duration of a few minutes up to an hour, whereas mood is more diffuse and of longer duration between a few hours and a few days. The problem of representing and reasoning with affective states and personality traits is analyzed in the research about embodied animated agents, see e.g. [André et al., 2000], [Klesen, 2002] and [Prendinger and Ishizuka, 2004] as well as in the research of human performance simulation, see [Silverman et al., 2003]. Frameworks for the generation of emotions in life-like agents are presented in [Lee, 1999], [Gebhard et al., 2004] and [Schmitt, 2005]. Emotional states are for example `happiness`, `anxiety`, `fear`, `love`, `hate` and `pride`. The list of currently modeled emotional states in the general user model ontology GUMO can be found

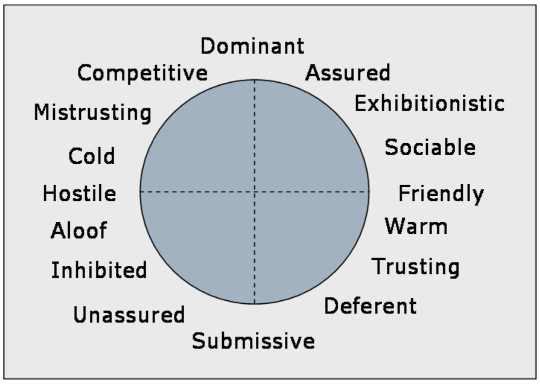


Figure 5.9: *The interpersonal circumplex according to Timothy Leary (1957) and Don Kiesler (1987), see [Acton, 1997]*

in figure 5.10(b). Mood as shown in figure 5.10(a) can be mapped to the compact values `good mood` and `bad mood` or to the Mehrabian temperament space as discussed below.

The *Cognitive Appraisal Theory* or *OCC Model* from Ortony, Clore and Collins [Ortony et al., 1988] has become popular in affective computing since it has a strong theoretical background on explaining the generation of emotions and computing their intensities. It converts an emotion eliciting situation by interpreting the situation in the cognitive level. Thus the generated emotion depends on the result of the interpretation of a given situation. According to [Gebhard et al., 2004] emotions are defined as appraised reactions to events, actions and objects: *event-based emotions*, desirability of events with respect to the agent‘s goals; *agent-based emotions*, praiseworthiness of actions with respect to a set of standards; *object-based emotions*, appealingness of objects with respect to a set of attitudes.

In [Morris, 1992] the distinction between `mood` and `emotion` is discussed. Moods are said to be concerned with larger, longer lasting, existential issues about the person's life and how it is going, while emotions are apt to be brief. Most moods do not seem to be clearly related to a single event, action or object as is the case of emotions. Even though such a distinction is not widely accepted in philosophy and psychology, it is interesting for the general user model ontology GUMO since an integrated model has most recently been developed by [Gebhard et al., 2004]. The *Pleasure-Arousal-Dominance (PAD) Temperament Framework*[6] that was introduced by [Mehrabian, 1996] is used there to represent moods. It consists of three dimensions of emotional states, namely `pleasure versus displeasure` *(+P and -P)* that distinguishes the positive-negative affective quality of emotional states; `arousal versus non-arousal` *(+A and -A)* that points to a combination of physical activity and mental alertness; and `dominance versus submissiveness` *(+D and -D)* that stems from control versus lack of control. A three-dimensional PAD emotion space with three scale ranges from -1 to +1 points towards specific emotion terms like `angry` [-0.51, 0.59, 0.25], `curious` [0.22, 0.62, -0.01] or `boredom` [-0.65, -0.62, -0.33],

[6]PAD Temperament Framework: `http://www.kaaj.com/psych/scales/temp.html`

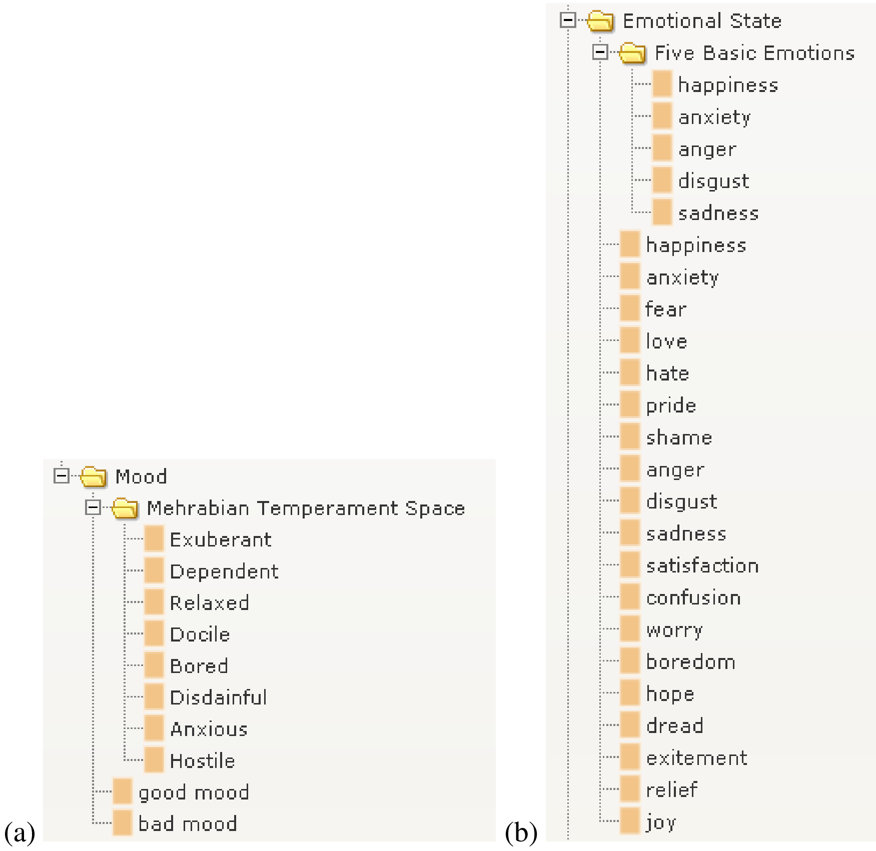


***Figure 5.10**: (a) Mood and (b) Emotions in the general user model ontology* GUMO

see for example [Klesen, 2002] for an introduction. Again, this model can be realized with the interplay of SITUATIONALSTATEMENTS and general user model ontology GUMO with a specialized range.

Humans communicate intentionally and unintentionally through a variety of emotional expressions and moods that are expressed for example in speech patterns, facial expressions [Ekman, 1993] and the body language of the individual. An interplay between emotions and learning is for example presented in [Kort et al., 2001] and [Goleman, 1995].

In the NECA[7] project a platform for the generation of affective animated conversations has been developed. [Krenn et al., 2003] identifies the following problems with emotion modeling that also make the task for the general user model ontology GUMO difficult:

- *multimodal expression of emotion*: most existing studies of emotion expression focus on a single modality (speech or facial expression) and do not address the question of the integration and the relative importance of different modalities.

[7]Net Environment for Embodied Emotional Conversational Agents: `http://www.oefai.at/NECA/`

- *sociocultural and situational influence*: from psychological experiments on human-human affect perception it is known that the situational and sociocultural context has strong influence on the emotion perception.

- *mapping between different approaches to emotion*: currently there is no principled or theory driven way to map between, for instance, the OCC model, the emotion dimensions and Ekman's basic emotion categories.

Especially the mapping between different approaches to emotion is important for the research in user model integration and exchange if the user's emotional states are described with different affective theories in mind.

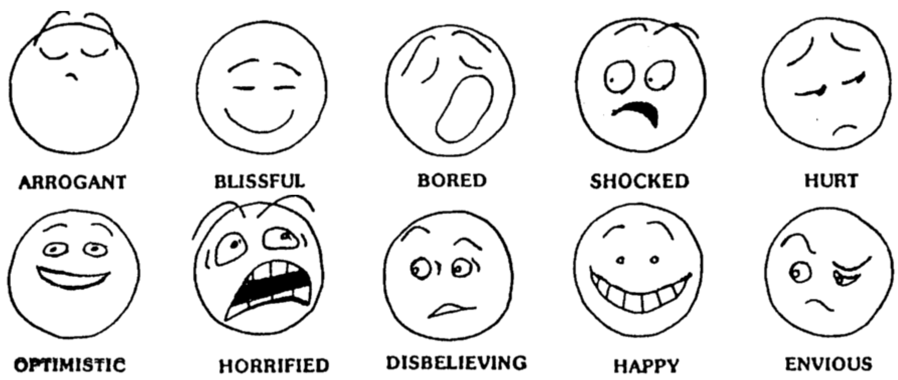


Figure 5.11*: Selected facial expressions in* GUMO

Another research area that is closely related to studies on emotion is the research on facial expressions[8]. Facial expression is according to [Ekman, 1993] among the most powerful, natural, and immediate means for people to communicate their emotions and intentions. The face can express emotion sooner than people verbalize or even realize their feelings. Even though facial expressions are often representative for emotional states, we decoupled them in our ontology. If a user shows some facial expression, we allow to represent this information in the UserJournal without interpreting the current emotional state of the user. Figure 5.11 shows some selected facial expressions that are modeled in GUMO.

5.2.2.4 Mental State and Physiological State

Research on identifying `mental states` like `time pressure`, `cognitive load` or `nervousness` is for example carried out in the READY[9] and BAIR projects. Further mental states like `depression` or `irritation` are modeled and shown in figure 5.12(a).

Indices of `physiological states` like `blood pressure`, `heartbeat` or `pupils dilation` can be measured by specialized bio-sensors. Large amounts of input data has to be processed and fairly complex inference mechanisms have to be de-

[8]Facial Expression Analysis: `http://www.ri.cmu.edu/projects/project_10.html`
[9]READY publications: `http://w5.cs.uni-sb.de/~ready/ready-pubs.html`

veloped, see [Brandherm and Schmitz, 2004]. Physiological states like `respiration`, `temperature`, `arousal` or `nourishment` are themselves used as symptomatic information for higher level user model dimensions like mood and emotions, as discussed above.

(a) (b)

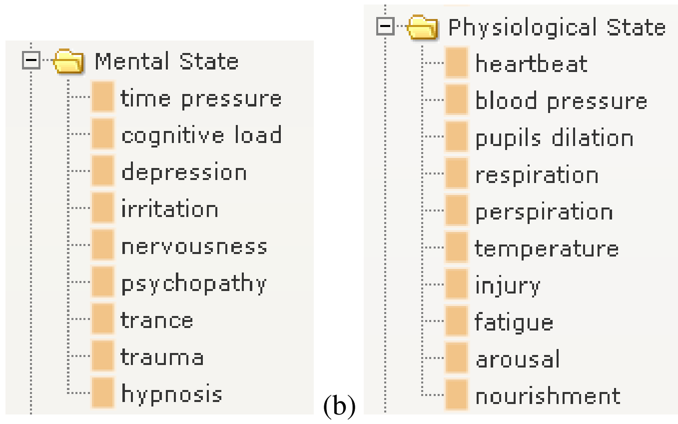


***Figure 5.12**: (a) Mental State and (b) Physical State as shown in the Ubis Ontology Browser*

According to [Jameson, 2001a] the reliability of the measurement of physiological states is fairly high for those that are closely associated with measurable symptoms. Furthermore the user has typically no cognitive effort since she or he doesn't have to give any explicit input. Using biodata as a source for ubiquitous user modeling is discussed in section 9.2.

5.2.2.5 Role, Profession and Proficiency

Especially in ubiquitous computing, the user can take several roles and change frequently between them: In some situation the user can be the `learner` while in another situation he is the `teacher`. The user can visit a town as `tourist` or as `businessman`. A hotel manager system like HAM-RPM could for example adapt its reservation to this information, see [Hoeppner et al., 1980] or [Jameson et al., 1980]. A pedestrian navigational system like the the M3I Personal Navigator as described in section 9.4.1, could for example adapt its route calculation according to the user's role. Further opposed roles are for example `employee` or `manager`, `child` or `parent` and `customer` or `salesman`.

A second group of basic user model dimensions that is mentioned in this subsection is the user's `profession` or `position`. However, since this topic has already been extensively modeled in the SUMO/MILO ontology, see section 3.3.1 on page 42, it has been integrated unmodified into the GUMO ontology. These dimensions can be used with the prefix `sumo:`

The third group in this subsection is the ability and proficiency of the user. These are especially important if disabilities handicap the user. Crucial abilities for the adaptation in human-computer interaction are `AbilityToSee`, `AbilityToHear` and the `AbilityToTalk`. Others are for example `AbilityToGrasp`, `AbilityToWalk` and interesting for the pedestrian navigation system: `AbilityToUseStairs`. GUMO models skills as subgroup of the abilities. Prominent exponents in user modeling are `TypingSkills` that reveal information about the familiarity with computers,

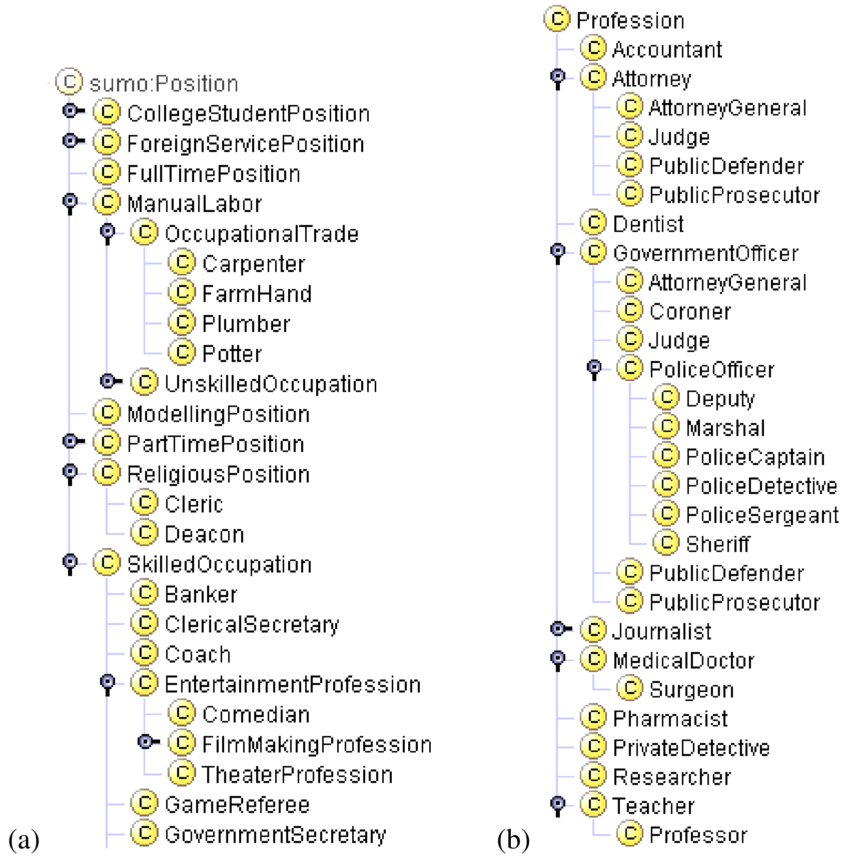


Figure 5.13: *(a) SUMO Position and (b) MILO Profession as part of* GUMO, *displayed with the Protege ontology browser*

`ReadingSkills` and `WritingSkills`.

5.2.2.6 Motion, Location and Orientation

The basic user model dimension `motion` allows to model if the user is `walking`, `sitting`, `lying`, `standing`, `goingUpStairs` or `goingDownStairs`. These dimensions could for example be detected by acceleration sensors as described in section 9.2, or by a mobile positioning system together with a detailed spatial model as described in section 9.4. Every motion can be attributed with a `speed` parameter. The dimension `SpatialLocation` covers all symbolic and geometric real world locations, while virtual positions in virtual worlds, or web pages are modeled by the dimension `VirtualLocation`. The underlying theory about *real and virtual versus original and reference* is introduced in section 2.2.5. The underlying spatial model is described together with the spatial ontology in the section 5.3.2.

5.2.3 Domain Dependent User Model Dimensions

Domain dependent user model dimensions differ from basic user model dimensions with respect to the required additional general world knowledge. If we want to express for example the interest of a user in a certain film category or the preference of that user for certain wines, we need ontologies about film genres and wines to base this on. Domain dependency in this interpretation should not be confused with the discussion about distinguishing user models from discourse models, as issued in [Wahlster, 1988] where a *user model* is defined as a knowledge source that contains explicit assumptions on all aspects of the user that may be relevant for the interaction behavior[10] of the system. According to this definition domain dependent user model data eventually belongs to the user model. However, this leads to the problem that any concept in the whole world is a potential candidate for expressing user model data about interests, preferences or knowledge, as discussed in section 5.2.1.

Our suggested solution to this problem is that we open the GUMO architecture to any external ontology and express user model data with the modularized SITUATIONAL-STATEMENTS. However, we additionally developed the user model interest categories as combining elements between GUMO and external ontologies.

5.2.3.1 User Model Interest Categories

The user model interest categories form a large listing of interest and preference categories like film genres, music trends, sports, pc-game genres, environmental topics and so on. Figure 5.14(a) shows the main categories that are realized so far, while figure 5.14(b) shows preference settings within the museum's domain that was used in the PEACH project as discussed in section 9.5.

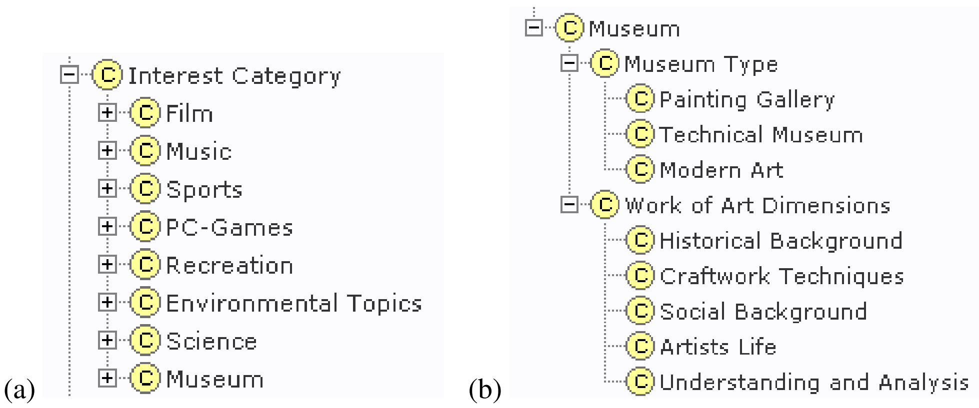


Figure 5.14: *(a) Main interest categories, (b) categories in the museum's domain*

The complete listing of the interest categories, which is based on the *Librarian's index to the internet*, can be found in the appendix B.1.4 on page 224. Figure 5.15 shows example statements about interests within a user model.

[10]*dialog behavior* is originally used in [Wahlster, 1988] since it is defined within a dialog system

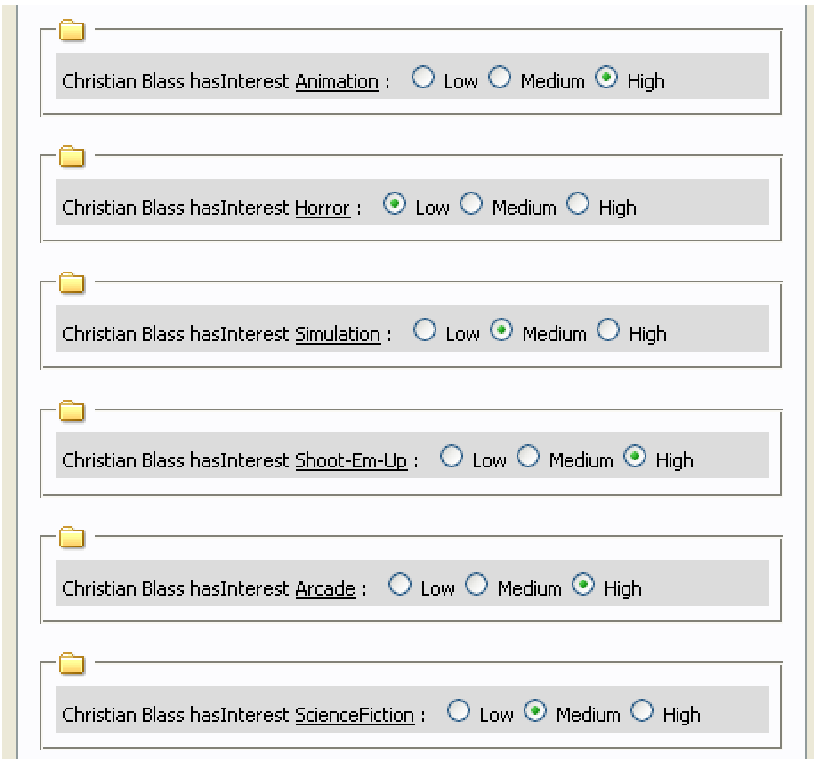


Figure 5.15: *Example statements in a user model about interests in the PC-Game genre*

5.2.3.2 Discussion

We did not model the "Medical Domain" within GUMO since there are large active projects in this so-called *eHealth* research domain, see e.g. IFOMIS[11], *Gesundheitskarte*[12] or MedCIRCLE[13].

[11]IFOMIS homepage: `http://www.ifomis.uni-saarland.de/`
[12]Gesundheitskarte homepage: `http://www.dimdi.de/de/ehealth/index.htm`
[13]MedCIRCLE homepage: `http://www.medcircle.org`

5.3 UBISWORLD and UBISONTOLOGY

Human-computer interaction in ubiquitous computing needs a uniform virtual world model in order to simulate, represent and compare research issues. The initial idea behind UbisWorld is the extension of the Blocks World, see [Slaney and Thiébaux, 2001], and the extension of the context toolkit, see [Salber et al., 1999a] to the special needs of situated interaction in ubiquitous computing with user modeling and privacy. The prefix *Ubis* abbreviates the term *Ubi*quitou*s*. The postfix *World* indicates, that our approach tries to be very broad and it refers to the blocks world. UbisWorld can be seen as a collection of concepts and models for location and time, for interaction and situation that are all prepared for ontological representation and data collection. UbisWorld has so far been described in [Heckmann, 2003a], [Heckmann, 2003c], [Stahl and Heckmann, 2004b], [Kruppa et al., 2005] and [Compiol, 2005]. With this virtual intelligent environment we are able to run experiments for ubiquitous computing. The next subsection shows the full interrelationship between the real world and UbisWorld.

From Real World to UbisWorld

UbisWorld can be used to represent some parts of the real world like an office, a shop, a museum or an airport. It represents persons, objects, locations as well as times, events and their properties and features. Apart from the representational function, UbisWorld can be used for simulation, inspection and control, but the question *"What parts of the real world should be represented?"* forms the first step in the diagram of figure 5.16.

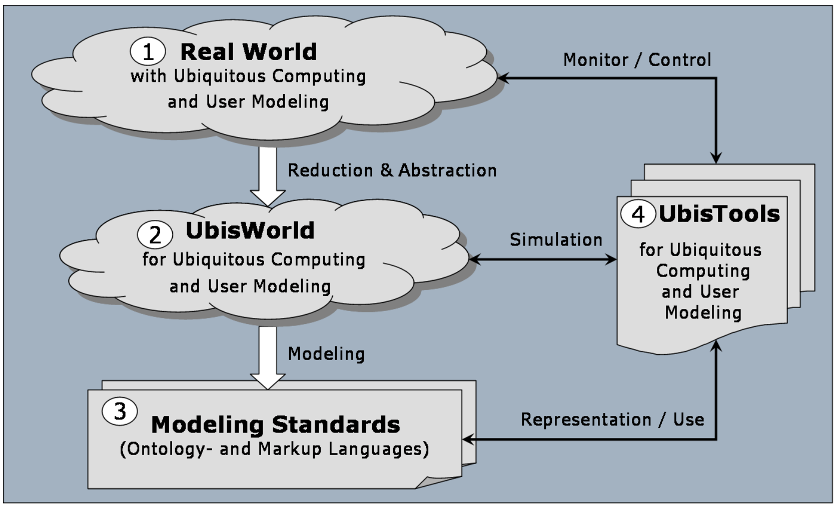


***Figure 5.16**: Conceptual overview: from real world with ubiquitous computing and user modeling (1) to UbisWorld (2) with modeling standards (3) and specialized tools (4)*

Cloud number (1) in figure 5.16 stands for the points of interest of the real world with ubiquitous computing and user modeling. UbisWorld, displayed as cloud number (2), is realized by reduction and abstraction from the real world. The modeling standards like markup-languages and ontology definitions are derived from UbisWorld abstractions.

It is interesting about the role of UbisTools (4) for ubiquitous computing and user modeling, that on one hand operate on the level of UbisWorld and on the other hand operate on the level of the real world.

A small example to illustrate the interrelation between a simulated world and a real world: *Imagine that there is a room in the real world with two doors, two light switches at each door and one light at the ceiling. All these elements will be represented in an abstract manner in the corresponding UbisWorld model. The UbisTools could simulate the light-on light-off behavior of the real room, such that if the virtual light switch gets pressed the virtual light in UbisWorld shines, independently from the real world. Secondly, the UbisWorld room could be used to monitor the real room in such a way that every time when the real switch has been used, the virtual light shows the status "shining". As a third possibility, the UbisWorld room could be used to control the real world room for example by turning the real light on or off, every time when the virtual light switch is used.* What follows - from this example - is that the ontology engineering of the UbisWorld ontology should be independent from the later task of representation, simulation, monitoring or even control of the real world.

A second important point, before one can start with modeling the UbisWorld ontology, is that we need at least a rough idea about the interaction model for ubiquitous computing with integrated user modeling and privacy. Figure 5.17 shows a simplified human-computer interaction model, where the context, the location, the time and the sensing & computation are schematically wrapped around the user with his or her user model and the system with its resource model.

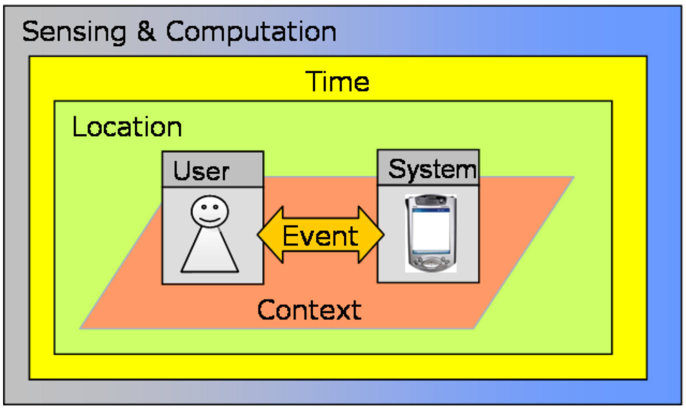


Figure 5.17: *Simplified human-computer interaction model for ubiquitous computing with user modeling and privacy*

The *User* icon and the *System* icon are also meant to represent user groups and collections of interacting systems. In section 2.2.4 a more complex interaction model has been discussed.

The colors of the simplified human-computer interaction model as shown in figure 5.17 and in figure B.1 on page 230 are used to classify the corresponding partial ontologies. Since the semantic web ontologies support and encourage the integration of several ontologies, the design decision for UbisWorld was to develop specialized partial ontologies, rather than trying to realize one ontology for all aspects. Figure 5.18 shows in one diagram the elements of the UbisWorld concept, which consists of classes and predicates, of individuals, and of relations. In UbisWorld the classes and predicates are defined in six additive ontologies, namely the *physical ontology*, the *spatial ontology*, the *temporal ontology*, the *activity ontology*, the *situation ontology* with situation describing dimensions that also cover the general user model ontology GUMO and finally the *inference ontology* which models the computational and intelligent behavior in ubiquitous computing environments.

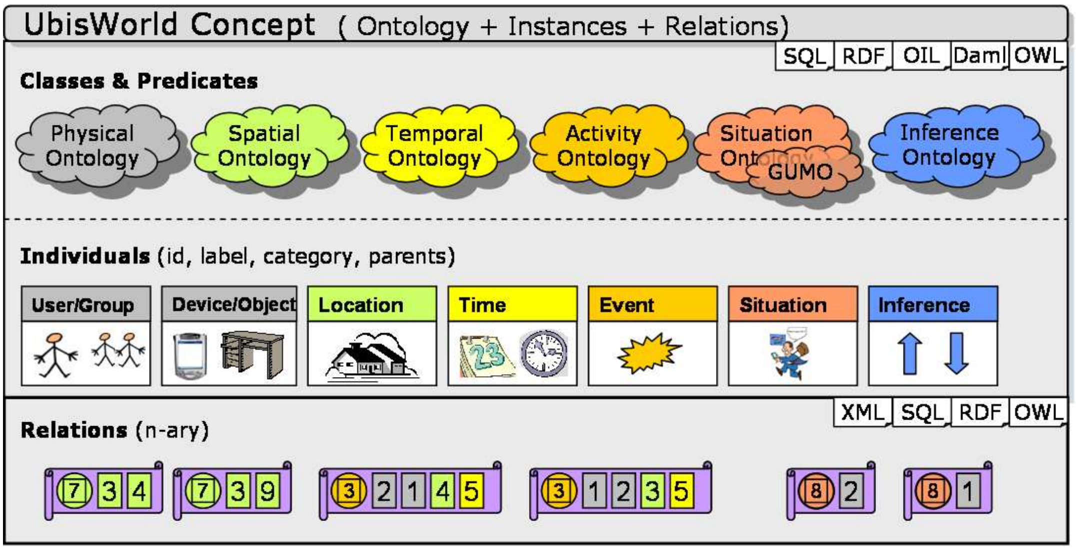


***Figure 5.18**: UbisWorld = Ontology + Instances + Relations*

Figure 5.18 indicates, that all ontologies are available in several ontology representation languages, among which are RDF, OIL, Daml and OWL. This verbosity allows a maximal reusability for different systems. All these different instantiations of the ontologies are realized by a generation and translation tool. Some parts of the ontologies can be found in the appendix B.2. In section 8.2.2 an ontology editor is described, that allows the online inspection, adding and editing of the ontological elements. Also important to mention is the fact, that the design decision for UbisWorld was to work with n-ary relations rather than binary relations only, even though they are not yet directly realized within the new semantic web approach. N-ary relations are especially important for the integration of SITUATIONALSTATEMENTS that can contain up to 25 positions per relation. Figure 7.7 on page 147 shows the integration of SITUATIONALSTATEMENTS into the rest of the ontology. In the following subsections, the concepts behind the partial ontologies are discussed.

5.3.1 Physical Ontology: Users, Groups, Devices and Objects

The physical ontology introduces physical objects, that are especially important under the given constraint of ubiquitous computing and user modeling. With physical elements we think of persons, devices, objects, furniture, goods, food and so on, everything that can play an important role in so-called intelligent environments. The ontology model divides between the three basic elements `Being`, `Thing` and `System`, and element groupings like `UserGroup`, `SystemGroup` and `MixedGroup`. Every basic element carries a default class icon. Important categories for the ubicomp[14] domain are `User`, `Device`, `Furniture` and `Vehicle`. All other graspable objects will be defined under the category `OtherObject`. Of course this categorization is arbitrary, however, it is of minor importance. More important is the correct identification and the inheritance of class properties to the instances. Figure 5.19 shows the partial top-level ontology of the physical ontology with the focus on the `DeviceElements`.

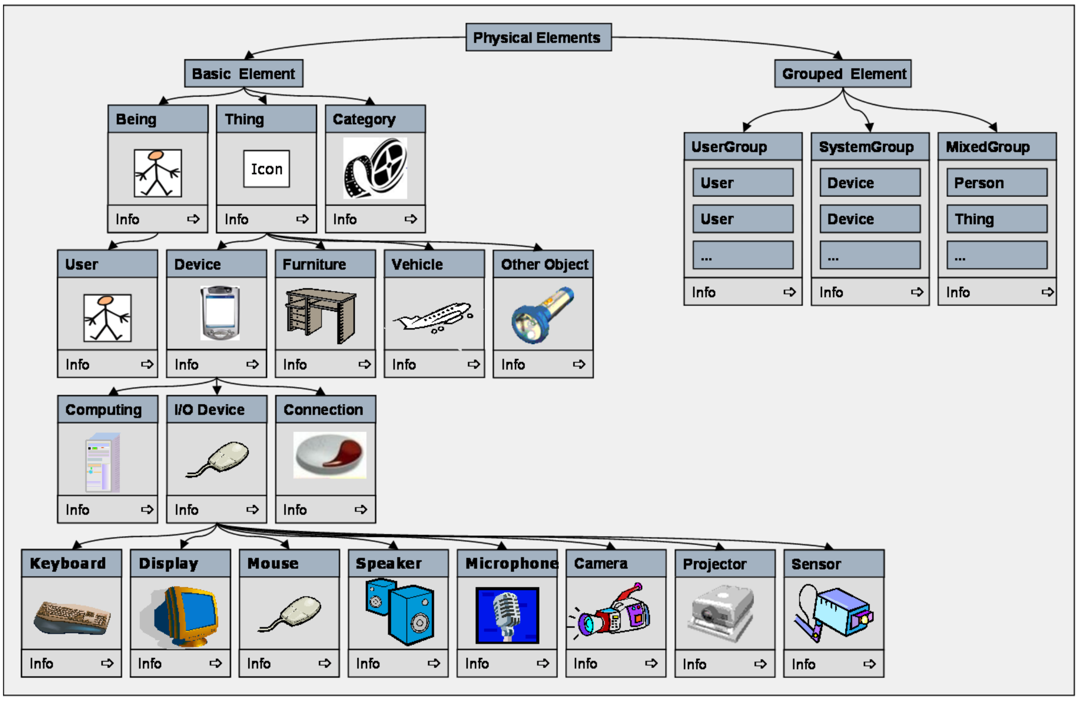


Figure 5.19: *Some physical elements in the so-called physical ontology with the focus on device elements*

Input/output device elements are for example: `Keyboard`, `Display`, `Mouse`, `Speaker`, `Microphone`, `Camera`, `Projector`, `IR Bark`, `ID TAG`, `Notebook`, `Mobile Phone` and so on. They are important to model the instrumentation in the environment.

[14]ubicomp = abbreviation for ubiquitous computing

Each element in the ontology has its own web page as a virtual representative. Figure 5.20 shows the class `SmartShopObjects` as sub-class of the class `Thing`. This illustrates the close connection between the real world scenarios and the formal ontology. Its ancestor elements and its child elements are presented on this web page as hyperlinks. The concept behind the interlinked elements in the UbisWorld ontologies is a mixture between *frames with slots* and *semantic nets*.

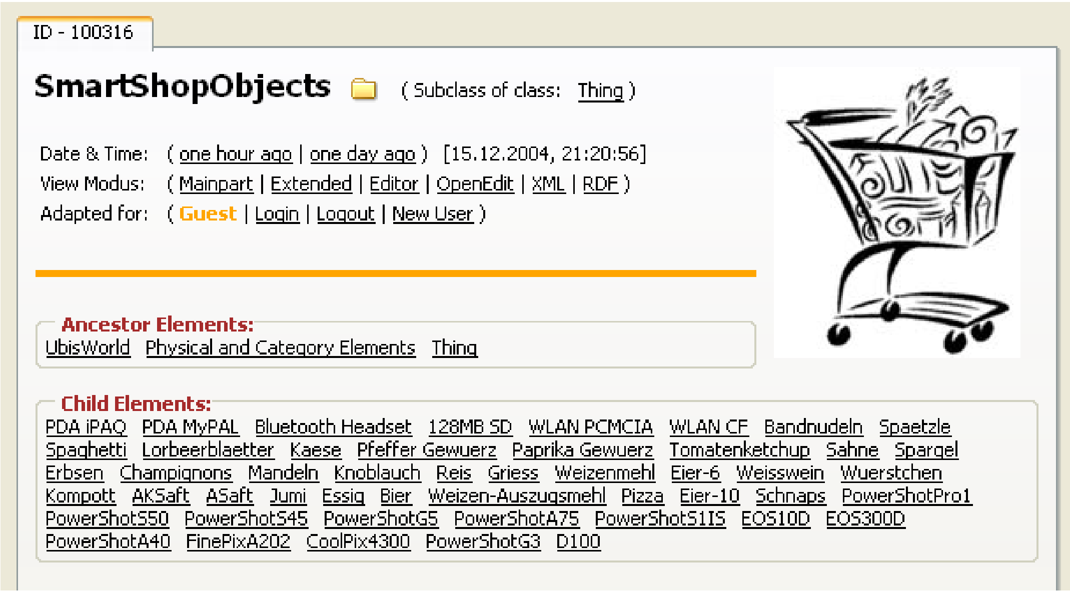


***Figure 5.20**: The class SmartShopObjects (in the UbisWorld Element Browser) is a conainter for several physical elements that are used in the smart-shop scenario*

Since our ontology allows multiple inheritance, an object like the *Canon PowerShot S50* can be the child in the device hierarchy, but at the same time be the child of the SmartShopObjects. Figure 5.21 shows the virtual representative of an instance of this camera. All products in the SmartShopObjects class have at least the product information *vendor* and *price*. No problem arises if we describe it as a product-type or product-instance, since both are equally handled as entities within UbisWorld. Further information about physical objects that is of importance for ubiquitous computing is the degree of mobility. Objects can be *mobile*, *movable* or *fixed*. Some objects may be small, but not necessarily mobile. A mobile object is for example a PDA. A movable object for example is a desktop PC. It is not meant to be be moved regularly. The degree of mobility has an influence on the default change model. Furniture like a table, a shelf, a bed, or a chair are also defined as being movable. If no statement is mentioned about a movable or fixed object, it is assumed that its location has not changed. See section 4.3.4 on page 67 for a discussion of the frame problem.

Another interesting property of objects and devices in the PhysicalOntology, which is especially important for ubiquitous computing, is the distinction between the *public* and

Figure 5.21: *The virtual representative of the Canon Powershop S50*

private status. The SPECTER concept as described in [Kleinbauer et al., 2003] is a private - mobile device, while the large screen in the instrumented environment as described in [Stahl et al., 2004] is a public - fixed device. In [Kruppa and Krüger, 2003] the importance of public and and private devices is discussed.

5.3.2 Spatial Ontology: Location, Topology, Orientation

Physical objects and devices are spatially arranged while the users move and interact in mobile and ubiquitous computing with their everyday environments. Services are location-aware while local awareness plays an important role. Thus a profound investigation of spatial concepts and models has to be undertaken, see [Stahl and Heckmann, 2004b] for an extended summary of this research. The first spatial concept analyzed here is the one of spatial granulation levels. As stated above, mobile and ubiquitous computing shifts the human-computer interaction from the desktop in one room into the halls of an airport or shops. In the travel domain, whole countries or continents can be the point of interest for description. Figure 5.22 shows nine significant spatial granularity levels from *Country* over *City* and *Building* to *Room* and even to the *Object Level* that can be identified in connection with the research about human interaction.

The distinction between *handling of objects*, *indoor moving within rooms* and *indoor moving between different rooms*, *outdoor moving*, *car&bus driving* as well as *airplane&train traveling* is interesting, since different techniques for the detection of the user's location are applied and different ways of adaptation are introduced according to their corresponding

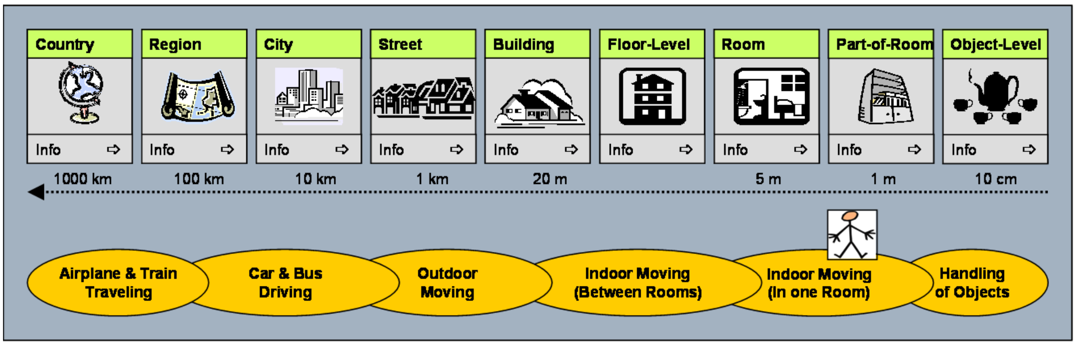


Figure 5.22: *Spatial granulation levels from "country" over "building" to "object level", together with corresponding human mobility interactions from "airplane and train traveling" to "handling objects"*

spatial granulation levels. Instances of qualitative spatial elements are rooms, buildings, cities, streets, and all can be mapped to these spatial granularity levels. Furthermore all instances of spatial and physical elements can be spatially arranged and different relations can hold between them. An example visualization of the spatial relation "is-nested-in" of physical and spatial elements is presented in figure 5.23.

This partial topology model puts the focus on the offices of the *WW-floor*[15] in *building E1 1* at the Saarland University. Even persons and their mobile objects that they carry along are displayed in the topology. This tree visualization enables a fold- and unfold-functionality of spatial locations. Thus the UBISLOCATIONMONITOR, that displays the context of locations, can be called for every qualitative location instance by browsing the location tree and clicking to any node. Figure 5.24 shows the UBISLOCATIONMONITOR for *Room 124* which is an instance of the class `Room`. The displayed information is adapted to the person that is currently logged in, in this case an anonymous guest.

The attribute `Location Path` shows the parents of room 124 and their is-nested-in relations. The question that now can be handled situation-aware by this information is *Where is Rainer?* The adapted answer in this spatial model could be *Room 124* for example for a call within the floor, it could be in *building E1 1* for a request within the university, or it could be *Saarbrücken* in some cases, or even *Germany*. To enable such adaptive answers automatically without a spatial inference system, the UBISLOCATIONMONITOR offers the whole location path in form of a web service. The attribute `Ancestor Elements` is related to the ontology and classifies the object as spatial element. The attribute `Physical Elements in this Location` says that *Rainer* is currently in this location. Furthermore, the `Temperature` is said to be high, while the `Noise Level` is said to be low. Thus spatial elements can have properties in form of SITUATIONALSTATEMENTS just like users or physical objects, however most of the predicates differ.

[15]This location model can be found at `http://www.u2m.org/ubisworld.htm`. It has partly been modeled by Holger Schultheis in [Morsing and Schultheis, 2004]

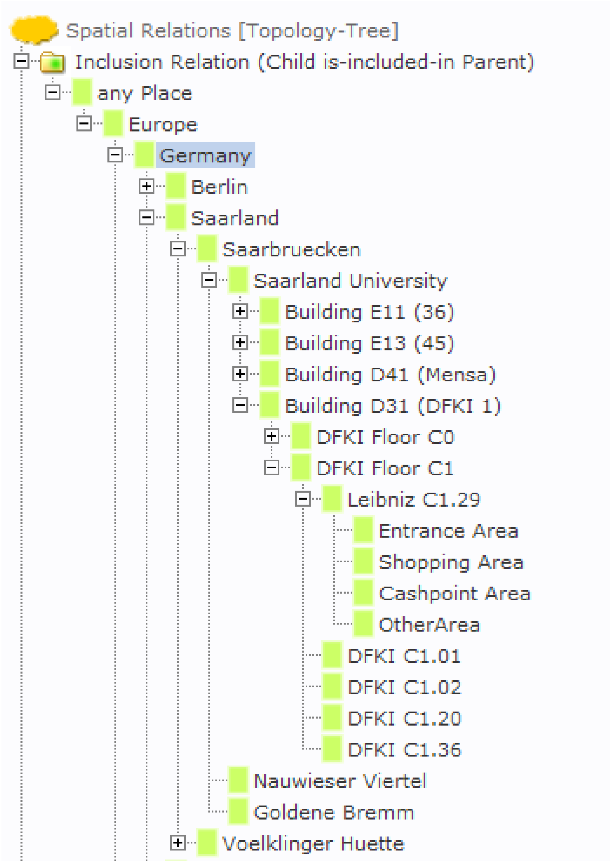


Figure 5.23: *Example visualization of the spatial relation "is-nested-in" in UbisWorld*

An important problem that has to be solved and integrated in the spatial ontology is the fundamental structural difference between qualitative or symbolic location models (as described so far) and quantitative or geometric location models with coordinates (as supplied by a GPS[16] receiver) in two, two and a half or three dimensions. In [Stahl and Heckmann, 2004b] a new hybrid location modeling has been introduced with the intention to model location in order to realize location aware applications with a focus on situated user interaction and pedestrian navigation. The tool YAMAMOTO[17] has been developed for the modeling of hierarchical geometrical maps. Figure 5.25 shows the map reference functionality that combines the symbolic topology on the right hand side with the corresponding coordinates on different maps on the left hand side.

The tool YAMAMOTO, that is described in section 8.12 on page 166, represents the geographical coordinates of real-world places in different granularity as resources in the world wide web. The geometrical model is joined with the symbolic model by uni-

[16]GPS = Global Positioning Service
[17]YAMAMOTO homepage: `http://w5.cs.uni-sb.de/~stahl/yamamoto/`

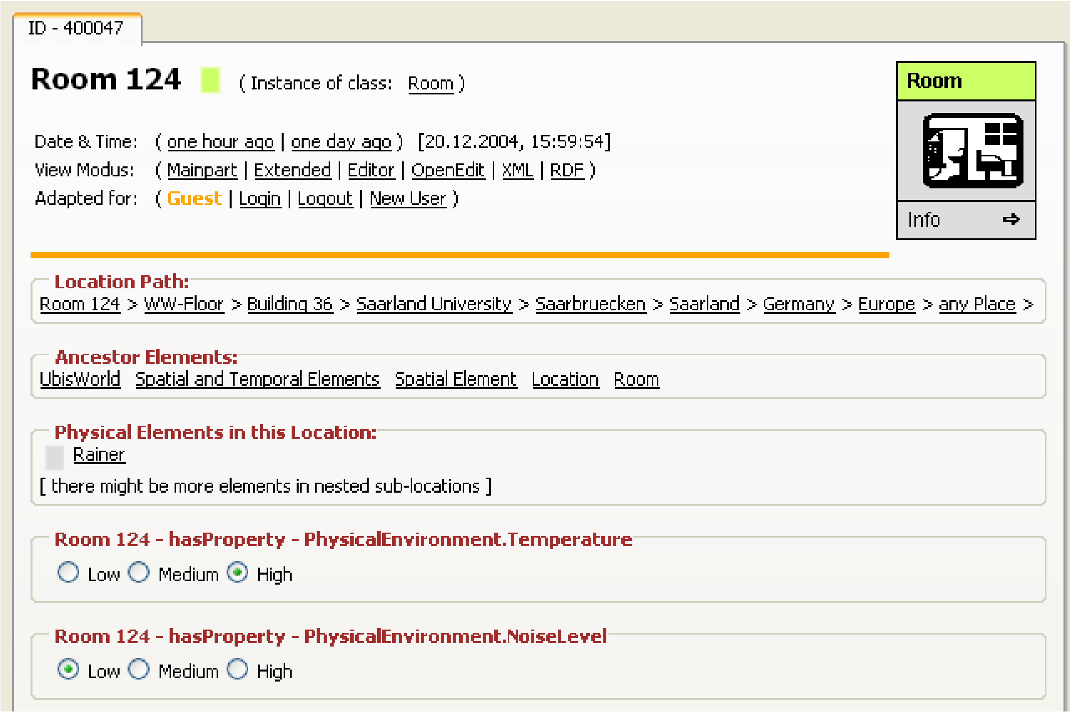


***Figure 5.24**: UbisLocationMonitor for Room 124*

form resource identifiers (URIs) as introduced in section 4.5 on page 74. The mobile systems in our scenarios locate themselves outside buildings using a GPS receiver, while indoors the intelligent environment provides localization based on an infrared beacon and RFID tag infrastructure, see [Stahl et al., 2005]. Symbolic and geometric location models are discussed in [Leonhardt, 1998] where a combined model is proposed. In [Jiang and Steenkiste, 2002] the aura location identifier ALI is introduced. It is based on hierarchical subspaces to realize a space service to handle spatial queries with a relational database. [Dürr and Rothermel, 2003] suggest a lattice instead of a tree structure to model hierarchical symbolic locations in order to express complex containment relationships between rooms to wings and floors of a building. In [Baus et al., 2005] an up-to-date survey of map-based mobile guides is presented. The NEXUS project[18] provides a platform for location-based services. It relies on an augmented world model, that represents real world objects as well as virtual objects. The world model is hierarchically decomposed into area models, which are provided by a spatial model server infrastructure. The positions of mobile objects are stored in location servers, which are designed to handle spatial queries on a large scale. The complete concepts of the UbisWorld spatial ontology, that are based on the related work discussed above, are shown in figure 5.26.

Spatial elements are divided into the two basic concepts of `Location` and

[18]NEXUS, Sonderforschungsbereich SFB 627, `http://www.nexus.uni-stuttgart.de`

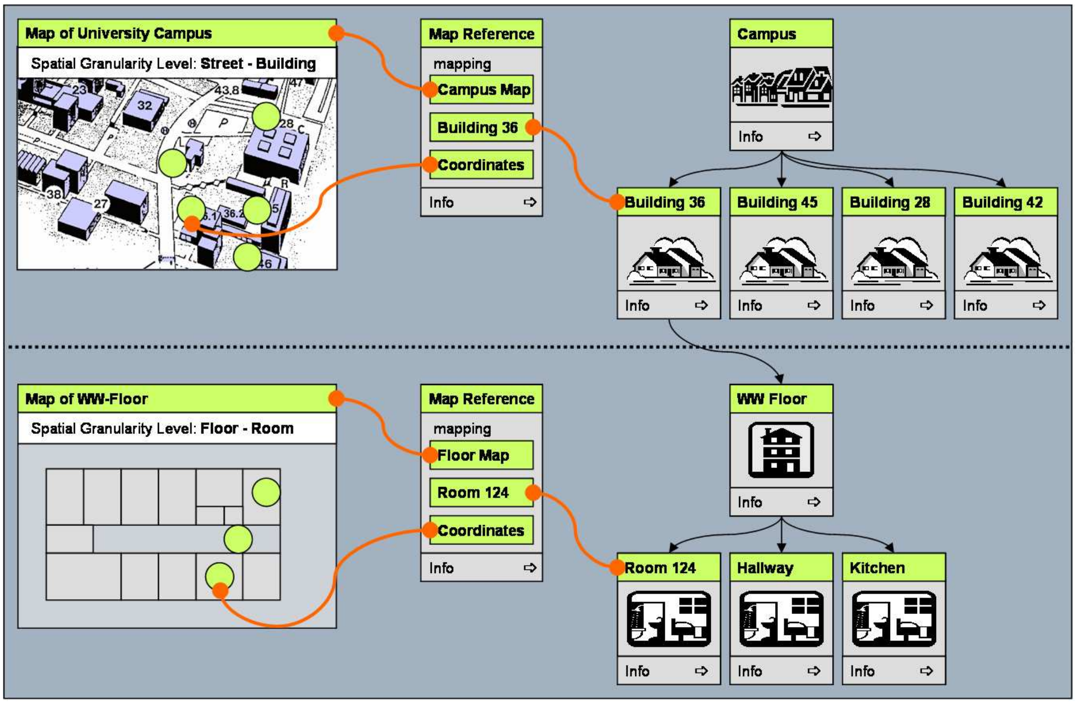


***Figure 5.25**: Map referencing and the hybrid location model*

`SpatialConstraints`. The first one can be a `Place` with the corresponding spatial granularity level, an `Area` where a spatial extension is given, an `Entity Location`, that defines the location next to a physical object, or a `Relative Location` that defines locations like *"between A and B"*. The second one, the spatial constraints, are `Connection`, that relates two locations by a connection, `Nesting` that relates two locations by the is-nested-in relation, `Map Reference` as discussed above, `Constraint` that allows for arbitrary spatial relations, and finally `Orientation`. This expressivity is helpful for modeling and inferencing in ubiquitous human-computer interaction.

5.3.3 Temporal Ontology: Point of Time, Interval, Temporal Constraints

The third partial ontology in UbisWorld is the temporal ontology. A clear model of time and time-intervals is essential, since most statements are related over the temporal dimension. The user interface design decisions allow for historical views of the user- and situation models. The real life offers verbose descriptions of time. Systems have to be synchronized, sometime the temporal knowledge is approximate or statements about the future have to estimate durations. In analogy to the spatial concept, a hybrid temporal concept has been developed. Figure 5.27 shows the concept of temporal granularity and the defining elements of a timestamp.

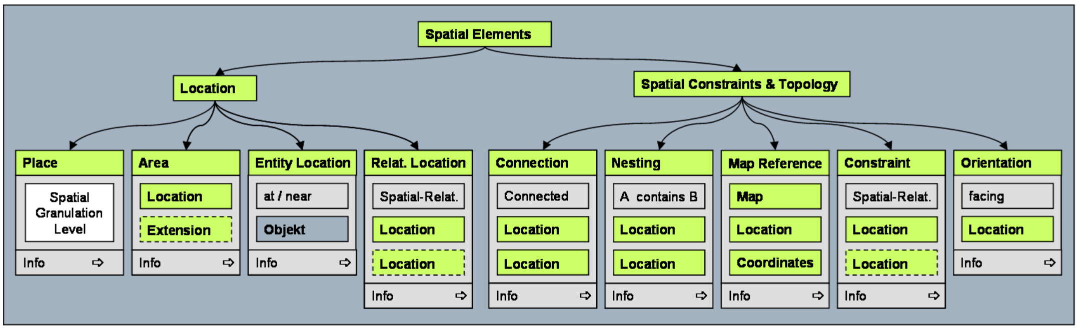


Figure 5.26: *Spatial elements that form the basic spatial concepts for UbisWorld*

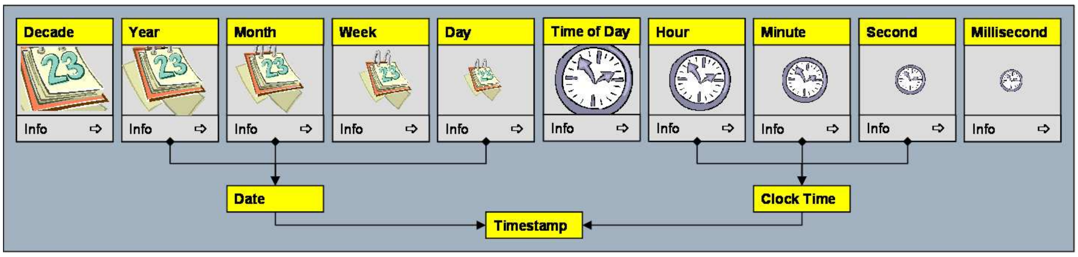


Figure 5.27: *Temporal granularity and timestamp*

The dimension of time is dense in such a way that every event has a temporal extension. Thus an interval seems to be the right concept to model time related issues, however by introducing the temporal granularity, one can also work with points of time. The duration aspect in the temporal domain corresponds to the concept of distance in the spatial domain. The `date` and `clock time` could be compared to the address in the spatial domain. The analogy between the temporal model and the spatial model can be seen if figure 5.26 and figure 5.28 are compared. If a time is asked in a question, it is not natural to simply state the full timestamp as answer, but the temporal granularity has to be adapted to the situation. For example to the question *When did something happen?* several temporal granularity levels could be the right one, like *in the 60s*, or *in 1969*, or *last Friday* or even a complex expression like *between event A and event B*. The first one describes the point of time by a whole decade, the second one by a year. The event itself could have happen within minutes, but the accuracy of the statement varies according so the situation. Thus the expressivity should be allowed. The temporal aspects within SITUATIONALSTATEMENTS can be found in section 4.2.

Figure 5.28 shows the temporal elements and divides between `Time` and `TemporalConstraints&Chronology`. The first one can be a `Point of Time` with its corresponding temporal granularity, or a `Time Interval` with a start time and either the duration or the end time. `Time by Event` defines the time by referring to an event. `Time by Relation` defines the time by unary or binary temporal relations. The second one, the temporal constraints and chronology , allow a `Partial Order` of the time by the `before` relation. They allow temporal `Nesting` or any other temporal `Constraint` like `after` or `overlap`. The currently implemented list of temporal elements as shown on the

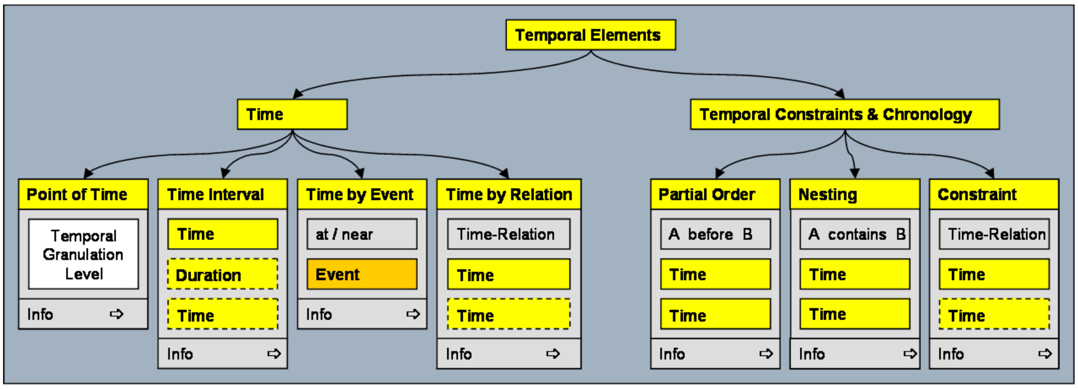


Figure 5.28: *Temporal elements: point of time, intervals, temporal constraints*

UbisOntologyBrowser can be found in the appendix B.2.3 on page 236.

5.3.4 Activity Ontology: Changing the World

With physical objects, spatial elements and temporal elements, as introduced in the three partial ontologies above, one can already describe complex but static worlds. Now, the activity elements describe the changes in the world and the most prominent one is *change of location*. Some events take place in point-of-times, others take place over time intervals. Four concrete example activities from the shopping domain are shown in figure 5.29. Such activities are described in SITUATIONALSTATEMENTS, such that the agents and time constraints and other important additional information can be stated or left underspecified.

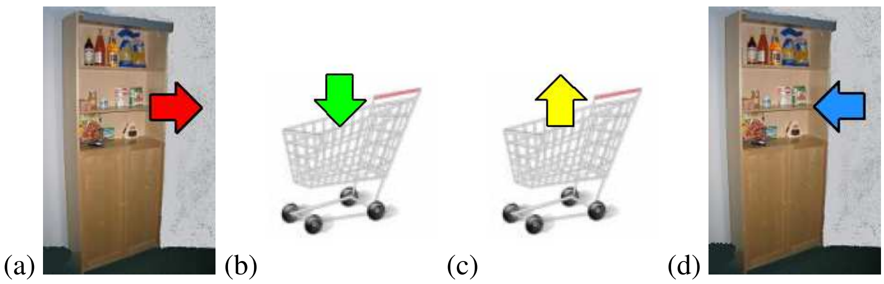


Figure 5.29: *Some examples of activities in the shopping domain: (a) shelf-out, (b) cart-in, (c) cart-out, (d) shelf-in*

Thus, the blocks-world that can be formed with elements from the UbisWorld ontology is not activity-centered or relation-centered. All types of elements (physical, spatial, temporal, activity, situation, inference) are equally important and arranged around SITUATIONAL-

STATEMENTS, that form the center of the whole approach. This view is established on the idea of *realism* in situation semantics, see [Barwise and Perry, 1983], where basic properties and relations are taken to be real objects and not sets of n-tuples or functions. Figure 5.30 shows some classes of activities in the view of the UBISONTOLOGYBROWSER. `Move`, `Take`, `Give`, `Put` and `Change` are the most basic ones. `Administration` activities are for example `Login` or `Logout`.

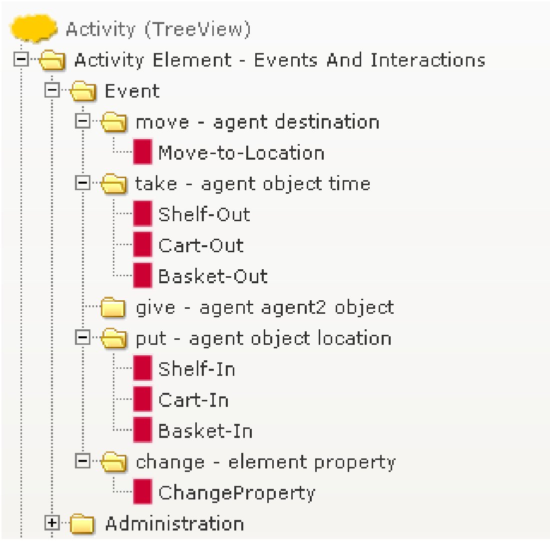


Figure 5.30: *Some activity classes as shown in the UbisOntologyBrowser*

This activity ontology finds its purpose in the software engineering and not in the research of proper classification and definition of concepts. We simply need resource identifiers to refer to the special activities and events, and the interoperability is realized by pointing to the same class or instance. Later on, in further research, either a profound ontological definition of the activity concepts will be needed, or attempts for further integration of the SUMO/MILO ontologies have to be undertaken. Figure 5.31 shows the conceptual diagram of the location events `Translation`, `Rotation` and `Pointing`. To represent an event or interaction, there are also slots with physical elements, spatial elements and temporal elements needed.

Not mentioned - so far - is the conceptual link to further information [Info $\Rightarrow$] as shown in the figures 5.31, 5.28, 5.27, and 5.19 for example, that describes the element or interaction in detail. This additional link allows to state contextual attributes and situational descriptions to every element or compound element structure in the conceptual model. The connection between the conceptual model and the UBISONTOLOGY is shown in the following subsection.

Figure 5.32 shows the graphical user interface of a basic activity editor to enter *move-to-location* events, and the *Actual List of Activities* monitor. Both functionalities need to be extended. The monitor displays the events that have been entered by the editor, but also the events that have been measured and inferred by the systems. Specialized monitors for the

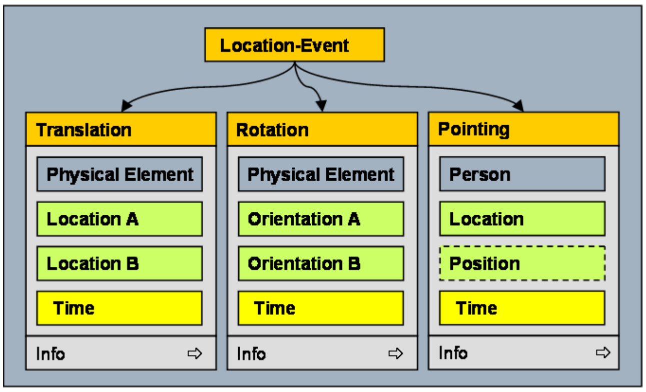


Figure 5.31: *Location Events in the Conceptual Model*

shopping domain and the pedestrian navigation domain are described in section 9.6.

5.3.5 Situation Ontology: Describing the World

Attributes, parameters or properties about users, systems, locations or activities, are collected in the *Situation Ontology*. A situational parameter for a location could for example be the `Noise Level`, the `Weather` conditions, or the available `Light`. A situational parameter for a person could for example be his/her `Blood Pressure`, his/her `Cognitive Load` or his/her `Interests`. Because of the importance and relevance for the research of ubiquitous user modeling, the situational parameters that concern users, have been analyzed and defined in more detail in section 5.2, starting from page 85. Hence, the defined general user model ontology GUMO can be considered as being part of the Situation Ontology. A situational parameter about a device or system could for example be its remaining `Battery Power` or its `Screen Size`. Figure 5.33(a) shows some situation elements from the context information subtree in the UbisOntologyBrowser.

The subclass `Social Environment` points towards the contextual group and team interactions. Examples for attributes from the `Physical Environment` are given in the screen shot in figure 5.24, and examples for attributes from the `Product Information` are given in the screen shot in figure 5.21.

A further distinction in the situation ontology can be made, between low level sensory data and higher level inferred data. Even though there is no clear possibility to distinguish between the different higher levels of inference[19], the low level sensor

[19]For example, the `Age` of a person can easily be estimated by the ages of elder and younger sisters and brothers. It could also be calculated in a complex inference process through speech analysis, as done by Müller in [Müller and Wittig, 2003], on the other hand, it could be analyzed by complex image processing from a photo on a birthday card. Thus the situational parameter itself is independent from the number of inference processes, that produce the value for the situational parameter.

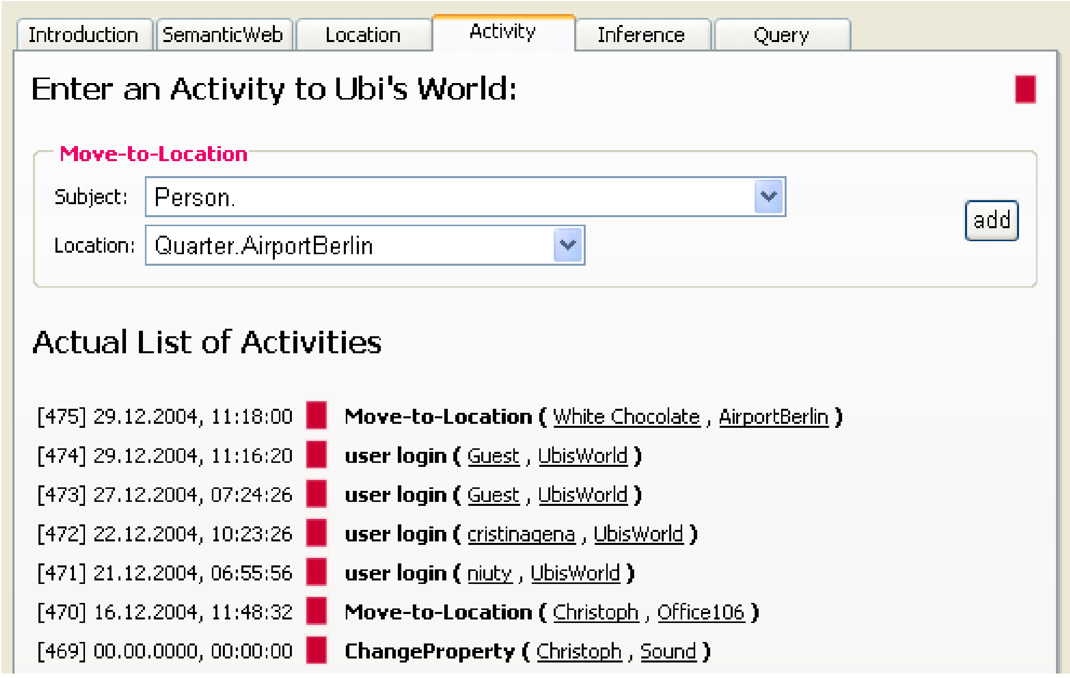


***Figure 5.32**: Activity Editor and the UbisActivityMonitor*

data can roughly be classified by the mass of uninterpreted data, being sent on a regular, frequent basis, from a special sensor device. Figure 5.33(b) shows some low level sensor data elements from the `Speech Parameters`, that were implemented in [Wasinger et al., 2003a], `Biometrical Sensor Data`, that were implemented in [Brandherm and Schmitz, 2004], and `Typing Behavior`, that were implemented in [Lindmark and Heckmann, 2000].

Possible values for the `Speaking Style` are *formal*, *informal* and *childish*. The question that arises is: *Should we add data types and ranges to the ontology or should we define them separately?* What turned out to be treatable in practice was modeling the distinct value sets within the ontology but to define the dense ranges as data types outside the ontology. Figure 5.34 shows several distinct value sets. Each data type receives an identifier and by that a referrable URI. Furthermore, each possible value receives an identifier and can thus be semantically described and interpreted by the means of the ontology. Most of the situation parameters are combined with a default data type, that could be overwritten if needed. For example the `Temperature` in a room could be measured in degrees *Fahrenheit* or in degrees *Celcius*, or its range could be the set { *cold*, *normal*, *warm*, *hot* }. As stated in the chapter about SITUATIONALSTATEMENTS, the `predicate` is decoupled from the used `range`. On the one hand, a data type like `PoorGoodPerfect.640040` can be used for many attributes, while on the other hand the data type `NoiseLevelType.640160` seems to be only useful for the physical environment parameter `NoiseLevel.820010`. However, this data type could for example also be used to classify the `happiness` of a baby: { *quiet*,

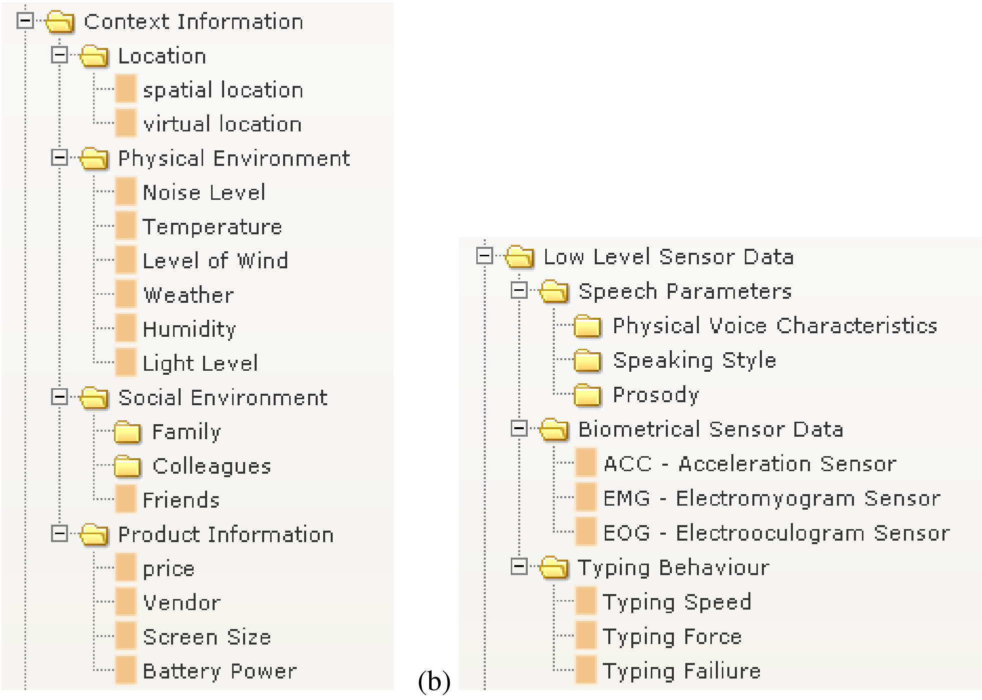


***Figure 5.33**: (a) Situation ontology elements for context information (b) Situation ontology elements, classified as low level sensor data*

normal, *noisy*, *loud*, *extremely loud* }.

5.3.6 Inference Ontology: Computing the World

Finally the power of computing and intelligent behavior enters the UbisWorld by the *Inference Ontology*. Inference elements define smart rules or proactive inference processes within intelligent instrumented environments. In ubiquitous computing and ambient intelligence, several scenarios have been proposed, among which the `home scenario` and the `office scenario` are the most prominent ones. In the *Georgia Tech's Aware Home Research Initiative*[20], see [Kidd et al., 1999], the house as shown in figure 5.35(a) is aware of its occupants whereabouts and activities. They analyze the question: *If we build such a home, how can it provide services to its residents that enhance their quality of life or help them to maintain independence as they age?* The Aware Home Research Initiative is an interdisciplinary research endeavor at addressing the fundamental technical, design, and social challenges presented by such questions and experimenting with the future of domestic technologies. The Microsoft Research project *EasyLiving*[21] is developing a prototype architecture and technologies for building intelligent environments. A survey of research on context

[20]Georgia Tech's Aware Home Research Initiative: `http://www.cc.gatech.edu/fce/ahri/`
[21]Microsoft Research EasyLiving homepage: `http://research.microsoft.com/easyliving/`

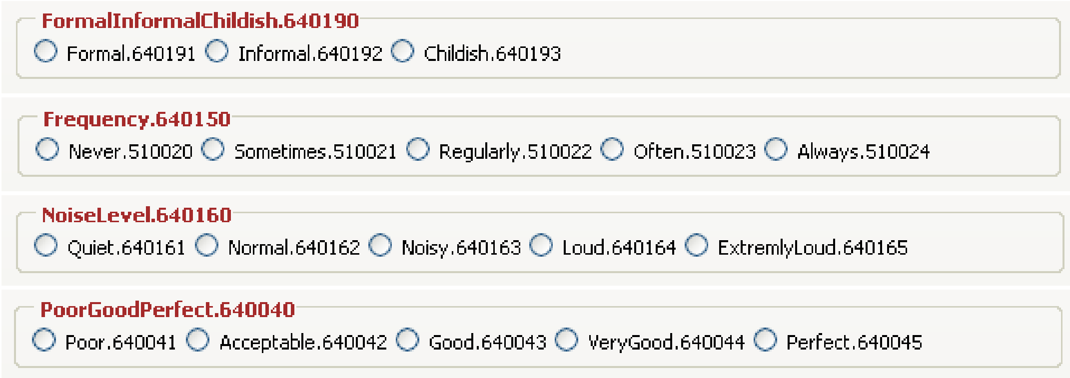


Figure 5.34: *Special sets defined as data types, as shown in the UbisOntologyBrowser*

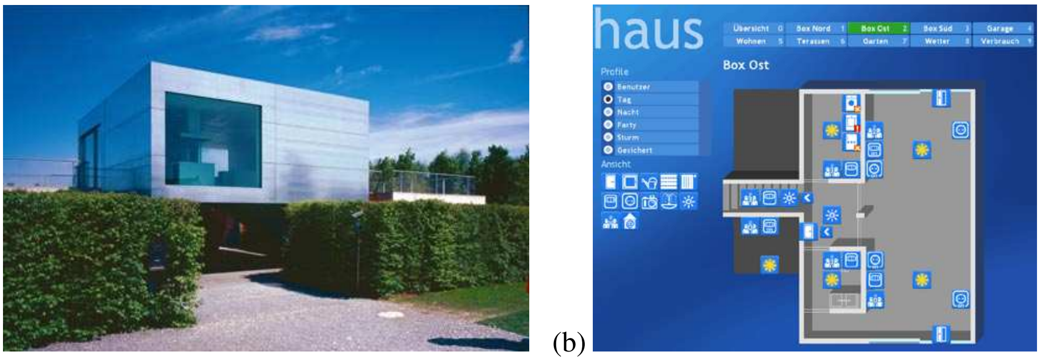
aware homes can be found at [Meyer and Rakotonirainy, 2003]. More recent European smart home research projects are for example the *Philips HomeLab*[22] in Eindhoven, the Netherlands, (figure 5.35(b) shows a prototypical display in a mirror), the "Haus der Gegenwart"[23] in Munich, Germany, see figure 5.36 and the "Intelligent House Duisburg Innovation Center - inHaus"[24] by Fraunhofer-Gesellschaft, see figure 5.37. The latter one includes apart from a residential home also a networked garden and a networked car. The basic idea is that not only all devices and components in the intelligent house communicate with each other, but they are also networked to the outside world, which allows distance monitoring and distance control.

(a) (b)

Figure 5.35: *(a) AwareHome by Georgia Tech, (b) HomeLab in Eindhoven by Philips*

An early application of intelligent objects in the ubiquitous computing office scenario

[22]Philips HomeLab: `http://www.research.philips.com/technologies/misc/homelab/`
[23]Haus der Gegenwart homepage: `http://www.haus-der-gegenwart.de`
[24]Fraunhofer-Gesellschaft inHaus homepage: `http://www.inhaus-duisburg.de/`



(a) (b)

***Figure 5.36**: "Haus der Gegenwart" in Munich with its graphical user interface by Microsoft*

(a) (b)

***Figure 5.37**: "inHaus" by Fraunhofer-Gesellschaft with its intelligent living room*

was the MediaCup[25] by the University of Karlsruhe, see figure 5.38(b). It is an ordinary coffee cup augmented with sensing, processing and communication capabilities that are integrated in the cup's bottom, to collect and communicate general context information in a given environment. The electronics in the cup senses how the cup is used. For example if someone drinks out of the cup, if someone plays with it and the temperature of the cup. This information is then communicated to other computerized objects in the room like a coffee machine or a Web server. Programs running on these objects use this data to support the human using these systems. `Small Meeting` of figure 5.38(a) describes the detection process of small meetings, for example if several cups of hot coffee realize their coexistence in a meeting room and deduce themselves that a meeting takes place, see [Gellersen et al., 1999].

The inference ontology, which is still under construction[26], is especially introduced to collect all inferences the lead to intelligent interaction behavior in the research of ubiquitous computing. `Remind me there` and `Remind me then` represent virtual notes that can be posted within the whole UbisWorld as *Digital Graffiti*. A real world example of this technology is for example realized in the SPELLBINDER[27] project. These there & then rules are used to demonstrate basic proactive behavior within this virtual world. Figure 5.39 shows

[25]MediaCup homepage: `http://mediacup.teco.edu/`
[26]The current status can be found at `http://www.ubisworld.org`
[27]SPELLBINDER homepage: `http://spellbinder.inf.ed.ac.uk/`

(a) (b)

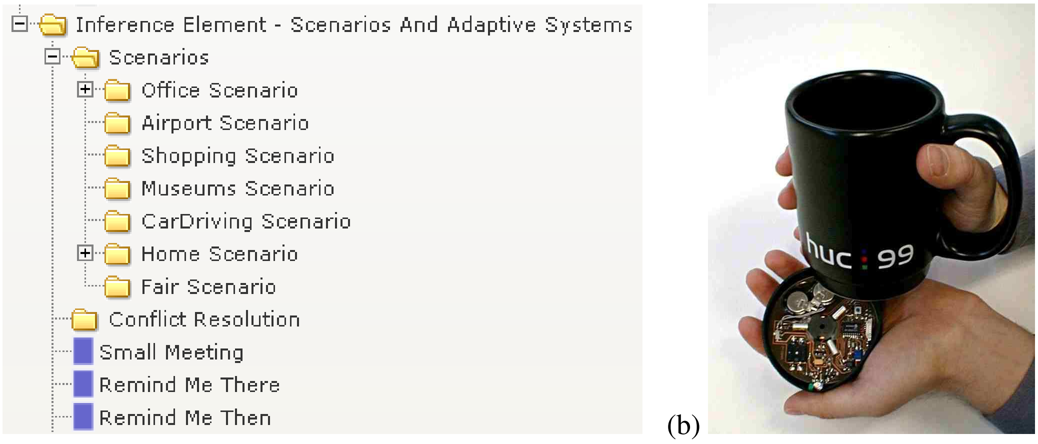


***Figure 5.38**: (a) Collection of inference elements, (b) MediaCup by the University of Karlsruhe*

the screen-shot of a digital graffiti trigger editor. The currently edited rule says that *everytime, the person Jörg enters the restaurant Schlemmer Eule, he will be reminded with a sentence that a certain dish is known to be very good there.*

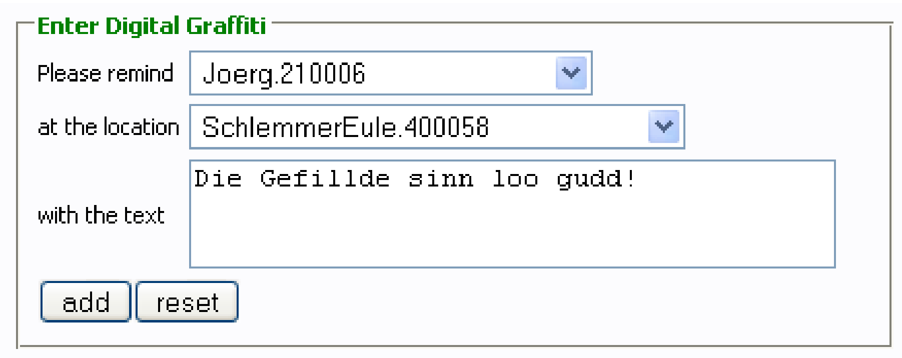


***Figure 5.39**: Screen-shot of the digital graffiti trigger editor in UbisWorld*

To summarize, *UbisWorld* has been developed under the impression of the real world with ubiquitous computing and user modeling. All elements are defined in six open ontologies: the physical ontology, the spatial ontology, the temporal ontology, the activity ontology, the situation ontology (which contains the general user model ontology GUMO), and the inference ontology. Altogether, they form the UBISWORLD ONTOLOGY. Several new conceptual models have been integrated into this ontology. All concepts have been implemented and tested, while several instances of UbisWorld servers are currently running, either locally or on the internet[28].

[28]UbisWorld can be tested at: `http://www.ubisworld.org/`

6 Situation Retrieval & Conflict Resolution

Information retrieval is the art and science of searching for information in documents, searching for documents themselves, searching for metadata which describe documents, or searching within relational databases or hypertext networked databases such as the Internet for text, sound, images or data, [Wikipedia, 2005].

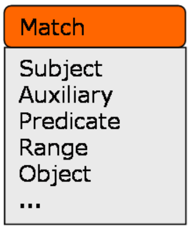
The newly defined term of *situation retrieval* describes specialized information retrieval within repositories and reports of SITUATIONALSTATEMENTS. In chapter 4 we have demonstrated that the model of the statements is independent from the actual used syntactical realization and in chapter 5 we have demonstrated that the model is even decoupled from the semantics which is moved to the ontology. For the information retrieval part this insight leads to the threefold relationship: one with the model, one with the syntax and one with the semantics. Since *context logs*, *user journals* and especially *user models* are also represented in this statement framework, situation retrieval directly covers the task of querying these logs, journals and models.

In this chapter, we first present the model of SITUATIONALQUERIES with its three groups of attributes. After that, we present the model of SITUATIONREQUESTS, followed by the query algorithm. Then, two sections about `UserQL` and conflict resolution follow. The architecture of situation retrieval with conflict resolution will be described in subsection 6.3.6, while the discussion on the design decisions for `UserQL` can be found in section 6.4. The integration into the overall architecture of the USERMODELSERVICE and into the procedural view to the USERMODELSERVICE will be explained in chapter 8 in the figures 8.1 and 8.2.

6.1 The Model of SITUATIONALQUERIES

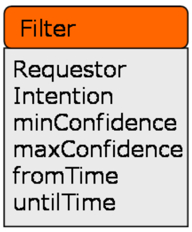
SITUATIONALQUERIES form the counterpart to SITUATIONALSTATEMENTS, since each situation attribute finds a corresponding attribute in the model of SITUATIONALQUERIES. SITUATIONALQUERIES consist of three identified groups of attributes: the `match`-attributes, the `filter`-attributes and the `control`-attributes. They are named after three macro-steps in the query evaluation process as displayed in figure 6.2. The `select`-attribute is modeled together with the `control`-attributes. The three groups of attributes are described as boxes below.

6.1.1 The Match Box



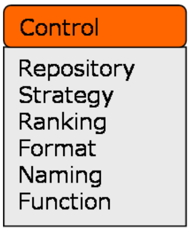
The match-attributes are named `subject`, `auxiliary`, `predicate`, `range`, `object` and so on, exactly after their corresponding attributes within the SITUATIONALSTATEMENTS. Actually any attribute from the model of statements can be used as match attribute in the model of queries. To differentiate between the two sets, the dot-notation is introduced as in `statement.subject` versus `query.subject`. When the *matching algorithm* is applied, only statements that match all given match-attributes are handed over to the *filtering algorithm*. Furthermore figure 6.4 - which shows the matching algorithm - reveals that it can be more complex than just the comparison for equality: semantic functionality like the *extension relation* can be applied. This relation maps to each class its subsumed instances. For example if the match attribute `query.subject` = *extension(*`person`*)* is given, all statements with `statement.subject` ∈ {Boris, Jörg, Margeritta, ...} are matched, namely instances that belong to the class (or the subclasses) of *person*. This elegant mechanism to integrate semantic inference into the syntax of the query language has already been prepared by the extended RDF resource concept of *UbisExpression*, as described in section 4.5 from page 74 onwards.

6.1.2 The Filter Box



Filter attributes set further restrictions on the returned statements. There are currently three filters implemented: a *privacy filter*, a *confidence filter* and a *temporal filter*. Figure 6.5 defines all three filters within the complete *filtering algorithm*. The attributes `query.requestor` and `query.intention` decide according to the attributes `statement.access`, `statement.purpose` and `statement.owner` if the privacy filter can be passed. If no permission is granted, the statement will be filtered out, even though it might contain the answer to the query. The privacy filter needs no activation since it is always activated. If no `requestor` and `purpose` attributes are stated in the query, the values `requestor`=*anonymous* and `purpose`=*commercial* are set as defaults, which results in a minimal access in privacy terms. The confidence attributes `minConfidence` and `maxConfidence` allow us to restrict the confidence value to a certain interval. For example it might be interesting to analyze only statements that carry a higher confidence value than 0.75 within the range [0, 1]. The confidence filter will be used automatically, if at least one of the attribute-value pairs `minConfidence`=*value* or `maxConfidence`=*value* is used. The *temporal filter* allows to put constraints on the statements' temporal aspects, since in our approach no statement is deleted, but either marked as *replaced* or marked as *expired* if it should not be used any more. This especially allows to neglect older statements with the attribute `fromTime`, or the other way round, to allow for historic views by filtering out all statements after a certain point of time with the attribute `untilTime`. The temporal filter can be activated by adding `fromTime`=*time* or `untilTime`=*time* to the query. To summarize, the filter box attributes lead to a fine grained mechanism for extending the expressivity of the situation retrieval process.

6.1.3 The Control Box



The attribute `repository` controls the `select`-step by defining the repository (or the set of repositories) to which the query is being applied to. The addressing schema to the decentralized and distributed `UserML`-repositories is defined in section 6.2.2. During the `control`-step, possible conflicts are resolved and the appearance of the returned `UserML` document is defined. The conflict resolution can be guided by the attributes `strategy` and `ranking` and the appearance of the result is influenced by the two attributes `format` and `naming`. The attribute `function` allows to post-process the so far calculated set of statements into any kind of result like an average number for example. The `control`-step procedures and especially the conflict resolution strategies are analyzed in section 6.3.

6.1.4 The Model of SITUATIONREQUESTS

A list of SITUATIONALQUERIES is called a SITUATIONREQUEST, see figure 6.1 for a schematic view. A SITUATIONREQUEST is sent to the user model and context service that resolves each query after the other in a row (or if possible in parallel) and returns the resulting SITUATIONREPORT.

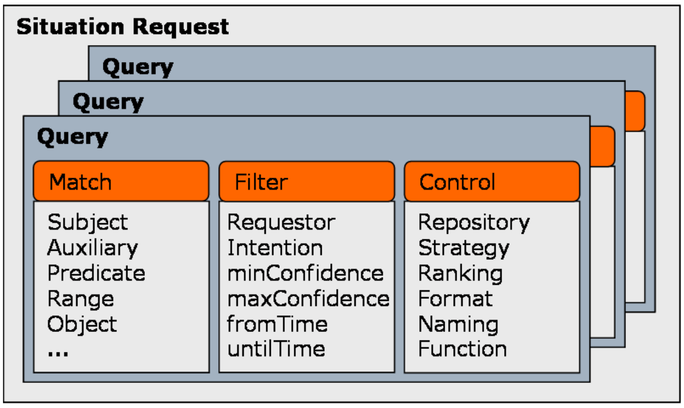


***Figure 6.1**: A SITUATIONREQUEST consists of a set of SITUATIONALQUERIES*

The naming convention of the two introduced concepts is defined in the following listing:

- A SITUATIONALQUERY is a collection of attributes as defined in table 6.1. The term QUERY without the prefix is used as shortcut.
- A SITUATIONREQUEST is a collection of SITUATIONALQUERIES, that belong to the same situational description. The term REQUEST without the prefix is used as shortcut.

Table 6.1 shows a detailed description of the intended attribute meanings within a SITUATIONALQUERY.

Match	*all attributes introduced for SituationalStatements*
`subject`	selecting the main statement entity, *default: any*
`auxiliary`	selecting the auxiliary part of the property, *default: any*
`predicate`	selecting the predicate part of the property, *default: any*
`range`	selecting the range part of the property, *default: any*
`object`	selecting the object, *default: any*
`id`	selecting the statement by id, *default: any*
`group`	selecting the group of statements, *default: UserModel*
`location`	selecting the spatial extension of the statements, *default: any*
⋮	see table 4.1 for a complete list of attributes since every situational statement attribute can be used as a matching attribute in the query

Filter	*a collection of filter attributes*
`requestor`	the requesting user or system, *default: anonymous*
`intention`	what is intended to be done with the statement, *default: commercial*
`minConfidence`	minimal confidence value that must hold, *default: 0*
`maxConfidence`	maximal confidence value that must hold, *default: 1*
`fromTime`	start of the time interval, *default: whenever*
`untilTime`	end of the time interval, *default: now*

Control	*a collection of control attributes*
`repository`	the chosen, respondent situation container, *default: system's choice*
`strategy`	conflict resolution strategies, *default: latestOnly*
`ranking`	sorting and ranking of the results, *default: newestFirst*
`naming`	manipulating the appearance of the names, *default: longName*
`format`	manipulating the appearance of the XML format, *default: UserML*
`function`	applying evaluation functions to the results, *default: none*

Table 6.1*: Attributes of SituationalQueries with default values*

6.1.5 The Query-Answer Algorithm

Figure 6.2 shows the concept of the macro-steps in the query evaluation process. The `select`-step chooses the report or repository to which the query is applied. The `match`-step returns all statements that match the corresponding query attributes. The `filter`-step filters out further unwanted statements, while the `control`-step performs conflict-resolution and transforms the final statements into the returned result. A query model as defined in section 6.1 needs to carry all the intended attributes and parameters to allow for this multi-step query evaluation process.

Figure 6.3 defines the query-answer algorithm as a sequence of the *selecting procedure*, *matching procedure*, *filtering procedure* and *control procedure*: after the selection of the repository (as shown in figure 6.2) has been done by the user model and context service in

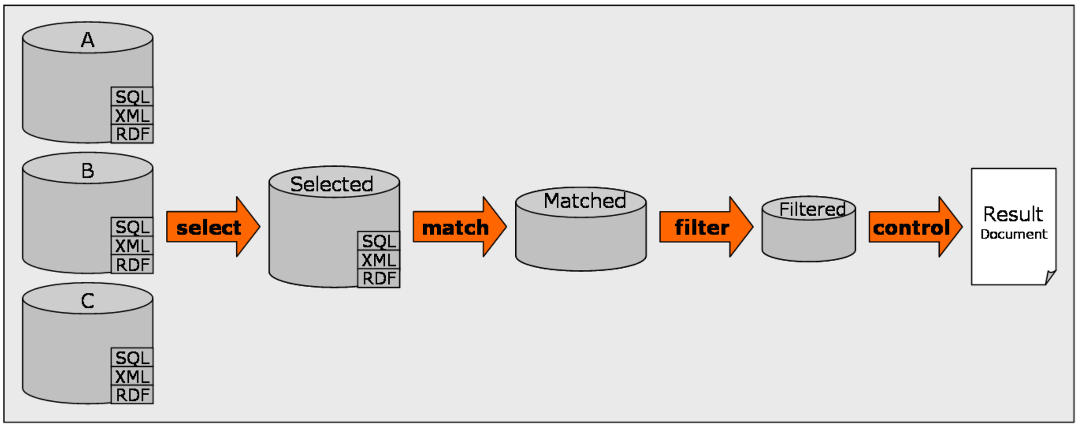


Figure 6.2: *Situation retrieval with the macro steps: Select, Match, Filter and Control*

the *selecting procedure*, the *answering procedure* linearizes the three remaining macro steps and returns the reduced set of statements in each step.

procedure *answering* (`query`, USERMODELSERVICE)
  SELECTINGRESULT := *selecting* (`query`, USERMODELSERVICE)
  MATCHINGRESULT := *matching* (`query`, SELECTINGRESULT)
  FILTERINGRESULT := *filtering* (`query`, MATCHINGRESULT)
  return RESULT := *control* (`query`, FILTERINGRESULT)

Figure 6.3: *The linearized answering procedure*

The *matching procedure* as shown in figure 6.4 compares all given match attributes with the corresponding statement attributes. Furthermore it integrates semantic functionality, here shown by the two examples: ontological *extension* and spatial *inclusion*. Further semantic functionality that could have interesting applications are for example the ontological *sameAs* and the spatial *closeBy*.

The *filtering procedure* as shown in figure 6.5 operates on the MATCHINGRESULT. Each statement is individually checked if it passes the *privacy filter*, the *confidence filter* and the *temporal filter*. The *privacy filter* checks if the `statement.access` is either set to public, or if it is set to friends, that the *friends relation* holds between the `query.requestor` and the `statement.owner`, or if it is set to private that the `query.requestor` is the same as `statement.owner`. Further filters can be added to this part of the algorithm. Especially a fine-grained spatial filter that covers more than the spatial *inclusion relation* (which is already realized by the match attributes) might become important.

The next section introduces two syntactical realizations of this SITUATIONALQUERY model.

```
procedure matching (query, SELECTEDREPOSITORY)
  forall statement ∈ REPOSITORY
    begin
      if statement.subject = query.subject
           or statement.subject ∈ extension( query.subject ))
         ∧ statement.auxiliary = query.auxiliary
         ∧ ( statement.predicate = query.predicate
           or statement.predicate ∈ extension( query.predicate ))
         ∧ statement.range = query.range
         ∧ statement.object = query.object
         ∧ statement.id = query.id
         ∧ statement.group = query.group
         ∧ statement.location = query.location
           or statement.location ∈ inclusion( query.location ))
         ∧ ...
      then add statement to MATCHINGRESULT
    end
  return MATCHINGRESULT
```

Figure 6.4: *The matching procedure with integrated semantic functionality*

6.2 The Syntax of `SituationQL` and `UserQL`

`SituationQL` defines instances of a *Situation Query Language* and the acronym `UserQL` stands for *User Model Query Language*. Both are syntactically equal and form the main concept in situation retrieval with SITUATIONALSTATEMENTS. They form the counterparts to `SituationML` and `UserML`. Since the USERMODELSERVICE which is introduced in section 8.1 is implemented as an HTTP web server, there exist the two methods *post* and *get* to send query information to the server. This results in two versions of the query language, namely `UserQL/XML` and `UserQL/URI`. The first one defines the queries in XML format and can be sent via the post method to the server. The second one defines the queries in URI format and can be sent via the get method to the server.

6.2.1 The `UserQL/XML` Query Language

The listing in figure 6.6 presents the default `SituationQL/XML` representation for SITUATIONALQUERIES. The root element is named `<query>`. All attributes are defined by XML-elements with their original name, while the corresponding attribute groups like `match` or `control` are omitted. The flat tree structure is the easiest for prototyping and can directly be transformed into a relational data model for database storage. SITUATIONAL-QUERIES define up to 37 attributes, however the average query will be short, since empty elements can be omitted. The variables from `q1` to `q37` can carry ordinary RDF node values but also more complex *UbisExpressions* as defined in section 4.5.

```
procedure filtering (query, MATCHINGRESULT)
  forall statement ∈ MATCHINGRESULT
    begin
      if ( statement.access = public
        or ( statement.access = friends
          ∧ query.requestor ∈ friends (statement.owner) )
        or ( statement.access = private
          ∧ query.requestor = statement.owner ))
      and statement.confidence ≥ query.minConfidence
        ∧ statement.confidence ≤ query.maxConfidence
      and statement.start ≥ query.fromTime
        ∧ statement.start ≤ query.untilTime
      then add statement to FILTERINGRESULT
    end
  return FILTERINGRESULT
```

Figure 6.5: *The filtering algorithm, using the externally defined friends-relation*

The XML application UserQL/XML forms only one concrete XML-instance of the model of SITUATIONALQUERIES. Syntactic variations of UserQL/XML could be defined if they promise advantages. Figure A.2 on page 215 shows for example the XML-Schema tree for an alternative realization.

UserQL/XML means that the query is represented in XML, but not necessarily the repositories. They are mostly stored in relational databases. However, XML repositories could be treated with XSL and XPath, a language for addressing parts of an XML document. The XPath data model provides a tree representation of XML documents. The result of an XPath expression may be a selection of nodes from the input documents, or an atomic value, or more generally, any sequence allowed by the data model, see W3C for a documentation.

6.2.2 The UserQL/URI Query Language

The UserQL/URI query language defines a structured subset of all possible URIs. Thus the attribute names and especially the attribute values have to fulfill the constraints that are implied by using URIs, see [Berners-Lee et al., 1998]. Figure 6.7 shows the partial definition of UserQL/URI in BNF-fromat, where the terminals are surrounded by ’-signs. In general, URIs allow us to define a list of attribute value pairs, the so-called *query string*, that is divided from the URL base with the ?-sign. The pairs are separated by &-signs, while the attribute name and the attribute value, a so-called *segment*, is syntactically separated by the =-sign. New in our extended query string approach for UserQL/URI is the additional evaluation of the |-sign as second-level list element within one single attribute value. The syntax is inherited from the EBNF grammar of UbisExpressions where for example the subject can be a |-sign-separated list of entities as in subject=Boris|Margeritta|Christian. The semantics for this list can be interpreted as logical **or**. The nested list structure and its semantics is defined in a relational model in section 7.1.1 on page 142.

```
<query>
      <!-- Match Attributes -->
      <subject>       q1   </subject>
      <auxiliary>     q2   </auxiliary>
      <predicate>     q3   </predicate>
      <range>         q4   </range>
      ...             ...  ...
      <unique>        q22  </unique>
      <replaces>      q23  </replaces>
      <group>         q24  </group>
      <notes>         q25  </notes>

      <!-- Filter Attributes -->
      <requestor>     q26  </requestor>
      <intention>     q27  </intention>
      <minConfidence> q28  </minConfidence>
      <maxConfidence> q29  </maxConfidence>
      <fromTime>      q30  </fromTime>
      <untilTime>     q31  </untilTime>

      <!-- Control Attributes -->
      <repository>    q32  </repository>
      <strategy>      q33  </strategy>
      <ranking>       q34  </ranking>
      <naming>        q35  </naming>
      <format>        q36  </format>
      <function>      q37  </function>
</query>
```

Figure 6.6: `SituationQL/XML` *is the default representation* SITUATIONALQUERIES

6.2.3 Example Queries with `UserQL/URI`

This subsection presents three example queries to the `u2m.org` USERMODELSERVICE in `UserQL/URI` format.

A) If one wants to ask *What is the age of Boris?* to the USERMODELSERVICE, one first has to identify the subject and the property of this question. In this example, the subject could be identified as "Boris" and the property could be described as "has age". Due to the design decisions in SITUATIONALSTATEMENTS, the property "has age" can be defined by three variables: the `auxiliary` *hasProperty*, the `predicate` *age* and a `range` like *integer* or *age group*. The default value for the range can also be defined in the general user model ontology GUMO and thus be omitted in the query. Figure 6.8 shows the identified attribute-value pairs. Figure 6.9 shows the query in `UserQL/URI` format.
String names like *Boris* or *age* are not case sensitive, it does not matter if they are written in upper case letters or lower case letters. However, since using the string names for referring to the intended elements is in general not sufficient to refer uniquely, the use of *UbisIds* as shown in figure 6.10 is also possible.
An interesting aspect about this user model request is that it could probably be answered without having a direct entry about *Boris' age* in the requested user model repository. The

```
UserQL/URI      --> 'http://www.u2m.org/service.php'
                    '? subject    =' Q1
                    '& auxiliary  =' Q2
                    '& predicate  =' Q3
                    ...
                    '& requestor  =' Q26
                    '& intention  =' Q27
                    ...
                    '& repository =' Q32
                    '& strategy   =' Q33
                    ...
                    '& format     =' Q36
                    '& function   =' Q37
Q1, ..., Q37    --> UbisExpression
```

Figure 6.7: *Partial EBNF grammar definition for* `UserQL/URI`*. The succeeding EBNF grammar for UbisExpression can be found in section 4.5.3 on page 77*

`query.subject` = *Boris*
`query.auxiliary` = *hasProperty*
`query.predicate` = *age*

Figure 6.8: *The attribute-value pairs that represent the query "What is the age of Boris?"*

USERMODELSERVICE could deduce this information for example from other information like *his birthday* or from the *ages of his elder and younger sisters*. Even though these two described inferences appear fairly simple they point out that `UserQL` queries applied to the USERMODELSERVICE differs from ordinary SQL queries applied to general databases. The interrelationship between the syntax and ontological reasoning in our approach will be discussed in chapter 7. On the other hand, instead of having no entry about *Boris' age* in the repository, there could be several conflicting entries by different `creators` with probably different `confidence` values. In such a case the USERMODELSERVICE has to resolve the answer by a given conflict resolution strategy. The challenge of conflict resolution is discussed in the following section 6.3.

B) If one is interested in a query like *What do you know about the personality of Margeritta?*, the predicate focuses on a whole class of user model dimensions, namely the *personality* entries like `extraversion`, `introversion`, `judging` or `perceiving`. The query syntax in `UserQL/URI` is the same as in example A), only that the resource identifier of the `predicate` attribute now refers to a class of concepts instead of to an instance. Figure 6.11 shows the corresponding URI request.

These requests are related to the structured query language SQL queries. Since the whole SITUATIONALSTATEMENTS are returned, it could be compared to a *SELECT * FROM (http://www.u2m.org/service.php) WHERE subject=Margeritta AND auxiliary=hasProerty AND predicate=Personality*. However, apart from the syntactical difference, there is a difference is the evaluation of the last attribute-value pair, since `Personality` is detected as

```
http://www.u2m.org/service.php?
subject=Boris&auxiliary=hasProperty&predicate=Age
```

Figure 6.9: *The* `UserQL/URI` *that represents the query "What is the age of Boris?"*

```
http://www.u2m.org/service.php?
subject=210002&auxiliary=600100&predicate=800302
```

Figure 6.10: *The* `UserQL/URI` *that also represents the query "What is the age of Boris?"*

a class and not as an individual, the ontological *extension* function is executed and the output result would represent the following triples, (compare with figure 8.13):

```
subject=Margeritta, predicate=open-minded, object=high
subject=Margeritta, predicate=indulgent, object=low
subject=Margeritta, predicate=tempered, object=high
subject=Margeritta, predicate=optimistic, object=medium
```

C) In the request *Tell me all interests of Christian!* one can omit the `predicate` attribute because the `auxiliary` attribute is sufficient, as shown in figure 6.12. Omitting an attribute in general means that there are no restrictions set on this attribute. Thus, if one omits all attributes, all entries of the USERMODELSERVICE will be returned according to the default `strategy`, `format` and `naming` settings.

6.2.4 How to Manipulate the Output Format?

This subsection presents some settings for manipulating the output format of the returned document. The two query attributes `query.format` and `query.naming` are discussed. The overall structure of the XML document can be chosen with the attribute `format`, while the attribute `naming` changes the appearance of the variables. "Boris' Age Example" with the chosen naming convention `nameWithParent` and the selected format `UserMLrdf` is shown in figure 6.13 below.

Table 6.2 shows the predefined values for the `UserQL` attribute `naming`. Compare the section about extended resource identification 4.5 on page 74 and the XML-Schema of SITUATIONREQUESTS in appendix A.3.3 on page 215.

The default setting for the `naming`-attribute is `longName`, which combines the written name and the id with a dot. Further examples for such "long names" are `BloodPressure.800131` and `Gender.800300`. Table 6.2 shows the predefined values for the `UserQL` attribute `format`. Compare the section about the syntax of `SituationML` and `UserML` in section 4.4 on page 67.

```
http://www.u2m.org/service.php?
subject=Margeritta&auxiliary=hasProperty&predicate=Personality
```

Figure 6.11: *This URI represents the request "Tell me all about the personality of Margeritta"*

```
http://www.u2m.org/service.php?
subject=Christian&auxiliary=hasInterest
```

Figure 6.12: *This `UserQL/URI` represents the query "Tell me all interests of Christian"*

```
http://www.u2m.org/service.php?
subject=Boris&auxiliary=hasProperty&predicate=Age
&naming=nameWithParent&format=UserMLrdf
```

Figure 6.13: *"What is the age of Boris?" with an alternative `naming` and `format` attribute*

Attribute	**Value**	**Description**
`naming`	`id`	unique numerical identifier in the format of a positive integer between 100000 and 999999
`naming`	`label`	textual identifier, possibly with blanks
`naming`	`shortName`	transformed `label` without blanks and only uppercase letters at the beginning of each new word
`naming`	`longName`	*(default)* combines the `shortName` with the `id` in the following way: "`shortName.id`"
`naming`	`nameWithParent`	combines the `shortName` with the `shortName` of the parent element as prefix: "`shortName`*(parent)*`.shortName`"
`naming`	`resource`	unique RDF resource, combining a URI with the fragment identifier `longName`: "*URI*`#longName`" or an RDF literal

Table 6.2: *Possible values for the `UserQL` attribute `naming`*

Attribute	**Value**	**Description**
`format`	`UserML`	*(default)* `UserML` definition where the `<statement>` element carries all information in its up to 25 subelements
`format`	`UserMLmin`	minimal variation of `UserML` where the `<statement>` element carries all information in its attributes
`format`	`UserMLmax`	XML application with the main element `<statement>` and its five subelements `<mainpart>`, `<situation>`, `<explanation>`, `<privacy>` and `<administration>`.
`format`	`UserMLrdf`	RDF representation of the same semantic information as in `UserML`
`format`	`PredicateObject`	returns only `predicate-object` pairs as a newlined list

Table 6.3: *Possible values for the query-attribute `format`*

6.3 Conflict Resolution

In widely spread distributed user modeling and context awareness one can not expect that all systems use the same standards for representing user models and context information. However, as a common basis and to narrow down the research area, we assume that all systems, that are involved in the information exchange, are able to use the framework of SITUATIONALSTATEMENTS: either as their main representation language or at least as an additional transforming feature. Even though we assume that all involved user-adaptive and context aware systems are supposed to communicate with the same *grammatical* framework of `SituationML` and `SituationQL`, there is still a whole list of interesting points of possible misunderstandings left to be solved. Systems that do not support `SituationML` and `SituationQL` can still cooperate with such systems on the level of SITUATIONAL-STATEMENTS in such a way that they consume or contribute new statements, or they could be connected with the help of specialized XSL transformations that have to be designed individually.

6.3.1 Situational Conflict Categories

As every user and every system is allowed to enter statements into repositories, some of this information might be contradictory. Conflicts among SITUATIONALSTATEMENTS like for example a contradiction caused by different opinions of different creators or changed values over time are loosely categorized in the following listing.

1. ON THE REPRESENTATIONAL LEVEL:
   each system can choose between a variety of possible representations to express the same information which leads to the so-called *variation mapping*

2. ON THE SYNTACTICAL LEVEL:
   statements can for instance differ in the use of the statement attributes like `subject`, `predicate`, `object`, `range`, `start` etc., clear *modeling guidelines* are necessary.

3. ON THE SEMANTICAL LEVEL:
   the systems are not forced to use the same vocabulary, that is to say the same ontology, to represent the meaning of the concepts, which leads to the user model integration problem number one: *ontology merging* and *semantic web integration*.

4. ON THE OBSERVATION AND INFERENCE LEVEL:
   several sensors can see same things differently and claim to be right, measurement errors can occur, systems may have preferred information sources

5. ON THE TEMPORAL AND SPATIAL LEVEL:
   information can be out of date or out of spatial range, a degree of expiry can hold, thus reasoning on temporal and spatial meta data becomes necessary

6. ON THE PRIVACY AND TRUST LEVEL:
   information can be hidden, incomplete, secret or falsified on purpose, a system of trustworthiness could be applied

All these aspects are content- and domain independent and can be relevant for user related as well as context related information. In the remaining part of this chapter, several conflicts are analyzed and categorized and possible solutions are presented. In relation to the situation retrieval task, the two concepts of *precision (How many retrieved statements are really relevant?)* and *recall (Is all relevant information retrieved?)* are also taken into consideration. This section is partly based on the research described in [Blass, 2004] and on ideas about meta rules and conflict resolution in OPS5, see for example [Brownston et al., 1985]. *Conflict resolution* forms the first part of the *control* macro-step as shown in figure 6.2.

6.3.2 Prefacing Examples with Conflicting Statements

A) Imagine you should play chess against user A and you have the following rule of thumb:

if the opponent is a good player, you start with a defensive opening
but if the opponent is a weak player, you choose an offensive opening.

You have heard the three following, conflicting statements about user A's ability to play chess: System C claims that A is a good player, system B claims that A is a medium good player while user A claims that she/he himself is a bad player. Figure 6.14 visualizes these

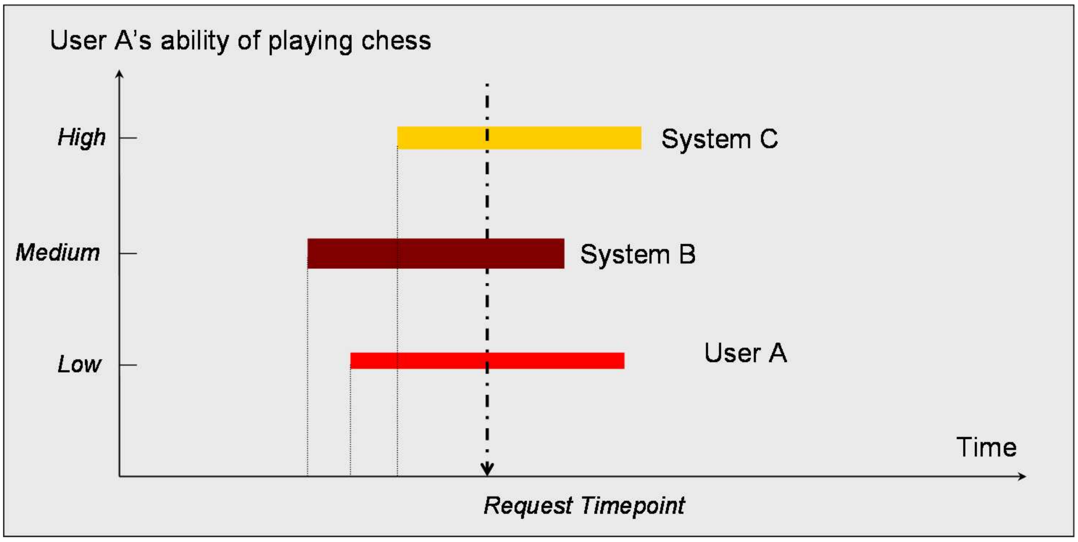


***Figure 6.14**: Example visualization of three conflicting statements*

three statements as rectangles that also indicate additional meta data dimensions: the time-axis shows the `start`, `end`, `durability` and expiry values which reveals that system B's statement is the oldest one, while system C's statement is the most recent one. The height of the rectangle relates to the `confidence` value: the confidence of system B's statement is the highest while the user's statement has the lowest confidence. So, how should we decide?

How do we resolve such conflicts? Naive approaches could ask: should we return the latest entry? Should we return a random element? Should we return the one with the highest confidence value or should we prefer the user's entry? Conflicting statements turned out to be a complex and general problem but with no general solution that is valid for all situations. The measures of belief that are based on evidences in the *Dempster-Shafer theory*, see section 4.2, could lead to a sophisticated solution. However, in our approach we support the means to express the intended conflict resolution methods as described in the next subsection. Conflicts on the semantic level are presented in the the following example.

B) Let us assume that a resource-adaptive mobile device uses the following energy saving and contrast optimizing adaptation rule if its device's battery is low:

if the surrounding brightness is low then set the display brightness to its minimum,
if the surrounding brightness is medium or high then set the display brightness to medium.

Since the mobile device has no light sensor by itself, it needs to retrieve and infer information about the *surrounding brightness* from the user model and context service. The current position of the mobile device should be *Room124* and the light situation there is given by the semantically conflicting statements:

1) `subject=`*Room124* `auxiliary=`*hasProperty* `predicate=`*brightness* `range=`*lowHigh* `object=`*low*
2) `subject=`*Saarbrücken* `auxiliary=`*hasProperty* `predicate=`*weather* `object=`*sunny with no clouds*
3) `subject=`*Light124.1* `auxiliary=`*hasProperty* `predicate=`*switched* `range=`*onOff* `object=`*on*

Figure 6.15: *Example statements that form a semantic conflict*

The most direct statement 1) about the brightness in *Room124* claims that it is dark. However, statement 2) claims that the sun is shining. Since the device is inside a building which could hinder the sun to lighten the surroundings the situation is not clear. Furthermore statement 3) claims that the lights in this room are switched on. This conflict is difficult to detect since ontological reasoning, spatial reasoning and qualitative reasoning might be necessary to handle this problem satisfactorily. Subsection 6.3.5 introduces a mechanism to detect such semantical conflicts.

6.3.3 Conflict Resolvers and Conflict Resolution Strategies

Conflict Resolvers are a special kind of filter that control the *conflict resolution process*. An ordered list of these resolvers defines the *conflict resolution strategy*. They are modeled in the `query.strategy` attribute. These resolvers are needed if the *match process* and *filter process* leave several conflicting statements as possible answers. Three kinds of conflict resolvers can be identified: the *most(n)*-resolvers that use meta data for their decision, the *add*-resolvers that add expired or replaced statements to the conflict sets, and the *return*-resolvers that don't use any data for their selection.

mostRecent(n) Especially where sensors send new statements on a frequent basis, values tend to change quicker than they expire. This leads to conflicting non-expired statements. The *mostRecent(n)* resolver returns the *n* newest non-expired statements, where *n* is a natural number between 1 and the number of remaining statements.

mostNamed(n) If there are many statements that claim A and only a few claim B or something else, than *n* of the "most named" statements are returned. Of course it is not certain that the majority necessarily tells the truth but it could be a reasonable rule of thumb for some cases.

mostConfident(n) If the confidence values of several conflicting statements can be compared with each other, it seems to be an obvious decision to return the *n* statements with the highest confidence value.

mostSpecific(n) If the `range` or the `object` of a statement is more specific than in others, the *n* "most specific" statements are returned by this resolver. For example if: `auxiliary`=*hasKnowledge*, `predicate`=*chess* and first `range`=*yesNo* while the second `range`=*Novice-Occasional-Professional-Expert-Grandmaster*, the statement with the second range contains a more specific information. Another specificity range ordering is for example: *yesNo* < *lowMediumHigh* < *0%-100%*

mostPersonal(n) If the `creator` of the statement is the same as the statement's `subject` (a self-reflecting statement), this statement is preferred by the *mostPersonal(n)* resolver. Furthermore, if an *is-friend-of relation* is defined, statements by friends could be preferred to statements by others. However, this resolver bears the problem, that users might not be their best judge. However due to privacy arguments, the user's own statements that are given (on purpose) should be preferred. (An alternative approach with the `creator` information could have been to define a *trusted-creator relation*.)

addExpired Per default the already expired statements are filtered out. However, if one wants to take them into consideration, the *addExpired*-resolver adds these statements to the conflict sets.

addReplaced Statements that are marked with the replaced-flag by other statements, are also per default filtered out and not considered in the situation retrieval process. The *addReplaced*-resolver brings these statements back into the process.

addPrivate Statements that do not pass the privacy settings are always filtered out. However, for development, testing and administrative reasons experimental private statements may also be recognized with the *addPrivate*-resolver.

returnAll If the remaining conflict set should not be resolved any further by the integrated mechanism, the resolver *returnAll* returns all remaining statements that can then be resolved by an external conflict resolution method, resolved by introspection or left unresolved since our approach also allows conflicting extensions in parallel.

returnNone If there still occurs a conflict that could not be resolved until the *returnNone* resolver is applied, no statement is returned. This is a very safe way not to say something wrong. This rule could be compared with *sceptical inheritance* in non-monotonic reasoning: *I don't know!*

returnRandom(n) if after applying several filters still no unique value is found but a unique answer is expected, a random pick will be offered by this resolver. This credulous behavior is selected by the requestor and therefore acceptable.

returnDialog if no unique value is found, an alternative conflict resolution strategy could be *clarification by dialog*[1]. In some cases an appropriate human-computer dialog will be initiated in this case.

These conflict resolver rules are based on common sense heuristics. An important issue to keep in mind is the problem that resolvers and strategies imply uncertainty. To contribute to this, the `confidence` value of the resulting statement is appropriately changed, furthermore the conflict situation is added to the `evidence` attribute. Further ideas like calculating the *average* value or the *maximum* value of the statement's `object` can be covered by the `function` attribute that allows to apply mathematical functions to the value of either one statement or a set of statements. The *function evaluation* forms the last part of the *control procedure*. Figure 6.16 shows several examples of conflict resolution strategies. In general, a strategy can be defined by every combination of resolvers, but not all make sense.

```
1.) strategy=returnAll
2.) strategy=mostRecent(1)
3.) strategy=mostRecent(4)|mostConfident(3)|returnRandom(2)
4.) strategy=mostNamed(5)|mostConfident(3)|mostSpecific(1)
5.) strategy=mostPersonal(3)|mostConfident(2)&function=average
```

***Figure 6.16**: Examples for conflict resolution strategies*

The first conflict resolution strategy is the empty one, which means that no conflict resolution will be applied and all statements that pass the match and filter process will be returned. Note, if the `query.strategy` attribute is left empty, a default strategy will be applied but not necessarily the *returnAll* one. The second conflict resolution strategy is adequate for statements that are frequently renewed by only one sensor, thus simply the last entry is returned. The third strategy is slightly more complex: the four most recent statements are handed over and checked for the three most confident ones. However, only two statements of these three are returned by random selection. The fifth and last conflict resolution strategy first selects the most personal ones and then the most confident ones and finally applies the average `query.function` to calculate the average value of the two remaining statement's `object`. This strategy could be interesting for user model dimensions that do not change over time like *personality traits*.

If we revisit the prefacing example of figure 6.14 and apply the strategies of figure 6.16, the second and the third strategy would return *high*, the fourth strategy would return *medium* while the fifth strategy would return *low*. However, the first strategy returns all three values but is no help in this situation. These completely different results show the power of the conflict resolution strategies, however they also indicate that not all conflicting statements can be satisfactorily solved by resolution strategies. As different strategies lead to different statements and resulting values, the choice of the "right" conflict resolution strategy isn't that obvious.

[1]The idea of *clarification by dialog* was recommend by Vania Dimitrova.

Different classes of problems seem to need different conflict resolution strategies. The open question is if there is any correlation of the best strategy in certain circumstances. We expect that already the user model dimension equivalence classes correlate with the best strategies. If so, this information could be added to the general user model ontology GUMO and the problem could be solved with the introduction of a new level of meta rules.

6.3.4 Conflict Ranking for Retrieved Statements

If several statements are returned, the `query.ranking` attribute controls the order of the statements. Three example rankings are presented in figure 6.17. They show that the same terms are used in the `query.ranking` and in the `query.strategy` attribute, however with a different impact.

1.) `ranking=mostRecent`
2.) `ranking=mostRecent|mostConfident`
3.) `ranking=mostNamed|mostSpecific|mostPersonal`

Figure 6.17: *Example rankings for situation retrieval*

The first ranking orders the resulting statements according to a temporal axis, the second ranking adds to the temporal axis the second level ordering on a confidence axis, and the third example ranking orders the statements under completely different settings. The so far discussed conflict resolution topics left the semantics of the statement's `mainpart` out of consideration. The following subsection integrates both: conflict resolution with semantic analysis.

6.3.5 Conflict Detection and Classification of Semantical Conflict Sets

In the prefacing chess example of subsection 6.3.2 the conflict was easy to detect since all three statements used the same `subject`, the same `auxiliary`, the same `predicate` and even the same `range`=*low-medium-high*. If one statement used instead the `range` *poor-average-good-excellent* a so-called SEMANTICRANGEMAPPING had to be executed, where for example *poor* is mapped to *low*, *average* and *good* are mapped to *medium*, and *excellent* is mapped to *high*. A more tricky problem is how can we detect conflicts if even the `auxiliary` and `predicate` in the conflicting statements (or in the query and the statements) differ even though the same ontology is used? For example, system B could use `auxiliary`=*hasKnowledge* and `predicate`=*chess* while system C uses `auxiliary`=*hasAbility* and `predicate`=*boardGames*. In such a case, we already need SEMANTICPROPERTYMAPPING[2]. See [Ram and Park, 2004] for a discussion of a semantic conflict resolution ontology. A closely related problem arises if statements refer to different ontologies. In such a case we additionally need ONTOLOGYMAPPING in order to be able to detect and solve these conflicts.

[2] PROPERTY is used here to denote the pair of `auxiliary` and `predicate`

For the challenge of detecting possible conflicts we use a technique which is also used in production systems: we classify statements into so-called *conflict sets*. Mathematically speaking these statements are classified into equivalence classes. The following listing analyzes several types of such equivalence classes for conflict resolution with SITUATIONALSTATEMENTS and arranges them into six levels with increasing computational load.

Conflict Detection Class 0: $\langle S, A, P, R \rangle$

Statements with the same `subject` (S), `auxiliary` (A), `predicate` (P) and `range` (R) are called to be in a potential conflict of level 0. These conflicts are easy to detect since only a syntactical comparison is necessary. However, every fast changing user model dimension like *heart beat* for example would constantly produce potential conflicts and thus the produced conflict sets are not very useful.

Conflict Detection Class 1: $\langle S, A, P, R \rangle^{nonExpired}_{nonReplaced}$

The problem of the *conflict detection class 0* is solved by removing all statements that are already expired or replaced. The following simple conflicts that only differ in the `object` value can be detected with the *conflict detection class 1*:

1.1) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*happy* `range=`*yesNo* `object=`*yes*
1.2) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*happy* `range=`*yesNo* `object=`*no*

Conflict Detection Class 2: $\langle S, A, P, R^* \rangle^{nonExpired}_{nonReplaced}$

The conflict sets that are defined by the *conflict detection class 2* only match against `subject`, `auxiliary` and `predicate` while the `ranges` R^* are transformed into each other with the SEMANTICRANGEMAPPING. The mappings between different ranges are defined in the GUMO ontology. The following two conflicting example statements differ in `range` and `object`.

2.1) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*happy* `range=`*yesNo* `object=`*yes*
2.2) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*happy* `range=`*lowHigh* `object=`*low*

Conflict Detection Class 3: $\langle S, A, P^*, R^* \rangle^{nonExpired}_{nonReplaced}$

The conflict sets that are defined by the *conflict detection class 3* can differ in the `predicate`, `range` and `object` values. P^* indicates that related predicates are classified together. This relation could for example be defined as *synonyms* (other predicates with the same meaning) or *hypernyms*[3] (concepts with a broader meaning) or *holonyms* (the whole concept of which the predicate is part of) or *opponyms* (concepts with opposite meanings like *happy* versus *sad* or extraverted versus introverted).

3.1) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*happy* `range=`*yesNo* `object=`*yes*
3.2) `subject=`*Tim* `auxiliary=`*hasProperty* `predicate=`*sad* `range=`*lowHigh* `object=`*high*

Conflict Detection Class 4: $\langle S, A^*, P^*, R^* \rangle^{nonExpired}_{nonReplaced}$

The *conflict detection class 4* is only a slight extension to level 3, where additionally the `auxiliary` attributes are semantically related. For example *hasLearned* and *hasKnowledge* are closely related. Also, if someone *hasInterest* in something, one can assume that at least a rudimentary *hasKnowledge* statement also holds. Furthermore if

[3] Compare the WordNet relations that are defined in subsection 3.3.3 on page 45

a *hasPreference* relation holds, a *hasInterest* relation should also hold. However, this level 4 has not yet been used for concrete conflict resolution.

Conflict Detection Class 5: $\langle S^*, A^*, P^*, R^*\rangle^{nonExpired}_{nonReplaced}$
Interesting about level 5 is that the statements can completely differ syntactically but still be recognized as being semantically conflicting. The newly extended S^* integrates the modeling of user groups. If the `subject` defines a group or is member of a group, the corresponding GROUPMEMBERMAPPING has to be applied in order to handle conflicts between "member-statements" and "group-statements". Furthermore S^* allows to handle the problem that one user is often identified in different systems with different logins or nicknames[4].

The denotations $\langle S, A, P, R\rangle$ to $\langle S^*, A^*, P^*, R^*\rangle^{nonExpired}_{nonReplaced}$ are called the *signatures* of the corresponding equivalent classes.

To summarize, in this subsection we have demonstrated that the complex task of detecting conflicting statements in a repository can be analyzed with a fine-grained definition of conflict sets, that represent syntactical as well as semantical conflicts. Ontological reasoning is mostly integrated by using semantic mapping functionality. However, further research (like the experimental analysis of the conflict detection classes) has to be done at this point.

6.3.6 Architecture of Smart Situation Retrieval

The architectural diagram in figure 6.18 shows the SMARTSITUATIONRETRIEVAL process. The focus is set on the semantic conflict resolution part. It represents the `control` macro-step of the query process as shown in figure 6.2.

The oval numbers indicate the reading direction.

(1) shows the request in `UserQL` that has to be parsed first.

(2) points to the distributed retrieval of SITUATIONALSTATEMENTS.

(3) summarizes the three macro-steps `select`, `match` and `filter` and presents the FILTERINGRESULT as input to the conflict resolution process.

(4) stands for the three syntactical procedures VARIATIONMAPPING, REMOVEEXPIRED and REMOVEREPLACED.

(5) shows the three semantical procedures GROUPMEMBERMAPPING, SEMANTICPROPERTYMAPPING and SEMANTICRANGEMAPPING

that are based on knowledge in the ontologies GUMO, UbisWorldOntology, SUMO/MILO and the knowledge base WordNet.

(6) shows the detection of conflicts and the construction of $\langle S^*, A^*, P^*, R^*\rangle^{nonExpired}_{nonReplaced}$ conflict sets.

(7) points to the post-processing of `ranking`, `format`, `naming` and `function` that control the output format.

(8) forms the resulting `UserML` report, that is sent via HTTP to the requestor.

[4] S^* multi-name example: In one implemented ubiquitous user modeling scenario, the positioning system (see section 9.4.2) used the device's IP for identification, UbisWorld used an internal UbisId, YAMAMOTO (see section 8.12) modeled it by URI while the SPECTER shopping assistant also used a different ontology representation.

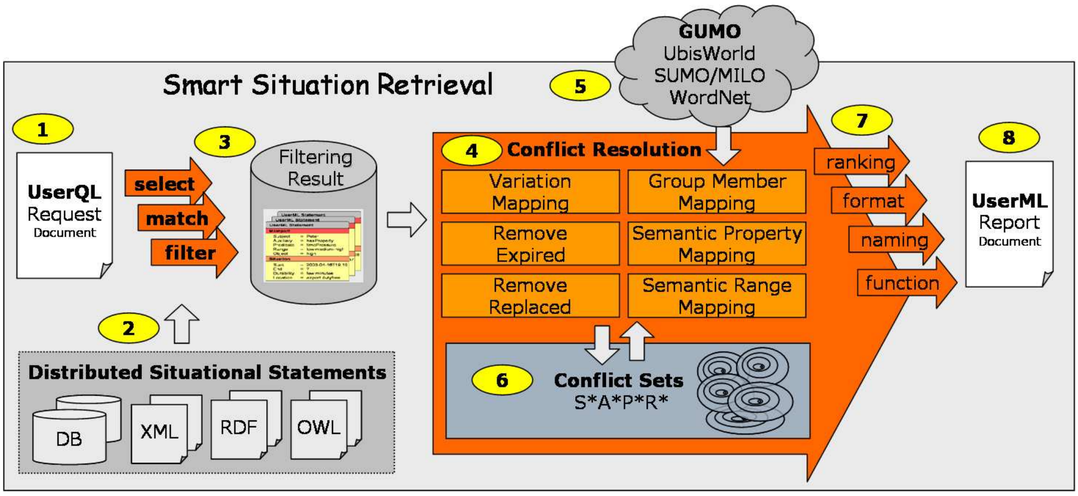


Figure 6.18: *Smart situation retrieval with focus on semantic conflict resolution*

6.4 Discussion and Design Decisions

In summarizing this chapter about information retrieval and conflict resolution, we will now discuss our design decisions: one design idea behind the user model query language `UserQL` was that it should be syntactically simple (realized as a flat structure) and close to the user model markup language `UserML`. The knowledge about the functionality should analogously be defined in an ontology such that the semantics of the queries' attributes and the statements' attributes are uniform. A tricky challenge was the integration of constraints of an XML query language, SQL query language and URI query language, since all three instances had to be realized by the general model of SITUATIONALQUERIES. However, the most important idea was to integrate syntactic and semantic conflict resolution into the query language.

Even though the so-called SMARTSITUATIONRETRIEVAL integrates various techniques of information retrieval and ontological reasoning, the dominating point of view should still be *information retrieval* and not the *ontology*. That is to say the query process which is mostly applied to databases with SITUATIONALSTATEMENTS dominates in our approach the ontological inference. Our situation models, that are part of the A-boxes, may be full of inconsistencies that need not to be resolved in general since we do not intend to change the statement repositories into one consistent world model. The statements that have been collected and sent around won't be changed or deleted. The reason is that in the highly distributed architecture, which is described in the next section, information is passed on and in most cases, we don't have access rights to edit the various instances of the statements. However, we allow to comment on them by reifying a statement into a new one and we allow a statement to be marked as being replaced. Furthermore we allow hiding them with the integrated privacy mechanism. The first advantage is that each user and each requesting system can individually choose the preferred conflict strategies and receive its individual model. The second advantage is demonstrated in the next chapter where we see that the integration of partial, decentralized user models can directly be realized using the SMARTSITUATION-

RETRIEVAL. The third advantage is that privacy handling is not the only mechanism that leads to individual models which results in better privacy.

The challenge that different users and different systems may want to use individualized ontologies that control the inferences could be solved in the following way: a new attribute with the name `ontology` is added to the *Control Box* (as defined on page 121) that carries one or several URIs pointing to the semantic web representations of these ontologies. This means that not only the set of statements differ individually and the conflict resolving strategies differ individually but also the inference ontologies can be chosen individually.

Another issue: how should we apply for example a confidence filter, if several statements in a conflict set don't carry confidence values? Or how should we apply the recency filter if the temporal constraints are not given? The answer is that the conflict resolution can only be as good as the available set of statements.

7 USER MODEL INTEGRATION & DECENTRALIZED USER MODELING

User Model Integration and *Decentralized User Modeling* are represented in figure 7.1 as the topmost brick in this stack model, that represents the complete conceptual overview over our implemented ubiquitous user modeling approach:

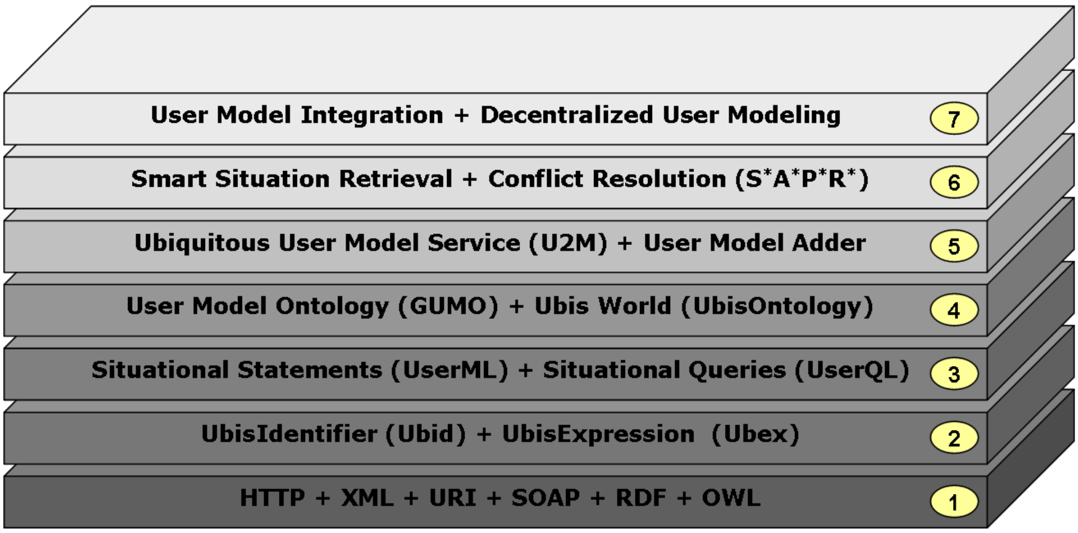


Figure 7.1: *Conceptual stack model of our ubiquitous user modeling approach*

On the bottom, brick number (1) forms the fundament. It represents the W3C internet standards, the web service standards and the semantic web standards. Brick number (2) extends the URI naming convention especially for ubiquitous computing. The two concepts `UbisIdentifier` and `UbisExpression` are introduced. They influence the next level of syntactic representation. Brick number (3) refers to the model and the syntax of SITUATIONALSTATEMENTS and SITUATIONALQUERIES. On top of this, the semantic level with the general user model ontology GUMO and the UBISONTOLOGY is presented by brick number (4). Number (5) stands for the ubiquitous user modeling service `U2M`, the *User Model Adder* and the distributed set of SITUATIONREPORTS. Brick number (6) denotes the

Smart Situation Retrieval and the *Semantic Conflict Resolution*. An interesting aspect of this highly modularized approach is that each module can be individually extended or even replaced without the need to change the others. And now, when all these modules are defined, the *User Model Integration* and the *Decentralized User Modeling* can be realized efficiently as shown in the following sections.

7.1 User Model Integration

User Model Integration can have several readings. The first one is the *merging of partial user models*, either from different user-adaptive systems, from different points of time in history or from different user model repositories. This reading leads to the general problem of *reusability* of user modeling data. The second reading is the integration of user models or SITUATIONREPORTS into the ontological representation of the world model (T-Box and A-Box of UBISONTOLOGY), in order to apply ontological reasoning. It could be called *ontology integration*. Solutions to both readings are presented in the succeeding subsections. But first, an adequate denotation to access every single element within large distributed sets of SITUATIONREPORTS is defined.

7.1.1 Denotation Model for Distributed SITUATIONREPORTS

A denotation model for situational information is presented in figure 7.2. It is based on the grammar-defined vocabulary of `UbisList`, see section 4.5.4 on page 79, which is a relational model of SITUATIONREPORTS and SITUATIONALSTATEMENTS. The new term of situation DISTRIBUTION is coined. It describes a distributed, but interconnected set of repositories.

$$
Denotation: \begin{cases}
Let\ \mathbb{D} = \{D_1, ..., D_d\} & \text{be the set of DISTRIBUTIONS} \\
Let\ \mathbb{R} = \{R_1, ..., R_r\} & \text{be the set of SITUATIONREPORTS} \\
Let\ \mathbb{S} = \{S_1, ..., S_s\} & \text{be the set of SITUATIONALSTATEMENTS} \\
Let\ \mathbb{A} = \{A_1, ..., A_a\} & \text{be the ordered set of statement attributes} \\
Let\ \mathbb{E} = \{E_1, ..., E_e\} & \text{be the set of UbisElements} \\
 & \\
D_{[h]} \rightarrow \{R_1, ..., R_w\} & \text{a distribution is a set of repositories} \\
R_{[i]} \rightarrow \{S_1, ..., S_x\} & \text{a repository is a set of statements} \\
S_{[j]} \rightarrow \{A_1, ..., A_y\} & \text{a statement is a list of named attributes} \\
A_{[k]} \rightarrow \{E_1, ..., E_z\}, & \text{an attribute is an ordered list of elements.} \\
 & \\
S_{(R=i,S=j)} & \text{denotes the } j^{th} \text{ statement of the } i^{th} \text{ repository} \\
S_{(R=i,S=j,A=k)} & \text{denotes the } k^{th} \text{ attribute of the } j^{th} \text{ statement of the } i^{th} \text{ repository} \\
S_{(R=i,S=j,A=k,E=l)} & \text{denotes the } l^{th} \text{ element of the } k^{th} \text{ attribute of the } j^{th} \text{ statement ...} \\
 & \text{with } i \in \{1,..,w\},\ j \in \{1,..,x\},\ k \in \{1,..,y\} \text{ and } l \in \{1,..,z\}
\end{cases}
$$

Figure 7.2: *Denotation model for the concept of distributed situational information*

7.1.2 User Model Merging and Reusability

We developed the user model exchange framework to enable decentralized systems to communicate about user models. The idea was to spread the information (if wished and allowed) among all user-adaptive systems, either with a mobile device or via ubiquitous networks. But the remaining question on the next higher level is about the merging and semantic integration of different user models. In [Orwant, 1996] the synchronization between the home computer user model and the work computer user model has already been suggested. Nowadays, data synchronization between mobile devices and stationary devices is omnipresent. However, the merging of partial, decentralized user models needs semantic reasoning similar to the one described within semantic conflict resolution. And indeed, since we implemented [Kay, 1995]'s accretion concept, our framework is able to reduce the *user model merging* task to the *semantic conflict resolution* task.

Figure 7.3(a-b) decomposes the general user modeling process schema of figure 2.2 on page 15 since *model communication* and *model integration* can take place between *model acquisition* and *model application*. Figure 7.3(c) shows the merge inference module: two user models are integrated into one.

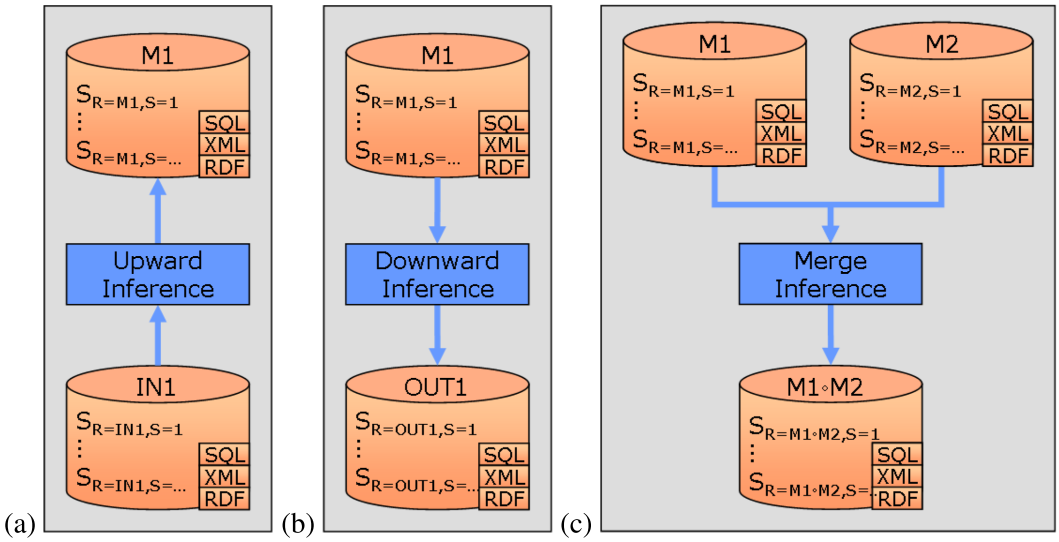


***Figure 7.3**: Upward inference, downward inference and merge inference*

Instead of inferring two user models individually as shown in figure 7.4(a) and then merging both user models, it could be more efficient in some cases to combine the different repositories and apply an integrated upward inference step, as presented in figure 7.4(b). Here, the inferential integration is automatically done by filters and conflict resolution strategies.

However, a new problem that occurs is the one of *double entries*. Sine the conflict resolution strategies are sensible to how often a statement is independently given (e.g. the *mostNamed filter*), we have to pay attention. So-called *Communication Doubles* refer to doubled statements that occur by communication and that do not imply further evidence to support the message. On the other hand, *Reinforced Doubles* refer to doubled statements,

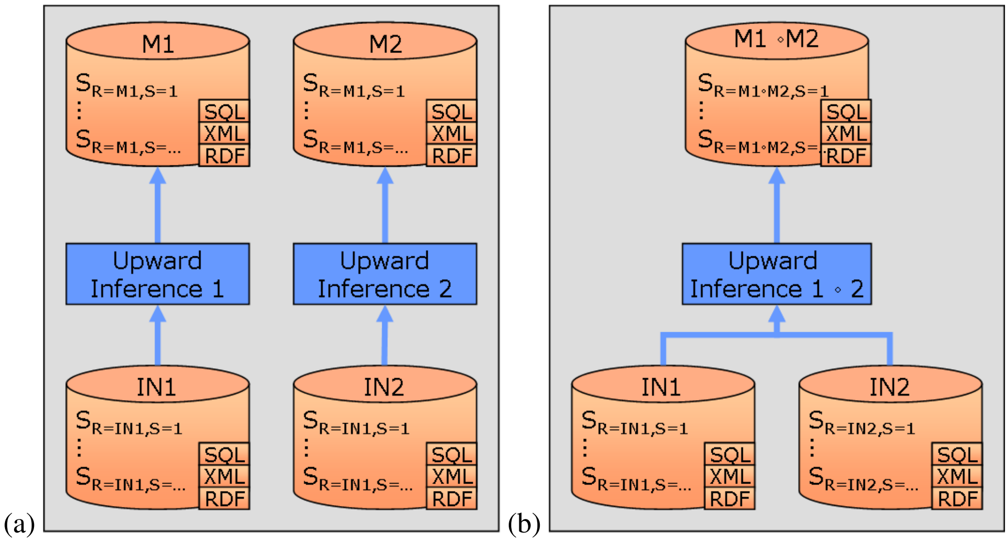


***Figure 7.4**: User model integration with an integrated upward inference step*

that reinforce the message. They do imply further evidence and are supposed to change the `confidence` attribute value. *Communication Doubles* can be detected if their globally unique identifiers are the same. This was the motivation to introduce the `unique` attribute within SITUATIONALSTATEMENTS, such that it is not lost during the communication process.

To summarize, because of the simple accretion concept within SITUATIONREPORTS, we can found *User Model Merging* on *Smart Situation Retrieval* and *Semantic Conflict Resolution*. Thus, the issue of *Reusability* of partial and distributed user models has been solved. A further discussion can be found in [Heckmann, 2005a].

7.1.3 Integrating SITUATIONALSTATEMENTS into the Ontology

According to [Holsapple and Joshi, 2002], a typical reason for constructing an ontology is to give a common language for sharing and reusing knowledge about phenomena in the world of interest. Ontological commitment is important. It is the agreement by multiple parties (persons and software systems) to adopt a particular ontology when communicating about the domain of interest. Where ontological commitment is lacking, it is difficult to converse clearly about the domain and benefit from knowledge of others. Thus we show in this subsection how to integrate the SITUATIONALSTATEMENTS (which are normally stored in databases, in XML files or in RDF files) into the general user model ontology GUMO and the UBISONTOLOGY. Some ideas go back to [Chepegin et al., 2004a] and [Chepegin et al., 2004b].

Figure 7.5 shows the conceptual relation between syntax and semantics in our approach. URIs that are called *semantic pointers*, point from the syntactical SITUATIONALSTATEMENTS to the ontologies. Default values for `expiry` and `privacy` are returned from the UBISWORLD ONTOLOGY, while further ontological effects from other ontologies are also used for syntactical purposes.

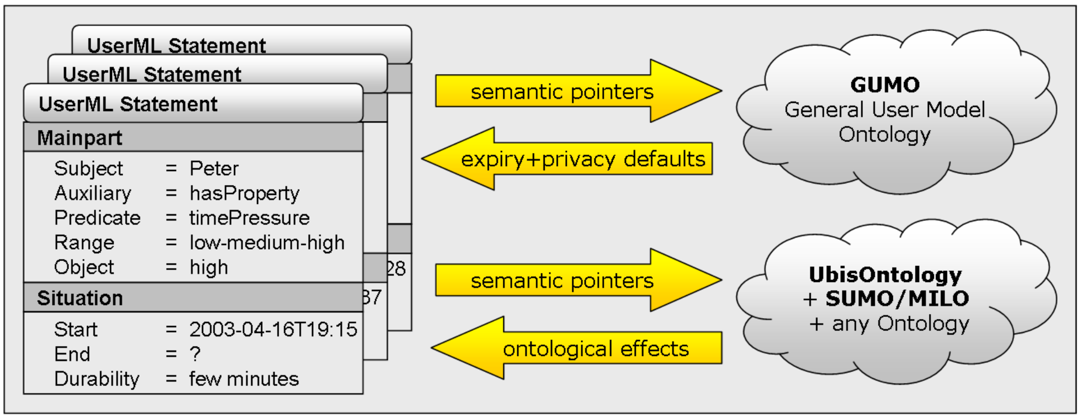


***Figure 7.5**: The Syntax-Semantics relation between UserML, GUMO and UbisOntology*

Figure 7.6 shows a condensed conceptual view to the vocabulary of OWL ontologies.

The statements within a SITUATIONREPORT are placed into the A-Box. Semantic web ontologies only allow binary relations to be represented directly. The SITUATIONALSTATEMENTS now enable one to represent n-ary relations directly. The *Ontology Legend* names the arcs in the diagram, while the *Category Legend* names the six categories or "colors" from UbisWorld that are used throughout the whole approach. Thus the individuals within the A-Box are colored. This rather overfull diagram is not discussed here in detail, it should only give an impression of the relation between statements and the rest of the ontology.

7.1.4 Integrating the Ontology into the Database

Mentionable is also our approach for the other way round: *How do we model the ontology into a relational database?* One could argue that this is not necessary, since no ontological reasoning is supported within relational databases. However, since we offer for each individual, each class and each property within our developed ontologies a virtual representative on the web, it is reasonable to integrate the ontology into the database. Basic information about each ontology element should be retrievable from the net. We even support online editing of the general user model ontology GUMO, see section 8.2.2, where multiple browsers can use the same application in parallel.

Figure 7.7 shows the central part of our database schema for representing parts of the ontology. This schema supports multi inheritance in the `element_parents` table. Furthermore it allows an arbitrary number of binary attribute-value pairs for each element in the `element_adds` table. The multi-inheritance topology relation is defined by the `element_topology` table. As result, we can navigate in the ONTOLOGYBROWSER from one ontology element to the next through the subsumption graph as well as through the topology graph.

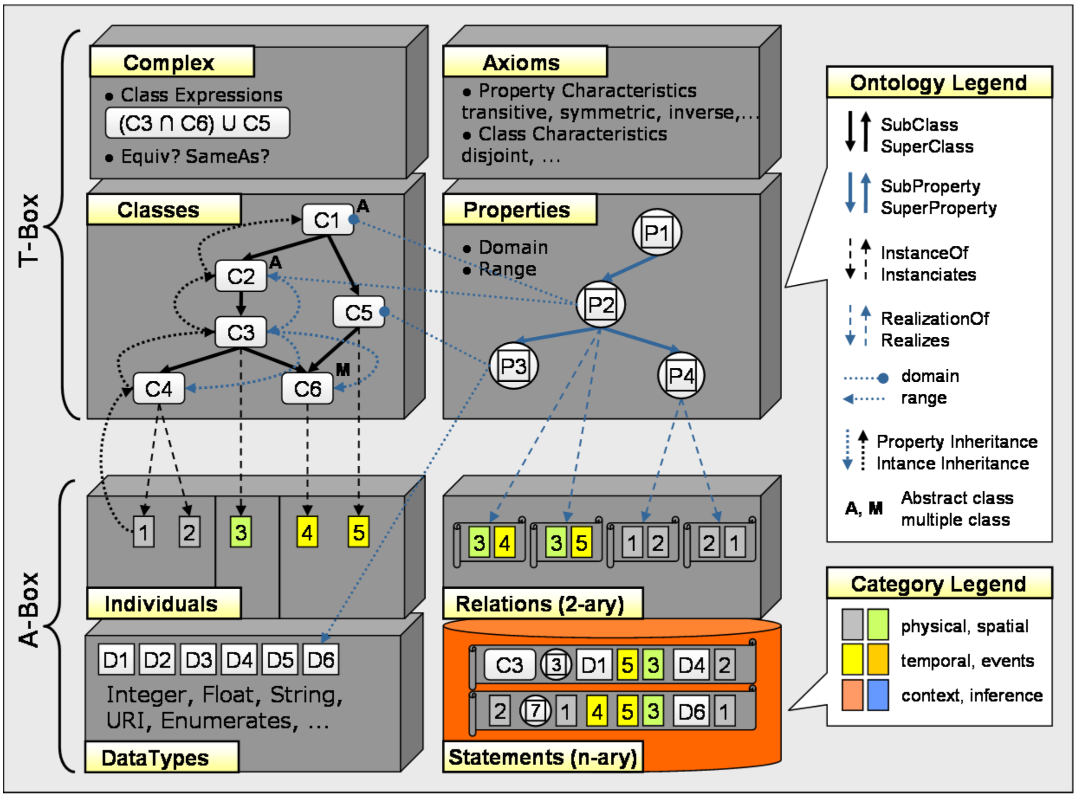


Figure 7.6: Integrating SITUATIONALSTATEMENTS into the OWL ontology theory

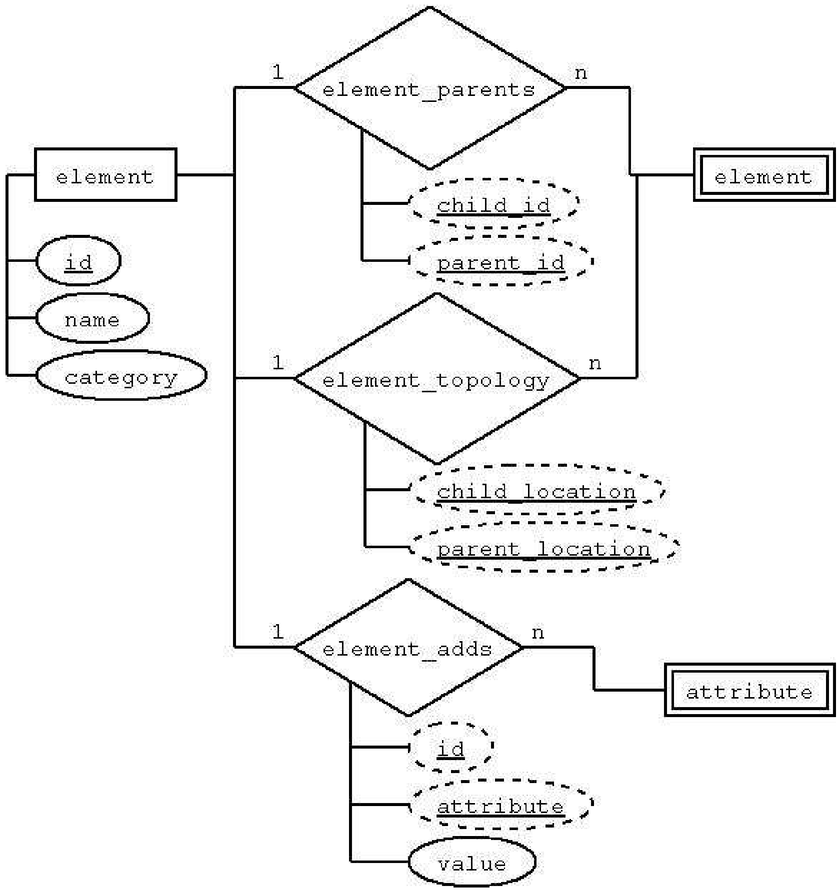


Figure 7.7: The central part of the database schema to represent the ontology

7.2 Decentralized User Modeling

A detailed analysis of decentralized user modeling can be found in [Fink, 2004] and [Heckmann et al., 2005b]. In this section we focus on *mobile user modeling* and the problem of *scalability*.

Each mobile and stationary device has its own repository of situational statements, see figure 7.8, either local or global, dependent on the network accessability since a mobile device can perfectly be integrated via wireless LAN or bluetooth into the intelligent environment, while a stationary device can be isolated.

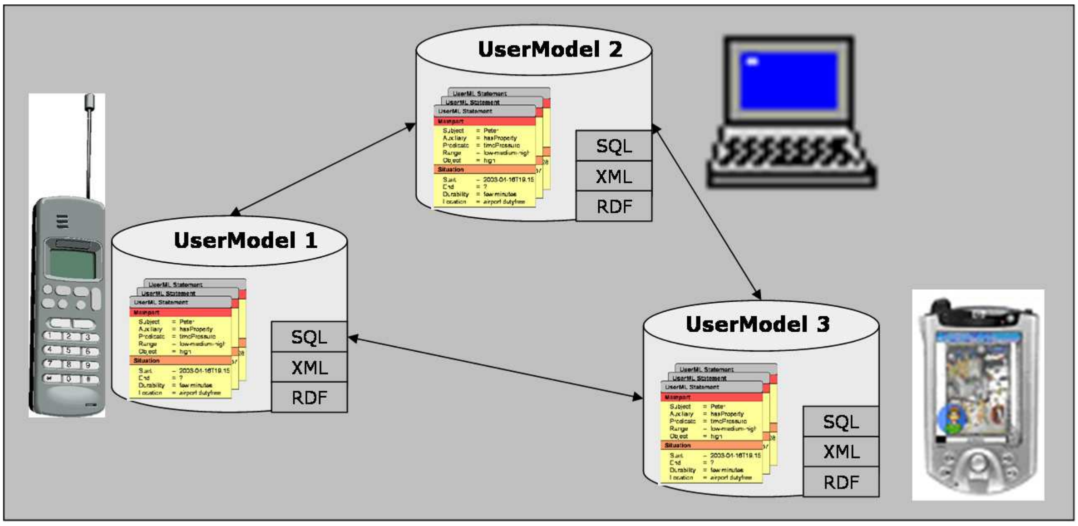


***Figure 7.8**: In decentralized user modeling each system can have its own user model*

7.2.1 User Modeling with Mobile Devices

A user model *service* manages information about users and contributes additional benefit compared to a user model *server*. The `u2m.org` user model service is an application-independent server with a distributed approach for accessing and storing user information, the possibility to exchange and understand data between different applications, as well as adding privacy and transparency to the statements about the user. The key feature is that the semantics for all concepts is mapped to the GUMO ontology. Applications can retrieve or add information to the server by simple HTTP requests, alternatively, by a `UserML` web service. A basic request looks like:

```
http://www.u2m.org/UbisWorld/UserModelService.php?
subject=Peter&auxiliary=hasProperty&predicate=Happiness
```

We now discuss possible roles of mobile devices in respect to user modeling, that has been presented in [Heckmann, 2002b]. Let us assume people carry a personalized user-modeling

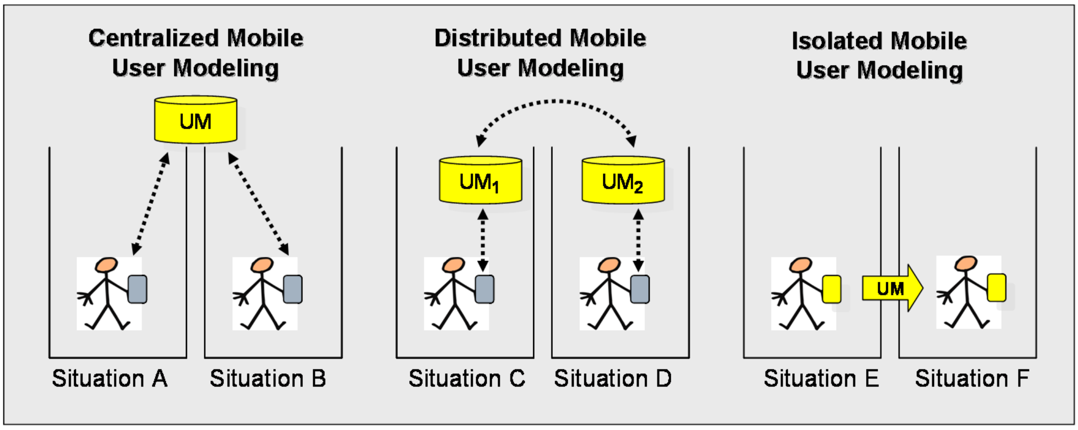


Figure 7.9: *Concept of centralized, distributed and isolated mobile user modeling*

system within a PDA and transmit their long-term properties and preferences anonymously to human-computer interaction systems. This could enable user-adaption from the beginning of using a new interaction system. After the interaction session, the system could transmit the partial user model of this session back to the personal agent within the PDA and also use the enriched data anonymously for collaborative filtering.

Mobile Devices and User Model Acquisition Advances in wireless networking facilitate large sensor networks, but the use of mobile devices themselves can already serve data and information about the user, i.e. analyze manual input like scrolling and pen-writing. Computational constraints on PDA-like devices restrict the possibilities of mobile devices to apply resource-intensive inference algorithms, thus special inference procedures are needed.

Mobile Devices as Collectors and Distributors In order to connect user-adaptive systems without network connectivity, a mobile device could serve as an intermediary, and it could also collect selected data. The distribution of data can take part between different systems and also between different time points.

Mobile Devices as User Model Clients The mobile user modeling agent can have an adaptive user interface. To adapt the user interface so that it fits better with the user's way of working, interface elements, like menus, icons, and the system's processing of signals from input devices such as mobile-keyboards can be adapted.

User Model Editing Tool If a computing system sets up a model about a user, the person has the right to inspect this model in a human readable format and edit the data. A private 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 according to the following section as well as user identification.

Mobile Security and Privacy A mobile device with a biometrical access system with a sensor that checks the finger-print of the user would enable an intuitive entry to secure user modeling, without the troublesome entry of passwords or PINs, while at the same time assuring the prevention of misuse of critical user data. Another argument for security is that the critical data like users demographic information for example date of birth or credit card numbers can be carried along and communicated to only selected systems.

7.2.2 Scalability of Distributed Situational Repositories

Figure 7.10 shows the concrete realized set of repositories, databases and computer servers that currently run to test the scalability of the distributed and ubiquitous approach. The distributed approach for the ontologies, situation models and low level sensor data is based on the following denotation model to identify each individual repository or each set of repositories efficiently:

```
<source>      ::= <repository> ( '|' <source> )*
<repository> ::= <table> '@' <database> '@' <server>
```

Our implemented arrangement of SITUATIONALSTATEMENTS with its database servers, databases and database tables already leads by `1:n:m` to almost 1600 repositories, which proves the scalability.

Another interesting approach to scalable and robust distribution is cooperation. A recent solution has been presented by *BitTorrent*[1], those users or systems that download the file or information tap into their upload capacity to give the file of information to others at the same time. And those that provide the most to others get the best treatment in return. Since this kind of cooperative distribution can grow almost without limit, because each new participant brings not only demand, but also supply, this approach seems to be highly relevant for upcoming large sets of SITUATIONALSTATEMENTS. However, the investigation of cooperative distribution of situational repositories for ubiquitous user modeling is encouraged for future work.

[1] BitTorrent homepage `http://www.bittorrent.com/`

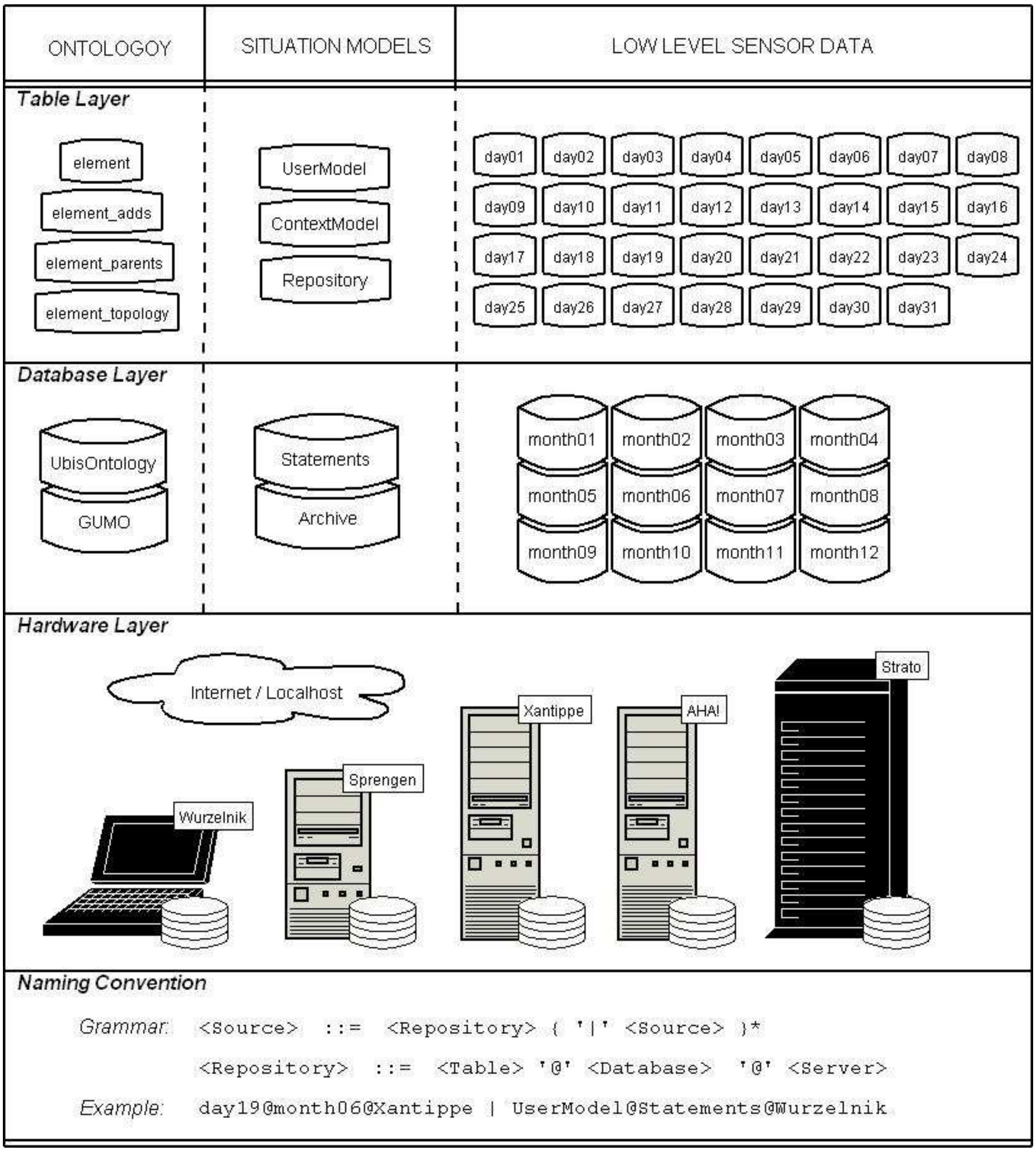


Figure 7.10: *The distributed approach for the ontologies, situation models and low level sensor data. This organization of* SITUATIONALSTATEMENTS *with its database servers, databases and database tables already leads by* `1:n:m` *to almost 1600 repositories within approximately 80 databases.*

Part III

System Architecture and Applications

8 USER MODEL SERVICE & USER INTERFACES

"In theory,
there is no difference between theory and practice"

8.1 USERMODELSERVICE

A *user model service* manages information about users, individuals or groups and contributes additional benefit compared to a pure *user model server* that only manages information. The `u2m.org` user model service is an application independent server with a distributed approach for accessing and storing user information, the possibility to exchange and understand data between different applications as well as adding privacy and transparency to the statements about the users. One key feature is that the semantics for all user model dimensions is mapped to the general user model ontology GUMO and thus the inter-operability between distributed user-adaptive systems is granted. The `u2m.org` user model service supports most of the newly introduced methods and data-types among which are the user model exchange languages `UserML` and `UserQL` in their semantic web representations XML, RDF and OWL. In this section we present the overall architecture of the user model service and show how to add new SITUATIONALSTATEMENTS to the server. The task of retrieving information from such services has already been introduced and analyzed in chapter 6.

8.1.1 Architecture of the User Model Service and its Environment

Figure 8.1 shows the main actors of the `u2m.org` USERMODELSERVICE and its environment in a static architectural plan. (The procedural dynamics within the user model service and its environment is presented in figure 8.2). The **Distributed Services** box is presented in the middle. It is literally surrounded by its conceptual environment. Even though the items are shown conceptually close to each other, they are spatially spread throughout the whole scenery. The box contains a set of internal modules that represent tasks and roles that are offered by the user model service.

- *User Model Server* or *Situation Server*, a web-server that manages the storage of the statements

- *User Model Adder* or *Situation Adder*, a parser that analyzes the incoming new statements and writes them to the distributed repositories.
- *Retrieval Filter*, a procedure that controls the retrieval of situational statements
- *Conflict Resolution*, a complex process that detects and resolves possible conflicts
- *Inference Engine*, a proactive engine that applies meta rules, writes new statements and triggers events (not completely implemented yet)
- *Interface Manager*, a control mechanism that integrates the user interfaces
- *Ontology Reasoning*, a reasoner that applies knowledge from the various ontologies

The **Applications** box on the left, sorts the applications that already cooperate with the USERMODELSERVICE according their application domain: *museum*, *navigation*, *shopping*, *biosensors*, *speech* and *e-learning*. The applications are discussed in chapter 9.

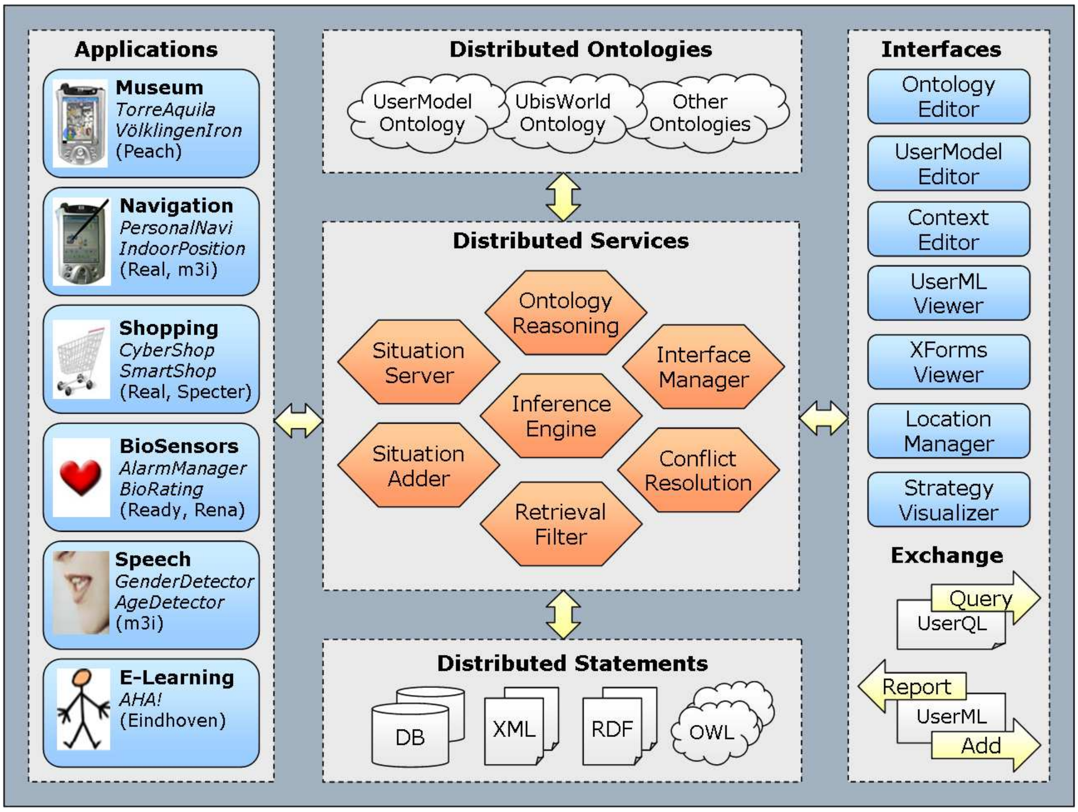


Figure 8.1: *Architecture of the* USERMODELSERVICE *and its environment*

The **Distributed Statements** box on the bottom points to the clear separation between data and software. The repositories are completely independent from the services which allows various services to operate independently on the same knowledge bases. This is only possible because the `privacy` and `administration` attributes are attached to each individual SITUATIONALSTATEMENT and not (as in most other systems) handled by the user modeling system.

The **Distributed Ontologies** box on the top points to the clear separation between the syntax and the semantics as discussed in chapter 5. These ontologies are used for the interpretation of the statements, for the detection of conflicts and for the definition of expiry defaults and privacy defaults.

The **Interfaces & Exchange** box on the right points to the clear separation between the user model service and the user interfaces and development tools which results in the advantage that each interface and tool can operate with different repositories, different ontologies and even different user model services. This is for example important for the spatially spread computational setting within ubiquitous computing. If the network connection is lost, the user interfaces can smoothly swap to device-local systems or integrate spatially restricted repositories. The developed user interfaces are described in the remaining part of this chapter. The communication between the boxes and items is indicated by the yellow bipolar arrows. `UserQL` is used to ask the queries, `UserML` is used to report the answers and to add new statements. The latter is described in the subsequent subsection 8.1.3, directly after the service architecture is looked at from its procedural view.

8.1.2 Procedural View to the User Model Service

Figure 8.2 shows the input and output information flows *Add*, *Request* and *Report* of the USERMODELSERVICE. They are denoted as yellow arrows with a short description of their format. The numbers in the orange ovals present the procedural order. Number (1) visualizes

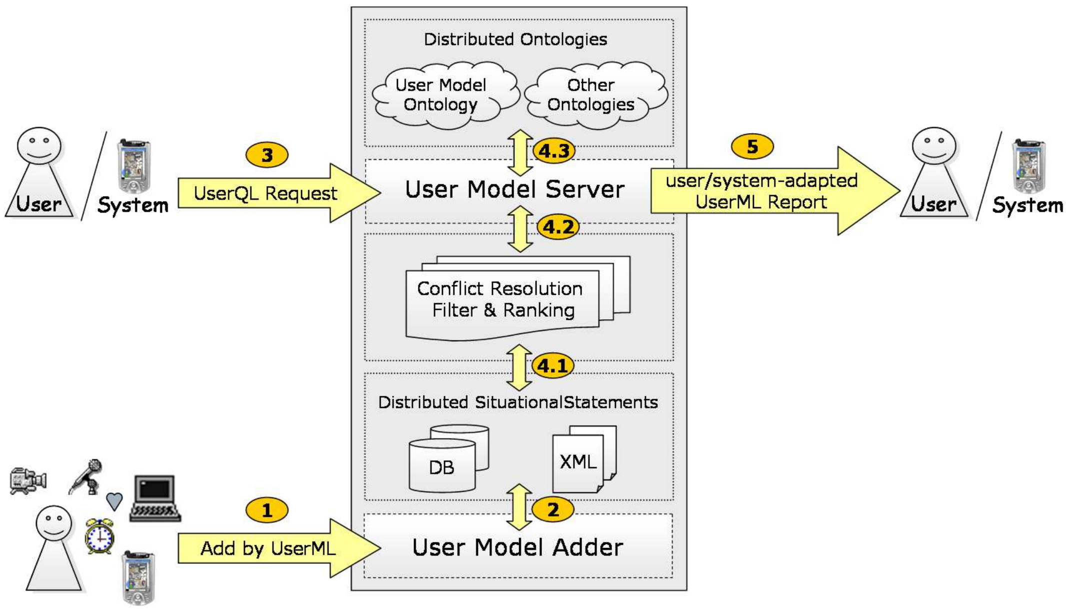


Figure 8.2: *Procedural view to the* USERMODELSERVICE

the sensors, users and systems that add statements via `UserML`. The statements are sent to the so-called *User Model Adder*, a parser that preprocesses the incoming data and distributes them to the different repositories, as indicated by number (2). If now a request is sent to the *User Model Server* via `UserQL` from a user or a system, see number (3), the according repositories are selected from which the statements are retrieved as shown at number (4.1). Then conflict resolution strategies are applied, see number (4.2), and the semantic interpretation as indicated by number (4.3). Finally, the user-adapted or system-adapted output is formatted and sent via HTTP in form of an `UserML` report back to the requesting user or system, see number (5).

8.1.3 Adding New Statements to the User Model Service

This subsection describes how to add new information to the USERMODELSERVICE. In analogy to situation retrieval, several methods to send data to the server are provided. The easiest way to enter new information to the USERMODELSERVICE is the "add by URI method". So-called *`AddURIs`* need to be assembled and executed in this approach. An *`AddURI`* is a Unified Resource Identifier that is able to contain a complete situational statement in the form of attribute-value pairs. These attribute-value pairs are sent to the USERMODELSERVICE with the HTTP get method, where the statement is added to the repository. The URI for the `u2m.org` USERMODELSERVICE is "`http://www.u2m.org/adder.php`" plus the attribute list which starts with a "?" and separates the attribute-value pairs with "&". Valid attributes are all attributes that are modeled within the concept of SITUATIONALSTATEMENTS.

Now, two example statements are presented. The first one is added by the *PeachMuseumsGuide* and the second one is added by the user himself. The *PeachMuseumsGuide* has the UbisId `200053` in UBISWORLD. Thus if it wants to inform the UserModelService about a new statement, it should append `creator=200053` to the *`AddURIs`* to identify itself. Otherwise, the statement will be marked with an "anonymous creator status", which results for example in a lower confidence value for the the conflict resolution strategies.

A) The situational statement to inform that *Michael is under high time pressure*, detected by the *PeachMuseumsGuide*, could be represented as shown in listing 8.3. In this example, the `object` and `range` are given in the `naming.id` form, where `640033` has the meaning *high* and `640030` has the meaning: *lowMediumHigh*.

```
http://www.u2m.org/adder.php?creator=200053&subject=Michael
&auxiliary=hasProperty&predicate=TimePressure&range=640030
&object=640033&confidence=0.8&location=TorreAquila
```

Figure 8.3: *This `AddURI` informs the server that "Michael is under high time pressure!"*

If no `start` time is mentioned, the actual time is implicitly taken as default starting time. If no owner is mentioned, then the actual `subject` is taken as the default `owner` of this piece of information. If no `end` time or `durability` is given, the latter is taken from the GUMO ontology.

B) The new information *Jörg is interested in football*, stated by Jörg himself, is shown in listing 8.4. In this example, the `object` and `range` are given in the `naming.label` form, where the `range` *yesNo* corresponding UbisId is `640010` and the `object` *no* corresponding UbisID would be `640012`. For each added statement, the user model adder returns a globally unique id like `<unique>S4243d9a06b16c</unique>` to the information creator, which manages this `unique` attribute to mark the statement as being `replaced` afterwards, when it updates the older statement with a new one.

```
http://www.u2m.org/adder.php?creator=210006&subject=Joerg
&auxiliary=hasInterest&predicate=football&range=yesNo
&object=yes&access=public&purpose=commercial
```

Figure 8.4: *This `AddURI` represents the information that "Jörg is interested in football."*

8.1.4 Remark on Modeling with Multi-Part Attributes

With the "interest-in-football-example" one could argue that all `mainpart` attributes are filled, but how should we differentiate between "*being interested in `playing` football*" versus "*being interested in `watching` football*"? One idea could be the extension of the ontology by adding the two classes *watching football* and *playing football* as subclass of the user model interest category *football*. However if we do this for all sports and games, the ontology will grow very fast. A better solution can be realized with the list structure for attributes as introduced in section 4.5.3 about UbisExpressions. Listing 8.5 shows multi-part `auxiliary` and `predicate` attributes.

```
(1.1) auxiliary=hasInterest|playing&predicate=football&object=no
(1.1) auxiliary=hasInterest|watching&predicate=football&object=yes
(2.1) auxiliary=hasInterest|watching&predicate=firstLeague|football
(2.1) auxiliary=hasInterest|watching&predicate=localTeam|football
```

Figure 8.5: *Using multi-part attributes to refine the statements*

Line (1.1) and (1.2) represent that *Jörg is intersted in watching football but not in playing football*. Line (2.1) and (2.2) define the even more specific properties "*being interested in watching `first-league` football*" and "*being interested in watching `localTeam` football*". Thus the `mainpart` of SITUATIONALSTATEMENTS with its five attributes `subject, auxiliary, predicate, range` and `object` can be much more flexible than one expects in a first impression. Even if the |-separated list were not expressive enough for future situational descriptions, it could be sufficient to extend the definition of UbisExpressions only (for instance with further operators like + or with semantic frames as in *FrameNet*, see section 3.3.4) and leave the syntax of `UserML` unattached.

8.2 User Interfaces for UbisWorld with UbisOntology

UbisWorld can be used to represent parts of the real world like an office, a shop, a museum, an airport or a city, as discussed in section 5.3. It represents persons, objects, locations as well as times, events and their properties and features. UbisWorld could be understood as a virtual colored world where each color represents a different category in the ontology. The user interfaces and tools are implemented in *PHP4*, *MySQL*, *Java*, *JavaScript*, *HTML* and *XML* with the modules *dtree*, *tabpane*, *phpMyAdmin*, *MagumaStudio* and *XForms*. This chapter briefly presents the main user interfaces that are also online available at `http://www.ubisworld.org`.

8.2.1 The UbisWorld Manager

The UbisWorldManager is the main user interface that leads to all other internal and external tools. It is divided into two pages, a left one and a right one, surrounded by an orange frame with logo, heading, imprint information and iconic links to the supporting institutions and projects. Figure 8.6 shows a screen shot of this user interface. Each of the two pages are organized in tab panes that vary according to the actual selections and the rights of access. The tab pane of the left hand side contains the tabs `[Introduction]` with a short description about UbisWorld and a login area, `[SemanticWeb]` with hyperlinks to the UbisOntologyEditor and the UbisOntology documents, `[Location]` with the topology tree and other spatial relations, `[Simulation]` with links to monitoring tools and forms to simulate changes in UbisWorld and finally the `[UbisEditor]` with forms to change the UbisOntology and the UbisLocation as described below.

The tab pane of the right hand side describes the randomly selected element *Fluid Desk* as instance of the *Fluidum Toolset*. The tab `[ID-406052]` describes the ontological information about *Fluid Desk*, the tab `[Context]` contains the actual context model of this system, while the tab `[ContextLog]` contains all logged context information.

UbisOntology is defined as the combination of the general user model ontology Gumo and the UbisWorld ontology that perfectly fit together and complement one another. These ontologies can be inspected and edited online with the developed tools as described below.

8.2.2 The Ontology Tree Browser

The UbisOntologyBrowser[1] is a tree browser that consists of one or several foldable trees as shown in figure 8.7. The trees should be considered as being a special representation of a semantic net, with classes and typed instances. Multiple inheritance is realized by node copying. The classes are represented by (C)-symbols while the instances are represented by colored rectangles. The color refers to the type as defined in figure B.1 on page 230. The parent-child connection between two classes in the tree implies implicitly the *concept subsumption* relation and the connection between a class and an instance implies implicitly

[1] UbisOntologyBrowser: `http://www.u2m.org/UbisWorld/UbisBrowser.php`

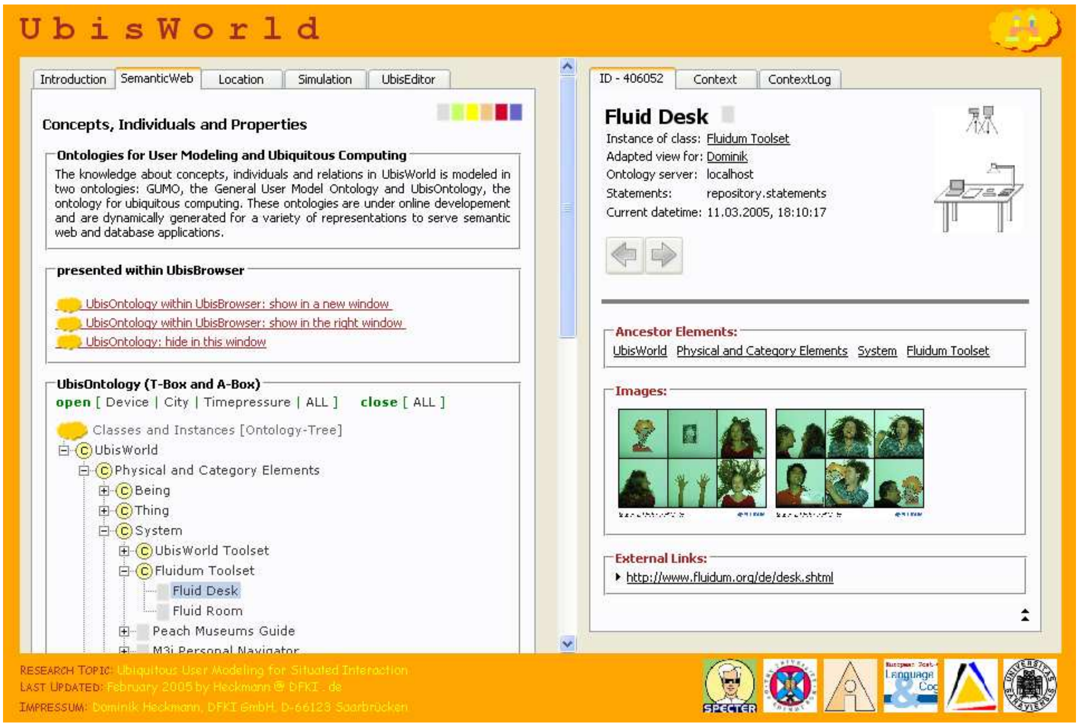


***Figure 8.6**: UbisWorld Manager with the selected tree browser on the left hand side and selected element FluidDesk on the right hand side*

the *is-a* relation. Further roles or relations are defined in adjacent tables and trees if they carry hierarchical structures like the *is-nested-in* relation as shown in figure 5.23 on page 107. The ontologies are also available as documents in the semantic web languages RDFS, InstanceOIL, DAML+OIL and OWL. These documents can be inspected with ordinary text editors or web browsers.

8.2.3 The Ontology Editor

To support the distributed construction and refinement of the general user model ontology GUMO and the UBISWORLD ontology, we developed a specialized online editor, that helps with introducing new concepts, adding their definitions and transforming the information into the required semantic web ontology language.

The UBISONTOLOGYEDITOR offers three different forms to enter new information or to change existing information. The first one allows to enter new elements, either a new class or a new instance. The colored type of the instance and the new element ID are automatically deduced from the selected parent class. The parent class can be selected via its name or its ID. The new concept name may contain empty spaces for better readability, which forms a difference to existing ontology editors. However for compatibility with other ontology editors, these empty spaces can be suppressed. Once the new element is entered, it is available in the drop-down boxes of all other user interfaces, including the ontology editor itself. Thus

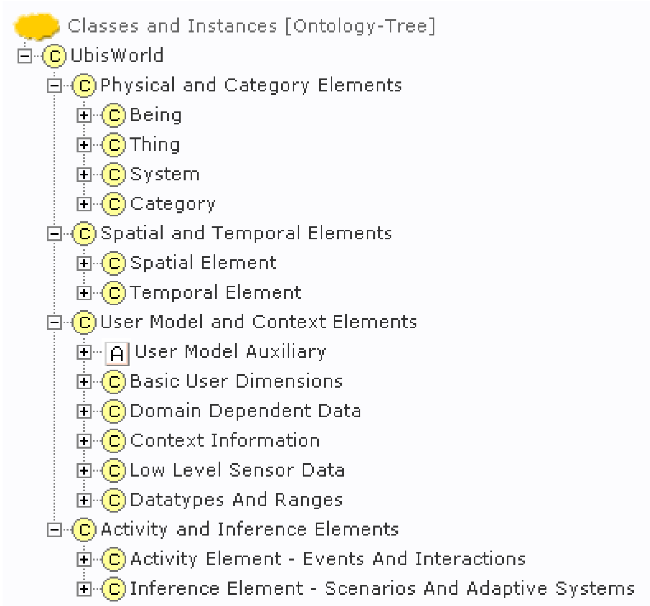


Figure 8.7: *The first three levels in the UbisOntology seen with the UbisOntologyBrowser*

further children in the subsumption hierarchy can be added. Figure 8.8 shows a screen shot of the *OntologyEditor* tab pane within the UBISONTOLOGY web site. It is only visible to certain logged-in users that have the rights to use this tool. Furthermore this editor is personalized in the way that it keeps track of who has entered or changed which elements.

The second form allows for changing the parent class of a concept. Thus the later rearrangement of misplaced concepts in the ontology is easy to handle. Since no online deletion of concepts is supported, a common way to do so is to mark the concepts as deleted by moving them to the so-called *Papierkorb* or ontology bin. Final deletion and changes of the concept's name or its relations can be done with a user interface called *phpMyAdmin*[2] that models the whole ontology in a newly developed relational representation.

The third form allows to enter new attributes like *icon*, *description*, *wordnet*, *domain*, *range*, *durability*, *privacy*, *link* and so on. The values of these attributes can be URIs with an optional prefix defining the namespace, ordinary strings or UbisIdentifiers. However, SITUATIONALSTATEMENTS are not entered here. They are handled within the user model editor, see subsection 8.4.1.

Apart from using UBISONTOLOGYEDITOR we modeled some DAML+OIL ontologies with the OilEd ontology editor[3], especially the first submitted version of the user model ontology, and some OWL ontologies with the Protégé ontology editor[4], especially the extensions to the existing SUMO and MILO ontologies.

[2]phpMyAdmin hompage: `http://www.phpmyadmin.net/`
[3]OilEd homepage: `http://oiled.man.ac.uk/`
[4]Protégé homepage: `http://protege.stanford.edu/`

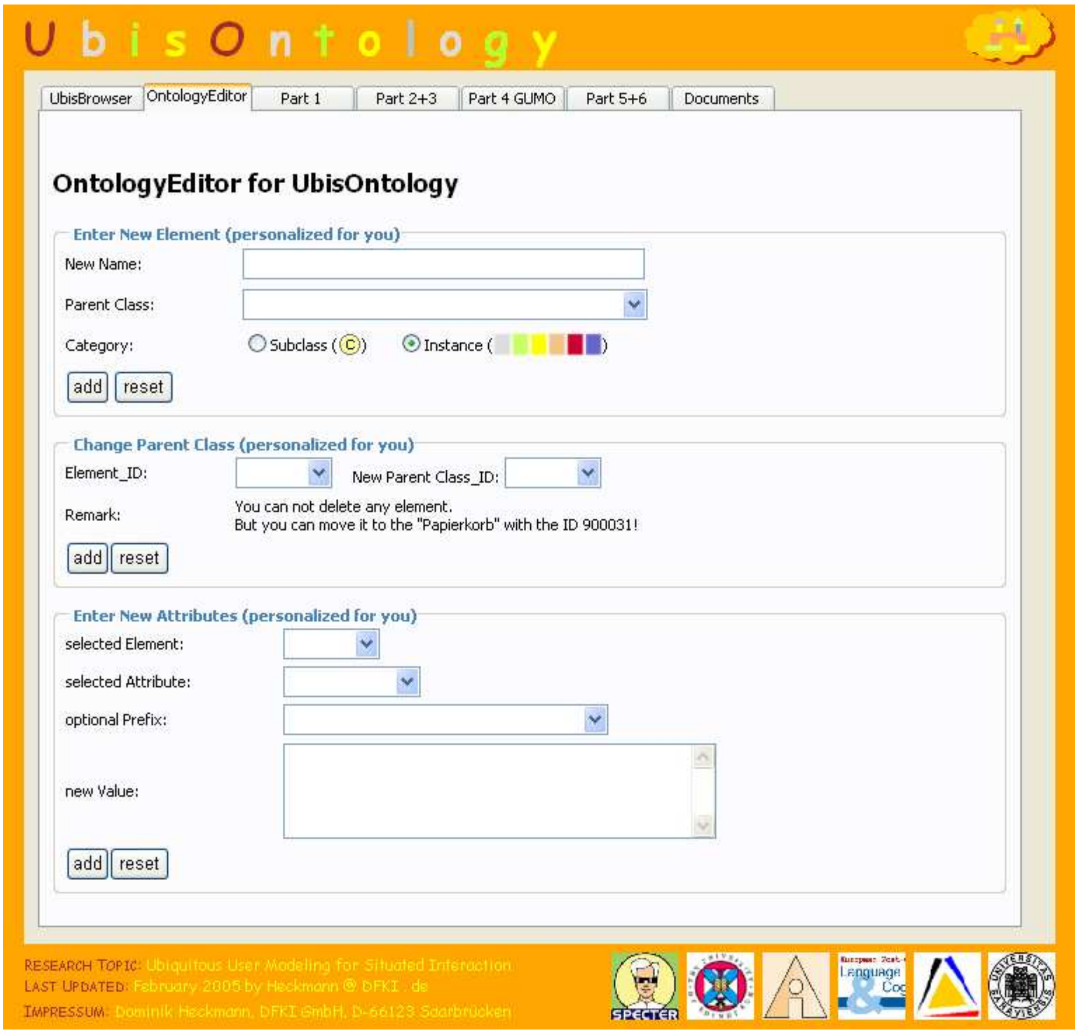


***Figure 8.8**: The online* UBISONTOLOGYEDITOR

8.3 User Interfaces for UBISLOCATION

UBISLOCATION[5] is a set of user interfaces and tools that cover most of the spatial aspects in UbisWorld. Several monitoring and editor interfaces have been developed, either for large screens or mobile devices. Furthermore, a map modeling tool for geometric location models has been developed that is based on UBISLOCATION.

8.3.1 The Location Monitor

The UBISLOCATIONMONITOR has already been introduced in section 5.3.2. Figure 5.23 shows a partially opened interactive tree that represents the "is-nested-in" or inclusion rela-

[5] UBISLOCATION homepage: `http://u2m.org/UbisWorld/UbisLocation.php`

tion. A screen shot of the mobile introspection tool for indoor walking will be presented in figure 9.4 on page 182. Another interesting application for location monitoring is for example the pet location tracking as shown in figure 8.9, all physical objects of a certain class with their latest known locations.

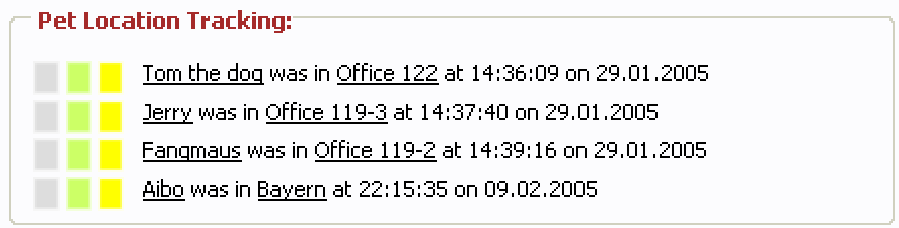


Figure 8.9: *Pet location tracking*

If you click on the underlined location you will be led to a location monitor where all physical elements of this location are listed. Even though this tracking task looks similar to the previous one, it is much more complex, since spatial inferences have to be applied. Figure 8.10 depicts all (modeled) physical elements in building 36 with all its nested sublocations.

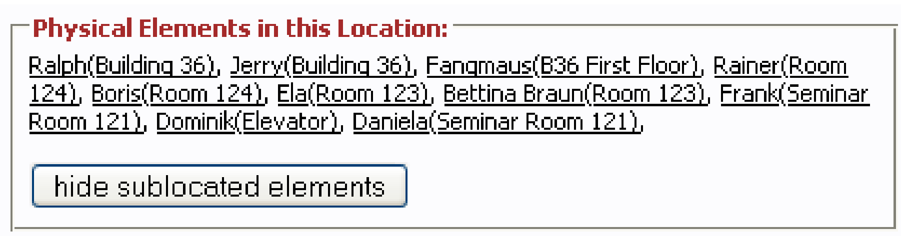


Figure 8.10: *Show all elements and sublocated elements at a certain location*

8.3.2 The Location Editor

This section describes how the static and dynamic spatial aspects are modeled in UbisWorld. The UBISLOCATIONEDITOR offers three different forms to enter information. The first one allows to enter rather static spatial inclusion relations between already existing locations. Every symbolic location can be spatially included in several other locations. It is an n:m binary relation that can be compared to the SUMO relation “is-partly-located”, see section 3.3.1 for a discussion of SUMO. New locations can be defined in the UBISONTOLOGY editor as described above.

The second input form allows to define spatial connections between two existing locations. These connections can either be bidirectional or unidirectional. This originates in the fact that there are one-way streets and even in pedestrian navigation there are doors and walk-throughs that can only be used in one direction. In contrast to the aforementioned inclusion relation, each connection is realized as a new element in order to attach further attributes

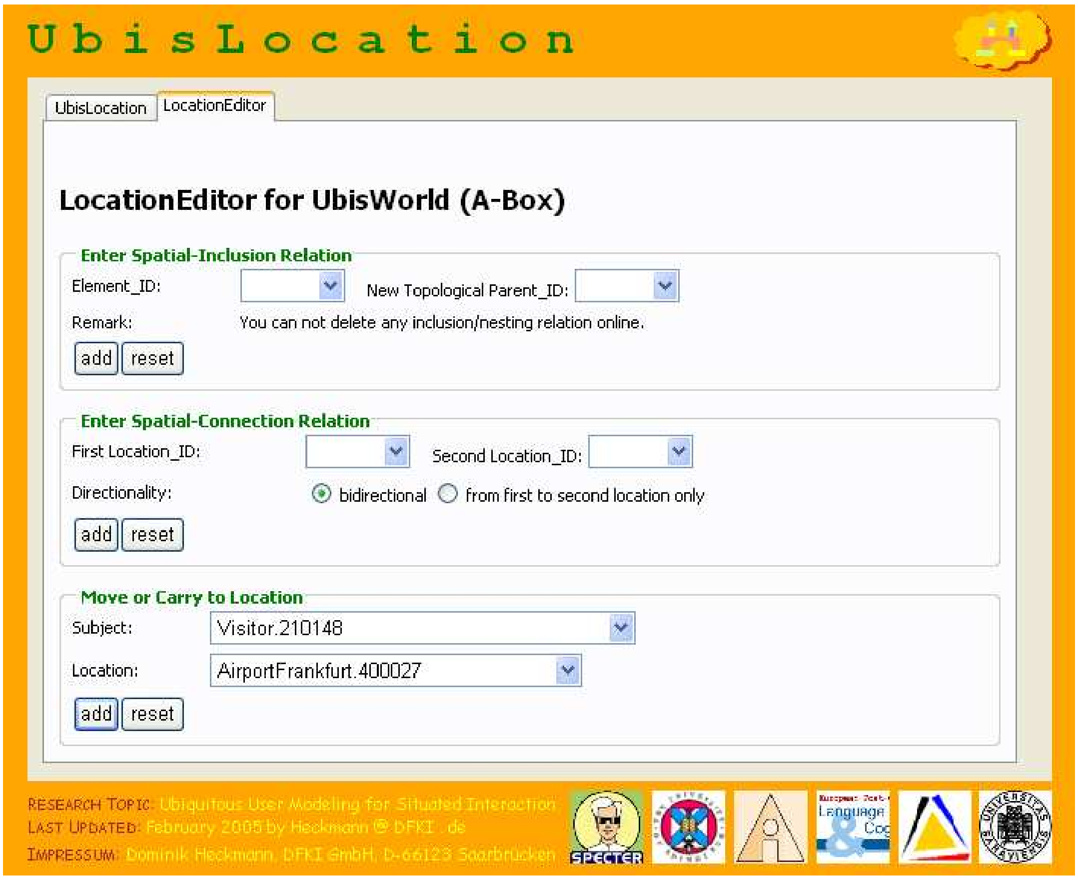


***Figure 8.11**: UBISLOCATIONEDITOR with forms to enter spatial relations and statements*

like a necessary key-card, the distance, the surface, the estimated walking time, included stairs and so on. These attributes are important for the next-generation pedestrian navigation planning, as discussed in section 9.4.

The third input form allows to report a moving or carrying event in terms of a SITUATIONALSTATEMENT to the U2M UserModelService. Thus, it does not change the ontology directly, but the separated repository of SITUATIONALSTATEMENTS. Other interesting attributes that can be attached to a location or a connection are the so-called *Digital Graffiti* as shown in figure 5.39 on page 118 or geometric maps as described in the following section.

8.3.3 The YAMAMOTO Map Modeling Tool

YAMAMOTO[6] is designed to create a hybrid symbolic and geometric location model for pedestrian navigation systems, with strong emphasis on positioning and route finding, see [Stahl and Heckmann, 2004b] for a detailed introduction. It addresses the scalability and

[6]YAMAMOTO = Yet another Map Modeling Tool.

operational overhead of such a model by the use of semantic web technology. In order to associate the spatial vocabulary of UbisWorld - that already provides a symbolic spatial model - with real-world coordinates such as those supplied by a GPS receiver, the YAMAMOTO tool for the creation of a hierarchical geometrical spatial model has been developed at the chair of Professor Wahlster. This model represents real-world places at different granularity levels as resources in the world wide web. The geometrical model is joined with UbisWorld by uniform resource identifiers as location identifiers.

Our physical environment is evolving constantly. We divide the world into a hierarchical tree structure, where each node refines one object of its parent node and represents one or more levels of granularity. Each refinement-node may specify its geometry using a separate coordinate system. The nodes contain both, geometric information and references to the symbolic spatial model of UbisWorld. They are encoded using XML and are made accessible on the world wide web. We use the semantic web language RDF to integrate the nodes to form a coherent model. This approach supports the growth of multiple models on different levels of granularity and their refinement as well as their union. Everyone is invited to use the editor and to add new nodes representing their own environment, and to integrate it into a larger scale model.

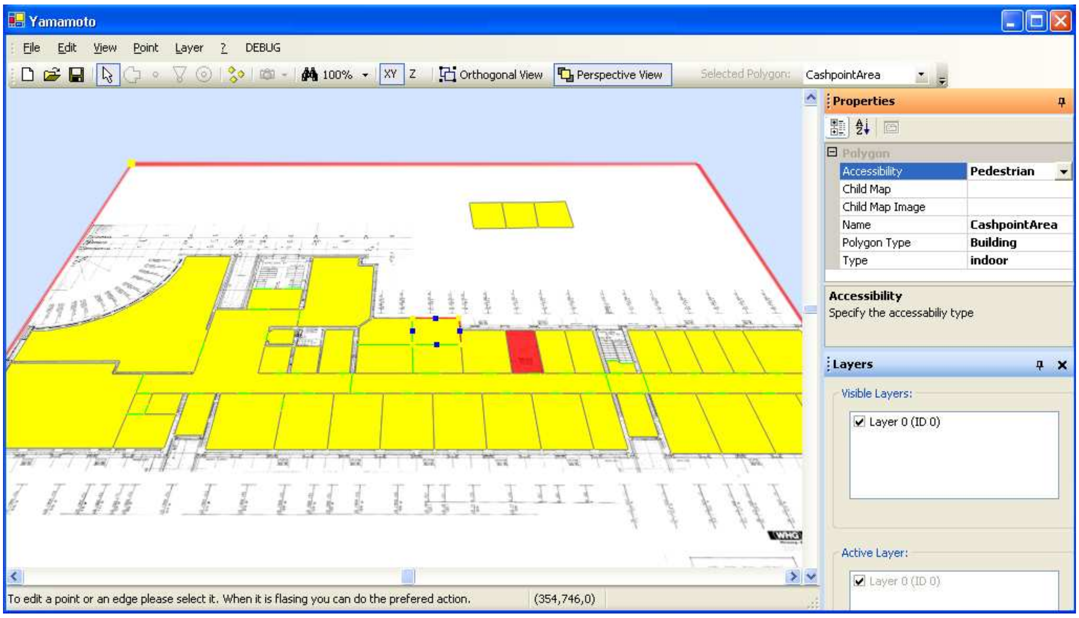


***Figure 8.12**: Screen shot of the* YAMAMOTO *map modeling tool*

The basic idea of the modeling process is to use graphical representations of the environment like architectural plans or aerial photographs as a source for the modeling. The geometric location model is drawn over the background of a bitmap image. A mesh of polygons is used to represent the logical and physical entities of the environment, like `streets`, `places`, `buildings` and `rooms`. An example screen shot is shown in figure 8.12. Since flexibility is especially in the indoor environment more relevant than precision, due to the lack of precise positioning technologies, existing AutoCAD-drawings may be imported as bitmap images to start the re-modeling with the YAMAMOTO editor. The polygons are used to represent physical boundaries in the environment by their edges: *outdoors* those are mostly discontinuities in the surface and *indoors* walls or doors. Polygons may also be used to define the boundaries of logical entities, like `buildings` or `floors`, which may be refined in separate nodes. The refinement of a polygon is specified by its URI. Modeling in the third dimension is supported by multiple layers, for example to represent the levels of a building.

For the positioning of the mobile device, we have to model some navigational fixpoints within the environment. For best results, heterogeneous sensor information should be used. In our current positioning infrastructure, a combination of RFID and infrared beacons is used. The editor provides various basic geometries to model the beam angle of the sender, such as point, disc and section. Whereas the RFID beacons radiate their position-identifier signal almost equally in all directions (disc model), the LEDs of the infrared beacons emit an directed id-signal with a range of 5 meters and a cone angle of 30 degrees (section model). Besides their position, the name and position-identifier of the beacon can be specified. In our scenario, the mobile device sends its IR- and RF-sensor readings to the positioning service, which uses the XML encoded geometrical location model to compute the device's position by triangulation between the beacon positions. It returns numerical coordinates for map visualization and a symbolic location identifier, which allows to query UbisWorld for the situational context of the surrounding area.

Besides the geometrical model, the editor allows the symbolical annotation of the polygons and their edges. URIs are used to refer to location instances in UbisWorld, where spatial relations like a lattice may be represented, which can not be expressed by the tree-structure of the geometrical refinements. In order to support route finding for pedestrian navigation, the model has to represent the connections between entities and their pass-ability. In comparison to cars, a pedestrian is not bound to a path network, but able to shortcut and cross large places. Therefore the system has to consider all routes across areas. The editor allows for naming and annotation of the edges which are shared by adjacent polygons. For example, a small path may be accessible for pedestrians, but not for cars, and some floors may require a key card outside working hours. These constraints are modeled as SITUATIONALSTATEMENTS. By naming edges uniquely, multiple representations of the same edge in other layers or even different nodes may be easily identified. This allows for route planning beyond the boundaries of the current granularity level. Using URI names instead of coordinates offers the freedom to abstract details from subsumed polygons on a higher level of granularity.

8.4 User Interfaces for User Model Inspection

For ubiquitous user modeling new tools are needed for inspecting and editing decentralized user models that put an extended focus on privacy. They could be compared with the so-called *scrutiny interfaces* in [Kay, 1995] and [Kay et al., 2002].

8.4.1 The User Model Editor

The USERMODELEDITOR offers an integrated web interface for inspecting and changing all user model information. It provides a site for each individual or element. See figure 8.13 for Margeritta's page. It is divided into an upper part and a lower part separated by a horizontal

***Figure 8.13**: USERMODELEDITOR with Margeritta's partial UserModel in Mainpart mode*

line. Pointer (1) shows the ID, a unique identifier for Margeritta within UBISWORLD. Pointer (2) shows the name of the element and its ontological parent class. Pointer (3) shows the current time of inspection on the right hand side and the commands *one hour ago* and *one day ago* for changing into the past, on the left hand side. With the commands in the row below, pointer (4), the view mode of the USERMODELEDITOR can be changed from *Mainpart* into *Extended*, *Editor*, *XML* and *RDF*. The *Mainpart* mode only shows the mainpart attributes of each statement, while the *Extended* mode also shows the situation, explanation and privacy attributes. Pointer (5) covers the login methods and shows the name of the current user, to whom the system is adapted. Pointer (6) displays the inheriting ancestor classes from the ontology. Pointer (7) shows the stored demographical information of Margeritta and pointer (8) shows the stored personality information.

Figure 8.14 shows the USERMODELEDITOR in *Editor* mode. Pointer (9) shows the form to enter a new statement body with subject-auxiliary-predicate, while pointer (10) enables one to change the default values for range and durability. Otherwise the default values are taken from the general user model ontology GUMO. Pointer (11) shows the subject-auxiliary-

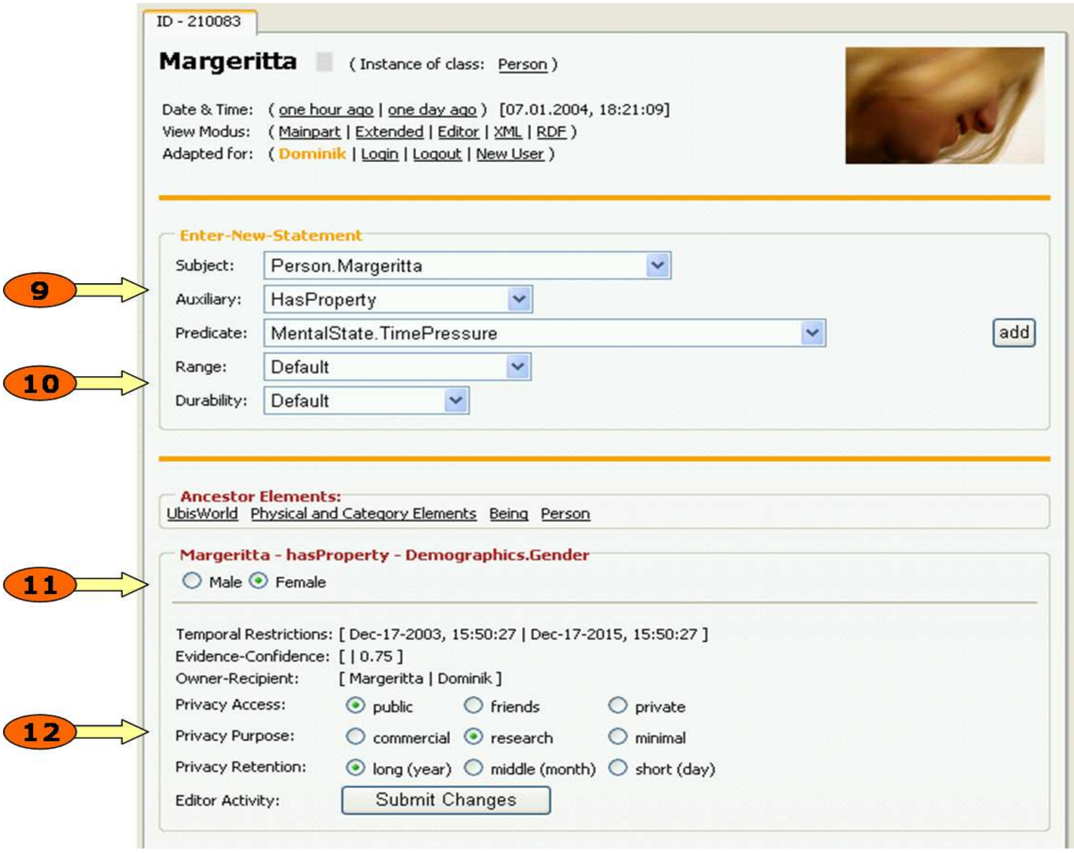


***Figure 8.14**: USERMODELEDITOR in Editor mode for Margeritta*

predicate part of the statement as a headline, the range-object separated by a line and the other attributes in the box below. Interesting is the possibility to change the privacy settings for each statement individually, see pointer (12) and the next section about privacy aspects.

8.4.2 Privacy Aspects and the Integrated Privacy Editor

In user modeling, context-awareness and resource-adaptive computing, information about the object of interest is gathered and further processed into higher level knowledge in order to receive hypotheses about possibilities for adaption of the system's behavior. Representing, storing and communicating information about the user like her `age`, or her current `time pressure`, `blood pressure`, `skin conductivity` or information about her `interests`, `goals` and `plans` in single isolated systems need privacy treatment. In the near future, users will live together with intelligent spaces and devices that communicate with each other about them. The problem that arises is that human-related adaptation and data storage in ubiquitous user modeling needs an extended treatment for privacy. One reason lies in law-restrictions as described in [Kobsa, 2002] and [Hinde, 2003] but the second - more persuasive - argument is the possible non-acceptance by the user if privacy is not handled with care, see [Kobsa, 2001b]. In mobile and ubiquitous computing the treatment of privacy issues seems to be especially difficult, since not every system will be able to interact with the user directly. Here, we describe the integrated PRIVACYEDITOR as part of the USERMODELEDITOR. It was first introduced in [Heckmann, 2003a]. In a discussion Professor Kobsa stated that there are four main arguments, that influence the users' decisions about allowing personalization with their personal data:

1. The first one is: *What will be done with their personal data?* This question focuses on the `purpose` of usage.

2. The second one is: *Who is going to use their personal data?* This question focuses on the `access` and the `recipient` or `requestor`.

3. The third one is: *Which kind of personal data is used?* Thus a differentiation between different classes of user model data is implied. (This influenced the design of GUMO).

4. The fourth one is: *In which mood or situation is the user currently?* This last point suggests a "user-adaptive user-adaptivity".

We tried to contribute to the problem of *how to integrate these privacy aspects into mobile and ubiquitous computing*. However, we did not investigate the problem of *security* in mobile or distributed systems, like encryption techniques or how to defeat attacks. We believe that this problem is orthogonal to privacy. In our approach we start with a trust based privacy concept and expect later security extensions to be added. The four main arguments from Kobsa as stated above could be extended to:

5. The fifth one: *How long will the personal data be stored?* This question rises the argument of `retention`, see [W3C-P3P, 2003] for a more detailed description.

6. The sixth one: *Can the user inspect or delete the personal data and decide to turn-off the whole user adaptivity?* This questions raises the demand for `control`.

To summarize, an important privacy aspect is that the user should be able to *control* the system's private information handling. One point will be the *inspection* of the stored data, another point will be the possibility to *change* it. Another point will be the possibility to turn the whole user-adaptivity off and on for individual systems or locations. The attributes `purpose`, `access`, `recipient`, `owner`, `retention`, `control` and `inspection` should play an important role in the treatment of privacy.

`UserML` already allows for the integration of privacy information together with the mainpart information on the most basic level of knowledge representation, see section 4.2. The onion model in figure 4.6 on page 59 shows that the privacy box is wrapped around the mainpart box, which means that the privacy settings have to be fulfilled in order to receive the mainpart information. Figure 8.15 presents two forms of the integrated PRIVACYEDITOR as part of the USERMODELEDITOR.

The intended meanings of the selected privacy attribute-value pairs are described in table 8.1. These privacy settings could be attached to individual statements within the user model, to selected user model dimensions like `birth place`, or to classes of user model dimensions like all `personality traits` within GUMO.

Figure 8.15: *Integrated* PRIVACYEDITOR *forms with (a) access:public, purpose:research, retention:short and (b) access:friends, purpose:commercial , retention:middle*

Attribute	**Value**	**Description**
`access`	`public`	everybody can be the recipient of this information
`access`	`friend`	only selected friends (persons, systems or locations) can be the recipient of this information
`access`	`private`	only the owner and the creator can be the recipient of this information
`purpose`	`commercial`	this information can also be used for commercial purposes like product recommendation
`purpose`	`research`	this information should not be used for commercial purposes, but only for research issues
`purpose`	`minimal`	this information can only be used for minimal purposes
`retention`	`long`	this statement must be deleted within a year
`retention`	`middle`	this statement must be deleted within a month
`retention`	`short`	this statement must be deleted within a day

Table 8.1: *The intended meaning of the nine privacy attribute-value pairs.*

The USERMODELEDITOR offers a privacy-adaptive view to the user models. Figure 8.16 shows for example the web page of Jörg, adapted to the fact that Jörg himself is logged in. He can inspect his current location and is able to change it on this page. He can also inspect his individual privacy settings and change them. Thus this page serves as an editing tool for user models of different user-adaptive systems. Especially in ubiquitous computing, not every user-adaptive system will have a user interface by its own. In such a case the USERMODELEDITOR can be used with its standard user interface.

Figure 8.18(a) shows the “Virtual Jörg”, adapted to the fact that not he himself, but a friend is logged in. This friend can see the current goals and the physical location of Jörg, but not the blood pressure, since this information is private. As a friend, he can only inspect this information, but not change it.

The version of the same page for a public viewer or recipient is even more restricted. Figure 8.18(b) shows that the values of the “current goals” of Jörg are hidden, but still the predicate “current goals” is stated, which expresses, that there is information about Jörg’s current goals, but they are not available for everybody.

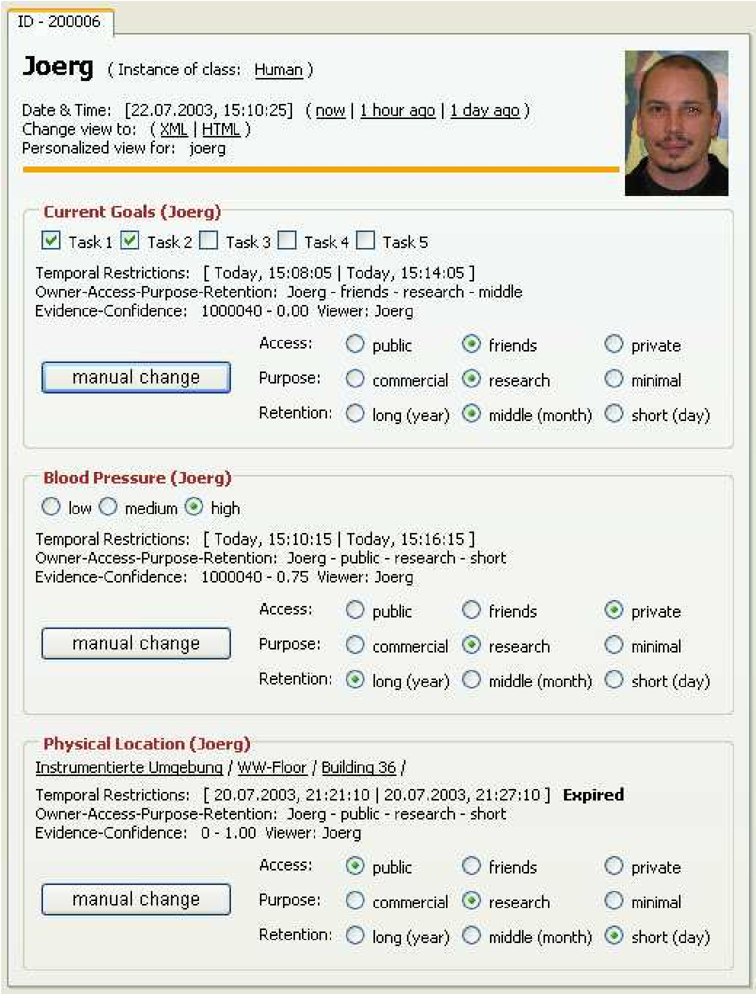


***Figure 8.16**: Jörg's personalized page for Jörg himself (with privacy editor)*

To summarize, in order to gain the user's acceptance of user-adaptivity in ubiquitous computing, the arguments of inspect-ability, controllability, access, purpose and retention are integrated into the basic data structure of the user model markup language `UserML`, the general user model ontology GUMO and the user interfaces like the USERMODELEDITOR. The integrated PRIVACYEDITOR allows the owner of the information to define the privacy settings individually.

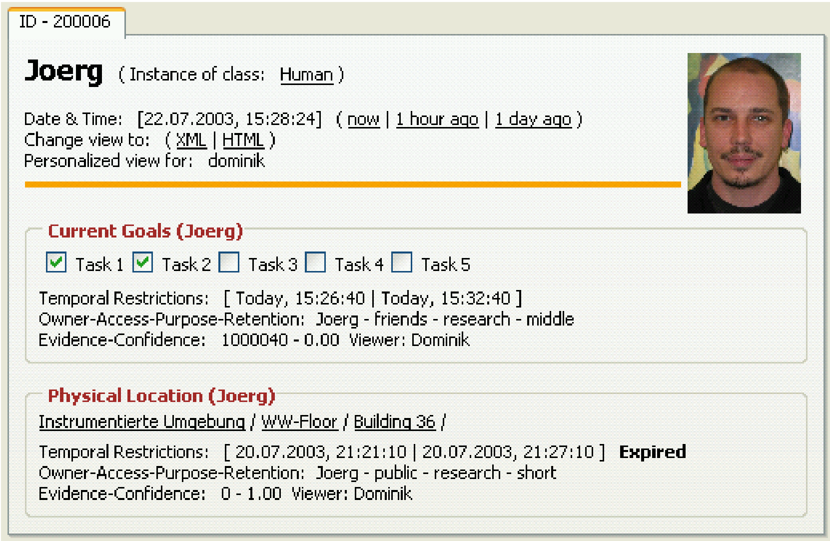


Figure 8.17: *Jörg's personalized page for friends*

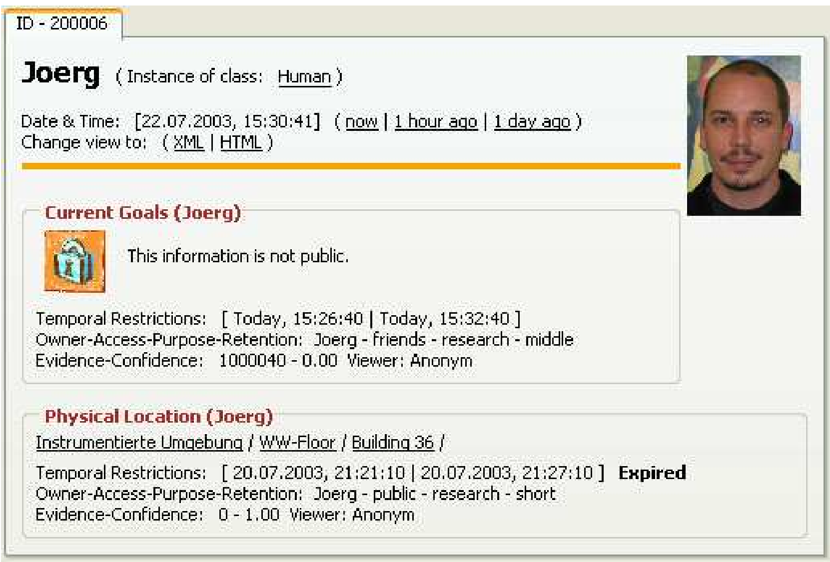


Figure 8.18: *Jörg's personalized page for public view*

9 APPLICATIONS & TESTING

This chapter presents sources and clients for ubiquitous user modeling. Especially applications that use and evaluate the u2m approach are discussed. Among others, there is the relation between a pedestrian navigation system, a museum and a shop and the ubiquitous user modeling approach presented.

9.1 Speech as a Source for Ubiquitous User Modeling

Speech plays an important role as an interaction modality in ubiquitous computing since in many application scenarios, the user can interact with the system by speech and gesture only in order to have the hands free.

9.1.1 M3I Gender and Age Detector

Speech contains meta information about the speaker. In most cases we can recognize the gender of a speaker, estimate the age and feel in what mood the speaker is in by hearing someone's voice. In [Müller and Wittig, 2003] and [Wittig and Müller, 2003] an approach is presented on how to use speech as as source for user modeling in a ubiquitous context. The results of their study imply that one can indeed successfully extract higher level information from the raw speech data. They divide the relevant features into three different levels of abstraction. On the lowest level, there are *acoustic* features that are related to the signal's power and frequency and their changes over time like jitter and shimmer. On the second level, there are *prosodic* features like intonation, stress, rhythm, speech rate and pauses. Finally, on the third level of abstraction, there are *linguistic* features that refer to the syntactical structure of the utterance, the number and category of the words and their semantic content.

The speech of the user is recorded on a mobile device (PDA) and then streamed to a server over a wireless network connection in the M3I[1] architecture. On the server side the relevant features are extracted with the help of machine learning techniques and Bayesian networks. The corresponding values are used to classify the speaker according to age data type: elderly/non-elderly and gender data type: female/male. The age of the user may

[1] M3I = a mobile, multi-modal and modular interface as part of the COLLATE project: Computational Linguistics and Language Technology for Real Life Applications http://www.collate.dfki.de/

contribute to derive adaptation decisions that can be important to make it easier for the elderly to access for example modern computing facilities in their everyday lives. According to [Jorge, 2001] elderly people often suffer from cognitive disabilities like age degenerative processes, short-term memory problems, motor impairments and reduced visual- and auditory capabilities. A reduction of the number of menu items per level in the menu hierarchy is suggested in [Wittig and Müller, 2003], while in [Müller and Wasinger, 2002] it is suggested that the speech output should be slower and the graphical user interface should be clearer in that the toolbars, buttons maps and text be displayed in a larger format. However, once the two user model demographics `age` and `gender` are sent to the U2M UserModelService, every client can independently decide on how to react and adapt to this information.

Speech symptoms of `cognitive load` and `time pressure` were analyzed in [Müller et al., 2001]. In the next subsection we do not use speech features to detect higher level user model information, but the other way round we use user model information from the user model server to adapt automatic generated speech.

9.1.2 Adapted Speech Generation

In [Wasinger et al., 2003a] we analyzed how to adapt spoken and visual output for a pedestrian navigation system, based on given SITUATIONALSTATEMENTS. In this section we focus on adapted speech generation. The formant synthesis[2] used in our system allows for a variety of acoustic parameters to be modeled. These can be explicitly set by the user through the programs menu, or implicitly defined by the identified causal relationships. These parameters can be modified to improve the quality and intelligibility of speech for a given context and for different locations along a users trip. The parameters are:

- different languages and dialects (e.g. UK English, Scottish).
- physical voice characteristics (e.g. pitch baseline, speed, volume).
- speaking style (e.g. whisper, monotone).
- prosody (e.g. emphasis, pauses, tones).

Adapting speech output is achieved by specifying a language and its dialectal form, along with the gender and age. The available voice characteristics include a range of physical factors like: pitch baseline and fluctuations, level of breathiness or roughness, as well as speech rate and volume. Additionally, depending on the context and environment, the speaking style can be adapted to suit a format such as whisper or monotone. To enhance clarity, or to avoid ambiguity of instructions, additional prosodic cues can be introduced. Such cues include emphasis, additional pauses, tones and phrase accents, which can all be assigned to the input. Prosodic signals such as cues for prominence and phrasing are crucial. We split the utterance into reasonable chunks and assign variables (e.g. pitch contour), appropriate for the type of output. In the two examples provided below, we manipulate physical voice characteristics and prosody, based on the two sentence types *instruction* and *description*.

[2] Formant Synthesis: `http://web.mit.edu/speech/www/history.html`

Two general studies that contributed to the adaptation of speech output comes from synthesized speech perception [Chasaide and Gobl, 2001] and German intonation [Grice and Baumann, 2002]. Both studies highlighted areas where it may be useful to modify the speech parameters. The following two examples by Dominica Oliver illustrate the results. The utterances (1) and (2) are of type description and instruction respectively.

1. ‘vs60 Das Gebäude ‘vs50 E1 1, ist der Sitz des Lehrstuhls ‘vs60 Professor ‘vs50 ‘2 Wahlster ‘/.

   [The building E1 1 is the location of Professor Wahlster’s professorship.]

   We increase the tempo of known tokens (‘vs60 Das Gebäude), put additional emphasis on the accented word (‘2 Wahlster) and assign a large pitch fall at the end of the sentence for a more perceived finality (‘/).

2. ‘vb60 Gehen ‘0 Sie 210 ‘ar Meter ‘%, ‘vs60 biegen Sie dann nach ‘vs40 rechts ab ‘%%, ‘vs50 in die ‘0 Max ‘2 Diamand-Strasse ‘/.

   [Walk 210 meters, and then turn right into Max-Diamand-Street.]

   We have a longer sentence, which has been split up into two chunks. Similarly, important pieces of information are pronounced slower (‘vs40 rechts) than the rest. The pitch baseline has been lowered to achieve a more factual and instructional tone (‘vb60). In the compound (Max-Diamand-Strasse), the middle token is additionally emphasised (‘2 Diamand) and the first token is de-accented (‘0 Max). At the end of the first phrase, we assign a rising tone to the last content word (‘ar Meter) and also add a pitch rise (‘%). We also add a phrase-final continuation rise (‘%%), which functions as a cue to the listener that more information follows.

Apart from achieving general clarity and ease of comprehension, speech parameters can also be modified to suit people with special needs such as the elderly. For example, studies show that the understanding of synthetic speech decreases with age, reaching around 60% loss for the older part of the population [Eskenazi and Black, 2001]. This loss can however be minimized through changes in speech parameters such as speech rate and volume. In the above study, it was also reported that in a semantically restricted domain, the ability to predict keywords within speech segments, remained constant with age. This is advantageous, because even if an elderly user does not understand all that was said, we can still rely on the user’s own cognitive ability to recognize the primary keywords in our navigation domain.

Psychological studies suggest that people use voice characteristics to assess personality [Nass and Lee, 2001]. When exposed to synthetic (clearly non-human) voices, people still assign personalities to the voices. Furthermore, the study showed that people seem to be attracted to voice characteristics exhibiting ’personality’ markers similar to their own. The voice characteristics found to be responsible were intensity, mean fundamental frequency, frequency range, and speech rate. By linking these parameters to the user model, we could in practice evoke trust or liking in the user, which is important for the interface design. In the next section we move from speech to biodata.

9.2 Biodata as Source for Ubiquitous User Modeling

Sensing and especially mobile sensing of physiological user data will become important for ubiquitous and wearable computing. A motivated application is to prevent workers in industry and transport from dangerous situations. This section presents an alarm manager as well as a rating mechanism based on so-called *biodata*.

9.2.1 AlarmManager

In [Brandherm and Schmitz, 2004] an alarm manager is described that is based on low level physiological user data. This alarm manager[3] is a service that triggers different kinds of notifications in an intelligent environment according to the estimated state of the user. The sensor data is interpreted by dynamic Bayesian networks on a personal mobile device. Figure 9.1(a) shows the experimental mobile setting. Since bio sensing data needs an extended privacy treatment, their system stores its data in form of SITUATIONALSTATEMENTS on the privacy enhanced U2M UserModelService, as described in [Heckmann et al., 2005a]. This server also offers a visualization component for introspection, see figure 9.1(b).

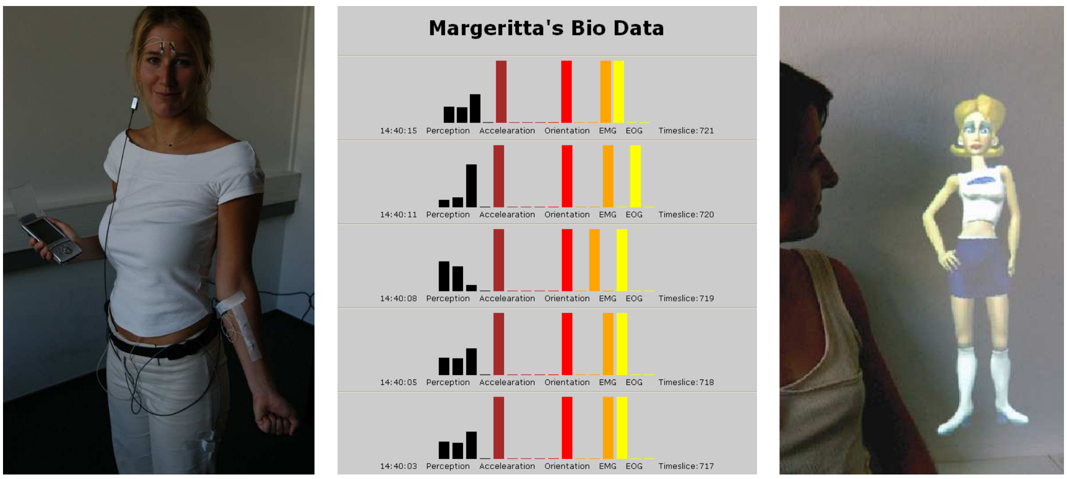


***Figure 9.1**: (a) Mobile BioSensing, (b) U2M's representation of physiological user data and (c) appearing avatar as final alarm notification*

The demonstration scenario is part of the airport scenario as introduced in chapter 1.1. The user waits for the departure of a flight and spends the remaining time shopping in the duty free area. When the boarding time is approaching, the AlarmManager starts to notify the user about the upcoming boarding of the plane, by acoustic and visual notifications that range from a short sound from the PDA to a virtual avatar appearing in the environment and calling the user directly until it is assumed that the user has responded to the message. If

[3] This research is being supported by the German Science Foundation in its Collaborative Research Center on Resource-Adaptive Cognitive Processes, SFB 378, Project EM 4, Real, and Project EM 5, READY.

the arousal value remains below a certain threshold, it must be assumed that the user did not notice the message and the next notification step will be initiated.

The biodata about the state of the user is gathered with a so-called *Varioport*, a mobile device that allows for recording signals from environmental and physiological sensors. It is a small and compact recorder especially for mobile acquisition of psychophysiological or environmental data. Up to nine sensors can be attached simultaneously. Available bio-sensors and environmental sensors are:

- *Electromyogram (EMG)* for acquisition of muscle tension
- *Electrooculogram (EOG)* for aquisition of eye movements and blinks
- *Electrocardiogram(ECG)* for acquisition of heart rate and interbeat intervals
- *ElectrodermalActivity (EDA)* sensor for acquisition of spontaneous fluctuations and average skin conductivity
- *Accelerometer (ACC)* for three-dimensional acquisition of accelerations (environmental sensor)
- *Movement sensor* to detect movements (environmental sensor)

The Varioport itself already preprocesses incoming data with filtering, pre-amplification and digitalization whereby the frequency of retrieving and storing data locally is configurable. In the example scenario an electromyogram sensor (EMG) was placed at the forearm, an electrooculogram sensor (EOG) between the eyebrows and an acceleration sensor at one thigh as shown in figure 9.2.

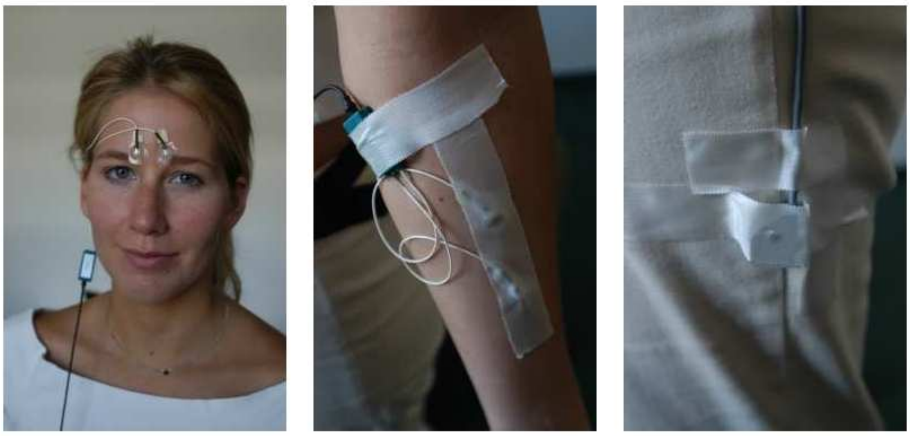


***Figure 9.2**: The three bio sensors (a) Electrooculogram (EOG), (b) Electromyogram (EMG) and (c) Acceleration (ACC) according to [Brandherm and Schmitz, 2004]*

9.2.2 BioRating

The BioRating interface also bases on data from bio-sensors. However the data is actively used in the opposite direction: by minimal muscle contraction, a rating `good`, `neutral` or `bad` is produced and sent to the U2M UserModelService. The internal processing of the sensory data is realized with specialized dynamic Bayesian Networks. The system has recently been developed by Boris Brandherm[4] and has been used within the SPECTER[5] shopping scenario. The idea behind BioRating is that if the user compares two products that already occupy the two hands, while additionally the user talks to the smart shopping assistant, the BioRating still fits smoothly into the interaction paradigm.

9.3 Implicit Manual Input as Source for User Modeling

Slightly more indirect than features in speech and biodata is the user's implicit manual behavior with human-computer interaction devices. The received manual input data is still low-level and needs to be interpreted with statistical methods first.

9.3.1 Symptoms of Cognitive Load and Time Pressure

In [Lindmark and Heckmann, 2000] we analyzed symptomatic behaviors caused by cognitive load and time pressure in manual input like *clicking an item* or *scrolling to a desired position* or *pressing a key on a keyboard*. Symptomatic behaviors for these manual inputs are for example *clicking next to target* or *over scrolling* or *high forced pressing*. We identified dependencies by literature research and implemented a Bayesian network as shown in figure 9.3(a) that evaluates such symptomatic behaviors and estimates the user model dimensions `cognitive load`, `time pressure` and `lack of knowledge` (of the current system). Figure 9.3(b) shows the conceptual model of this approach.

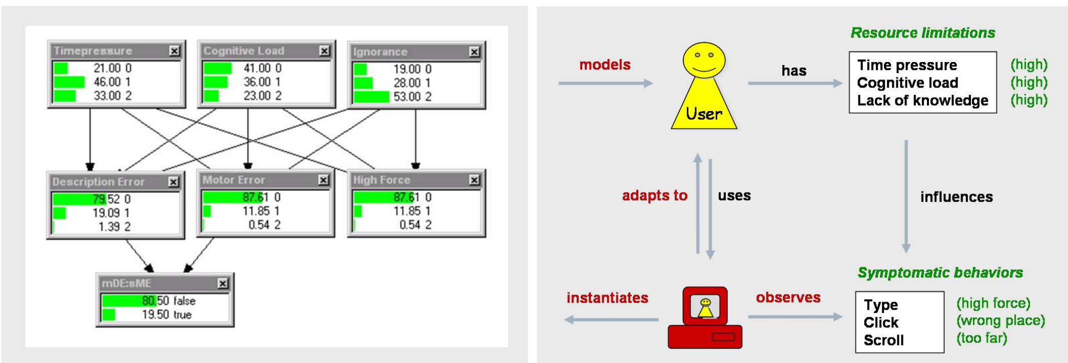


Figure 9.3: *Manual input with its (a) Bayesian Network and (b) conceptual model*

[4] BioRating contact: `http://w5.cs.uni-sb.de/~borisbra/`
[5] SPECTER homepage: `http://www.dfki.de/specter/`

9.4 Navigation as Source for Ubiquitous User Modeling

Being at the right place at the right time is an essential precondition for most user interaction within the real world. Thus for any given user task, it is likely that a navigational sub-goal exists. In [Stahl et al., 2004] we give a survey of the research project REAL[6] where we relate navigation with other tasks in instrumented environments. Furthermore, UbisWorld is integrated into the overall system architecture. Walking, ambling, rushing, visiting, all these different pedestrian navigation tasks reveal interesting facts about the user. The location, position and orientation plays an important role in ubiquitous computing and user modeling.

9.4.1 M3I Personal Navigator

The M3i Personal Navigator, see [Wasinger et al., 2003b] or [Krüger et al., 2004], connects a variety of specialized user interfaces to achieve a personal navigation service spanning different situations. It combines a desktop event and route planner, a car navigation system, and a multimodal, in- and outdoor pedestrian navigation and exploration system for PDA's. The PDA component is multimodal in that it provides output in the form of combined 2D/3D graphics and synthesized speech, and permits input via a combination of speech and gesture interaction. The Personal Navigator is a hybrid navigation system on the PocketPC hardware platform. The mobile device sends all positioning data received by GPS and beacons to this service, which fuses the different sensor information and matches them with spatial knowledge of the environment. The positioning service returns a symbolic location identifier as modeled in the UbisWorld location model as well as geometric coordinates for map visualization and the generation of situated navigational aid. The user interface is based on a three-dimensional interactive map, see figure 9.4(a), showing the M3I Personal Navigator system running on a mobile device. Figure 9.4(b) shows the introspection of the visited places during the indoor walking in the U2M UserModelService.

In the M3I Personal Navigator, output adaptation is used to modify the type of objects and speech presented to the user (based on building types, street types, and points of interest), the number and size of the objects to present to the user, the type of modality used when presenting information to the user (speech, speech and graphics, or just graphics), and their format (2D/3D graphics, formant/concatenative speech synthesis). In [Wasinger et al., 2003a], we described a variety of SITUATIONALSTATEMENTS that can be broadly categorized into device, user and environment types. Device statements are listed as including `memory constraints` and `screen size`, while user statements include `role`, `age`, `gender`, `load`, `interests` and `preferences`. Environment statements include `noise level`, `light level`, and `crowdedness`. The following table illustrates SITUATIONALSTATEMENTS that could have an effect on output in the Personal Navigator.

User interest profile Interests such as *shopping* or *sightseeing* may be used to adapt the type of graphical objects and audio presented to the user, with a bias towards one.

[6]REAL is a Project of the Collaborative Research Program 378 *Ressource-Adaptive Cognitive Processes*

(a) (b)

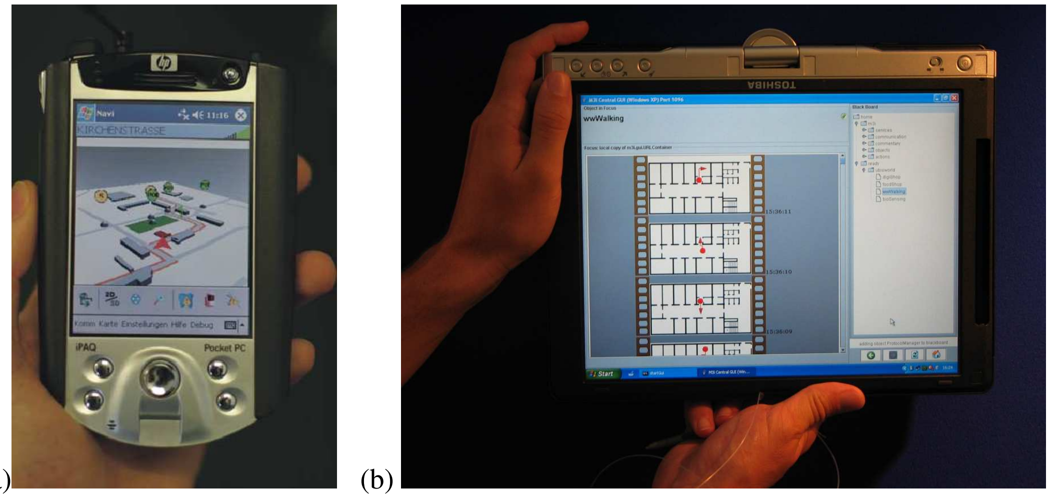


***Figure 9.4**: (a)* M3I *Personal Navigator and (b) U2M introspection of indoor walking*

User role Roles such as navigating from one place to another (targeted at business people), and exploration of a city (targeted at tourists) may be used to adapt the type of output to suite either navigation (e.g. “walk 100m, and then turn right into Stuhlsatzenhausweg”) or exploration (e.g. “you will see the cafeteria to your right”).

Collaborative user filters Models based on previous users’ interaction with a given map region may be used to determine the likelihood of an object being shown, especially if no user interests are known.

User location, orientation and time constraints Objects nearby to a user may be displayed and announced to a user before objects that are further away. Depending on time, more or less information may be presented to the user.

User load and environment context User loads arise when the user’s hands are busy (e.g. carrying a bag), the user’s eyes are busy (e.g. walking down stairs, or in a crowded environment), or specifically for input interaction the user’s voice is busy (e.g. talking to someone, or using a mobile phone). Depending on the type of load, different modality combinations may be used (e.g. speech only, speech and graphics, or just graphics). The speech and graphics output may also be simplified to compensate for user load.

Device memory Working memory in the Personal Navigator may be affected by the map size and map annotations (3D models of objects). Depending on the available working memory, the 3D graphics can be replaced by 2D graphics, and the concatenative speech synthesizer can be replaced in real time by a formant synthesizer.

Device location The location of the device will also have an effect on the output modalities that may be used (e.g. inside a bag, or in the user’s hands).

To illustrate the use of pre- and post-action adaptation, consider for example: *a user* *running down a street while trying to interact with the system.* In such a situation the effectiveness of intra-gestures and even handwriting may be affected by the speed of the user or the size of the graphical objects on the display. Pre-action adaptation would entail simplifying the graphics and speech output in an attempt to reduce the user's load[7]. This would be done by presenting fewer and larger objects on the display, and by only including speech grammars for currently visible objects, and only in keyword form. In comparison, post-action adaptation - assuming conflicting input information between the speech and gesture modalities exists - would then entail positively biasing the speech input over the gesture input, under the assumption that speed reduces the accuracy of gesture input. To conclude, the M3I Personal Navigator produces and uses a variety of SITUATIONALSTATEMENTS for adaptation and communicates them to the U2M UserModelService.

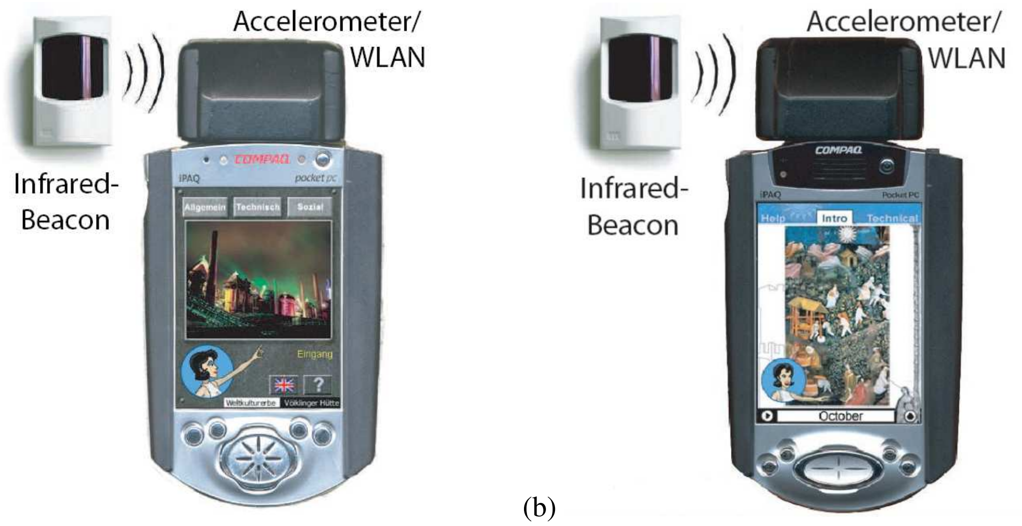
9.4.2 Indoor Positioning Service

Another application that feeds location data into the U2M UserModelService is the indoor positioning service as described in [Brandherm and Schwartz, 2005] and [Heckmann et al., 2005a] which is based on [Baus et al., 2002]. This service runs completely on a PDA, currently on an Hewlett Packard iPAQ. It uses infrared beacons and active RFID tags that are installed in the environment. The built-in infrared sensor and the active RFID reader card which is attached to the PDA via PCMCIA are used to calculate the current position of the device. The information about where the tags and beacons are installed in the environment, the so-called geo-coordinates, is stored in the RFID tags themselves. The sensor fusion and position calculation is done with the help of dynamic Bayesian networks. The calculated position together with the identification code of the PDA are then send to the U2M UserModelService via wireless LAN using `UserML/URI`. If the user model service already carries the information that this identified PDA is currently being used or held by a specific user, it can infer that the device position and the user position are the same and can thus update the user model accordingly. Furthermore, the indoor positioning service itself asks the U2M UserModelService if the user is wearing an acceleration sensor and if so, it can use this additional sensor data to reduce movement artifacts that sometimes appear due to the noisy nature of the RFID sensors.

9.5 Museum Visits as Source for Ubiquitous User Modeling

Two museums that have been instrumented for research on ubiquitous computing and user modeling are the *UNESCO world cultural heritage site "Old Völklingen Ironworks"* in Germany and the *Museo Castello del Buonconsciglio* in Italy.

[7]Professor Oberlander suggests to adapt the recognition grammar in the automatic speech recognition in this case as well because the user is likely to change the speaking behavior under high cognitive load.



(a) (b)

Figure 9.5: *(a) Old Völklingen Ironworks and (b) Museo Castello del Buonconsciglio*

9.5.1 Peach Museums Guide

The PEACH museums guide[8] as described in [Kruppa et al., 2005] and [Kruppa et al., 2003] is dedicated to the enhancement of cultural heritage appreciation. Using both mobile and stationary devices, it provides an educational and entertaining experience suited to each individual's background, needs and interests. In planning the personalized presentations, the system uses both an internal user model, and the U2M UserModelService from UbisWorld. Communicated SITUATIONALSTATEMENTS include the user's `location` and `orientation`, `modality`, `stereotype`, `visiting history` and `remaining visiting time` as discussed below. One goal is to build a museum guide system which combines the portability of personal digital assistants, see figure 9.5, with the multimedia capabilities of large, stationary devices. We introduce different characters which stand for different stereotypes within the museum context. According to the topic in the *Museo Castello del Buon Consciglio*, we chose an `artist` and an `aristocrat women`. Users are free to choose and exchange either character at will. While the artist will present the content in relation to artistic and technical background, the aristocrat women will focus on social and historical information. The character is chosen by the user at a so-called *Virtual Window*. It then moves from the virtual window to the mobile device of this particular user. Now the user may walk through the museum accompanied by the character of choice. Two different layouts were designed for each character, one for the mobile device, and one for the large screen virtual window.

The PEACH museums guide offers several degrees of freedom that have to be regarded in order to appropriately select and plan the personalized presentations for mobile devices

[8] PEACH homepage: `http://peach.itc.it`

Figure 9.6: *The museum visitor Jörg's user model in the mobile* PEACH *setting editor (on the left) and the U2M UserModelService introspection and editor tool (on the right)*

and virtual windows. The visitor's user model[9] introspection tools are depicted in figure 9.6 above while the situative context of the visitor is described below:

Location The location of the visitor in the museum. This location can be an absolute position regarding a well defined frame of reference (e.g. the visitor is in room A) or a relative position (e.g. the visitor is close to exhibit B). Both can be modeled in UbisWorld's qualitative location model as introduced in section 5.3.2 .

Orientation In a museum the orientation of the visitor is of utmost importance, since presentation should refer more to what the user is looking at than to information related to where the user is located.

Modality This is a list of all modalities that can be used to communicate with the visitor. Speech is the modality that is supposed to work all the time. However, the use of graphics and gestures depends on whether the visitor looks at the device or not.

Stereotype The actual stereotype that is used to classify the visitor. In our previous work, we have experimented with dynamic and more refined visitor models like interest models or visiting styles. In the current experimentation, we are simply using two groups: the more general or the more technically interested visitor.

Visiting History This history contains all information that the system could collect from the interaction with the visitor and information on the presentations that were generated and presented to the visitor. This includes visitor requests for more information on a

[9] User Model: `http://u2m.org/UbisWorld/UserModelEditor.php?subject=Joerg`

topic, how long the visitor looked at each exhibit, and whether the visitor interrupted certain presentation units as well as a representation of the text, the graphics and the videos that were generated during the visit.

Time left for visit This helps the system to distinguish between visitors who are in a hurry and visitors who have a lot of time available. Of course, those values should depend on the size and structure of the museum.

The situational context of the system is derived from different sensors, which are connected to the mobile device in corporation with the U2M UserModelService. Figure 9.5 shows the components of the mobile device that are used for this purpose. The position of the mobile device is determined by the use of long-life infrared beacons that are installed throughout the museum. Several beacons with different sending ranges, which are installed in the same location, allow us to roughly distinguish the distance of the mobile device to that location. Accelerometers provide the 3D-orientation of the device. This allows to estimate the orientation of the visitor and to determine whether the user is looking a the screen of the device since the device is held within a certain range of vertical angle range. All visitor interactions with the mobile device are recorded and sent to the service, where the visitor's situational context is constantly updated in form of SITUATIONALSTATEMENTS. To summarize, instrumented museums form an ideal playground to experiment with ubiquitous user modeling since the information can be shared with earlier and upcoming museum visits, either in the same or in different museums. Even the shopping behavior within the museums shop can be integrated within the user model. The next subsection analyzes shopping as a source for ubiquitous user modeling.

9.6 Shopping as a Source for Ubiquitous User Modeling

Another interesting area to apply results from artificial intelligence to everyday life is the shopping domain. In contrast to explicit human-computer interaction, where a user controls an application's behavior through buttons, menus, dialogs, or some kind of written or spoken command language, the shopping scenario allows for implicit interactions and is driven by the user's actions in the environment, as described in [Stahl et al., 2004].

9.6.1 Smart Shopping Assistant

The prototype of an intelligent, adaptive shopping assistant, named SMART SHOPPING ASSISTANT[10], offers value-added services in a shopping scenario, see [Schneider, 2003] and [Schneider, 2004]. The assistant uses Radio Frequency IDentification (RFID) sensors and plan recognition techniques to observe a shopper's actions directly. From the observed actions the system infers the shopper's goals. Using this information a proactive mobile assistant mounted on a shopping cart offers support during shopping. The type of value-added services offered, e.g. while buying groceries, range from product comparisons and analysis of goods in the shopping cart for cross-selling recommendations to the suggestions of

[10] SMART SHOPPING ASSISTANT homepage: `http://www.misch.net/ssa/`

recipes. Another speciality of the system lies in its ability to make use of a user model in order to adapt value-added services to special preferences like `nutrition habits` and `shopping lists` of the individual customer. During shopping, relevant user actions for example include `moving` around in the store, `looking` at items of interest or advertising displays, or physically `interacting` with products, see figure 5.29 on page 111. Relevant context includes the SITUATIONALSTATEMENTS in the U2M UserModelService, a Ubis-World location model and the products available in the store or involved in a user's action, see figure 5.20 on page 104. The SMART SHOPPING ASSISTANT comprises a central server, a modified shopping cart or basket and instrumented shelves. Both, the shopping basket and the shelf, have been equipped with RFID readers to allow for the recognition of products tagged with RFID transponders. In addition to the basket, a PDA is used as the primary user interface.

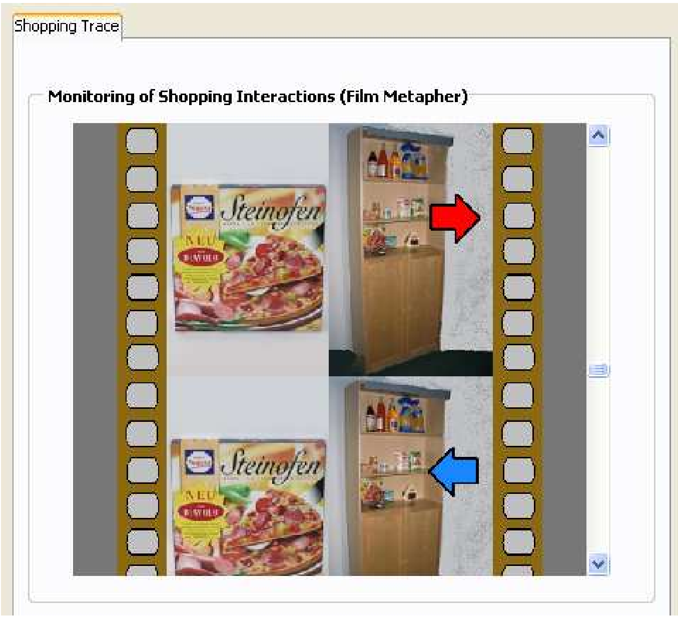


***Figure 9.7**: U2M UserModelService introspection of food shop interactions: film metaphor*

Alternatively to the basket, we have a shopping cart with integrated RFID antenna and a tablet PC mounted on the cart's handrail, hosting the shopping assistant, controlling the RFID reader and connecting the cart with the central server via wireless network. With this setup, user actions like taking a product and putting it down as depicted in figure 9.7[11] and figure 9.8[12] can be recognized by repeatedly polling the transponder's IDs in the antenna field of each shelf and shopping cart. These observations are fed into the shopping assistant application, which is provided as a service by the intelligent environment. The application reacts to these observations and the context provided by the user. It offers value-added services like product comparisons or cross-selling recommendations. Furthermore, the information

[11]Food Shop Interactions: `http://u2m.org/UbisWorld/FoodShopMonitorFilm.php`
[12]Food Shop Interactions for PDA: `http://u2m.org/UbisWorld/FoodShopMonitorPDA.php`

***Figure 9.8**: U2M UserModelService introspection of food shop interactions: resource-adaptive for a mobile device*

can also be sent to a presentation planning service, which selects the appropriate display and generates a resource-adaptive and user-adaptive presentation. Several versions of instrumented smart shopping shelves have been implemented, among which are a food shop, a digi-cam shop, a book shop and a mixed grocery shop.

9.7 Discussion of the Developed Tools

The developed tools for ubiquitous user modeling have been positively tested in several, independent projects with external experts[13]. It turned out that SITUATIONALSTATEMENTS are general enough to represent data in user modeling, context-awareness and resource-adaptive computing. `UserML` has shown to be "information equivalent" and "computationally efficient" in respect to RDF. `UserML/URI` is a powerful tool to exchange user model data via ordinary http requests. It has been used by most applications for the exchange of partial user models. The `u2m.org` user model and context service has proven flexibility, scalability and universal accessability in various projects. In one testing phase, the user model service handled over 40.000 statements without running into computational problems. In a pure ontology-based approach this would not have been possible. Several user interfaces have been built on `UserML` for introspection. The use of `Gumo` was especially important to

[13] SFB 2001, SFB 2004, SAB 2005

guarantee semantic uniformity in chains of adaptation that have been demonstrated in the museums and shopping scenarios. Another possible field for ubiquitous user model applications that has not been discussed so far is *eLearning* together with *LearnerModeling*. The analysis of the auxiliary `hasKnowledge` and the process of knowledge acquisition `hasLearned` could play an important role in further research. First experiments with applying `UserML` and integrating the `u2m.org` user model service into the *AHA! learning environment* have been undertaken and lead to promising results. No user studies have been executed so far to analyze the developed user interfaces. However, studies are planned. Especially the USERMODELEDITOR, the online tool to inspect, control and change one's personal data has to be evaluated.

Part IV

Discussion

10 CONCLUSION & OUTLOOK

The first section of this final chapter "Conclusion & Outlook" revisits the main research questions and summarizes their results and effects. The second section presents the scientific contributions, sorted by research areas, which is followed by an overall conclusion. The next section describes the impact of our research in terms of numbers and research cooperations. Finally, further research opportunities are compiled in the last section.

10.1 Revisiting the main Research Questions

In the outline of this thesis, we have formulated eleven research questions. These questions have driven the investigations and developments around the main theme of formalizing and exchanging user model knowledge in the era of semantic web and ubiquitous computing. Let's now revisit them and discuss their results briefly.

1. *How can we conceptualize the complex process of situated interaction for ubiquitous user modeling?*

   Situated interaction belongs to the research area of *Human-Computer Interaction*. However, since the user in ubiquitous user modeling does not interact with only one "computer" any more, the main task was to identify and conceptualize the "interaction partner": according to [Weiser, 1991], the interaction partner in ubiquitous computing is an *effective place*, which is nowadays called an *intelligent environment*, see [Coen et al., 1999]. We analyzed the user mobility and the interface transparency according to [Maffioletti, 2001], the symmetric multimodal interaction according to [Wahlster, 2003], and intelligent user interfaces according to [Maybury and Wahlster, 1998]. We divided between explicit, implicit and indirect interaction according to [Schmidt, 2003] and defined a clear model that integrates all relevant elements for mobile computing and ubiquitous computing with user-adaptivity, resource-adaptivity and context-awareness, according to [Kray, 2003].

2. *How can a user model exchange language be designed that is especially well-suited for the communication between different user modeling applications ?*

   Since descriptions of whole situations are relevant for ubiquitous user modeling, the first step towards a user model exchange language is to unify user-adaptivity, resource-adaptivity and context-awareness. This is realized on the basis of *Situation Semantics*, see [Barwise, 1981], which is for example extended with a privacy layer, according to [Kobsa, 2001b], and an explanation layer, according to [Kay et al., 2003]. The result is the new concept of SITUATIONALSTATEMENTS, which proves according to [Larkin and Simon, 1987] and the defined tasks (i.e. human readability) computational efficiency. The second step towards the design of the user model exchange language is according to [Stuckenschmidt and van Harmelen, 2005] the use of semantic web technology. We developed the RDF-based exchange language `UserML` with the innovation that the semantics is consequently defined in an external semantic web ontology.

3. *How can partial user models be communicated via the information structures within intelligent environments?*

   First of all, the implemented accretion method, see [Kay et al., 2002], allows to partition all user models into small partial models. Then, these partial user models can be represented in a variety of `UserML` versions, for example in file-size optimized, minimal XML documents for low band-width environments. However, the intelligent environments like the “Haus der Gegenwart” in Munich or the “inHaus” by the Fraunhofer-Gesellschaft provide web-service infrastructures with high bandwidths. We have implemented centralized and decentralized, as well as global and local web service networks with XML messaging that allow wireless and network-connected communication via HTTP protocols. The implemented `u2m.org` USERMODELSERVICE has been used by several ubiquitous applications to test the communication under real conditions.

4. *How can knowledge about user model dimensions be well organized for a semantic web ontology?*

   We started with the overview of modeled properties in user-adaptive systems by [Jameson, 2001a] and [Kobsa, 2001a] and investigated existing user modeling systems. We deduced specialized, domain specific extensions and engineered the general user model ontology GUMO. It is represented in the semantic web languages DAML+OIL and OWL. Especially the introduction of the user model `auxiliaries` and the user model `predicates` influenced the organization of the ontology. In order to enable the distributed extension and refinement of this ontology, we implement the online-available UBISONTOLOGYEDITOR that uses a tree browser for visualization.

5. *How can the changing physical and virtual environment around the user be represented uniformly?*

   On top of our new human-environment interaction model for ubiquitous computing we built an extended virtual world with the name UBISWORLD, that was inspired by the Blocks World, see [Slaney and Thiébaux, 2001]. The corresponding ontology UBISONTOLOGY with physical elements, spatial elements, temporal elements, activity elements, situation elements and inference elements defines a large T-Box and several A-Boxes to refer to the real world. In [Stahl and Heckmann, 2004b], we define a state of the art hybrid location model, according to [Leonhardt, 1998], to represent symbolic and geometric spatial references. Especially the spatial granularity levels, inspired by

[Dürr and Rothermel, 2003] and the UBISLOCATIONMONITOR complete the uniform representation of the environment.

6. *How can entities like locations and objects be identified uniquely and efficiently in distributed real-world applications with multi-user and multi-systems?*

   Uniform resource identifiers (URIs), see [Berners-Lee and Fischetti, 1999], were interesting candidates. However, more powerful, globally unique identifiers were needed for efficient identification in ubiquitous user modeling. We developed the unique naming concepts *UbisIdentifier* (Ubid) and *UbisExpression* (Ubex) as extensions to the URI naming concept. They allow to represent several resources in one expression. Together with *UbisFunctions* like `extension()` and `ancestors()` semantic preprocessing has been ported into the formerly pure knowledge representation part.

7. *How can users inspect and control their distributed user models?*

   The so called *scrutiny interfaces* according to [Kay, 1995] form a problem especially for ubiquitous user modeling, since not every system or device will be able to interact with the user directly. In such a case we use the virtual representative within UBISWORLD. Furthermore, according to [Hinde, 2003] and [Kobsa, 2001b] an extended amount of privacy issues have to be considered for distributed user models. We realized a USERMODELEDITOR as a web browser application for mobile and large-screen devices that already incorporates the privacy dimension of the SITUATIONAL-STATEMENTS. With this specialized online editor the user can inspect, change, delete and control his or her personal data that is stored on distributed systems, devices or user model repositories.

8. *How can the huge amount of (sensor) data be handled technically by the server?*

   Our solution points towards load balancing and ontological databases. Our approach allows the smooth integration of ontologies and distributed databases by offering `UserML` models in RDF and SQL. We have defined a set of hierarchical databases that balance the load of low level sensor data dynamically according to days, weeks, and months. Thus, we have developed a URI conform denotation model to efficiently identify each repository and statement.

9. *How can situation retrieval and conflict resolution be managed in such a distributed approach?*

   Information retrieval on distributed SITUATIONALSTATEMENTS is handled with its four steps: `select`, `match`, `filter`, `control`. It is partly based on SQL, while the developed SITUATIONALQUERIES and query language `UserQL` allow to access distributed repositories in parallel. Interesting is the integrated semantic functionality within the query process like *sameAs*, *closeBy*, or spatial *inclusion*. Our approach is based on conflict resolution strategies with situational conflict categories, comparable to [Brownston et al., 1985] and [Ram and Park, 2004]. The retrieval mechanism contains a multi-level conflict resolution strategy that resolves queries according to given preferences on meta information. The problem of detecting conflicting statements has been approached with fine-grained $\langle S^*, A^*, P^*, R^* \rangle$ conflict sets.

10. *How can the integration of instantiated partial user models be realized within ubiquitous user modeling?*

    Sharing and reusing of knowledge is according to [Holsapple and Joshi, 2002] a typical reason for constructing an ontology. However, since SITUATIONALSTATEMENTS are rather database oriented, we have demonstrated a method to integrate them into GUMO and UBISONTOLOGY. For the other way round, we enabled the integration of our ontologies into a database and presented a user model integration method that reverts back to the conflict resolution strategy for SITUATIONREPORTS. The conflicts that may occur when partial user models are integrated are resolved at the time of request.

11. *How does a reasonable architecture for decentralized user modeling look?*

    We analyzed the role of mobile devices in decentralized user modeling according to [Fink, 2004] and [Maffioletti, 2001] and presented the modularized architecture of the `u2m.org` USERMODELSERVICE. It is an application independent service with a distributed approach for accessing and storing spatially distributed partial user models, which clearly separates between statements, services, ontologies, interfaces and applications. The communication between modules and applications is web service based with `UserML` and `UserQL`.

10.2 Scientific Contributions (sorted by Research Areas)

Ubiquitous User Modeling incorporates integrating aspects from the three research areas *User Modeling*, *Ubiquitous Computing* and *Semantic Web*. We have presented new results, models, methods, tools and ontologies that contribute to all of these research areas.

10.2.1 User Modeling

[Orwant, 1996]'s initial conclusion and demand to the user model community was ...

> *What we need is a protocol for encoding information about users, so that the applications and techniques developed at each site will be usable at every other site. ... There are so many things to sense about people, and so many scenarios and uses for the resulting inferences, that any communication mechanism must be open ended, so that when new sensors are developed, or new behavior domains tracked, or new modeling techniques employed, they can be incorporated without breaking previous implementations. ... But we risk creating something too general to be useful.*

... which has, for the first time, been entirely solved with semantic web technology in our presented approach with the (first and only) specialized user model exchange language `UserML` and the (first and only) general user model ontology GUMO. Both are influencing contributions to the user modeling community. The combination of `UserML`, with the integrated new mechanism of `UbisIdentifiers` and `UbisExpressions`, the according query language `UserQL`, and the ontology application language `UserOL` form the needed flexibility and power to contribute an *open ended protocol*, as issued by [Orwant, 1996], which is still not *too general to be useful*. The developed general user model ontology with its already large set of basic user dimensions can constantly be extended by the user model community with the newly developed UBISONTOLOGYEDITOR, that can easily be accessed via a web browser. The general user model ontology GUMO currently collects and organizes over 1000 user model dimensions. The major classes of `contact information`, `demographics`, `ability`, `proficiency`, `personality`, `characteristics`, `emotional state`, `mental state`, `role`, `motion` and `nutrition` cover most of the dimensions that are used in related user-adaptive systems. The ontology profits from the design of SITUATIONALSTATEMENTS since *user model auxiliaries*, *user model dimensions* and *user model ranges* are modeled individually, which increases the expressivity. Furthermore, GUMO cooperates with other existing ontologies, and it contains information for optimizing the conflict resolution strategies during the smart situation retrieval process. The users can introspect and edit their user models with the newly developed USERMODELEDITOR.

The third major issue that has been solved (both theoretically and practically) in this thesis is the one of *centralization versus decentralization* of user models: should the user model be stored locally or globally, centralized or decentralized? While according to [Fink and Kobsa, 2002] previous user modeling systems stored user models in centralized databases and knowledge representation systems, our presented solution starts with the *Open World Assumption*, which is already reflected in the design of the model of SITUATIONAL-STATEMENTS. They allow any body, any system and any sensor to state what is known about the world, even if these distributed sets of statements are conflicting. A part of the solution is a complex, requestor-adaptable conflict resolution mechanism that can be applied for each request. The decentralization problem is solved by the web service idea where the data is not stored centrally and nor is it communicated to a central point, but registry information about user model repositories and user model servers integrate all distributed statements virtually. Thus a *requestor-adaptive, virtually centralized user model* is generated dynamically, which is spatially distributed, but logically centralized and uniform.

The fourth user modeling issue, that has been incorporated into the whole family of user modeling tools is *privacy handling*. Even though this topic has been recognized from the beginning of the research in user modeling, the newly realized idea in our approach is that the privacy status of information is not only handled by the user model server, which decides if any requester is allowed to receive information or not, but the intended privacy settings are already built into the basic structure of each individual SITUATIONALSTATEMENT. The great advantage is that in the highly distributed world of different ubiquitous computing environments, the privacy settings go along with the user model information and are not lost during the communication process.

10.2.2 Ubiquitous Computing

Our major scientific contributions to the research area of ubiquitous computing is the so-called UBISWORLD and its UBISONTOLOGY. UbisWorld is the first large scale blocks world for simulating, monitoring and controlling ubiquitous computing environments that is based on flexible semantic web ontologies. It represents persons, objects, locations together with their properties and features, also times, events and inferences. UbisWorld provides a virtual counterpart to each real world element. Several tools were developed to support

UbisWorld, including the UBISWORLDMANAGER, the UBISLOCATIONMONITOR and the UBISONTOLOGYEDITOR.

Another result that was induced by ubiquitous computing and its model of situated interaction was the integrated treatment of user models, context models and resource models. Thus the three research areas of user modeling, context-awareness and resource-adaptivity are unified within our approach.

10.2.3 Semantic Web

The semantic web technologies were mainly used as a means to combine user modeling and ubiquitous computing; however, two scientific contributions have also been developed in this research area. The first one is the extension of the RDF resource concept: more flexibility is reached by the `UbisId` naming convention, that elegantly solves the problem of unique identifiers within large, globally-connected instrumented environments, which are still human readable. The more general concept of `UbisExpressions` allows us to define a list of resources within one XML attribute plus semantic functionality, thus already existing XML applications become more expressive. Newly developed applications are informationally equivalent according to [Larkin and Simon, 1987], however they can be designed to be more dense, which again helps human readability. This technique is used to model groups of individual users within one resource.

The second contribution solves the problem of *scalability* in ubiquitous user modeling, which involves the question of how to handle the mass of data that is constantly produced. If we simply add all situational information to the semantic web ontology, the system won't be tractable. The solution is a flexible integration of databases and ontologies. The mass of situational statements are modeled as n-ary relations and stored and retrieved from online available databases, while the semantic concepts are stored in semantic web ontologies. Both mechanisms are unified, since the ontological data also has a database representation for fast retrieval and to control the virtual counterparts and vice-versa, the retrieved situational statements also have a semantic web language representation that is used for ontological reasoning.

10.3 Conclusion

To conclude, we presented a platform-independent, semantic web based, ubiquitous user modeling service for the deployment of (augmented) real world environments as well as world wide web applications, which has vastly been tested by several independent applications. One great advantage of our broad, general service approach, which covers all necessary aspects of semantic management of distributed user models, is that we can design and adapt all modules as needed. For example, we can move a concept from the syntactical representation part into the inferential conflict resolution part, or we can take something from the underlying conceptual model and put it into the ontology and find the most efficient overall system. With this flexible and powerful tool set, we are able to keep pace with the currently fast changing paradigms in human-computer interaction. Furthermore, we have the great opportunity to experiment in all directions. Even though most of the modules and methods are not complex by themselves, the whole system altogether forms a new, complex, distributed system for the integrated management of distributed user models. We presented new and innovative services for user modeling and ubiquitous computing. Furthermore we have proven that ongoing user modeling with a variety of systems and applications is possible, since we have solved the problems of scalability and distribution.

10.4 Impact of our Research

During the last years of this thesis, the newly defined topic of *Ubiquitous User Modeling* has gained more and more interest throughout several research communities. The two latest external publications are *Ubiquitous User Modeling in Recommender Systems*, see [Berkovsky, 2005] and *Agent-Based Ubiquitous User Modeling*, see [Lorenz, 2005].

There are currently over one hundred registered users in UBISWORLD, and over the last months the visits to the `u2m.org` website are constantly increasing to approximately 2500 pageviews per month. Unfortunately the number of downloads of the GUMO ontology is not known, since the distributer's domain `www.daml.org/ontologies`, where our user model ontology has been published since 2003, does not provide a counter. However, in this year's user modeling conference (2005) in Edinburgh, GUMO was often cited or mentioned in panel discussions as being the only specialized ontology for user models.

With the poster and paper: *A user modeling markup language (UserML) for ubiquitous computing*, see [Heckmann and Krüger, 2003], we won the "best poster award" at the user modeling conference 2003 in Johnstown, Pennsylvania.

A further impact of our research is indicated by the increased number of research visits and cooperation with other research groups. The following listing names most of them that have already been taken place or that are planned for the coming months:

- Paul de Bra, Lora Aroyo, Vadim Chepegim, *Eindhoven University of Technology, The Netherlands*: Applying SITUATIONALSTATEMENTS and `UserML` to AHA!
- Helmut Prendinger, *National Institute of Informatics in Tokyo, Japan*: Using GUMO for Life-Like Characters
- Cristina Gena, Francesca Carmagnola, *Universita degli Studi di Torino, Italy*, Vania Dimitrova, Alexander Kröner, *German Research Center for Artificial Intelligence*: Exchange of user modeling rules and adaptation rules, based on the `u2m` approach
- Judy Kay, Bob Kummerfeld, *University of Sydney, Australia*: Exploring possible collaborations between the `u2m` USERMODELSERVICE and the PERSONIS server
- Julita Vassileva, Gordon McCalla, *University of Saskatchewan, Canada*: Decentralized user modeling
- Tsvi Kuflik, Shlomo Berkovsky, *University of Haifa, Israel*: Co-organizing a workshop on ubiquitous user modeling
- Tatiana Gavrilova, *St. Petersburg State Polytechnical University, Russia*: Ontology modeling for user modeling

- Erica Melis, Martin Homik, *German Research Center for Artificial Intelligence*: Using GUMO and the u2m approach with ePortfolio
- Peter Brusilovsky, Michael Yudelson, Sergey Sosnovsky, *University of Pittsburgh, USA*: Integration of GUMO and the user modeling meta-ontology
- Heiner Stuckenschmidt, *Vrije Universiteit Amsterdam, The Netherlands*: Semantic web services for information exchange and ontology merging
- Adam Pease, *Knowledge Systems at Teknowledge, USA*: Integration of GUMO and SUMO
- Berardina Nadja De Carolis, *University of Bari, Italy*: Applying UserML
- Andreas Schmidt, *FZI Research Center for Information Technologies Karlsruhe, Germany*: Unified user context management
- Rodrigo Campiolo, *Universidade Federal de Santa Catarina, Brasil*: Simulation and modeling in pervasive computing with UBISWORLD

Another impact of our research is given by the two workshops: *Workshop on User Modeling for Ubiquitous Computing*, that was held in conjunction with the International User Modeling Conference (UM 2003) and the *Workshop on Ubiquitous User Modeling* (UbiqUM'06), that will be held in conjunction with the European Conference on Artificial Intelligence (ECAI 2006).

10.5 Further Research Opportunities

Even though most models, formalisms, methods, tools and ontologies that are introduced in this thesis already reached a somehow mature state of development, most of them could still be refined or extended. However, the following list of questions points to further research opportunities, that promise interesting results and insights for the research on ubiquitous user modeling:

- *How would a conceptual model for ubiquitous user-adaptive interaction look like, if we incorporate all ideas from mobile computing, ubiquitous computing, ambient intelligence, virtual reality and augmented reality?*
  The research area of human-computer interaction is undergoing a tremendous change from the desktop metaphor via the mobile-computing metaphor to the augmented-ubiquitous metaphor. The concepts for context-awareness and user-adaptivity have to be adjusted according to these new models.
- *How would the conceptual model for ubiquitous user-adaptive interaction look like, if we extend the adaption concept from one single user to groups of users?*
  User-group modeling is a new research issue that could probably build on the presented solution for the integration of distributed single-user models, and on the S^* extension within the conflict resolution. Group modeling could be useful for adaptive museum

guides, family adaptation or interaction devices (like large screens) that are used by several users at the same time.

- *Could possible correlations between conflict resolution strategies and auxiliary and predicate equivalence classes lead to a better situation retrieval algorithm?*
  This new research issue came up while analyzing smart situation retrieval and conflict resolution. If such a correlation can be found, recommended resolution strategies could be attached to the GUMO ontology elements.

- *How can location-awareness and spatial meta-data lead to new spatial filters for UserQL and smart situation retrieval?*
  Once the realized intelligent environments of ubiquitous computing leave the size of a room or a house and reach the size of a smart village or city, the implication of spatial filters in the smart situation retrieval process has to be analyzed and extended.

- *How can the transparency be increased by an explanation component for the conflict resolution strategies?*
  In the smart situation retrieval process, filtering decisions that highly influence the report are automatically made by the system. However, a new explanation component could support better transparency and probably explain the decisions to the user.

- *How could an extension of the privacy and security management look?*
  The `access` attribute in the privacy process could be extended from the *public-friends-private* setting to the integration of different groups like family, colleagues or sport teams with extended rights management which would probably lead to an increased transparency.

- *How could a profound evaluation of developed user interfaces, especially the user model introspection tools, be designed?*
  These evaluations could for example reveal significant results on the user behavior during the introspection and privacy editing phase. Hence, the corresponding underlying models could be adjusted to the user's need.

- *How could the semantic transfer of user model information be realized, if the information that has been collected by different applications use different ontologies or if the applications use different interpretations for the same concepts in different contexts?*
  Expected research results in ontology mapping, ontology merging or semantic translation will most probably have a great influence on ubiquitous user modeling.

Part V

Appendix

A Appendix - Syntax Specifications

The SITUATIONALSTATEMENTS and SITUATIONALQUERIES are defined in three syntax variations, namely *Min*, *Mix* and *Max*. All three can be used in parallel. The *Min* variation is the shortest one and good for exchanging the data. The *Mix* variation is easy to be read for inspection because of more structure, while the *Max* variation can carry element data instead of attribute data only.

A.1 Syntactic Specifications of SITUATIONALSTATEMENTS

A.1.1 Specification: SITUATIONALSTATEMENT / XML

In this major XML-representation for SITUATIONALSTATEMENTS, the actual information is put into XML-elements, while the according attribute-groups like *explanation* or *privacy* are omitted. Namespaces can be added. The variables `a1` to `a25` carry information in the *UbisExpression* format. Empty elements can be omitted.

```
<statement>
      <subject>    a1  </subject>
      <auxiliary>  a2  </auxiliary>
      <predicate>  a3  </predicate>
      <range>      a4  </range>
      <object>     a5  </object>
      <start>      a6  </start>
      <end>        a7  </end>
      <durability> a8  </durability>
      <location>   a9  </location>
      <position>   a10 </position>
      <source>     a11 </source>
      <creator>    a12 </creator>
      <method>     a13 </method>
      <evidence>   a14 </evidence>
      <confidence> a15 </confidence>
      <key>        a16 </key>
      <owner>      a17 </owner>
      <access>     a18 </access>
      <purpose>    a19 </purpose>
      <retention>  a20 </retention>
```

```
      <id>         a21 </id>
      <unique>     a22 </unique>
      <replaces>   a23 </replaces>
      <group>      a24 </group>
      <notes>      a25 </notes>
</statemtent>
```

A.1.2 Syntactic Variation: SITUATIONALSTATEMENT / XML (Max)

In this maximal syntactic variation, the attribute-groups introduce a new level in the XML tree. It is a one-to-one equivalent realization to the model of SITUATIONALSTATEMENTS as defined in 4.4. The namespace `st:` should be read as "statement". It is mapped to `http://www.u2m.org/2003/02/situation/`.

```
<st:statement>
      <st:mainpart>
            <st:subject>    a1  </st:subject>
            <st:auxiliary>  a2  </st:auxiliary>
            <st:predicate>  a3  </st:predicate>
            <st:range>      a4  </st:range>
            <st:object>     a5  </st:object>
      </st:mainpart>
      <st:situation>
            <st:start>      a6  </st:start>
            <st:end>        a7  </st:end>
            <st:durability> a8  </st:durability>
            <st:location>   a9  </st:location>
            <st:position>   a10 </st:position>
      </st:situation>
      <st:explanation>
            <st:source>     a11 </st:source>
            <st:creator>    a12 </st:creator>
            <st:method>     a13 </st:method>
            <st:evidence>   a14 </st:evidence>
            <st:confidence> a15 </st:confidence>
      </st:explanation>
      <st:privacy>
            <st:key>        a16 </st:key>
            <st:owner>      a17 </st:owner>
            <st:access>     a18 </st:access>
            <st:purpose>    a19 </st:purpose>
            <st:retention>  a20 </st:retention>
      </st:privacy>
      <st:administration>
            <st:id>         a21 </st:id>
            <st:unique>     a22 </st:unique>
            <st:replaces>   a23 </st:replaces>
            <st:group>      a24 </st:group>
            <st:notes>      a25 </st:notes>
      </st:administration>
</st:statemtent>
```

A.1.3 Syntactic Variation: SITUATIONALSTATEMENT / XML (Mix)

The two following syntactic variations represent the SITUATIONALSTATEMENT-attributes as XML-attributes. This has the advantage of compactness, but the disadvantage of restricted expressivity. The Mix version is a mixture between Max and Min and as it turns out, statements in this version are good for online reading by humans.

```
<statement>
      <mainpart
            subject =    "a1"
            auxiliary =  "a2"
            predicate =  "a3"
            range =      "a4"
            object =     "a5" />
      <situation
            start =      "a6"
            end =        "a7"
            durability = "a8"
            location =   "a9"
            position =   "a10" />
      <explanation
            source =     "a11"
            creator =    "a12"
            method =     "a13"
            evidence =   "a14"
            confidence = "a15" />
      <privacy
            key =        "a16"
            owner =      "a17"
            access =     "a18"
            purpose =    "a19"
            retention =  "a20" />
      <administration
            id =         "a21"
            unique =     "a22"
            replaces =   "a23"
            group =      "a24"
            notes =      "a25" />
</statement>
```

A.1.4 Syntactic Variation: SITUATIONALSTATEMENT / XML (Min)

This minimal syntactic variation only uses XML-attributes and thus has the disadvantage of restricted expressivity. Nevertheless, it is a premium choice if bandwith for mobile systems is a problem.

```
<statement
      subject =    "a1"
      auxiliary =  "a2"
      predicate =  "a3"
      range =      "a4"
      object =     "a5"
      start =      "a6"
      end =        "a7"
```

```
      durability = "a8"
      location =   "a9"
      position =   "a10"
      source =     "a11"
      creator =    "a12"
      method =     "a13"
      evidence =   "a14"
      confidence = "a15"
      key =        "a16"
      owner =      "a17"
      access =     "a18"
      purpose =    "a19"
      retention =  "a20"
      id =         "a21"
      unique =     "a22"
      replaces =   "a23"
      group =      "a24"
      notes =      "a25"
/>
```

A.2 Document Type Descriptions for SITUATIONAL-STATEMENTS

For the reason of completeness, the document type descriptions of all four syntax variations of SITUATIONALSTATEMENTS/XML are listed in this section of the appendix.

A.2.1 SITUATIONALSTATEMENT / DTD

```
<!ELEMENT  statement (subject, auxiliary, predicate, range, object,
  start, end, durability, location, position,
  source, creator, method, evidence, confidence,
  key, owner, access, purpose, retention,
  id, unique, replaces, group, notes) >

<!ELEMENT  subject  (#PCDATA) >
<!ELEMENT  auxiliary  (#PCDATA) >
<!ELEMENT  predicate  (#PCDATA) >
<!ELEMENT  range  (#PCDATA) >
<!ELEMENT  object  (#PCDATA) >

<!ELEMENT  start  (#PCDATA) >
<!ELEMENT  end  (#PCDATA) >
<!ELEMENT  durability  (#PCDATA) >
<!ELEMENT  location  (#PCDATA) >
<!ELEMENT  position  (#PCDATA) >

<!ELEMENT  source  (#PCDATA) >
<!ELEMENT  creator  (#PCDATA) >
<!ELEMENT  method  (#PCDATA) >
<!ELEMENT  evidence  (#PCDATA) >
<!ELEMENT  confidence  (#PCDATA)>

<!ELEMENT  key  (#PCDATA) >
<!ELEMENT  owner  (#PCDATA) >
<!ELEMENT  access  (#PCDATA) >
<!ELEMENT  purpose  (#PCDATA) >
<!ELEMENT  retention  (#PCDATA) >

<!ELEMENT  id  (#PCDATA) >
<!ELEMENT  unique  (#PCDATA) >
<!ELEMENT  replaces  (#PCDATA) >
<!ELEMENT  group  (#PCDATA) >
<!ELEMENT  notes  (#PCDATA) >
```

A.2.2 SITUATIONALSTATEMENT / DTD (Max)

```
<!ELEMENT  statement
  (mainpart, situation, explanation, privacy, administration) >

<!ELEMENT mainpart (subject, auxiliary, predicate, range, object) >
<!ELEMENT situation (start, end, durability, location, position) >
<!ELEMENT explanation (source, creator, method, evidence, confidence) >
<!ELEMENT privacy (key, owner, access, purpose, retention) >
```

```
<!ELEMENT administration (id, unique, replaces, group, notes) >

<!ELEMENT  subject  (#PCDATA) >
<!ELEMENT  auxiliary  (#PCDATA) >
<!ELEMENT  predicate  (#PCDATA) >
<!ELEMENT  range  (#PCDATA) >
<!ELEMENT  object  (#PCDATA) >

<!ELEMENT  start  (#PCDATA) >
<!ELEMENT  end  (#PCDATA) >
<!ELEMENT  durability  (#PCDATA) >
<!ELEMENT  location  (#PCDATA) >
<!ELEMENT  position  (#PCDATA) >

<!ELEMENT  source  (#PCDATA) >
<!ELEMENT  creator  (#PCDATA) >
<!ELEMENT  method  (#PCDATA) >
<!ELEMENT  evidence  (#PCDATA) >
<!ELEMENT  confidence  (#PCDATA)>

<!ELEMENT  key  (#PCDATA) >
<!ELEMENT  owner  (#PCDATA) >
<!ELEMENT  access  (#PCDATA) >
<!ELEMENT  purpose  (#PCDATA) >
<!ELEMENT  retention  (#PCDATA) >

<!ELEMENT  id  (#PCDATA) >
<!ELEMENT  unique  (#PCDATA) >
<!ELEMENT  replaces  (#PCDATA) >
<!ELEMENT  group  (#PCDATA) >
<!ELEMENT  notes  (#PCDATA) >
```

A.2.3 SITUATIONALSTATEMENT / DTD (Mix)

```
<!ELEMENT  statement
  (mainpart, situation, explanation, privacy, administration) >

<!ATTLIST mainpart subject CDATA >
<!ATTLIST mainpart auxiliary CDATA >
<!ATTLIST mainpart predicate CDATA >
<!ATTLIST mainpart range CDATA >
<!ATTLIST mainpart object CDATA >

<!ATTLIST situation start CDATA >
<!ATTLIST situation end CDATA >
<!ATTLIST situation durability CDATA >
<!ATTLIST situation location CDATA >
<!ATTLIST situation position CDATA >

<!ATTLIST explanation source CDATA >
<!ATTLIST explanation creator CDATA >
<!ATTLIST explanation method CDATA >
<!ATTLIST explanation evidence CDATA >
<!ATTLIST explanation confidence CDATA >
```

```
<!ATTLIST privacy key CDATA >
<!ATTLIST privacy owner CDATA >
<!ATTLIST privacy access CDATA >
<!ATTLIST privacy purpose CDATA >
<!ATTLIST privacy retention CDATA >

<!ATTLIST administration id CDATA >
<!ATTLIST administration unique CDATA >
<!ATTLIST administration replaces CDATA >
<!ATTLIST administration group CDATA >
<!ATTLIST administration notes CDATA >
```

A.2.4 SITUATIONALSTATEMENT / DTD (Min)

```
<!ELEMENT statement (#PCDATA) >
<!ATTLIST statement subject CDATA >
<!ATTLIST statement auxiliary CDATA >
<!ATTLIST statement predicate CDATA >
<!ATTLIST statement range CDATA >
<!ATTLIST statement object CDATA >

<!ATTLIST statement start CDATA >
<!ATTLIST statement end CDATA >
<!ATTLIST statement durability CDATA >
<!ATTLIST statement location CDATA >
<!ATTLIST statement position CDATA >

<!ATTLIST statement source CDATA >
<!ATTLIST statement creator CDATA >
<!ATTLIST statement method CDATA >
<!ATTLIST statement evidence CDATA >
<!ATTLIST statement confidence CDATA >

<!ATTLIST statement key CDATA >
<!ATTLIST statement owner CDATA >
<!ATTLIST statement access CDATA >
<!ATTLIST statement purpose CDATA >
<!ATTLIST statement retention CDATA >

<!ATTLIST statement id CDATA >
<!ATTLIST statement unique CDATA >
<!ATTLIST statement replaces CDATA >
<!ATTLIST statement group CDATA >
<!ATTLIST statement notes CDATA >
```

A.3 XML-Schemata for SITUATIONREPORTS and SITUATION-REQUESTS

A.3.1 SITUATIONREPORT / XSD (Tree)

Figure A.1 shows the XMLSchema tree for SITUATIONREPORTS, produced with XMLSpy.

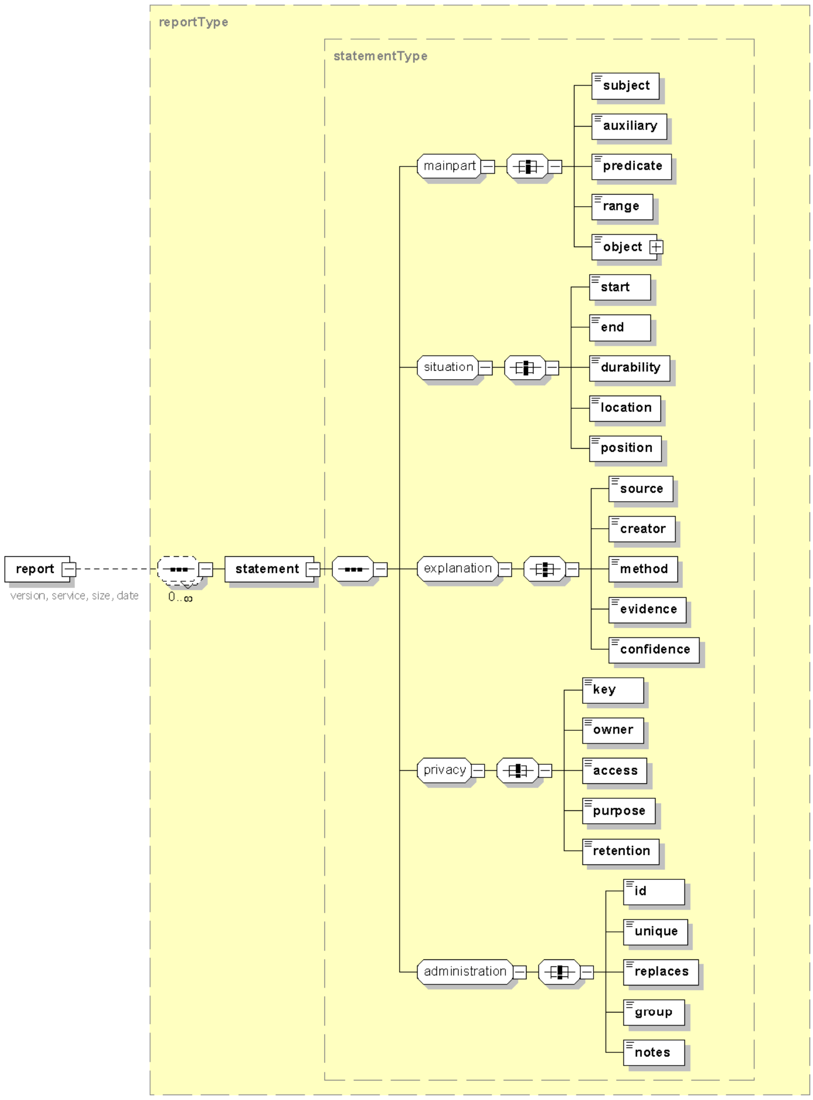


Figure A.1: SITUATIONREPORT *XML-Schema tree, produced with XMLSpy*

A.3.2 SITUATIONREPORT / XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Authors: Dominik Heckmann and Christian Blass-->
<!--Document Version: UserML 2.0 vom 30. April 2004 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="report">
        <xs:annotation>
            <xs:documentation>version, service, size, date</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="reportType">
                    <xs:attribute name="version" type="xs:anySimpleType"/>
                    <xs:attribute name="service" type="xs:anySimpleType"/>
                    <xs:attribute name="size" type="xs:nonNegativeInteger"/>
                    <xs:attribute name="date" type="xs:dateTime"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="reportType">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="statement" type="statementType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="statementType">
        <xs:sequence>
            <xs:group ref="mainpart"/>
            <xs:group ref="situation"/>
            <xs:group ref="explanation"/>
            <xs:group ref="privacy"/>
            <xs:group ref="administration"/>
        </xs:sequence>
    </xs:complexType>
    <xs:group name="mainpart">
        <xs:all>
            <xs:element name="subject" type="xs:anySimpleType"/>
            <xs:element name="auxiliary" type="xs:anySimpleType"/>
            <xs:element name="predicate" type="xs:anySimpleType"/>
            <xs:element name="range" type="xs:anySimpleType"/>
            <xs:element name="object" type="xs:anyType"/>
        </xs:all>
    </xs:group>
    <xs:group name="situation">
        <xs:all>
            <xs:element name="start" type="xs:dateTime"/>
            <xs:element name="end" type="xs:dateTime"/>
            <xs:element name="durability" type="xs:anySimpleType"/>
            <xs:element name="location" type="xs:anySimpleType"/>
            <xs:element name="position" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="explanation">
```

```
        <xs:all>
            <xs:element name="source" type="xs:anySimpleType"/>
            <xs:element name="creator" type="xs:anySimpleType"/>
            <xs:element name="method" type="xs:anySimpleType"/>
            <xs:element name="evidence" type="xs:anySimpleType"/>
            <xs:element name="confidence" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="privacy">
        <xs:all>
            <xs:element name="key" type="xs:string"/>
            <xs:element name="owner" type="xs:anySimpleType"/>
            <xs:element name="access" type="xs:anySimpleType"/>
            <xs:element name="purpose" type="xs:anySimpleType"/>
            <xs:element name="retention" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="administration">
        <xs:all>
            <xs:element name="id" type="xs:nonNegativeInteger"/>
            <xs:element name="unique" type="xs:string"/>
            <xs:element name="replaces" type="xs:string"/>
            <xs:element name="group" type="xs:anySimpleType"/>
            <xs:element name="notes" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
</xs:schema>
```

A.3.3 SITUATIONREQUEST / XSD (Tree)

Figure A.2 on page 215 shows the XMLSchema tree for SITUATIONREQUESTS, produced with XMLSpy.

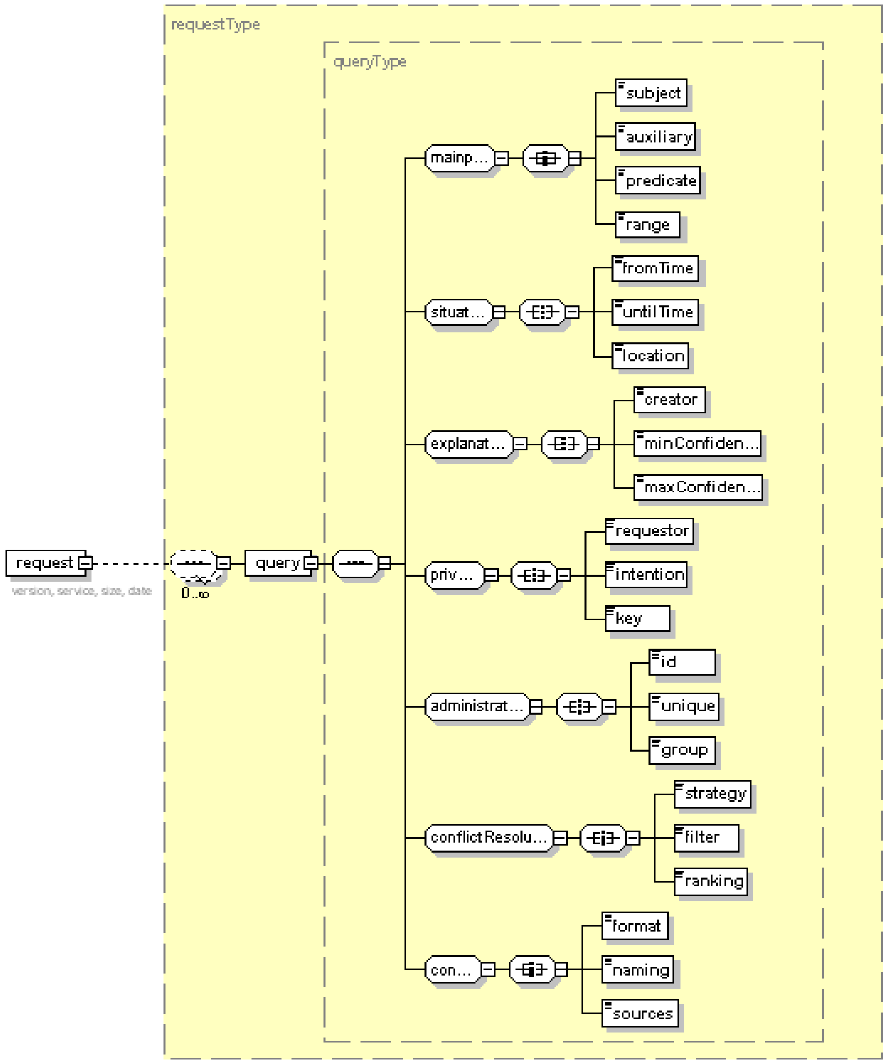


Figure A.2: SITUATIONREQUEST *XML-Schema tree, produced with XMLSpy*

A.3.4 SITUATIONREQUEST / XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Authors: Dominik Heckmann and Christian Blass-->
<!--Document Version: UserQL 2.0 vom 30. April 2004 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:annotation>
            <xs:documentation>version, service, size, date</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:complexContent>
                <xs:extension base="requestType">
                    <xs:attribute name="version" type="xs:anySimpleType"/>
                    <xs:attribute name="service" type="xs:anySimpleType"/>
                    <xs:attribute name="size" type="xs:nonNegativeInteger"/>
                    <xs:attribute name="date" type="xs:dateTime"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="requestType">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="query" type="queryType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="queryType">
        <xs:sequence>
            <xs:group ref="mainpart"/>
            <xs:group ref="situation"/>
            <xs:group ref="explanation"/>
            <xs:group ref="privacy"/>
            <xs:group ref="administration"/>
            <xs:group ref="conflictResolution"/>
            <xs:group ref="control"/>
        </xs:sequence>
    </xs:complexType>
    <xs:group name="mainpart">
        <xs:all>
            <xs:element name="subject" type="xs:anySimpleType"/>
            <xs:element name="auxiliary" type="xs:anySimpleType"/>
            <xs:element name="predicate" type="xs:anySimpleType"/>
            <xs:element name="range" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="situation">
        <xs:all>
            <xs:element name="fromTime" type="xs:dateTime"/>
            <xs:element name="untilTime" type="xs:dateTime"/>
            <xs:element name="location" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="explanation">
        <xs:all>
```

```
            <xs:element name="creator" type="xs:anySimpleType"/>
            <xs:element name="minConfidence" type="xs:anySimpleType"/>
            <xs:element name="maxConfidence" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="privacy">
        <xs:all>
            <xs:element name="requestor" type="xs:anySimpleType"/>
            <xs:element name="intention" type="xs:anySimpleType"/>
            <xs:element name="key" type="xs:string"/>
        </xs:all>
    </xs:group>
    <xs:group name="administration">
        <xs:all>
            <xs:element name="id" type="xs:nonNegativeInteger"/>
            <xs:element name="unique" type="xs:string"/>
            <xs:element name="group" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="control">
        <xs:all>
            <xs:element name="format" type="xs:anySimpleType"/>
            <xs:element name="naming" type="xs:anySimpleType"/>
            <xs:element name="sources" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
    <xs:group name="conflictResolution">
        <xs:all>
            <xs:element name="strategy" type="xs:anySimpleType"/>
            <xs:element name="filter" type="xs:anySimpleType"/>
            <xs:element name="ranking" type="xs:anySimpleType"/>
        </xs:all>
    </xs:group>
</xs:schema>
```

B APPENDIX - ONTOLOGY SPECIFICTIONS

B.1 The USERMODELONTOLOGY

Relevant concepts from the general user model ontology GUMO are described in this part of the appendix. They can also be accessed online via the UbisOntology Browser under the URL `http://www.u2m.org/UbisWorld/UbisBrowser.php` or in RDFS, DAML and OWL under
`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology_OWL`
`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology_DAML`
`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology_RDFS`.

B.1.1 List of User Model Auxiliaries

The list of user model auxiliaries together with their corresponding UbisIds can be found in table 5.1 on page 87.

B.1.2 List of User Model Property Groups

The basic user model dimensions are arranged in several property groups as described in figure 5.6 on page 88.

Table B.1: *List of user model classes with UbisIds*

Group	Class	Id
UserModelPropertyGroup	MentalState	700017
UserModelPropertyGroup	PhysicalState	700016
UserModelPropertyGroup	Demographics	700010
UserModelPropertyGroup	ContactInformation	700008
UserModelPropertyGroup	Role	700019
UserModelPropertyGroup	EmotionalState	700014
UserModelPropertyGroup	Personality	700012
UserModelPropertyGroup	Characteristics	700013
UserModelPropertyGroup	AbilityAndProficiency	700011

B.1.3 List of Basic User Model Dimensions

The list of basic user model dimensions is constantly under development. Please compare the online representation as stated above.

Table B.2: *List of basic user model dimensions with UbisIds*

Group	Name	Id
MentalState	TimePressure	800100
MentalState	CognitiveLoad	800102
MentalState	Depression	800104
MentalState	Irritation	800106
MentalState	Nervousness	800108
MentalState	Psychopathy	800110
MentalState	Trance	800112
MentalState	Trauma	800114
MentalState	Hypnosis	800118
PhysiologicalState	Heartbeat	800130
PhysiologicalState	BloodPressure	800131
PhysiologicalState	PupilsDilation	800132
PhysiologicalState	Respiration	800133
PhysiologicalState	Perspiration	800134
PhysiologicalState	Temperature	800135
PhysiologicalState	Injury	800136
PhysiologicalState	Fatigue	800137
PhysiologicalState	Arousal	800138
PhysiologicalState	Nourishment	800139
Location	SpatialLocation	800200
Location	VirtualLocation	800201
Demographics	Gender	800300
Demographics	AgeGroup	800301
Demographics	Age	800302
Demographics	Birthday	800303
Demographics	Birthplace	800304
Demographics	FirstLanguage	800306
Demographics	SecondLanguage	800308
Demographics	FamilyStatus	800312
Demographics	EducationLevel	800314
Demographics	Employment	800316
Demographics	Salary	800318
Demographics	Wealth	800320
ContactInformation	GivenName	800410
ContactInformation	MiddleName	800411
ContactInformation	FamilyName	800412
ContactInformation	FullName	800414
ContactInformation	Street	800416
ContactInformation	HouseNumber	800418
ContactInformation	PostalCode	800420
ContactInformation	City	800422

continued on next page

User Model Dimensions – continued from previous page		
Group	**Name**	**Id**
ContactInformation	State	800424
ContactInformation	Country	800426
ContactInformation	TelephoneNumber	800430
ContactInformation	MobilePhoneNumber	800432
ContactInformation	FaxNumber	800433
ContactInformation	Email	800434
ContactInformation	Homepage	800436
Role	Tourist	800460
Role	Businessman	800462
Role	Employee	800464
Role	Manager	800466
Role	Learner	800468
Role	Teacher	800470
Role	Child	800472
Role	Parent	800474
Role	Customer	800476
Role	Salesman	800478
Role	User	800480
Role	Developer	800482
Role	Author	800484
Role	Reader	800486
Role	Producer	800488
Role	Consumer	800490
EmotionalState	Happiness	800600
EmotionalState	Anxiety	800601
EmotionalState	Fear	800602
EmotionalState	Love	800604
EmotionalState	Hate	800606
EmotionalState	Pride	800608
EmotionalState	Shame	800610
EmotionalState	Anger	800612
EmotionalState	Disgust	800614
EmotionalState	Sadness	800616
EmotionalState	Satisfaction	800618
EmotionalState	Confusion	800620
EmotionalState	Worry	800622
EmotionalState	Boredom	800624
EmotionalState	Hope	800626
EmotionalState	Dread	800628
EmotionalState	Exitement	800630
EmotionalState	Relief	800632
EmotionalState	Joy	800634
Personality	Extravert	800702
Personality	Introvert	800704
Personality	Thinking	800706
Personality	Feeling	800708
Personality	Sensing	800710
Personality	Intuiting	800712
		continued on next page

User Model Dimensions – continued from previous page

Group	**Name**	**Id**
Personality	Judging	800714
Personality	Perceiving	800716
Personality	Controled	800718
Personality	Optimistic	800720
Personality	Pessimistic	800722
Personality	Tempered	800724
Personality	Neurotic	800726
Personality	Agreeable	800728
Personality	Open-minded	800730
Personality	Intelligent	800732
Personality	Excessiv	800734
Personality	Indulgent	800736
MyersBriggsTypeInventory	ExtravertVersusIntrovert	800802
MyersBriggsTypeInventory	SensorVersusIntuiter	800804
MyersBriggsTypeInventory	ThinkerVersusFeeler	800806
MyersBriggsTypeInventory	JudgerVersusPerceiver	800808
EysencksThreeFactorPENModel	Normality-Psychoticism	800812
EysencksThreeFactorPENModel	Introversion-Extraversion	800814
EysencksThreeFactorPENModel	Stability-Neuroticism	800816
FiveFactorModel	Extraversion-Energy-Enthusiasm	800822
FiveFactorModel	Agreeableness-Altruism-Affection	800824
FiveFactorModel	Conscientiousness-Control-Costraint	800826
FiveFactorModel	Neuroticism-NegativeAffectivity-Nervousness	800828
FiveFactorModel	Openness-Originality-OpenMindedness	800830
Characteristics	Talkative	800902
Characteristics	Assertive	800904
Characteristics	Dominant	800906
Characteristics	Quiet	800908
Characteristics	Reserved	800910
Characteristics	Shy	800912
Characteristics	Retiring	800914
Characteristics	Sympathetic	800916
Characteristics	Kind	800918
Characteristics	Warm	800920
Characteristics	Helpful	800922
Characteristics	Fault-finding	800924
Characteristics	Cold	800926
Characteristics	Unfriendly	800928
Characteristics	Organized	800930
Characteristics	Thorough	800932
Characteristics	Efficient	800934
Characteristics	Careless	800936
Characteristics	Disorderly	800938
Characteristics	Frivolous	800940
Characteristics	Tense	800942
Characteristics	Anxious	800944
Characteristics	Moody	800946
Characteristics	Worrying	800948

continued on next page

User Model Dimensions – continued from previous page		
Group	**Name**	**Id**
Characteristics	Stable	800950
Characteristics	Calm	800952
Characteristics	Contented	800954
Characteristics	Imaginative	800956
Characteristics	Artistic	800958
Characteristics	Inventive	800960
Characteristics	Commonplace	800962
Characteristics	Cooperative	800964
AbilityAndProficiency	AbilityToSee	801002
AbilityAndProficiency	AbilityToSmell	801004
AbilityAndProficiency	AbilityToHear	801006
AbilityAndProficiency	AbilityToTaste	801008
AbilityAndProficiency	AbilityToFeel	801010
AbilityAndProficiency	AbilityToTouch	801012
AbilityAndProficiency	AbilityToGrasp	801014
AbilityAndProficiency	AbilityToWalk	801016
AbilityAndProficiency	AbilityToUseStairs	801018
AbilityAndProficiency	AbilityToSwim	801020
AbilityAndProficiency	AbilityToCycle	801022
AbilityAndProficiency	AbilityToDrive	801024
AbilityAndProficiency	AbilityToTalk	801026
AbilityAndProficiency	ReadingSkills	801028
AbilityAndProficiency	WritingSkills	801030
AbilityAndProficiency	TypingSkills	801032
Motion	Walking	801070
Motion	Sitting	801072
Motion	Lying	801074
Motion	Standing	801076
Motion	GoingUpStairs	801078
Motion	GoingDownStairs	801080
end of table		

B.1.4 List of User Model Interest Categories

The user model interest category listing offers a large number of interest and preference categories like film genres, music trends, sports, pc-game genres, environmental topics and so on. It is based on the *Librarian‘s Index to the Internet*[1]. Even though it is incomplete and under development, it forms a good starting point for interest categories.

Table B.3*: List of user model interest categories with UbisIds*

Group	Name	ID
Category	Film	120020
Category	Music	120030
Category	Sports	120040
Category	PC-Games	120045
Category	Recreation	120050
Category	EnvironmentalTopics	120060
Category	Science	120070
Sports	Athletics	120104
Sports	AutomobileRacing	120106
Sports	Badminton	120108
Sports	Baseball	120110
Sports	Basketball	120112
Sports	Boating	120114
Sports	Boomerangs	120116
Sports	Bowling	120118
Sports	Boxing	120120
Sports	Chess	120122
Sports	Climbing	120124
Sports	Cricket	120126
Sports	Cycling	120130
Sports	Fencing	120132
Sports	FieldHockey	120134
Sports	FlyingDiscs	120136
Sports	Football	120138
Sports	Golf	120140
Sports	Gymnastics	120142
Sports	Hockey	120144
Sports	HorseRacing	120146
Sports	IceHockey	120148
Sports	IceSkating	120150
Sports	Karting	120152
Sports	Luge	120154
Sports	Motorcycles	120156
Sports	Olympics	120158
Sports	Rodeos	120160
Sports	Running	120162
Sports	Sailing	120164
Sports	ScubaDiving	120166

continued on next page

[1]Librarian‘s Index to the Internet: `www.lii.org`

Interest Categories – continued from previous page		
Group	**Name**	**Id**
Sports	Skateboarding	120168
Sports	Skydiving	120170
Sports	Skating	120172
Sports	Skiing	120174
Sports	Snowboarding	120178
Sports	Soccer	120182
Sports	Softball	120184
Sports	Surfing	120186
Sports	Swimming	120188
Sports	TableTennis	120190
Sports	Tennis	120192
Sports	Triathlon	120194
Sports	Volleyball	120196
Sports	Windsurfing	120200
Sports	WinterSports	120202
Sports	Wrestling	120204
Recreation	Aquariums	120220
Recreation	ArtGalleries	120221
Recreation	Backpacking	120222
Recreation	Beaches	120224
Recreation	Bicycling	120226
Recreation	BirdWatching	120228
Recreation	Books	120230
Recreation	Camping	120232
Recreation	CardTricks	120234
Recreation	Circus	120236
Recreation	Collecting	120238
Recreation	Crafts	120240
Recreation	CrosswordPuzzles	120242
Recreation	Dance	120243
Recreation	Fairs	120244
Recreation	Festivals	120246
Recreation	Fireworks	120248
Recreation	Fishing	120250
Recreation	Fortune-telling	120252
Recreation	Gambling	120254
Recreation	Games	120256
Recreation	Gardening	120258
Recreation	Hiking	120260
Recreation	Humor	120261
Recreation	Hunting	120262
Recreation	JigsawPuzzles	120264
Recreation	Juggling	120266
Recreation	Kites	120268
Recreation	Lotteries	120272
Recreation	MagicTricks	120274
Recreation	Movies	120276
Recreation	Museums	120278
		continued on next page

Interest Categories – continued from previous page		
Group	**Name**	**Id**
Recreation	Pets	120280
Recreation	Photography	120282
Recreation	Puppets	120284
Recreation	Rafting	120286
Recreation	Radio	120287
Recreation	RailroadModels	120288
Recreation	Television	120291
Recreation	Toys	120292
Recreation	Travel	120294
Recreation	WordGames	120296
Recreation	Zoos	120298
Music	Blues	120304
Music	Celtic	120306
Music	Children	120308
Music	ChoralMusic	120310
Music	Classical	120312
Music	Composers	120314
Music	Concerts	120316
Music	Country	120318
Music	Dance	120320
Music	Electronic	120322
Music	Folk	120324
Music	HeavyMetal	120326
Music	Hip-Hop	120328
Music	HumorAndComedy	120330
Music	Hymns	120332
Music	Improvisation	120334
Music	Indian	120336
Music	Jazz	120338
Music	Jewish	120340
Music	Karaoke	120341
Music	Lyrics	120342
Music	MusicalInstruments	120346
Music	Musicals	120348
Music	NewAge	120350
Music	Opera	120352
Music	Popular	120354
Music	Psychedelic	120356
Music	PunkRock	120358
Music	Quartets	120360
Music	Ragtime	120362
Music	Rap	120364
Music	RaveCulture	120366
Music	Reggae	120368
Music	Religion	120369
Music	Rhythm-n-Blues	120370
Music	Rock	120372
Music	SymphonyOrchestras	120374
		continued on next page

Interest Categories – continued from previous page		
Group	**Name**	**Id**
Music	WesternSwing	120376
Music	WorldMusic	120378
EnvironmentalTopics	AirPollution	120400
EnvironmentalTopics	BiologicalDiversity	120402
EnvironmentalTopics	ClimaticChangesAndGlobalWarming	120404
EnvironmentalTopics	CoralReefs	120406
EnvironmentalTopics	DesertsAndDesertification	120408
EnvironmentalTopics	Disasters	120410
EnvironmentalTopics	EndangeredSpecies	120412
EnvironmentalTopics	EnergyConservation	120414
EnvironmentalTopics	EnvironmentalEducation	120416
EnvironmentalTopics	EnvironmentalHealth	120418
EnvironmentalTopics	Fire	120420
EnvironmentalTopics	ForestsAndDeforestation	120422
EnvironmentalTopics	FuelCells	120424
EnvironmentalTopics	GovernmentalPolicy	120426
EnvironmentalTopics	Habitat	120428
EnvironmentalTopics	HazardousSubstances	120430
EnvironmentalTopics	HazardousWastes	120432
EnvironmentalTopics	IndustrialSafety	120434
EnvironmentalTopics	LightPollution	120436
EnvironmentalTopics	Littering	120438
EnvironmentalTopics	NaturalHistory	120442
EnvironmentalTopics	NoisePollution	120444
EnvironmentalTopics	NuclearEnergy	120446
EnvironmentalTopics	Pesticides	120448
EnvironmentalTopics	PestsAndDiseases	120450
EnvironmentalTopics	PetroleumIndustry	120452
EnvironmentalTopics	Ponds	120456
EnvironmentalTopics	Protection	120458
EnvironmentalTopics	RadioactiveWaste	120460
EnvironmentalTopics	RainForests	120462
EnvironmentalTopics	Recycling	120464
EnvironmentalTopics	RenewableEnergy	120466
EnvironmentalTopics	RuralDevelopment	120468
EnvironmentalTopics	SolarEnergy	120470
EnvironmentalTopics	Toxicology	120472
EnvironmentalTopics	Water	120474
EnvironmentalTopics	Weather	120476
EnvironmentalTopics	Wetlands	120478
EnvironmentalTopics	WildlifeConservation	120480
EnvironmentalTopics	WindPower	120482
Science	Agriculture	120500
Science	Animals	120504
Science	Anthropology	120506
Science	Archaeology	120508
Science	Astronomy	120510
Science	Biology	120514
		continued on next page

Interest Categories – continued from previous page

Group	**Name**	**Id**
Science	Botany	120516
Science	Chemistry	120518
Science	Cloning	120520
Science	ComputerScience	120522
Science	Environment	120526
Science	Evolution	120528
Science	Expeditions	120530
Science	Genetics	120532
Science	Geography	120534
Science	Geology	120536
Science	HumanGenome	120538
Science	Inventions	120540
Science	LifeSciences	120542
Science	Mathematics	120548
Science	Medicine	120550
Science	Microorganisms	120552
Science	Molecules	120554
Science	NatureSounds	120556
Science	NobelPrizes	120558
Science	NuclearEnergy	120560
Science	Ontologies	120562
Science	PhysicalSciences	120564
Science	Physics	120566
Science	Plants	120568
Science	Psychology	120570
Science	Scientists	120576
Science	SemanticWeb	120578
Science	Standardization	120580
Science	Taxonomy	120582
Science	Technology	120584
Science	UbiquitousComputing	120586
Science	UserModeling	120588
Science	Weather	120590
Science	Zoology	120592
Film	Action	120600
Film	Adventure	120602
Film	Animation	120604
Film	ChildrenAndFamily	120606
Film	Classics	120608
Film	Comedy	120610
Film	Crime	120612
Film	Documentary	120616
Film	Drama	120618
Film	Fantasy	120620
Film	Foreign	120622
Film	Horror	120624
Film	Independent	120626
Film	MartialArts	120628

continued on next page

Interest Categories – continued from previous page		
Group	**Name**	**Id**
Film	MusicAndConcert	120630
Film	Musicals	120632
Film	Mystery	120634
Film	Romance	120636
Film	ScienceFiction	120638
Film	Soaps	120640
Film	SportsAndExercise	120642
Film	Television	120644
Film	Thriller	120646
Film	War	120648
Film	Westerns	120650
Film	Other	120652
PC-Games	Adventure	120660
PC-Games	Action	120662
PC-Games	Arcade	120664
PC-Games	BoardGames	120666
PC-Games	Children	120668
PC-Games	Fighting	120670
PC-Games	Platform	120672
PC-Games	Puzzle	120674
PC-Games	Racing	120676
PC-Games	Shoot-Em-Up	120680
PC-Games	Simulation	120682
PC-Games	Sports	120684
PC-Games	Strategy	120686
end of table		

B.2 The UBISWORLD ONTOLOGY

This appendix adds some additional material to the so-called UBISWORLD ONTOLOGY which is instroduced in chapter 5.3. Figure B.1 recalls the six main parts of this ontology. This appendix contains a link to the UBISWORLD ONTOLOGY in RDF/RDFS and DAML+OIL. It presents the temporal elements as shown in the UbisOntologyBrowser, and lists parts of the ontology in InstanceOIL.

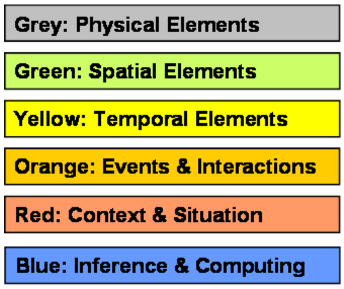


Figure B.1: *Six colors for the six main subparts of the* UBISONTOLOGY

B.2.1 Partial UBISWORLD ONTOLOGY in Instance OIL

Some parts of the UbisWorld ontology in InstanceOIL are presented in this subsection of the appendix to demonstrate how general the developed ontology tools are. OIL is a pre semantic web language that influenced the development of DAML+OIL.

```
Ontology-container
    title  UbisWorld Ontology in InstanceOIL
    creator  Dominik Heckmann
    subject  Ubiquitous Computing, User Modeling, Situational Statements
    description  A partial ontology for research in ubiquitous computing
             and user modeling
    description.release  0.20
    description.highlighting  green: slots, blue: classes, red: instances,
             bold: terminals
    type  ontology
    date  27.12.2004
    format  pseudo-xml
    identifier  http://www.u2m.org/UbisWorld.php?cmd=extern&window=InstanceOIL
    source  http://www.u2m.org/
    language  InstanceOIL
    language  en-uk

Ontology-definitions
    slot-def  is-nested-in  inverse  nests  properties  transitive
         domain  "Location"  range  "Location"
    slot-def  is-part-of  inverse  has-part  properties  transitive
```

```
     domain   "Physical Element"   range   "Physical Element"
slot-def   is-owned-by   inverse   owns   properties   transitive
     domain   "Physical Element"   range   "Human"
slot-def   is-property-of   inverse   has-property
     domain   "Situational Element"   range   "Physical Element"
class-def   "UbisWorld"
     slot-constraint   name   value-type string
     slot-constraint   id   value-type integer
     slot-constraint   category   value-type integer
class-def   "Physical and Category Elements"   subclass-of   "UbisWorld"
class-def   "Being"   subclass-of   "Physical and Category Elements"
class-def   "Thing"   subclass-of   "Physical and Category Elements"
class-def   "Device"   subclass-of   "Thing"
class-def   "Furniture"   subclass-of   "Thing"
class-def   "Vehicle"   subclass-of   "Thing"
class-def   "Other Object"   subclass-of   "Thing"
class-def   "Keyboard"   subclass-of   "Device"
class-def   "Display"   subclass-of   "Device"
class-def   "Mouse"   subclass-of   "Device"
class-def   "Speaker"   subclass-of   "Device" "Furniture"
class-def   "Microphone"   subclass-of   "Device"
class-def   "Camera"   subclass-of   "Device"
class-def   "Projector"   subclass-of   "Device"
class-def   "Sensor"   subclass-of   "Device"
class-def   "Network"   subclass-of   "Device"
class-def   "IR Bark"   subclass-of   "Device"
class-def   "ID TAG"   subclass-of   "Device"
class-def   "Reading"   subclass-of   "Other Object"
class-def   "Food"   subclass-of   "Other Object"
class-def   "Book"   subclass-of   "Reading"
class-def   "Newspaper"   subclass-of   "Reading"
class-def   "Magazine"   subclass-of   "Reading"
class-def   "Webpage"   subclass-of   "Reading"
class-def   "Potatos"   subclass-of   "Food"
class-def   "Rice"   subclass-of   "Food"
class-def   "Pasta"   subclass-of   "Food"
class-def   "Bread"   subclass-of   "Food"
class-def   "Milk"   subclass-of   "Food"
class-def   "Coke"   subclass-of   "Food"
class-def   "Flowers"   subclass-of   "Other Object"
class-def   "Chocolate"   subclass-of   "Food"
class-def   "System"   subclass-of   "Physical and Category Elements"
class-def   "Handheld"   subclass-of   "Device"
class-def   "Notebook"   subclass-of   "Device"
class-def   "Desktop"   subclass-of   "Device"
class-def   "Server"   subclass-of   "Device"
class-def   "Mobile Phone"   subclass-of   "Device"
class-def   "Airplane"   subclass-of   "Vehicle"
class-def   "Train"   subclass-of   "Vehicle"
class-def   "Bus"   subclass-of   "Vehicle"
class-def   "Car"   subclass-of   "Vehicle"
class-def   "Bike"   subclass-of   "Vehicle"
class-def   "Person"   subclass-of   "Being"
class-def   "Pet"   subclass-of   "Being"
class-def   "Dog"   subclass-of   "Pet"
```

```
class-def   "Cat"   subclass-of   "Pet"
class-def   "Almost Human"   subclass-of   "Being"
class-def   "Mouse"   subclass-of   "Pet"
class-def   "Category"   subclass-of   "Physical and Category Elements"
class-def   "Film"   subclass-of   "Category"
class-def   "Music"   subclass-of   "Category"
class-def   "Sports"   subclass-of   "Category"
class-def   "PC-Games"   subclass-of   "Category"
class-def   "Recreation"   subclass-of   "Category"
class-def   "Environmental Topics"   subclass-of   "Category"
class-def   "Science"   subclass-of   "Category"
class-def   "Activity and Inference Elements"  subclass-of  "UbisWorld"
....
instance-of   "Boris"  "Person"
instance-of   "Dominik"  "Person"
instance-of   "Doris"  "Person"
instance-of   "Joerg"  "Person"
instance-of   "Thorsten"  "Person"
instance-of   "Frank"  "Person"
instance-of   "Michael Kruppa"  "Person"
instance-of   "Mira"  "Person"
instance-of   "Christoph Endres"  "Person"
instance-of   "Toni"  "Person"
instance-of   "Andreas"  "Person"
instance-of   "Rainer"  "Person"
instance-of   "Fluffy"  "Dog"
instance-of   "Minka"  "Cat"
instance-of   "Toxy"  "Dog"
instance-of   "Aibo"  "System" "Dog"
instance-of   "Pussy"  "Cat"
instance-of   "Fangmaus"  "Cat"
instance-of   "Kralline"  "Cat"
instance-of   "Jerry"  "Mouse"
instance-of   "Tom"  "Cat"
class-def   "Continent"   subclass-of   "Location"
class-def   "Country"   subclass-of   "Location"
class-def   "Region"   subclass-of   "Location"
class-def   "City"   subclass-of   "Location"
class-def   "Quarter"   subclass-of   "Location"
class-def   "Street"   subclass-of   "Location"
class-def   "Street-Segment"   subclass-of   "Location"
class-def   "Building"   subclass-of   "Location"
class-def   "Floor"   subclass-of   "Location"
class-def   "Section"   subclass-of   "Location"
class-def   "Room"   subclass-of   "Location"
class-def   "Room-Segment"   subclass-of   "Location"
...
instance-of   "Europe"  "Continent"
instance-of   "Berlin"  "City"
instance-of   "Saarland"  "Region"
instance-of   "MidLothian"  "Region"
instance-of   "Saarland University"  "Quarter"
instance-of   "Hessen"  "Region"
instance-of   "Frankfurt"  "City"
instance-of   "Muenchen"  "City"
```

```
instance-of   "Saarbruecken"  "City"
instance-of   "Edinburgh"  "City"
instance-of   "Paris"  "City"
instance-of   "Nauwieser Viertel"  "Quarter"
instance-of   "Georg Square"  "Quarter"
instance-of   "Airport Frankfurt"  "Quarter"
instance-of   "Airport Edinburgh"  "Quarter"
instance-of   "Voelklinger Huette"  "Quarter"
...
instance-of   "User Model Auxiliary"  "User Model and Context Elements"
instance-of   "hasProperty"  "User Model Auxiliary"
instance-of   "hasInterest"  "User Model Auxiliary"
instance-of   "hasKnowledgeOrBelief"  "User Model Auxiliary"
instance-of   "hasPlanOrGoal"  "User Model Auxiliary"
instance-of   "hasPreference"  "User Model Auxiliary"
instance-of   "hasRegularity"  "User Model Auxiliary"
instance-of   "performsActivity"  "User Model Auxiliary"
instance-of   "hasDone"  "User Model Auxiliary"
instance-of   "hasLocation"  "User Model Auxiliary"
instance-of   "yes no"  "ElementSets "
instance-of   "male female"  "ElementSets "
instance-of   "low medium high"  "ElementSets "
instance-of   "PoorGoodPerfect"  "ElementSets "
instance-of   "from 1 to 6"  "ElementSets "
instance-of   "Privacy Access"  "ElementSets "
instance-of   "Privacy Purpose"  "ElementSets "
instance-of   "Privacy Retention"  "ElementSets "
instance-of   "frequency"  "ElementSets "
instance-of   "noise level"  "ElementSets "
instance-of   "walking speed"  "ElementSets "
instance-of   "shopping list"  "ElementSets "
class-def   "User Model and Context Elements"   subclass-of   "UbisWorld"
class-def   "Basic User Dimensions" subclass-of "User Model and Context Elements"
class-def   "Domain Dependent Data" subclass-of "User Model and Context Elements"
class-def   "Contact Information"   subclass-of   "Basic User Dimensions"
class-def   "Demographics"   subclass-of   "Basic User Dimensions"
class-def   "Ability And Proficiency"   subclass-of   "Basic User Dimensions"
class-def   "Personality"   subclass-of   "Basic User Dimensions"
class-def   "Characteristics"   subclass-of   "Basic User Dimensions"
class-def   "Emotional State"   subclass-of   "Basic User Dimensions"
...
related   is-nested-in   "Germany"  "Europe"
related   is-nested-in   "France"  "Europe"
related   is-nested-in   "Scotland"  "Europe"
related   is-nested-in   "Europe"  "any Place"
related   is-nested-in   "Berlin"  "Germany"
related   is-nested-in   "Saarland"  "Germany"
related   is-nested-in   "MidLothian"  "Scotland"
related   is-nested-in   "Saarland University"  "Saarbruecken"
related   is-nested-in   "Hessen"  "Germany"
related   is-nested-in   "Frankfurt"  "Hessen"
related   is-nested-in   "Muenchen"  "Bayern"
related   is-nested-in   "Saarbruecken"  "Saarland"
related   is-nested-in   "Edinburgh"  "Scotland"
related   is-nested-in   "Paris"  "France"
```

```
related   is-nested-in   "Nauwieser Viertel"  "Saarbruecken"
related   is-nested-in   "Georg Square"  "Edinburgh"
related   is-nested-in   "Airport Frankfurt"  "Frankfurt"
related   is-nested-in   "Airport Edinburgh"  "Edinburgh"
related   is-nested-in   "Voelklinger Huette"  "Saarland"
related   is-nested-in   "Goldene Bremm"  "France"
related   is-nested-in   "Goldene Bremm"  "Saarbruecken"
related   is-nested-in   "First Floor"  "Building 36"
related   is-nested-in   "Second Floor"  "Building 36"
related   is-nested-in   "Building 36"  "Saarland University"
related   is-nested-in   "Building 45"  "Saarland University"
related   is-nested-in   "Mensa Saarbruecken"  "Saarland University"
related   is-nested-in   "WW-Floor"  "Building 36"
related   is-nested-in   "Mensa Second Floor"  "Mensa Saarbruecken"
related   is-nested-in   "B36 First Floor"  "Building 36"
related   is-nested-in   "Room 125"  "WW-Floor"
related   is-nested-in   "Room 124"  "WW-Floor"
related   is-nested-in   "Room 123"  "WW-Floor"
related   is-nested-in   "Room 122"  "WW-Floor"
related   is-nested-in   "Seminar Room 121"  "WW-Floor"
related   is-nested-in   "Hallway"  "WW-Floor"
related   is-nested-in   "Dining Hall"  "Mensa Second Floor"
related   is-nested-in   "Airport Paris"  "Paris"
related   is-nested-in   "Airport Berlin"  "Berlin"
related   is-nested-in   "HS 1"  "Building 45"
related   is-nested-in   "HS 2"  "Building 45"
related   is-nested-in   "HS 3"  "Building 45"
related   is-nested-in   "Schlemmer Eule"  "First Floor Mensa"
related   is-nested-in   "First Floor Mensa"  "Mensa Saarbruecken"
related   is-nested-in   "Gasgeblaesehalle"  "Voelklinger Huette"
related   is-nested-in   "Bayern"  "Germany"
related   is-nested-in   "Buccleuch Place"  "Edinburgh"
related   is-nested-in   "Buccleuch 1"  "Buccleuch Place"
related   is-nested-in   "Buccleuch 2"  "Buccleuch Place"
related   is-nested-in   "Buccleuch 4"  "Buccleuch Place"
related   is-nested-in   "Highlands"  "Scotland"
related   is-nested-in   "Borders"  "Scotland"
...
```

B.2.2 UBISWORLD ONTOLOGY in RDF/RDFS and DAML+OIL

```
<?xml version="1.0" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:daml="http://www.daml.org/2000/10/daml-ont#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/" xmlns:user="http://u2m.org/2003/02/UserModelOntology.rdf"
  xmlns:u2m="http://u2m.org/2003/U2M/" xmlns:lexicon="http://u2m.org/2003/Lexicon/"
  xml:base="http://u2m.org/2003/02/UserModelOntology.rdf">
- <rdf:Description rdf:about="">
    <dc:title>UbisWorldOntology in RDFS</dc:title>
    <dc:creator>Dominik Heckmann</dc:creator>
    <dc:date>27.12.2004</dc:date>
    <dc:description>The UserModelAndContextOntology defines concepts, classes, predicates and instances about the
      situation of the user, the system and the environment.</dc:description>
    <rdfs:comment>This ontology is based on tutorials from Anthony Jameson and Jon Oberlander</rdfs:comment>
  </rdf:Description>
- <rdfs:Class rdf:ID="UbisWorld.100000">
    <rdfs:label>UbisWorld</rdfs:label>
    <u2m:identifier>100000</u2m:identifier>
    <u2m:website rdf:resource="http://u2m.org/UbisWorld/show.php?subject=100000" />
    <rdfs:subClassOf rdf:resource="#UbisWorld.100000" />
  </rdfs:Class>
- <rdfs:Class rdf:ID="PhysicalAndCategoryElements.100001">
    <rdfs:label>Physical and Category Elements</rdfs:label>
    <u2m:identifier>100001</u2m:identifier>
    <u2m:website rdf:resource="http://u2m.org/UbisWorld/show.php?subject=100001" />
    <rdfs:subClassOf rdf:resource="#UbisWorld.100000" />
  </rdfs:Class>
- <rdfs:Class rdf:ID="Being.100002">
    <rdfs:label>Being</rdfs:label>
    <u2m:identifier>100002</u2m:identifier>
    <u2m:website rdf:resource="http://u2m.org/UbisWorld/show.php?subject=100002" />
    <rdfs:subClassOf rdf:resource="#PhysicalAndCategoryElements.100001" />
```

***Figure B.2**: Heading of the UbisOntology in RDF/RDFS in the IE XML Viewer*

The full ontologies, that are under online community development, can be found at:
`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UbisWorldOntology_RDFS`
`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UbisWorldOntology_DAML`

B.2.3 Temporal Elements shown in UbisOntologyBrowser

The temporal ontology is part of the UbisWorld ontology and is described in chapter 5.3.3 on page 109. This appendix shows the tree view in the UbisOntologyBrowser, where classes are represented as folder symbols and relations are represented as circled sheets.

***Figure B.3**: Some temporal classes and relations as shown in the UbisOntologyBrowser*

B.2.4 Media Types as shown in UbisOntologyBrowser

The classification of media types and media categories[2] for the research in ubiquitous computing, as presented in figure B.4, has been developed with the UbisOntologyBrowser, and is used within a multimedia presentation planner.

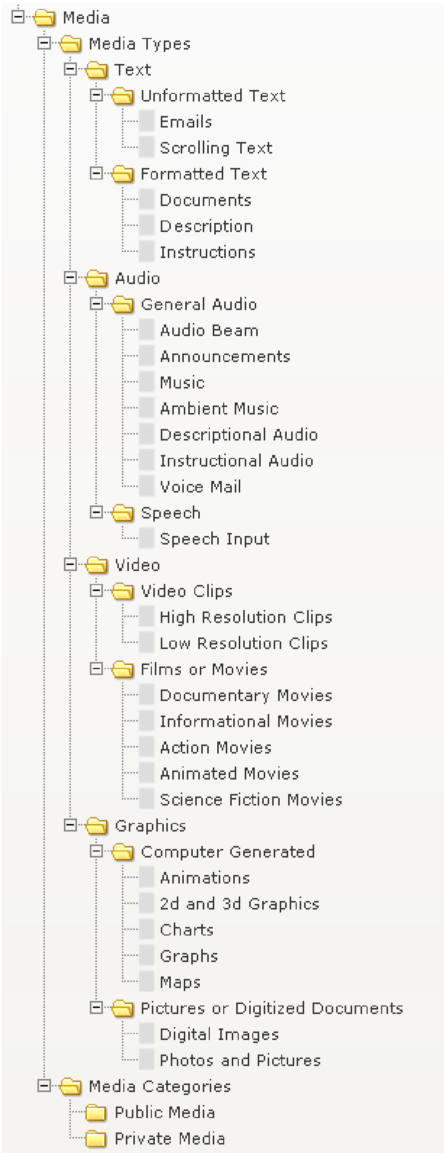


***Figure B.4**: Media types and media categories, modeled with the UbisOntologyBrowser*

[2]This media taxonomy has mostly been implemented in a student project by Usman Rafiq.

C List of Acronyms

ACL	Agent Communication Language
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
AR	Augmented Reality
ASCII	American Standard Code for Information Interchange
CORBA	Common Object Request Broker Architecture
CC/PP	Composit Capabilities / Preference Profile
CPEX	Customer Profile Exchange
CRM	Customer Relationship Management
CYK	Cycorp Knowledge Base
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DASUM	Decentralized, Agent Based and Social Approaches to User Modeling
DBMS	Database Management System
DOM	Document Object Model
DTD	Document Type Definition
FIPA	Foundation for Intelligent Physical Agents
GPS	Global Positioning System
GUMO	General User Model Ontology
HCI	Human Computer Interaction
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
HumanML	Human Markup Language
IBM	International Business Machines Inc.
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
iHCI	implicit Human Computer Interaction
IUI	Intelligent User Interfaces
KAON	Karlsruhe Ontology Tool Suite
KIF	Knowledge Interchange Format
KQML	Knowledge Query Mark-up Language
LAN	Local Area Network

LDAP	Lightweight Directory Access Protocol
MILO	Mid-Level Ontology
MIT	Massachusetts Institute of Technology
OASIS	Organization for Advancement of Structured Information Standards
ODBC	Open Database Connectivity
OKBC	Open Knowledge Base Connectivity
OIL	Ontology Inference Layer / Ontology Interchange Language
OWL	Ontology Web Language
OWL-S	Ontology Web Language for Services
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
PJ	Personal Journal
PDA	Personal Digital Assistant
QEL	Query Exchange Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RDQL	Resource Description Query Language
RQL	RDF Query Language
SALSA	The Saarbrücken Lexical Semantics Acquisition Project
SAX	Simple API for XML
SituationML	Situation Markup Language
SituationQL	Situation Query Language
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SUMO	Suggested Upper Merged Ontology
U2M	Ubiquitous User Modeling
Ubid	UbisIdentifier
Ubex	UbisExpression
Ubli	UbisList
UC	Ubiquitous Computing
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UserML	User Model Markup Language
UserOL	User Model Ontology Language
UserQL	User Model Query Language
VR	Virtual Reality
W3C	World Wide Web Consortium
WSDF	Web Service Description Framework
WSMF	Web Service Modeling Framework
WWW	World Wide Web
XHTML	Extensible Hypertext Mark-up Language
XML	Extensible Mark-up Language
XPath	XML Path Langage
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

List of Figures

List of Tables

Index

Bibliography

[Aarts et al., 2003] Aarts, E., Collier, R., Loenen, E. v., and Ruyter, B. d., editors (2003). *Ambient Intelligence*, volume 2875 of *Lecture Notes in Computer Science*. Berlin, Heidelberg, New York: Springer.

[Aarts and Marzano, 2003] Aarts, E. and Marzano, S., editors (2003). *The New Everyday: Vision on Ambient Intelligence*. 010 Publishers, Rotterdam, The Netherlands.

[Acton, 1997] Acton, G. S. (1997). Great ideas in personality. online book, http://www.personalityresearch.org. Rochester Institute of Technology, Department of Psychology.

[Allgayer et al., 1989] Allgayer, J., Harbusch, K., Kobsa, A., Reithinger, N., and Schmauks, D. (1989). XTRA: A natural-language access system to expert systems. *International Journal of Man-Machine Studies*, 31:161–195.

[André et al., 2000] André, E., Klesen, M., Gebhard, P., Allen, S., and Rist, T. (2000). Exploiting models of personality and emotions to control the behavior of animated interactive agents. In *Proceedings of the workshop on "Achieving Human-Like Behavior in Interactive Animated Agents"*, pages 3–7.

[Aslan and Krüger, 2004] Aslan, I. and Krüger, A. (2004). The bum bag navigator (bbn): An advanced pedestrian navigation system. In *Workshop on Artificial Intelligence in Mobile Systems (AIMS) in conjunction with UbiComp 2004*, pages 15–19, Nottingham, England.

[Azuma, 1997] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355 – 385.

[Barwise, 1981] Barwise, J. (1981). Some computational aspects of situation semantics. In *Proceedings of the 19th annual meeting on Association for Computational Linguistics*, pages 109–111, Morristown, NJ, USA.

[Barwise and Perry, 1983] Barwise, J. and Perry, J. (1983). *Situations and Attitudes*. Bradford Books, MIT Press, Cambridge, Massachusetts.

[Baus et al., 2005] Baus, J., Cheverst, K., and Kray, C. (2005). *Map-based mobile services - Theories, Methods and Implementations*, chapter A Survey of Map-based Mobile Guides, pages 197–216. Springer.

[Baus et al., 2002] Baus, J., Krüger, A., and Wahlster, W. (2002). A resource-adaptive mobile navigation system. In *Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI2002*, pages 15–22. New York: ACM Press.

[Berkovsky, 2005] Berkovsky, S. (2005). Ubiquitous user modeling in recommender systems. In *Proceedings of the 10th International Conference on User Modeling*, pages 496–498, Edinburgh, Scotland. LNAI 3538: Springer, Berlin Heidelberg.

[Berners-Lee et al., 1998] Berners-Lee, T., Fielding, R., and L.Masinter (1998). *RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax*. IETF.

[Berners-Lee and Fischetti, 1999] Berners-Lee, T. and Fischetti, M. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco, New York: HarperCollins.

[Blass, 2004] Blass, C. (2004). Conflict resolution strategies for ubiquitous user modeling. Student report, Department of Computer Science, Saarland University, Saarbrücken, Germany.

[Blocher, 1999] Blocher, A. (1999). *Ressourcenadaptierende Raumbeschreibung: Ein beschränkt-optimaler Lokalisationsagent*. PhD thesis, Departement of Computer Science, Saarland University, Saarbrücken, Germany.

[Bohnenberger, 2004] Bohnenberger, T. (2004). *Decition-Theoretic Planning for User-Adaptive Systems: Dealing with Multiple Goals and Resource Limitations*, volume 289 of *DISKI*. Aka Verlag, Berlin. ISBN 3-89838-289-3.

[Bohnenberger et al., 2002] Bohnenberger, T., Brandherm, B., Großmann-Hutter, B., Heckmann, D., and Wittig, F. (2002). Empirically grounded decision-theoretic adaptation to situation-dependent resource limitations. *Künstliche Intelligenz*, 16(3):10–16.

[Bohrer and Holland, 2000] Bohrer, K. and Holland, B. (2000). *Customer Profile Exchange (CPExchange) Specification*. http://www.cpexchange.org/standard/cpexchangev1_0F.pdf.

[Brandherm and Schmitz, 2004] Brandherm, B. and Schmitz, M. (2004). Presentation of a modular framework for interpretation of sensor data with dynamic Bayesian networks on mobile devices. In *LWA 2004, Lernen Wissensentdeckung Adaptivität*, pages 9–10, Humboldt-Universität zu Berlin, Germany.

[Brandherm and Schwartz, 2005] Brandherm, B. and Schwartz, T. (2005). Geo referenced dynamic Bayesian networks for user positioning on mobile systems. In Strang, T. and Linnhoff-Popien, C., editors, *Proceedings of the International Workshop on Location- and Context-Awareness (LoCA), LNCS 3479*, pages 223–234, Munich, Germany. Berlin, Heidelberg: Springer.

[Brickley and Guha, 1999] Brickley, D. and Guha, R. (1999). *Resource Description Framework (RDF) Schema Specification*. W3C. W3C proposed recommendation.

[Brooks, 2002] Brooks, R. (2002). Human markup language primary base specification 1.0. Technical report, OASIS Committee Specification 01. online at oasis-open.org, Working Draft.

[Brooks and Cagle, 2002] Brooks, R. and Cagle, K. (2002). The web services component model and HumanML - building personalization into the emerging web component architecture. OASIS, Technical Report, online at oasis-open.org.

[Brownston et al., 1985] Brownston, L., Farrell, R., Kant, E., and Martin, N. (1985). *Programming Expert Systems in OPS5: An Introduction to Rule-based Programming*. Addison-Wesley, Reading, Massachusetts.

[Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129.

[Brusilovsky et al., 1998] Brusilovsky, P., Vassileva, J., and Kobsa, A., editors (1998). *Adaptive Hypertext and Hypermedia*. Dordrecht: Kluwer Academic Publishers.

[Candela, 2003] Candela, P. (2003). W3C Glossary and Dictionary. http://www.w3.org/2003/glossary/.

[Carolis et al., 2003] Carolis, B. D., Pizzutilo, S., and Palmisano, I. (2003). D-Me: Personal interaction in smart environments. In Brusilovsky, P., Corbett, A., and de Rosis, F., editors, *Proceedings of User Modeling '03*, pages 388–392, Johnstown, PA, USA. LNAI 2702: Springer, Berlin Heidelberg.

[Castel, 2002] Castel, F. (2002). Ontological computing. *Communications of the ACM*, 45(2):29–30.

[Champin, 2001] Champin, P.-A. (2001). *RDF tutorial for developers*. Univ. Lyon 1, France. http://www710.univ-lyon1.fr/ champin/rdf-tutorial/.

[Chasaide and Gobl, 2001] Chasaide, A. N. and Gobl, C. (2001). Voice quality and the synthesis of affect. In Keller, E., Bailly, G., Monaghan, A., Terken, J., and Huckvale, M., editors, *Improvements in Speech Synthesis*. Wiley, UK.

[Chen and Kotz, 2002] Chen, G. and Kotz, D. (2002). Solar: A pervasive-computing infrastructure for context-aware mobile applications. Technical Report TR2002-421, Dartmouth College, Computer Science, Hanover, NH.

[Chen et al., 2003] Chen, H., Finin, T., and Joshi, A. (2003). Semantic web in a pervasive context-aware architecture. In Krüger, A. and Malaka, R., editors, *Proceedings of Artificial Intelligence in Mobile Systems*, pages 33–40, Seattle, USA.

[Chepegin et al., 2004a] Chepegin, V., Aroyo, L., Bra, P. D., and Heckmann, D. (2004a). User modeling for modular adaptive hypermedia. In *Workshop on Applications of Semantic Web Technologies for Educational Adaptive Hypermedia in conjunction with AH'04*, pages 366–371, Eindhoven, The Netherlands.

[Chepegin et al., 2004b] Chepegin, V., Heckmann, D., and Aroyo, L. (2004b). User modeling in ubiquitous environments - minutes for AWIS meeting. Technical report, Eindhoven University of Technology & Saarland University.

[Coen et al., 1999] Coen, M., Philips, B., Warshawsky, N., Weisman, L., Peters, S., and Finin, P. (1999). Meeting the computational needs for intelligent environments: The metaglue system. In Nixon, P., Lacey, G., and Dobson, S., editors, *Proceedings of the First International Workshop on Managing Interactions in Smart Environments*, pages 201–212. Springer London.

[Compiol, 2005] Compiol, R. (2005). Pervasive computing: Aspectos de modelagem e simulao. Master thesis, Department of Computer Science, Federal University of Santa Catarina, Brazil.

[Cycorp, 2004] Cycorp (2004). What‘s in Cyc? http://www.cyc.com.

[Dempster, 1968] Dempster, A. (1968). A generalization of bayesian inference. *Journal of the Royal Statistical Society, Series B*, 30:205–247.

[Dervin, 1999] Dervin, B. (1999). Chaos, order, and sense-making: A proposed theory for information design. In Jacobson, R., editor, *Information Desing*, chapter 3, pages 35–57. MIT Press, Cambridge Massachussetts, First MIT Press paperback edition 2000 edition.

[Dewaele and Furnham, 2000] Dewaele, J.-M. and Furnham, A. (2000). Personality and speech production: a pilot study of second language learners. *Personality and Individual Differences*, 28:355–365.

[Dey and Abowd, 1999] Dey, A. K. and Abowd, G. D. (1999). Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, U.S.A.

[Dürr and Rothermel, 2003] Dürr, F. and Rothermel, K. (2003). On a location model for fine-grained geocast. In Dey, A. K., Schmidt, A., and McCarthy, J. F., editors, *Proceedings of the Fifth International Conference on Ubiquitous Computing*, pages 18–35. LNCS 2864.

[Eberhart, 2003] Eberhart, A. (2003). *Ontology-based Infrastructure for Intelligent Applications*. PhD thesis, University of Saarbrücken, Germany.

[Ekman, 1993] Ekman, P. (1993). Facial expression of emotion. *Americal Psychologist*, (48):384–392.

[Erk et al., 2003a] Erk, K., Kowalski, A., and Pado, S. (2003a). The SALSA annotation tool. In Duchier, D., editor, *Proceedings of the Workshop on Prospects and Advances in the Syntax/Semantics Interface*, pages 1–4, Nancy, France.

[Erk et al., 2003b] Erk, K., Kowalski, A., Pado, S., and Pinkal, M. (2003b). Building a resource for lexical semantics. In *Proceedings of the Workshop on Frame Semantics, XVII. International Congress of Linguists (CIL)*, pages CD–Rom, Prague, Czech Republic.

[Eskenazi and Black, 2001] Eskenazi, M. and Black, A. (2001). A study on speech over the telephone and aging. In *EUROSPEECH-2001*, pages 171–174.

[Evans, 1981] Evans, D. A. (1981). A situation semantics approach to the analysis of speech acts. In *Proceedings of the 19th annual meeting on Association for Computational Linguistics*, pages 113–116, Morristown, NJ, USA.

[Eysenck and Eysenck, 1991] Eysenck, H. and Eysenck, S. (1991). The eysenck personality questionnaire-revised. Hodder and Stoughton, Sevenoaks.

[Fensel, 2001] Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Berlin Heidelberg.

[Fensel et al., 2003] Fensel, D., Liebermann, J., and Wahlster, W., editors (2003). *Spinning the Semantic Web. Bringing the World Wide Web to Its Full Potential.* MIT Press, Boston.

[Fillmore, 1985] Fillmore, C. J. (1985). Frames and the Semantics of Understanding. *Quaderni di semantica*, VI(2):222–255.

[Fink, 2004] Fink, J. (2004). *User Modeling Servers - Requirements, Design, and Evaluation*, volume 289 of *DISKI*. Aka Verlag, Berlin. ISBN 3-89838-277-X.

[Fink and Kobsa, 2000] Fink, J. and Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Modeling and User-Adapted Interaction*, 10(2-3):209–249.

[Fink and Kobsa, 2002] Fink, J. and Kobsa, A. (2002). User modeling for personalized city tours. *Artificial Intelligence Review*, 18(1):33–74.

[Foster et al., 2001] Foster, I., Kesselman, C., and Tuecke, S. (2001). The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:1–24.

[Gebhard et al., 2004] Gebhard, P., Klesen, M., and Rist, T. (2004). Coloring multi-character conversations through the expression of emotions. In *Proceedings of the Tutorial and Research Workshop on Affective Dialogue Systems (ADS'04)*, pages 128–141, Kloster Irsee.

[Gellersen et al., 1999] Gellersen, H.-W., Beigl, M., and Krull, H. (1999). The MediaCup: Awareness technology embedded in an everyday object. *Lecture Notes in Computer Science*, 1707:308–310.

[Genesereth, 1991] Genesereth, M. R. (1991). Knowledge interchange format. In Allen, J., Fikes, R., and Sandewall, E., editors, *Proceedings of the 2nd International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, pages 238–249. Morgan Kaufman Publishers Inc.

[Gil and Oberlander, 2002] Gil, A. and Oberlander, J. (2002). Taking care of the linguistic features of extraversion. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, pages 363–368.

[Goleman, 1995] Goleman, D. (1995). *Emotional Intelligence*. Bantam Books: New York.

[Grice and Baumann, 2002] Grice, M. and Baumann, S. (2002). Deutsche intonation und gtobi. In *Linguistische Berichte*, pages 267–298.

[Heckmann, 2001] Heckmann, D. (2001). Ubiquitous user modeling for situated interaction. In *Proceedings of the 8th International Conference on User Modeling*, pages 280–282. LNCS 2109: Springer, Berlin Heidelberg.

[Heckmann, 2002a] Heckmann, D. (2002a). Proposal for a user modeling markup language (UserML). In Henze, N., editor, *ABIS-Workshop: Personalization for the mobile World*, pages 17–21, Hannover, Germany.

[Heckmann, 2002b] Heckmann, D. (2002b). Towards user modeling in ubiquitous computing. In Jacoby, D., Krüger, A., and Malaka, R., editors, *AIMS2002*, pages 58–65, Lyon, France. in conjunction with the 15th European Conference on Artificial Intelligence.

[Heckmann, 2003a] Heckmann, D. (2003a). Integrating Privacy Aspects into Ubiquitous Computing: A Basic User Interface for Personalization. In Krüger, A. and Malaka, R., editors, *Artificial Intelligence in Mobile Systems (AIMS 2003)*, pages 106–110, Seattle, USA. in conjunction with the Fifth International Conference on Ubiquitous Computing.

[Heckmann, 2003b] Heckmann, D. (2003b). Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In Hoto, A. and Stumme, G., editors, *LLWA Lehren - Lernen - Wissen - Adaptivität (ABIS2003)*, pages 283–286, Karlsruhe, Germany.

[Heckmann, 2003c] Heckmann, D. (2003c). A specialized representation for ubiquitous computing and user modeling. In Chevrest, K., de Carolis, N., and Krüger, A., editors, *User Modeling in Ubiquitous Computing*, pages 26–28, Johnstown, PA, USA. in conjunction with the Ninth International Conference on User Modeling.

[Heckmann, 2003d] Heckmann, D. (2003d). A user model ontology in DAML. Published online at http://www.daml.org/ontologies/444.

[Heckmann, 2005a] Heckmann, D. (2005a). Distributed user modeling for situated interaction. In *35. GI Jahrestagung, Informatik 2005 - Situierung, Individualisierung und Personalisierung*, Lecture Notes in Informatics (LNI), pages 266–270, Bonn, Germany.

[Heckmann, 2005b] Heckmann, D. (2005b). Situation modeling and smart context retrieval with semantic web technology and conflict resolution. In *IJCAI-Workshop Modeling and Retrieval of Context (MRC2005)*, pages 128–132, Edinburgh, Scotland.

[Heckmann et al., 2005a] Heckmann, D., Brandherm, B., Schmitz, M., Schwartz, T., and von Wilamowitz-Moellendorf, B. M. (2005a). GUMO - the general user model ontology. In *Proceedings of the 10th International Conference on User Modeling*, pages 428–432, Edinburgh, Scotland. LNAI 3538: Springer, Berlin Heidelberg.

[Heckmann and Krüger, 2003] Heckmann, D. and Krüger, A. (2003). A user modeling markup language (UserML) for ubiquitous computing. In *Proceedings of the 8th International Conference on User Modeling*, pages 393–397, Johnstown, PA, USA. LNAI 2702: Springer, Berlin Heidelberg.

[Heckmann et al., 2005b] Heckmann, D., Schwartz, T., Brandherm, B., and Kröner, A. (2005b). Decentralized user modeling with UserML and GUMO. In Dolog, P. and Vassileva, J., editors, *Decentralized, Agent Based and Social Approaches to User Modeling, Workshop DASUM-05 at 9th International Conference on User Modelling, UM2005*, pages 61–66, Edinburgh, Scotland.

[Heider, 1958] Heider, F. (1958). *The Psychology of Interpersonal Relations*. New York: Wiley.

[Hinde, 2003] Hinde, S. (2003). Privacy legislation: a comparison of the US and European approaches. *Computers and Security*, 22(5):378–387.

[Hoeppner et al., 1980] Hoeppner, W., Jameson, A., and Wahlster, W. (1980). The natural language system HAM-RPM as a hotel manager: Some representational prerequisites. In Wilhelm, R., editor, *GI-10. Jahrestagung Saarbrücken*, pages 459–473. Berlin, Heidelberg, New York: Springer.

[Holsapple and Joshi, 2002] Holsapple, C. W. and Joshi, K. D. (2002). A collaborative approach to ontology design. *Communications of the ACM*, 45(2):42–47.

[Jacobson, 1999] Jacobson, R., editor (1999). *Information Design*. The MIT Press Cambridge Massachusetts.

[Jameson, 2001a] Jameson, A. (2001a). *Designing User-Adaptive Systems*. Tutorial presented at IUI, Santa Fe, New Mexico.

[Jameson, 2001b] Jameson, A. (2001b). Modeling both the context and the user. *Personal Technologies*, 5(1):29–33.

[Jameson, 2001c] Jameson, A. (2001c). *Systems That Adapt to Their Users: An Integrative Perspective*. Department of Computer Science, Saarland University, Saarbrücken, Germany.

[Jameson, 2003] Jameson, A. (2003). Adaptive interfaces and agents. In Jacko, J. A. and Sears, A., editors, *Human-computer interaction handbook*, pages 305–330. Mahwah, NJ: Erlbaum.

[Jameson and Buchholz, 1998] Jameson, A. and Buchholz, K. (1998). Einleitung zum Themenheft "Ressourcenadaptive kognitive Prozesse" [Introduction to the special issue on "Resource-Adaptive Cognitive Processes"]. *Kognitionswissenschaft*, 7:95–100.

[Jameson et al., 1980] Jameson, A., Hoeppner, W., and Wahlster, W. (1980). The natural language system HAM-RPM as a hotel manager: some representation prerequisites. In Wilhelm, R., editor, *Proceedings of 10th GI Jahrestagung*, pages 459–473. Springer, Berlin.

[Jensen, 2001] Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Statistics for Engineering and Information Science. New York, Berlin, Heidelberg: Springer.

[Jiang and Steenkiste, 2002] Jiang and Steenkiste (2002). A hybrid location model with a computable location identifier for ubiquitous computing. In *Proceedings of the Fourth International Conference on Ubiquitous Computing, Ubicomp*, pages 246–263.

[Jorge, 2001] Jorge, J. A. (2001). Adaptive tools for the elderly: new devices to cope with age-induced cognitive disabilities. In *EC/NSF workshop on Universal accessibility of ubiquitous computing*, pages 66–70. ACM Press.

[Kamp and Reyle, 1990] Kamp, H. and Reyle, U. (1990). *From Discourse to Logic / An Introduction to the Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

[Kay, 1995] Kay, J. (1995). The UM toolkit for reusable, long term user models. *User Modeling and User-Adapted Interaction*, 4(3):149–196.

[Kay et al., 2002] Kay, J., Kummerfeld, B., and Lauder, P. (2002). Personis: A server for user models. In de Bra, P., Brusilovsky, P., and Conejo, R., editors, *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, LNCS 2347, pages 203–212. Springer-Verlag Berlin Heidelberg.

[Kay et al., 2003] Kay, J., Kummerfeld, B., and Lauder, P. (2003). Managing private user models and shared personas. In Cheverst, K., de Carolis, N., and Krüger, A., editors, *UM03 Workshop on User Modeling for Ubiquitous Computing*, pages 1–11.

[Kidd et al., 1999] Kidd, C. D., Orr, R. J., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., and Newstetter, W. (1999). The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the Second International Workshop on Cooperative Buildings - CoBuild'99*, pages 191–198, Pittsburgh, USA.

[Kim et al., 2004] Kim, M.-J., Lee, E., Oh, B.-W., and Kim, M. (2004). The spatial data server based on open gis standards in heterogeneous distributed environment. In *12th International Conference on Geoinformatiocs*, pages 242–244, University of Gävle, Sweden.

[Kleinbauer et al., 2003] Kleinbauer, T., Bauer, M., and Jameson, A. (2003). SPECTER - a user-centered view on ubiquitous computing. In *ABIS 2003: 11th GI-Workshop "Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen"*, pages 299–306, Karlsruhe, Germany.

[Klesen, 2002] Klesen, M. (2002). NECA: Net environment for embodied emotional conversational agents: Report on affective reasoning and cultural diversity. Technical Report D9b, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany.

[Kobsa, 1990] Kobsa, A. (1990). Modeling the user's conceptual knowledge in bgp-ms, a user modeling shell system. *Computational Intelligence*, 6:193–208.

[Kobsa, 2001a] Kobsa, A. (2001a). Generic user modeling systems. *User Modelling and User-Adapted Interaction Journal*, 11(1-2):49–63.

[Kobsa, 2001b] Kobsa, A. (2001b). Invited keynote: Tailoring privacy to user's needs. In Bauer, M. and Gmytrasiewicz, P. J., editors, *Proceedings of the 8th international conference on User Modeling*, pages 303–313, Sonthofen, Germany. Springer LNAI 2109.

[Kobsa, 2001c] Kobsa, A., editor (2001c). *User Modeling and User-Adapted Interaction - Ten Year Anniversary Issue*, volume 11, Netherlands. Kluwer Academic Publishers.

[Kobsa, 2002] Kobsa, A. (2002). Personalized hypermedia and international privacy. *Communications of the ACM*, 45(5):64–67.

[Kobsa et al., 2001] Kobsa, A., Koenemann, J., and Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationships. In *The Knowledge Engineering Review*, volume 16, pages 111–115. Cambridge University Press.

[Kono et al., 1994] Kono, Y., Ikeda, M., and Mizoguchi, R. (1994). Themis: a nonmonotonic inductive student modeling system. *Journal of Artificial Intelligence in Education*, 5(3):371–413.

[Kort et al., 2001] Kort, B., Reilly, R., and Picard, R. W. (2001). An affective model of interplay between emotions and learning: Reengineering educational pedagogy - building a learning companion. In *Proceedings of International Conference on Advanced Learning Technologies ICALT 2001*, pages 43–48, Madison Wisconsin, USA.

[Kray, 2003] Kray, C. (2003). *Situated Interaction on Spatial Topics*, volume 274 of *DISKI*. Aka Verlag, Berlin. ISBN 1-58603-391-3.

[Krenn et al., 2003] Krenn, B., Schröder, M., and Klesen, M. (2003). Emotion modelling in neca: Generating affective conversations between ecas. Technical report, NECA project.

[Kröner, 2001] Kröner, A. (2001). *Adaptive Layout of Dynamic Web Pages*, volume 248 of *DISKI*. Aka Verlag, Berlin. ISBN 3-89838-248-6.

[Kröner, 2005] Kröner, A. (2005). On building an episodic memory for a context-aware personal assistant. In *1st International Workshop on Exploiting Context Histories in Smart Environments, in conjunction with Pervasive 2005*, Munich, Germany.

[Krüger et al., 2004] Krüger, A., Butz, A., Müller, C., Stahl, C., Wasinger, R., Steinberg, K. E., and Dirschl, A. (2004). The connected user interface: Realizing a personal situated navigation service. In Vanderdonckt, J., Nunes, N. J., and Rich, C., editors, *Proc. of the 9th International Conference on Intelligent User Interfaces*, pages 161–168, Funchal, Madeira, Portugal. ACM.

[Kruppa et al., 2005] Kruppa, M., Heckmann, D., and Krüger, A. (2005). Adaptive multimodal presentation of multimedia content in museum scenarios. *KI Journal*, 1:56–59.

[Kruppa and Krüger, 2003] Kruppa, M. and Krüger, A. (2003). Concepts for a combined use of personal digital assistants and large remote displays. In Schulze, T., Schlechtweg, S., and Hinz, V., editors, *Simulation und Visualisierung 2003*, pages 349–361, Magdeburg, Germany. SCS Verlag.

[Kruppa et al., 2003] Kruppa, M., Krüger, A., Rocchi, C., Stock, O., and Zancanaro, M. (2003). Seamless personalized tv-like presentations on mobile and stationary devices in a museum. In *Proceedings of ICHIM*.

[Larkin and Simon, 1987] Larkin, J. H. and Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11:65–99.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. R. (1999). *Resource Description Framework (RDF) Model and Syntax Specification*. W3C. W3C recommendation.

[Lee, 1999] Lee, K. (1999). Integration of various emotion eliciting factors for life-like agents. *ACM Multimedia*, Part 2:155–158.

[Leonhardt, 1998] Leonhardt, U. (1998). *Supporting Location-Awareness in Open Distributed Systems*. PhD thesis, Imperial College London, Department of Computing.

[Lindmark and Heckmann, 2000] Lindmark, K. and Heckmann, D. (2000). Interpreting symptoms of cognitive load and time pressure in manual input. In Müller, M., editor, *Adaptivität und Benutzermodelierung in interaktiven Softwaresystemen (ABIS 2000)*, pages 93–101, Osnabrück, Germany.

[Lönneker, 2003] Lönneker, B. (2003). *Konzeptframes und Relationen: Extraktion, Annotation und Analyse französischer Corpora aus dem World Wide Web*. PhD thesis, Fachbereich Sprach-, Literatur- und Medienwissenschaft der Universität Hamburg. (Chapter 3.5).

[Lorenz, 2005] Lorenz, A. (2005). Agent-based ubiquitous user modeling. In *Proceedings of the 10th International Conference on User Modeling*, pages 512–514, Edinburgh, Scotland. LNAI 3538: Springer, Berlin Heidelberg.

[Mackay, 1998] Mackay, W. E. (1998). Augmented reality: linking real and virtual worlds: a new paradigm for interacting with computers. In *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, pages 13–21, L'Aquila, Italy. ACM Press, New York, USA.

[Maffioletti, 2001] Maffioletti, S. (2001). Requirements for an ubiquitous computing infrastructure. *Cultura Narodov Prichernomoria journal*, 3:35–40. Mezhvuzovskii publisher, Simferopol, Ukraine, ISSN 1562-0808.

[Malaka, 2003] Malaka, R. (2003). Context and user adapted mobile interaction. In *Nexus Workshop on Context-Aware Systems*, Stuttgart, Germany.

[Malaka and Zipf, 2000] Malaka, R. and Zipf, A. (2000). DEEP MAP - challenging it research in the framework of a tourist information system. In Fresenmaier, D., Klein, D., and Buhalis, D., editors, *Information and Communication Technologies in Tourism 2000*, pages 15–27. Springer.

[Manola and Miller, 2004] Manola, F. and Miller, E. (2004). Rdf primer, w3c recommendation draft. Published online at http://www.w3.org/TR/rdf-primer/.

[Martin et al., 2004] Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K. (2004). Bringing semantics to web services: The OWL-S approach. In *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, California, USA.

[Mattern, 2002a] Mattern, F. (2002a). Physical-virtual integration, ubicomp applications & implications. In *Summer School on Ubiquitous and Pervasive Computing*, chapter 9, pages 1–95. ETH Zürich Folder, Schloss Dagstuhl, August 7-14, Germany.

[Mattern, 2002b] Mattern, F. (2002b). Smart identification. In *Summer School on Ubiquitous and Pervasive Computing*, chapter 1, pages 1–75. ETH Zürich Folder, Schloss Dagstuhl, August 7-14, Germany.

[Matthews et al., 2003] Matthews, G., Deary, I. J., and Whiteman, M. C. (2003). *Personality Traits*. Cambridge University Press, 2nd edition.

[Maybury and Wahlster, 1998] Maybury, M. and Wahlster, W., editors (1998). *Readings in intelligent user interfaces*. San Francisco, CA: Morgan Kaufmann.

[McCarthy and Buvac, 1998] McCarthy, J. and Buvac, S. (1998). Formalizing context (expanded notes). In Aliseda, A., van Glabbek, R. J., and Westerstahl, D., editors, *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. Center for the Study of Language and Information, Standford University, CA.

[McCrea and John, 1992] McCrea, R. and John, O. (1992). An introduction to the five-factor model and its implications. *Journal of Personality*, (60):171–215.

[McGuinness and van Harmelen, 2003] McGuinness, D. L. and van Harmelen, F. (2003). OWL web ontology language overview. http://www.w3.org/TR/owl-features/ W3C Recommendation.

[Mehrabian, 1996] Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temprament. *Current Psychology: Developmental, Learning, Personality, Social*, (14):261–292.

[Meyer and Rakotonirainy, 2003] Meyer, S. and Rakotonirainy, A. (2003). A survey of research on context-aware homes. In Johnson, C., Montague, P., and Steketee, C., editors, *Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing*, volume 21 of *Conferences in Research and Practice in Information Technology*, pages 159–168, Adelaide, Australia. ACS.

[Miller et al., 1993] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, G., and Miller, K. (1993). Five Papers on WordNet. Technical Report, Cognitive Science Laboratory, Princeton University.

[Morris, 1992] Morris, W. N. (1992). More on the mood-emotion distinction. *Psycoloquy*, 3(7):Mood (6). http://psycprints.ecs.soton.ac.uk/archive/00000231/.

[Morsing and Schultheis, 2004] Morsing, A. and Schultheis, H. (2004). Location service for REAL. Fopra, Saarland University, Saarbrücken, Germany.

[Müller et al., 2001] Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., and Wittig, F. (2001). Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P., editors, *UM2001, User Modeling: Proceedings of the Eighth International Conference*, pages 24–33. Springer, Berlin. Available from http://dfki.de/∼jameson/abs/MuellerGJ+01.html.

[Müller and Wasinger, 2002] Müller, C. and Wasinger, R. (2002). Adapting multimodal dialog for the elderly. In *Proceedings of the ABIS-Workshop 2002 on Personalization for the Mobile World*, pages 31–34, Hannover, Germany.

[Müller and Wittig, 2003] Müller, C. and Wittig, F. (2003). Speech as a source for ubiquitous user modeling. In *Workshop on User Modelling for Ubiquitous Computing, in conjunction with User Modeling*, pages 46–50, Pittsburgh, PA, USA.

[Narayanan et al., 2003] Narayanan, S., Baker, C., Fillmore, C., and Petruck, M. (2003). FrameNet meets the Semantic Web: Lexical Semantics for the Web. In Fensel, D., editor, *ISWC 2003*, LNCS 2870, pages 771–787. Springer Verlag Berlin Heidelberg.

[Nass and Lee, 2001] Nass, C. and Lee, K. M. (2001). Does Computer-Synthesized Speech Manifest Personality? Experimental tests of Recognition, Similarity-Attraction, and Consistency-Attraction. *Journal of Experimental Psychology Applied*, 7(3):171–181.

[Niles and Pease, 2001] Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In Welty, C. and Smith, B., editors, *In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine.

[Oberlander, 2004] Oberlander, J. (2004). Introduction to cognitive science, a third-year undergraduate module. University of Edinburgh, http://www.inf.ed.ac.uk/teaching/courses/ics/.

[Oberlander and Gill, 2004] Oberlander, J. and Gill, A. (2004). Individual differences and implicit language: personality, parts-of-speech and pervasiveness. In *Proceedings of the 26th AnnualConference of the Cognitive Science Society*, pages 1035–1040. Hillsdale, NJ: LEA.

[Ortony et al., 1988] Ortony, A., Clore, G., and Collins, A. (1988). *Cognitive Structure of Emotions*. Cambridge University Press.

[Orwant, 1994] Orwant, J. (1994). Apprising the user of user models: Doppelgänger's interface. In *Proceeding of the Forth International Conference on User Modeling*, pages 151–156, Massechusetts, USA. MITRE, User Modeling Inc, Hyannis.

[Orwant, 1995] Orwant, J. (1995). Heterogeneous learning in the Doppelgänger user modeling system. *User Modeling and User-Adapted Interaction*, 4(2):59–106.

[Orwant, 1996] Orwant, J. (1996). For want of a bit the user was lost: Cheap user modeling. *IBM Systems Journal*, 35(348):398–416.

[OWL-S Coalition, 2005] OWL-S Coalition (2005). *OWL-S 1.0 Release*. DAML, At http://www.daml.org/services/owl-s/1.1/.

[Pease et al., 2002] Pease, A., Niles, I., and Li, J. (2002). The suggested upper merged ontology: A large ontology for the semanticweb and its applications. In *AAAI-2002Workshop on Ontologies and the Semantic Web*, Working Notes. http://projects.teknowledge.com/AAAI-2002/Pease.ps.

[Pennebaker and King, 1999] Pennebaker, J. W. and King, L. A. (1999). Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77(6):1296–1312.

[Picard, 1997] Picard, R. W. (1997). *Affective Computing*. MIT Press, Cambridge, MA.

[Prendinger and Ishizuka, 2004] Prendinger, H. and Ishizuka, M. (2004). *Life-Like Characters - Tools, Affective Functions, and Applications*. Springer, Berlin Heidelberg.

[Ram and Park, 2004] Ram, S. and Park, J. (2004). Semantic Conflict Resolution Ontology (SCROL): An ontology for detecting and resolving data and schema-level semantic conflicts. *IEEE Transactions on Knowledge and Data Engeneering*, 16(2):189–202.

[Reed and Lenat, 2002] Reed, S. L. and Lenat, D. B. (2002). Mapping ontologies into Cyc. In *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, pages 1–6, Edmonton, Canada.

[Salber et al., 1999a] Salber, D., Dey, A. K., and Abowd, G. D. (1999a). The context toolkit: Aiding the development of context-enabled applications. *CHI*, pages 434–441.

[Salber et al., 1999b] Salber, D., Dey, A. K., and Abowd, G. D. (1999b). Defining an HCI Research Agenda for an Emerging Interaction Paradigm. Technical report, GVU Center and College of Computing, Georgia Tech.

[Schmidt, 2003] Schmidt, A. (2003). *Ubiquitous Computing - Computing in Context*. PhD thesis, CSEG, Lancaster University, Lancaster, England, U.K.

[Schmidt, 2005] Schmidt, A. (2005). A layered model for user context management with controlled aging and imperfection handling. In *Proceedings of the 2nd International Workshop on Modeling and Retrieval of Context (MRC 2005)*, pages 27–38, Edinburgh, Scotland.

[Schmitt, 2005] Schmitt, M. (2005). *Dynamische Modellierung interpersoneller Beziehungen zwischen virtuellen Charakteren*. PhD thesis, Computer Science Department, Saarland University, Germany.

[Schneider, 2003] Schneider, M. (2003). A smart shopping assistant utilizing adaptive plan recognition. In Hoto, A. and Stumme, G., editors, *Lehren - Lernen - Wissen - Adaptivität (LLWA04)*, pages 331–334.

[Schneider, 2004] Schneider, M. (2004). Towards a transparent proactive user interface for a shopping assistant. In *Workshop on Multi-User and Ubiquitous User Interfaces (MU3I)*, pages 31–35, Funchal, Madeira, Portugal.

[Scowen, 1998] Scowen, R. S. (1998). Extended BNF - a generic base standard. Technical report, ISO/IEC 14977. http://www.cl.cam.ac.uk/ mgk25/iso-14977.pdf.

[Sevcenko, 2003] Sevcenko, M. (2003). Online presentation of an upper ontology. In *In Proceedings of Znalosti 2003*, Ostrava, Czech Republic.

[Shafer, 1979] Shafer, G. (1979). *Mathematical Theory of Evidence*. Princeton: University Press.

[Shedroff, 1999] Shedroff, N. (1999). Information interaction design: A unified field theory of design. In Jacobson, R., editor, *Information Design*, chapter 11, pages 267–292. The MIT Press Cambridge Massachusetts, 1st edition.

[Silverman et al., 2003] Silverman, B. G., Cornwell, J., and O'Brien, K. (2003). *Metrics and methods in human performance research toward individual and small unit simulation*, chapter 9 Human Performance Simulation. Human Systems Information Analysis Center, Washington, DC.

[Slaney and Thiébaux, 2001] Slaney, J. K. and Thiébaux, S. (2001). Blocks world revisited. *Artif. Intell.*, 125(1-2):119–153.

[Spassova et al., 2005] Spassova, L., Wasinger, R., Baus, J., and Krüger, A. (2005). Product associated displays in a shopping scenario. In *4th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 05)*, pages 210–211, Vienna, Austria.

[Stahl et al., 2005] Stahl, C., Brandherm, B., Schmitz, M., and Schwarz, T. (2005). Navigational and shopping assistance on the basis of user interactions in intelligent environments. In *The IEE International Workshop on Intelligent Environments*, pages 182–191, Colchester, UK.

[Stahl and Heckmann, 2004a] Stahl, C. and Heckmann, D. (2004a). Using semantic web technology for ubiquitous hybrid location modeling. In *UbiGis 2004 - First International Workshop on Ubiquitous GIS*, pages 1–9, Gävle, Sweden.

[Stahl and Heckmann, 2004b] Stahl, C. and Heckmann, D. (2004b). Using semantic web technology for ubiquitous location and situation modeling. *The Journal of Geographic Information Sciences CPGIS: Berkeley*, 10(2):157–165.

[Stahl et al., 2004] Stahl, C., Krüger, A., Baus, J., Heckmann, D., Wasinger, R., and Schneider, M. (2004). REAL: Situated dialogue in instrumented environments. In *Workshop on Invisible and Transparent Interfaces at AVI 2004*, pages 10–15, Gallipoli Lecce, Italy. ITI Workshop, Part of AVI 2004 (Advanced Visual Interfaces).

[Stuckenschmidt and van Harmelen, 2005] Stuckenschmidt, H. and van Harmelen, F. (2005). *Information Sharing on the Semantic Web*. Springer Verlag Berlin Heidelbert.

[Studer et al., 2003] Studer, R., Hotho, A., Stumme, G., and Volz, R. (2003). Semantic web - state of the art and future directions. *Künstliche Intelligenz*, 3(3):5–9.

[Torre, 2000] Torre, I. (2000). A modular approach for user modelling. In *Adaptive Hypermedia and Adaptive Web Based Systems*, Lecture Notes in Computer Science 1892, pages 415–420. Springer Verlag.

[Ullmer and Ishii, 2001] Ullmer, B. and Ishii, H. (2001). *Human-Computer Interaction in the New Millennium*, chapter Emerging Frameworks for Tangible User Interfaces, pages 579–601. Addison Wesley Professional.

[Vassileva et al., 2003] Vassileva, J., McCalla, G., and Greer, J. (2003). Multi-agent multi-user modeling in i-help. *User Modeling and User-Adapted Interaction*, 13:179–210.

[W3C-P3P, 2003] W3C-P3P (2003). Platform for privacy preferences (p3p) project. http://www.w3.org/P3P/.

[Wahlster, 1988] Wahlster, W. (1988). Distinguishing user models from discourse models. *Computational Linguistics*, 14(3):101–113.

[Wahlster, 2003] Wahlster, W. (2003). Towards symmetric multimodality: Fusion and fission of speech, gesture, and facial expression. In Günther, A., Kruse, R., and Neumann, B., editors, *KI 2003: Advances in Artificial Intelligence. Proceedings of the 26th German Conference on Artificial Intelligence*, pages 1–18, Hamburg, Germany. Berlin, Heidelberg: Springer, LNAI 2821.

[Wahlster and Kobsa, 1986] Wahlster, W. and Kobsa, A. (1986). Dialog-based user models. In *Journal Proceedings of the IEEE, Special Issue on Natural Language Processing*, pages 948–960.

[Wahlster and Kobsa, 1989] Wahlster, W. and Kobsa, A. (1989). User models in dialog systems. In Kobsa, A. and Wahlster, W., editors, *User Models in Dialog Systems*, pages 4–34. Springer, Berlin. (Symbolic Computation Series).

[Wahlster and Tack, 1997] Wahlster, W. and Tack, W. (1997). Ressourcenadadaptive kognitive Prozesse. In Jarke, M., Pasedach, K., and Pohl, K., editors, *Informatik 97 – Informatik als Innovationsmotor*, pages 51–57. Springer, Berlin, Heidelberg, New York.

[Wasinger et al., 2003a] Wasinger, R., Olivia, D., Heckmann, D., Braun, B., Brandherm, B., and Stahl, C. (2003a). Adapting spoken and visual output for a pedestrian navigation system, based on given situational statements. In Hoto, A. and Stumme, G., editors, *LLWA Lehren - Lernen - Wissen - Adaptivität (ABIS2003)*, pages 343–346, Karlsruhe, Germany.

[Wasinger et al., 2003b] Wasinger, R., Stahl, C., and Krüger, A. (2003b). M3I in a pedestrian navigation and exploration system. In *Proceedings of the Fifth International Symposium on Human Computer Interaction with Mobile Devices*, pages 481–485, Udine, Italy.

[Wasinger and Wahlster, 2005] Wasinger, R. and Wahlster, W. (2005). *The Anthropomorphized Product Shelf: Symmetric Multimodal Human-Environment Interaction*, chapter in Emile Aarts and José Luis Encarnação (eds.), True Visions: The Emergence of Ambient Intelligence, page (in press). Heidelberg: Springer. ISBN: 3-540-28972-0.

[Weiser, 1991] Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):94–104.

[Weiser, 1993] Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications on the ACM*, 36(7):75–84.

[Wellner et al., 1993] Wellner, P., Mackay, W., and Gold, R. (1993). Special issue on computer augmented environments: back to the real world. *Communications of the ACM*, 36(7):24 – 27.

[Wikipedia, 2005] Wikipedia (2005). Wikipedia, the free online encyclopedia. http://en.wikipedia.org/.

[Wittig and Müller, 2003] Wittig, F. and Müller, C. (2003). Implicit feedback for user-adaptive systems by analysing the users' speech. In Hoto, A. and Stumme, G., editors, *LLWA Lehren - Lernen - Wissen - Adaptivität (ABIS2003)*, pages 347–350, Karlsruhe, Germany.

[Wrightson, 2005] Wrightson, A. (2005). Semantics of well formed xml as a human and machine readable language: Why is some xml so difficult to read? In *Proceedings of Extreme Markup Languages*, pages 1–11, Montréal, Québec. IDEAlliance.

[Wu, 2002] Wu, H. (2002). *A reference architecture for Adaptive Hypermedia Applications*. PhD thesis, Technische Universiteit Eindhoven, The Netherlands.

[Yudelson et al., 2005] Yudelson, M., Gavrilova, T., and Brusilovsky, P. (2005). Towards user modeling meta-ontology. In *Proceedings of the 10th International Conference on User Modeling*, pages 448–452, Edinburgh, Scotland. LNAI 3538: Springer, Berlin Heidelberg.

[Zhuge, 2005] Zhuge, H. (2005). Semantic grid: Scientific issues, infrastructure, and methodology. *Communications of the ACM*, 48(4):117–119.