A Specification for Agent-Based Distributed User Modelling in Ubiquitous Computing

Andreas Lorenz

Fraunhofer Institute for Applied Information Technology
Schloss Birlinghoven
53754 Sankt Augustin, Germany
Andreas.Lorenz@fit.fraunhofer.de

Abstract. This paper introduces an approach for applying agent technology for user modelling in ubiquitous computing. It illustrates the research issues in distributing the knowledge about the user across active entities and distributed user-model acquisition and application methods, and specifies the agents using a defined communication framework for distributed user-modelling for ubiquitous computing. Regarding the requirements in ubiquitous computing, co-operating agents build ad-hoc networks for receiving information from other entities and distributing knowledge to other components in the network. Therefore, the specified agents are able to react both to their environment and to messages received from neighbouring components.

Introduction

In the classic approach for *personalized system development*, the application contains specific user-modelling components inside the personalization engine. By user model acquisition, information about the user is extracted from sensoring the environment and knowledge from explicit and implicit user feedback is inferred [1]. The inferred knowledge usually is written to an internal database, mapping user attributes to their values. In the next step, both the component listening to sensor data and the knowledge-base about the user are separated from the internal application logic. In the first case, *sensor-servers* retrieve data streams from different sensors placed in the environment and deliver the information to the application. Using remote sensor-servers distributing sensor data on a network, different applications can concurrently receive the same data. In the second case, *User-Model Servers* [2] work as an application-external knowledge-base. The derived knowledge about the user is delivered to the server that hosts the information for different applications. For mobile applications, this enables systems on small devices even with limited memory and computing power to have access to meaningful user models. Furthermore, it enables different applications to have access to the same knowledge and to adapt consistently.

In the vision of ubiquitous computing the user has *one personal information space* independent of devices and the system manages the information spaces of its users.

For future application development in ubiquitous computing we expect centralized design-approaches to be confronted with uncountable clients on heterogeneous devices with different properties. In our vision, *distributed user modelling* approaches need to replace monolithic centralized user modelling by distributed user model fragments [3]. To become true, this vision requires several pre-conditions to be fulfilled:

1. The network of distributed components needs to be self-adapting, especially regarding available communication partners and technology.
2. The information needs to migrate between different hosts and platforms without being central-controlled.
3. The communication infrastructure and technical details need to be hidden from the user modelling components, and their developers.
4. Typically, application designers building distributed applications have to guarantee the following non-functional requirements: scalability, openness, heterogeneity, fault-tolerance, and resource sharing.

Facilitating communication and coordination of distributed components, we will implement cooperating agents as active components hosting on the devices and using a defined communication framework. In contrast to other approaches for applying agent-technology, sets of agents will be implemented for distributed user-modelling, user-model acquisition and user-model-application instead of a one-to-one relationship between the user and a User-Agent. Each local component might detect a section of the global state, but the network of agents will piece together these partial states for *distributed representation of knowledge about the user*. This paper demonstrates our specification of such agents.

Agent-Technology in User-Modelling

Recent *agent-based user modelling* approaches usually consists of two parts: a User Modelling Service and a User Agent (often also referred to as Personal Agent). The former keeps track of the user's interaction with the application and within the environment, and stores the inferred user and environmental characteristics. The latter usually represents the user in the system. For mobile / nomadic users, mobile agents can move with the user between devices and applications. In this section we describe recent attempts of combining user modelling and agent technologies for the application fields mentioned in the Introduction to this paper. To get an overview of the variety of agent definitions, modelling techniques, and architectures in this field, the reader is referred to [4].

Driven by the boom of web-applications in the late 1990s, the value of personalization was increasingly recognized in the field of intelligent information access on the WWW. Pazzani and Billsus [5] have introduced adaptive web site agents that recommend relevant documents to the user in an Amazon.com-like manner. They argued that the information is best used to change the behaviour of an animated agent (avatar) to assist the user. In Billsus and Pazzani [6] an intelligent information agent is considered to be a personal assistant that gradually learns about users' interests. Like the adaptive web agents presented in Menczer and Belew [7], agent technology is either used for personalized information acquisition or for individual information presentation.

In the domain of eLearning, Vassileva *et al.* [8] base the adaptation within the I-Help system [9] on models of human users maintained by personal agents: "*Each personal agent manages a user model containing information about the user's goals (help requests, current goal), about* knowledge resources / competencies *on certain topics or tasks, and about the* relationships *existing between the user and other users.*" The Baghera project [10] has implemented personal interface agents for students and teachers, and tutor agents that base whose didactical decisions on a student model. In order to integrate human-like intelligent tutors into collaborative learning environments, Goodman *et al.* [11] have also proposed to integrate tutoring agents. These approaches have in common that student and tutor agents are connected with external user models.

Furthermore, agent technologies have been applied for personalizing location-based services like city- and tourism-guides. The Deep Map Agents introduced in Fink and Kobsa [12] provide tour recommendations, analyse spoken text, generate speech output etc. These agents, which loosely adhere to the FIPA agent specification [13], communicate to a User Modelling Server (UMS) about the user's interaction with the system and query the UMS for user characteristics. In the EU-founded CRUMPET project [14], FIPA compliant user agents are hosted on the end user terminal devices and provide the user with the service GUI. These agents adapt the information presentation to the platform evaluating the usage profile of the user.

In summary of these approaches, the agents usually query an external user model. In terms of multi-agent system development, the internal knowledge-base of such an agent actually is or refers to a user model; in terms of the general scheme of an adaptive system [1], team working agents improve user model acquisition resp. user model application.

Agent-based distributed User-Modelling

To be able to fulfil the requirements of ubiquitous computing, we propose to have a network of small active entities on the client side. Resent research in smart *sensor-networks* enables for placing huge numbers of intelligent senor-components ("smart dust") in the environment. Smart sensors are equipped with small processors that enable for intelligent information acquisition [15]. In *self-organizing networks*, such as Intel's iMote approach [16], sensor technologies build ad-hoc sensor-networks and deliver requested information on demand. Similar procedures can be applied on higher layers in the system-design. For example, modelling components receive sensor-data and distribute inferred knowledge in something like a "modelling-network", which will have effect on controlling components and so forth. In difference to sensor-networks, the components actually receive pre-processed data from virtual components instead of direct measuring the physical environment. Therefore, we propose to have *distributed active entities receiving data from and delivering information to other entities*. As active entities, software-agents have their own thread of control making them appear like "active objects with initiative" [17] localizing not only code and state but their invocation as well. In other words, when and how an agent acts is determined by the agent. Regarding the assumption that in ubiquitous computing there won't be a central server hosting databases for all entities, the knowledge-base and the decision-finding process will be distributed across the agents. There will exist neither central user modelling nor information-acquisition/knowledge-application components.

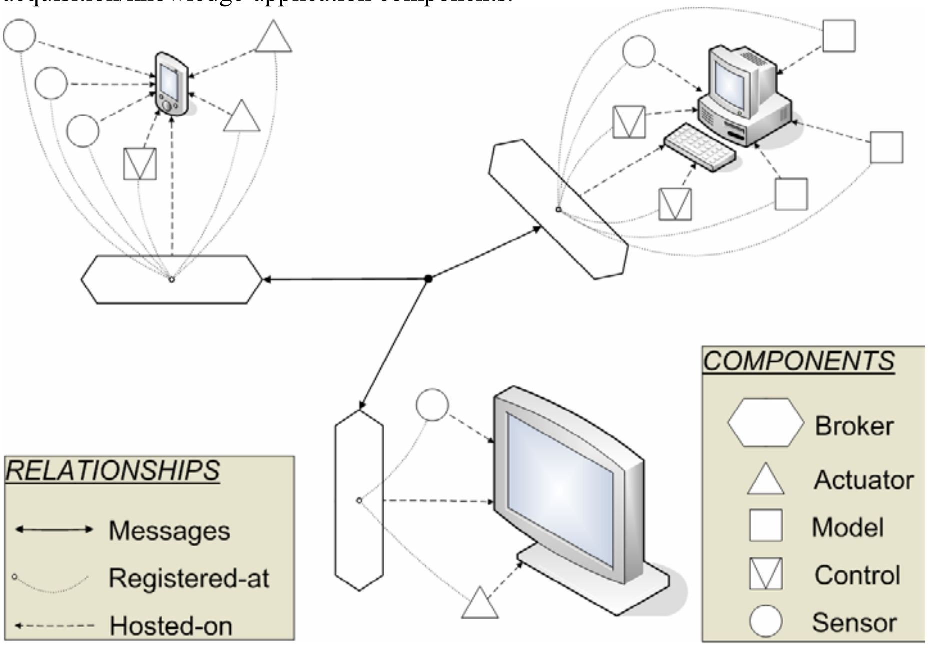

Fig. 1 Distributed User Modelling Platform

Due to the distribution of functionality and knowledge, the agents will be categorized virtually. This ensures encapsulated inter-package communication inside and broadcasting to a specific category. Though system developers are able to integrate their own packages, we propose to have four categories of *sensoring*, *modelling*, *controlling* and *actuating agent*s [18, 19, 20]. For each category, networks of highly specialized software-agents process small tasks like delivering one information snippet or deciding to display data on a particular device. Each category will be distributed over different devices, e.g. among others the light sensor of a PDA, the infrared sensor of an automatic door and the GPS-sensor of the car are part of the sensor-package regardless to their physical location and environment. In turn, each device potentially hosts agents of several categories, e.g. a PDA independently hosts sensors for light-conditions, background-noise and pen-input as well as controlling agents for content-selection and actuators for video-streaming or adjustment of the display-brightness. Fig. 1 illustrates the distributed agents hosted on different devices and their relations to each other.

Example
To illustrate the information flow between the distributed components we will shortly describe one of the application scenarios. In this scenario, the hosts illustrated in Fig. 1 are a kiosk-system and a public information display in an airport and the personal device of the traveller. The kiosk and the display are connected via LAN and the user's PDA can establish Bluetooth-connections to the kiosk-system, which is able to read the RFID-Tag fixed to the flight-ticket of the customer. The service offered to the customer is time- and route planning on the large airport: When a traveller passed by the kiosk, the public display shows the flight-number and destination, and guides the traveller to the gate anonymously, including the estimated needed time. If the customer wants to have a personal plan, she can accept the Bluetooth-connection between her PDA and the kiosk.

The communication platform

The basic underlying cooperation-approach between the agents is *cooperation by information-exchange*. Like a middleware, brokers hide the complexity of communication from the other agents. This concept can be seen in between of the blackboard-approach and the message-sending approach well-known in the field of multi-agent systems [21]: For local agents, the broker provides access to a message-board whereas the information exchange between devices is based on message-sending between the brokers (Fig. 1). The agents register at the board based on defined check-in/check-out mechanisms, announcing what information they provide and what information they request.

In the example, the brokers of the kiosk, the information-display and a database-server are connected by Ethernet continuously. The sensoring-agent, logged in to the kiosk-broker, fires the event that an identifier has been received from the customer. A controlling-agent on the kiosk listens to the event and sends out a request for the number and destination of that flight. The broker sends the request to all known brokers in the network, which is answered by an agent on the database-server. The answer – broadcasted between the brokers – is received by a listening controlling agent on the information display, who generates the command to display the corresponding data for the rendering agents on the large screen display. The information has then migrated between the distributed components with different capabilities.

So far, the knowledge about the traveller's personal data is very limited. For privacy reasons, the display will not show any private information of the customer. If desired, the traveller can accept the Bluetooth connection at the kiosk. After the connection has been established, the broker on the kiosk covers roaming between different communication-technologies: Messages received from the Ethernet-connections are also forwarded via Bluetooth to the broker at the PDA. At a glance, the network of reachable agents is extended to cover the modelling-agents of the personal attributes, goals and task of the traveller. If the traveller passes by the kiosk system, which receives and sends out the RFID-identifier on the network, the agents on the PDA also receive the information from their local broker. Local agents can now request the position of the kiosk in order to update a local map, or controlling agents can re-arrange the time-schedule in order to skip the visit of the book-store because of potential time pressure. In turn, controlling agents can even generate commands for rendering agents on the kiosk, e.g. to display a map of nearby restaurants because of the traveller's habitat to have a coffee before boarding.

The advantage of the platform is that by standardized communication with local components distributed agents are released from discovering communication partners using different technologies. As the broker establish / loose connections to other brokers, the user model structure changes automatically and the sets of accessible information-sources and -destinations adapt to the current environment. The division of agents in different categories additionally distributes similar functionality on different devices. Regarding privacy concerns, the user has control on providing private information by allowing / disabling connections of private devices with other ones. If the connection was enabled, the global accessible user model is extended to include the attributes from personal devices at once.

Specification

Currently, our main objective is to provide a well-defined conceptual basis, in particular specifying the architecture and agents, communication and information-exchange, and cooperation-techniques and conflict-management [22], e.g. if many agents are potentially able to process the same information or agents receive ambiguous answers to a request. The realization phase has already been launched starting with the implementation of the framework and the specified communication-protocols for

- check-in/check-out
- subscribe-inform mechanism
- question-reply mechanism
- command-delivery
- acknowledgement
- exceptions

The messages are defined in EBNF and sending / receiving of such messages was implemented in several projects for receiving data from distributed sensors. In the next steps, we will finish the work on the specification, continue to implement the framework and focus on the implementation of the specified agents.

Agent-Specification

In general, we need two simple types of agents: *Information delivering agents* and *information receiving agents*, which include intelligent processing of the received information. Fig. 2 illustrates the derivation of the agents from those two basic types.

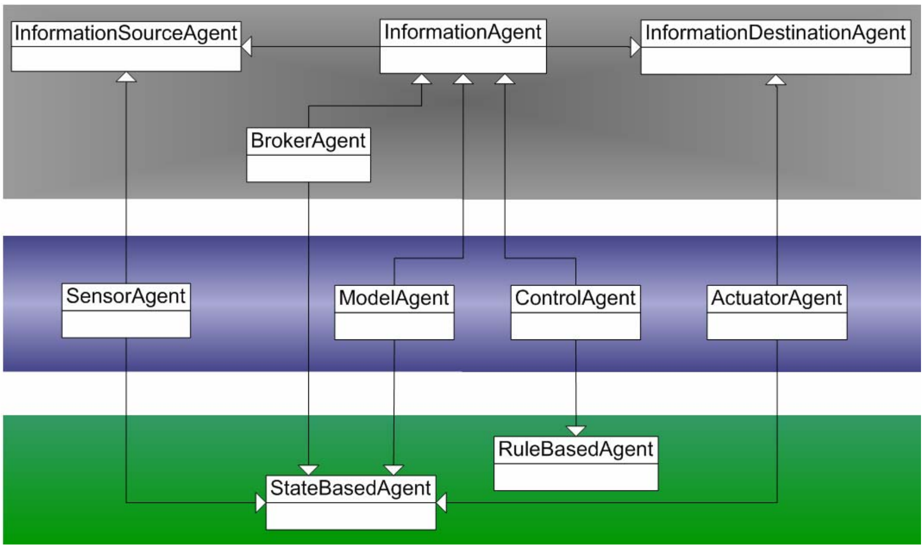

Fig. 2 The Agent's inheritance diagram

On top of Fig. 2, we have information sources on the left and information destinations on the right. Information sources do only deliver information towards other components; therefore they contain a list of all attributes they provide. The delivery can be performed either by throwing events or by answering requests from other components for specific information. In contrast, the information destinations are only able to receive information, either by listening to events or by pro-actively requesting data from others. They have an internal list of information they demand and are able to register as listeners. Derived from those two basic types, the third agent-type, the *information agent*, is able to send information as well as to receive and process information from others. As a special type of an information agent, the *broker agent* only forwards incoming information either towards other local agents or towards other brokers.

On the bottom of the figure, we depict two kinds of agent-specification from the field of multi-agent systems. Generally, we decided to model all agents in a state-based manner, except the controlling-agents. Incoming messages will trigger transitions in state-based agent-modelling, which sufficiently supports reactive behaviour and is also understandable for human developers in future. Fig. 3 exemplifies the states-diagram for the information source agents. The transitions between the states are usually defined in a transition-table.

Unfortunately, state-based agent-modelling is not applicable for knowledge-based agents. Beside the complexity of a state-diagram with many states for complex decision finding, the knowledge representation is implicit coded by the developer in the states and transitions. Adhering to state-based modelling, each derived agent would have its own state-diagram with specific conditional transitions. To determine the overall system's behaviour, a *rule-based approach* for controlling agents seems to be more appropriate. In this approach, the behaviour is coded by sets of rules of *IF condition THEN action*. Incoming messages trigger the interpretation of the rule's conditions and fire all rules with fulfilled pre-conditions. In conclusion, we have four different models for the agents taking part in the user-modelling process:

1. Sensory agents as state-based information source agents,
2. Modelling agents as state-based information agents,
3. Controlling agents as rule-based information agents and
4. Actuating agents as state-based information destination agents.

In the next subsection we will exemplary describe the state-diagram of an information source agent in more detail.

Information source agent

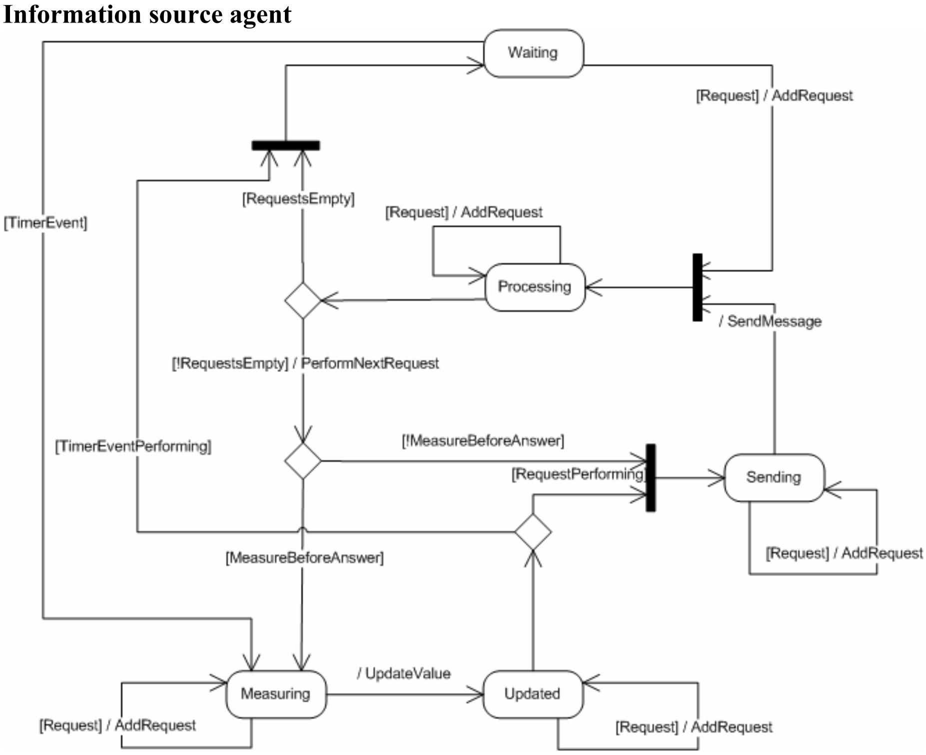

Fig. 3 The States of an Information Source Agent

The information source agent is connected either with an environmental (physical) sensor or another information source agent. The goal of the agent is to observe the parameter and to inform other agents about changes of the value. As shown in Fig. 3, an information source agent waits for external requests from other components or for timer events triggering the measurement of the observed parameter. In the former case, the incoming request is added to a list of requests to be processed, and as long as the list is not empty, the agent delivers its contained knowledge. If the agent was triggered by a timer event, or the agent is configured to measure the current value before answering a request, the agent updates its internal knowledge-base by obtaining the current value. If the agent measures the parameter directly, it reads the physical sensor value. If it requests the information from other entities, the "Measuring"-state branches out to an internal "Requesting"-state, the agent sends the requests and waits for the answer. When the agent realises that the value has changed, it fires an event that will be delivered to all agents registered as listeners to this attribute.

Conclusions and Future Work

In this paper we have illustrated our approach of applying agent technology for user modelling in ubiquitous computing. In contrast to resent approaches, we broke the one-to-one relationship between the user and her representing User-Agent. As true for sensor-networks we choose to have many small entities cooperating in ad-hoc networks on the different devices of a user. This allows for a flexible representation of the user by assembling the knowledge of all agents reachable in the current context. For releasing the need for a mobile-agent platform, we aim at information migrating between devices instead of Mobile Agents physically moving to an unknown platform.

In the current state of platform-specification and agent-modelling, the platform and the messages being sent between the agents are implemented in several projects of our institute. In the next steps, we will focus on the implementation of the agents based on the specification presented here.

References

[1] A. Jameson. Systems that adapt to their users: An integrative overview. In *Tutorial presented at 9th International Conference on User Modelling*, Johnstown, PA, USA, 2003.

[2] J. Fink and A. Kobsa. A review and analysis of commercial user modelling servers for personalization on the world wide web. *User Modelling and User-Adapted Interaction*, 10(2-3):209–249, 2000.

[3] J. Vassileva. Distributed User Modelling for Universal Information Access. In Stephanidis C. (ed.), *Universal Access in Human - Computer Interaction*, Lawrence Erlbaum: Mahwah, N.J., 122-126, 2001.

[4] M.J. Wooldridge. *Intelligent Agents II.* Springer-Verlag, 1996.

[5] M.J. Pazzani and D. Billsus. Adaptive web site agents. *Autonomous Agents and Multi-Agent Systems*, 5(2):205–218, 2002.

[6] D. Billsus and M.J. Pazzani. User modelling for adaptive news access. *User Modelling and User-Adapted Interaction*, 10(2):147–180, 2000.

[7] F. Menczer and R.K. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning*, 39(2/3):203–242, 2000.

[8] J.I. Vassileva, J.E. Greer, and G.I. McCalla. Openness and disclosure in multi-agent learner models. In *Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling*, Le Mans, France, 1999.

[9] J. Vassileva, G. McCalla, and J. Greer. Multi-Agent Multi-User Modelling. *User Modelling and User Adapted Interaction*, 13(1):1-31.

[10] C. Webber, L. Bergia, S. Pesty, and N. Balacheff. The baghera project: a multi-agent architecture for human learning. In J.I. Vassileva, ed., *Workshop on Multi-Agent Architectures for Distributed Learning Environments*, pp. 12–17, San Antonio, USA, 2001.
[11] B. Goodman, J. Hitzeman, F. Linton, and H. Ross. Towards intelligent agents for collaborative learning: Recognizing the roles of dialogue participant. In P. Brusilovsky, A. Corbett, and F. de Rosis, editors, *9th International Conference on User Modelling*, Johnstown, PA, USA, 2003.
[12] J. Fink and A. Kobsa. User modelling for personalized city tours. *Artificial Intelligence Review*, 18(1):33–74, 2002.
[13] FIPA Foundation for Intelligent Physical Agents. http://www.fipa.org.
[14] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf. CRUMPET: Creation of user-friendly mobile services personalised for tourism. In *2nd International Conference on 3G Mobile Communication Technologies*, pages 26–29, London, UK, 2001.
[15] M. Satyanarayanan, Of Smart Dust and Brilliant Rocks. *Pervasive Computing*, 2(4):2-4, 2003.
[16] D.E. Culler and H. Mulder. Smart Sensors to Network the World. *Scientific American*, 290(6):52-59, 2004.
[17] J. Odell. Objects and agents compared. *Journal of Object Technology*, 1(1):41–53, 2002.
[18] A. Lorenz and J. Denzinger. Functional agent systems for user modelling. *Proceedings of the Workshop on Artificial Intelligence, Information Access, and Mobile Computing*, 82–86, Acapulco, Mexico, 2003.
[19] A. Lorenz. Towards a new role of agent technology in user modelling. *Proceedings of the Workshop on Adaptivity and User Modelling in Interactive Systems*, Karlsruhe, Germany, 2003.
[20] A. Zimmermann, A. Lorenz, and M. Specht. User modelling in adaptive audio-augmented museum environments. In P. Brusilowsky, A. Corbett, and F. de Rosis (eds), *Proceedings of the 9th international conference on user modelling*, pp. 403–407, Johnstown, USA, Springer-Verlag, 2003.
[21] N.R. Jennings and M.J. Wooldridge. *Agent Technology*. Springer, 1998.
[22] C. Tessier, L. Chaudron, and H.-J. Müller. Conflicting Agents: Conflict management in multi-agent systems. Kluwer Academic Publishers , 2000.