An Agent-Based Framework for Context-Aware Services

Axel Bürkle[1], Wilmuth Müller[1], Uwe Pfirrmann[1], Nikolaos Dimakis[2], John Soldatos[2], and Lazaros Polymenakos[2]

[1] Fraunhofer Institute for Information and Data Processing, Karlsruhe, Germany
{Axel.Buerkle,Wilmuth.Mueller,Uwe.Pfirrmann}@iitb.fraunhofer.de
[2] Athens Information Technology, 19.5 Km Markopoulou Ave., Peania, GR 19002
{ndim,jsol,lcp}@ait.edu.gr

Abstract. A major challenge of Ambient Intelligence lies in building middleware that can ease service implementation through allowing the application developer to emphasize only the service logic. In this paper we describe the architecture of an Ambient Intelligence system established in the scope of the European research project CHIL (Computers in the Human Interaction Loop). CHIL aims at developing and realizing computer services that are delivered to humans in an implicit and unobtrusive way. The framework presented here supports the implementation of human-centric context-aware applications. This includes the presentation of the sensors used in CHIL spaces, the mechanisms employed for controlling sensors and actuating devices, as well as the perceptual components and the middleware approach for combining them in the scope of applications. Special focus lies on the design and implementation of an agent based framework that supports "pluggable" service logic in the sense that the service developer can concentrate on coding the service logic independently of the underlying middleware. Following the description of the framework, we elaborate on how it has been used to support two prototype context-aware human centric and non-obtrusive services.

1 Introduction

The emerging Ambient Intelligence (AmI) paradigm aims at the convergence of pervasive and intelligent systems, exploiting the full range of sensors, devices and networks available to transparently provide services, regardless of time and the location of the end user [1]. AmI services are essentially context-aware services that acquire and process information about their surrounding environment. Context-aware applications execute service logic not only based on input explicitly provided by end users, but also based on information that is implicitly derived [2]. Implicit information is usually derived based on a rich collection of casually accessible, often invisible sensors that are connected to a network structure. Sensors provide information streams, which are accordingly processed by middleware components towards deriving context cues. As a characteristic example, complex perceptive interfaces and recognition algorithms process audiovisual streams and extract context relating to the identity and location of people and objects. Having an initial set of context cues, information fusion algorithms are applied to derive more sophisticated context, leading to situation recognition. Context-acquisition can then trigger the service logic of an AmI service. The latter is likely to be composed

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2007, LNCS 4555, pp. 53–62, 2007.

of a rich set of invocations to soft-computing services, including commands to actuating devices, sensor control and regulation of the environment. Therefore, an infrastructure for Ambient Intelligence should at least include:

- A transparent sensing infrastructure, which is as non-intrusive as possible.
- An infrastructure for controlling sensors and actuating devices.
- A collection of perceptual components gaining elementary context cues from the various sensor streams.
- Information fusion components combining elementary context cues towards deriving more sophisticated context.
- An infrastructure for service provision.

These infrastructure and families of components can be seen as the common denominator of any non-trivial application. Building this common denominator infrastructure is a particularly challenging task. Indeed several major projects have allocated significant effort in building such hardware and middleware infrastructures [2][3][4][5][6]. As soon as these infrastructures are established, another challenge is to reuse these components to realize a variety of services. The challenge lies in building middleware that can ease service implementation through allowing the application developer to emphasize only the service logic rather than wondering about the integration with the underlying context-aware components. Such middleware is a key prerequisite for the wide adoption of Ambient Intelligence, given that it can significantly minimize the effort required to develop AmI applications.

In this paper we describe the sensing and middleware infrastructure established in the scope of the Computers in the Human Interaction Loop (CHIL) project [7] to support the implementation of human-centric context-aware applications for in-door environments. The paper is structured as follows: Section 2 gives an overview of the CHIL system architecture. Section 3 discusses the sensing infrastructure and context-aware middleware components established within CHIL. Section 4 presents the design of the agent-based framework that facilitates service access control and personalization, while also enabling service developers to plug service logic in a seamless fashion. Section 5 emphasizes two applications of the framework in the scope of two distinct services. Section 6 concludes the paper.

2 Architecture of the CHIL System

Due to the scale of the CHIL project with its large number of partners contributing to the CHIL system with a diversity of technical components such as services and perceptual components, as well as their complexity, a flexible architecture that facilitates integration of components at different levels is essential. A layered architecture model was found to best meet these requirements and allow a structured method for interfacing with sensors, integrating technology components, processing sensorial input and composing services as collections of basic services. Furthermore, the architecture described here supports flexible exchange of components and the replacement of unavailable components with simulators through well-defined interfaces.

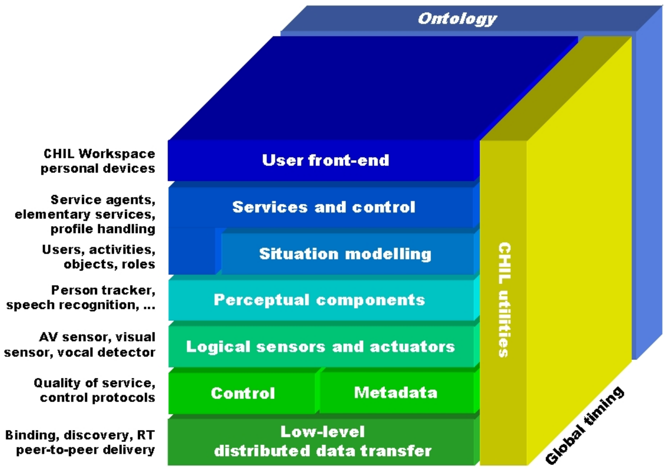

Fig. 1. CHIL architecture layer model

2.1 CHIL Layer Model

In order to realize proactive and intelligent services, both context-delivering and collaborating components have to be integrated. In our project, context is delivered both by perceptual components and learning modules. Perceptual components continuously track human activities, using all perception modalities available, and build static and dynamic models of the scene. Learning modules within the agents model the concepts and relations within the ontology. Collaboration is enabled by a set of intelligent software agents communicating on a semantic level with each other. The layered system architecture that facilitates the integration of different components is presented in Figure 1. The lower layers deal with the management and the interpretation of continuous streams of video and audio signals in terms of event detection and situation adaptation and contain the perceptual components. The upper layers of the infrastructure enable reasoning and management of a variety of services and user interfacing devices, based on agent communication. All layers use a common ontology as a backbone. A detailed description of the layers is given in the following section. The agent infrastructure of the CHIL system is described in more detail in section 4.

2.2 Description of the Layers

User Front-End: The *User Front-End* contains all user-related components such as the Personal Agents, Device Agents and the User Profile of a CHIL user. The Personal Agent acts as the user's personal assistant taking care of his demands. It interacts with its master through personal devices (e.g. notebook, PDA, smartphone) which are represented by corresponding Device Agents. The User Profile stores personal data and service-related preferences of a user.

***Services and Control*:** This layer comprises the service agents and their management. Services include reusable basic services as well as complex higher-level services composed of suitable basic services. The interaction with other agents within this layer and the *User Front-End* layer uses the communication mechanisms of the agent platform, while communication with the other layers uses internal mechanisms.

***Situation Modelling*:** This layer is a collection of abstractions representing the environmental context in which the user acts. An ontological knowledge-base maintains up-to-date state of objects (people, artifacts, situations) and their relationships. Additionally it serves as an "inference engine" that regularly deduces and generalizes facts during the process of updating the context models as a result of events coming from the underlying layer of *Perceptual Components*.

***Perceptual Components*:** The *Perceptual Components* layer provides interpretation of data streams coming from various audio and video sources represented by the underlying layer. *Perceptual Components*, such as for example speech recognition, process the raw data input and generate higher-level semantics events used by the *Situation Modelling* layer and/or by the *Services and Control* layer.

***Logical Sensors and Actuators*, *Control and Metadata*, *Low-level Distributed Data Transfer*:** These layers contain components which provide abstractions of the physical sensors and actuators, allow data annotation, storage, search and transfer of sensor data and control information.

***CHIL Utilities*:** This layer provides functionality that is needed by several or all components in the framework, e.g. global timing.

***Ontology*:** In order to enable the intended cognitive capabilities of the CHIL software environment, it is necessary to conceptualize entities and to formally describe relations among them. Through the ontology, CHIL software components both know the meaning of the data they are operating on and expose their functionality according to a common classification scheme.

3 Sensors Infrastructure and Context-Awareness

Sensing infrastructures are a key prerequisite for realizing context-aware AmI applications such as those developed, studied and evaluated within CHIL. Therefore, several sites within this project have constructed in-door environments comprising multi-sensor infrastructures with multiple microphones and cameras. These infrastructures are called 'smart rooms' and include at least:

- One 64 channel linear microphone array [8].
- Microphones for localization, grouped in inverse T-shaped clusters.
- Four fixed cameras, used for overall monitoring of the room.
- One active camera with pan, tilt and zoom (PTZ camera).
- A panoramic (or fish-eye) surveillance camera.

Based on these sensor infrastructures, a variety of perceptual components have been built and evaluated such as 2D (and 3D)-visual perceptual components, acoustic components, audio-visual components, as well as output perceptual components such as multimodal speech synthesis and targeted audio. Examples of such perceptual components which are operational in the AIT 'smart room' are:

- Acoustic identification and localization of the speaker [9].
- Person identification based on visual sensors [10].
- Detection and tracking of peoples' movements [11].
- Detection of speech activity [12].

3.1 Situation Recognition

Perceptual components derive elementary contextual information, and in general lack information about the overall current status of the people's interactions and environmental conditions. To be able to "fuse" many of these perceptual components together in order to determine more sophisticated and meaningful states, additional middleware interfaces have been developed to facilitate the intercommunication of these components, as well as higher layer agents which act as receptors of the whole range of the elementary context cues and process them in order to map the resulting contextual information into a complicated situation. This process is called situation recognition and is a major part of every human-centric Ambient Intelligence application.

Situation recognition, as developed in our implementation, follows the network of situation approach. This scheme allows the interconnection of distinct cases (situations), which are connected with arcs, forming a directed graph. A transition from one situation to another occurs if given constraints are satisfied. As soon as this transition is feasible, the service logic is applied, and the active state is reflected by the newly reached one. Context-awareness is hence modeled by this network. Figure 2 illustrates such a network of situations which can be used to track situations during meetings in a 'smart room'.

4 Agent Infrastructure

Ambient Intelligence services are usually based on complex heterogeneous distributed systems comprising sensors, actuators, perceptual components, as well as information fusion middleware. In projects like CHIL, where a number of service developers concentrate on radically different services, it is of high value that a framework ensures reusability in the scope of a range of services. To this end, we have devised a multi-agent framework that meets the following target objectives:

- Facilitates integration of diverse context-aware services developed by different service providers. Facilitates services in leveraging basic services (e.g. sensor and actuator control) available within the smart rooms.
- Allows augmentation and evolution of the underlying infrastructure independently of the services installed in the room.
- Controls user access to services.
- Supports service personalization through maintaining appropriate profiles.
- Enables discovery, involvement and collaboration of services.

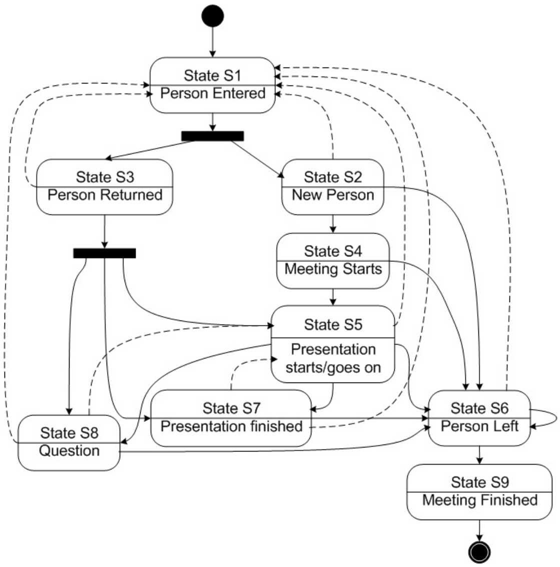

Fig. 2. An network of situations model for tracking meetings in a smart room

The CHIL software agent infrastructure, shown in Figure 3, is composed of three levels of agents. It covers the upper two layers of the CHIL architecture (cf. Figure 1). Agents and components close to the user form the level of User-Related Agents and are situated in the *User Front-End* layer, whereas the *Services and Control* layer contains the Basic Agents level providing elementary tasks and including a communication ontology as well as the level of specific Service Agents.

User-related Agents: Every person in the CHIL environment has his own Personal Agent acting as a personal secretary. Users interact with the system only via their self-adapting Personal Agents assigned during the login procedure. The Personal Agent manages interactions with its master through its associated Device Agents, whereby each device known to the system is represented by one Device Agent. It selects the input and output devices based on their availability and its master's context and preferences. Personal Agents and Device Agents act as the user interface of the different services provided by the Service Agents. Furthermore, the Personal Agent provides and controls access to its master's profile and preferences, thus ensuring user data privacy.

Basic Agents and Ontology: The main basic agents are the CHIL Agent and the CHIL Agent Manager. CHIL Agent is the basic abstract class for all agents used in the CHIL environment. It provides methods for agent administrative functionality, for registration and deregistration of the services provided by the agent instantiations, and additional supporting utility functions like creating and sending ontology-based messages, extracting message contents and logging. The CHIL Agent Manager is a central instance encapsulating and adding functionality to the JADE [13] Directory Facilitator (DF). Other CHIL agents register their services with the Agent Manager

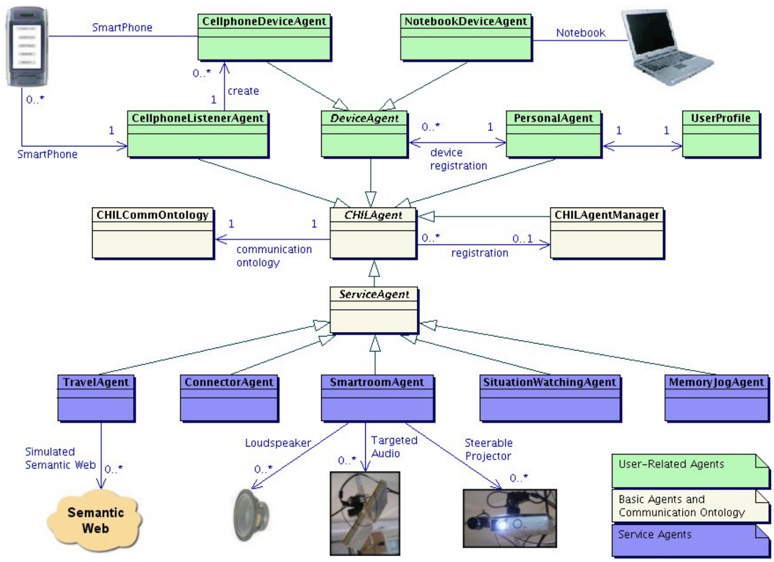

Fig. 3. Agent infrastructure

including required resources for carrying out these services. Service Agent is the abstract base class for all specific service agents.

Service Agents: Service Agents provide access to the service functionality. Based on a common ontology they map the syntactical level of the services to the semantical level of the agent community enabling the use of other Service Agents to supply their own service. The Memory Jog and the Connector Agent implement the functionality of the corresponding services. Both are described in detail in the next section. The Travel Agent can arrange and rearrange itineraries according to the user's profile and current situation. To gain information about flight and train schedules, it uses a simulated semantic web interface that gathers the required data from several online resources on the internet. The Smart Room Agent controls the service-to-user notification inside the smartroom using appropriate devices such as loudspeakers, steerable video projectors and targeted audio. The Situation Watching Agent provides access to the situation model.

5 Prototype Applications

We have exploited this architectural framework in implementing two different non-intrusive Ambient Intelligence applications, namely the 'Memory Jog' and the 'Connector'. These two applications demonstrate how this agent framework is indeed appropriate for developing context-aware applications. Specifically, implementations have shown that it is possible to use this framework to interface with situation modeling logic, as well as to invoke basic services. Nevertheless, demonstrating the full value of the framework, demands that two or more services are plugged into the same instance of the framework based on the same underlying infrastructure for sensing and context-awareness. The following paragraphs present briefly the current implementations.

5.1 The Memory Jog

The goal of the Memory Jog Service is to provide non-obtrusive information to boost productivity in the scope of meetings, lectures and presentations in 'smart rooms'. The current functionality of the Memory Jog focuses on tracking humans and their activities within a room, and accordingly providing pertinent information to participants. Information is provided both when users request it, but also automatically, based on context. Information pertains to people, as well as their activities in the scope of a meeting or lecture, such as the topics, keywords, agenda items.

The Memory Jog Service makes use of all underlying sensor and perceptual component infrastructure. In addition to this set of components which provide the input to the Memory Jog Service, additional software and hardware modalities act as the computer-human interfaces, such as a TTS (Text-to-Speech) service, or a targeted audio device. Moreover, situation recognition is featured and managed by the Situation Watching component. This component is wrapped as a Service Agent (i.e. Situation Watching Agent) and is accordingly integrated into the agent framework.

An agent implementing the service logic for the Memory Jog Service was developed by making use of all these infrastructure components. The agent was registered at the Agent Manager as a provider for the Memory Jog Service and the Agent Manager was extended by pluggable behaviors to integrate the agent into the system. The service logic includes:

- Display of peoples' identities and locations on an appropriate user interface.
- Tracking and displaying the meeting agenda.
- Automatic display of presentations on the participants' screens.
- Automatic e-mailing of presentations at the end of the meeting.
- Services invoked upon users' requests to play audio prompts, display messages to other participants, search a database based on the current contextual state.

The information provided by the Memory Jog is delivered to the user via his Personal Agent which invokes the appropriate Device Agent based on the user's preferences and context. The Device Agent filters the gathered information according to device limitations such as screen size and processing power, factors which should be taken under consideration in the case of e.g. a PDA.

5.2 The Connector

The goal of the Connector Service is to facilitate human communication during events that occur in in-door environments. It manages intelligent communication links (connections and notifications) and handles connection and notification requests, either simple requests (e.g. a meeting participant wants to talk to another) or more complex requests (e.g. a meeting participant wants to notify all other participants). The corresponding Connector Agent mediates between various Personal Agents and handles communications affecting two or more Personal Agents (Personal Agents may communicate with each other directly if there is only a direct communication between two Personal Agents). The Connector Service stores all pending connections and finds a suitable point of time for these connections to take place.

In order to describe the complex behavior of the Connector Service, we consider a sample scenario described in detail in [14]. In this scenario a meeting is scheduled in a CHIL smart room, where most of the meeting participants are already present in the meeting room. Another participant (Jeff) realizes that he will be late for the meeting and wants to inform the other participants about his delay.

Jeff uses his smartphone to trigger his Personal Agent (via the smartphone's Device Agent) which informs the Agent Manager of the delay. The Agent Manager asks the Meeting Agent for the meeting room number and a list of all meeting participants to inform them in the most efficient way. With this information the Agent Manager requests the responsible Connector Agent to deliver Jeff's message in an appropriate manner. After contacting the situation model via the Situation Watching Agent, the Connector Agent (being aware of a list of participants already in the meeting room), decides to notify them via the Smart Room Agent which is aware of the output devices available in the meeting room. The Smart Room Agent chooses to display the delay message with the central video projector. For those participants who are not yet in the meeting room, the Connector Agent informs their Personal Agents which in turn may choose an appropriate output medium for their masters (e.g. notebook, PDA, mobile phone, etc.) according to their current situation and user profile. Finally Jeff is informed again by his Personal Agent that his message has been delivered to all meeting participants.

Please note that the scenario described in [14] covers further functionality of the Connector Service and involves additional devices and agents. A detailed illustration of the complete scenario would be beyond the scope of this paper.

6 Conclusion

Common constituents of sophisticated Ambient Intelligence services are: a non-intrusive sensing infrastructure, infrastructures for controlling sensors and actuating devices, collections of perceptual components gaining elementary context cues from the various sensor streams, as well as a rich set of information fusion components combining elementary context cues towards more sophisticated context. Having these components at hand, a major research challenge is to reuse these components to realize a variety of Ambient Intelligence services. The challenge lies in building middleware that can ease service implementation through allowing the application developer to emphasize only the service logic, rather than wondering about integration with underlying components. In this paper we have presented an architecture of an Ambient Intelligence system with special focus on a distributed agent framework allowing developers of different services to concentrate on their service logic, while exploiting existing infrastructures for perceptual processing, information fusion, sensors and actuators control. The core concept of this framework is to decouple service logic from context-aware and sensor/actuator control middleware. Hence, service logic can be plugged in a specific placeholder based on well defined interfaces. The agent framework has been implemented based on the JADE environment, and accordingly instantiated within two real life smart rooms comprising a wide range of sensors and context-aware middleware components. The benefits of this framework have been manifested in the development of two radically different applications, one providing memory support during meetings and conferences, and another facilitating human communication.

Acknowledgements

This work is part of the project CHIL, an Integrated Project (IP 506909), partially funded by the European Commission under the Information Society Technology (IST) program. The authors acknowledge valuable help and contributions from all partners of the project.

References

1. Aarts, E., Marzano, S., (eds.): The New Everyday: Views on Ambient Intelligence. 010 Publishers, Rotterdam, The Netherlands (2003)
2. Dey, A.K., Salber, D., Abowd, G.D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16 (2001)
3. Coen, M., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., Finin, P.: Meeting the Computational Needs of Intelligent Environments: The Metaglue System. In: 1st Int. Workshop on Managing Interactions in Smart Environments, Dublin, Ireland (1999)
4. Johanson, B., Fox, A.: The Event Heap: An Coordination Infrastructure for Interactive-Workspaces. In: 4th IEEE Workshop Mobile Computer Systems and Applications (WMCSA 2002), Piscataway, N.J, IEEE Press, New York (2002)
5. Judd, G., Steenkiste, P.: Providing Contextual Information to Pervasive Computing Applications. In: IEEE Int. Conference on Pervasive Computing (Percom 2003), Dallas, USA (2003)
6. Saif, U., Pham, H., Paluska, J.M., Waterman, J., Terman, C., Ward, S.: A Case for Goal-oriented Programming Semantics. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, Springer, Heidelberg (2003)
7. CHIL - Computers in the Human Interaction Loop (2007), `http://chil.server.de`
8. Brandstein, M., Ward, D.: Microphone Arrays: Techniques and Applications. Springer, Heidelberg (2001)
9. Constantinides, A., Polymenakos, L., Talantzis, F.: Estimation of Direction of Arrival Using Information Theory. IEEE Computer Society Press, Los Alamitos (2005)
10. Pnevmatikakis, A., Polymenakos, L.: An automatic face detection and recognition system for video streams. In: 2nd Joint Workshop on Multi-Modal Interaction and Related Machine Learning Algorithms, Edinburgh (2005)
11. Pnevmatikakis, A., Polymenakos, L.: Kalman tracking with target feedback on adaptive background learning. In: Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (2005)
12. Rentzeperis, E., Stergiou, A., Boukis, C., Souretis, G., Pnevmatikakis, A., Polymenakos, L.: An adaptive speech activity detector based on signal energy and lda. In: 2nd Joint Workshop on Multi-Modal Interaction and Related Machine Learning Algorithms (2005)
13. Java Agent DEvelopent Framework (2007), `http://jade.tilab.com`
14. Danninger, M., Flaherty, G., Bernardin, K., Ekenel, H., Köhler, T., Malkin, R., Stiefelhagen, R., Waibel, A.: The connector: facilitating context-aware communication. In: 7th Int. Conference on Multimodal Interfaces (ICMI'05), ACM Press, New York (2005)