Building Scalable Services: The CHIL Agent Framework

Axel Bürkle¹, Nikolaos Dimakis², Ruth Karl¹, Wilmuth Müller¹, Uwe Pfirrmann¹, Manfred Schenk¹, Gerhard Sutschet¹

¹ Fraunhofer Institut für Informations- und Datenverarbeitung, IITB, Karlsruhe, Germany
² Athens Information Technology, Peania, Attiki, Greece

The services realized within the CHIL project are implemented by a set of collaborative software agents communicating with each other on a semantic level. In order to ensure this collaboration as well as a scalable service composition, coordination, and configuration, an agent framework and infrastructure was designed. A special feature of the CHIL agent infrastructure is the “pluggable behaviors” mechanism. This concept allows implementing service-specific code in agent behaviors, which will be plugged into the agents. It keeps the agent free from service functionality and enables a service-oriented scalable configuration. Service-specific communication ontologies can be plugged into the system without recompiling the source code. Furthermore, the autonomy feature of the CHIL architecture facilitates self-healing and restarting of agents, both in a stateless mode and in a stateful mode, while a directory service leverages a knowledge base that services requests and handles registration of any component in the architectural framework.

28.1 The CHIL Agent Infrastructure

Ubiquitous services are usually based on complex heterogeneous distributed systems comprising sensors, actuators, perceptual components, and 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 multiagent 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 independent of the services installed in the room;
- controls user access to services;
• supports service personalization by maintaining appropriate profiles;
• enables discovery, involvement, and collaboration of services.

28.1.1 Software Agents

An agent is a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives.

This definition by Jennings et al. [6] emphasizes three key concepts – situatedness, flexibility, and autonomy – which perfectly meet the main requirement for the CHIL architecture: to support the integration and cooperation of autonomous, context-aware services in a heterogeneous environment. Summarized, the major goal for the infrastructure incorporates the discovery, involvement, and collaboration of services as well as competition between services in order to perform a certain task the best way possible. Standardized communication mechanisms and protocols have to be considered to raise information exchange onto a semantic level and to ensure location transparency.

The following sections describe the CHIL agent infrastructure and how we achieved these objectives. Moreover, they demonstrate how we realized a multiagent system that is capable to “solve problems that are beyond the individual capabilities or knowledge of each problem solver” [6].

28.1.2 Agent Description

The CHIL software agent infrastructure covers the upper two layers of the CHIL architecture. Agents and components close to the user are situated in the user front-end layer, e.g., the user’s personal agent and the device agents, whereas the services and control layer contains the basic agents and communication as well as the service agents. Figure 28.1 shows an excerpt of the agent infrastructure.

Basic Agents

CHIL agent: CHIL agent is the basic abstract class for all agents used in the CHIL environment. It provides methods for basic agent administrative functionality (setup, takedown), directory facilitator service (DF Service) functions (register/deregister agents, modify agent descriptions, search service-providing agents based on a semantic service description ontology), and additional supporting utility functions for creating and sending messages, extracting message contents, and logging. Special importance is attached to keep the agent communication conforming to FIPA (Foundation for Intelligent Physical Agents) standards [3], i.e., to comply with the FIPA Interaction Protocols and the FIPA Communicative Acts. The message transfer is based on a well-defined communication ontology including agent actions, content concepts, and predicates.