Ubiquitous Healthcare: The OnkoNet Mobile Agents Architecture

Kirn S
stefan.kirn@tu-ilmenau.de
Germany

Abstract: This paper introduces ubiquitous healthcare, addressing the access of health services by individual consumers applying to mobile computing devices. This access requires knowledge about the individual health status, which may involve (as far as available) the current personal situation (e.g., accident), relevant recent diseases etc., current symptoms or already available diagnosis. Addressing the related IT challenges - e.g., patient-centric knowledge processing, situation awareness, local control - we develop in detail the OnkoNet Mobile Agents Architecture involving architectures on the macrolevel and microlevel as well as cooperation protocols, inference models controlling system behaviour, and a health ontology.

Keywords: cooperative problem solving, mobile agents, multiagent systems, OnkoNet, ubiquitous healthcare

1 Introduction

In recent years, healthcare organisations worldwide have undergone major reorganisations to meet the demands of improved health services. Part of this is the management, and use of rapidly growing quantities of health data. Much of that information is already available in digital form, and lots of efforts are addressing the problems of system and information integration, resulting from heterogeneity in all aspects [SL90], e.g., a broad variety of formats, unsatisfying standardisation of data models (e.g., Dicom3), proprietary software architectures and interfaces, etc. As a result, an increasing number of health services (and the respective processes to produce, coordinate, deliver, and consume them) are subject to extensive digitalisation. The most impressive vision of digital healthcare is that one of the ”teleportal hospital” [Mün02].

However, the emerging technology of mobile computing leads already to new challenges. Health is a fully individualised concern, and is therefore inherently mobile. Whereas today we generally visit a health specialist in case of indisposition, in future we may use mobile devices to access health services at any time and any place, just depending on our current demand. Examples are car or sports-accidents, health communities [Cos02], home care, and diseases which run over long periods of time like allergies, asthma, diabetes, cancer, etc.

In this context the paper introduces the concept of ubiquitous healthcare, being the next step beyond web-enabled healthcare computing [Ede00]. Ubiquitous healthcare refers to the disposition of any type of health services such that individual consumers through mobile computing devices can access them. The (automatic/semi-automatic) access necessarily (to reduce complexity, to avoid information overflow) involves knowledge about the health status of the respective person. This knowledge may involve (as far as available) the current personal situation (e.g., car accident), possibly relevant past diseases/operations/therapies, current symptoms or already available diagnosis, etc. Some of that knowledge may be provided by the person herself (e.g., through mobile devices), other knowledge may be accessible in databases at different locations (e.g., hospitals, family doctors, specialists, pharmacy, hospitals) and even other knowledge may be generated as a result of human or machine-based search and/or problem solving. Further, ubiquitous healthcare is supposed to offer competitive advantages to health service providers, in particular through numerous new opportunities for market differentiation, increase of service process efficiency, of professional specialisation, of service quality, and improvements in patient relationship management.

In order to meet the IT-related challenges arising from ubiquitous healthcare, this paper develops a mobile agents architecture supporting knowledge intensive cooperation among humans and software agents involved in producing, delivering, controlling, and consuming health services.

Organisation of the paper: Section 2 introduces to the reader OnkoNet Thuringia, a virtual organisation supporting the cooperation of all actors involved in cancer diagnosis, and therapy (patients and their families, family doctors, nurses, cancer specialists, etc.). A major task within this initiative is to efficiently support communication, coordination, and collaboration among the actors. Section 3 outlines the conceptual approach we have applied in order to achieve strong patient-orientation, and situation awareness in cooperative knowledge processing. Section 4 presents system design and implementation, with a particular focus on the multiagent system schema, the OnkoNet mobile agents architecture(s), the cooperation protocol together with several multiagent inference models, and the OntHoS-Ontology. Section 5 reports the project status, and outlines our plans for future work. Section 6 contains the references.

2 OnkoNet Thuringia

2.1 Overview

The current focus of healthcare information management is still on patient records, especially on information about clinical courses - often on different levels of abstraction. Local perspectives are dominating, neglecting the interrelations among different actors (patient, family doctors, different specialists, etc.), and different institutions, each with their own individual patient-/disease-related knowledge. Another problem concerns the typically insufficient participation of patients in treatment-related information management and infor-

Figure 1: OnkoNet Thuringia

mation processing. These shortcomings cause (often unnecessary) limitations in treatment quality, they may lead to enhanced workload of medical professionals, and they definitely increase healthcare budgets.

In order to approach these problems, in 1998 OnkoNet Thuringia was founded as a virtual healthcare organisation in oncology. The aims of the OnkoNet initiative are

- to develop, and to provide a WWW-based information platform for cancer treatments,
- with local, and mobile access for authorised persons/institutions storing, maintaining, retrieving, and accessing patient-/disease-related data, and information
- in order to achieve a much better integration of patients, and of medical professionals in all information-related activities during diagnosis, therapy, and care.

An overview of the current participants of OnkoNet Thuringia is depicted in fig. 1. Because of the extremely high degree of specialisation in oncology, any successful system development requires a sound understanding of the concrete problem(s) to be addressed, of the possible role(s) and risk(s) of IT in the environment considered, and of the relevant success factors from the – generally very different – perspectives of patients, nurses, doctors, and hospital management. OnkoNet Thuringia is thus an almost ideal empirical setting to study these and all related issues, and to provide us with all information required for system design, implementation, validation and test.

2.2 System Requirements

An important part of OnkoNet Thuringia is the development of an appropriate IT infrastructure supporting all types of individual as well as collaborative communication-related and information-related activities of humans as well as of software agents. In this section we summarise the most important results of a requirements engineering study in 2000. This study has identified the following major system requirements:

1. Health service providers:
   - two-phase production of health services requiring the explicit (often: active) integration of the patient as so-called ”external” factor of production,
   - process management support,
   - re-integration of small pieces of processes (”process fractals”) to complete processes
   - activity and process coordination,
   - identification, and elimination of institutional, organisational, media related breaks,
   - thoroughly consistent, integrated, adequate and reliable information dissemination,
   - patient-centric knowledge processing
2. Health service customers (patients):
   - Demand for health services is inherently individualised,
   - demand for health services is inherently mobile (at work, at home, sports, leisure),
   - related demand remains to be analysed, quantified – and satisfied,
   - single patients can easily be group by their diseases (or disease profiles) which makes it relatively easy to estimate the market volume.
3. Relevance of mobile computing:
   - potentially high to very high numbers of users,
   - in more or less all areas of private and work life,
   - on all levels of public, and private health systems,
   - to all parts of healthcare value chains,
   - key factor for customer contact / patient relationship management,
   - broad variety of mobile devices (which often require small bandwiths only) provides for a lot of different mobile healthcare services, and
   - relevance for B2B commerce in whole healthcare sector.

3 Conceptual Approach

3.1 Agent Technology

Software agents (in short: agents) are an innovative technology for the efficient realisation of complex, distributed, and heterogeneous information systems. They are software components with the ability to act on their own (autonomy), to process external events (reactivity) as well as to pursue own goals (pro-activity), to adapt their behaviour to changing demands/internal and external parameters (learning, adaptivity), to cooperate with other agents if necessary to fulfil their tasks (creating multiagent systems), and to move from one place in a network to another (migration) [Wei99, OJ96].

Agents can be specialised for particular tasks. An important example are user agents. They can learn user profiles and preferences from their interactions with their users, and they are able to adapt this knowledge if their users' tasks and preferences change over time.

Agents are created and "living" on so-called agent platforms. These are specialised middleware systems providing the agents with basic services, e.g. communication channels/protocols, white and yellow pages, etc.

Commonly accepted models for agents decompose their architecture into communicator, head, and body [SHM91]. The agent body constitutes the internal problem solving expertise and provides the basic functionality of the agent (e.g., hospital information system, pharmaceutical database, calendar or email system of a patient). It may be independent of any multiagent system, as a standalone application, or participate in different systems. The agent body is wrapped through an API controlling its behaviour. The agent head is that component which enables the agent to participate in cooperation processes with other agents, or humans. To this purpose it is connected to the agent body on the one side, and to the communicator on the other side. The agent communicator provides basic communication services, e.g. converting of logical in physical addresses, delivering of messages on behalf of the agent head through appropriate channels to the receivers. Furthermore, the communicator maintains lists of incoming and outgoing messages and continuously informs the agent head about the status of any communication the agent is involved in.

Messages are highly structured, and must apply to an accepted agent communication language such as FIPA ACL as well as to an agreed ontology [Fip02]. Incoming/outgoing messages are processed through communication protocols, according to the relations of agents, to their tasks or intentions, or due to any priority rules.

Responsible for the standardisation of agent technology is the Foundation of Physical Agents (FIPA), founded in 1996. The emphasis of FIPA is on the practical commercial and industrial uses of agent systems. Besides the FIPA Abstract Architecture Specification there are standards (or standardisation processes) for Agents Message Transport, Agent Management, Agent Communication, Agent Application Specifications, and Agent Security.

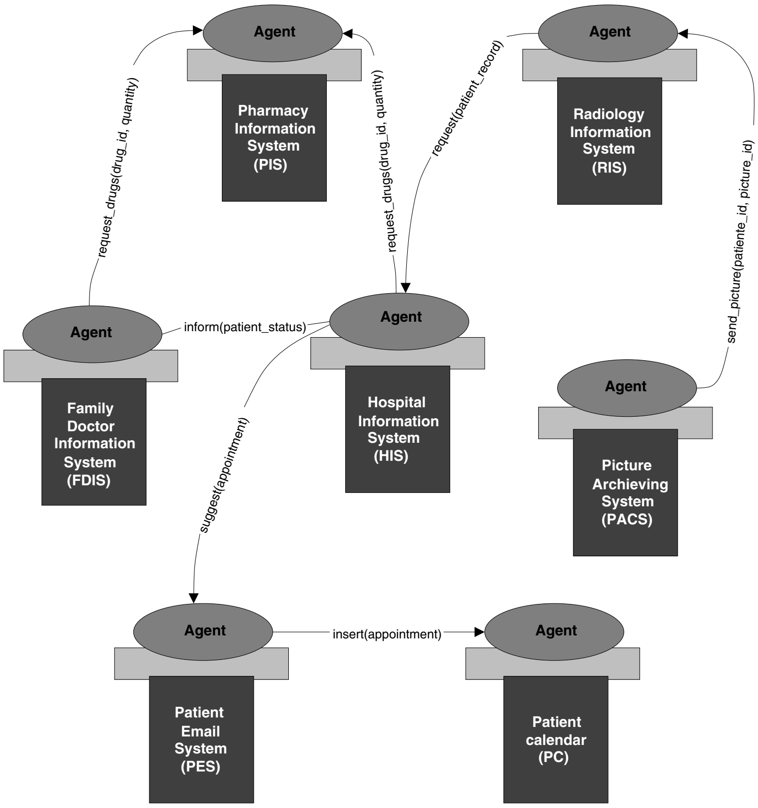

Figure 2: Agent-based cooperation support with wrapping of local knowledge sources

3.2 Transforming Local Systems into Cooperative Problem Solvers

Fig. 2 describes how different local information sources are wrapped and integrated with agent heads (here: including communicators) in order to communicate with each other and to perform cooperative problem solving, if necessary (e.g., "suggest(appointment)"). This architecture allows preserving the local autonomy of each information source even if it agrees to cooperate with other information sources in a multiagent system.

3.3 LEAP – Lightweight Extensible Agent Platform

As development tool and as agent management platform we are using LEAP, which has been developed as part of the European project Lightweight Extensible Agent Platform.

LEAP emerged as an independent development branch of JADE under the LGPL license

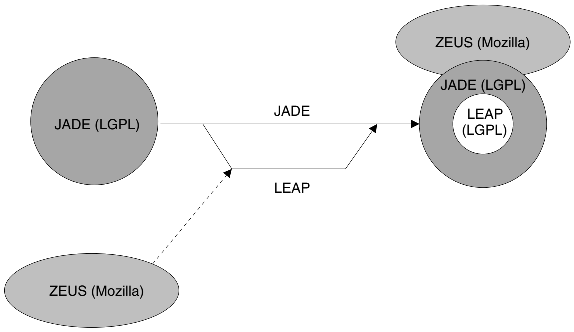

Figure 3: LEAP development process[BB01]

[Gnu02]. It is planned that LEAP shall replace JADE's kernel (fig. 3). As JADE continues its evolution towards environmental functions such as monitoring facilities, visualisation, ontologies, and policies, LEAP concentrates on lightweight and extensible aspects. In parallel the project LEAP is also developing the agent application development tool ZEUS under the Mozilla license. Detailed descriptions are available in [BB01, Jad02, Lea02, Age02]. LEAP is an integrated agent platform in that it is capable of generating and executing agent applications. LEAP is lightweight enough to be deployed on small (mobile) devices using Java 2 Micro Edition/Connected, Limited Device Configuration (J2ME/CLDC), instead of standard JAVA (Java). It thus supports running of agent applications implemented over a large family of mobile devices (e.g., laptops, PDAs, mobile phones, etc.) and communication mechanisms (TCP/IP, WAP, etc.). It is transport layer independent. In particular it supports transport protocols suitable for both wireline and wireless communication. Finally, it is easily extensible such that, when deployed on a PC it can provide additional functionality such as agent mobility support, user-defined ontology, and platform management tools.

4 Implementation

4.1 Technical Approach

The technical approach chosen to meet the system requirements summarised above exhibits the following characteristics:

- local and global problem solving is separated through wrapping of local systems by APIs, and through applying to an agent-based software architecture
- multiagent systems are dynamically created/dissolved referring to the paradigm of ad-hoc networks [TWLC01]
- multiagent systems establish partially open problem solving spaces,

- inference models provide services for adapting multiagent systems (including services for agents to join/leave the global system) as well as for patient-centric knowledge identification, knowledge processing and knowledge presentation,
- consequent individualisation through agent technology, and cooperative problem solving (push and pull strategies).

Further, the approach applies to a four-tier-model for agent definitions:

- *layer 0:* each user owns $m \geq 1$ user agents. Each user agent controls $n_0 \geq 1$ processes.
- *layer 1:* each business process owns exactly 1 process agent. Business processes are either atomic or complex. Complex business processes are composed of several atomic or complex processes. Each process agent accesses $n_1 \geq 1$ business functions.
- *layer 2:* each business function (such as "`send_a_letter`") owns exactly 1 function control agent implementing exactly one atomic business process. Further, each function control agent accesses $n_2 \geq 1$ resources,
- *layer 3:* each resource owns exactly 1 resource agent. Each resource agent controls exactly $n_3 = 1$ resource.

System design is following the FIPA compliance rules. In particular we are applying to the FIPA application specification definition, the FIPA ACL, and the FIPA Abstract Agent Definition specification. As already mentioned, the development and execution of agents are performed through LEAP.

4.2 Multiagent System Schemas

The general multiagent system schema developed assumes, that the cooperation begins when an agent (called: InitiatorAgent) on a mobile device (e.g., PDA) receives a user request (see fig. 4). The agent then migrates to the "main container" (stationary agent platform, e.g., running on a workstation) to contact DBAgents, each controlling/accessing its own local data/knowledge base. Next, the agent negotiates its demand with these DBAgents in order to decide who of them shall query its local database. A detailed description of the underlying cooperation protocol is given in section 4.4. In the current implementation of LEAP only agents running on PDAs with pJava (e.g., Windows CE) are able to migrate. For MIDP devices (e.g., J2ME) we thus developed a different multiagent schema (fig. 5). In this case, the agent needs to send its request via standard wireless communication to a DBBrokerAgent accessing a Directory Facilitator (DF) who identifies the appropriate database together with all parameters necessary to perform a query. The result is given back to the InitiatorAgent, which then accesses the DBAgent, which has been suggested.

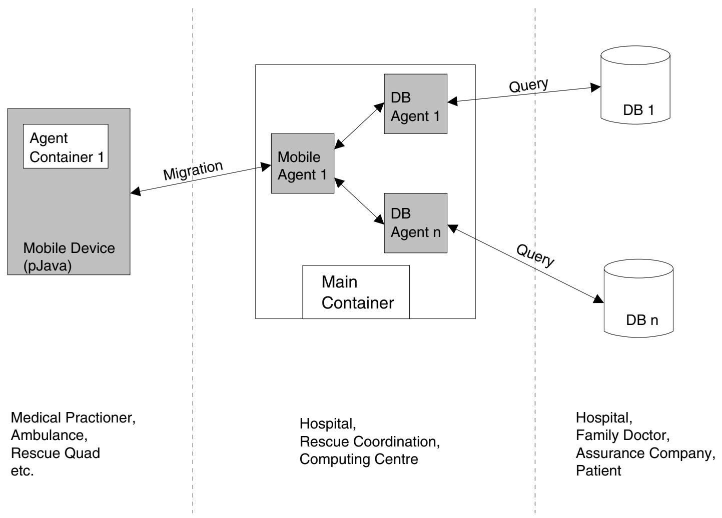

Figure 4: Multiagent system schema I (pJava device)

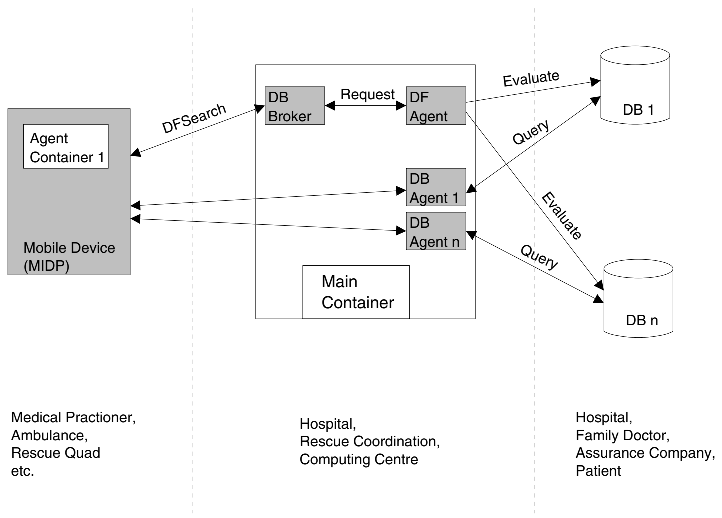

Figure 5: Multiagent system schema II (MIDP device)

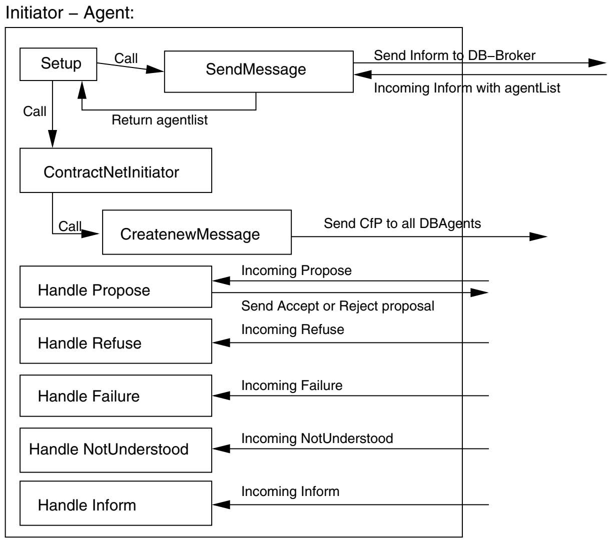

Figure 6: InitiatorAgent

4.3 OnkoNet Mobile Agents Architectures

According to the two multiagent systems schemas there are three agent architectures: the InitiatorAgent (located on the mobile device), the DBAgent, and the DBBrokerAgent (each located on main container). Their architectures are depicted in fig. 6-8. For details see [Gro02].

4.4 Cooperation Protocols

Cooperation among the agents is controlled through bilateral negotiation protocols. The top-level protocol is depicted in fig. 9. It controls the selection of a local database stating to believe that it would be able fulfilling an announced query through the DBBrokerAgent, and the subsequent data access of the InitiatorAgent via the responsible DBAgent. In order to be able to really execute the multiagent system further cooperation protocols and multiagent system level inference mechanisms have been developed. E.g., there are inference models for agents to join/leave the multiagent system, to adapt the macro structure of the system to changing demands and environmental parameters. Other inference models control the emergence, the access, and the dissemination of organisational knowledge on the multiagent system level, and micro-level as well as macro-level learning. These have been published in [KG98].

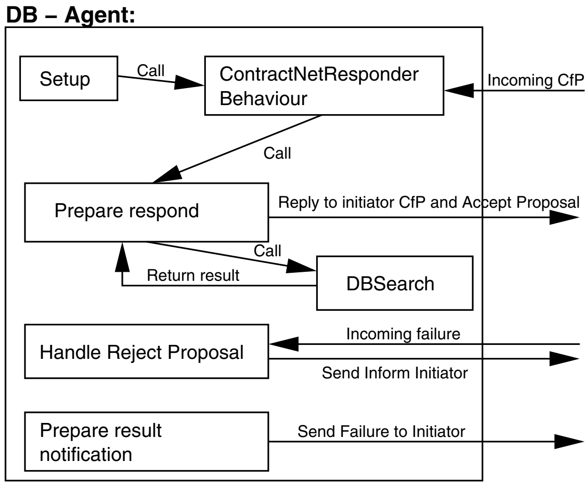

Figure 7: DBAgent

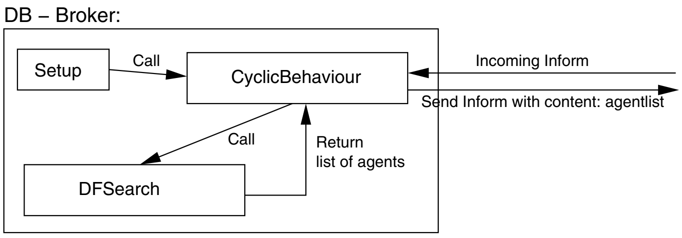

Figure 8: DBBrokerAgent

4.5 OntHoS - An "Ontology for Hospital Scenarios"

OntHoS is an ontology collection terms and definitions to represent organisational structures, tasks, and processes in hospitals. It is emerging from a national priority research program on intelligent agents and business applications, funded by the German Research Foundation (DFG) from 2000-2006 [SPP02]. OntHoS is being developed with Protégé-2000 [Pro02]. It will be available as open source probably in early 2003.

5 Summary

This paper introduced the concept of ubiquitous healthcare, and developed towards an agent-based architecture appropriately supporting the any-time/any-place access of health services via mobile computing devices.

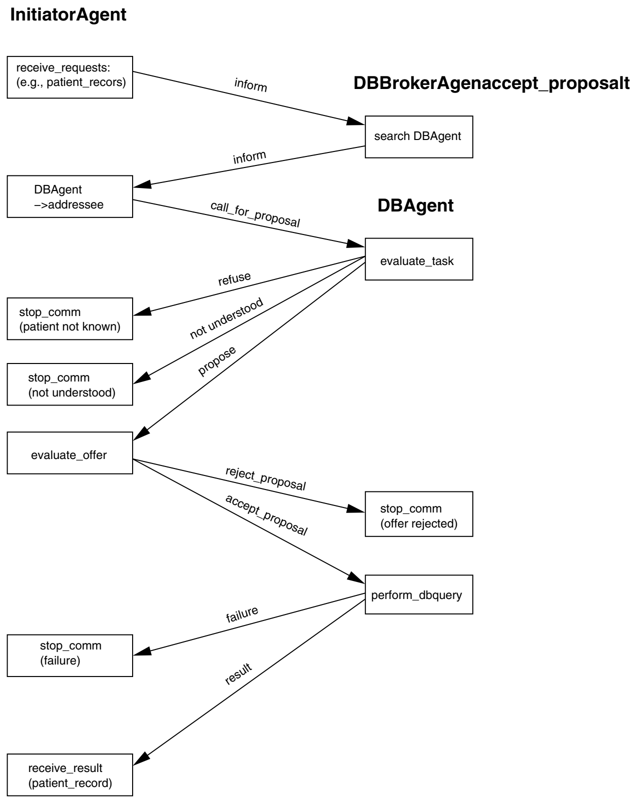

Figure 9: Selected Cooperation Protocol

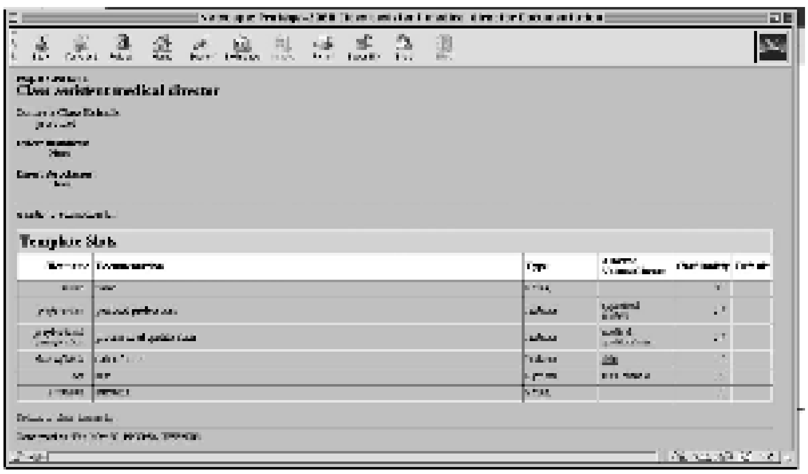

Figure 10: Class assistent medical director (part of the health ontology)

The work presented here emerged from an ongoing project covering all relevant issues from empirical process studies in cancer diagnosis/therapy down to system implementation, test, and validation. Due to space limitations we concentrated our presentation on the framework as such, and on the results already achieved.

However, besides the empirical studies suffering from several still unsolved methodological problems, the most interesting part in our research concerns the development and evaluation of patient-centric inference models for cooperative problem solving on the macrolevel of the multiagent system. Such strategies are required to avoid information overflow, and they provide for reducing search complexity in combined search spaces down to a computational tractable level.

Acknowledgements

Part of this work has been supported by the German Research Foundation under contract KI-720/3-1. Special thanks go to those members of the above mentioned German priority research program (SPP 1083) who are currently developing the OntHoS ontology.

Bibliography

[Age02] `http://www.labs.bt.com/projects/agents.htm`.

[BB01] M Berger and B Bauer. LEAP - A Scalable Agent Platform Enabling Next Generation Distributed Information Systems. In S Jablonski, St Kirn, M Plaha, EJ Sinz, A Ulbrich-vom Ende, and G Weiš, editors, *Verteilte Informationssysteme auf der Grundlage von Objekten, Komponenten und Agenten. Proceedings VertIS 2001*, Rundbrief der GI-Fachgruppe 2.5.2 und Rundbrief der GI-Fachgruppe 5.10 (8. Jahrgang, Heft 2), Oktober 2001, pages 191–206, Univ. Bamberg, 4./5. October 2001.

[Cos02] `http://www.cosmos-community.org`.

[Ede00] L Eder, editor. *Managing Healthcare Information Systems with Web-Enabled Technologies*. Idea group Publishing, 2000.

[Fip02] `http://www.fipa.org/`.

[Gnu02] `http://www.gnu.org/copyleft/lgpl.html`.

[Gro02] C Grož. Mobile Computing for Rescue Support - A sample Application based on the JADE-LEAP Agent Platform. Master's thesis, TU Ilmenau, June 2002. (in German).

[Jad02] `http://sharon.cselt.it/projects/jade/`.

[KG98] St Kirn and L Gasser. Organizational Approaches to Coordination in Multi-Agent Systems. *it + ti*, (4):23–29, 1998. (German journal, article in English).

[Lea02] `http://leap.crm-paris.com`.

[Mün02] E Münch. Teleportal-Klinik. Internal presentation, TU Ilmenau, 9. April 2002.

[OJ96] G O'Hare and N Jennings, editors. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons Inc., 1996.

[Pro02] http://protege.stanford.edu/index.html, 2002.

[SHM91] D Steiner, H Haugeneder, and D Mahling. *Cooperating Knowledge Based Systems*, chapter Collaboration of Knowledge Bases via Knowledge Based Coordination, pages 113–129. Springer-Verlag, 1991.

[SL90] AP Sheth and JA Larson. Federated Database Management Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, 09 1990.

[SPP02] http://www.wirtschaft.tu-ilmenau.de/wi/wi2/SPP-Agenten/index.html.

[TWLC01] Y-C Tseng, S-L Wu, C-M Liao, and C-M Chao. Location Awareness in ad Hoc Wireless Mobile Networks. *IEEE Computer*, pages 46–52, June 2001.

[Wei99] G Weiš, editor. *Multiagent Systems - A Modern Approach To Distributed Artificial Intelligence*. MIT Press, 1999.