The Data Abstraction Layer as Knowledge Provider for a Medical Multi-agent System

Montserrat Batet[1], Karina Gibert[2], and Aida Valls[1]

[1] ITAKA, Intelligent Technologies for Advanced Knowledge Acquisition, Dept. of Computer Engineering and Maths, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{montserrat.batet,aida.valls}@urv.cat
[2] Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Campus Nord, Ed. C5, c/ Jordi Girona 1-3, Barcelona, Catalonia, Spain
karina.gibert@upc.edu

Abstract. The care of senior patients requires a great amount of human and sanitary resources. The K4Care Project is developing a new European model to improve the home care assistance of these patients. This medical model will be supported by an intelligent platform. This platform has two main layers: a multi-agent system and a knowledge layer. In this paper, it is reviewed the initial design of the system, and some improvements are presented. The main contribution is the introduction of an intermediate layer between agents and knowledge: the Data Abstraction Layer. Using this additional layer agents can have a transparent access to many different knowledge sources, which have data stored in different languages. In addition, the new layer would make possible to make intelligent treatment of the queries in order to generate answers in a more effective and efficient way.

Keywords: knowledge representation, knowledge engineering, home care, multi-agent systems.

1 Introduction

The care of senior patients that suffer chronic diseases requires life long treatments under the continuous supervision of a group of people in charge of providing medical care. The European project K4CARE: "Knowledge-Based Homecare eServices for an Ageing Europe" will develop a distributed platform to improve the capabilities of the new EU society to manage a personalized Home Care assistance of the increasing number of senior population.

The main two objectives of K4Care project are: (1) to create a sanitary model at European Level for the care of senior patients that suffer chronic diseases or are disabled and need a treatment; (2) to design and implement an Information and Communication Technology (ICT) platform that supports this model. With respect to the first goal, it is tried to unify the way in which medical centers of different countries work, so that the exchange of information between centres will

D. Riaño (Ed.): K4CARE 2007, LNAI 4924, pp. 87–100, 2008.

become easier, as well as the transfer of patients from one place to another. With respect to the second goal, a Multi-Agent System that supports the different kinds of people involved in home care treatment (physicians, nurses, patients, etc.) will be implemented. An intelligent agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. Then, a multiagent system consists of a number of agents that cooperate with one another to solve a complex problem that could not be solved individually [1]. In the K4Care system, patients and professionals are the agents that should be able to coordinate their activities and also to share anytime and anywhere all the necessary health information (x-rays, analysis, prescriptions, etc.) in an integrated way.

In this context, the agents of the system will need to access the home-care model defined in the K4Care project [2] and to the Electronic Healthcare Record that stores data related to the patients, such as treatments, prescriptions or information about their health. These data and K4Care knowledge are distributed in different sources and encoded in different formats (for example, health information about the patients is in XML documents and diseases are defined in an ontology). This multiple language representation makes the communication between the agents and the information system difficult. In this paper, it is proposed and argued the use of an intermediate layer that helps to manage the communication, which is called Data Abstraction Layer [3]. This layer plays an integrating role between the Multi-Agent System and the knowledge sources. This layer is in charge of providing the data that the agents, facilitating to obtain and to store data in a transparent manner not depending on the localization and format of the data.

In the following sections it will be explained the process that lead us to design the Data Abstraction Layer. A real example extracted from the K4CARE project will be used to illustrate the construction of this intermediate layer and its use of the knowledge. This document is organized as follows: in section 2 the first version of the K4Care model is presented; it is devoted some special attention to the codification of procedures in section 3, since this is an important contribution derived from the revision of the first version of the model. Section 4 describes the new layer incorporated to the model, the Data Abstraction Layer, and its components. In section 5 it is presented a case study to explain the *Data Abstraction Layer* in detail. Finally, the section 6 presents some conclusions and future work.

2 K4Care Information Model

The first design of the K4Care model was composed, basically, by a platform of agents and some knowledge sources (see figure 1). In this model, two main layers are found: the first one, the K4Care platform, is divided in three components the Servlet, the Gateway agent and the Multi-Agent System. The second one is the Knowledge Layer, which stores all the medical knowledge about patients, diseases, and healthcare professionals.

The K4Care platform is a web-based application. The multi-agent system is composed by two kinds of agents: *actor agents* that represent practitioners and patients, and *execution agents* that are in charge of facilitating the execution of certain types of complex actions. Actors interact with the system through a web browser. The connection between the web-based application and the agents is made through a servlet and a set of gateway agents.

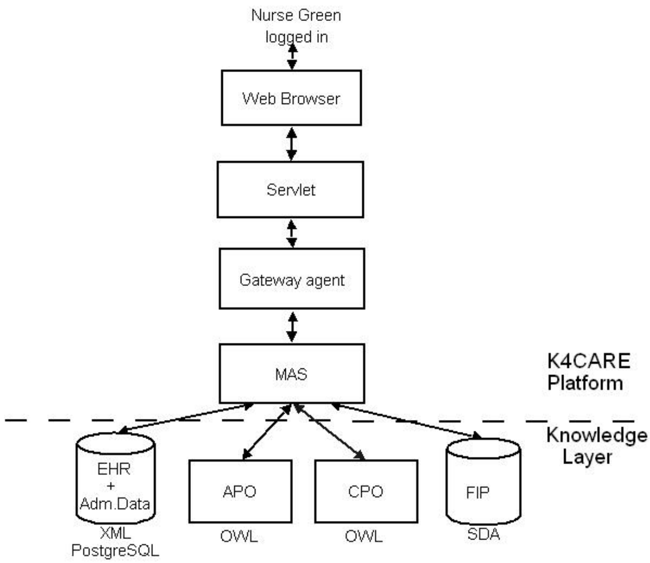

Fig. 1. K4care initial model

The knowledge in the Knowledge Layer is distributed in the following data sources:

– Ontologies help to build better and more interoperable information systems [4]. As it is defined in [5]: "An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group". In K4Care medical ontologies are mandatory. The communication of complex medical concepts is a crucial feature in medical information systems. In these systems there must be a medical terminology with a clear and non-confusing meaning [6]. Ontologies are widely used in medicine. Kumar et. al. [7] created a task ontology named Context-Task Ontology to represent the knowledge required to define clinical guidelines. Isern et al. [8] propose the inclusion of an especially designed ontology into an agent-based medical platform. The ontology represents medical terminology, models healthcare entities with its relations, and collects semantic categories of those medical concepts. Moreover, Davis and Blanco [9] used taxonomies to model the clinical lifecycle knowledge. In our case, a group of physicians developed two ontologies in order to reach an efficient way to store and communicate general medical knowledge and patient-related information:

- The Actor Profile Ontology (APO) stores the data of the actors involved into the K4Care home-care model [10]: healthcare professionals, patients and relatives, citizens and social organisms. They are organised in a hierarchical structure. Moreover, the APO also stores data about the services, actions, documents, and permissions of the different kinds of actors. K4Care provides services to the patients. The APO contains the services in which actors can take part, the procedures of those services (a non-ordered list of actions that conform a service) and the particular actions that they can perform. In addition, the ontology stores documents, and permissions of the different kinds of actors to read and/or write a document. The APO is represented by an ontology encoded in OWL [11].
- Case Profile Ontology (CPO) contains knowledge about some syndromes that will be considered in the K4Care prototype and cognitive impairment. All the information regarding these syndromes is organised in the CPO: symptoms, treatments, diseases, etc. The CPO is represented by an ontology encoded in OWL.

– Formal Intervention Plans (FIP) represent procedural knowledge about treatments of syndromes and diseases. Patients' treatments can be modelled using clinical guidelines. The US National Cancer Institute defines Clinical Guidelines as documents developed to help health care professionals and patients make decisions about screening, prevention, or treatment of a specific health condition. Clinical Guidelines contain a set of directions or principles to assist the health care practitioner with patient care decisions about appropriate diagnostic, therapeutic, or other clinical procedures for specific clinical circumstances [12]. FIPs store the guidelines to assist patients who suffer from particular ailments or diseases. Some of them have been obtained from the recommendations of the World Health Organization, others are being defined by the doctors' team that participate in the project. FIPs are represented in a graphical language called SDA* (States, Decisions, Actions) [13].
– The Electronic Health Record (EHR) and the administrative data. The information of particular users of the K4care system is stored in a database. On one hand, the database contains administrative information about the users (f.i. name, address, phone, login name). On the other hand, health care

information is stored in an EHR [14][15]. This EHR contains personal documents and general document schemas that health professionals manage during the patient assistance. It also contains Individual Intervention Plans (IIP). An IIP is a FIP adapted to the health particularities of a single patient. In the K4Care project, the Postgres database is used [16]. The Electronic Health Record is stored in form of XML documents [17].

As it can be seen, there are many different types of knowledge distributed into different knowledge sources and encoded in different formats. This causes that the agents must know the location of each information item, how it is codified and how to access to it. Therefore, to reach the information becomes a difficult and complicated task, mainly when the agents are due to make complex consultations in which information of different data sources takes part.

3 Procedure Codification

The home-care model defined in the K4Care project [2] is centred in two main concepts: actors and services. The actors represent the people involved in home care, and the services are those that professional actors perform to take care of the patients. Each service is structured into a procedure composed by a set of simple actions or calls to other services that must be done to successfully complete the service. Each action can be done by a subset of actors and uses a subset of documents. Initially, the procedure was represented as a simple list of actions.

During the design of the knowledge layer, a revision of the initial model was done. It was found that the available information contained a lot of implicit and incomplete information. In particular, the flow between the procedure steps has to be explicitly defined, since a procedure is not always a simple sequence of actions, but may contain loops and decisions. It was required to make explicit the sequence of actions, which are the decision points and to identify if some actions could be done in parallel. In addition, doctors provided a list of documents for the service but they did not specify in which action they are used.

Some essential elements required to execute a K4Care procedure were identified:

- The concrete list of actions implied in a procedure. The actions that compose each procedure and which is the control flow of these actions in the procedure must be known.
- The actors implied in each action. That is, who are the people allowed to perform the action.
- The documents required to perform the actions.
- The actors' permissions to read/ write the documents.

A model to represent the information of a procedure was studied [18]. This model should allow to represent basically: states, actions, decisions and the flow between them; but it also must be able to include information about the actors that performs an action, which are the documents associated to this action and its permissions. The solution adopted to encode procedures was the formalism called SDA*. The SDA* model [13] is stored in XML and represents the repetition of *states*, *decisions* and *actions* in order to describe health care procedural knowledge. *States* are used to describe patient conditions or situations. *Decisions* represent alternative options whose selection depends on the available information about the patient. Finally, *Actions* are proper treatment steps. States, decisions, and actions are combined to form a joined representation of how to deal with a particular health care situation. This formalism has some properties that make it suitable to be used in procedure representation: it supports the storage of the actors involved in each action, it supports concurrence, it stores time-related information like in which moment an action has to start, to end, or if it has to be repeated.

Initially, this language was only considered to represent the Formal Intervention Plans used to treat patients. However, taking in account the above arguments, it is now also used to represent the Procedures of the medical services provided by the K4Care system [18]. However, there is still some information that is not included in the SDA*: the link between documents and actions. The solution proposed is to store this relation into the Actor Profile Ontology, which already has information about types of documents and their access rights.

An example of the codification of procedures using SDA* will be explained in section 5.

4 Data Abstraction Layer

In the K4Care model presented, the agents must be able to manage data from many sources; each source requires the use of a different language. That solution is inefficient because the information is not transparent for the multi-agent system. Our proposal consists on creating an intelligent intermediate layer that: (1) implicitly understands the different languages used to represent the data and (2) that knows where the knowledge is located. Therefore it will is able to manage the communication requirements to the proper knowledge sources.

This intermediate mediator is called *Data Abstraction Layer*(DAL), see figure 2. This layer permits the agents of the Multi-Agent System to access the knowledge through Java calls, whatever is the real data representation language. The DAL is composed by different Application Programming Interfaces (APIs) and a new element called Data Access Interface. The APIs are a set of Java methods that work as a bridge between the knowledge stored in a particular place and the rest of the system.

The EHR API provides the information contained in the EHR and the administrative data, which are stored in the database. The results of the EHR API are given in the Java *K4CareStorage* object. The SDA API provides access to the data stored in the SDA representation, for example the Formal Intervention Plans that define the guidelines to treat the patients who suffer from particular ailments or diseases or procedures which are executed to perform a service. The APO API and a CPO API have been created to consult the knowledge contained

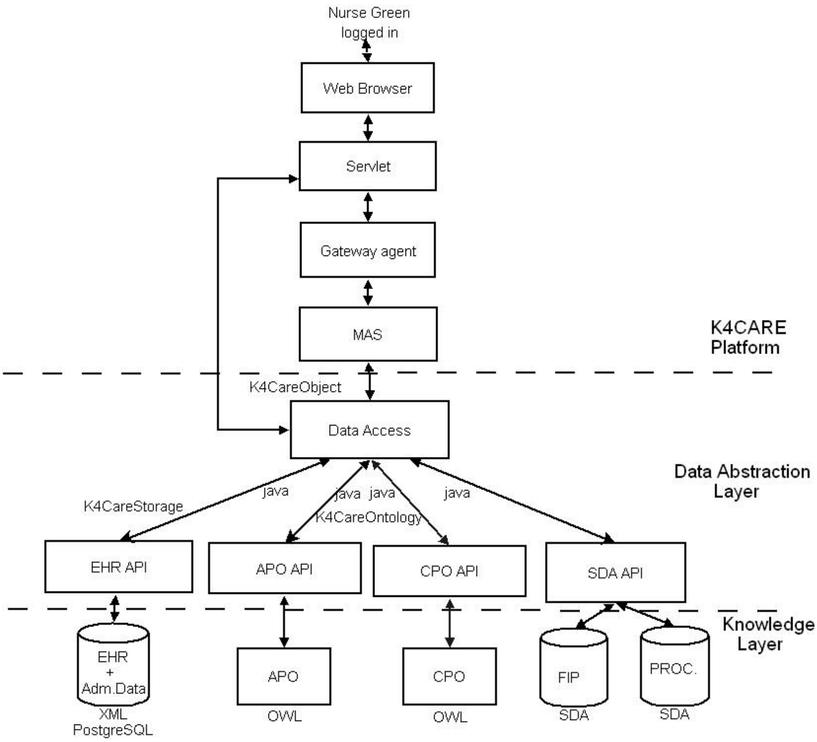

Fig. 2. K4care extended model

in the APO and the CPO ontologies. These interfaces allow high level consultations to the ontologies. These APIs, that return data through a *K4CareOntology* object, permit consultations to the specific knowledge contained in the APO or the CPO, such as the services that an actor can start or the types of documents contained in the EHR.

The most important part of the Data Abstraction Layer is the Data Access Interface (DAI). This element works with many different knowledge sources and translates the data to fit into the Java objects required by the multi-agent system. The Data Access Interface has also an active role, it is in charge of the permissions of access to the different data sources; it prepares the forms for the data to be displayed to the browser; it combines information from different sources to give a unique answer to high-level queries, etc.

4.1 Data Access Interface

The Data Access Interface (DAI) is a set of classes that are able to compose simple and complex queries from the petitions done by the agent, retrieves the corresponding information contained in different knowledge sources and composes a answer to the agents. It is not considered the possibility that the DAI provides functions to modify the ontologies because the medical model must not be changed by the agents. The data that can be updated through this layer is the one related to the medical care of specific patients (this is, the EHR) and the administrative user's data, and the SDAs of FIPs adapted to a particular patient.

To illustrate the use of the DAI we present some examples of the type of situations in which data access is required:

– First, when an agent starts, an authentication should be performed in order to establish permission policies for this agent.
– Obtain administrative information from the data base, such as the list with the roles that an actor can play. Each actor can play more than one role in different sessions, for instance an actor can play the family doctor role, and the specialist physician role.
– Provide a list of services that an actor can start when he is playing a role.
– When the user starts a service an agent will ask the procedure of this service and its related information (SDA flow diagram, documents associated).
– Once the procedure is started each action is performed by an agent that needs to know which documents are needed. The DAI must verify the rights of the actors to read, or write documents.Provide to the agents the document forms from document schemas. Then, the document forms can send to the agents empty or they can be partially auto-completed with values obtained from the different data sources (for example, with personal data of patients, such as identification number, name or date of birth).
– Provide any kind of data about diseases, symptoms, Formal Intervention Plans or Individual Intervention Plans.

At the moment, the functions of the DAI are structured depending on their use in the system. There are 8 groups of functions:

– Log in related functions. They are required by the servlet when a person wants to access the system.
– Functions to generate a template for an agent of a particular type of actor. There are some functions to create the different kinds of actors (agents that represent doctors, nurses, patients, etc). These functions extract the knowledge from the APO to create these templates.
– Functions to start the K4Care agents. DAI provide functions to obtain data from the database to initialise the agents.
– Functions to manage users. DAI provide functions to create or delete users, consult information about the actors, to know the patients treated by a particular physician, etc.
– Service related functions. This kind of functions provides information needed to perform a service, such as the list of different procedures for a service, the services that a particular actor can start or the actions of a procedure.

- Functions to manage the actors' roles. During care procedures there are several people interacting (patients, relatives, physicians, etc) and an actor can play one or more of these roles. These functions are used to know all the roles of an user or the actual role.
- Functions that manage Evaluation Units. An Evaluation Unit (EU) is a team of professionals that are in charge of assess the problem, define an individual intervention plan, identify the proper procedures, evaluate the results and verify the achievement of the goals defined by the IIP. The EU is composed by a Family Doctor, a Physician in Charge, a Head Nurse, and a Social Worker. The Data Access allows to create EUs or to obtain the members of the EU.
- Functions to manage Individual Intervention Plans and documents. The EHR contains different kinds of documents. The DAI provide functions to get or to store documents, or to ask to the APO the permissions of an actor to read or write a document.

In addition, we can consider another classification of these functions into two types: (1) basic functions that only interact with one knowledge source, for example, when the agent asks for the members of an EU, the DAI queries to the database the members on that Evaluation Unit identified by its identifier, and (2) complex functions that access to more than one knowledge source, for example, when an agent requests a document. This function returns a list of the identifiers of a document that an actor reads or writes, when he is playing a specific role for the patient identified by the identifier of the patient. This is a high level consultation, the response of it requires query the APO and the EHR. Initially, it must be identified which is the APO of the actor. Each actor has his own APO (or sub-APO, a sub-set of the APO). An APO must be instantiated in a particular physician, patient or citizen before it is applicable. This instantiation process will permit that a particular actor could introduce his or her particular vision of his role in the K4Care model. Then the DAI consults the permissions of the actor, that is, which documents can read or write an actor playing a specific role. Then the DAI query to the EHR which are the documents of an actor, playing a role, and related with a patient.

The use of the DAI in the K4Care platform is done by means of instantiations of the DAI by the entities that need it. These entities are the agents of the multi-agent system and the servlet. The Data Access always interacts with the agents except when an actor logs into the system. According to the design of the multi-agent system, an agent representing an actor starts when the corresponding user's authentication has been done. Since this time, the agent uses its instantiation of the DAI to access the data. The case of deploying this model to a real setting has been considered. With this model, the Data Access will be capable to attend consultations of hundreds of agents of medical centres (such as the agents in charge of the execution of procedures or Formal Intervention Plans, the agents that represent family doctors, nurses, social workers, etc.), because each agent instantiates this interface.

5 Case Study: Comprehensive Assessment

In this section a case study is used to illustrate the construction, contents and use of the Data Abstraction Layer. One of the services available in the K4Care project has been selected as case study: the Comprehensive Assessment (CA) [2]. This is a service to assess the condition of the patient during the first encounter, and whenever a re-evaluation is required. The CA service is devoted to detect the whole series of patients' diseases, conditions, and difficulties, from both the medical and social perspectives. It is performed at admission, at periodical or end-treatment re-evaluation time and during the evaluation of the patient condition through the time. Comprehensive Geriatric Assessment (CGA) [19,20] served as a model for the process of assessment: it is a multidimensional process designed to assess an elderly person's functional ability, physical health, cognitive and mental health, and socio-environmental situation.

As argued before, the main problems detected with the original specification of procedures were the lack of control flow definition and the no connection between actions and documents. A solution has been presented in section 3: procedures will be encoded using the SDA* representation, and the link between documents and actions will be stored in the Actor Profile Ontology. In this example, the Comprehensive Assessment, the initial information available was the list of steps to be followed for performing a CA (see Table 1). After agreeing with our proposal, the doctors participating in the K4Care project defined the relation between actions and documents in the CA procedure and the permissions for the actors on those documents (see Table 2). After having that information, we have worked together with the experts to represent the flow of actions in the CA service in SDA* (see figure 3). The SDA* diagram represents the precedences among the actions, it also indicates which documents are used in each action. Notice that some actions can be done alternatively by an actor or another.

After analysing the procedure represented in Figure 3, a list of petitions for the DAI has been generated:

- When the user starts the CA service, the user's personal agent will require to the DAI the SDA* procedure of this service.
- Once the procedure is started each action is performed by an agent that sends its requests to the DAI. For example, in action BO.03 the agent requests the document schema D10 to be retrieved from the EHR. But in step 7 of the CA the agent requests for a new procedure, the procedure of the service S3.4.
- The DAI will verify in the APO the rights of the actors to read, or write documents. In the previous example, in action BO.03 only the Head Nurse or the Physician in Charge will get the document.
- During the action the actors will fill in the document D10. When the action finishes, the DAI must provide the functions to store the document in the patient's EHR.

Table 1. Comprehensive Assessment Procedure

Comprehensive Assessment Procedure	
Code	**Description**
BO.03 refer the admitted patient for CA	The PC or the HN refers the admitted patient for a CA.
BO.05 assign members of EU BO.13 actor confirmation	The HN assigns the members of the EU.
BO.08 send message to the patient	The HN sends a message to the patient to make an appointment.
P.1 confirm appointment	The HCP confirms the appointment
EU.1 evaluate through scales	The EU makes the patient's assessment at home according to a standardized interview (Multi-Dimensional Evaluation).
S3.3 Clinical Assessment S3.4 Physical Examination	FD or PC performs Clinical Assessment and Physical Examination.
S3.8 Social Needs and Network Assessment	The SW performs the Social Needs and Social Network Assessment.
BO.01 provide information	The HCP provides the necessary information. In case of a non-compliant or non reliable HCP, the CCP provides the necessary information.
BO.06 confirm or modify waiting lists BO.07 schedule activity	The HN performs Case Management or Back Office proper actions

Table 2. Comprehensive Assessment Documents

Comprehensive Assessment Documents			
Name	**Purpose**	**Description**	**Permissions**[1]
D10	Request	Request of CA	Head Nurse, Physician in Charge:W/R
D11	Anamnestic	MDE Scales	Evaluation Unit:W/R
D12	Anamnestic	Clinical History	Physician in Charge, Family Doctor: W/R
D13	Anamnestic	Physical Examination Report	Physician in Charge, Family Doctor: W/R
D1	Request	Actor assignment	Head Nurse: W/R
D2	Authorization	Actor Confirmation	Evaluation Unit: W/R
D5	Request	Message to the patient	Head Nurse: W/R
D6	Authorization	Patient Confirmation	Patient: W/R

– In other steps of the CA procedure, the agents needs to start another service. For example, in step 7 the DAI must verify if the actor has the permission to start the service S3.4. If it is the case, the agent will require the corresponding SDA*.

[1] W/R=Write/Read

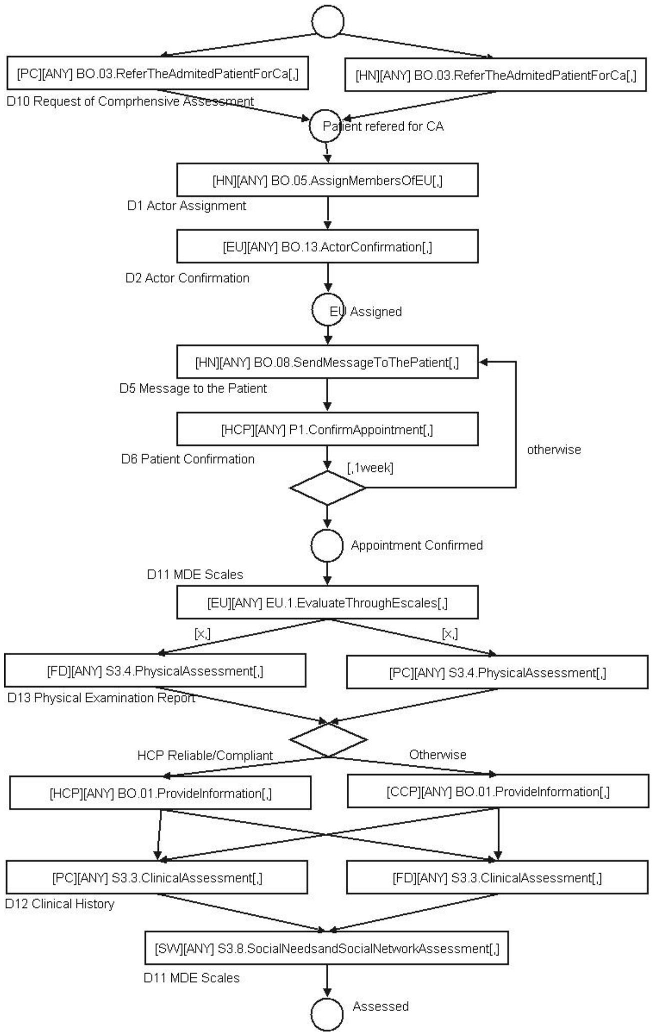

Fig. 3. Comprehensive Assessment SDA

6 Conclusions and Future Work

The fact of adding an intermediate layer to the K4Care model and the creation of the DAI has provided a series of improvements: the model is best structured; there is independence between layers; the different codifications of the knowledge are transparent for the multi-agent system; and the multi-agent system will be able to make complex petitions involving information stored in many sources. In short, agents do not need to know how the Knowledge Layer is implemented.

The study done to solve the initial problem has contributed with other improvements to the model. For example, the need of making explicit the relations between actions and documents in the knowledge data. It was also identified the need of having a representation for the flow of the actions into the service procedures, proposing to use the SDA model (which has been accepted by the K4Care consortium). Finally, the development of specific application interfaces for the different types of data sources has been proposed.

Although the Data Access Interface has been designed and implemented as part of the K4Care system, it could be integrated into any other medical system that organizes the information following the K4Care information model. Also, the integration to a medical system with other EHRs or ontologies structures is really easy if the APIs maintain the same form. This generalization only requires the reimplementation of the internal queries that APIs send to the corresponding information source.

At this moment, each agent makes an instance of the DAI. As a future work, it would be interesting to identify which functions are used for each type of agent, and, then, generate different interfaces depending on the agent. With this personalisation, agents will only instantiate the functions that they will use.

Acknowledgments

This work has been funded by the K4CARE project (IST-2004-026968), the HIGIA project (TIN2006-15453-C04-01) and by the Student Research Grant of the University Rovira i Virgili. The authors also acknowledge the help of David Sánchez and the doctor Fabio Campana, as well as to all participants of WP6 of K4Care project in charge of MAS design.

References

1. Wooldridge, M.: An introduction to multiagent systems. Wiley, Chichester (2002)
2. Campana, F., Annicchiarico, R., Riaño, D.: D01 - the K4CARE model. In: Internal deliverable for the K4CARE project (2007)
3. Batet, M.: How to facilitate the connection between distributed data and agents in the k4care project. Master's thesis, Escola Técnica Superior d'Enginyeria, Universitat Rovira i Virgili, Campus Sescelades, Av. Països Catalans 26. Sant Pere i Sant Pau (43007), Tarragona. Catalunya (2007)
4. Pisanelli, D.: If ontology is the solution, what is the problem? IOS Press, Amsterdam (2004)
5. Studer, R., Benjamins, V., Fensel, D.: Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering 25(1-2), 161–197 (1998)
6. López-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Enginnering. Springer, Heidelberg (2003)
7. Kumar, A., Ciccarese, P., Smith, B., Piazza, M.: Context-based task ontologies for clinical guidelines. In: Pisanelli, D.M. (ed.) Ontologies in Medicine. Proceedings of the Workshop on Medical Ontologies. Studies in Health Technology and Informatics, vol. 102, pp. 81–94. IOS Press, Amsterdam (2004)

8. Isern, D., Sánchez, D., Moreno, A.: An ontology-driven agent-based clinical guideline execution engine. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, Springer, Heidelberg (2007)
9. Davis, J.P., Blanco, R.: Analysis and architecture of clinical workflow systems using agent-oriented lifecycle models. In: Silverman, B., Jain, A., Ichalkaranje, A., Jain, L. (eds.) Intelligent Paradigms for Healthcare Enterprises. Studies in Fuzziness and Soft Computing, vol. 184, Springer, Heidelberg (2005)
10. Casals, J., Gibert, K., Valls, A.: Enlarging a medical actor profile ontology with new care units. In: 11th Internacional Conference on Artificial Intelligence in Medicine, Workshop from Knowledge to Global Care, pp. 11–19 (2007)
11. O.W.L. (2007), http://www.w3.org/tr/owl-features/
12. Isern, D., Moreno, A.: Computer-based management of clinical guidelines: A survey. In: Proc. of Fourth Workshop on Agents applied in Healthcare in conjunction with the 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, pp. 71–80 (2006)
13. Riaño, D.: The SDA model v.1.0: a set theory approach. Technical Report DEIM-RT-07-001, Dept. of Computer Engineering and Maths, Universitat Rovira i Virgili,Tarragona, Spain (2007)
14. Iakovidis, I.: Towards personal health records: Current situation, obstacles and trends in implementation of electronic healthcare records in europe. Int. J. Medical Informatics 52(1), 105–115 (1998)
15. Batet, M., Valls, A., Gibert, K.: Survey of electronic health records standards. Research Report DEIM-RR-06-004, Department of Computer Engineering and Maths, Universitat Rovira i Virgili (2006)
16. PostgresSQL (2007), http://www.postgresql.org/
17. XML: (2007), http://www.w3.org/xml/
18. Batet, M., Valls, A., Gibert, K.: The SDA as a model for flow control in k4care. Research Report DR 2007/8, Department of Statistics and Operational Research, Universitat Politècnica de Catalunya, Barcelona, Catalunya, Spain (2007)
19. Stuck, A., Siu, A., Wieland, D., Adams, J.: Comprehensive geriatric assessment: A meta-analysis of controlled trials. Lancet 342(8878), 1032–1036 (1993)
20. Nikolaus, T., Specht-Leible, N., Bach, M., Oster, P., Schlierf, G.: A randomized trial of comprehensive geriatric assessment and home intervention in the care of hospitalized patients. Age and Ageing 28, 543–550 (1999)