A Multi-Agent System for Organ Transplant Management

Monique Calisti, Petra Funk, Sven Biellman, Thomas Bugnon

Abstract. This paper describes the *Organ Transplant Management* (OTM) system, an agent-based platform to be deployed by medical practitioners as a smart software support for data management and decision making. The twofold goal is to identify the main challenges and benefits that the OTM paradigm brings along in order to validate and/or eventually modify and refine our choices. This also hopes to provide the readers with a concrete feedback on the feasibility and utility of deploying software agents in today's medical environments.

1. Introduction

During the last few years, there have been significant improvements in the area of organ transplant. Transplants are no longer the last-option kind of therapies, and recent important successes are contributing to increase the number of transplants all over Europe. However, despite significant advances in the surgery process itself, the coordination of the preliminary activities involved in an organ transplant operation is still a very challenging, complex, and not yet well understood process [18] (see also concrete examples in [4, 8, 21]). The way distinct OTM tasks are currently performed still lacks from a comprehensive computational support and it is to a great part not or poorly coordinated. Medical experts have to match available organs to long lists of potential recipients with very weak local computerized support in processing large amount of data. Coordination is mainly achieved by telephone calls, pagers and/or other highly human-dependent means to track down suitable recipients, and contact physicians at both the extraction and insertion sites. Furthermore, the storage and the transport of organs involve a set of activities under the direct control of external and distinct organizations, which need to interact in a rapid way (available organs cannot be preserved for long time).

In this context, several recent initiatives are trying to address the need for more flexible and powerful support for medical practitioners involved in the OTM process by working at different key levels (i.e., legal procedures, hospital infrastructures, coordination centers organization, software support, etc.). The main goal of the work presented in this paper is the definition and implementation of a software agent-based approach,

Key words and phrases. organ transplant, agents, medical knowledge representation, eHealth, ontology.

namely the OTM system, as a comprehensive solution offering computational support for two crucial sets of tasks which medical teams involved in the OTM process have to face:

- *Information gathering and data management tasks*: both, potential recipients (i.e., patients waiting for a transplant) and donors need to be characterized by a certain amount of data that has to be collected, stored, processed and dynamically maintained by the transplant centers and/or specific institution/s involved in the OTM process. This is a very challenging issue, as no common comprehensive standard framework is available nowadays to uniquely represent, store, process and visualize medical data, in particular in the OTM context. The critical point is not only to agree upon a given formalism for data representation (e.g., XML, RDF, DAML+OIL, etc.), but rather to provide a semantic grounding for knowledge representation that can be easily understood by humans and effectively manipulated by software entities at the same time.
- *Decision making processes*: when organs become available, the responsible medical teams in the involved transplant centers have to process a significant amount of information (organ and corresponding waiting patients descriptions) to verify whether there is a compatible recipient on the corresponding waiting lists. This decision has to be taken under very strong time constraints by considering a large set of interrelated match making criteria, including several deterministic medical factors (about the 80% of the total amount of data to be checked) plus a number of aspects (soft constraints) that are more difficult to quantify and estimate, but that can heavily influence the final selection of a specific patient. The additional issue here is that different “degrees of compatibility” can be determined in relation to those soft constraints. Therefore, the final answer could rather be a list ranked according to the compatibility rates.

This paper describes the OTM system aiming at identifying the main challenges and benefits that our agent-based approach brings along to validate and/or eventually modify our choices. This hopes to stimulate discussion and provide feedback on the feasibility and utility of deploying software agents in today’s medical environments. The remaining of the paper is organized as follows. Section 2 focuses on the description of the OTM system. Section 3 reports on the knowledge modelling and OTM ontology definition work. The experience achieved in building the agent-based OTM demonstrator leads to the discussion of the main issues encountered so far as well as the expected benefits out of the deployment of software agents (see Section 4), before concluding the paper with an overview of ongoing work (Section 5).

2. The Organ Transplant Management System

The OTM system [1] is a modular software agent-based platform, including adaptive and user friendly graphical interfaces to facilitate the access for nurses, physicians, surgeons, etc. (who are not necessarily IT experts) to both the data and the mechanisms required for effective coordination of the main OTM tasks. Agent technology is one of the most promising approaches for designing and implementing autonomous, intelligent and social software assistants capable of supporting human decision making [9]. As mentioned earlier, the fundamental idea in the OTM context is to provide computational help for decisions that have to be taken by considering a substantial number of medical factors as well as legal rules and requirements under very strong time pressure.

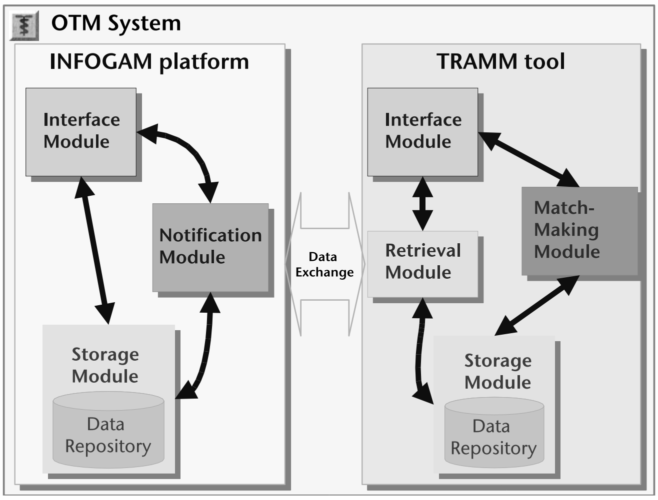

FIGURE 1. An overview of the OTM system architecture.

In this perspective, we propose a pro-active, modular and flexible approach consisting of two main sub-systems: the *Information GAthering Module* (INFOGAM) and the *TRAnsplant Match Making* (TRAMM) tool (see Figure 1). Various agents populating these two sub-systems interact and coordinate on behalf of medical practitioners, transplant coordinators and patients in order to improve the OTM process. The development tools that have been deployed to implement the OTM system are:

- *Jade* v. 3.0b1, a software development framework to develop multi-agent systems and applications conforming to FIPA standards for intelligent agents. (http://sharon.cselt.it/projects/jade/).
- The Sun Java Development Kit v. 1.4 (http://www.javasoft.com).
- *Protege* v. 1.8, an ontology editor used to create the OTM ontology in combination with the UMLS tab and the Bean Generator plug-in. (http://protege.stanford.edu/)

Before a more detailed description of the various software components, the definitions of some recurrent terms are given for clarity's sake.

Definition 2.1. Donor: a person who donates organs for organ transplantation.

Definition 2.2. Hard constraint: a compatibility condition on the selection of a patient for a given available organ that must be satisfied.

Definition 2.3. Patient: a person who receives medical attention, care, or treatment. A patient can be a recipient as well as a potential donor.

Definition 2.4. Recipient: a patient who is waiting for an organ transplant. A recipient is registered in one or more waiting lists (for different organs) in the OTM system within a given transplant center.

Definition 2.5. Soft constraint: a compatibility condition on the selection of a patient for a given available organ that can be satisfied.

Definition 2.6. User: a person accessing the OTM system. Various user types (e.g., patients[1], nurses, doctors, surgeons, etc.) have different access rights.

2.1. The Information Gathering Module

The INFOGAM platform consists of three main parts. The *Storage Module*, SM, contains one or possibly more databases in which all the relevant OTM data (user profiles, Electronic Health Records (EHR), etc.) is stored and the agents responsible for the various data management operations, including the database access control. The *Interfacing Module*, IM, consists of all the required components enabling interactions with human users, i.e., enabling human-agent interaction to be carried on in a flexible and user-friendly way. The third component, i.e., the *Notification Module*, NM, allows the system to pro-actively notify users. The category of notifications we actually implemented in the system includes reminders, alerts and suggestions (1) to patients having to take some medical tests, (2) to selected recipients for quickly contact the hospital, in case they are not there, (3) to nurses for recalling patient's medication and/or special treatments, or for alerting about shortage in resources, (4) to physicians whenever organs become available, etc. Reactive (mainly as consequence of events monitoring and user commands) and pro-active (mainly as output of goal-driven agents reasoning) notifications represent a key aspect of the capabilities of software agents and thereby of the services supplied by the OTM system. While *reactivity* mainly guarantees to face expected categories of events (such as shortage of blood of a certain type, lack surgical material, etc.), *pro-activity* is intended in this context as the capability of goal-driven deductions/actions whenever unexpected situations occur (e.g., suggesting specific medical crossed patients tests involving more than one medical team), taking the initiative of performing given actions in a way to prevent undesired events to take place (for instance, suggesting to order additional medical material not as a consequence of a shortage in the current resources, but because of a forecasted greater need whenever the number of registered patients in the hospital increases).

The Storage Module. The database contained in this module is maintained by the DataBase Connectivity Agent (DBCA). Any entity in the system having to store information requests the DBCA to perform this service. This centralized approach has been chosen mainly for consistency, modularity and security reasons. By having a unique access point, authorized operations and data consistency can be more easily controlled. On the other hand, this requires to set up appropriate mechanisms to ensure failure recovery (in our case, the back up of the DBCA in case of crashes) and scalability (in the OTM system, the DBCA can delegate part of its tasks to authorized agents). There are four other types of agents in the SM, all heavily interacting with the DBCA. Each of these agents

[1] Patients are users of the system in the sense that they might directly enter and keep up to date their personal data (address, phone number, etc.) and they might also have access for visualizing their own medical status.

is responsible for manipulating a specific part of the knowledge base: the User DataBase Agent (UDBA), the Patient DataBase Agent (PDBA), the Waiting List Agent (WLA) and the Notification DataBase Agent (NDBA). The UDBA manages the user information, i.e. the identity, the access rights, contact information, etc. The PDBA maintains the patients EHRs: it creates, modifies, fetches EHRs, etc. The UDBA and PDBA interact mainly with the Interfacing Module. The WLA maintains the waiting lists, i.e. adds new patients, update their conditions, notifies the medical staff about missing information, etc. The WLA interacts with all the other INFOGAM modules and most important with the TRAMM tool. The NDBA stores the end users notification preferences and mainly interacts with all other agents in the NM. All the SM agents can refer to a specific user or patient using his/her unique identifier, which enables an efficient separation of the different sets of data. Note that the UDBA is also acting as Key Distribution Center (KDC) for security purposes. Each user has indeed restricted access to the data stored in the OTM platform. Therefore, the system needs to know which user is requesting data and/or willing to perform any action before allowing any access/execution. The proposed solution is to make use of a Kerberos-like approach[2] and has been designed and implemented according to [10], [11]. This is a username/password based mechanism in which the authentication credentials are needed only once for the whole session and they are not stored, cached nor sent into the network. The protocol is based on *tickets*. To have access to a service, an OTM agent needs the proper ticket. An agent asks the ticket granting service (TGS) for tickets. Each ticket can be used only once. To avoid having to specify the password each time a service has to be accessed, the agent first gets a special *ticket granting ticket* (TGT) from a unique TGS in the system called Authentication Server (AS). Such TGT is valid during a whole session and usually expires after eight hours. Once an agent obtained the TGT it will use it in order to request from the TGS the proper ticket for specific services. Most of the time, the AS and the TGS are the same server called then the Key Distribution Center. As stated above, in the OTM system the UDBA acts as KDC.

The Interface Module. This module provides the medical staff with a broad range of specific interfaces to access the platform. This highly user-centered approach aims to facilitate the personalization of tools, services and actions to be performed, but also to appropriately customize the access control to the OTM system. The embedded Graphical User Interface Agent (GUIA) takes care of contacting the specialized interface agent (SIA) with the proper features corresponding to a given user with specific access rights. Personalized graphical interfaces are then instantiated for patients (PIA), medical practitioners (MIA), nurses (NIA), administrators (AIA) and laboratories (LIA). This choice of separating the common features, such as authentication (performed by the GUIA), from the role specific features by means of different specialized agents has been done for modularity and security reasons. Since there exists a large range of (more or less) standard forms, reports and letters related to organ transplantation that need to be filled in, printed and/or modified, we also created specific devoted software entities: the Input/Output Agents (IOAs). An IOA can provide an empty form to be filled in by a user,

[2] More details on access control and security within the OTM system are addressed by the authors in a separate paper to appear.

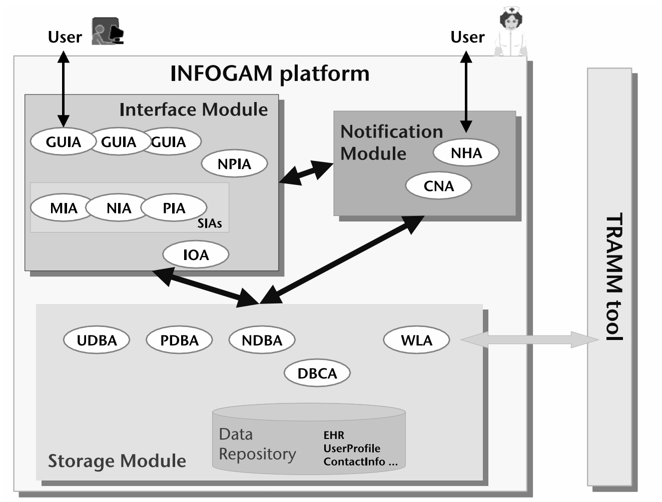

FIGURE 2. An overview of the INFOGAM architecture.

save a form in the database or print it, and fill in (when authorized/required) forms by itself with data directly retrieved from the database. Finally, the Notification Preferences Interface Agent (NPIA) provides an interface accessible to all OTM users so that they can set their notification preferences. A notification preference specifies by which means the final end users can be notified (e.g., telephone call, SMS, e-mail, etc.), depending on the type and priority of the notification[3]. The NPIA asks then the NDBA to store the stated preferences.

The Notification Module. Proactive OTM agents have the capability of reporting events, notifying (e.g., a deadline is crossed and requires somebody's attention), taking the initiative of performing specific actions and thereby informing end users, asking questions or suggesting/requesting to perform specific actions, e.g. data can be missing. To do so, the Notification Module embeds two main agents: the Central Notification Agent (CNA) and the Notification Handler Agent (NHA). When an agent has to notify a given user *U* about a specific event *E*, it sends the proper request, specifying a given priority level *L*, to the CNA. The latter checks the notification preferences of user *U* (first interacting with the NDBA to recover the notification priority profile) by adopting the following procedure: (1) if user *U* has set a notification preference for event *E*, make use of the specified notification means *M* for *E*; (2) if no preference is pre-defined for event *E*, make use of the specified notification means *M* for the given priority level *L*; (3) if no means is pre-selected for *L*, make use of the means for the lowest priority level *L'* higher than level *L* the user *U* has set. We force users to select at least one priority means to be deployed for notification. When the notification means *M* is selected, the CNA contacts the suitable

[3]Besides pre-defined notifications, OTM agents can pro-actively decide to notify end-users depending on their goal-driven behavior.

NHA. Each NHA is devoted to perform a specific type of notification (SMS, pager, e-mail, phone call, etc). At the moment, OTM agents are able to send text messages via SMS, e-mail and graphical displays (i.e., text and alert icons in the GUIs).

2.2. The Transplant Match Making Tool

The TRAMM tool architecture has been articulated over 4 main modules, by following design requirements dictated by current practice in the hospitals we are collaborating with, but also for guaranteeing a more easily manageable separation of concerns, access rights and responsibilities of the various agents. Although, some of the embedded agents perform similar duties and present analogous characteristics to the corresponding agents in the INFOGAM modules (like enabling access to the system -interface management- or enabling database management), the corresponding actions into the system are centered on supporting the match making process and act on different sets of data.

The Interface Module. This module, by means of the Graphical User Interface Agent (GUIA), manages the various graphical interfaces required to access the TRAMM platform. This includes: (1) the login interface used to identify and authenticate a user by means of the user name and password; (2) the match making interface needed to enter donor and organ specific data, display the ranked matching list and possibly indicate (communicate back to the system) the final decision of medical experts; and (3) the administrator interface deployed to create, modify and delete user profiles. Once a user has been identified and authorized to access the platform, it is possible to have access to all the different functions offered by the TRAMM system. The Match Making Interface Agent (MMIA) collects data about a donor and the donated organ(s), as they are entered by responsible nurses and/or physicians, and display the result of the match making process, i.e. the organ specific ranked matching list. The Administrator Interface Agent (AIA) agent provides the administrator of the TRAMM tool with an appropriate interface to manage the user profiles, e.g. creating/deleting user profiles, modifying user roles, etc. Finally, the Input Output Agent (IOA) takes care of input and output actions like printing forms, mapping forms to objects and vice-versa, saving/retrieving forms in/from the database, and receiving data from Swisstransplant (national coordination center).

The Interchange Module. This modules coordinates the flow of data and information exchanged between the INFOGAM and the TRAMM sub-systems. By means of the embedded Data Management Agent (DMA) all information needed by the TRAMM agents (i.e., waiting lists information, patients characterization, etc.) and maintained by INFOGAM software entities is retrieved. This data flow is then appropriately processed and forwarded so that information can be communicated to the interface module (whenever visualization of information to medical practitioners is needed/required), and to the match making module (for recipient selection out of selected waiting lists). On the other hand, whenever decisions are taken (a recipient is selected) and waiting lists need to be updated the DMA notifies the WLA in the INFOGAM platform.

The Storage Module. Analogously to what happens in the INFOGAM platform, this module is responsible for the data management part. In this case, however, the information managed concerns (1) donors (e.g. create, insert, update, view donor data records); (2) available organs (e.g. storage/retrieval of organ specific characteristics); (3) TRAMM

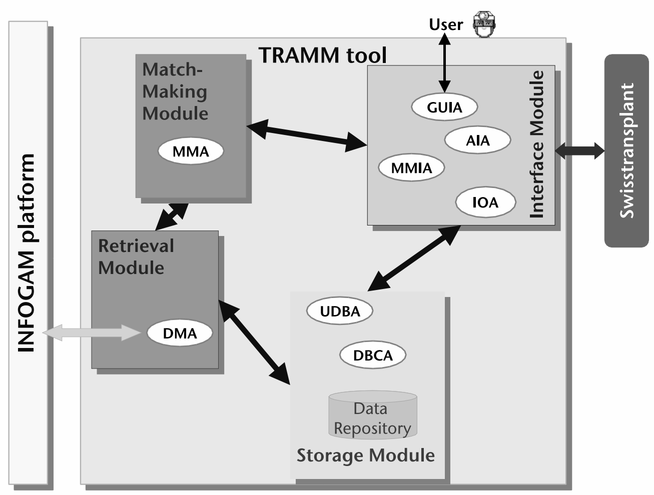

FIGURE 3. An overview of the TRAMM architecture.

users data (e.g. storage/retrieval of user names, passwords, access rights to the TRAMM services); and (4) final transplant decision (e.g. keeping track of selected recipients for given organ/s, reasons, motivations, match making results, etc.). The DBCA agent is responsible for managing the TRAMM database and retrieve/store/update information according to the various OTM agents and users requests. The UDBA is responsible to create, modify, update, delete and retrieve user information by processing requests from agents in the TRAMM Interface Module and by identifying and authenticating users that access the TRAMM services.

The Match Making Module. The Match Making Module (MMM) provides mechanisms for the selection of appropriate recipient candidates whenever organs become available. This includes facilities for the Match Making Agent (MMA) to be able to perform (1) the selection of organ specific match making policies to be used; (2) the retrieval of organ specific match making rules, including hard and soft constraints to be applied; (3) the selection of a set of compatible candidates from a waiting list according to organ specific hard constraints; (4) the computation of a score for each selected potential compatible recipient according to organ specific soft constraints. The MMA compiles a ranked matching list of potential recipients (when available), by taking into account statistics data about former and similar decisions on the computation of the matching score. The MMA provides this list together with an explanation of the computed scores, listing possible positive factors and risk factors. The organ specific matching rules can be retrieved from the SM database by interacting with the DBCA and as a future extension directly from XML based file repository directly managed by the National transplant coordinator. With regard to the match making process, although we have looked at some solid work done in the same problem area [24], we have currently implemented a constraint satisfaction based approach [7] combined with case-based reasoning techniques [12]. The set of hard constraints (between 4 and 10 depending on the organ type[4]) is used in order to select which patients' descriptions to recover from the waiting lists (search space pre-pruning). Based upon the pre-selected list of patients satisfying hard constraints, the MMA runs then consistency techniques [5] on the set of soft constraints for final scoring and ranking of the patients. Upon request of surgeons, case based reasoning methods can also be used in order to correlate current cases to previous choices recorded by the system (if any).

3. Modelling and Representing the OTM Knowledge Base

For multi-agent systems, re-usable and sharable conceptualizations of domain-dependent contexts are vital communication enablers as they provide common definitions of terms including the relations holding among them, which thereby allows the elaboration of coordinated plans and solutions. It is beyond the scope of this paper to report on extensive works for formalizing and defining compact, standard and reusable ontologies in multi-agent systems. The aim here is to report on the experience gained when building the OTM ontology, in order to discuss the main concrete issues that need to be faced when building structured knowledge basis for software agents in the organ transplant area.

3.1. Requirements

The OTM ontology has to include all concepts required to support the various agents when performing the OTM *application tasks* for information management (i.e., creation, maintenance, update, deletion of health state descriptions of patients, electronic health records, waiting list management etc.) and multi-criteria matchmaking (specific laboratory values analysis and interpretation, organ specific rules and factors etc.). This requires (1) a set of notions that describe the possible agents' actions, (2) a set of concepts, including medical, personal (i.e., user related), and logistic terms (e.g., organ descriptions, transplant indications) that define the corresponding actions' domains and objects, and (3) a way to express how actions and concepts (or terms) relates to each other in the given OTM context (i.e., a set of predicates for describing the state of the world in terms for instance of actions' results, objects' values, etc.). When referring to medical concepts (e.g., organ description), the re-use and integration of existing medical ontologies and/or vocabularies[5] becomes essential. This is very important because well-structured and organized medical know-how, when available, can be more appropriately outsourced, understood and used by agent developers (or IT people in general), and more easily maintained and deployed by medical practitioners. For this reason, when building the OTM ontology, we carefully considered the re-use of existing pre-defined medical concepts (when available).

[4] The current OTM prototype enables the management of liver and kidney transplants. The main reason is that given the collaboration with two specific Swiss hospitals, we have been able to recover more detailed information for those two types of organs.

[5] By vocabularies we means collection of medical terms, where relations among them have not been directly included.

3.2. Relevant Medical Knowledge Sources

Some of the existing medical vocabularies address specialized tasks, while others are intended to cover broader knowledge providing the medical foundation of various electronic health care applications. Among the most well known structured medical knowledge bases, GALEN [16], SNOMED [19], and UMLS [23] have been more extensively described [15] as possible candidates for usage in agent-oriented applications.

In our framework, after having analyzed and discussed several optional choices, we decided to build the OTM ontology by making use of the *Unified Medical Language Systems* (UMLS). The main reason resides in the meta-description level of medical concepts that is provided. The creators and maintainers of the UMLS (the American National Library of Medicine), NLM [14], collect and distribute electronic medical knowledge sources provided by medical experts, organizations and companies. The NLM has built a relational database connecting over 60 vocabularies and medical ontologies by means of a conceptual unification. The main goal is to make the various existing medical *Knowledge Sources* (KS) accessible and usable through a common meta-level interface. UMLS users can choose distinct KS for specific purposes/applications by simply expressing which KS-dependent definition is associated to a given meta-level UMLS concept. The advantage of having a standard high-level interface is that when adding, updating or changing the various KS included by UMLS one can simply adapt its knowledge base by appropriately updating the mappings between UMLS concepts and KS definitions. UMLS creates for each concept (i.e., medical term) a unique identifier (CUI). This is used to associate the UMLS meta-level definition (if any) of the given concept to all the various instances in the different KS, in which a concrete definition of the specific concept (or any of its synonyms) is given. For instance, the term *organ* is a UMLS concept used in 4 distinct KS. By means of its UMLS CUI, i.e., C0178784, it can be uniquely identified. It is then possible to associate to this same concept different meanings (or different concrete definitions) by referring to one of the 4 different KS. If any of the knowledge sources is changing, it is still possible to use the same concept (and thereby the same ontology structure) either by referring the new definition or by switching to another KS (i.e., changing the association from the CUI to the selected KS). Of course, when changing concrete KS definitions, the corresponding objects in the OTM ontology might need to be updated, but eventually the relationships among various concepts do not need to be modified. Finally, by referring to meta-level descriptions, it is possible to adapt the OTM ontology to different KS (adopted for instance in distinct hospitals) by simply changing the mapping between UMLS concept identifiers and specific definitions in selected KS.

3.3. Building the OTM Ontology

The currently defined and used OTM knowledge base consists of approximately 75 concepts plus the relationships holding among these. This includes:

- A sophisticated and complete electronic health record, partially based on [22], including all necessary related medical concepts;
- Users identity and authentication related concepts for access control to the OTM system;

- Matchmaking factors and organ specific rules for recipient selection.

While for the OTM specific functionality, such as access control and recipient selection criteria, we defined the required objects, predicates and actions from the scratch, we referred (when possible) to UMLS for most of the medical concepts. There are two alternative ways of making use of UMLS definitions:

- *Direct Connection*: the UMLS knowledge sources can be directly accessed through the Internet by means of a JAVA API or plug-ins for tools such as Protégé-2000 [17]. However, while this remote access to the overall UMLS databases infrastructure can be quite useful for agent developers, it is quite slow for agents access given the huge amount of data to be processes.
- *Compliancy*: UMLS can be used as a reference knowledge base for the creation and formalization of the required domain-related concepts. This means to make sure that the medical concepts deployed in the OTM ontology (e.g., organ descriptions, transplant indication descriptors etc.), including definitions, relations and hierarchies are compliant to UMLS.

In the OTM context, we adopted this latter approach, because while it is still possible to easily adapt the OTM knowledge base by changing the mapping from meta-level concepts to specific knowledge base definitions (few values have to be changed in the OTM ontology specification), it is much faster and efficient for agents to directly access the required medical concepts in a local knowledge base.

4. Discussion

After several discussions with OTM experts, extensive studies of other works, research projects [2], [3], and existing systems offered in the market (see [1] for various references), we realized that even though some tools are available the majority of transplant centers have to face a serious fragmentation in the way such support is provided. There is no comprehensive approach, as far as we know, providing a compact computational solution integrating patients' data management tasks, match making and decision support procedures, which includes *proactive monitoring and goal-driven notification about patients conditions*. The OTM software agents are monitoring, interpreting and reacting to events in the environment they sense (such as changes in a patient's conditions, shortage of a specific type of blood for a given surgery operation), autonomously reasoning and pro-actively notifying the medical responsible interacting with the system (see Section 2). The main idea is to provide help in tracking events and managing data (i.e., faster and personalized access to stored information, user friendly visualization, etc.), in order to enable smart computational support for interpreting and explaining why data changed in a certain way and what such changes can mean in relation to dynamic health care patients conditions. In this perspective, autonomous software entities acting on behalf of patients and/or medical practitioners can:

- *Speed-up and automate many tedious and simple tasks currently performed by humans*. The resources saved by relying upon software instruments could be re-used for more demanding and delicate tasks;

- *Reduce the complexity of various tasks* such as multi-criteria match-making, dynamic coordination under stringent constraints by implementing smart techniques that can support the human decision making and optimize several processes;
- *Offer added value services*, such as the integration of some activities performed during the OTM with other medical services and vice versa, or the personalization and custom-tailoring of some tasks for which there is currently no way of differentiating the way patients are monitored and cured.

For the formalization of the medical knowledge, as anticipated above, we referred, when possible, to existing standard UMLS concepts and definitions. Besides, finding the correct balance between application driven knowledge, medical concepts and the integration efforts required to make this combination usable within an agent environment, we faced several challenging issues:

- Most of the existing medical ontologies and vocabularies are very complex, difficult to understand for not medical experts, and have not been necessarily designed to be used within an agent-based environment.
- Given the huge amount of available medical knowledge bases, it is important to carefully analyze the usefulness and usability of these in the context of the OTM application, and thereby select the most appropriate one.
- The integration of an existing medical ontology/terminology into an agent-based context, especially in compliance with the communication stack given by the FIPA [6] standards or in relation to the deployment of a specific agent platform, such as JADE [20] in our case, possibly requires changes or adaptation in the way the knowledge is formally represented and thereby encoded.

The closest and most comprehensive approach we have been looking at has been developed by the GruSMA group [13] and it is tailored to the Spanish transplant framework and rules. In our case, the overall OTM data modelling and system functionalities have been defined in relation to the Swiss context.

5. Conclusion and Future Work

In the health care scene, while the automation of some activities and the deployment of auxiliary software tools have partly enabled to address pressing requirements for more effective information management and decision-making, there is still a strong need for addressing the complexity of integrating new solutions within existing hospitals' infrastructures (i.e., pervasive health care). This requires taking into account users' needs and various existing constraints (medical, legal, etc.) in a more flexible, dynamic and personalized way, in particular in the OTM context. We strongly believe in the potential of our agent-based approach to be used as a distributed software support tool that would enable a better taking care of the individuality of the patients, saving precious resources and would finally facilitate the tasks of physicians and coordinators.

The currently available OTM prototype requires various refinements and extensive validation and testing on the field. Both the INFOGAM and TRAMM modules have been implemented and preliminarily validated. Agents deploying the OTM ontology work on a set of simulated organ transplant cases. At the moment, we are working at different levels:

- Improving the various graphical interfaces in order to facilitate the deployment of the system.
- Working in close collaboration with some medical experts to better define and model the OTM ontology, the soft constraints and the various deployed match making rules and policies.
- Increasing security for system access and data exchange.
- Refining the cased-based reasoning part based upon the feedback from medical experts.

For the future, the aim is to validate the OTM approach on real clinical cases, besides covering the matchmaking process of other organ types.

References

[1] M. Calisti, P. Funk, and P. Brunschwig. Software support for organ transplant management. In *Proceedings of Medical Informatics Europe 2003*, 2003.

[2] U. Cortés, A. López-Navidad, J. Vázquez-Salceda, D. Busquets, M. Nicolás, S. López, A. Vázquez, F. Vázquez, and F. Caballero. UCTx: a multi-agent approach to model a transplant coordination unit. In *In Proceedings of the 3rd. Congrés Català d'Intelligència Artificial. CCIA 2000*, October 2000.

[3] U. Cortés, A. López-Navidad, J. Vázquez-Salceda, A. Vázquez, D. Busquets, M. Nicolás, S. Lopes, F. Vázquez, and F. Caballero. Carrel: An agent mediated institution for the exchange of human tissues among hospitals for transplantation. Technical Report LSI-00-33-R, Software Department. UPC, http:/www-lsi.upc.es/dept/techreps/ps/R00-33.ps.gz, 2000.

[4] M. R. Costanzo. Selection and treatment of candidates for heart transplantation. Seminar report on thorax cardiovascular surgery, American Heart Association, Inc, 1995. http://www.americanheart.org/presenter.jhtml?identifier=1298.

[5] Rina Dechter. Constraint networks. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 276–285. Wiley, 1992. Volume 1, second edition.

[6] Fipa. http://www.fipa.org.

[7] Eugene C. Freuder and Richard J. Wallace. Suggestion strategies for constraint-based matchmaker agents. In *Principles and Practice of Constraint Programming*, pages 192–204, 1998.

[8] S. Goddard and D. H. Adams. New approaches to immunosuppression in liver transplantation. *Journal Gastroenterol Hepatol*, 17(2):116–126, February 2001.

[9] N. R. Jennings. Building complex software systems. *Communications of the ACM*, 44(4):35–41, June 2001.

[10] Kerberos 5 release 1.2. http://web.mit.edu/kerberos/www/krb5-1.2/index.html.

[11] The moron's guide to kerberos. http://www.isi.edu/ brian/security/kerberos.html.

[12] David Leake. Case Based Reasoning in context: The present and future. Technical report, 1996.

[13] A. Moreno, A. Valls, and J. Bocio. A multi-agent system to schedule organ transplant operations. *Inteligencia Artificial (Revista Iberoamericana de Inteligencia Artificial), Special issue on Development of Multi-Agent Systems*, (13):36–44, 2001.

[14] National library of medicine. http://www.nlm.nih.gov/.

[15] J. L. Nealon and A. Moreno. The application of agent technology to health care. In *AAMAS02 Workshop on the Application of Agent Technology to Health Care*, 2002.

[16] openGALEN. http://www.opengalen.org.

[17] Protégé-2000. http://protege.stanford.edu.

[18] Meeting the organ shortage: Current status and strategies for improvement of organ donation. Newsletter Transplant, 4(1):5–17, 1999.

[19] SNOMED. http://www.snomed.org.

[20] Jade Open Source Development Team. Java Agent Development Environment. Technical report, Open Source Project, 2001.

[21] U. Thalheimer and F. Capra. Liver transplantation: making the best out of what we have. *Digital Distributed Science*, 47(5):945–953, May 2002.

[22] A. S. Tonnesen. Contents of the Electronic Patient Record (EPR). http://www.google.ch/search?q=cache:ZOOtMRRlFk4C:www.uth.tmc.edu/ atonnese/cprdata.html, 1996.

[23] UMLS. http://www.nlm.nih.gov./research/umls.

[24] A. Valls, A. Moreno, and D. Sànchez. A multi-criteria decision aid agent applied to the selection of the best receiver in a transplant. In *4th. International Conference on Enterprise Information Systems (ICEIS)*, pages 431–438.

Acknowledgment

We are very grateful to Prof. C. Lovis from the Informatics Department of the University Hospital of Geneva for his precious support. Many thanks also to the Swisstransplant team for the valuable input and to the rest of the Whitestein team, in particular to Patrick Brunschwig.

Gotthardstrasse 50, CH-8002, Zurich, Switzerland
E-mail address: `mca@whitestein.com`