MET4: Supporting Workflow Execution for Interdisciplinary Healthcare Teams

Szymon Wilk[1], Davood Astaraky[2], Wojtek Michalowski[2], Daniel Amyot[3], Runzhuo Li[3], Craig Kuziemsky[2], and Pavel Andreev[2]

[1] Institute of Computing Science, Poznan University of Technology, Poland
[2] Telfer School of Management, University of Ottawa, Canada
[3] School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Abstract. This paper describes MET4, a multi-agent system that supports interdisciplinary healthcare teams (IHTs) in executing patient care workflows. Using the concept of capability, the system facilitates the maintenance of an IHT and assignment of workflow tasks to the most appropriate team members. Moreover, following the principles of participatory medicine, MET4 facilitates involving the patient and her preferences in the management process. We describe conceptual foundations of the system and present the system design combining multi-agent and ontology-driven paradigms. Finally, we present a prototype implementation of MET4 and illustrate operations of the prototype using an example involving chronic pain management in palliative care.

1 Introduction

An interdisciplinary healthcare team (IHT) includes healthcare practitioners from different domains/specialties, who work together to achieve a common goal of providing evidence-based care to a patient [14]. In clinical practice, team-based approach is used to manage complex medical cases – this practice has been shown to reduce patient management costs and time, improve service provision and enhance patient satisfaction [2].

Very often patient management by an IHT follows a patient care workflow that is generally defined as a sequence of steps corresponding to clinical tasks and decisions [8]. While executing these tasks, members of an IHT collaborate, communicate by exchanging information about the patient's state and care provided (directly, or through an electronic health record – EHR), and coordinate their activities [15].

The overall goal of our research is to build MET4 – a system to support an IHT in executing a patient care workflow and to operationalize principles of participatory medicine that posit patient's active involvement in a care process. This general goal is decomposed into specific goals that are addressed in this paper, namely (1) developing a conceptual model to facilitate our understanding of the problem at hand, (2) developing a formal design of an agent-based system and (3) implementing a prototype version of the designed system to support execution of the chronic pain management workflow. We build on our earlier research on designing and developing multi-agent clinical decision support systems (CDSSs) [18, 1]. Specifically, we expand the results published in [1] by extending the conceptual foundations of the system to include the concepts of *team leadership* and *patient participation* in the management process and by revising how workflows are represented. We also describe a prototype implementation of MET4 and present its operations.

2 Related Work

Challenges of collaboration and coordination in healthcare and other domains produced intensive research in the area of computer-supported cooperative work (CSCW) [17]. The multi-agent paradigm has shown to be suitable for modeling CSCW because of the inherent cooperative nature of agents, as demonstrated in two projects aimed at designing and developing assisted living applications – K4CARE [10] and CASIS [11].

K4CARE's main objective was to create, implement, and validate a knowledge-based healthcare model (based on patient care workflows) of professional assistance to senior patients at home. While K4CARE supports IHTs, it provides relatively basic support for team management, as the team members have to be explicitly associated with workflow tasks they are capable to execute. CASIS, on the other hand, is an event-driven, service-oriented and multi-agent framework, whose main goal is to provide context-aware healthcare services to elderly residents.

In the area of diagnosis, CDSSs often need the integration of different sources of data and involve the collaboration of different members of a team. The multi-agent paradigm in such a context was successfully used in the OHDS [16] system. OHDS supports physician in establishing a diagnosis using information pertinent to each patient and leveraging bio-medical data contained in various external databases. The system uses agents and ontologies to organize unstructured biomedical data into structured disease information.

A common challenge faced when developing a CDSS is related to integration of patient data with the CDSS engine, as this requires interoperability between existing systems and medical devices. The MobiGuide [7] project aims to address this challenge by providing guideline-based clinical decision support integrated with a personal health record.

3 Conceptual Model of MET4

In this section, we present the conceptual model of MET4 as a description of assumptions, concepts and relations between these concepts. The goal of this model is to facilitate our understanding of workflow execution by an IHT and to lay foundations for the system design.

We assume that an IHT manages a patient according to a presentation-specific workflow (where by presentation we mean a disease, a specific type of trauma, etc.). The team is formed when the patient management needs to start (e.g., at the time of hospital admission after preliminary diagnosis has been made), and it is disbanded when the execution of the workflow is completed.

The team has a *leader* responsible for overseeing the execution of a workflow, for handling exceptional situations and for delegating workflow tasks to suitable team members (i.e., members who are qualified to perform these tasks) – in this sense the leader plays a role very similar to the role of the *most responsible physician* [3]. Moreover, following the principles of *participatory medicine* [9], the leader can consult with the patient before making any relevant decisions, such as selecting a therapy or selecting a team member. While there can be only one team leader at a time, she can change during workflow execution (e.g., at one point a surgeon leads a team, while at some other time she is replaced by an anesthesiologist) – this is similar to what has been proposed in [12].

In the conceptual model we employ a *hybrid team formation* that combines static and dynamic approaches (see [10] for their detailed description). The leader is selected when the workflow starts, and stays with an IHT until the workflow is competed, or the need for a new leader is explicitly indicated by the workflow (static approach). Other team members are recruited dynamically to execute tasks from the workflow and released from the team afterwards. Specifically, a new member is recruited only if the leader is not able to perform the task at hand. Such simplicity is the major advantage of the hybrid approach – there is no need for scheduling or thorough analysis of the workflow. At the same time, it is also a disadvantage, as it is impossible to form a team in a more efficient manner (e.g., to re-use team members for different tasks).

We distinguish between normal and urgent workflow tasks (we assume such information is given as part of the task description) – the former can be temporary suspended, while the latter have to be executed with no delay. If a practitioner qualified to perform an urgent task is busy with a normal task, then execution of this normal task is suspended and the practitioner is recruited for the urgent task.

In order to provide fine-grained characteristics of a practitioner and a task, we introduce the concept of *capability*, which is defined as an ability to perform a certain clinical task [6]. A capability is additionally annotated with a *capability level* that indicates the level of competency in performing this task. Generally, the capability level should be expressed using some ordinal scale – it could be a simple numerical score or a qualitative value. For simplicity, we assume that capability levels are given in the former form.

A practitioner may have multiple capabilities – we refer to these capabilities as *possessed capabilities*. On the other hand, capabilities associated with workflow tasks are called *required capabilities*. A practitioner is able to perform a certain task if she possesses all the required capabilities, and the levels of possessed capabilities at least meet the levels of required capabilities.

Obviously, there may be many practitioners capable to perform a given task. In such situation, a team leader should consult the patient in order to evaluate and rank possible candidates. If the patient does not have preferences, or there is a tie between the candidates, then additional criteria need to be employed. For example, the leader can select the practitioner with the best capabilities (i.e., capabilities with the highest levels), or take the one with the lowest workload.

The concept of required capability is also used for defining requirements that need to be satisfied by a team leader. Specifically, we assume a patient care workflow starts with a leader appointment task that is associated with the capabilities required from the leader. Unlike the selection of a team member, the leader selection relies solely on capabilities possessed by the practitioners considered for this position, and patient preferences are not taken into account. Moreover, if a workflow invokes another workflow, and this invoked workflow involves appointment of a new leader, then the previous leader is temporary suspended, and then restored when the invoked workflow terminates.

In order to fully conform to principles of participatory medicine that advocate integration and equal participation of IHT members and patients or family members [9], we introduce the concept of *patient representative* that complements the concept of *patient*. A patient is a passive receiver of care and a "provider" of clinical data, whereas a patient representative represents a patient and actively participates in all decision-making activities that affect patient care. While in most cases the patient and the patient representative are the same person, there are situations where such distinction is necessary, e.g., during the management of a pediatric patient when the parent or legal guardian become a representative. Also we assume that there is a single patient representative associated with the patient and the team.

4 Design and Implementation of MET4

In this section we show how the conceptual model of MET4 has been formalized into the system's design, and how the design has been implemented as a prototype system. The design was created using Organization-based Multiagent System Engineering (O-MaSE) – a flexible methodology for analyzing and designing multi-agent systems (MASs) [5]. O-MaSE starts with system specifications (conceptual model in our case) and iteratively translates them into a number of models (domain model, goal model, agent class model, protocol model, plan models) that describe various aspects of a MAS and that drive subsequent implementation.

Due to the space limit, we present below only the domain and agent class models – the remaining models present hierarchy and dependencies between goals to be achieved by the system (goal model), communication protocols employed by agents (protocol models), and plans and algorithms employed by the agents to achieve goals (plan models). A detailed description of O-MaSE and its clinical application (designing a CDSS for patient management in the emergency department) is given in [18].

4.1 Domain Model

The domain model provides an ontology (Fig. 1) formalizing the system's domain knowledge. This ontology should be considered as a schema to construct ontological models that represent specific workflows, practitioners and patients. The ontology is divided into three areas that define concepts and relations associated with patients, teams and workflows. These areas overlap – in particular the concepts of capability and presentation are shared between the areas.

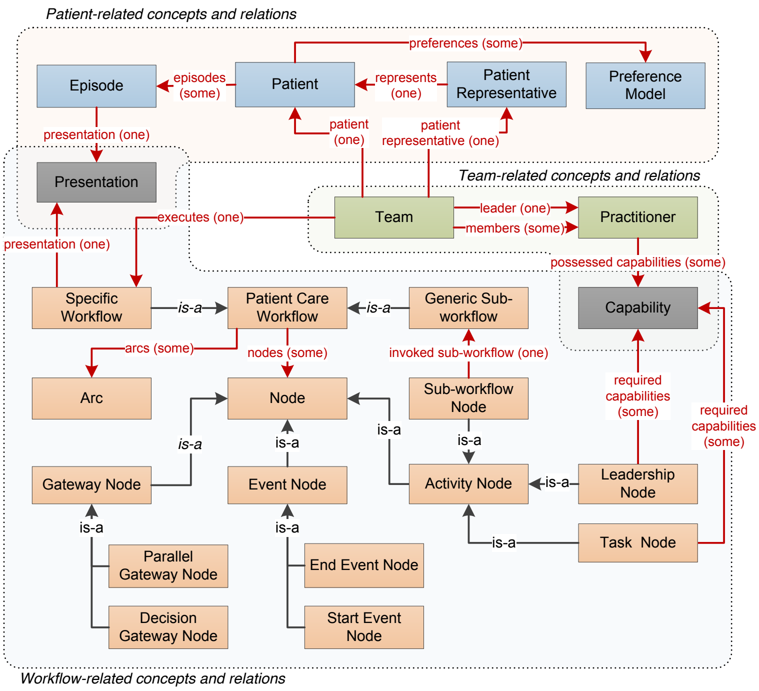

Fig. 1. Domain ontology

Most of the patient- and team-related concepts have been described in Section 3. A new concept here is *preference model*, which captures patient's preferences with regards to a specific type of decision (for example, a patient may have one preference model for evaluating a therapy and another for evaluating interventions). We assume preference models are represented using various formats (functional, relational and rule-based) and they are employed by a patient representative.

The workflow-related part of the domain ontology is inspired by the Business Process Model and Notation (BPMN), and it has been tailored to address the particularities of IHTs (we could not directly use BPMN because it does not allow for representing capabilities). The central concept is *patient care workflow*, specialized into *specific workflow* and *generic sub-workflow*. The former represents a "top-level" patient care workflow that is associated with a specific presentation. The latter is a generic workflow (i.e., not associated with any presentation) that can be shared and re-used (i.e., invoked by other workflows).

Each patient care workflow is composed of *arcs* and *nodes*, so it can be represented as a directed graph. The node concept is specialized into *gateway node*, *event node* and *activity node*, depending on its purpose. Gateway node is further specialized into *decision gateway node* and *parallel gateway node*, which allow for conditional branching and parallel paths, respectively. Event nodes are used to indicate starting and ending points in a workflow (by *start event node* and *end event node*). Finally, activity node is specialized into *task node*, *sub-workflow node* and *leadership node* corresponding to three types of basic activities in a workflow – executing a clinical task, invoking a sub-workflow and appointing a team leader, respectively.

4.2 Agent Class Model

An agent class model defines what types of agents are needed to achieve the goals, and it shows interactions between the agents and actors (users, external systems and other entities, such as repositories or databases). The agent class model derived for MET4 is given in Fig. 2. We need to note that several goals associated in the conceptual model with a leader (e.g., supporting team maintenance and overseeing workflow execution) have been offloaded to autonomous agents classes (*team manager* and *workflow executor*) in order to minimize the practitioner's workload associated with the leadership activities.

Most of the agent classes introduced for MET4 have been already described in detail in [18, 1], therefore, we focus here on new or expanded ones. The *patient representative* agent class has been introduced to act on behalf of the patient or guardian, it manages a set of patient's preference models and uses them to evaluate possible decision choices. The *practitioner assistant* class has been extended to achieve the goals associated with consulting decision choices with a patient representative whenever necessary. Finally, the *team manager* class has been modified to support leaders' management (i.e., maintaining lists of active and temporarily suspended leaders, identifying practitioner assistant agents, who can be team leaders, informing selected agents about appointment, and informing leaders about their status). It uses information from *yellow pages* – a shared directory where agents from the *practitioner assistant* class register their possessed capabilities and availability (i.e., free, busy with a normal task, busy with an urgent task).

4.3 Prototype Implementation

The prototype of MET4 has been implemented using *Workflows and Agent Development Environment* (WADE, http://jade.tilab.com/wade). WADE allows for developing, deploying and running multi-agent systems, where agents cooperate and coordinate their activities according to a set of workflows. It provides

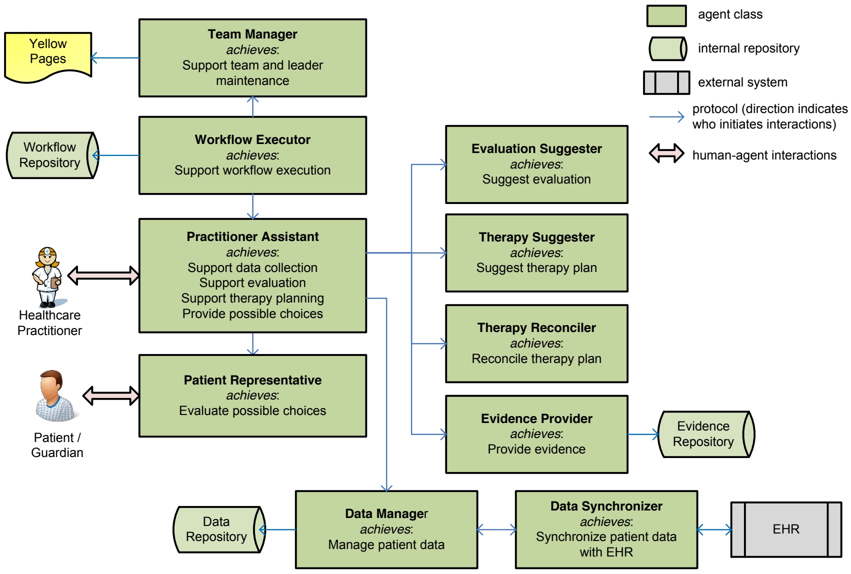

Fig. 2. Agent class model for MET4

Java programming libraries, visual workflow development tools and an execution environment acting as a middleware for developed systems.

The domain model (see Fig. 1) has been implemented and stored in Protégé (http://protege.stanford.edu). Protégé is also used as a knowledge base to store ontological models representing specific workflows, specific practitioners and patient data collected using MET4.

Finally, as indicated in Fig. 2, we assume MET4 interacts with an EHR available in a given healthcare organization to exchange patient data. For the prototype implementation, we used the OpenMRS system [19], an open-source and web-based EHR. OpenMRS and MET4 (specifically the *data synchronizer* agent) communicate using REST web services and JSON for encoding exchanged information. Data integration is achieved through the mechanism of concept maps that allow associating data concepts used in OpenMRS with concepts employed in other systems.

5 Illustrative Example

5.1 Chronic Pain Management in Palliative Care

Palliative care aims at improving the quality of life of the patient, who suffers from terminal diseases, by managing symptoms (e.g., pain, nausea or fatigue), coordinating care, and providing emotional and spiritual support [13]. Chronic pain management is a significant component of any palliative care management protocol, as it is estimated that 70% of advanced cancer patients experience pain

[4]. The complexity of chronic pain management in terminal patients necessitates that care delivery is provided by a team of practitioners from different specialties (e.g., physicians, psychologists, and physiotherapists, nurses, and occupational therapists).

Building on existing pain management guidelines and consulting with the domain experts, we created a set of workflows to structure all activities associated with assessing, diagnosing, and managing pain for a palliative care patient. These workflows were implemented in WADE, and the top-level workflow and selected sub-workflows are illustrated in Fig. 3. This figure also presents required capabilities associated with the leadership and task nodes.

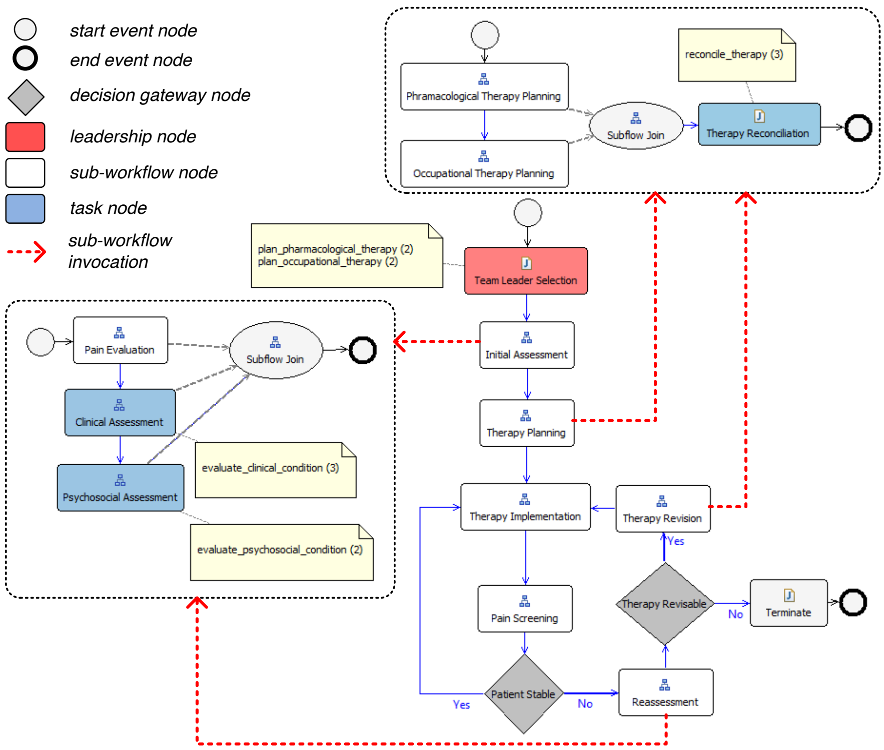

Fig. 3. Chronic pain management workflow

The workflows in Fig. 3 contain no parallel paths (and thus no parallel gateway nodes), because such paths are not directly supported by WADE. Fortunately, they can be easily simulated as a sequence of asynchronous activity nodes. Such a sequence requires a *subflow join* node that is WADE-specific and therefore has not been considered in the domain model (it may become unnecessary if we use another workflow-related technology). For example, *pain evaluation*, *clinical assessment* and *psychosocial assessment* in the initial assessment sub-workflow are actually all executed in parallel, although it is not evident from the WADE-specific representation.

5.2 Supporting Chronic Pain Management by MET4

To illustrate operations of MET4 we use three scenarios: team leader selection, team maintenance and therapy reconciliation. In all these scenarios we assume a configuration of the MET4 system presented in Fig. 4, where we use "*label : agent class*" as a notation to describe specific agents. For the sake of readability, the figure does not show the decision support agents (they are irrelevant in the considered scenarios).

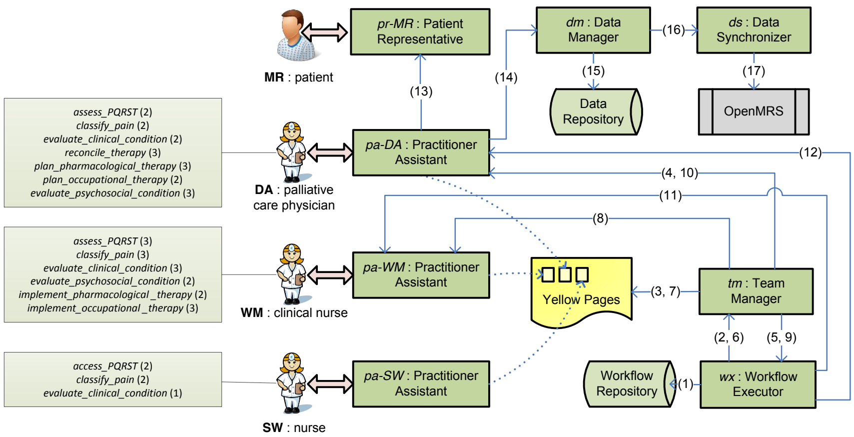

Fig. 4. Sample configuration of MET4 for chronic pain management

Figure 4 also shows interactions in the system occurring in the considered scenarios. Patterns associated with selected interactions (e.g., between the *tm* agent and the *yellow pages*) have been established during the design phase and are embedded in agents' plans. However, the overall sequence of interactions depends on a particular workflow and available practitioners, thus it is established during run-time.

For simplicity we limit the set of available practitioners to three, identified in Fig. 4 as *DA*, *WM* and *SW*. They are further described with possessed capabilities, and each has a corresponding practitioner assistant agent (*pa-DA*, *pa-WM* and *pa-SW*, respectively) described with the same capabilities as its human owner and registered in the *yellow pages*.

Finally, we assume a patient (*MR*), who has a *patient representative* agent (*pr-MR*) with a preference model for evaluating therapies – so this is a situation, where a patient and a patient representative is one person. The preference model is realized as a value function that computes an overall score for possible therapeutic options. The patient has no other preference model, which means he is indifferent to choices and options associated with other decisions, including selection of team members.

Team Leader Selection Scenario. Once the *MR* patient has been admitted, the *wx* agent retrieves from the *workflow repository* the patient care workflow for managing chronic pain ((1) in Fig. 4). The first activity node involves appointing a leader. In order to do this, the *wx* agent sends a request to the *tm* agent to find a practitioner assistant agent that possesses the capabilities required for leading the IHT (2), so the corresponding practitioner may become the leader. The *tm* agent consults the *yellow pages* for the information about available agents and their capabilities (3). The only agent that possesses the required capabilities is *pa-DA*, therefore, it is notified about being selected (4), and this choice is reported to the *wx* agent (5).

Team Maintenance Scenario. Here, the *wx* agent proceeds to the next activity node in the workflow, which is the initial assessment. Execution of this activity involves invoking a sub-workflow. One activity in this sub-workflow is *pain evaluation*, which we assume has been already completed during previous visit, and the results have been recorded in OpenMRS. The other two tasks nodes (*clinical assessment* and *psycho-social assessment*) need to be executed in parallel (see Fig. 3).

First, the *wx* agent sends a request to the *tm* agent to find two practitioner assistant agents that could perform the above tasks (6). According to the capability-based selection procedure, *pa-DA*, who is already the team leader, qualifies for *psychosocial assessment*. However, for the *clinical assessment* task node, the *tm* agent has to search for a qualified candidate in the *yellow pages*, where *pa-WM* is identified as potential team member (7). Then, *tm* asks *pa-WM* to confirm that it agrees to be part of the team (8). Assuming that practitioner *WM* accepts this request, the information about these two agents is sent back to *wx* by *tm* (9). Then, *pa-DA* and *pa-WM* are sent requests to complete the two tasks discussed here (10 and 11, in any order) and these agents through user interface (see Fig. 5) prompt practitioners *DA* and *WM* to collect patient data.

Therapy Reconciliation Scenario Here, we assume that the *pa-DA* agent has already been selected for the *therapy reconciliation* task. First, the *wx* agent sends a request to *pa-DA* to perform this task ((12) in Fig. 4). In response, *pa-DA* checks for possible interactions between pharmacological and occupational therapies derived earlier and constructs possible combined therapies. Then, it communicates with the *pr-MR* agent to learn about patient's preferred combined therapy (13). This is accomplished by applying the preference model to possible therapies and reporting resulting scores. The *pa-DA* agent selects the therapy with the highest score and reports this choice to the *dm* agent (14). The *dm* agent stores the selected combined therapy in the *data repository* (15) and notifies the *ds* agent (16), which in turn notifies OpenMRS about the update (17).

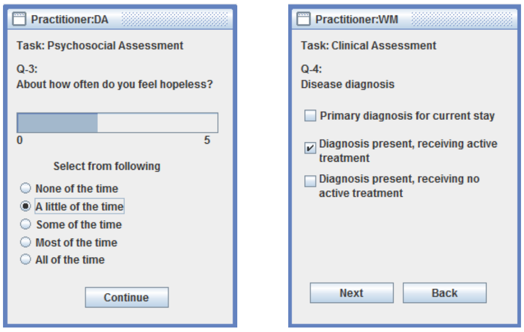

Fig. 5. Sample user interface presented to practitioners *DA* and *WM*

6 Discussion

We have presented conceptual foundations, design, implementation and operations of the MET4 system that supports an IHT in executing a patient care workflow. MET4 maintains a team and its leader and assigns workflow tasks to the most qualified team members. This is accomplished by employing capabilities to characterize practitioners and workflow tasks and by following a hybrid strategy to maintain the team.

MET4 also implements the principles of participatory medicine by introducing the concept of a patient representative and a corresponding agent class. Such solution allows for including patient preferences in the management process. Moreover, in order to support communication and awareness of practitioners, the system manages data collected during management and synchronizes it with an EHR (OpenMRS).

As part of future research we plan to enrich the team maintenance strategy by accounting for situations where a group of practitioners has to manage multiple patients at the same time – in such case the global allocation of practitioners has to be considered. Moreover, we are going to develop more sophisticated plans for the practitioner assistant agent that will allow handling exceptional and ad-hoc situations in automatic or semi-automatic manner. We are also working on considering patient preferences to modify the structure of a workflow. Finally, to ensure data security and privacy, we plan to adopt access control techniques described in [12] and to combine them with the team maintenance strategy.

References

1. Astaraky, D., Wilk, S., Michalowski, W., Andreev, P., Kuziemsky, C., Hadjiyannakis, S.: A multiagent system to support an interdisciplinary healthcare team: a case study of clinical obesity management in children. In: Proceedings of the VIII Workshop on Agents Applied in Health Care. pp. 69–80. Murcia, Spain (2013)
2. Borrill, C., Carletta, J., Carter, A., Dawson, J., Garrod, S., Rees, A., Richards, A., Shapiro, D., West, M.: The effectiveness of health care teams in the National Health Service. University of Aston, Birmingham (2001)

3. Capen, K.: When patient care is shared, who is the most responsible physician? CMAJ 154(6), 885–886 (1996)
4. Franks, P.J., Salisbury, C., Bosanquet, N., Wilkinson, E.K., Kite, S., Naysmith, A., Higginson, I.J.: The level of need for palliative care: a systematic review of the literature. Palliat. Med. 14(2), 93–104 (2000)
5. Garcia-Ojeda, J.C., DeLoach, S.A., Oyenan, W.H., Valenzuela, J.: O-MaSE: a customizable approach to developing multiagent development processes. In: Agent-Oriented Software Engineering VIII. vol. 4951, pp. 1–15. Springer (2008)
6. Gardner, A., Hase, S., Gardner, G., Dunn, S.V., Carryer, J.: From competence to capability: a study of nurse practitioners in clinical practice. J. Clin. Nurs. 17(2), 250–258 (2008)
7. González-Ferrer, A., Peleg, M., Verhees, B., Verlinden, J.M., Marcos, C.: Data integration for clinical decision support based on openEHR archetypes and HL7 virtual medical record. In: Proceedings of the 2012 International Conference on Process Support and Knowledge Representation in Health Care. pp. 71–84. Springer (2013)
8. Grando, M.A., Peleg, M., Cuggia, M., Glasspool, D.: Patterns for collaborative work in health care teams. Artif. Intel. Med. 53(3), 139–160 (2011)
9. Hood, L., Auffray, C.: Participatory medicine: a driving force for revolutionizing healthcare. Genome Med. 5(12), 110 (2013)
10. Isern, D., Moreno, A., Snchez, D., Hajnal, A., Pedone, G., Varga, L.Z.: Agent-based execution of personalised home care treatments. Appl. Intell. 34(2), 155–180 (2011)
11. Jih, W.r., Hsu, J.Y.j., Tsai, T.: Context-aware service integration for elderly care in a smart environment. In: Proceedings of the 2006 AAAI Workshop on Modeling and Retrieval of Context Retrieval of Context. pp. 44–48. AAAI Press, Menlo Park, CA (2006)
12. Le, X., Doll, T., Barbosu, M., Luque, A., Wang, D.: An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow. J. Biomed. Inform. 45(6), 1084–107 (2012)
13. Morrison, L.J., Morrison, R.S.: Palliative care and pain management. Med. Clin. North. Am. 90(5), 983–1004 (2006)
14. Oandasan, I., D'Amour, D., Zwarenstein, M., Barker, K., Purden, M., Beaulieu, M., Reeves, S., Nasmith, L., Bosco, C., Ginsburg, L.: Interdisciplinary education for collaborative, patient-centred practice: research and findings report. Health Canada, Ottawa (2004)
15. Ruan, J., MacCaull, W., Jewers, H.: Enhancing patient-centered palliative care with collaborative agents. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). vol. 3, pp. 356–360. IEEE (2010)
16. Ulieru, M., Hadzic, M., Chang, E.: Soft computing agents for e-health in application to the research and control of unknown diseases. Inform. Sci. 176(9), 1190–1214 (2006)
17. Weerakkody, G., Ray, P.: CSCW-based system development methodology for health-care information systems. Telemed. J. E Health. 9(3), 273–282 (2003)
18. Wilk, S., Michalowski, W., OSullivan, D., Farion, K., Sayyad-Shirabad, J., Kuziemsky, C., Kukawka, B.: A task-based support architecture for developing point-of-care clinical decision support systems for the emergency department. Methods Inf. Med. 52, 18–32 (2013)
19. Wolfe, B.A., Mamlin, B.W., Biondich, P.G., Fraser, H.S., Jazayeri, D., Allen, C., Miranda, J., Tierney, W.M.: The OpenMRS system: collaborating toward an open source EMR for developing countries. AMIA Annu. Symp. Proc. p. 1146 (2006)