

Process-based Multi-agent Architecture in Home Care

B. Bořanský¹

¹Dept. Of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague

Supervisor: Doc. Ing. Lenka Lhotská, Csc.

Summary

Utilization of procedural knowledge in the form of organizational processes and formalized medical guidelines can be useful in decision support systems (DSSs) in health care domain. The problem of using this form of knowledge arises when a multi-agent paradigm is to be applied in a DSS due to differences in specification of behavioural models of agents and process formalisms. In this work we continue in enhancing a novel process-based multi-agent architecture and demonstrate its integration into an existing DSS (K4care) focused on home care.

We analysed available documentation of the complex system K4Care and identified possible mutual common functionalities of implemented multi-agent system with the new architecture. These were the entry points, using which we further enhanced the K4Care platform with respect to the process-based multi-agent architecture.

The analysis proved that the integration is not only possible, but thanks to the general design of the process-based multi-agent architecture can be done with only small changes in the existing K4Care model. Immediate improvements in supporting human experts were identified and possible further improvements of the system were discussed.

Adopting the process-based multi-agent architecture can be beneficial even for existing DSSs and can open new possible features emerging from the multi-agent paradigm.

Keywords: multi-agent systems, multi-agent architecture, organizational processes, formalized medical guidelines, health care, home care, K4Care

1. Introduction

Decision support systems (DSSs) have established their place in the field of health care and assist physicians with their decisions. Several studies proved that

decision support systems (DSSs) can be useful for preventing errors, or improving the quality of provided care, they can reduce costs and save time [1-5]. The decision support systems can be helpful in practice only when they offer unobtrusive interface to users, such as physicians and/or other health care specialists. This observation is proved in a systematic review of existing DSSs [2], where systems indicated as most helpful were working transparently in the background and automatically prompted the user in case of a reminder or an alert (e.g. systems [6], [7]). The knowledge used by DSSs is usually based on medical guidelines (e.g. in systems [8], [9]) that describe a recommended procedures for diagnosing specific diseases and/or their treatment. However, based on the analysis of existing DSSs [1-9] we conclude that a successful DSS is fully integrated into a medical facility where it is used in order to fit current circumstances. Therefore organizational processes must be considered when creating the procedural knowledge for a DSS. Such a procedural knowledge can then be used by a DSS for process validation, simulation or verification of their execution in practice.

One of the fields of health care where applying DSSs can be highly beneficial, is the domain of Home Care (HC) [10], [11]. Growing number of chronic elderly patients result in growing need for a lifelong treatment under continuous health care specialist supervision. Moreover placing the patients at their home environment can have better results compared to improper effects of institutionalization [11]. As the home care domain is distributed in nature, there is a need for appropriate DSS. Health care specialists visit patients at home in order to provide appropriate assistance, however, they cannot easily consult specific questions related to unfamiliar situations or complications. Therefore,

a DSS supporting and monitoring their actions can improve provided care.

The distributed nature of the home care have to be reflected by a DSS as well, therefore a multi-agent paradigm is often used [11-15]. As a multi-agent system we term a system where a group of autonomous intelligent agents jointly interact in order to achieve predefined goals. The behaviour of an agent can be specified using various behavioural models, such as (1) purely reactive models where the agent only reacts to the changes present in the environment, (2) deliberative models, where the agent create its own model of the environment and reason about its future actions, and (3) hybrid models, combining the two previous models. These behavioural models can be formalised using general programming languages, such as Java, or they can be formalised using some agent programming languages, such as Jazzyk [16], JASON [17], or versions of the APL language [18]. The problem of existing agent programming languages is that they offer only limited capabilities of effective modelling of implicit cooperation and coordination between agents in order to make them concurrently follow both private and common goals that require cooperation. On the other hand, organizational processes and medical guidelines are highly based on cooperation of involved actors and provide the information about the order and conditions for execution of specific activities (e.g. perform an examination prior to a surgery, which also need specific equipment, etc.). Therefore in the previous work we defined a new way of creating the behavioural models for intelligent agents based on a novel multi-agent architecture that would directly support such cooperation [19], [20].

The approach is based on enhanced processes, in which a procedural knowledge can be formalised. Based on this information we create a set of agents in a specific architecture that are able to work with these processes in terms of their validation, simulation, or verification of their execution and critiquing in practice. For explanatory reasons we refer to this architecture as ProMA in this work.

The developed architecture (ProMA) is designed to work with general process-like formalization. Hence the goal of this paper is to demonstrate a possible application of ProMA for extending an existing architecture for a home care system K4Care [11], where the formalism SDA* [21] is used for storing the procedural knowledge. We analyze the impact of an integration of a new multi-agent architecture into the K4Care system in terms of necessary model changes and new features. The paper is organized as follows. In the next section we introduce the concept of processes in health care, following by the description of the ProMA architecture that is able to work with them. Next section is devoted to the brief description of the K4Care system, used architecture and functioning of the system. Following sections describe the integration of the ProMA architecture into the K4Care system. This integration acts as a proof of concept of applicability of ProMA. Finally, we propose possible new features for the K4Care system based on the integration, and we conclude our approach in the final section of the paper.

2. Process-based Multi-agent Architecture (ProMA)

In this section we define processes in health care, identify two basic types of the processes, and describe the ProMA architecture that is able to work with them. We understand the term process to be a sequence of activities, states, decision points, steps splitting the sequence, and synchronization points. Processes in health care can be divided into two main categories: organizational processes and medical treatment processes (medical guidelines).

The organizational processes in the health care domain are closely related to other

business areas. There are several studies (e.g. in [22], [23]) that practically examine problems of applying process modelling or workflow management systems within the health care domain. There is a mutual agreement that a successful implementation of this approach can improve the quality of health care, reduce the time of necessary hospitalization and lead to the reduction of costs.

Medical guidelines are for some time now a part of the process of standardization of medical treatment and act as pillars of evidence-based medicine. Guidelines contain recommended actions, directions, and principles for diagnosis or treatment specific diseases approved by appropriate expert committees. However, the work-methods regarding the guidelines (such as consulting or using in practice) are still usually based on a textual form, which is natural for health care specialists. The textual form is, however, not sufficient for a quick consultation during an examination of a patient, or an identification relevant changes in the new version of the document. Moreover, guidelines in the textual form cannot be directly used for computer interpretation, or as a knowledge base for decision support systems. Therefore a wide part of research in biomedical informatics is related to the formalization of medical guidelines into an electronic form and various process

languages are used in order to capture the knowledge of a textual medical guideline into an electronic and structured form (e.g. PROforma [24], Asbru [25], GLIF [26], etc.).

Both of these types can be jointly seen as processes in health care and have to be reflected in the procedural knowledge base of a DSS. In following we describe the architecture of a multi-agent system that is able to work with procedural knowledge captured using a generic process-like formalism. The scheme of the architecture is depicted in Figure 1.

2.1 Environment Agent

An agent-based simulation is situated in some environment that is represented by the Environment Agent in the ProMA architecture. With respect to the level of detail that we want to model using the system, the environment could represent a virtual world (e.g. database of patients' health records, a whole department of a hospital, etc.) with existing objects (e.g. laboratory results, RTG or EEG machines, beds, etc.). There is one difference between handling data from patient's electronic health record (EHR) and other objects in the environment. As it is shown in the figure, the agent managing the patients' EHR is requested by the Environment Agent as patients' data certainly are a part of environment.

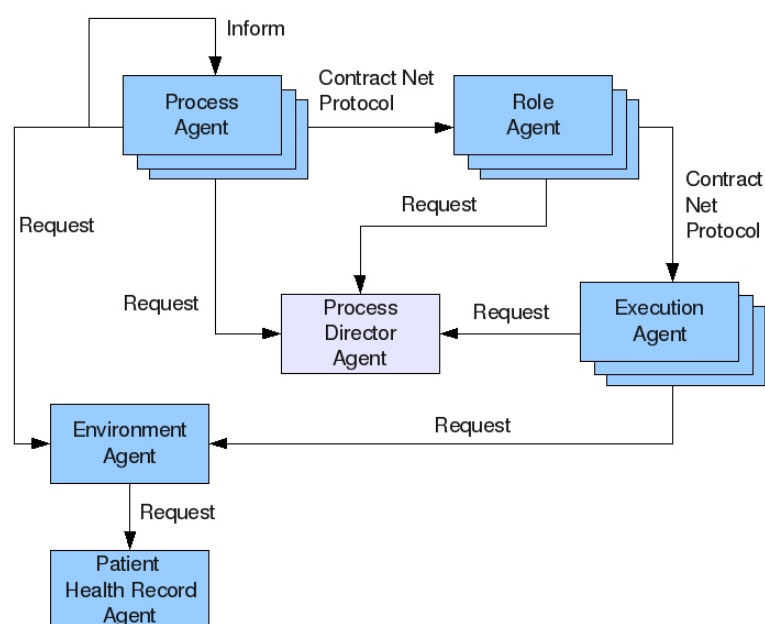


Fig. 1. Scheme of the process-based multi-agent architecture.

However, as it would be described in the next sections, there could be a need for a distinction between real data stored in the record (Patient Health Record Agent) and data used for a simulation (EnvironmentAgent).

2.2 Execution Agent

Execution Agents (EA) represent concrete physicians, nurses, patients, and/or possibly other employees of a facility involved in the processes. These agents are based on a reactive architecture in the form of hierarchical rules, which can be automatically generated based on possible activities that the specific EA can participate in (see [20] for details on the transformation algorithm). Each EA has several predefined rules that cover basic behaviour in the environment (i.e. responding to messages sent by other agents, sending appropriate messages to the Environment Agent during the execution of the activity, etc.). Then, for each activity that the agent (hence the represented person) can participate in, one additional rule is generated. These rules can be activated (when the conditions for the process execution are met, and the EA can perform this action) or deactivated (execution of this process is no longer possible) by a message sent by appropriate a Role Agent (see below). Finally, the Execution Agent autonomously chooses which of the activated processes it executes based on its internal priority.

2.3 Role Agent

Role Agents (RAs) represent the roles in the environment (i.e. general roles for patient, nurse, physician, etc.). RA receives a proposal from a Process Agent (see below) and finds appropriate Execution Agent(s) (EA). Special agents for roles reflect the typical hierarchical structure of roles at a workplace (e.g. a secretary, a nurse, or a doctor are all also employees, etc.). Therefore, when a RA receives a proposal from a Process Agent, it starts to find the appropriate EA between agents that possess this role (using a contract-net protocol (CNP)), but also roles, that are more general in the hierarchy.

If multiple EAs are able to execute given activity and only one is needed, the RA will select the best of them according to its

internal rules and send the message activating appropriate rule to the selected EA. The method of selecting an EA is always domain or role dependent (e.g. the EA that is idle for the longest time can be chosen, etc.). Moreover, RAs are responsible for finding another EA in case the current one needs to suspend its participation on currently executed activity (e.g. one physician is needed in a case with a higher importance and he/she can be replaced in the current one, etc.).

2.4 Process Agent

For every step in the process notation (i.e. for each activity, event, decision point, etc.) there is one Process Agent (PA) created in the system. PA(s) are responsible for a proper execution of the associated step. At first, a Process Agent checks whether the initial conditions for the process are met:

- whether the previous PA(s) has successfully finished its execution (i.e. whether the PA has received the inform message from its predecessor(s)),
- whether all input objects have correct values (the PA uses a simple request protocol to retrieve these data from the EnvironmentAgent),
- whether there exist appropriate agents that will execute this action (the PA uses a CNP for those RAs that are connected to this activity).

If all mandatory conditions hold, the PA starts the execution of the step (e.g. simulation of an activity, a calculation, or a decision process, etc.) and after a successful finish, the PA is responsible for notifying the Environment Agent about results of the activity (using simple request protocol) and the next succeeding Process Agent about a successful finish (using simple inform protocol).

Our approach takes into account a possibility of a temporal suspension of activities and reflecting partial results in the environment, replacing executing EA with another, coordination of several EAs participating on a single activity, or optional input conditions.

2.5 Process Director Agent

The last agent is an auxiliary agent that reads formalized processes, creates other

agents, and answers them to their requests regarding information about the processes (e.g. input data, predecessors, etc.). Note that this is only an implementation issue and all agents reason in terms of processes.

3. K4Care

The aim of the K4Care project [11] was to create, implement, and validate a knowledge-based health care model for the professional assistance to senior patients at home. The procedural knowledge is represented by clinical guidelines, termed within this project as Formal Intervention Plans (FIPs). However, in order to capture specific conditions, in which each patient is present, the K4Care project introduces Individual Intervention Plans (IIPs) that merge both medical and social characteristics of the patient with the available medical knowledge that is represented by the FIPs.

All procedural knowledge related to both types of Intervention Plans (Formal and Individual) is represented in K4Care using the SDA* language [21]. The language has a process-like structure, in which there are three main components (called terms): (1) state terms, (2) decision terms, and (3) action terms. State terms are used to describe feasible patient conditions or situations in the area of interest (e.g., terms as high-blood-pressure are used to emphasize the state). Decision terms represent the terminology that health care professionals use to differentiate possible alternatives (e.g. term antecedents-of-heart-problems would derive the course another direction if it is true). Finally, the action terms identify the medical, clinical or management activities performed by actors (e.g., actions like take-beta-blocker, avoid-salt-in-meals, make-blood-analysis, etc.). Compared to typical process-like formalisms, the actors, as a specific component, are missing in the SDA* formalism. There is, however, a fixed set of actors assigned to each action. More specifically there are two sets of actors - the set of *petitioners* and the set of *performers*. Actors in the first set are allowed to request the related action to be executed. Actors in the set of performers are those persons that are allowed to execute the action.

Formalized procedural knowledge is used by human experts (physicians, or medical centre managers) at first. In order to provide a new service to a request made by a patient registered in the system, a team of specialists identify possible diseases based on the signs and symptoms of the patient. The platform automatically retrieves a set of Formal Intervention Plans (FIPs) that considers all of identified factors. All of these FIPs are merged together and modified by a human expert in order to cover specific circumstances creating this way an Individual Intervention Plan (IIP). In an IIP some other actors can be involved, hence their approval is necessary. After the approvals of the procedure, the responsibilities resulting from the IIP are generated by the SDA* execution engine and specific tasks are generated for involved actors. After a health care specialist executes specified task and he/she fills a necessary data into a document related to the action, and another task is generated according to the IIP.

4. Integration of ProMA into the K4Care System

In this section we provide a description of an integration of the existing K4Care model and the ProMA architecture.

We describe necessary conceptual and technical changes at first, following by a proposition of further capabilities of the modified K4Care system. In this section we follow the typical use-case of the K4Care system described at the end of the previous section and illustrate the integration within this context. An immediate advantage of implementing the ProMA architecture into the K4Care system is in improved decision support for a human expert creating a new IIP for a patient. However, further possible improvements are available as we discuss in the next section.

4.1 Environment

We are focused on the multi-agent system part in the K4Care architecture (see Figure 2). First of all, we unify existing functionality of the K4Care system with ProMA.

The connection to the health record data and procedural knowledge of the multi-agent system represented in ProMA as the Patient Health Record Agent and the Process Director Agent is in K4Care realized by the Data Abstraction Layer (DAL). This layer provides Application Program Interface (API) for requests about patient's data from the health record or guidelines formalized in SDA*.

However, the Environment Agent must still be present in order to temporarily store simulation-based data (that are not persistent). The DAL in the current version is not able to satisfy partial request on the structure of the formalized processes as initialized by Process or Role Agents in ProMA (DAL was designed only for retrieving complete documents). Therefore, both, the Process Director Agent and Patient Health Record Agent, should be created in the multi-agent system as well in order to respond to these requests.

4.2 Execution Agents

The Execution Agents (EAs) in ProMA conceptually correspond to Actor Agents in the K4Care system - they represent the actual users of the system and executors of the actions. In order to fit the ProMA model completely, Actor Agents must be able to communicate with Role Agents (RAs), and autonomously react on their requests for participation in an activity. Note, that in the current version of the K4Care system providing this response is solely based on the activity of human users.

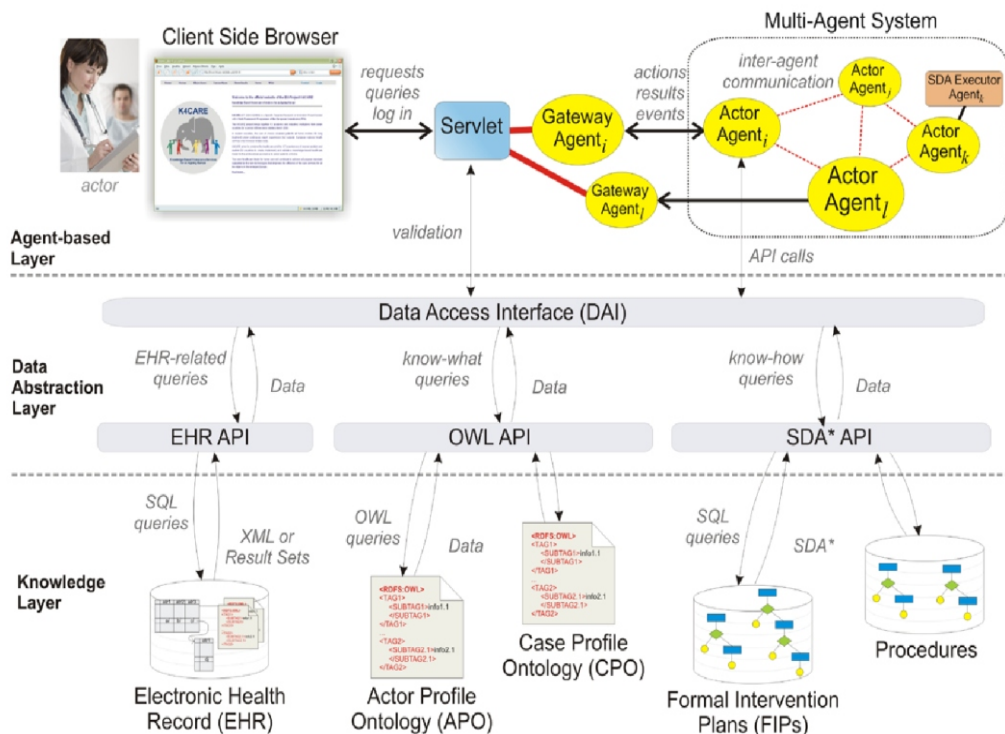


Fig. 2. Scheme of the K4Care system as presented in [13].

However, due to the persistent presence of each Actor Agent in the system, their functionality can easily be enhanced with the ability to respond to the contract-net protocol (CNP) initiated by Role Agents (RAs). The respond can be based on currently associated tasks and/or availability of the associated user (e.g. if a medical specialist has left the facility for a longer time, the Actor Agent could possess this information and automatically deny his/her participation on proposed activities). Moreover, the SDA* concepts typically have the notion of time-related information stored (i.e. time interval and step of the occurrence for a state or an activity), hence Actor Agents can provide much more sophisticated answers based on currently planned actions and the associated IIPs, and act as a basic time-scheduling algorithm easing a human expert when creating a new IIP for a patient (see the next section for new possible features).

4.3 Role Agents

The rest of the types of agents presented in the ProMA architecture do not have their alternative in the K4Care model, hence they have to be created. In the following we describe their full integration into the K4Care system. Note that for Role Agents (RAs), neither Process Agents (PAs) is necessary to be persistent in the system within proposed use-case of creating an IIP (only relevant roles and processes should be created based on requested FIPs).

Role Agents (RAs) conceptually correspond to types of actors in the K4Care system. For each actor type we create one RA as follows: Patient, Family Doctor, Head Nurse, Physician in Charge, Social Worker, Nurse, Specialist Physician, Social Operator, Continuous Care Provider, and Informal Care Giver. Moreover, some of these types belong in the K4Care system into groups Evaluation Unit and Additional Care Givers. Therefore for each of these groups one RA is created as well and a hierarchy between the RAs is established. Moreover, further hierarchy can be seen in various activities (most of the activities of a nurse can be executed by a head nurse as well, etc.).

4.4 Process Agents

Finally, for each state, decision or an activity in requested FIP SDA* models one Process Agent (PA) is created in the multi-agent system. When the IIP is being created by the user, for each step one PA is created as well. The collective behaviour of the PAs during the execution of the IIP by health care specialists correspond to the behaviour implemented the SDA* Execution Engine. However, the advantages of using PAs for each step in the SDA* procedure is during the process of creation of a new IIP by merging the requested FIPs. This process can be partially automated, hence offer some preliminary results that a human expert can use and further improve according to his/her knowledge.

All of the characteristics of the SDA* steps in the K4Care system are formalized using an exhaustive ontology structure, hence we can utilize these information and try to automatically merge some of the steps from requested FIPs in order to simplify the final model. The equivalence of several steps can be identified by appropriate Process Agents (PAs) comparing associated terms and context. If they match, a single PA can be created in the IIP with the union of needed documents and output data (e.g. a blood-analysis can be performed only once with a union of desired information). Although the idea of semi-automated merging whole SDA* procedures was briefly introduced in [13], we propose merging of separate steps of the procedures using autonomous agents. This can be further improved by using a basic simulation (will be described below) of all of the processes concurrently (which ProMA offers). The PAs can identify the order in which they can be executed based on fulfilment of their prerequisites. By the term "basic simulation" we mean a simulation, where the PAs try to simulate associated steps by temporarily filling necessary data fields of the patients' data profile with varying arbitrary values and analyse which of following PAs have fulfilled their prerequisites. We further improve this insight in the next section by more advanced simulation.

Finally, using the CNP Protocol, Process Agents (PAs) can delegate a problem of selecting appropriate health care specialist

to associated Role Agents (RAs). Request for a set of Actor Agents must obey security restrictions given by the current step in the IIP and the current patient. Based on returned responses the RAs can offer to the PAs a best fit, or a sorted list of possible candidates for a human expert selection.

5. Further Improvements and Discussion

In the previous section we described the integration of the ProMA architecture into the K4Care system, which required only minimal conceptual changes in the model. The key advantage of this integration is in opening new possibilities for the K4Care system easily realized using ProMA. These, however, would require additional knowledge about the activities supported by the system.

The key factor is to use advanced simulation during the process of creating a new IIP. As we showed in the previous section, even a basic simulation can find some constraints that can be visualized and ease the process of creating a new IIP for a human expert.

Therefore, let us assume additional information available in the system. First of all, we can enhance the ontology for each action with additional properties, such as duration and type of the progress function (see [20] for more details on the progress function). Secondly, the map of a local area that the system is used in can be integrated into the multi-agent system as a virtual environment. Finally, we assume that the agent can access other relevant patients' data from the EHCR database during the process of creating the IIP (while obeying the security restrictions). All of these additional parameters allow us to simulate modelled procedures on a more detailed level. Based on a proposition of a new IIP a simulation can be ran and the results of the simulation can be used as an evaluation of the current version of the IIP and association of the actors.

The simulation of the IIP advance as in ProMA, which means that each of Process Agents is trying to successfully simulate associated step by selecting appropriate Role and Actor Agents.

As the Actors Agents are limited by other tasks already set in the system (i.e. there are several IIPs that are simulating at one time), they have to select the actions in which they currently participate according to their capabilities and priority. Let us demonstrate it on an example: we need to associate a nurse with an activity N.15 Blood sugar measurement at patient's home. We have a nurse Alice and a head nurse Eve. Nurse Alice has already planned a single task on the other side of the town, head nurse Eve has planned two shorter tasks in the centre. Note, that we use actors with different roles, however, there is hierarchy allowed within these roles and activity (i.e. a nurse is a more general role than the head nurse role). The ProMA architecture is able to handle this hierarchy and the RA correspondent to the nurse role would also send the proposition to RA correspondent to the head nurse role. In order to select an appropriate actor for this activity we use a multi-agent simulation in the virtual environment. That means the Actor Agents simulate execution of their already planned tasks within this environment. When the request for the N.15 activity is proposed by the RA associated with the nurse role, both Actor Agents are in different locations in the virtual environment and based on their other tasks they can estimate the time of completion of proposed task. Hence, in this simplified case, Eve would be offered for a human expert as an appropriate actor for this activity. However, if the proposed activity would have a higher priority for Alice, and her current task could be temporarily suspended, she could have been selected. In order to be the simulation of the steps as accurate as possible, PAs must have access to patients' health records to correctly estimate results of simulated activities (similarly to the critiquing mode of ProMA functioning describe in [19]) which can have impact of future course of execution of the IIP. Moreover, after the IIP is created and stored in the system, the ProMA can support critiquing and verifying the execution of these procedures and alert the health care specialist in charge in case of inconsistencies (see [19]).

6. Conclusions

In this paper we analysed the applicability of a generic multi-agent architecture into an existing decision support system, identified possible changes in both models and highlighted immediate advantages as well as possible future directions that are open by this integration. More specifically, we described processes as an appropriate notation for formalising medical procedural knowledge and described a process-based multi-agent architecture (ProMA). Then we focused on the existing support system, K4Care, where a procedural knowledge is stored in a SDA* formalism and used by a multi-agent system in order to provide support for home care specialists. We merged these two architectures using the minor model and conceptual changes and identified improvement in the process of creating a new Individual Intervention Plan for a patient. Finally, we identified possibilities for new features that the integration of ProMA into the K4Care system can further bring.

Acknowledgements

Acknowledgements belong to doc. Ing. L. Lhotska, CSc. for granting the access to K4Care documentation. The work was supported by the grant SVV-2010-265513, and by the Czech Ministry of Education, Youth and Sports under Research Programme no. MSM6840770038.

References

- [1] Balas E.A., Weingarten S., Garb C.T., Blumenthal D., Boren S.A., Brown G.D.: Improving preventive care by prompting physicians. *Arch Intern Med.* 2000 Feb 14;160(3):301308.
- [2] Garg A.X., Adhikari N.K., McDonald H. et al.: Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA.* 2005 Mar 9;293(10):12231238.
- [3] Grimshaw J.M., Russell I.T.: Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations. *Lancet.* 1993 Nov 27;342(8883):13171322.
- [4] Hunt D.L., Haynes R.B., Hanna S.E., Smith K.: Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. *JAMA.* 1998 Oct 21;280(15):13391346.
- [5] Kawamoto K., Houlihan C.A., Balas E.A., Lobach D.F.: Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *Bmj.* 2005 Apr 2;330(7494):765.
- [6] Burack R.C., Gimotty P.A.: Promoting screening mammography in inner-city settings: the sustained effectiveness of computerized reminders in a randomized controlled trial. *Med Care.* 1997;35:921-931.
- [7] Demakis J.G., Beauchamp C., Cull W.L. et al.: Improving residents' compliance with standards of ambulatory care: results from the VA Cooperative Study on Computerized Reminders. *JAMA.* 2000;284:1411-1416.
- [8] Overhage J.M., Tierney W.M., McDonald C.J.: Computer reminders to implement preventive care guidelines for hospitalized patients. *Arch Intern Med.* 1996;156:1551-1556.
- [9] Nilasena D.S., Lincoln M.J.: A computer-generated reminder system improves physician compliance with diabetes preventive care guidelines. *Proc Annu Symp Comput Appl Med Care.*
- [10] Egan M., Wells J., Byrne K., Jaglal S., Stolee P., Chesworth B. et al.: The process of decision-making in home-care case management: implications for the introduction of universal assessment and information technology. *Health & Social Care in the Community.* (2009, July); 17(4): 371-378.
- [11] Campana F., Moreno A., Riano D., Varga L.Z.: K4care: Knowledge-based homecare e-services for an ageing europe. *Agent Technology and e-Health.* 2008;p. 95115.
- [12] Aubrecht P., Lhotska L., Dolezel J., Dolezel J.: Mobile Devices for e-Services in Home Care. In: 4th European Conference of the International Federation for Medical and Biological Engineering. Springer; 2009. p. 10061009.
- [13] Isern D., Moreno A., Sanchez D., Hajnal A., Pedone G., Varga L.: Agent-based execution of personalised home care treatments. *Applied Intelligence.* 2009;p. 126.
- [14] Isern D., Moreno A.: Computer-based execution of clinical guidelines: A review. *International Journal of Medical Informatics.* 2008;77(12):787 808.
- [15] Isern D., Sanchez D., Moreno A.: HeCaSe2: A Multi-agent Ontology-Driven a Guideline Enactment Engine. In: CEEMAS '07: Proceedings of the 5th international Central and Eastern European conference on Multi-Agent Systems and Applications V. Berlin, Heidelberg: Springer-Verlag; 2007. p. 322324.
- [16] Novak P.: Jazyk: A programming language for hybrid agents with heterogeneous knowledge representations. *Programming Multi-Agent Systems.* 2009;p. 7287.

- [17] Bordini R.H., Hubner J.F., Wooldridge M.J.: Programming multi-agent systems in AgentSpeak using Jason. Wiley-Interscience; 2007.
- [18] Hindriks K.V., De Boer F.S., Van der Hoek W., Meyer J.J.C.: Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*. 1999;2(4):357401.
- [19] Bosansky B., Lhotska L.: Agent-based process-critiquing decision support system. In: 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009; 2009. p. 16.
- [20] Bosansky B., Brom C.: Agent-Based Simulation of Business Processes in a Virtual World. In: Corchado E, Abraham A, Pedrycz W, editors. *Hybrid Artificial Intelligence Systems*. vol. 5271 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2008. p. 8694.
- [21] Riano D.: The SDA* Model: A Set Theory Approach. CBMS 2007, Maribor, Slovenia, (2007).
- [22] Lenz R., Blaser R., Beyer M., Heger O., Biber C., Baumlein M. et al.: IT support for clinical pathways Lessons learned. *International Journal of Medical Informatics*. 2007;76(Supplement 3):S397 S402. Ubiquity: Technologies for Better Health in Aging Societies - MIE 2006.
- [23] Kumar A., Smith B., Pisanelli M., Gangemi A., Stefanelli M.: Clinical Guidelines as Plans: An Ontological Theory. *Methods of Information in Medicine*. 2006;2.
- [24] Fox J., Johns N., Rahmanzadeh A., Thomson R. PROforma: A method and language for specifying clinical guidelines and protocols. In: Amsterdam; 1996.
- [25] Shahar Y., Miksch S., Johnson P.: The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*. 1998;14:2951.
- [26] Peleg M., Boxwala A., Ogunyemi O.: GLIF3: The Evolution of a Guideline Representation Format. *Proc AMIA Annu Fall Symp*. 2000;p. 645649.

Contact

Mgr. Branislav Bošanský

Agent Technology Center

Dept. Of Cybernetics

Faculty of Electrical Engineering

Czech Technical University in Prague

Technická 2

166 27 Prague 6

Czech Republic

e-mail: bosansky@labe.felk.cvut.cz