Proceedings of the 7th International Conference on Applied Informatics
Eger, Hungary, January 28–31, 2007. Vol. 2. pp. 409–416.

Supporting MAS complex service modelling through agents substitutability*

Gianfranco Pedone, László Zsolt Varga

System Development Department,
MTA SZTAKI, Budapest
e-mail: {gianfranco.pedone,laszlo.varga}@sztaki.hu

Abstract

In knowledge intensive domains the underlying information system is asked to provide a behaviour that match with highly-changeable plan guidelines. Such is the case of health-care contexts, in which patients treatment can be highly customized to subject's requirements. In these e-Health scenarios Multi-Agent Systems are showing a rapid increase, thanks to their ability to manage complex tasks and adapt gracefully to unexpected events. Nevertheless, the necessity of modelling run-time personalized and complex services can lead the user to conceptualize inconsistent or unavailable services. This is mainly due to the lack of a process for agents substitutability evaluation.

This paper proposes a general approach in assisting the modelling of complex agent and ontology-based services for e-Health, by advancing an Agent Replacement Methodology (ARM). The most important aspects of the latter can be summarized in a principle of behavioural equivalence and in the minimalization of cost components (such as communications, read-write operations, or expected inputs) inherent to this.

1. Introduction

The work presented in this paper is part of the K4CARE project, whose aim is to design and develop a prototype system, based on Web technology and intelligent agents, that provide services defined in a Home Care (HC) model. The project is developing a platform to manage the information needed to guarantee an ICT-based Home Care service and, particularly, it is requiring the generation of Intervention Plans (IPs) from the personalized health care treatments and the configuration of capability-based decision support methodology for the arrangement of agent-based complex medical services. This last aspect is the focus of the present paper.

*This paper was prepared in the context of the K4CARE project, funded under the 6th Framework Programme of the European Community (Contract N. 026968).

The health care of a patient is particularly complex not only because of the great amount of resources required to guarantee a quality long-term assistance and the amount of interactions in a medical treatment, but even because physicians are often asked to define at highly personalized health care services by orchestrating capabilities of a potentially large number of domain actors. These capabilities are represented by agents skills and embedded in their behavioural model. Some authors in literature are targeting the different objective to verify and validate Multi-Agent Systems behaviour, as reported in [1, 2]. But how can a physician trust agents identities involved in an IP definition? How to replace unavailable agents while assuring the system an equivalent behaviour? In this process an important role can be played by the process of agents replacement through the evaluation of their capabilities equivalence. The paper proposes a double possibility in achieving this objective: the first, based on the ontological equivalence of services and agent capabilities (described in K4Care project ontologies), and the second by tracing simulation guidelines in the replacement of unavailable agents and by measuring the system reaction in terms of expected behaviour. Our novel proposal takes into account behavioural characterizations typical of Contract-driven Systems while trying to extend and adapt them to the agent paradigm: this is summarized and presented in the Agent Replacement Methodology (ARM). The document's content is structured as follows: Section 2 briefly highlights K4Care project and the complexity in arranging new agent-based services (medical treatments); Section 3 reports the theoretical background necessary to the ARM and briefly presents the ARM methodology guidelines. Section 4 is dedicated to conclusions and future works. Acknowledgements close the present paper.

2. e-Health complex services in K4Care

The K4CARE model is explained by the paradigm shown on the Figure 1. In

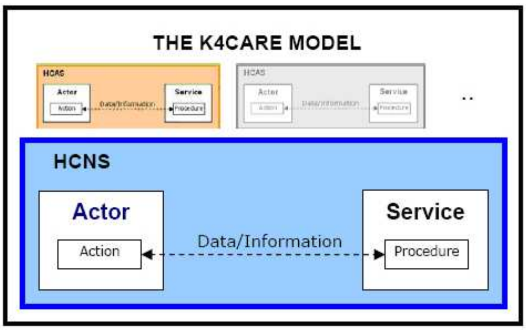

Figure 1: The K4CARE Model Architecture for HC

the model services are distributed by local health units and integrated with the social services of municipalities. The model is represented by a modular structure that can be adapted to different local opportunities and needs. K4CARE model is based on a home care nuclear structure (HCNS) which comprises the minimum number of common elements needed to provide a basic HC service. Each one of the HC structures consists of the same components: ***Actors*** are all the sort of human figures included in the structure of HC; ***Professional Actions*** and ***Liabilities*** are the actions each actor performs to provide a service within the HC structure; ***Services*** are all the utilities provided by the HC structure for the care of the HCP; ***Procedures*** are the chain of events that leads an actor in performing actions to provide services; ***Information*** are the documents required and produced by the actors to provide services in the HC structure. There are two kinds of knowledge embedded in system services: *declarative* and *procedural*. The former contains the information on the basic elements of the K4Care model and the organizational relationships between the system actors. The latter is concerned with the representation of the sequences of actions involved in the provision of a service.

Declarative knowledge. *Ontologies*, as a set of concepts, properties and relations, constitute a feasible paradigm to represent the declarative knowledge used in system services [4]. In K4Care an ontology named ***Actor Profile Ontology*** (APO), details the all basic elements of the K4Care HC model (actors, actions, services, procedures, documents) and the relationships between them.

Procedural knowledge. Procedural knowledge codifies complex medical tasks and it is required to define the set of available actions performed by all actors in the platform. This knowledge is coded using a flowchart-based representation called SDA* [3]. The basic elements of SDA* structures are "*states*", "*decisions*" and "*actions*". States describe patient condition situations. Decisions code alternative options required to guide the enactment of a plan. An action is one of the activities that an actor can perform in the treatment of a patient. Between those elements, directed edges define the direction of the steps and can be labelled with temporal constraints. The SDA* formalism is used in K4Care to represent ***3*** kinds of elements:

Procedures = descriptions of the steps to be taken within the K4Care platform to provide one of the HC services;

Formal Intervention Plans (FIPs) = general descriptions, defined by health care organizations, of the way in which a particular pathology should be treated;

Individual Intervention Plans (IIPs) = descriptions of the specific treatment that has to be provided to a particular patient.

In order to simplify the model of a service provision, agent activities have been represented by 4-tupled tasks, as synthesized by the following expression:

$$Task = (subject, object, action, doc). \tag{2.1}$$

Task (an agent action) represents the atomic logical step that must be taken by an agent in the provision of a service, in conformity to a procedure. Procedures are nation-dependent, so that a specific service realization in a country can differ from the realization of the same in another one.

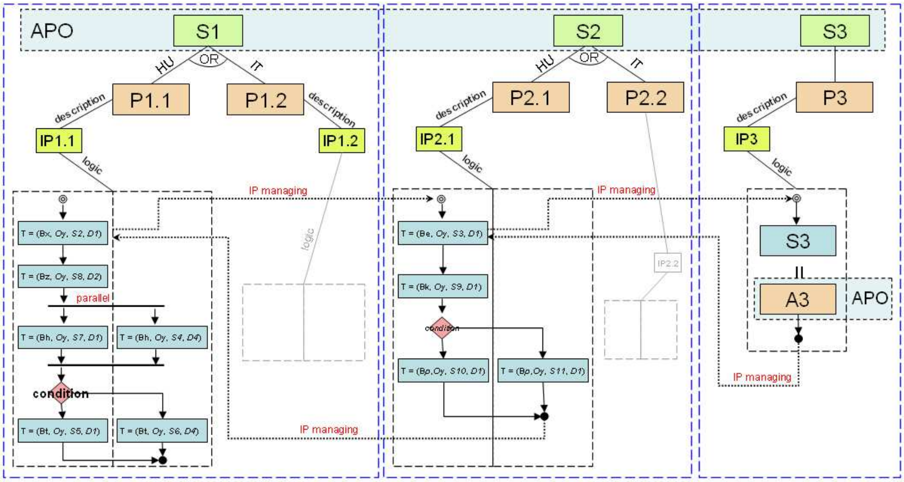

Figure 2: K4CARE service execution model

Tasks corresponds to a message sent to the "*subject*" agent to execute the named "*action*"; "*object*" is the patient, "*doc*" is the electronic document relating to the service that must be accessed and updated during the execution of the action. The logic for the advancement of service provision is owned by the SDA Engine, which is a special agent with the fundamental capability to lead and control the interpretation at run-time of the service flow-chart (for example an Intervention Plan execution, as in Figure 2), by transforming it into a suitable graph, first, and then coordinating agents actions in the service. Figure 2 evidences a service invocation diagram in which a nested control reference occurs: a service (*S1*) requires the execution of another service (*S2*) (contained into it), which refers to a final invocation for a third service *S3*). The expression "*IP managing*" in the figure stands for the SDA* execution capability by the Engine agent. *Px.y* represents the procedure describing the x service in the y version for the proper country. In conclusion, *IPx.y* is the specific instantiation of the Intervention Plan, whose execution logic is decompounded into atomic tasks, as previously reported.

3. Agent capabilities equivalence in K4Care

K4Care is a large-size, knowledge-intensive, agent-based system in which agent interactions dynamics must be formally governed. This reason led us to consider the introduction of a usable and flexible, but at the same time trustable and rigorous, methodology in evaluating agents capabilities substitutability: an *ontological* *approach*, based on the semantic equivalence of agent-oriented services definition in K4Care, and a *behavioural evaluation approach*, through capabilities simulation. The latter will require the definition of explicit contracts between users and agents, as derived from the theory on Contract-Driven Systems [10] and theoretically presented in the following of the paper. In the health care, despite the availability of general medical guidelines, treatments are almost always personalized to the particular patient case (which is, in practice, "unique" in its characteristics). The care practitioner has to orchestrate domain actors (agents) capabilities in order to model new care service. As we have already seen, in K4Care we have trace of these in the APO. The physician has to determine a new work-flow of actions to be submitted to the SDA Engine, which will coordinate the service provision. The capabilities of actors are differentiated and, considering the level of responsibility and the sensitive nature of treatments, the "designer" wants to have the absolute assurance of the validity of these capabilities. This problem is even amplified in the case of unavailability of requested actors: physicians will have to investigate available agents, in order to collect "*equivalent*" behaviours, trying to minimize the "cost" of a possible mismatch (communications, read-write operations, expected inputs) with those capabilities initially required. This equivalence over agents capabilities can be "*trusted*" (reported in the following as ontological equivalence) or deeply *investigated* (by simulating agent behaviours).

3.1. Ontological equivalence

Ontologies are a standard *AI* knowledge representation mechanism. Their main components are a set of concepts (or classes) C, which are taxonomically related by the transitive *is-a* relation and non-taxonomically related by named object relations *R*C*C*String*. They represent the knowledge assets of the K4Care application and the catalyst for the agents behavioural model, as well as the fundament for the agents code generation. A necessary and sufficient class definition in an ontology (Description Logic-based system), which consists of restrictions on a set of properties, implies that an individual which satisfies the property restrictions, belongs to the class. This is the philosophy of the first proposed approach in evaluating agent capabilities equivalence. The new service designer could need, for example, to invoke capabilities from a *family doctor*, which results unavailable. On the basis of the analysis of object properties, another actor can be addressed (*physician in charge*), which evidences the same capabilities of interest (from the APO: "*does action BO.01*" and "*initiates service for brochure consultation*"). This leads to an ontologically equivalent actor replacement.

3.2. Behavioural equivalence

Our intention is to systematically validate agents behavioural capabilities. This requires the introduction of an agent-oriented engineering method for explicitly defining the expected effects of agents activities. In other words, the idea is to consider agents behaviours as conceptualized agreements, on the basis of which they assure the reliability and validity of their actions. We are trying to extract and adapt behaviours definition guidelines proper of Object-Oriented systems to a Multi-Agent System (like K4Care). Undoubtedly there are several adaptation points that still have to be clarified. We try to report on some in the following. One of the most important notions to ensure correctness of software components is the concept of abstract data types (ADTs), introduced in the 1970s by Liskovand [5], and relying on foundation work done by Hoare [6] and Parnas [7, 8]. Correctness formulas of the form $\boldsymbol{P\{Q\}R}$ (also called Hoare triples) are a mathematical notation and form the basis for assertions. On the basis of this notation, we can introduce concepts such as "preconditions", "post-conditions", and "invariants", to express the correctness properties of methods and classes. The idea of assertions led to the concept of a contract, which binds a method to its clients and thereby imposes obligations and grants rights for both of them. This concept is also called Design by Contract (DBC) and it was introduced by Meyer [9]. When trying to adapt contract assertions to the agent paradigm, in any case, different aspects must be taken into account:

1) agents are an extension of the OO paradigm, in the sense they manifest autonomy, proactivity, social and learning capabilities;

2) the correct approach in designing agents does not involve methods-based interactions (whether it is always possible) but the definition of a behavioral model that is triggered by a mechanism of message passing;

3) agents can reason on their own over the environment configuration, even through ontologies;

4) on the basis of the previous consideration, agents cannot be obliged to follow a contract in the strict sense introduced above, while they represent a kind of "intelligent" entity in the system, rather than a "simply reactive" object.

These lead to the following considerations:

1) DBC theory can still be adapted to a Multi-Agent System design, but preconditions, post-conditions and invariants concepts must be referred to agents behavioral model. That is, they have to reason over preconditions in order to grant the expected contract-based assertions (invariants and post-conditions);

2) an agent does not own environmental parameters within its class, but rather these should be considered as an external resource accessed by the agent;

3) each agent can be seen as an "intelligent" black-box, whose behaviour can even remain unclear to the user perception (Q terms in Hoare triple), but whose effects are constrained and validated by the definition of a contract (P and R terms in Hoare triple).

On the basis of all previous considerations, we can now affirm that validating an agent's behaviour implies the verification of the contract conditions (agreement) between the two involved parties (user and the agent, or even two agents). At the moment we are still investigating on the proper formalism to adopt in the definition of the agent-oriented contracts, evaluating DBC guidelines contained into [10].

3.3. Agent replacement methodology

In this section we outline the general guidelines to be followed by the responsible physician in the provision of a new agent-based medical service in K4Care. The ARM process is, naturally, characterized and triggered by the unavailability of one or more agents embedding the requested capabilities (in terms of actions).

The methodology should consists of the followings:

1. Realize the unavailability of a requested agent's activity;

2. Analyze the main objectives of the new medical treatment service;

3. Try to capture their hierarchical importance by a parameterized rating (time, physical location of agents, overall capabilities, other);

4. Decide which approach to use (ontological, behavioural, both);

5. Apply parameterizations to agents availability results;

6. Analyze results. The user (physician) is asked for a responsibility decision about the patient treatment;

8. Step on with the composition of the treatment.

This methodology is to be intended as a medical decision support tool, to be coupled to the knowledge and experience of physicians (who remain the responsible for the intervention plan), not the intention to substitute to it.

4. Conclusions and future works

We consider the evaluation process of agents substitutability as a necessary support for agent-based complex service modelling, specially in the context of the electronic medical assistance and interventions, as in K4Care project. We have presented a methodology for supporting physicians in run-time modelling of agent-based intervention plans. Due to the sensitive nature of the application domain an the critical event of unavailability of those agents involved in patients treatment, physicians may need to evaluate the equivalence of different agents before replacing some of them. Two approaches have been proposed to evaluate agents role's equivalence: the first, ontological (based on the domain conceptualizations from the APO), is being designed in practice within the K4Care), and the second, behavioural (evaluating the extendibility and applicability of DBC concepts to MASs). Theoretical affinity between behavioural aspects of objects as reported in DBC and behavioural model of agent paradigm inspires our investigations for the complete adaptability of this approach to our intent. We are at the moment analyzing and evaluating the exploitation of existing tools for the definition of agent-oriented contracts. Our future works involve, in addition to the previous, the inspection of MAS development methodologies which could model and embed contract-based concepts into agent-based paradigm.

Acknowledgements. The authors would like to acknowledge the work of all the K4Care partners, especially Fabio Campana, Roberta Annicchiarico and the medical partners (K4Care model), Sara Ercolani, Aïda Valls, Karina Gibert, Joan

Casals, Albert Solé, José Miguel Millán, Montserrat Batet and Francis Real (ontologies, data abstraction layer and service execution).

References

[1] Fallah-Seghrouchni, A. E., Degirmenciyan-Cartault, I., Marc, F., Modelling, control and validation of multi-agent plans in dynamic context, *Autonomous Agents and Multiagent Systems, AAMAS 2004. Proceedings of the Third International Joint Conference on Volume*, Issue, (2004), 44–51.

[2] Jonker, C. M., Treur, J., Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness, *International Journal of Cooperative Information Systems*, (2002).

[3] Riaño, D., The SDA* Model v1.0: A Set of Theory Approach, *Technical Report (DEIM-RT-07-001)*, University Rovira i Virgili. `http://deim.urv.es/recerca/reports/DEIM-RT-07-001.html` (accessed 03-04-2007).

[4] Fensel, D., Ontologies: A silver bullet for knowledge management and electronic commerce, Germany: Heidelberg, (2001).

[5] Liskov, B., Zilles, S., Programming with abstract data types, *Computation Structures Group, Memo No. 99, MIT Project MAC*, Cambridge, MA (1974), [see also SIGPLAN Notices 9(4), 50–59 (April 1974)].

[6] Hoare, C. A. R., Proof of correctness of data representation, *Acta Informatica 1*, (1972), 271–281.

[7] Parnas, D. L., A technique for software module specification with examples, *Communications of the ACM*, 15(5), (May 1972), 330–336.

[8] Parnas, D. L., On the criteria to be used in decomposing systems in to modules, *Communications of the ACM*, 15(12), (December 1972), 1053–1058.

[9] Meyer, B., Object-Oriented Software Construction, *2nd edn (Prentice Hall, Upper Saddle River*, 15(12), NJ (1997).

[10] Plosch, P., Contracts, Scenarios and Prototypes – An Integrated Approach to High Quality Software, ISBN 3-540-43486-0. *Springer-Verlag* Berlin Heidelberg (2004).

Gianfranco Pedone, László Zsolt Varga
Lágymányosi u. 11.
H-1111 Budapest
Hungary