Towards an Environment under which Executing LAPs

J.B. Mocholí, D. Domínguez. C. Fernández
Tecnologías para la Salud y el Bienestar (TSB), Instituto ITACA,
Universidad Politécnica de Valencia (UPV)
{juamocag, dadotor, cfllatas}@itaca.upv.es

Abstract

In this paper we describe the preliminary work developed in the use of a multi-agent based environment for the execution of guidelines protocols in the field of healthcare. We have used the LAP (Life Assistance Protocol) concept in order to extend the classical medical guidelines. A LAP is the concrete and practical set of guidelines, recommendations and prescriptions (actions) for a specific need, such as pathologies (diabetes, heart failure…), concerns (stopping smoking, following a healthy lifestyle…) or special needs (pregnancy, elderly…). Using a LAP as input, an executor will be developed in order to improve the management of this type of processes and facilitate these tasks to the healthcare professional.

1. Introduction

The agent based paradigm has been used in a large different sort of domains by using their characteristics: autonomy, re-activity, pro-activity, and social ability. As an agent is a social object, it can interact with other agents in order to perform a task, and is then when we are talking about a multi-agent system (MAS). In these systems there are some different types of agents which offer their services and consume the services provided by others. Through time, different approaches have appeared to implement the agent based paradigm, also standards and platforms have appeared in which execute this systems.

The healthcare domain has been one of the frameworks where computer science research have developed and applied new technologies in order to find solutions to historical problems. Some examples of such technologies could be, for instance, ontologies used to represent medical language and concepts, reasoners and knowledge systems in order to support the doctors' decisions, standards to share medical data, etcetera… There are also languages to specify medical guidelines and environments which execute them, and also approaches to a multi-agent based guideline executor.

The execution of medical guidelines and LAPs is a distributed task, involving complex processes, some time with long term results. The aim of these medical guides is to provide to the healthcare professional an easy but robust way to perform and follow up a specific patient medical treatment. This is a scenario in which the use of a multi-agent system has confirmed its efficiency.

The architecture developed to implement this environment, is not only focused on the workflow management, also covers information processing, prediction and diagnostics to offer them to the healthcare professional as data to support their decisions in order to assert or correct the medical treatment of a specific patient. The architecture uses an agent as the figure of coordinator to check and evaluate each particular LAP; it uses also specialized agents in order to perform the tasks described in the LAP. Finally there is also a GUI agent that shows and interacts directly with the patient and the healthcare professional.

1.1. Basics

The classical approaches that we can find in the literature related to the specification, representation and execution of medical guidelines are based on different models. They could be sorted as logic-based, rule-based, Network-based, and Workflows as Petri Nets [10]. The aim of these models is creating computer-interpretable guidelines that facilitate decision support, and those models cover both computable data and clinical knowledge. There are also other models focused on the clinical knowledge representation by using XML guideline document models which in some cases can be also computable by it.

Due to the nature of LAPs and other medical guidelines, it is required a workflow mechanism which allows parallel execution, and also allows the fulfilment and following of the restrictions that are used.

Until now the approaches to modelling computable languages for medical guidelines were based on monolithic languages executed under a "unique" engine. However, by using a language definition based on workflows, and working in adapting the medical guidelines to it, it can be used to write medical guidelines which can be decomposed in smaller tasks which can be easily executed under the figure of a coordinator. This model is basically focused on the workflow's flow of the medical acts; the actions to be performed are not taken into account at the beginning, only the workflow's nodes are important. Actions are used later to take the jumping conditions and restrictions from a node to other node of the workflow using their responses or results as indicators to be applied in the transitions between nodes. To do that, this workflow based model needs:

- A theoretical base under which executing workflows.
- A LAP's specification language workflow oriented.
- A CASE and verification tool based on graphical symbols to define the workflow.
- A compiler used in addition to the CASE tool to generate workflow LAP templates.
- A tool used to merge both the LAP template and the patient's clinical record in order to make a computable instance of the LAP.

In addition to the workflow model specification, the execution model appears. It must support multiple LAPs and their tasks concurrently executed, but support also autonomy, flexibility and interaction with the users. The election of the MAS was based on that. In this system, there are agents under execution that represents a workflow node from the LAP and computes its list of actions.

The final LAP instance will have such amount of (qualitative and quantitative) information that will be easily computable. The chosen language used to write the LAP templates and the instances is XML; it can be easily used to describe the workflow, the nodes and its transitions, and also to reflect the tasks and actions per node, and the stages that can be extracted from the workflow. This XML is filled using a set of ontologies in order to provide support to write the correct target tasks, the goals and the parameters of the actions. The actions that are used during the LAP's specification are a reflection taken from the actions that the system can perform.

In the following sections we are introducing the steps and topics that we have used in the definition of what the LAP is (its conceptual model), the workflow model specification, and the execution model.

2. Life Assistance Protocols (LAP)

One of the common issues that has come into vogue in the recent years, in the domain of the e-Health technologies for citizen centred health systems, is the way to integrate disease prevention, control and treatment, with the person daily life in a personalized and non-invasive way. This is one of the seeds that has led to the creation of the a new concept: the Personal Health or p-Health.

The *Life Assistance Protocol* (LAP) is defined within this framework, and it is going to be the starting point of the architecture described in this paper. A LAP is a set of guidelines, recommendations and prescriptions (actions) for a concrete need of the user, including not only health care actions but lifestyle.

The user's needs can be divided into three different categories: pathologies; such as diabetes, heart failure, overweight…; special conditions; such as pregnancy, elderly… and main concerns; such as stopping smoking, following a healthy lifestyle, etcetera... The LAP defines, then, a set of actions in order to achieve the goal that should solve the need of the user. These actions represent the cycle (workflow) of the life of a person, in which each stage takes into consideration all the person dimensions: motivation, clinical status, personal context, etc… These dimensions are continuously re-evaluated to trigger moves forward and backwards in the stages of the LAP.

After this brief introduction, the following definitions set the boundaries of the conceptual model of the LAP, which is placed in a formal framework based on workflow execution models.

Definition 1. A **LAP** is a set of guidelines, recommendations and prescriptions (actions) for a concrete need of the user. It includes a general *objective* and set of *stages*.

Definition 2. A **STAGE** is a set of active nodes in a concrete moment of time. It includes a description and a set of *nodes*. There are two special stages to be defined here, the *initial stage* of the LAP and the *set of objective stages*, which will be reached because whether the general objective of the LAP has been achieved or the patient is in a stage in which the LAP is not able to continue (i.e. the person needs to go urgently to a hospital).

Definition 3. A **NODE** is a set of atomic actions, which are independent between them, transactionally simultaneous, that will be executed over or by one or several determined actors. Each node will be defined by a description, a set of *actions,* a subset of the user relevant data (Clinical Record), indicators and a *reaction*.

Definition 4. An **INDICATOR** is any actor's feature that is atomic, dynamic, measurable and discrete and is relevant for the LAP. It is defined by a range of values, a

description and a formula to calculate the indicator value. Time can also be an indicator, since the occurrence of a certain event, after a period of time, can trigger the transition from one node to another.

Definition 5. An **ACTION** is represented in the LAP as any atomic interaction between the actors and the system (or vice versa). It includes an origin (actor), an addressee (actor), and a description. However, the LAP actions have some semantics in their definition. This means

Definition 6. An **ACTOR** is considered to be any user of the system.

Definition 7. A **REACTION** is any alteration of the system's status.

All these concepts will be used in order to create templates, which will be filled by the domain experts and instantiated by the system for a concrete person. This instance needs to be internally represented in a formal language that will be executed by the architecture described in the following sections. In the Figure 1 these layers are represented.

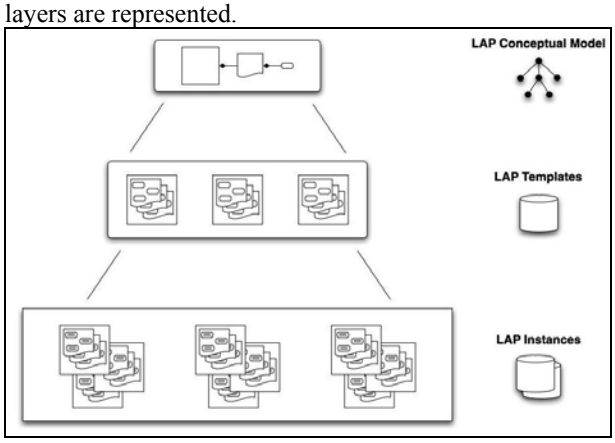

Figure 1. LAP layers from abstract to concrete

3. LAP formal framework

After the introduction of the conceptual model of the LAP, this section introduces the formal representation of the model.

Definition 8. A Life Assistance Protocol is a tuple LAP = {O, E, N, δ, γ, Σ, Δ, σ, E_o, P}, where:

$\sigma = \{a_1, a_2, a_3 \ldots a_n\}$, finite set of actors;

$\Sigma = \{i_1, i_2, i_3 \ldots i_n\}$, finite set of indicators;

$P = \{p_1, p_2, p_3 \ldots, p_n\}$, finite set of processes;

$\Delta = P \times \sigma^2$, a_1 = origin, a_2 = addressee, set of actions with one actor as origin and one as addressee;

$N = \{n_1, n_2, n_3 \ldots, n_n\}, \forall\ n_i \subseteq \Delta^*$, finite set of nodes;

$E = \{e_1, e_2, e_3 \ldots, e_n\}, \forall\ e_i \subseteq N^+$, finite set of stages;

$\delta: E \times \Sigma^* \rightarrow E$, set of transitions between stages;

$\gamma: N \times \Sigma^* \rightarrow N$, set of transitions between nodes;

$E_0 \in E$, initial stage;

$O \subseteq E$, set of goal stages.

All the definitions can be easily identified in the formalization. Only the reaction has been divided into two total functions to represent all the transitions in the workflow.

4. Workflow model

The theoretical model used in the computable specification of our LAP workflow is taken from the preliminary studies of *Timed Parallel finite Automata* (TPA)[1] and this is also based on the *Parallel finite Automata* (PFA). PFA is an improvement of the *Determinist Finite Automata* (DFA) and presents an automaton with capability to execute parallel activities. A parallel automaton implies that more than one node are active at the same time -like in Petri Nets- but they also are supported by the concept of stages, that approximates the PFA to the DFA by joining the parallel active nodes into a different stage, and finally this based on stages model is an abstraction that maps the PFA to a DFA which is a well controlled framework. TPA is a formal framework that appends timers to the PFA by adding the concept of *Clocks* to the PFA's definition and also using the *Clocks'* timers as indicators.

In the Figure 2 is shown a TPA's workflow schema which represents a (care plan) medical actuation for a disease treatment.

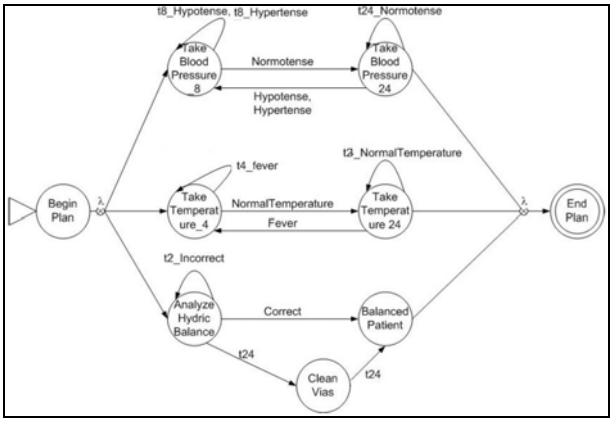

Figure 2. A TPA Example extracted from [1]

In [1] is also explained how the TPA framework covers a series of classical data workflow patterns, like the lineal sequence pattern, the parallel split pattern, the synchronization pattern, the multi-choice pattern, the multi-merge pattern, the discriminate ending pattern, the milestone pattern, the Interleaved Parallel Routing pattern etc.

The TPA formal framework is defined by using these concepts:

Definition 9. A Timed Parallel Automaton is a tuple $A = \{C, P, N, Q, T, \phi, \Sigma, \gamma, \sigma, q_0, F\}$ where:

C is a finite set of Clocks,

P is a finite set of Processes,

$N \subseteq P \times C^+$ is a finite set of Nodes,

$Q \subseteq N^+$ is a finite set of states,

T is a finite set of time labels that can be generated by the Clock set C,

ϕ is finite set of symbols,

$\Sigma \subseteq T \cup \phi^+ \cup T \times \phi^+ \cup \{\lambda\}$ is the finite input alphabet,

$\gamma : N^+ \times \Sigma \rightarrow N+$ is the Node Transition function,

$\sigma : Q \times \Sigma \rightarrow Q$ is the state transition function,

$q0 \in Q$ is the initial state,

$F \subseteq N$ is the set of final states.

Finally we can say that LAPs are TPAs because the LAP formalization is based on stages like TPAs, and also uses temporized indicators and other parameters that are used in the formal specification of TPAs. It is clear that there are also differences between LAPs and TPAs but it can be summarized saying that:

- The finite set of indicators used in the LAP's definition are equivalent to the finite input alphabet used in the TPA's definition, it est., $\Sigma^*_{LAP} \equiv \Sigma_{TPA}$. In the LAP's definition we assume that the *Clock* used in the TPA is a process, and its temporized labels are directly used as indicators.
- The set of actions of a LAP is equivalent to the TPA's set of processes, $\Delta \equiv P$, but adding to the TPA's processes an addressee and an origin.

The differences between LAPs and TPAs are only in components that are not implied in the regular formulization of the LAPs, so we can still saying that LAPs have a regular definition. Finally, in most of the cases, LAP stages are composed by a single node, due to the nature of the problems treated and range of guidelines provided. However, the model supports more complex health care protocols definitions.

5. Execution model

As we said in previous sections, the environment under which the LAPs will be executed is a MAS in which are used different types of agents that perform the tasks in the system. The MAS is also used to divide and distribute the tasks and processes of each running LAP's instance correctly by using the autonomy and knowledge of the present agents. The situations and possible LAPs that we took in mind don't require a time strict execution or response, and neither are critical actuations or treatments, so the time response of the MAS will not affect to the system. We said also that the final XML instance of the LAP has a workflow inside that can be easily extracted in order to compose the different nodes present in the workflow, and as a less complex process, it can be easily computed. The task to "extract" the workflow from the LAP is done by the figure of the LAP's coordinator. In addition, the coordinator is who follows the execution flow, annotates the active nodes and also the active stage, and finally is who evaluates the indicators in order to perform the transition between stages. Stages are important because they provide us a determinist execution. The Figure 3 shows an overview of this execution model.

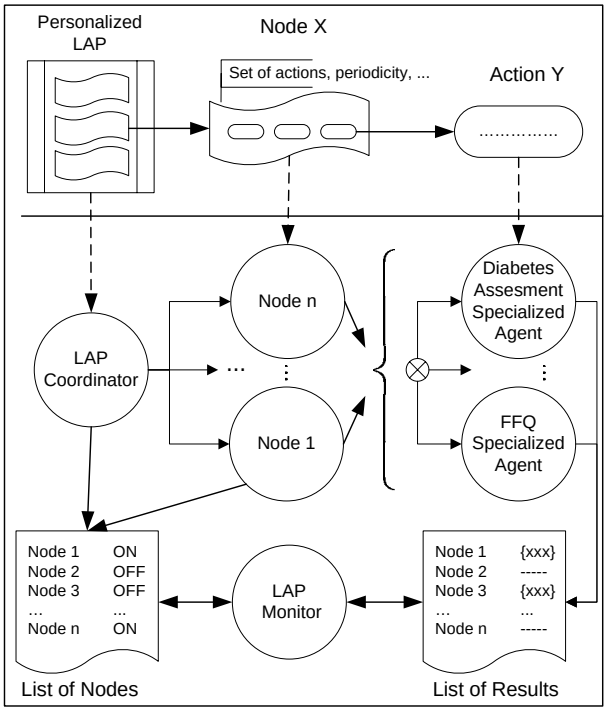

Figure 3. LAP schema overview

When the coordinator has the LAP decomposed in nodes, it takes the mission of creating and starting an agent by node. These agents have all the information related to a node: the tasks, the set of actions, and its parameters and results to be passed to the coordinator to be used as indicators for the conditions of each transition between stages. The node agent inspects the actions to perform, and like actions were fully specified since the beginning of the template by using ontologies, the agent knows that exist almost one specialized agent which performs those actions. Then the node agent requests the action to the specialized agent. The specialized agents provide solutions, perform actions, make suggestions and also check patient's signals and health care parameters. The actions and treatments could take time while returning results or taking time to be finished, so both the agent coordinator and the nodes hold a record with information of each active stage and nodes per time

instant, the actions performed, the results, etc. so they can easily recover their status if is need to "kill" or put in "stand by", and also can be easy to redraw the flow and navigate through it to see all the previous steps and stages that the patient has crossed.

Other Figures that appear in the system are the Main Coordinator -also called "Coordinator of coordinators"-, the agent monitoring the flow and processes per LAP, a set of agents that interact with the actors used also as a GUI, and a set of agents specialized on sending warnings and alarms. The Main coordinator has the tasks of being the entrance point of the LAP; it creates a LAP's coordinator per LAP and also is the starting point of recovering from a stand by process.

We chose the JADE platform as the environment under which create and execute our MAS system. This well known platform provides a set of utilities and services in order to communicate the agents between them. Here a text extracted from the JADE's official page: *"JADE simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases"*.

6. Conclusions

This paper has given a view of a possible solution for executing personalized life assistance sets of guidelines, named here LAPs, with the purpose of solving people needs, such as pathologies, life style concerns, etc…

In the paper, we propose the use of a multi agent system, based on the formalization of the LAP model by using the concept of the timed parallel automaton, as an added value solution over the already existing workflow engines in the market.

All the definitions in the conceptual model, as well as the formalization obtained from this model, need to be revised and validated, in order to consolidate the terms this paper describes, and verify the feasibility of the execution model.

7. Acknowledge

We would like to thank the European Commission for funding the PIPS Project (Contract 507019), in which all this work is being carried out, as well as to the other partners who helped in its development.

8. Bibliography

[1] C. Fernández, J.M. Benedí, *"Workfow mining basado en Indicadores: una aproximación teórica"*.

[2] L. Lhotska, O. Stepankova, *"Multi-Agent Systems as an Integration Environment for Classical and AI Components"*.Fourth International ICSC Symposium on ENGINEERING OF INTELLIGENT SYSTEMS (EIS 2004).

[3] J.L. Nealon, A. Moreno, *"Agent-Based Health Care Systems"*, EU-LAT workshop on e-Health 03.

[4] D. Isern, D. Sánchez, A. Moreno, A. Valls, *"Personalisation of Medical Services"*, EU-LAT workshop on e-Health 03.

[5] D. Isern, *"Agent-Based Management of Clinical Guidelines"*, Thesis Project at the Technical University of Catalonia.

[6] D. Sutton, J. Fox, *"Syntax and Semantics of PROforma"*, PROforma Technical paper, Version: 1.3.38, March 2003.

[7] M. Peleg, et al. "*Comparing computer-interpretable guideline models: a case-study approach"*. J Am Med Inform Assoc. 2003 Jan–Feb; 10(1): 52–68.

[8] M Peleg, et al.,*"Guideline Interchange Format 3.5 Technical Specification"*, Intermed Collaboratory, May 4 2004.

[9] P.A. de Clercq, J.A. Blom, et al. "*Approaches for creating computer-interpretable guidelines that facilitate decision support*". Artif Intell Med. 2004 May;31(1):1-27

[10] Web Page: http://openclinical.org/gmmintro.html, last access February 2006.

[11] *"Asbru: a global ontology for guideline-application tasks"* Web Page: http://smi-web.stanford.edu/projects/asgaard/AsbruL.html, last access February 2006.

[12] Java Agent Development Framework (JADE),*"JADE Programmer's Guide v3.3"*, last access March 2006. More information available on http://jade.tilab.com/papers-txt.htm

[13] JADE API v3.3, last access March 2006. More information available on http://jade.tilab.com/doc/api/index.html