

Adaptive Medical Workflow Management for a Context-Dependent Home Healthcare Assistance Service

L. Ardissono, A. Di Leva, G. Petrone, M. Segnan, M. Sonnessa¹

*Dipartimento di Informatica
Università di Torino
Torino, Italy*

Abstract

The provision of health-care services and home assistance to the elderly and chronic patients is a challenging application scenario for Web Services composition, which support the integration of distributed, heterogeneous services in complex workflows. However, in order to support the management of the typical routine activities carried out when dealing with a patient, several contextual conditions, such as his health conditions and the presence of specialized personnel at his home, must be taken into account. Thus, context-awareness must be added to the developed services.

In this paper, we present the architecture of a framework integrating workflow management and context-aware action execution to support the personalized management of medical guidelines in home health-care assistance services. The framework is based on the integration of Web Service and Autonomous Agents techniques, which enhance the execution of medical guidelines, handled as abstract workflows, with a context-sensitive execution of actions.

Key words: Web Services, context-aware workflow execution, autonomous agents.

1 Introduction

Decentralized health-care services and home assistance are key tools to achieve two objectives: enabling patients to spend their time in a familiar environment and reducing the hospital expenses. In order to efficiently manage home health-care assistance, a virtual connection between patient and hospital has to be established, which supports the monitoring activities, the request of

¹ Email: liliana,dileva,giovanna,marino,sonnessa@di.unito.it

services and the management of the medical protocols that the patient undergoes.

The development of this type of application is far from trivial because the service should be tailored to different actors (patient and/or relatives,² nurses, doctors) and it should integrate distributed, heterogeneous sub-services to mediate the interaction between user and service suppliers. Moreover, the service has to manage medical guidelines in a context-aware way in order provide the user with instructions that are appropriate to the patient's situation.

We believe that the recent results achieved in Service Oriented Computing [11] and Context Aware Computing [5] can be integrated with traditional approaches developed in the Autonomous Agents research [9] to make home health-care assistance a reality. Autonomous Agents research has in fact developed advanced action execution techniques that support a sophisticated selection of the actions to perform to achieve goals, taking contextual conditions into account. The idea is to combine an abstract workflow management with a fine-grained specification of the actions to be performed, within each activity of the workflow to be carried out. The selection of actions can thus be done by analyzing the relevant contextual conditions.

In this paper, we propose an architecture for the development of health-care assistance services managing medical guidelines at the patient's home in a personalized and context-dependent way. The architecture is aimed at supporting the interaction with the hospital personnel and services by means of desktop and mobile client devices, as well as the automated storage of data about the patient's health state in the clinical record. Our proposal relies on Web Service composition standards [1] and Autonomous Agents techniques [9] to integrate distributed applications in a personalized service.

It should be noticed that the elderly and chronic patients often undergo different medical protocols in parallel. The service must therefore work as a set of assistants which manage the guidelines to be carried out and interleave the execution of activities accordingly. In this scenario, a doctor initializes each guideline to be followed via Web interface; then, the service should assist the users involved in the guideline execution. The service should support different users (doctors, nurses, patient) by showing them the actions to perform next, on the basis of their role. The service should also automatize the activities that can be carried out via Web and compose heterogeneous services: e.g., the delivery of health-care products, the patient transportation and the scheduling of medical check ups. Flexibility and abstraction are key aspects of the service: in fact, an effective guideline management service should attract the user's attention on the execution of the important aspects of a medical guideline (e.g., which exams have to be scheduled next), treating low-level details about the guideline execution in a semi-automated way (e.g, in order to perform such

² We model patient and relatives as a single "patient" role as they may to receive similar information and support.

exams the patient has to be carried to a lab).

The rest of this paper is organized as follows: Section 2 discusses the open issues in the representation of context-aware medical guidelines and provides some background on the kind of approach we propose to adopt. Section 3 presents the guideline execution model we propose to adopt and sketches the framework we are developing. Section 4 concludes the paper.

2 Context-aware Execution of a Medical Guideline

A medical guideline can be described as a workflow executed by scheduling activities and meeting synchronization constraints. The activities may include both on-line and physical actions to be performed by different types of users. Moreover, the management of the guideline involves alerting such users when it is time to perform an action and assisting them in the action execution.

In principle, the set of active guidelines (all the medical protocols that apply to a patient) can be managed as parallel workflows whose progress depends on the patient's health conditions. However, the specification of all the factors determining how to personalize the execution of an activity leads to the definition of very complex and detailed workflows. For instance, each activity might be specified in different ways, depending on contextual conditions, and the workflow specification might become large and redundant if the guideline prescribes to perform similar activities. Furthermore, the selection of the most convenient alternative might require the evaluation of complex contextual conditions.

In order to support a highly adaptive execution of guidelines and to maintain a simple representation style, we propose to combine workflow management, which is particularly suitable for the scheduling of activities, with advanced activity execution techniques supporting context-aware behavior. Specifically, we propose to describe medical guidelines as metalevel workflows, whose activities map to different courses of action, depending on contextual conditions. The guidelines are managed by a workflow engine, but the individual activities are performed by an activity actuator which selects the course of action to follow on the basis of contextual information. In this way, the high-level management of a medical guideline can be separated from the details of the execution of the specific courses of actions to take.

In the Autonomous Agents research, the BDI (Belief Desire Intention) agent model [7,9,3,6,13] has been defined to support the development of deliberative agents acting in an environment to achieve goals in a context-dependent way. Among the various models of autonomous behavior that have been developed, BDI agents have proved to be particularly flexible and, at the same time, sufficiently lightweight to suit the requirements of realistic applications. Therefore, we propose to adopt the BDI model for the intelligent action execution component to be developed. The basic ideas characterizing the BDI model can be summarized as follows.

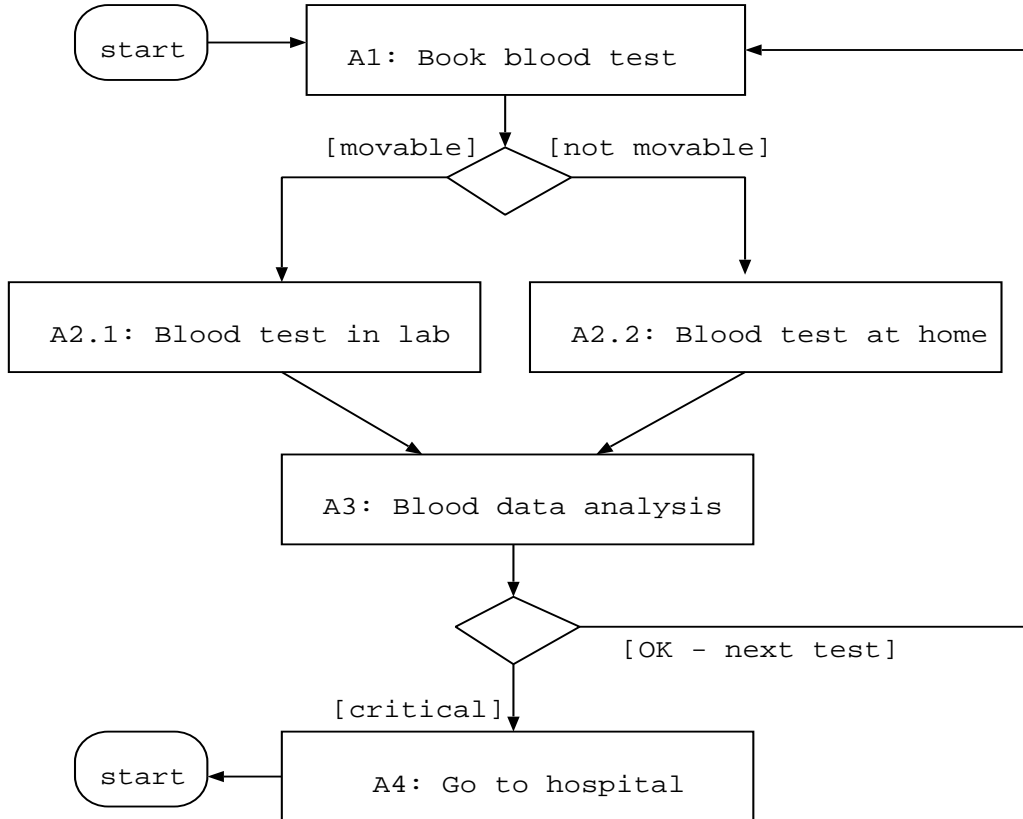


Fig. 1. Medical guideline specifying blood thinner management

- The capabilities of a BDI agent are described as *actions*: the agent has an *action library* specifying all the actions it can perform. Each *action* is aimed at achieving a *goal* (a boolean condition on the agent's beliefs, which concern the state of the world, as the agent perceives it) and is characterized by *applicability conditions* (contextual conditions determining the correct execution of the action); moreover, each action has a *body* describing a sequence of steps to be executed.
- A BDI agent acts in an environment to achieve its own goals, which may be permanent goals, or can be posted to the agent during its lifetime. The agent's behavior is based on the execution of an interpreter, which cycles testing the goals to be satisfied and sensing the world in order to select the next action to perform. Basically, the agent retrieves the actions aimed at achieving one of its goals, then it checks the applicability conditions of such actions, and it performs one of the actions whose applicability conditions are satisfied.

BDI agents have been applied in rather different application domains, such as softbots, Web-based information systems, and robotics; e.g., see [8,12].

3 Management of Adaptive Guidelines

Our framework supports the management of the guidelines at two granularity levels. The activities specified by a guideline can be executed by a workflow engine (Guideline Manager). At a finer grained level, when the Guideline Manager selects an activity for execution, it triggers an Activity Execution Agent which performs the action by taking contextual conditions into account. The connection point between the two levels is provided by the specification of the goals achieved by each individual activity. This means that the medical guideline is treated as a partial ordered set of goals to be achieved and the achievement of such goals is separately managed.

Guideline Management. The workflow representing a medical guideline includes a partial ordered set of *generic activities* to be scheduled. This workflow is represented in an executable process language; see Section 3.1. The Guideline Manager executes a workflow, starting from its entry point, by selecting each time the next activity A to be performed and invoking the Activity Execution Agent on A . For instance, Figure 1 depicts a medical guideline describing blood treatment: first, an appointment for the blood test must be defined (*Book blood test*); later, depending on whether the patient can be easily moved from his home or not, the blood test is carried out at his home or in a lab. Then the blood data is analyzed to plan either the date of the next test, or a visit to the hospital, depending on the patient’s health state.

Activity Execution. The main difference with respect to traditional workflow execution is in the way how each activity is performed. We propose to specify, for each (generic) activity, the main goal pursued by the activity, in order to separate the scheduling of goals to be satisfied (managed by the Guideline Manager) from the decision of which is the most convenient way to achieve such goals. The association of a goal to each activity supports the run time selection of alternative behaviors, depending on contextual conditions.

Each alternative behavior is represented as an action aimed at achieving the main goal of the activity and is characterized by applicability conditions and body, according to the BDI agent model. For instance, consider the medical guideline depicted in Figure 1. Activity $A1$ (*Book blood test*) is aimed at achieving goal *get blood test appointment*, which can be achieved by fixing the appointment at a lab and, if the patient cannot be moved from home, by requesting a nurse who goes to the patient’s home, takes the blood and carries it to the lab.

Figure 2 depicts the two actions in an informal way (see Figure 4 for the detailed representation of one of the two actions, in the formalism we adopt). Both actions are aimed at achieving goal *get blood test appointment* but they differ in their applicability conditions (*Appl condition* field in descriptor) and in the operations to be carried out (*Body*). The actions are a simple set of alternative ways to perform activity $A1$ of the medical guideline, but they

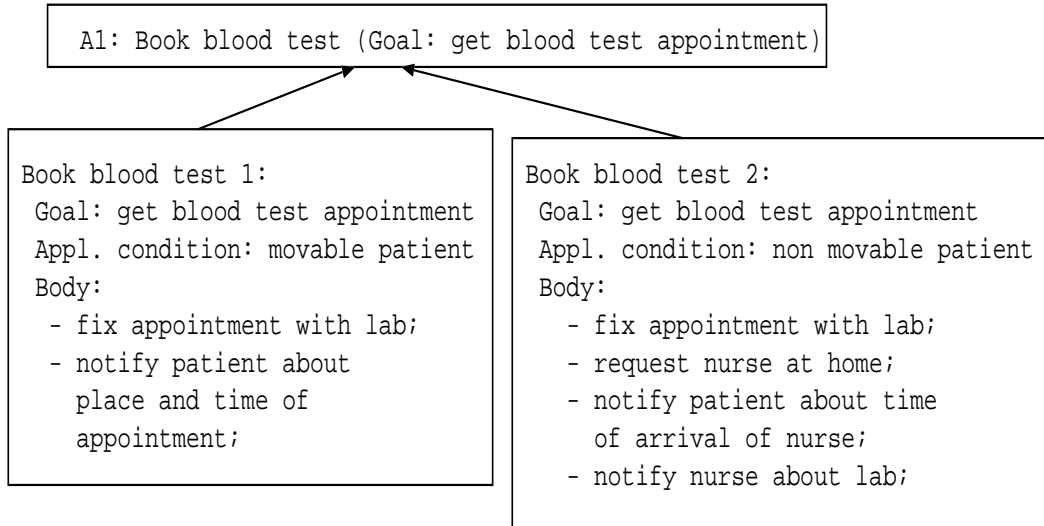


Fig. 2. Alternative ways to perform an activity

show a clear example of context-aware action execution: depending on the patient's health state (movable patient or non movable patient), only one of the actions is applicable and can be performed when the activity is scheduled by the Guideline Manager. Notice also that other alternatives (having more specific applicability conditions) could be defined, to further specialize the expected behavior.

Notice that the two actions prescribe the interaction with different types of users and the generation of alerts that are specific for the responsibilities of the actor involved. For instance, in *Book blood test 1*, the patient must be notified about the lab for the test; in the other action, the nurse must be notified about the appointment, too.

The Activity Execution (BDI) Agent selects the action to be performed (out of those associated to A , by checking their applicability conditions), executes it and returns the results to the Guideline Manager. The execution of actions may include the invocation of remote supplier services and the presentation of instructions to the user on the User Interface of the home health-care service.³ The generation of instructions depends on the role played by the user interacting with the service; e.g., a patient and a nurse have to be notified about different types of information.

3.1 Framework

Figure 3 shows the architecture we propose for the design of the system:

- Supplier boxes represent service providers exploited by the overall service.

³ Actions may include the invocation of Web Services but we impose that they do not represent a finer-grained workflow. In this way, the Activity Execution Agent does not need to be another workflow management system.

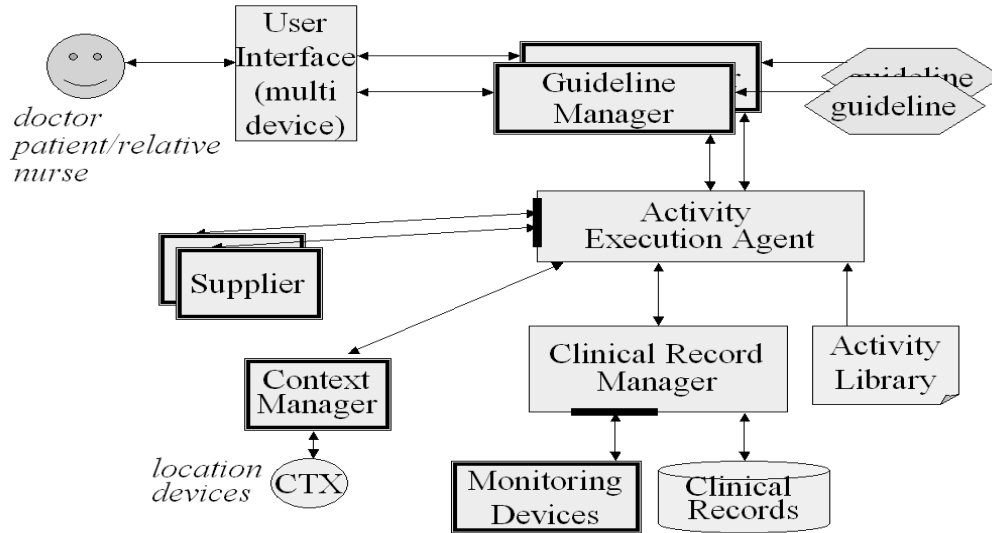


Fig. 3. Architecture

For instance, the nursery service and the emergency service providing ambulances.

- A Clinical Record Manager handles the patients' clinical records and enables the retrieval and storage of patient data (blood parameters, pending appointments, ...).
- A User Interface component manages the interaction with the user.
- A set of parallel Guideline Managers manage the workflows associated with medical protocols set up for treating specific health conditions of the patient.
- An Activity Execution Agent performs individual activities in a context-dependent way. The agent retrieves the associations between activities and actions and the definition of actions from an Activity Library. Moreover, it interacts with remote suppliers in order to request (one-shot) services. Finally, it retrieves information about the patient from the Clinical Record Manager.
- Monitoring Devices are instruments connected to the patient, such as pace makers. Although today this kind of machinery works in isolation, we assume that these devices send the patient data to the Clinical Record Manager in real time, as this is going to happen in the near future.

Notice that the architecture is distributed; the core of the system might be located within an intranet, but the suppliers might be remote services. In order to support the integration of such distributed entities, we defined some architecture components as Web Services. In Figure 3, thick boxes represent components developed as Web Services, while thick short lines represent the Web Service invocation capabilities of the component.

```

Book blood test 1:
  Parameters: patientID;
  Goal: getAppointment(patientID, bloodTest);
  Applicability conditions: movable(patientID);
  Body: {
    Appointment info = fixBloodTestAppointment(patientID);
    notify(patientID, info);
  }

```

Fig. 4. Representation of action *Book blood test 1*

We are developing our framework in Java and standard communication and representation languages. Specifically, the User Interface component is based on XML techniques to support multi-device access (desk-tops, cell phones, PDA). Moreover, for the specification and management of guidelines we have selected the BPEL process language [2] because it is emerging as a standard in Web Service composition and has some reference implementations available to manage the workflows; e.g., the Oracle BPEL Server and BizTalk BPEL server [10]. Each Guideline Manager exploits an engine working on a BPEL1.1 workflow that describes a medical guideline. Finally, the Activity Execution Agent wraps a lightweight BDI agent which we are implementing by exploiting the Seta2000 agent infrastructure [4]: the Seta2000 infrastructure can be exploited to implement autonomous agents that select, out of a set of alternatives, the actions to be performed, and the action execution ends up in the execution of a simple script including one or more steps that can be sequentially performed. Figure 4 shows the representation of action *Book blood test 1* prescribed by the Seta2000 infrastructure. Notice that the slots of the action are Java methods to be executed by the Seta2000 interpreter; this is rather common in the infrastructures for the development of agents (e.g., see [3] and [13]) because it supports an efficient selection and execution of actions at run time.

3.2 Management of Multiple Active Guidelines

If multiple guidelines apply to a patient, they are carried out as parallel workflows, each one managed by a Guideline Manager. The interaction between the users and the Guideline Managers is handled by JSPs which invoke (simple or composed) Web Services. Similarly, the activities within parallel workflows send results back to the users through invocation of appropriate JSPs. Of course, the active guidelines have to be coordinated, e.g., in order to prevent the scheduling of conflicting activities. The coordination must be carried out by synchronizing the access and update of the patient's clinical record, which includes not only the patient data, but also the pending appointments and tasks the patient is involved in. While the basic access synchronization is directly carried out by the DBMS, suitable policies to select dates avoiding conflicts have to be developed in our future work.

The parallel guideline management is the basis for the activation of emergency guidelines: a guideline can be started by a doctor (at guideline initialization time), but also by the internal system components. For instance, the Clinical Record Manager could be extended to start an emergency guideline which alerts the user that the patient must be taken to the hospital when anomalous conditions are detected in his record.

4 Conclusions

We presented a framework supporting the execution of medical guidelines tailored to contextual conditions concerning the patient and the execution environment. Our framework manages the main steps of a guideline by abstracting and separately treating the details affecting the execution of operations in specific contextual conditions.

The separation of workflow management from action execution simplifies the description of medical guidelines and enhances the flexibility in their execution. In fact,

- The description of the different ways to perform an activity can be separated from the workflow, therefore making the workflow concise;
- An intelligent component may be employed as an activity execution module which selects the best way to perform the guidelines, given the user interacting with the service and the patient state.

Notice that the management of metalevel guidelines does not exclude the presence of context-dependent paths in the workflow; the fact that a guideline evolves in different ways, depending on some contextual conditions, can be expressed by defining alternative goal achievement paths in the workflow.

References

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services - Concepts, architectures and applications*. Springer, 2004.
- [2] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services version 1.1. <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>, 2003.
- [3] AOS. JACK Intelligent Agents [tm]. <http://www.agent-software.com/shared/products/index.html>, 2002.
- [4] L. Ardissono, A. Goy, G. Petrone, and M. Segnan. A multi-agent infrastructure for developing personalized web-based systems. *ACM Transactions on Internet Technology (TOIT)*, 5(1):47–69, 2005.

- [5] J. Bardram. Applications of context-aware computing in hospital work. In *Proc. of Symposium on Applied Computing (SAC)*, pages 1574–1579, Nicosia, Cyprus, 2004.
- [6] F. Bellifemine, A. Poggi, and G. Rimassa. JADE - A FIPA2000 compliant agent development environment. In *Proc. 5th Int. Conf. on Autonomous Agents (Agents '01)*, pages 216–217, Montreal, CA, 2001.
- [7] M. Georgeff and F.F. Ingrand. Decision-making in an embedded reasoning system. In *Proc. 11th IJCAI*, pages 972–978, Detroit, Michigan, 1989.
- [8] J.A. Giampapa, M. Paolucci, and K. Sycara. Agent interoperation across multiagent system boundaries. In *Proc. of 4th Int. Conf. on Autonomous Agents (Agents'2000)*, pages 179–186, Barcelona, 2000.
- [9] N.R. Jennings, K.P. Sycara, and M. Wooldridge. A roadmap of agent research and development. In *Autonomous Agents and Multi-agent Systems*, pages 275–306. Kluwer Academic Publishers, Boston, 1998.
- [10] M.B. Juric, B. Mathew, and P. Sarang. Business Process Execution Language for Web Services. An architect and developer's guide to orchestrating Web Services using BPEL. <http://www.bpelbook.com/>, 2005.
- [11] M.P. Papazoglou and D. Georgakopoulos, editors. *Service-Oriented Computing*, volume 46. Communications of the ACM, 2003.
- [12] K.P. Sycara, A. Pannu, M. Williamson, and D. Zeng. Distributed intelligent agents. *IEEE Expert*, December:36–45, 1996.
- [13] Tilab. Java Agent DEvelopment Framework. <http://jade.tilab.com/>, 2005.