

# Personalising Situated Workflow Systems for Pervasive Healthcare Applications

Giovanni Russello, Changyu Dong, Naranker Dulay  
Imperial College London  
{g.russello, changyu.dong, n.dulay}@imperial.ac.uk

**Abstract**—In this paper, we present an approach where a workflow system is combined with a policy-based framework for the specification and enforcement of policies for healthcare applications. In our approach, workflows are used to capture entities' responsibilities and to assist entities in fulfilling them. The policy-based framework allows us to express authorisation policies to define the rights that entities have in the system, and event-condition-action (ECA) policies that are used to adapt the system to the actual situation. Authorisations will often depend on the context in which patients' care takes place, and our policies support predicates that reflect the environment. ECA policies capture events that reflect the current state of the environment and can perform actions to accordingly adapt the workflow execution. We show how the approach can be used for the Edema treatment and how fine-grained authorisation and ECA policies are expressed and used.

## I. INTRODUCTION

*Healthcare Applications* are characterised by the integration of software systems in healthcare environments. Real-world medical environments present several research challenges that need to be addressed for developing robust healthcare applications.

As a showcase for our approach, in this paper we focus on technology-assisted living, where the domains to be administered are units of personal living. Here, someone may be living alone, perhaps in sheltered housing, perhaps post-operative, perhaps with one or more disabilities [8], perhaps elderly and infirm. In this scenario, the healthcare applications are composed of several services to monitor the patient condition, and to assist the patient or the carer in performing the appropriate treatments.

Monitoring the physical condition of a patient is carried out by means of body sensors [9]. Several types of sensors are commercially available to measure blood pressure, blood sugar, pulse rate, etc. Another form of monitoring can be achieved using infra-red cameras that avoids the invasiveness of video surveillance. Combining infra-red cameras with motion detection, such as in the the Irisys technology [7], makes it possible to detect the number of people or animals such as guide dogs, that are present and to make a note of any visitors, in order to find out whether carers are visiting according to schedule. This information, together with the data gathered from sensors could be used to detect critical conditions for the patient and raise an alarm to summon some help.

Carers and relatives can visit the patient's house. This brings up several issues that need to be addressed. First of all, carers

and patients needs to be assisted in performing their tasks. Although carers have to comply to standard procedures, often treatments require to be personalised to tailor the needs of each patients. The system should assist carers in complying with personalised procedures. Another issue is that of managing resources and entities. Such management needs to be very flexible since entities and resources need to be added and removed according to the situation, for instance in response to an emergency call. Privacy is also another issue that needs to be addressed. Healthcare systems can be thought of as collaborative systems where access to sensitive medical resources should only be provided to authorised entities. Although privacy is a very stringent concern, security mechanisms should support dynamic binding of access rights to entities acting in the system. Finally, the overall system should be able to react and adapt to the actual context according to pre-defined rules.

To this end, we propose an approach where a workflow system is coupled with a policy-based framework. Workflows are used to assist entities in the execution of their tasks. Workflow systems are well established in the medical community. The notion of workflow as an ordered collection of tasks can be used for modeling several aspects of a medical environment. Workflow systems have been used for capturing diagnosis [4], therapy/treatment [6], [12], and hospital administration [5], [11]. Workflow specifications capture the responsibilities and obligations that entities have to fulfil. For example, a carer may set the schedule for the various forms of medication a patient must take. This can be complex and confusing when someone comes home from hospital with many new medications. Another example could be of carers visiting several patients with different conditions. Here carers must be assisted in performing correctly their tasks. Workflows can be personalized to the actual needs of each patient.

The use of a policy-based framework is an effective solution of management of entities and resources [13]. Policies can be specified for defining entities' access rights as well as event-condition-action (ECA) rules that can be used to adapt and react to contextual events. The access right policies are typically to control access to the patients medical data. ECA policies can be used to capture events associated with threshold values and patterns of sensed data values. For example, ECA policies can be specified for acknowledging and logging that a patient took a prescribed medication. The event could be generated by a swallow sensors [3] and the data logged in a local database. In case of an emergency ECA policies can

raise an alarm for summoning medical care.

The enactment of workflows is then realised by means of a workflow engine. To capture the interaction between an entity executing a workflow and the workflow engine, ECA policies are used. In this way, a feedback loop is created between the workflow engine and the environment where entities interact. On the one hand, the workflow engine can announce which is the next task to execute or how much time is left before the execution must be completed. On the other hand, events can be automatically sent when a task is completed by an entity (for instance by means of sensors), triggering ECA policies to communicate back to the engine the fulfilment of a task. If during the treatment the conditions of a patient change then ECA policies can detect such behaviour and dynamically adapt the execution of the workflow to the actual needs of the patient.

We illustrate the feasibility of our approach by means of an implementation of a healthcare application based on our framework.

The outline of this paper is as follows. In Section II, we provide a description of the workflow management system used in our approach. Section III gives an overview of how entities and resource are managed, and policies specified. Section IV we provided a scenario for the Edema treatment realised with our approach. We conclude in Section V.

## II. WORKFLOW MANAGEMENT SYSTEM

In our framework, workflows are used to capture the responsibilities of entities and the execution flow, that is the sequence in which the entity must execute the workflow's tasks. Workflow Management Systems (WMS) are particular relevant in medical situations where failure in performing an action could lead to serious consequences. In recent years, WMSes have been applied to medical environments. Several experimental workflow systems have been proposed for helping in carrying out diagnosis [4], treatment/therapy [6], [12] and for management in hospital administration [5], [11].

The WMS that we use in our framework is based on the YAWL environment [1], [2]. The YAWL environment provides a very powerful workflow language together with a workflow execution engine, and an editor for workflow specifications.

The YAWL environment can be customised to export to external components certain events that occur in the life-cycle of workflow instances. On receiving a task-enabled event, a component may decide to 'check-out' the task from the engine. On doing so, the engine marks the task as *executing* and effectively passes operational control for the task to the component. When the component has finished executing the task, it will check it back in to the engine, at which point the engine will mark the task as *completed*, and proceed with the workflow execution.

It should be realised that our framework is independent of the specific workflow language/engine used as long as the workflow environment provides means for interacting with our framework.

## III. POLICIES FOR HEALTHCARE APPLICATIONS

In our framework, access rights and ECA rules are specified by means of policies. Policy-based management is an effective solution for the distribution, automation and dynamic adaptation of systems used for supporting medical environments [13]. In this paper, we use the Ponder2 policy language developed at Imperial College London [10]. The language supports the specification of policies that are rules governing the choices in the behavior of a system [14].

The interpreter organises the entities on which policies operate in hierarchical domains of *managed objects*. A managed object has a management interface that the object has to implement in order to be managed by the interpreter. Domains allow the classification and grouping of managed objects in a hierarchy. Furthermore, domain paths can be used to address managed objects in policy specifications. Managed objects can be used to represent resources (e.g., data repositories, printers, X-ray machines, etc.), devices (e.g., sensors), and entities (i.e., nurses, doctors, GPs, etc.).

An assisted living scenario is highly dynamic, as entities, including medical professionals, carers and sensor equipment, may frequently enter and leave a particular care domain. This dynamism means the middleware must autonomously manage these entities, by identifying them, assigning the requisite privileges and triggering various events and/or obligation flows. Another aspect of entity management, particularly important in the healthcare domain, concerns the performance of an entity. Actively monitoring and evaluating the behaviour of an entity allows for a higher quality of service, by allowing active intervention whilst functioning as a deterrent against misbehaviour.

In our framework, entities (such as carers) are represented as special managed objects called *Entity Templates*. An Entity Template defines the type of entity to which it must be applied and it contains a set of data that the interpreter uses for carrying out management operations, including authentication, authorisation and the responsibilities (express in terms of workflows) that the entity has to fulfil. Before an entity can access resources of a domain structure it has to be authenticated. Entities are authenticated by means of credentials (appointment certificates) issued by the domains in which the carers are employed as proof of their qualifications and fitness for performing tasks. We envisage that such credentials might be carried on an electronically readable identification token, to be used for personal authentication on entry to a unit. For instance, peripatetic carers could use a decentralised network of certification authorities and using the notion of *trust* for authenticating only trusted entities.

Once the credentials have been authenticated, the authentication service finds the Entity Template that captures that set of presented credentials. Once the matching Entity Template is found a *Entity* managed object is created in the domain structure. When a *Entity* is created the system activates all the policies and workflows defined in the Entity Template. An *Entity* managed object represents an active entity in

the system. To make an analogy with the Object Oriented paradigm, an Entity Template is similar to a Class specification whether an Entity is like an instance of an Object.

The use of Entity Templates caters for a more dynamic management of entities and policies. In fact, the set of credentials, policies, and workflows that should be associated with a managed object can be dynamically retrieved from well-known repositories. For instance, in our scenario of sheltered home if the Entity Template of the doctor is not defined locally or is out-of-date it can be retrieved from an NHS repository. A domain structure can also import part of an Entity Template. For instance, the domain structure could allow the device used by a carer to provide workflows for new therapies that the carer has to execute when visiting the patient. Such workflows are imported in the carer Entity Template of the domain structure of the sheltered home. During execution, the therapies can be personalised to cater for the needs of the specific patient. These adaptations are made to the local workflows stored in the patient's home domain and therefore used only for that specific patient.

#### IV. IMPLEMENTATION OF A HOME CARE SCENARIO FOR EDEMA

In this section, we present the implementation of a simple healthcare application for the Edema treatment based on our framework.

Edema refers to swelling caused by excess fluid in body tissues. Monitoring involves a carer taking various measurements/performing tests. This allows us to show in more detail several aspects of our framework such as:

- 1) Authentication of the carer when entering the home healthcare domain.
- 2) Careplan for Edema treatment assigned to the carer
- 3) Carrying out treatment procedures
- 4) Departure of the carer from the home environment

Carers are provided with portable devices that help them in performing their tasks. In the scenario shown in Figure 1, a nurse carries with her a PDA. The PDA has already a domain structure instantiated where a Nurse Entity is already instantiated.

The domain structure for the sheltered home is instantiated on the home PC. In the sheltered home, resources (such as body sensors and records) are provided as managed objects that can be accessed by authorised entities. The structure also provides an Entity Template for Nurse that provides the set of policies required for the treatment of Edema pathology.

##### A. Entering the Home Domain and Assignment of Careplan

When the nurse enters the patient household, her PDA and the home PC discover each other. The discovery of the home domain will raise an event on the nurse device that triggers an ECA policy. This policy forces the local Nurse Entity to present the nurse credential to the Authentication Service of the home domain, as shown in Figure 1.

The following steps are executed: (1) after the Nurse Entity presents its credentials to the authentication service, (2) the

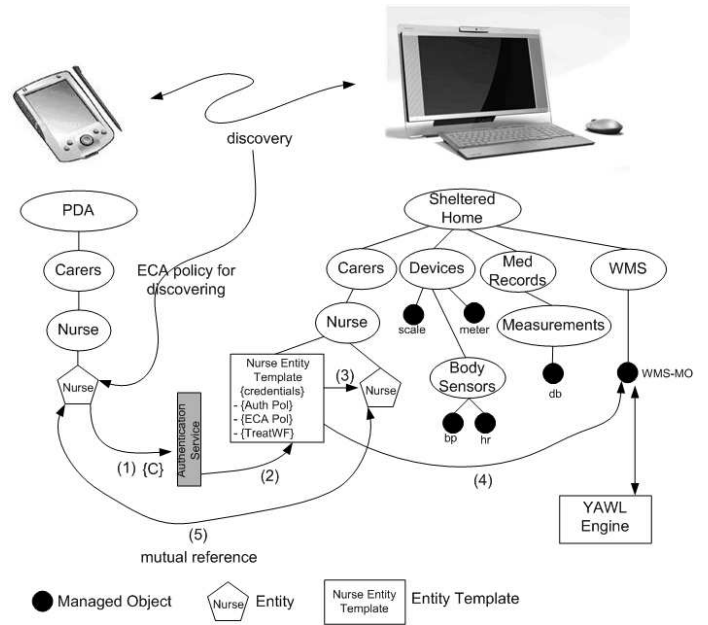


Fig. 1. Domain structures for a carer device and the shelter home devices.

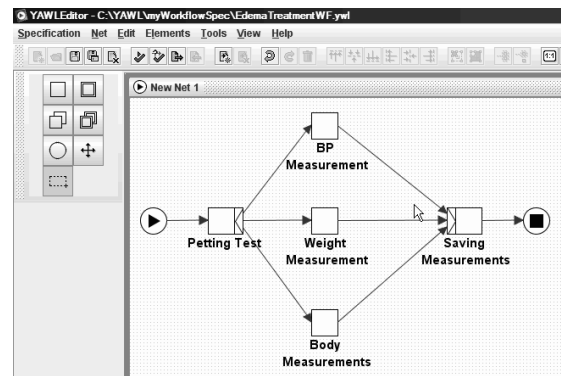


Fig. 2. Specification of the Edema Treatment workflow using the YAWL editor.

credentials are verified and the service finds the matching Entity Template that is invoked to instantiate the Nurse Entity in the local domain(3). Moreover, the Entity Template takes care that the set of authorisation and ECA policies are activated. The Entity Template also instantiates the workflow for the Edema Treatment that the Nurse Entity is required to follow. This is done by means of the WMS-MO (4), a managed object that is interfaced with the YAWL engine instantiated in the home domain. To conclude, the Nurse Entity in the PDA and the Nurse Entity in the home domain exchange a mutual reference (5). In this way, the nurse can follow the Edema treatment procedures for this particular patient using her device. The Nurse Entity instantiated on the sheltered home domain acts as a proxy for the Nurse Entity running on the nurse's device. As we will see in the next section, this allows the specification and enforcement of policies locally to the domain structure where the resources are instantiated.

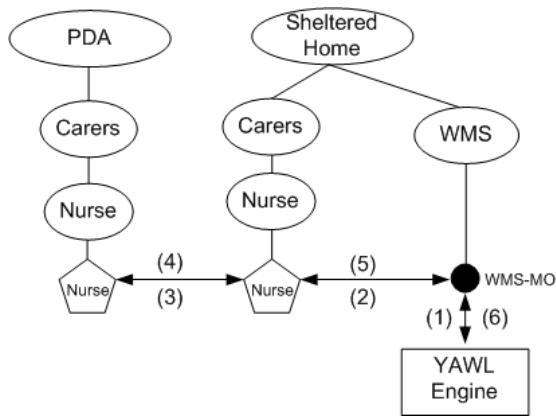


Fig. 3. Controlling the enactment of tasks.

### B. Treatment Procedures

The specification of the workflow for the Edema treatment is shown in Figure 2. The workflow is specified using the YAWL editor (shown in Figure 2) and then saved as an XML file. When an instance of the workflow needs to be instantiated, the workflow engine loads up the XML specification and enacts the first task.

The Edema treatment involves the following tasks. First the petting test is executed. Afterwards, measurements for blood pressure, weight, and body measurements are collected. In the end, the measurements are saved in the local database.

These measurements are performed using devices and body sensors, such as scale, meters and the blood pressure sensor (BPS) already present in the home. These devices are represented as managed objects in the domain structure of the sheltered house, as shown in Figure 1.

The nurse can use her device for following the workflow execution. The YAWL workflow engine is responsible for the enactment of tasks. Figure 3 shows how the engine communicates to the entities involved in the execution of a given instance of a workflow. Let us assume that the nurse completed all the measurement tasks. The only task left is the recording of the measurements into the local database. The YAWL engine notifies the WMS-MO which is the next task to be executed ('Saving Measurements') (1). The WMS-MO keeps track of which instance of an entity is involved in the execution of a workflow. In this case, the WMS-MO knows that the task is to be executed by the Nurse Entity and therefore it communicates the information about the task to be completed (2). The Nurse Entity in the home domain sends this information to the Nurse Entity instantiated in the PDA domain (3) that visualises the information to the nurse via the PDA screen. The nurse checks the measurements are correct and requests the logging of the information by pressing the 'log measurements' button that will appear on the PDA screen. Once the operation is completed, the task fulfilment is communicated back to the engine following the inverse path (steps (4), (5) and (6) in Figure 3).

The WMS-MO can be used also for adapting the execution

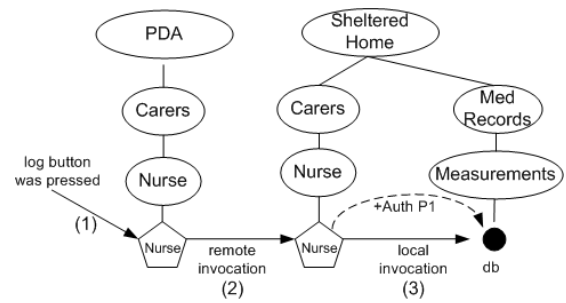


Fig. 4. Controlling access to resources via authorisation policies.

of workflows to the actual situation of the environment. For instance, if the biometrics of the patient present an anomaly, the sensors can raise an alarm/event that triggers ECA policies. Depending on the situation, the ECA policies can require the WMS-MO to stop the current workflow execution and switch to an emergency workflow to cater for the actual condition of the patient. The changing of the workflow is also notified to the nurse via her PDA.

To control access to the database, authorisation policies are used. As shown in Figure 4, once the nurse presses the button (1) the Nurse Entity will try to connect to the database in the home domain. The invocation of the action on the database goes through the Nurse Entity on the home domain (2). The local Nurse Entity (that represents an authenticate but remote Nurse Entity) must be authorised to execute such an action. In other words, there must exist an active authorisation policy that allows the nurse Entity to access the database for recording the measurements. As Figure 4 shows, such a policy exists (policy *P1* shown as a dashed arch from the nurse Entity to the database) and it is specified as part of the nurse Entity Template shown in Figure 1. The policy is specified as a positive authorisation for the Nurse Entity to access the method `recordMeasurement()` of the `db` managed object. Therefore the system will allow the execution of the action from the Nurse Entity on the database (3). The authorisation policy *P1* is activated when the Nurse Entity is instantiated.

### C. Leaving the Home Domain

When the nurse completes her visit and leaves the patient's home the resources allocated for her in the home domain must be relinquished. For instance, the Nurse Entity must be removed from the home domain and the policies associated with it deactivated. These operations are executed by means of an ECA policy that is triggered by the event `NurseOutEvt`. The `NurseOutEvt` is sent when the system detects that the nurse left the patient's home. When the ECA policy is triggered it disables the policies associated with the entity managed object and invokes the remove action of the domain where the Nurse Entity is contained.

The `NurseOutEvt` event can be sent in several ways:

- manually by the carer: the nurse presses a button to notify that the treatment is completed (according to her) and she is ready to leave.

- automatically by the system: when the nurse's device is out-of-range from the home PC or after a certain amount of time allocated for the carer expires.

These methods can be used in different situations. For instance, if the nurse has to leave before completing her procedures due to an emergency in another sheltered home, then either the manual disconnection or the out-of-range detection allows the system to promptly react (i.e., the system can report via an alarm that some procedures need to be completed). On the other hand, the time-out disconnection can be used by the system to recover from erroneous situation (i.e. the nurse left behind her device without notifying the system when she departed from the patient's house).

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we have outlined a workflow-based approach for the specification and enforcement of workflow for personalised and situated healthcare applications. In the approach, the workflow system is integrated with a policy-based framework. This allows us a more flexible management of resource, the enforcement of authorisation policies to express the rights that entities have in the system, and ECA policies to capture events and react to them in order to adapt the system to the actual situation.

Workflows are enacted by a workflow engine but are made context-aware by means of ECA policies. This allows the system to interact with the surrounding environments and dynamically adapt to it.

We showed how our approach could be used for Edema treatment and how fine-grained authorisation and ECA policies are expressed and enforced.

We are investigating how the notion of trust can also be associated with the authentication process. Trust allows a more dynamic authentication mechanism where the entity being authenticated is not known to the system in advance.

## ACKNOWLEDGMENTS

This research was supported by the UK's EPSRC research grant EP/C537181/1 and forms part of the CareGrid, a collaborative project between the University of Cambridge and the Imperial College London. The authors would like to thank the members of the Policy Research Group at Imperial College for their support.

## REFERENCES

- [1] W.M.P. van der Aalst, L. Aldred, M. Dumas, and A.H.M. ter Hofstede. "Design and implementation of the YAWL system." In A. Persson and J. Stirna, editors, *Proceedings of The 16th International Conference on Advanced Information Systems Engineering (CAiSE 04)*, volume 3084 of LNCS, pp. 142–159, Riga, Latvia, June 2004. Springer Verlag.
- [2] W.M.P. van der Aalst and A.H.M. ter Hofstede. "YAWL: Yet Another Workflow Language." *Information Systems*, 30(4), pp. 245–275, 2005.
- [3] O. Amft, G. Troster "Methods for Detection and Classification of Normal Swallowing from Muscle Activation and Sound." In *1st International Conference on Pervasive Computing Technologies for Healthcare*, December 2006.
- [4] L. Ardissono, A. Di Leva, G. Petrone, M. Segnan, and M. Sonnessa. "Adaptive Medical Workflow Management for a Context-Dependent Home Healthcare Assistance Service." In *Electronic Notes in Theoretical Computer Science*, Elsevier, 2005.

- [5] P. Dadam, and M. Reichert. "Towards a new dimension in clinical information processing." In *Stud. Health Technol. Inform.*, 77, pp. 295–301, 2000.
- [6] L. Dazzi, and M. Stefanelli. "A patient workflow management system built on guidelines." In *Proc. of AMIA 97*, pp. 146–150, 1997.
- [7] Irisys. <http://www.irisys.co.uk>
- [8] Y. Liu, J. Bacon, and R. Wilson-Hinds. "On smart-care services: Studies of visually impaired users in living contexts." In *First International Conference on the Digital Society (ICDS07)*, page 32. IEEE, Jan 2007.
- [9] B. Lo and G. Z. Yang. "Key technical challenges and current implementations of body sensor networks." In *IEEE Second Workshop on Body Sensor Networks (BSN 2005)*, April 2005.
- [10] The Ponder2 Project <http://ponder2.net>
- [11] M. Poullymenopoulou, and G. Vassilacopoulos. "A Web-based Workflow System for Emergency Healthcare." In *Medical Informatics Europ*, 2002.
- [12] S. Quaglini, E. Caffi, A. Cavallini, G. Micieli, and M. Stefanelli. "Simulation of a Stroke Unit Careflow." In *Medinfo*, 2001.
- [13] M. Sloman, J. Magee and K. Twidle and J. Kramer. "An Architecture for Managing Distributed Systems." In *Proc. 4th IEEE Workshop on Future Trends of Distributed Computing Systems*, pp. 40–46, 1993.
- [14] M. Sloman and E. Lupu. "Security and Management Policy Specification." In *IEEE Network*, pp.10–19, Vol. 16, Issue 2, March, 2002.