PERSONALIZED MEDICAL WORKFLOW THROUGH SEMANTIC BUSINESS PROCESS MANAGEMENT

Jiangbo Dang, Amir Hedayati, Ken Hampel and Candemir Toklu
Siemens Corporate Research, Princeton, NJ 08540, U.S.A.

Keywords: Medical workflows, Business process management, Semantic web services, Ontology, Knowledge base.

Abstract: Business Process Management (BPM) systems are becoming the runtime governance of emerging Service Oriented Architecture (SOA) applications. They provide tools and methodologies to design and compose Web services that can be executed as business processes and monitored by BPM consoles. Ontology, as a formal declarative knowledge representation model, provides semantics upon which machine understandable knowledge can be obtained, and as a result, it makes machine intelligence possible. By combining ontology and BPM, Semantic Business Process Management (SBPM) provides a novel approach to align business processes from both business perspective and IT perspective. Current healthcare systems can adopt SBPM to make themselves adaptive, intelligent, and then serve patients better. Our ontology makes our vision of personalized healthcare possible by capturing all necessary knowledge for a complex personalized healthcare scenario including patient care, insurance policies, drug prescriptions, and compliances. This paper presents a hospital workflow management system that allows users, from physicians to administrative assistants, to create context-aware medical workflows, and execute them on-the-fly using an ontological knowledge base.

1 INTRODUCTION

Workflows are becoming ubiquitous in enterprise applications from industry to healthcare. They encompass both system processes and human workflows. Herein, we use the process and workflow terms interchangeably. Semantic Web services (Berners-Lee, 1998) are intended to be applied dynamically by the services themselves through automatic and autonomous selection, composition, and execution. BPM systems provide real-world methodologies to govern semantic service selection, composition, execution, and monitoring. In this paper, an ontology is developed to describe a healthcare network including hospital resources and processes, and to serve as a knowledge base for our Web-based workflow management system. This ontology provides the semantic layer to BPM applications and helps business rules, enterprise policies, and running environment context be described at the semantic level. Therefore, services can be described, advertised from a business perspective, and these functionalities can be discovered, and composed by business specialists, instead of IT professionals. Moreover, ontology helps business rules, enterprise policies, and context to be described in a machine understandable way to support adaptive workflow composition and execution.

Most existing workflow systems don't support semantic process design or composition. They are not context-aware and do not support run-time process composition and execution. In contrast, this paper advances the state of the art by (1) developing an ontological to unify healthcare processes, resources, organizations, and compliances; (2) applying semantic to the lifecycle of medical workflows: from modeling to monitoring, (3)providing a prior knowledge base for task scheduling and resource allocation at semantic level, and (4) bridging the gap between IT and healthcare business. Our system allows users to: (1) create and deploy personalized medical workflow in the run time; (2) control and monitor the process flow that a patient will pass through; and (3) maintain historical process data for further diagnosis. Note that our intended users are healthcare professionals such as physicians, nurses, and administrative assistants who have no or little IT background. Our workflow system, together with the ontological knowledge base, will help these users to create, manage, and monitor workflows without intervention from IT people.

Dang J., Hedayati A., Hampel K. and Toklu C. (2009).
PERSONALIZED MEDICAL WORKFLOW THROUGH SEMANTIC BUSINESS PROCESS MANAGEMENT.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 122-127
DOI: 10.5220/0002006401220127

2 BACKGROUND

2.1 Semantic Web Services and BPM

Web Ontology Language for Services (OWL-S) (Coalition, 2004) is an initiative to facilitate automatic discovery, invocation, composition, and monitoring of Web services through their semantic description. It augments BPEL (Business Process Execution Language) processes with preconditions and results to encode the side effects. A process can be represented as a set of composed Web services. BPM systems provide process modeling tools, execution engine, and key performance indicators (KPIs) to monitor and measure end-to-end processes (in BPEL) against operational targets. BPM address business needs and design flexible processes that are based on services, which can be implemented later with SOA infrastructure. New and changed processes modeled in the BPM may be implemented in the SOA infrastructure more rapidly because the SOA decouples the described service from the specific implementation of particular service. (Laukkanen and Helin, 2003) and (Mandell and McIlraith, 2003) present methodologies for semantic Web service composition.

2.2 Ontology and Knowledge Base

An ontology is a schema composed of concepts and relationships among these concepts. A knowledge base may use ontology to specify its structure (entity types and relationships) and its classification scheme. Ontology, with its instances, constitutes a knowledge base that includes: (1) a set of concepts, properties, and relationships among them, (2) high-level rules in the form of constraints, (3) low-level rules defined in semantic rule markup language (Horrocks, 2004), (4) semantic service descriptions and. It builds the metadata needed by a program to understand its environment and status, reason about them, and then automatically compose services to create new tasks, which in turn can be scheduled with optimized resource while complying with business policies and compliances.

Siemens MED group (Dickmann, 2006) has used a methodological approach to define hierarchical medical processes. (Emanuele and Koetter, 2006) surveyed BPM and workflow technologies applied in current healthcare systems. (ST Liaw and Lewis, 2006) proposed a modeling methodology to define prescribing process using BPEL. Healthcare systems can adopt BPEL processes, semantic Web services, and ontology when dealing with workflows applied in a hospital. By doing this, these systems can deliver personalized medical workflows to patients, monitor workflow status and optimize the flow in real-time, and further coordinate these workflows to improve the service they deliver. This paper aims to present a system that delivers these features. In the rest of the paper, Section 3 introduces a motivating medical workflow scenario. Section 4 discusses the ontological knowledge base. Section 5 describes the system architecture with an emphasis on dynamic process orchestration. Section 6 presents methodology for process monitoring, and Section 7 concludes.

3 SYSTEM OVERVIEW

3.1 A Hospital Scenario

Here we introduce a simplified healthcare workflow with five participant roles: *RegistrationOfficer*, *DischargeOfficer*, *Nurse*, *Physician*, and *SupportStuff*. For each role, there might be more than one participant (instance). For example, one physician reviews patient's medical record, while another physician prepares for the operation. The ADMIT phase starts when a patient comes into a hospital and register in a waiting list. An administrative assistant can check the relevant data including insurance information, medical history, patient address, emergency contact, etc. Then the patient will go through the DETECT phase when a healthcare professional like physician might do some tests, such as blood test or X-ray test. The TREAT phase follows and treat the patient's disease for instance by applying a surgery and/or a plaster. The last phase is DISCHARGE when the patient leaves the hospital and all relevant data including executed processes is gathered for record and future diagnosis.

Due to uncertainty, it is difficult to handle all events and emergencies in a hospital. From a patient's view, patient may have to have the treatment plan updated from time to time. This uncertainty and dynamics exist because of: (1) newly come findings from test, unexpected outcome from treatment, (2) lack of resource, and (3) emergency. From a hospital's view, a hospital may want to dynamically route tasks among physicians and resources to provide better care, timely response with minimized cost.

3.2 System Interfaces

Our workflow system provides role-based online access to patients and hospital staffs. As shown in Fig 1, a user can log into the system with the corresponding roles. On his/her welcome page, there is a worklist that shows all the pending tasks for the user as well as the in-progress tasks. Using the worklist, the user can

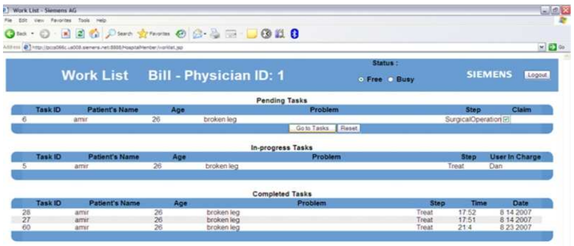

Figure 1: User Interface - Worklist.

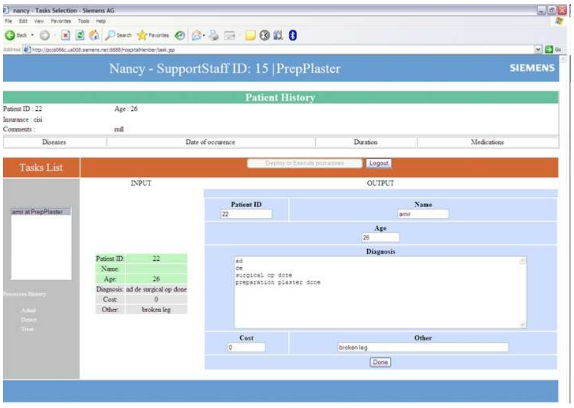

Figure 2: User Interface - Taskpane.

claim a task, review the patients which are still in the workflow. More flexibly the user can set his/her working status to allow or avoid newly coming tasks. All the status will be monitored for resource optimization and performance evaluation purpose.

Fig 2 shows a physician's task panel. A physician can check patient's medical record, add new observation, diagnosis information, and operation suggestion. As defined in our ontology, a physician have the right to deploy new processes. By clicking on the button "Deploy or Execute Processes", a physician can create a new process by choosing and arranging necessary atomic tasks from a task pool, which is validated by the ontological knowledge base, then deploy this process by clicking on "Apply Processes". All necessary files are created and the process is deployed on the server. After clicking on "OK", this process is executed and its status will be updated in real-time.

4 ONTOLOGICAL KNOWLEDGE

BPEL is widely accepted in industry but it does not provide semantic. The newly published WS-BPEL 2.0 specification does not talk about semantic annotations neither. As our effort toward Semantic BPM, we enhance current BPM with semantics by adopting the following methodologies: (1) extending the process groundings from WSDL to WSDL-S, (2) defining an OWL-based process model for process orchestration, and (3) mapping between OWL-S services and BPEL processes. In this paper, we combine (2) and (3), and develop an ontological knowledge base to facilitate process orchestration and execution. This ontological knowledge base also captures business intelligence behind the healthcare process hierarchy. By adopting the ontological knowledge base, we separate business rules represented in knowledge base from process logic defined in a process.

Figure 3 shows the schema of our healthcare ontology. This knowledge base covers the domains that a healthcare enterprise encompasses - in particular a hospital - from the medical or administrative tasks, to hospital assets, medical insurances, patient records, drugs and regulations. Our ontology include different views: roles, resources, organization, KPIs, and processes. Processes are defined in terms of their actors, inputs, outputs, preconditions, and results (AIOPRs). KPIs are also defined in the AIOPRs for monitoring purpose. All concepts in processes are defined in other views. Process view unifies all other views by: (1) consuming assets and people from resource view, (2) specifying actors with roles defined in role view, and (3) providing process activity information and business data to calculate KPIs from the KPI view. Business rules include business policies and logics. Properties, constraints, and relations are defined to describe high-level business policies. Meantime, low-level business rules including business logic are described explicitly with SWRL clauses. Therefore, it allows this ontological knowledge base to capture all necessary knowledge for a more complex scenario involving patient care, insurance policies, and drug prescriptions, and compliances.

5 ADAPTIVE WORKFLOW SYSTEM FOR HOSPITALS

5.1 System Architecture

As shown in Figure 4, We built our workflow system on Oracle Application Server and Oracle Business Activity Monitoring (BAM) server. In the Application Server side, we implemented and deployed Web services for process composition and execution. In the BAM server side, we defined BAM data objects and designed dashboards, and then monitored running processes. For the system front end, we developed role-based Web interface for end users, and use Protégé (Stanford,) as the ontology viewer and

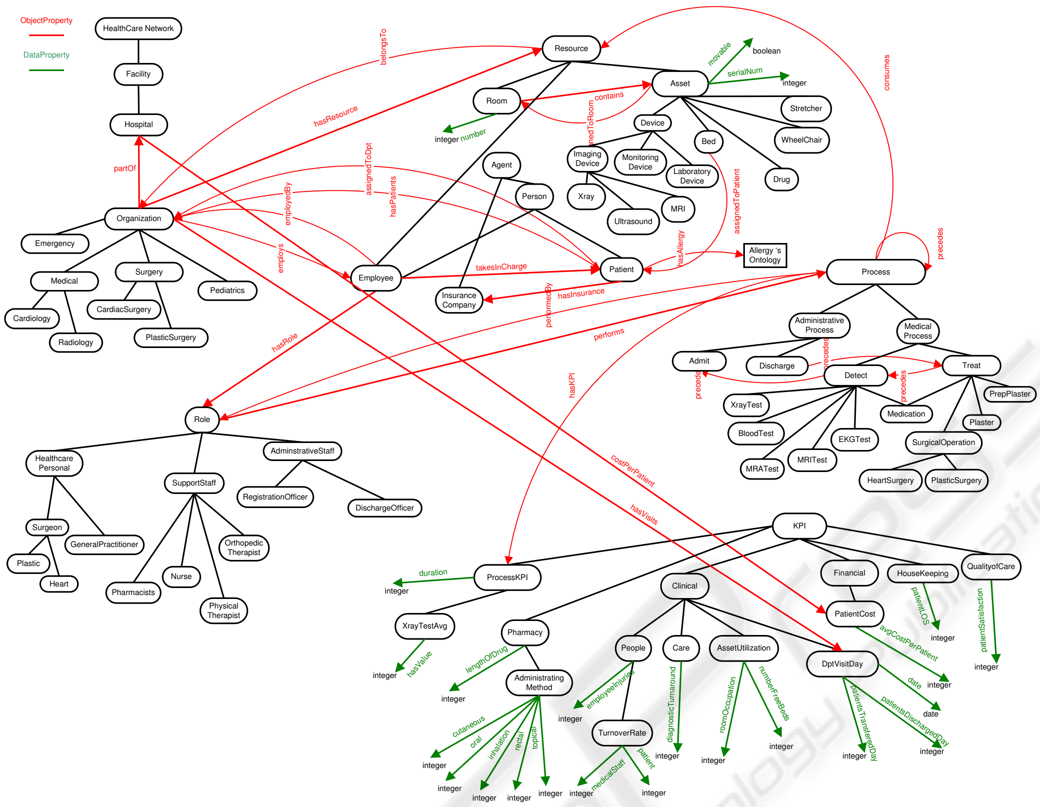

Figure 3: A simplified Hospital Ontology.

editor. For the system back end, we adopt Oracle database and MySQL database to store the ontological knowledge base and workflow data separately.

Task Assigner assigns tasks to the corresponding roles. When a patient registers, our system will automatically generate a patientID for this patient. Personal informationlike patientID, name and age are passed to the *Task Assigner* Web service. These information allows the *Task Assigner* service to generate a task and select a role for this task. The task is then inserted into taskpool in the database, waiting to be accomplished by the user who has the proper role. After that *Task Assigner* checks the status of the task. When it is accomplished, the task is marked as finished. The taskpool keeps track of all executed processes or pending processes.

Knowledge Retrieve Engine extends Jena API (api, 2007) to extract ontological knowledge. Jena API can be utilized to store and retrieve RDF, RDF Schema and OWL data. By combining a Jena query engine SPARQL and an OWL-S API (api, 2004) for Jena, we developed this knowledge retrieve engine to extract properties, classes, and instances of classes from the OWL specifications in our knowledge base. Meanwhile, we retrieve inputs, outputs, preconditions, and results from the OWL-S description, and provide the retrieved knowledge to the process composition service for service selection and composition.

Process Composition Service is called once a user has composed his own process and wants to deploy and execute it on the BPEL server. This service gathers information concerning *TransactionID*, *ProcessInstanceID*, *ParentID*, patient's name, and structure information regrading the newly composed process. After then this service deploys the new process onto the application server after creating all files needed by the BPEL server to properly deploy the process. Finally the service executes the newly deployed process on the BPEL server by calling it. At this step, A unique ID is generated and inserted into the process. The ID is returned to this service to monitor the status of sub-processes of the composed process. After execution, this ID is used to construct a tree of linked processes, which allows tracking the process history for future use.

5.2 Dynamic Process Orchestration

The dynamic process orchestration is one of the core features. Our system makes personalized medical workflow possible by letting healthcare specialists create and execute new medical processes for special

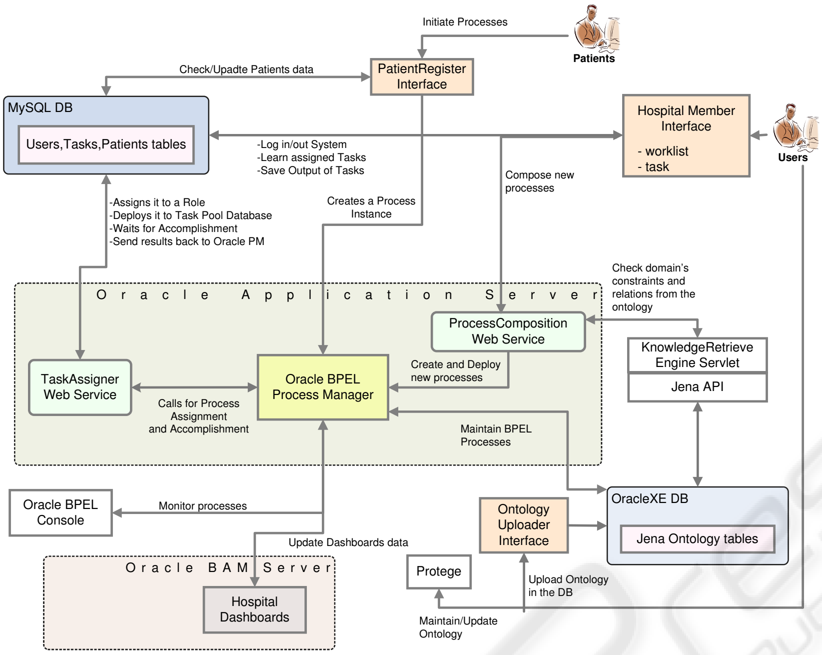

Figure 4: System Architecture.

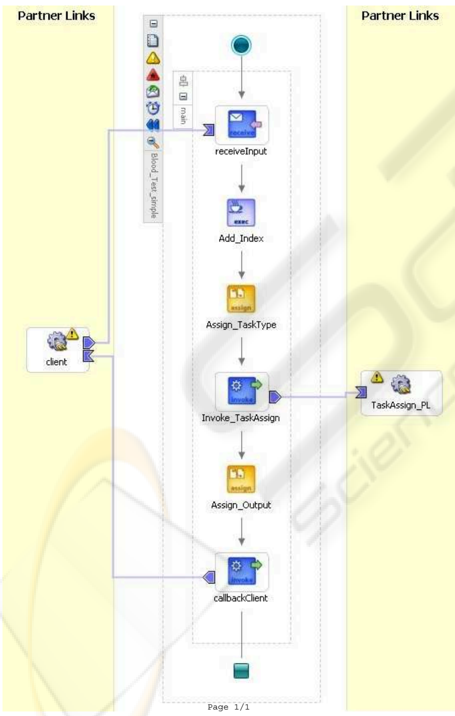

Figure 5: BloodTest Process.

patients without IT infrastructure or BPEL knowledge. This feature can be seen as the first accomplishment to bridge healthcare needs and IT technologies in a hospital. Meanwhile, our system maintains a process repository on the BPEL server. This repository allows users to save newly composed processes and reuse them in the future.

To support dynamically composed processes, we design one main BPEL process - *MainPatient* process encompassing the flow of all processes. The *MainPatient* process is triggered each time a patient registers with the system. In the *MainPatient* process, there are four sub-processes in sequential: *Admit*, *Detect*, *Treat* and *Discharge* . The whole process and all the calls to sub-processes are asynchronous to allow human interventions. Moreover, we develop a pattern for all atomic processes to make them compos-able using our methodology. Figure 5 shows an example of atomic processes. As you can see, the *BloodTest* process has the following components:

- A *receive* element receives events as a start point.
- An *Add_Index* element generates a unique ID for the current instance of the executed process. We use this ID to retrieve process information and extend it to retrieve the status of the executed processes. Moreover, we developed a context-aware process monitoring component, which utilizes this element to attach the newly deployed process to its context.
- An *Assign_TaskType* element updates the coming event by inserting the task name and sending it to the *Task Assigner* Web service.
- An *invoke* element invokes the *Task Assigner* Web service that will assign the task to a role, and will call back when the task is done by a user.

- An *Assign_Output* element further updates variables sent back by the *TaskAssigner* Web service. This element implements process-specific logics
- Finally, a callback to the requester if it is a newly composed process.

By adopting the above pattern, we define the template for all atomic processes and make the runtime process composition and execution possible.

6 BUSINESS ACTIVITY MONITORING

Business Activity Monitoring refers to the aggregation, analysis, and presentation of real time information about business activities inside organizations. It provides a real-time summary of business processes to business managers and upper management. A set of dashboards can be created to monitor running workflows and system performance, which can be utilized to further improve the performance.

6.1 Design BAM Objects

First, business analysts define KPIs that need to be monitored from a workflow. Based on the KPIs, BAM data objects are created as a set of variables, each of which is composed of multiple fields to store its property values. For instance, a field of process starting time and another filed of process ending time can be added and used to calculate the duration of a process. On the BAM server, these data objects will be filled with real-time data extracted from running processes. Dashboards are designed to display these data for different user groups. Figure 6 shows a hospital main dashboard with multiple views: Hourly Patients distribution, Average process times, etc.

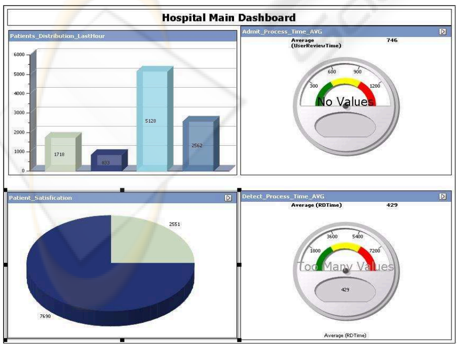

Figure 6: BAM Dashboards.

6.2 Map BAM Objects to Processes

BAM views depend on information from running processes. Therefore, the linkages between process data and BAM data need to be created. In our system, BAM Data objects are linked to process data via sensors and sensor actions. A sensor is configured in a process to gather data to the BAM server when a BPEL process is triggered. The activity sensors are created to gather process activity information. For instance, one can configure a BAM sensor to send the data of the time when a process is triggered and the time when a process is finished. Then on the BAM server side, a new field in the data object can be created to calculate the duration of the process by subtracting the starting time from the ending time.

7 FUTURE WORK

There are several directions for future work. First, we are going to fully explore semantic Web service to elaborate automated process orchestration and execution. Second, we will automate and optimize runtime resource allocation among processes. Third, we will adopt a semantic rule engine to enhance the knowledge retrieve and reasoning procedure.

REFERENCES

(2004). API. http://www.daml.ri.cmu.edu/owlsapi/.

(2007). Jena A Semantic Web Framework for Java. http://jena.sourceforge.net/.

Berners-Lee, T. (1998). Semantic Web Road Map.

Coalition, T. O. S. (2004). OWL-S: Semantic Markup for Web Services.

Dickmann, C. (2006). Workflow support in healthcare IT systems. Siemens MED.

Emanuele, J. and Koetter, L. (2006). Workflow opportunities and challenges in healthcare. Siemens MED.

Horrocks, I. (2004). SWRL: A Semantic Web Rule Language combining OWL and RuleML. http://www.w3.org/Submission/SWRL/.

Laukkanen, M. and Helin, H. (2003). Composing workflows of semantic web services. In *Proceedings of Web-Services and Agent-based Engineering*.

Mandell, D. J. and McIlraith, S. A. (2003). Adapting bpel4ws for the semantic web: The bottom-up approach to web service interoperation. In *International Semantic Web Conference*, pages 227–241.

ST Liaw, E. Deveny, I. M. and Lewis, B. (2006). Clinical, Information and business process modeling to promote development of safe and flexible software. *Health Information Journal*.

Stanford. Protégé. http://protege.stanford.edu/.