Developing a Service Oriented Process Management System for University Quality Assurance

PROF. DR. TAHER TAWFEK HAMZA
Dept. of computer science, faculty of computer science and informatics, mansoura university, Egypt
phone: +20105603913

DR. AHMED HASSAN
Dept. of electrical engineering, faculty of engineering, mansoura university, Egypt
phone: +20104992497

MOHAMED ALI ALHADIDY
Dept. of computer science, faculty of computer science and informatics, mansoura university, Egypt
m_hadidi@hotmail.com
phone: +20103622323

ABSTRACT
Quality assurance in universities is an increasing interest in quality and standards reflecting both the growth of higher education and its cost. Institutions should have a policy and associated processes for the assurance of the quality and standards of their programs. To achieve this, institutions should develop and implement a strategy for the continuous enhancement of quality. Formal policies and processes provide a framework within which higher education institutions can develop and monitor the effectiveness of their quality assurance systems.

Process management is most suitable model for automating quality assurance in university. Process management is about modeling, design, execution, and monitoring. In this paper we describe quality assurance process management system built on service oriented architecture (SOA). SOA has been chosen for the transparently, scalability, and Integrity of this architecture.

Key words: University Quality Assurance (UQA) – Business Process Management (BPM) – Process Modeling (PM) - Service Oriented Architecture (SOA)

1. Introduction
Procedural guidance can give more detailed information about the ways in which the policy is implemented and provides a useful reference point for those who need to know about the practical aspects of carrying out the procedures. The policy and the associated processes are expected to include[2]:
1. The institution’s strategy for quality and standards.
2. The organization of the quality assurance system.
3. The responsibilities of departments, schools, faculties and other organizational units and individuals for the assurance of quality.
4. The ways in which the policy is implemented, monitored and revised.

Process management is most suitable model for automating quality assurance in university, Quality Assurance process management system (QAPMS) can provide the following capabilities:
1. Ability to integrate with university systems as shown in figure 1, like university data management system, learning management systems, and document repository system.
2. Ability to integrate people and systems that participate in quality assurance processes.
3. Ability to simulate quality processes to design the most optimal processes for implementation
4. Ability to monitor, control, and improve quality assurance processes in real time.
5. Ability to effect change on existing quality assurance processes in real time without an elaborate process conversion effort.
The Process Management Life Cycle shown in figure 2 consists of phases that are related to each other. The phases are organized in a cyclical structure, showing their logical dependencies. The order of performing these phases can be modified by some different development methodologies[6].

2. Quality Assurance Process Modeling
In university quality assurance center, Process is a set of activities performed by its teamwork and monitoring activities performed faculties' quality assurance units. These units deal with sub-processes, each of which includes the set of activities performed by the teamwork of the unit, teaching staff, and students. The quality assurance process mainly involves two kinds of stockholders. Planners, who play the main role of planning scheduled tasks and activities, monitoring the execution of these tasks, approving the completed tasks, and sometimes execute parts of these tasks. The other stockholders are executers, who are mostly executing tasks like assessments, and documents preparing. The use case of the quality assurance process can be represented as in figure 3.
When the planner initiate the plan, he must identify the main properties of the plan like the expecting start and end dates of the plan, plan reviewer, approver role. Then he starts identifying the tasks of this plan. Tasks are scheduled, ordered, prioritized, and must have specific user(s) to execute it. The planning process model is shown in figure 4.

![Figure 4. Planning Process Model](Image)

Each Task are represented by a set of sub-tasks (activities) having the same properties of task. One task could have many scheduled activities that should be executed by one or more users, note that the approval status of a task is engaged to the approval of its all activities, this will provide monitoring advantage on the execution level. Each task or activity could have a risk management substitute one. That should be included automatically in the plan when the main task is not executed in its dedicated time. Execution is shared between users and admins, both will have some particular action to give the task or the activity a known status. Figure 5 shows the execution process.

![Figure 5. Task/Activity Execution Process Model](Image)

During the execution of plan tasks, the admin is responsible for monitoring the plan execution to take the appropriate actions like redesigning the plan either by adding/removing tasks, or changing in schedules and priority.

3. QAPMS Implementation Methodology
A process implementation has several elements that must be developed and interoperate to accurately reflect the requirements contained in the original quality assurance process. These elements usually include automated functions, manual procedures, user interfaces, task management, and plan management [6].

3.1. Automation Process model
The automation process model is defined by the model shown in figure 6. The primary input to the model is the quality assurance requirements as represented by the quality assurance process. Secondary inputs are (1) university standards that will affect some aspect of the development or deployed process and (2) stakeholder needs, which represent the interests of the different classes of individuals interacting with the deployed process. The output is the implemented and deployed business process the users employ in performing their work.

3.2. Process Implementation Architecture
Service oriented approach is strongly related to process management systems. As the system components can be represented as a set of separated services. The SOA model shown in figure 7 reflects on scalability, integrity, transparency, and performance.
The system architecture mainly consists of a set of layered components shown in figured 8. The services layer is grouped into the categories:

a. Entity services
Services that primarily manage access to entities, like students data or assessments results. These services are independent of process services.

The database entity services implement the IEntityDAO interface which abstracts the implementation of a set of functions like Add, Update, Delete, GetAll. An example of task entity service is shown in figure 9.

![Figure 8. Service Composition](image-url)
b. Decision services

Services that execute rules to provide decisions. Decision services generally provide yes/no answers to complex questions, or they support frequently changing externalized rules, such as tax regulations. The set of decision services are presented in the following table.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approvals</td>
<td>Includes the set of functions to manage the approval life cycle of plans, tasks, and activities.</td>
</tr>
<tr>
<td>Risk Manager</td>
<td>Perform the risk management tasks.</td>
</tr>
<tr>
<td>Scheduler</td>
<td>Responsible for timing of tasks execution and additional activities like sending e-mails.</td>
</tr>
<tr>
<td>Rules</td>
<td>Manages the rules of execution and users</td>
</tr>
<tr>
<td>Constants</td>
<td>Contains the main configuration keys and their corresponding values.</td>
</tr>
<tr>
<td>Role Rights</td>
<td>Controls the privileges of each role of the system.</td>
</tr>
<tr>
<td>Communication</td>
<td>Communicates the system to other university environment system.</td>
</tr>
</tbody>
</table>

Table 1. QAMPS decision services

c. Process services

Services that implement a quality assurance a function, activity, or task. They represent the process management system main functionalities like planning, monitoring, task management, and rule management. These services are interacting together to implement and enhance the process as shown in figure 10. The Planning service depends on the task management and the rules service to implement the process, an example of the output process is shown in figure 11. Then the monitoring process can adjusts plans, rules and tasks to enhance the process.
The integration service layer integrates functions and or data from existing systems and exposes itself as a service. Combining these different types of services together provides flexible capabilities that support the activities of quality assurance process.

3.3. Process implementation functionality

The standard in Web services composition is the Business Process Language for Web Services, WS-BPEL, or BPEL. It is the outcome of a merger of the Web Services Flow Language by IBM and XLANG by Microsoft[12]. Web Services Flow Language can be considered an XML serialization of Flow Definition Language, the script language that was used in IBM’s work-flow system, enhanced by concepts to access Web services. Data dependencies are specified by data flow between activities. Process behavior is specified by transition conditions.
attached to control flow links. XLANG is a block structured language that was used in BizTalk, Microsoft’s enterprise application integration software, focusing on the integration of heterogeneous back-end systems using processes. In block-structured languages, a strict nesting of control flow blocks is used to structure business processes. The language can be used to characterize both abstract processes and concrete processes. Abstract processes describe the externally visible behavior of a business process. They mainly serve communication purposes, so operational details are not obtained. Concrete processes contain information required to execute the Web services of the service composition. The following types of activities are available in BPEL[11]:
1. Invoke: Invoke an operation offered by a Web service; this invocation may or may not have a response
2. Receive: Wait for a message to arrive
3. Reply: Send a reply in response to a receive message
4. Wait: Wait for a specified time period
5. Assign: Assign data values, for instance, from received messages to process variables
6. Throw: Indicate that an error has occurred; used for exception handling
7. Terminate: Complete the process
Activities in BPEL can be related to each other using the following control flow structures:
1. Sequence: Define a block consisting of an ordered sequence of activities
2. Switch: Based on an expression, select a particular activity from a set of possible alternatives
3. Pick: Wait for a suitable message to arrive or for a time-out event. On receipt of the message (or the time-out event), start a defined activity
4. While: Execute a set of activities as long as a condition is evaluated to true
5. Flow: Concurrently execute a set of activities
6. Link: Execution constraint between activities
An example of a quality process expressed in BPEL is graphically represented in figure 12.

![Graphical representation of Web services composition in the BPEL format](image)

**Figure 12.** Graphical representation of Web services composition in the BPEL format

**4. QAPMS**
QAPMS will help developing, monitoring, and optimizing quality assurance strategic and action plans. Quality assurance stakeholders will be able to automate their plans and task and monitor its execution. The system contains two main modules. The process designer: Planner can specify roles, tasks performed by the various roles, and the sequence the process should follow. The quality tasks could be human tasks like on site auditing and writing quality reports, another type tasks that can be automated like monitoring the schedule of execution, planning, or statistics.
The process designer allows the planner to set the conditions and the flows of these exceptions and sub-processes. Once the quality plans have been set, tasks and activities will be presented to users in the executers’ module. The system forms will be generated for each task and activity with maintaining scheduled and risk management tasks. The web presentation has a familiar look and feel that is more acceptable for users.

The system will serve two kinds of stakeholders:

1- Executor
   He will receive tasks assignments to execute. He will have the ability to modify the status of the task. When the executer login to the system, a list of his pending tasks will appear. And he will be able to execute his assigned tasks as shown in figure 13, update the status of the task. The task will be completed if the user have finished all the task activities, these activities could be an assessment, document uploading (see figure 14), or checklist marks. After the user completes executing the task, an e-mail will be sent to the task creator to approve the completion of this task.

   ![Figure 13. Assigned tasks list](image1)

2- Planner
   a. He will have the ability to manage the quality assurance plans, tasks, and their associated actions. Also he can assign the tasks for specific executers in a time frame (estimated time for execution) as shown in figure 15. Then he will be able to approve or reject the execution of any task/action and finally he can monitor the whole plan to adjust it if it needs. Create/update plan. The user will be able to set the plan actual starting date/ending date, the approval status of the plan, and the plan owner.

   ![Figure 14. Document uploading activity](image2)
b. Create/update tasks. Tasks are created explicitly and then added to plans. Some tasks are common in the QA process, this assures eliminating duplication of cost. Tasks consists of sub-tasks (activities) which has the same properties of tasks like priority, order, schedule, owner, estimated/actual start/end dates of the task, task status, and its approval status. An example of tasks management is shown in figure 16.

c. Create/Update activities, which can be categorized into three types (i) Document uploading activity (ii) Checklist marking activity (iii) user defined activity.

d. Add/Update system users.

e. In addition, admin can monitor the execution progress, and status of his plans through plan monitor shown in figure 17.
5. Conclusion

QAPMS serves as a supervisory system that controls workflow in university quality assurance center. The supervisory aspect of QAPMS provides the abilities to monitor, control, and improve quality assurance processes. Process owners can obtain statistics such as the wait time before a task is performed and cost data. To help quality assurance team in planning and process design, using the process designer the admin can design the initial plans which will be enhanced and optimized through the monitoring process. The QAPMS will give the quality assurance team the ability to implement real-time process improvement without extensive process conversion effort. The original quality plans are already exist in process designer. Then, when there is a need for process improvement the quality team will be able to set a new process by modifying the schedule of tasks execution or add/remove tasks from the process. Using QAPMS process improvement could be made without disruption to process output which is very useful to continuous improvements.

SOA approach is agile in responding to usual changing needs and process improvements. It speeds up the application development process and systems become more adaptable. Furthermore, the client, which communicates with the services, is independent of the services itself. It does not have to know about the platform, the services run on, or the programming language they are written in; which provides more scalability and transparency of the system.

6. References

[12]Van der Aalst-W.M.P., Pattern-Based Analysis of BPML (and WSCI), 2002