**Chapter Four – Part II**

**COURSE MANAGEMENT SYSTEM**

1. **Introduction**

This chapter presents the proposed CMS. Proposed CMS facilitates integration among different CMSs. An automated course search, import, and deposit process is presented. This automated process requires governance of business rules. Business rules might be limiting system efficiency, so they must be monitored and modified when needed. Manage business rules process is presented to achieve this goal. Utilizing SOA to integrate Web services and software agents in CMSs highlighted the unlimited advantages of Web services and its capabilities to facilitate software agents’ integration within systems.

1. **CMS Analysis**

Proposed CMS addresses two processes namely Search and Manage Rules as depicted in the use case diagram at figure 4.8. Search process enables instructor and CMS to search for courses within course repositories managed and maintained by internal and external CMSs, import, and display those courses. There are managerial issues that should be addressed to enable this process. Managerial issues are managed at Manage Rules process.



Figure 4.8: CMS Use-Case

* 1. **Proposed CMS Architecture**

Figure 4.9 depicts the proposed CMS architecture that enables automation of courses integration. From architectural perspective, proposed system consists of two main layers that present its frame: Presentation layer, and Service layer.

**Presentation Layer**

It is the layer that presents user interface. System’s scope is limited to instructors and business managers. Those are the end users of the system. Display service is the service that is responsible for preparing dynamic output prepared to users, calling the suitable web page to display data. More than web page is prepared to address different functionalities. Display service passes suitable set of data to appropriate pages, and acquires input from users incase there are. Separation of interface design and implementation from business logic has proven many advantages.

**Service Layer**

Service layer consists of three sub layers: Orchestration Layer, Application Services Layer, and Agents Layer.

**Orchestration Layer**

Orchestration layer manages interaction details required to ensure that service operations are executed in a specific sequence. Sequences are determined based on processes supported by system. Orchestras within this layer are: Course Manager, Rules Manager, Exception Manager, Financial Manager, Mail Manager, Faculties Manager, Instructor’s Manager, and Feedback Manager.

**Application Services Layer**

Application Services are set of stateless services perform certain. Process is the summation of tasks performed by one or more application services layer in the sequence that is maintained at orchestration layer services. Services of Application Services Layer are: Discover/Recommend Display, Check Capability, Import, Insert Rule, Pay, Raise Exception, Manage Rules, Send Mail, Manage Courses, Search Process Exception Manager, Manage Rules Exception Manager, Manage Faculties, Manage Instructors, and Manage Feedback.



Figure 4.9: Proposed CMS Architecture

**Agents’ Layer**

Specific task agents are required to serve system. Software agents are always the optimum solution for track and analysis tasks. Software agents are: Discoverer Agent, Ranker Agent, Analyzer Agent, and Tracker.

* **Discoverer:** It is the software agent that is responsible for preparing list of external course providers and indexing metadata about courses.
* **Ranker:** It is the software agent that is responsible for ranking different system’s components based on ranking rules. System components that should be ranked are: Universities, Faculties, Instructors, and Courses.
* **Tracker:** Proposed CMS depends heavily on instructor’s feedbacks to enhance system’s performance.
* **Analyzer:** Analyzer is the software agent that is responsible for analyzing previously stored failed imports by detecting the most cause of failed imports and suggesting new conditions to decrease such problem. Suggested rules are inserted temporarily in the suggested rules database, waiting for the approval/denial of business manager.

Course Management System is one of the Learning Management System components that is interested in automating the activities related to courses discovery, downloading, and tracking.

* 1. **Search Process Analysis**

Search process is initiated by Instructor, managed and maintained by Course Manager. Figure 4.10 depicts analysis of Search process. Highlighted shapes present activities performed by course manager. Other shapes present non-course manager activities. Non-course manager activities are presented within the architecture as stand alone services that are called explicitly by course manager.

**Figure 4.10: Search Process Analysis**

* 1. **Manage Rules Process Analysis**

Manage Rules process is initiated by Business Manager, managed and maintained by Rules Manager as shown in figure 4.11. Highlighted shapes present activities performed by rules manager. Other shapes present non-rules manager activities. Non-rules manager activities are presented within the architecture as stand alone services that are called explicitly by rules manager.

**Figure 4.11: Manage Rules Process Analysis**

* 1. **Software Agents Analysis**

This part presents the design process of the four software agents highlighted in the CMS. The four software agents are: Discoverer, Ranker, Tracker, and Analyzer.

**2.4.1 Discoverer Agent Analysis**

It is the software agent that is responsible for preparing list of external course providers and indexing metadata about courses. Discoverer agent makes use of standard SOAP/XML based Web services that integrates internal and external applications [1]. Discoverer software agent integrates internally with exposed Web services like Course manager in order to insert data and metadata about courses. Figure 4.12 depicts Discoverer software agent architecture. There are three main interfaces for Discoverer agent: Internal interface, external interface, and human interface. Discoverer software agent integrates externally with educational institutions exposed Web services in order to initiate and maintain an internal catalog of external courses providers and courses. Web services technologies enabled integration flexibility and maintainability [1]. Internal interface presents integrating Discoverer software agent with CMS’s Web services on service level. Discoverer needs to consume Manage Courses service in order to update courses database. Discoverer is not directly inserting data into courses database. All operations have to be done through Manage Courses Web service in order to obtain service level integration benefits. Human interface is required to enable instructor and business manager insert data about courses and courses’ providers. Programmer interface is required to inform Discoverer about available external Web services. Universal Description Discovery and Integration (UDDI) provides a platform-independent way of describing services, discovering businesses, and integrating business services using the Internet [30].



Figure 4.12: Discoverer Agent Architecture

**2.4.2 Ranker Agent Analysis**

It is the software agent that is responsible for ranking different system’s components based on ranking rules. System components that should be ranked are: Universities, Faculties, Instructors, and Courses. Different formulas used to rank each of them. Ranker agent integrates internally with the proposed system on service level via system’s exposed Web services. Ranker agent consumes suitable Web services to insert, update, retrieve, or delete items that to be rank. Figure 4.13 depicts a Course Ranker process design. Course Ranker consumes two services in the early phase of running, and another two services as depicted.

**Figure 4.13: Courses Ranking Process Analysis**

**2.4.3 Tracker Agent Analysis**

Proposed CMS depends heavily on instructor’s feedbacks to enhance system’s performance. Figure 4.14 depicts of tracking instructor’s feedback of imported courses process design.



**Figure 4.14: Imported Courses Tracker Process Analysis**

**2.4.4 Analyzer Agent Analysis**

Analyzer is the software agent that is responsible for analyzing previously stored failed imports by detecting the most cause of failed imports and suggesting new conditions to decrease such problem. Suggested rules are inserted temporarily in the suggested rules database, waiting for the approval/denial of business manager. Analysis of failed imports tend mainly to increase system overall performance by adapting system to meet/exceed instructor’s expectations. Figure 4.15 depicts Analyzer agent process analysis.

**Figure 4.15: Analyzer Agent Process Analysis**

1. **Design of CMS**

CMS design include design details of the Search, Manage Rules Process Design, and designing the software agents. Each process design includes defining composing services and designing each of them.

**3.1 Search Process Design**

Figure 4.16 depicts Search process design. Search process utilizes more than one service. Services that are utilized by Search process are: Discover/Recommend Display, Check Capability, Manage Courses, and Import.

**3.1.1 Discover/Recommend Service**

It is the service that is responsible for searching internal and external list of ranked courses to prepare a list of ranked courses that satisfies user query. Recommend is the service responsible for ranking results prepared by Discover service by relevance of its keywords and user query keywords. Detailed design for Discover/Recommend activities is depicted in figure 4.16 under Discover/Recommend side title.

**Figure 4.16: Search Process Design**

Figure 4.17 depicts tables required by Discover/Recommend to achieve tasks. Courses\_External\_Ranked, and Courses\_Internal\_Ranked are the tables required by Discover to prepare list of courses. Keywords table is the table required by Recommend to rank list by relevance to user’s keywords.



Figure 4.17: Tables of Discover/Recommend Service

* + 1. **Display Service**

It is the service that is responsible for preparing data that will be displayed, passing this data to the proper page at presentation layer, and acquiring input from end user. Display service manages different kinds of data. Those data include:

* Display list of courses: at presentation layer, there should be a display course page that is waiting for the dataset that will be passed by course manager.
* Display Notification message: about rules, there should be a display rules page that is waiting for the dataset that will be passed by course manager.

Display service takes input from instructor and business manager. Example of instructor input is Course\_ID. Example of Business Manager Input is Rules\_Approved. Figure 4.16 depicts detailed activities performed by Display service to support Search process.

* + 1. **Check Capability Service**

It determines either the course satisfies organizational rules or not, by doing so, it actually determines if course can be imported or not. Rules not only include financial issues; like limits provided for each instructor, but they also reflect pedagogical aspects of the educational institution. Educational institutions must guarantee certain pedagogical level of courses. Check Capability Service requires database tables depicted in figure 4.18.



Figure 4.18: Tables of Check Capability Service

* + 1. **Import Service**

It consumes other services to achieve the required task, which is download course file and update database. In order to update database, import service calls Manage Course appropriate services. Download Course It is the functionality that holds programming logic that is responsible for downloading the course file into FTP server.

* + 1. **Manage Courses Service**

Manage Courses refer to the capability to implement basic databases operations upon certain entity. Basic database operations are: insert, update, and delete [29]. As courses are the entity of interest to search process, add, remove, and update operations should be supported. Courses are classified into three categories: Internal, External, and Imported. Internal courses refer to courses developed by institution’s instructors. External courses refer to courses that are available in other educational institutions and have not been imported yet. Imported courses refer to downloaded courses form external sources. Internal and External courses are ranked in different tables. Manage Course Service is capable of performing basic database operations on those different tables. Figure 4.19 depicts required database tables to support basic database activities.



Figure 4.19: Tables of Manage Courses Service

* + 1. **Pay Service**

Responsible for depositing money into accounts determined by the transaction. Figure 4.16 depicts detailed activities by Pay service. Pay service should manage transactions carefully by adding transaction entry to transactions table before attempting the process for it to be able to rollback non successful transactions. Figure 4.20 depicts required table to support financial transactions by the Pay service.



Figure 4.20: Tables of Pay Service

* + 1. **Raise Exception Search Process Service**

Incase any kind of errors happened during the Search process, a Search Process Exception is raised. Raise Exception Service performs two main functionalities: Adds an entry to Failed Imports table, and adds an entry to Log table. Figure 4.16 depicts activities performed by Raise Exception Search Process Service. Figure 4.21 depicts required tables by Raise Exception Search Process Service.

**Figure 4.21: Tables of Raise Exception Search Process Service**

**3.2 Manage Rules Process Design**

Figure 4.22 depicts Manage Rules Process Design. Manage rules process is initiated by Business Manager and managed by Rules Manager. Manage Rules Process utilizes the following services: Retrieve Suggested Rules, Display, Insert Rule, and Raise Exception.

**Figure 4.22: Manage Rules Process**

* + 1. **Manage Rules Service**

Manage Rules Service supports basic database operations on rules. There are three rule’s categories: Rules, Analysis Rules, and Suggested Rules. Rules table presents rules of educational institution to rank courses and determining applicability of course import. Analysis rules table is the table that holds analysis rules will be used by Analyzer agent to apply them on failed imports, in order to generate suggested rules. Suggested rules table is the table that holds either new rules or edited current rules suggested by Analyzer agent and waiting for approval by Business Manager. Figure 4.23 depicts tables required by Manage Rules Service



Figure 4.23: Tables of Manage Rules Service

* + 1. **Raise Exception Manage Rules Service**

Activities performed by Raise Exception Manage Rules Service are depicted within Figure 4.16. Raise Exception Manage Rules Service requires only log table in order to add entry. Figure 4.21 includes Log, Events, and Description\_Items tables design.

* 1. **Software Agents Design**

This part will cover the design of two of the systems software agents. Those are: Tracker, and Ranker.

**3.3.1 Tracker Agent**

Proposed CMS depends heavily on instructor’s feedbacks to enhance system’s performance. In proposed system, tracking focuses mainly on instructors filling their feedback forms. Every instructor imported a course should give her/his feedback for this course. Imported courses feedback tracking requires consumption of three Web services. Figure 4.24 depicts tracking required tables.



Figure 4.24: Imported Courses Tracking Required Table

**3.3.2 Ranker Agent**

Course Ranker consumes two services: Retreive Course, and Update Course. Pedagogical and managerial rules are presented with different priorities to reflect system’s interest. Figure 4.25 depicts ranking rules required tables. Ranking is either based on instructor’s feedback, or mathematical formula. Courses, Instructors, and Faculties ranking is based on instructors’ feedback for those different items. Several feedback forms with one to five grading are presented to instructors to evaluate different items. Rank items and inserted values are saved, and Ranker gets average of all those items to present final rank of any of the saved items. Thresholds are presented in different tables to enable Ranker to classify ranked item either accepted ‘ranking value over threshold’ or not accepted ‘ranking value is below threshold’. Universities ranking is presented as a formula that is the average of all faculties composing this university ranks.



Figure 4.25: Tables of Ranker Software Agent

1. **Summary**

This chapter presented a proposed CMS that facilitates integration among different CMSs. An automated course search, import, and deposit process is presented. This automated process requires governance of business rules. Business rules might be limiting system efficiency, so they must be monitored and modified when needed. Manage business rules process was presented to achieve this goal. Utilizing SOA to integrate Web services and software agents in CMSs highlighted the unlimited advantages of Web services and its capabilities to facilitate software agents integration within systems. CMS should be thought of as a collection of stateless Web services. SOA provides a fine granularity and modularity that solves many integration problems, but adds complexity to systems design. SOA is a design pattern that helped enterprises overcome integration obstacles, and gain agile and interoperable advantages within architectures. Pedagogical, social, and managerial advantages of added processes include:

* Overcome lack of internal courses.
* Get use of external, higher pedagogical features courses.
* Shareability among different educational institutions.
* Competition increment adds to quality (indirect effect).
* Increase Return-On-Investment by selling courses.