

Architectural Proposal for a Syllabus Management System using the ISO/IEC/IEEE 42010

Anthony Meza-Luque¹

Bantotal Perú S.A.C

Arequipa, Perú

Alvaro Fernández Del Carpio², Karina Rosas Paredes³

Jose Sulla-Torres⁴

Systems Engineering Program

Universidad Católica de Santa María

Arequipa, Perú

Abstract—The efficiency in the academic and administrative procedures of higher education clearly marks competitive advantage in aspects of quality, which consists in the continuous improvement and improvement for the achievement of educational objectives. In our institution, the syllabus treatment is currently carried out manually, delaying many educational processes. Therefore, it is proposed to innovate through a software architecture approach based on the standard “ISO/IEC/IEEE 42010: Systems and software engineering - Architecture Description” to describe the architecture Syllabus Management System software. It is developed in three stages: Analysis, Design and Verification. This will allow professors to develop their research, training, teaching and presentation of timely reports, with the measurement of achieved skills and abilities of students, managers and academic authorities, and to make decisions based on the results obtained by the tool allowing an improvement in the quality of the contents and development flow of the syllabus.

Keywords—Management; architecture; software; syllabus; skills; ISO/IEC IEEE 42010

I. INTRODUCTION

The syllabus is a plan for teaching and learning, as it contains the important meaning between the professor and students. However, most current program management systems offer simple functionality including creating, modifying, and retrieving the unstructured program [1].

The management of syllabus in a quality educational environment within a teaching-learning process has become an urgent need for universities [2]. The “Universidad Católica de Santa María” (UCSM) has obtained the ISO 9001 standard for its quality management systems (QMS) in order to meet the needs of customers and stakeholders. Thus, it is very appropriate to implement good practices specifically in the continuous improvement for the achievement of educational objectives.

The properly chosen software architecture helps to overcome potential problems and allows you to take advantage of this model [3]. For the software architect, it is essential to understand what a software architecture ready application is and what requirements it must meet.

In this context, for the academic management of the educational content of the subjects taught at the university, they have been innovating through a software architecture approach, improving the accessibility and usability of software products

through standards for minimize errors in the development of syllabus, an important instrument in the development of subjects.

Currently there are several proposals for software architecture, but as far as it has been reviewed, very few oriented towards syllabus management. In [4], they conducted a comparative study of the structure of the curriculum at Latin American universities, discovering that Syllabus are still seen as a registration document and not as a learning tool.

For this reason, the Syllabus Management System is proposed and developed in three stages. The analysis stage, with the detail of the functional and non-functional requirements, visits and interviews with the directly involved actors (professors and managers) to gather needs and current problems that lie mainly in response times since the process is carried out manually. The design stage, that starting from a prototype, automates the manual process, generating information on subjects from the Center for Academic Development to the different professional schools. This process minimizes common mistakes that the professor made, focusing only in sections relevant to its work. Finally, the verification stage, to validate the results of the previous stages. The architecture description can be modeled using the ISO/IEC/IEEE 42010 standard that allows for alternative options and decisions to be selected, where the rationale and trade-offs for each decision are documented and understood as necessary to inform subsequent decisions to stakeholders [5]. These concerns combined with the environment and system scenarios provide an architectural design context that clarifies the motivation to make decisions.

These functionalities and characteristics will allow professors to develop their research, training, teaching and presentation of timely reports. The measurements of skills and abilities achieved by students, managers and academic authorities are included in the proposal as well as to make decisions based on the results obtained. This tool helps to get an improvement in the quality of the contents and development flow of the syllabus through an intuitive and friendly system.

This article is organized as follows: the background is explained in Section II, the methodology in Section III, and the results of experimentation in Section IV, and finally the conclusions and future work in Section V.

II. BACKGROUND

In universities, the syllabus document defines the contents of a course, and other important information that ensures the quality of teaching-learning. This document, in addition to capturing the contents of learning, includes the mechanisms for the achievement of learning and the development of skills and abilities. It is an instrument in which the contents are defined by units or themes, credits are specified, in some cases prerequisite courses are established and the learning outcomes to be achieved are specified.

In the literature, various proposals have emerged for the construction of technological supports and mechanisms that help the good development of study programs. Such is the case of [6], which presents an abstract hierarchical model for syllabus management based on Model Directed Architecture, specifying a set of attributes such as levels, hours, credits, etc. It generates various types of reports, such as for accreditation and general purpose. The model is linked to the Open Syllabus system, a system for creating, editing and publishing syllabus.

In [7], an XML-based syllabus repository system integrates syllabus information from a set of universities was proposed. This system includes a template for data entry, the process of integration and search of syllabic contents. This system makes it easier for students to search for courses.

Similarly, [8] developed a service model for syllabus in order to standardize the method of classification of syllabus items and allow collaboration between environments. They modeled the syllabus scheme to facilitate the search between different domains, developed the automatic generation of the syllabus based on a markup language and implemented multilayer search agents. In addition, the model includes data creation, storage and retrieval functions.

Likewise, in order to assist in the search for topics of the courses in the study programs, [9] introduced a web-based tool using probabilistic models. This allows to identify the degree of similarity between the content of a syllabus in relation to a given topic. The tool extracts similar courses from a set of highly ranked universities. The benefit of the tool is twofold, as it helps students understand courses and professors to improve their study program.

In itself, World Wide Web technologies have transformed the design, development, implementation and deployment of decision support systems. The DSS web-based academic literature focuses primarily on applications and implementations, and only a few articles examine architectural issues or provide design guidelines based on empirical evidence [10]. For the development of technological proposals, as is the case of the present investigation, the software architecture approach is important, since it improves the accessibility and usability of software products for the benefit of high quality advanced education systems throughout the world [11]. It also allows improving learning experiences through collaborative services that are context-aware: software architecture and prototype system [12].

In [13], they analyzed the courses of archival study programs in North America. In identifying the convergences and divergences of the topics, they sought to understand the relationship between the two courses and obtain information on how these courses continue to serve as an integral component of archival studies education. The research examined three different aspects of the syllabus: textbooks, required articles and weekly topics. The syllabus was analyzed as separate data sets (RM syllabus and ERM syllabus), which was followed by a comparative analysis of the two types of syllabus. This may allow the Design of Knowledge Management Syllabus [14].

Júnior, Misra and Soares [15] indicate that software architecture as a development product is useful for technical activities, such as describing the views and concerns of the future software products, as well as for management activities, including assigning tasks to each team and as an input for project management activities. A major problem in describing software architecture is knowing what elements should be included in the architecture and at what level of detail, towards a Reference Architecture [16].

In [17] address the problem of how the software architectures of a System of Systems (SoS) should be described. For this purpose, they present an approach based on the standard "ISO / IEC / IEEE 42010: Systems and software engineering - Architecture description" to describe the software architectures of a SoS, as well as to develop an approach to modeling SoS using an architecture description language (ADL) [18].

As described, most jobs vary in the application of ISO/IEC/IEEE 42010. In addition, the approach used depends on the nature of course management at a University. Taking into account that most of these works are implemented for the design of general architecture, that is why in this work we try to use the suggestions of the standard in the university academic field. The objective is to use the guide of the ISO/IEC/IEEE 42010 standard to obtain a good software architecture in the management of syllabus in the university.

III. ELABORATION: ARCHITECTURAL PROPOSAL

Given the need for improvement within the Syllabus Management process in the UCSM and with the aspiration to have a standard and minimize errors in its preparation; the analysis, design and results points are detailed as follows:

A. Analysis

An agenda of meetings with interested parties was established to have a dialogue and understand the context in which syllabus management is currently taking place. These meetings allowed us to understand the process cycle from general to detail (see Fig. 1). The current guidelines established by the Center for Academic Development of which all professors use are also documented.

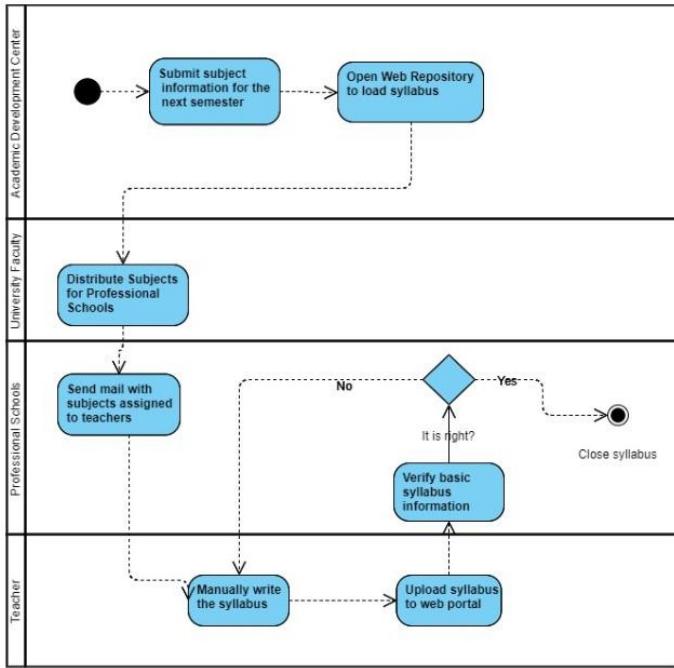


Fig. 1. Current Syllabus Management Process.

With the information collected we proceed to detail the functional and non-functional requirements that will contribute to the improvement within the process of Syllabus Management.

Functional Requirements:

- Allow the creation, modification and elimination of users.
- Perform the creation, modification and elimination of roles.
- Navigate correctly in the menu corresponding to the assigned role.
- Consult the syllabus entered through a query interface.
- Be able to update the status of the syllabus to move on to its next stage.
- Execute the information load, transferring it from the database of the “Universidad Católica de Santa María” to the database for Syllabus Management.
- Develop the content of the syllabus focusing on the sections that require professor participation.
- Generate PDF file as final product after developing the syllabus correctly.

Nonfunctional Requirements:

- The information entered into the system will be protected from unauthorized access and disclosure.
- Ensure the integrity of the information that has been entered into the system. Redundant or unnecessary information will not be allowed.

- Ensure adequate access to users according to their assigned role. This will happen through users that professors already have assigned with Windows environment authentication.
- Facilitate the understanding of the system with the help of user manuals, which will allow professors to have a flow guide implemented in the system.
- Allow optimal supports to be made by documenting the design of the system that will help the maintenance user understand the system agilely.
- Implement a system incident log by means of error and audit logs, so that anomalies that have occurred may be monitored.

B. Design

A prototype of the Syllabus Management system is currently under development that allows automating and streamlining this process. For this, the process definition has been made and a new model to be executed is proposed (Fig. 2).

This new model automates two important points:

1) In the first place, there is automation of the issuance of information on the subjects by the Center for Academic Development towards the different professional Schools. After executing the SSIS packages, the predefined load of already established information is performed and the Syllabus Management database is fed for the process to be carried out.

2) Secondly, the professor only focuses on developing those sections within the syllabus that cannot be obtained with a predefined load. This minimizes common mistakes.

C. External System Architecture

In this section, we present an overview of the standard ISO/IEC/IEEE 42010 entitled “Architecture Description” [19] on which our approach is based.

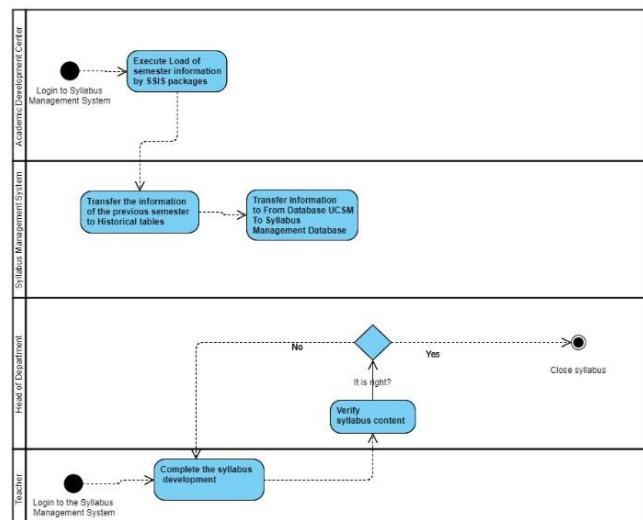


Fig. 2. Proposal for the New Syllabus Management System.

In Fig. 3, a system is situated in an environment. It exhibits an architecture expressed by an architecture description. A system has a number of stakeholders. Each stakeholder has interest in the system presented by a number of concerns.

To obtain an architecture description, the first step is to identify the stakeholders having concerns considered fundamental to the architecture of the system and to identify these concerns. The second step is to identify the architecture viewpoint: providing a name for the viewpoint, providing a listing of architecture-relevant concerns to be framed by this architecture viewpoint, providing a listing of the typical stakeholders of a system and identifying each model kind used in the viewpoint. To identify a model kind, we must provide its name, describe the conventions for models and identify the notation used in models [17].

1) Concerns and stakeholders: We present the stakeholders of the Smart Building System:

- Professors (Professors) of the Professional Program are the users and operators of the system.
- The department heads of the faculty are the ones who review the system.
- A software development application represents courses and their syllabus, developers, builders, and system maintainers.

Then, we present, the concerns considered fundamental to the architecture of the system:

- The purposes of the syllabus management system are to guarantee the correct entry, validation and monitoring of syllabus for courses.
- To guarantee the suitability and feasibility of the architecture to achieve the purposes of the system, our objective is to implement this system based on SOA technologies.
- To intercept the risks and potential impacts of the system for its stakeholders throughout its life cycle and to ensure the maintenance capacity and evolution capacity of the system, we will use a monitoring mechanism that allows intercepting SOAP messages [20].

2) Viewpoint and model kind: For the Syllabus Management System illustrated in our case study, we propose a structural viewpoint. In fact, a structural viewpoint deals with the purposes of each system participating in our proposal, the suitability of the architecture for achieving the system's purposes and the feasibility of constructing and deploying the system.

The external architecture of the system will be based on web (see Fig. 4) and the interaction can be done through the internet or intranet of the university. Professors connect through Windows authentication with the users they were provided with. The information is validated and access is provided.

A relationship is generated between the UCSM database and the database used to manage the syllabus. The database feed to manage the syllabus is done through the execution of SSIS packages. The SSIS packages were developed in the Microsoft SQL Server Integration Services tool that provides the necessary platform to perform required data extraction, transformation and loading (ETL) from one Database to another.

The server that supports the web application is the same where the various solutions that the UCSM have put into production for their development are hosted.

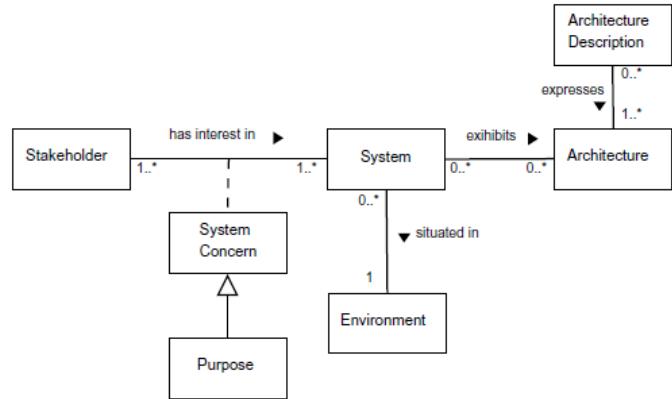


Fig. 3. Context of Architecture Description [19].

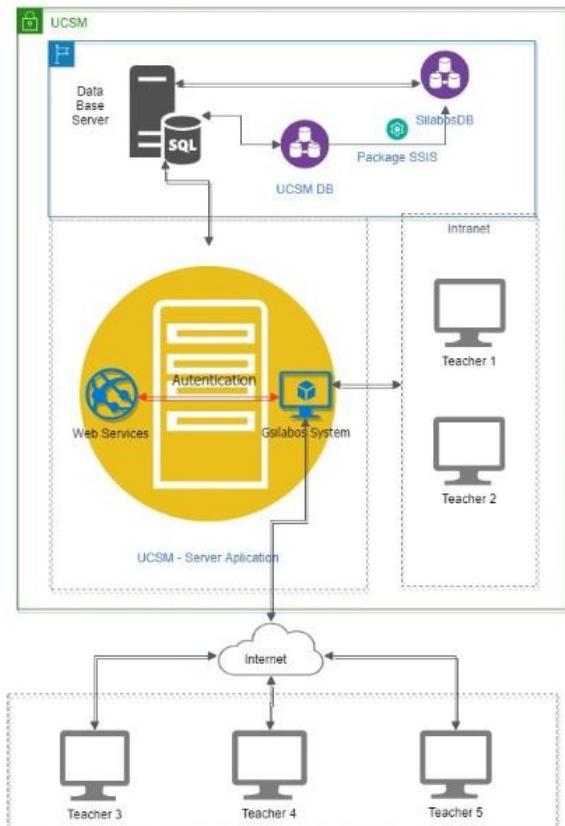


Fig. 4. External Architecture of the Syllabus Management System.

D. Internal System Architecture

The internal architecture of the system is based on MVC (see Fig. 5). Various technologies that enriched its functionality and aesthetics were integrated into this three-layer architecture.

For the view part, the Bootstrap 4 framework was integrated, which will help with typography, forms, buttons, boxes, navigation menus, etc. Also, the JQuery library was added to manipulate the DOM elements effectively. AJAX to be able to handle asynchronously the obtaining or sending of information without reloading the page. Data tables to show the information in the queries and that these are looked at in a distributed and dynamic way.

ReportViewer, a component that allows you to design reports, was used in this case to generate the final product that is the syllabus in PDF format. Razor was used to embed C # code in views and add functionality. Log4net was applied to be able to store the record of errors or incidents for audit. Linq and Lambda were used to manage the information in the database.

Likewise, Codefirst was applied to build the database model, and SQL Server was included as the database engine where the system data will be hosted.

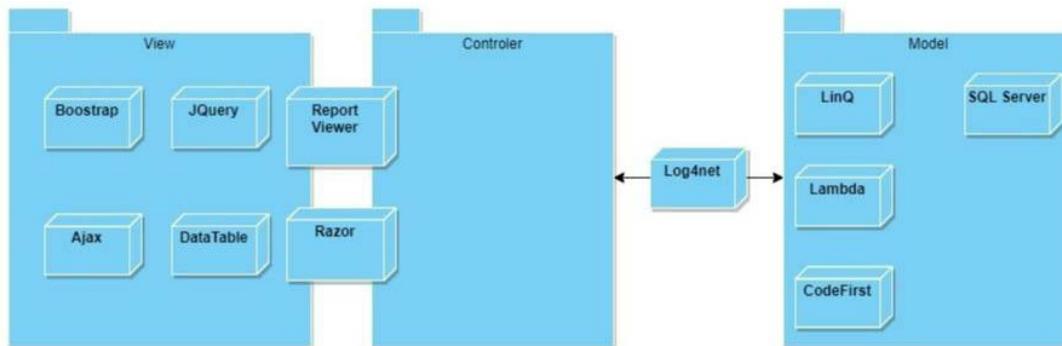


Fig. 5. Internal Architecture of the Syllabus Management System.

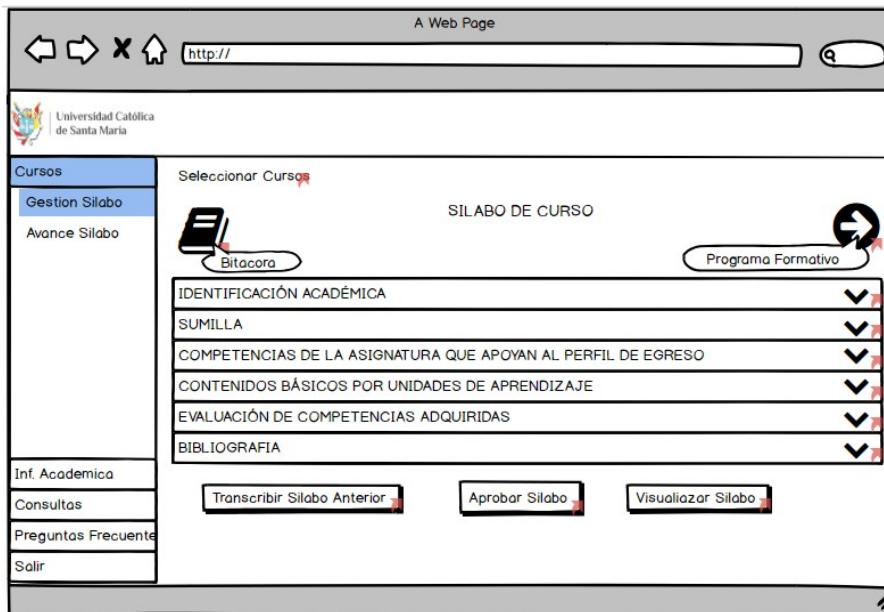


Fig. 6. Drop-Down Menu for the Syllabus Section.

E. Main Screen Design

This section presents the designs of the main screens of the syllabus management system. Fig. 6 shows the interface of the syllabus structure. From here, we proceed to enter or modify the syllabus information.

Fig. 7 shows the specification of the contents and competences grouped by units as well as the matrix corresponding to the research activities according to the learning phases of the academic semester.

Fig. 8 presents a summary of the file upload status. In this section the SSIS packages are executed and their processing is verified; if it was successful or had an error.

Fig. 9 shows the generation of the report in pdf of the syllabus document, according to the format established by the UCSM.

F. Testing

The tasks for the implementation of the support tool for Syllabus Management were tested, and the following results were obtained (see Table II). The Priority Task is in Table I.

Fig. 7. Drop-Down Menu of the Training Program Section.

Fig. 8. Load Menu.

Fig. 9. Final Product, PDF File of the Syllabus in the Format Provided by the Center for Academic Development.

TABLE I. PRIORITY TASK – VALUE

Priority	Value
Low	1
Medium	2
High	3

TABLE II. TASKS TESTED

ID	Task	Pre Requisite (Id Task)	Priority	Expected Results	State	Department Head Review	Functional Testing
1	Application Login		1	Successful teacher access	Development - Integration	No	In Integration Test
2	Application Home		2	Recognition of assigned role. Load views associated with role	Development – Integration	No	In Integration Test
3	File upload Curriculum		3	Successful execution of SSIS packages. Correct information migration to BD	Development – Integration	No	In Integration Test
4	Load Academic Load Files		3	Successful execution of SSIS packages. Correct information migration to BD	Development – Integration	No	In Integration Test
5	Professor File Upload Screen		3	Successful execution of SSIS packages. Correct information migration to BD	Development – Integration	No	In Integration Test
6	Screen Load Bibliography Files		3	Successful execution of SSIS packages. Correct information migration to BD	Development – Integration	No	In Integration Test
7	Select Course Screen	3, 5	1	Course recognition by teacher. Association of information by course	Validated	Yes	Done
8	Display Syllable Screen	3, 4, 5, 6	3	Course information load. Creation of syllabus.	Validated	Yes	Done
9	Training Program Screens	3, 4, 5, 6	3	Course information load. Training Program Creation	Validated	Yes	Done
10	Enable Syllables	3	2	Display of entered syllabus. Change of states within the management process	Validated	Yes	Done
11	Display Syllable Screen	3	2	Display of entered syllabus. Search for past syllabus. Option to display the PDF document of the syllabus.	Validated	Yes	Done
12	Display Screen PDF Syllables	8, 9	3	Syllabus in PDF format	Development – Integration	No	Done

IV. CONCLUSION

This work presents a tool as a syllabus support which allows an improvement in the quality of the contents by providing the tool with suggestions for professors in the various sections that compose it. For this purpose, we had the idea of using the rules and the basis provided by the standard ISO/IEC/IEEE 42010 “Systems and Software Engineering - Architecture Description”.

The evaluations carried out by the pilot show that the flow of syllabus development was intuitive and friendly. The various sections that make up the syllabus, which can be provided by the Center for Academic Development and that do not require professor intervention, are automatically resolved through the load. When you start the syllabus management flow, you immediately see the relationship between the syllabus and its preloaded information.

Likewise, the probability of error decreases because the professor has to implement a large percentage of options within the syllabus through selection; for example, the competencies of the graduation profile associated with the subject, which is expected information. That information requires to edit the criteria and participation; for example, the competences of the subject.

The relationship between the different sections of the syllabus, such as the Academic Identification and Formative Program was successful since it is verified that the data entered in the first one are displayed with coherence in the second one.

V. FUTURE WORKS

Activities related to business intelligence will be included in the tool for analytical processing, text mining and predictive analysis of syllabus content. Topics of the courses and teaching-learning strategies will be guided by an intelligent assistant for avoiding inconsistencies when it is filled by professors. The tool will suggest curricula recommendations from technical organizations when defining subjects of the course.

Finally, textual and graphical reports will be generated to show syllabic compliance during this execution.

ACKNOWLEDGMENTS

This research has been funded by the Universidad Católica de Santa María, through the project “Syllabus management through business intelligence techniques for support in decision making at the FCIFF-UCSM”.

REFERENCES

- [1] H. S. Chung and J. M. Kim, "Semantic model of syllabus and learning ontology for intelligent learning system," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2014.
- [2] A. Lile and E. Muka, "University/Student Management System: a case study of Sports University of Tirana," *Proc. Multidiscip. Acad. Conf.*, 2018.
- [3] A. Sharma, M. Kumar, and S. Agarwal, "A Complete Survey on Software Architectural Styles and Patterns," in *Procedia Computer Science*, 2015.
- [4] M. Tapia-Leon, M. L. Merchán-Gavilanez, A. Carrera-Rivera, and S. Luján-Mora, "Comparative Study Of Syllabus Structure In Latin American Universities," in *ICERI2017 Proceedings*, 2017.
- [5] K. E. Harper and J. Zheng, "Exploring Software Architecture Context," in *Proceedings - 12th Working IEEE/IFIP Conference on Software Architecture, WICSA 2015*, 2015.
- [6] O. Gerbé, J. Raynauld, and R. Camarero, "Learning outcomes in a model-based approach to curriculum design," in *5th International Conference on Internet and Web Applications and Services, ICIW 2010*, 2010, pp. 550–553.
- [7] S. Chatvichienchai, "An XML-based syllabus repository system for inter-university credit exchange systems," in *ICCTD 2009 - 2009 International Conference on Computer Technology and Development*, 2009.
- [8] T. Kawaba, T. Tsuchiya, and K. Koyanagi, "Research on inter-domain collaborative syllabus services with cross-retrieval - IEEE Conference Publication," 2012. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6268513>. [Accessed: 30-May-2020].
- [9] T. Sekiya, Y. Matsuda, and K. Yamaguchi, "A web-based curriculum engineering tool for investigating syllabi in topic space of standard computer science curricula," in *Proceedings - Frontiers in Education Conference, FIE*, 2017.
- [10] H. K. Bhargava, D. J. Power, and D. Sun, "Progress in Web-based decision support technologies," *Decis. Support Syst.*, 2007.
- [11] G. R. Morales and J. P. Benedí, "Towards a reference software architecture for improving the accessibility and usability of open course ware," in *ACM International Conference Proceeding Series*, 2017.
- [12] N. Dimakis, L. Polymenakos, and J. Soldatos, "Enhancing learning experiences through context-aware collaborative services: Software architecture and prototype system," in *Proceedings - Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, WMUTE 2006*, 2006.
- [13] D. C. Force and J. Zhang, "Knowledge discovery from within: An examination of records management and electronic records management syllabi," *Rec. Manag. J.*, 2016.
- [14] R. Yucel, "Designing Knowledge Management Syllabus: How Business Administration Students Learn at Summer School?," in *Proceedings of the 9TH International Conference on Intellectual Capital, Knowledge Management & Organisational Learning*, 2012.
- [15] A. A. C. Júnior, S. Misra, and M. S. Soares, "ArchCaMO - A Maturity Model for Software Architecture Description Based on ISO/IEC/IEEE 42010:2011," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019.
- [16] M. Vidoni and A. Vecchietti, "Towards a reference architecture for advanced planning systems," in *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems*, 2016.
- [17] M. Chaabane, I. B. Rodriguez, and M. Jmaiel, "System of systems software architecture description using the ISO/IEC/IEEE 42010 standard," in *Proceedings of the ACM Symposium on Applied Computing*, 2017.
- [18] M. Chaabane, I. Bouassida Rodriguez, R. Colomo-Palacios, W. Gaaloul, and M. Jmaiel, "A modeling approach for Systems-of-Systems by adapting ISO/IEC/IEEE 42010 Standard evaluated by Goal-Question-Metric," *Sci. Comput. Program.*, 2019.
- [19] International Organization of Standardization, "ISO/IEC/IEEE 42010:2011 - Systems and software engineering -- Architecture description," ISO/IEC/IEEE 420102011E Revis. ISOIEC 420102007 IEEE Std 14712000, 2011.
- [20] M. Chaabane, F. Krichen, I. B. Rodriguez, and M. Jmaiel, "Monitoring of service-oriented applications for the reconstruction of interactions models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.