XML-BASED COURSE SYLLABI
An Electronic Implementation of the CDIO Syllabus

Ricardo Camarero, Clément Fortin, Gregory Huet
Département de Génie mécanique, Ecole Polytechnique de Montréal, Montréal, Canada

Jacques Raynauld, Olivier Gerbé
Chair for Teaching and Learning in Management Education, HEC, Montréal, Canada

Keywords: XML, Syllabus, CDIO, Open courseware, LMS, Learning objectives.

Abstract: A syllabus forms the structure of a course (or program) offering and is used to assemble its constituents. Course websites and LMS's, also contain similar information, and in some sense, are quite similar to course outlines or syllabi, and replicate some of the information that can also be found in administrative websites of universities. From a system perspective, this duplication and dispersion of information can be a source of confusion and hinder the set-up of an integrated workflow approach to manage all the course information for visualisation or reporting purposes.
Course outlines or syllabi, whatever the support used for distribution, are essentially "paper" documents that are most often created using word processors or limited web forms in course management systems. Currently, there is no accepted standard format for representing syllabi that could facilitate automatic production or processing of their contents.
In this paper, we present a model for a specialized tool to create, edit and publish course syllabi that can be used alongside LMS's and other administrative environments. The proposed prototype is based on XML to semantically tag the set of detailed elements of the CDIO Syllabus towards a complete and consistent implementation of an electronic syllabus. The objective is to cast the programs and course outlines into a schema, including both contents and learning outcomes, that allows the incorporation of the formulation and mechanisms for verification that the goals are fulfilled.

1 INTRODUCTION

Syllabi are a key element of university teaching. In most cases, syllabi take the form of a paper or electronic document where the key aspects of the course are described: instructor information, course material, assessment and grading, course calendar, etc. In other cases, syllabi are embedded in a course web site and can include numerous electronic resources. For students, syllabi are the most important source of course information. For university faculty and administrators, syllabi are at the center of pedagological course design and program offering.

Although universities usually provide course descriptions in online catalogs, detailed syllabi are usually not available in a centralized location (MIT OpenCourseWare is a nice exception): depending on the university, they can be found in departmental or faculty personal web pages or inside protected Learning Management Systems (LMS). When syllabi are posted in different locations, nothing guarantees that the content will be the same everywhere. When course syllabi are embedded in a course web site, the same information is often available in the HTML pages as well as in a PDF file, leading to unnecessary work for the instructor or the administrative staff. Furthermore, updates are often not synchronized and can mislead the students.

Within a department or a faculty, the course web site interfaces can be very different, adding unnecessary browsing difficulties for users in search of a very specific piece of information. The course web sites are most often designed and maintained using different software tools, adding difficulties when staff or instructors are assigned to different courses or departments. Finally, in many cases, no procedure insures

Camarero R., Fortin C., Huet G., Raynauld J. and Gerbé O. (2009).
XML-BASED COURSE SYLLABI - An Electronic Implementation of the CDIO Syllabus.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 160-165
DOI: 10.5220/0001848401600165

that all the course web sites are properly archived.

Either in plain web pages or embedded in a CMS, course syllabi of a faculty or a department are structured quite differently, which unnecessarily complicates the reading tasks of students. Although universities provide templates for paper course syllabi, these guidelines are not applied for the design of web based syllabi which are quite different and probably need specific interfaces more appropriate for electronic browsing. As a result, users are exposed to quite different web interfaces and some key elements like assignments due dates are displayed in different locations, not necessarily optimal from a usability perspective.

Finally, the information contained in a LMS or in a course website is often duplicated in parallel administrative systems. For reporting or administrative purposes, many universities seek specific course information like the objectives of the course, the evaluation method (assignments, exams, etc) and broad descriptions of topics covered. This is particularly the case for professional accreditation purposes, which require the formulation of learning outcomes and specification of mechanisms for verification that these goals are fulfilled. As these parallel administrative systems, when they exist, are not generally linked to course syllabi, special reports seeking precise elements of teaching and learning are not possible.

In this paper, we present a framework designed to provide a solution to these important questions, combining Open Syllabus (Gerbé and Raynauld, 2005) and the CDIO Syllabus (Crawley E.F., 2007). Open Syllabus is an electronic environment using a XML model-based approach, which aims to replace both simple traditional syllabi and more elaborate web syllabi by providing a structured approach using templates, reserved vocabulary, unified web user interface, centralized location and integration with administrative systems. The CDIO Syllabus is a comprehensive codification of knowledge, skills and attitudes which constitutes a requirements document for contemporary engineering education.

Building on the work carried out at HEC Montréal on `Zone Cours` (http://zonecours.hec.ca) and at École Polytchnique on the CDIO implementation, a methodology to structure and assemble the various elements required in an electronic syllabus or course web site is investigated.

2 THE SYLLABUS

2.1 Organization of a Course Outline

Due to their importance as the cornerstone of teaching and learning, universities provide guidelines, suggestions and even templates on the proper use of course syllabi. Numerous researchers have confirmed the central role played by syllabi. For (Parkes and Harris, 2002), syllabi serve three major roles: the syllabus as a contract, the syllabus as a permanent record and the syllabus as a learning tool, and provide a very detailed example of a course syllabus with 12 specific elements or sections. In a recent paper, (Marcis and Carr, 2004) report on a student survey about the relative importance of 23 distinct elements usually present in a course outline. Table 1 illustrates a simple mapping of the purposes of a syllabus and its associated elements.

Table 1: Mapping of the various syllabus elements into its purposes.

1. **The syllabus as a learning tool**
   - 1.1. Title and authors of textbooks and readings
   - 1.2. Course goals and objectives
   - 1.3. Course format (for example, lecture, discussion, videos, classroom activities)
   - 1.4. Attendance policy
   - 1.5. Late assignment policy
   - 1.6. Academic dishonesty policy
   - 1.7. Class participation requirements
   - 1.8. Examination and quiz dates
   - 1.9. Schedule of topics to be covered
   - 1.10. Holidays observed
   - 1.11. Amount of work (for example, amount of reading, number and length of other assignments)
2. **The syllabus as a contract**
   - 2.1. Grading procedure and policies
   - 2.2. Number of examinations and quizzes
   - 2.3. Kind of assignments (for example, readings, papers, presentations, projects)
   - 2.4. Reading material covered by each examination or quiz
   - 2.5. Type of examinations and quizzes (for example, multiple choice, essay)
3. **The syllabus as an administrative record**
   - 3.1. Instructor information (for example, name, title, office, location, phone number, e-mail address)
   - 3.2. Instructors office hours
   - 3.3. Course information (for example, course number and title, section number, credit hours)
   - 3.4. Course description
   - 3.5. Days, hours, and location of class meetings
   - 3.6. Required prerequisite coursework necessary to enroll in the course

These papers as well as the analysis of numerous course outlines (including those of the MIT Open-

CourseWare initiative) show that it is quite feasible to propose a model or a template that would incorporate most if not all the basic constituents of a course syllabus. The wording of the different elements could change across universities but, as the tables make it clear, the meaning of the different elements can be made quite precise. One can imagine the construction of a set of automatic rules that could help to translate syllabi of different universities.

3 THE CDIO INITIATIVE

3.1 The Syllabus

Four leading engineering universities[1] have proposed a new engineering education model, named CDIO (D. R. Brodeur and Ostlund, 2002). The CDIO Initiative seeks to re-emphasized the role of actual engineering practice in balance with the current engineering science model. This aims at training expert engineers who master the technical fundamentals of their field while acquiring the skills required to function in industrial enterprises, developing complex value-added engineering systems in modern team-based, multidisciplinary environments.

The learning activities are tightly integrated with the engineering process of product development around four phases: Conceiving-Designing-Implementing-Operating.

In previous papers, Ed Crawley and his colleages have developed and codified a comprehensive understanding of abilities needed by the contemporary engineer (Crawley, 2001). Known as the CDIO Syllabus, this codification is the set of knowledge, skills and attitudes expected from a graduating engineer. Initially designed as a requirements document for designing and implementing undergraduate engineering programs, it turned out to closely correlate with the criteria for accrediting engineering programs (ABET, 2000). It is consists in four high level requirements, which can be further detailled into second, third etc... levels objectives. This hierarchical breakdown makes the transition from the high level goals, to the level of course units or lectures. The depth or degree of granularity is the matter of specific curriculum for each field.

For conciseness and to maintain a generic presentation, a condensed form of the CDIO Syllabus with

[1]Chalmers University of Technology, Linköping University, and the Royal Institute of Technology, in Sweden, and the Massachusetts Institute of Technology in the USA, and which have been joined by several other engineering schools throughout the world.

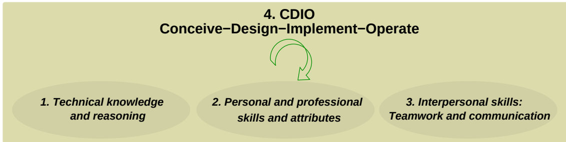

Figure 1: Structure of the CDIO Syllabus.

the 1st and 2nd levels is shown in Table 2. (Bankel J., 2005)

Table 2: Condensed Presentation of First Two Levels of the CDIO Syllabus.

1. TECHNICAL KNOWLEDGE AND REASONING
1.1. KNOWLEDGE OF UNDERLYING SCIENCES
1.2. CORE ENGINEERING FUNDAMENTAL KNOWLEDGE
1.3. ADVANCED ENGINEERING FUNDAMENTAL KNOWLEDGE
2. PERSONAL AND PROFESSIONAL SKILLS AND ATTRIBUTES
2.1. ENGINEERING REASONING AND PROBLEM SOLVING
2.2. EXPERIMENTATION AND KNOWLEDGE DISCOVERY
2.3. SYSTEM THINKING
2.4. PERSONAL SKILLS AND ATTRIBUTES
2.5. PROFESSIONAL SKILLS AND ATTITUDES
3. INTERPERSONAL SKILLS: TEAMWORK AND COMMUNICATION
3.1. TEAMWORK
3.2. COMMUNICATIONS
3.3. COMMUNICATION IN FOREIGN LANGUAGES
4. CONCEIVING, DESIGNING, IMPLEMENTING, AND OPERATING SYSTEMS IN THE ENTERPRISE AND SOCIETAL CONTEXT
4.1. EXTERNAL AND SOCIETAL CONTEXT
4.2. ENTERPRISE AND BUSINESS CONTEXT
4.3. CONCEIVING AND ENGINEERING SYSTEMS
4.4. DESIGNING
4.5. IMPLEMENTING
4.6. OPERATING

3.2 Specification of Learning Outcomes

In addition to the Syllabus, where the contents are outlined, the CDIO initiative introduces the goals or learning outcomes for knowledge and skills in a systematic way. To deal with these aspects, (Bankel J., 2005) have proposed the Introduce-Teach-Utilize (ITU) concept.

Introduce(I) : topics that are introduced in the course, but not subject to examination.

Teach(T) : knowledge and skills that are taught in the course and also subject to examination.

Utilize(U) : knowledge and skills that have been acquired in previous courses and are considered to be prerequisites.

Table 3 illustrates a generic application of this concept for a course. A more thorough presentation is given

Table 3: Example of a Course Level ITU-matrix.

	SYLLABUS LEVELS											
COURSE CONTENT		**1** *1.x.x*			**2** *2.x.x*			**3** *3.x.x*			**4** *4.x.x*	
Topic 1	T			I						U		
Topic 2						T				U		
Topic 3		I										
Topic			T			U						

in (Gunnarsson S., 2007) including courses as well as entire curriculum applications.

4 THE OPEN SYLLABUS MODEL

In the spirit of (Tungare and al, 2006), who advocate the use of standardized representations, (Gerbé and Raynauld, 2009) have proposed a model, illustrated in Figure 2, that could accommodate numerous types of syllabi. The nodes are defined as follows:

- CourseOutline corresponds to the syllabus or the course outline.
- CourseStructure organizes CourseUnits.
- CourseUnit is a section of a course outline. Instructor information, Lectures calendar and General course information are examples of CourseUnit elements.
- CourseUnitStructure is a sub-section of a section. CourseUnitStructure organizes the information of a CourseUnit. For example, the lecture could be devided in two parts.
- CourseUnitContent is the key pedagogical container of all the learning resources. CourseUnitContent includes the readings, the files and the assignments typically found in a lecture.
- CourseProxyResource is a proxy that serves to qualify the use of a particular resource in a CourseUnitContent. For example, a reading could be mandatory in a course but optional in another.
- Ressource is a very low level element that corresponds to some specific learning material; for example PowerPoint files, citations, urls, text files, assignment guidelines.....

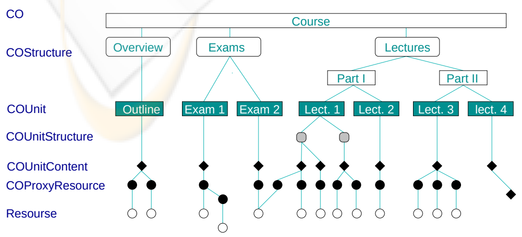

Figure 2: Structure of the Proposed Course Outline Model, adapted from (Gerbé and Raynauld, 2009).

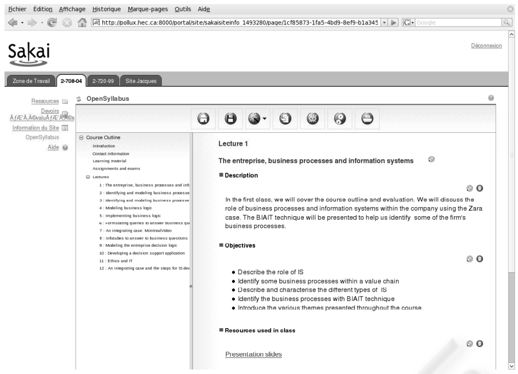

Figure 3: Page for Lecture 1.

Figure 3 illustrates the model in the context of a simple course outline created with OpenSyllabus Sakai 2.5 tool based on GoogleWebToolKit(GWT). The tree on the left displays the CoStructure elements (Introduction, Contact information, Learning material, Assignment and Exams and Lectures). There are CourseUnit elements represented by 12 different lectures. The resources in the Lecture 1 page on the right hand side panel are part of a CoUnitContent. The description is a text resource with the rubric Description put on the CoproxyResource. The objectives are also a text resource with the rubric Objectives put on the CoproxyResource. The last resource is a Power Point file with the title Presentation slides and the rubric Resources used in class put in the CoproxyResource.

The XML file corresponding to the course outline of Figure 3 is given in Figure 4. As explained before, the CoUnitContent contains a text Resource with associated CoProxyResource set to CoContentRubric Rubric=Description and some other useful properties like the security which is set to public. CDATA contains the text. This XML description is the building block for the OpenSyllabus GWT editor and can lead to very sophisticated operations for reused or display.

5 COURSE INFORMATION MANAGEMENT

5.1 LMS-XML Integration

To our knowledge, most Learning management Systems such as Moodle, do not currently offer tools or plug-ins to create and edit XML-based course outlines. Although, it quite easy to construct and display a course website that will mimic the vocabulary, the presentation and the content of Figure 3, the underlying structure cannot be manipulated since it is not

```
<COUnitContent type='lecture' scrty='public'>
<COResourceProxy type='text' scrty='public'>
<COContentRubric type='description'/>
<properties>
<visible>true</visible>

</properties>
<COResource type='text' scrty='public'>
<properties>
<text>
<![CDATA[ <P>In the first class, we will cover the
course outline and evaluation. We will discuss the
role of business processes and information systems
within the company using the Zara case. The BIAIT
technique will be presented to help us identify
some of the firm's business processes.
</P>]]>

</text>
</properties>
</COResource>
</COResourceProxy>
...
</COUnitContent>
```

Figure 4: XML code fragment for the production of the Page of Figure 3.

model-based. LMS's cannot recognize that the content underneath the title:`Objective` in that figure are indeed the learning objectives of Lecture 1.

In some preliminary work, we have explored the possibility of linking XML-based syllabi to Moodle. Figure 5 illustrates the results of a plug-in that can read an OpenSyllabus XML file similar to the one presented here. The XML file imported for that example included the description and the objectives of the course as well as the title of the different lectures and their content.

With the current prototype, the instructor can edit the resulting content but the changes will not be added to the underlying XML file. Further work is necessary to enable both import and edit/export XML capabilities in Moodle. However, the prototype opens interesting avenues concerning the integration of standardized course related content in LMS's. For example, one can imagine that official description of courses, learning objectives and university policies can be pushed to course websites in a similar fashion. It is also possible to develop and use simple XML editors (for example the GWT OpenSyllabus editor) to generate the XML file to be imported in th LMS for final use by the instructor and the students.

5.2 Reporting: Accreditation

For professional accreditation purposes, an essential fonction is to measure the program outcomes which

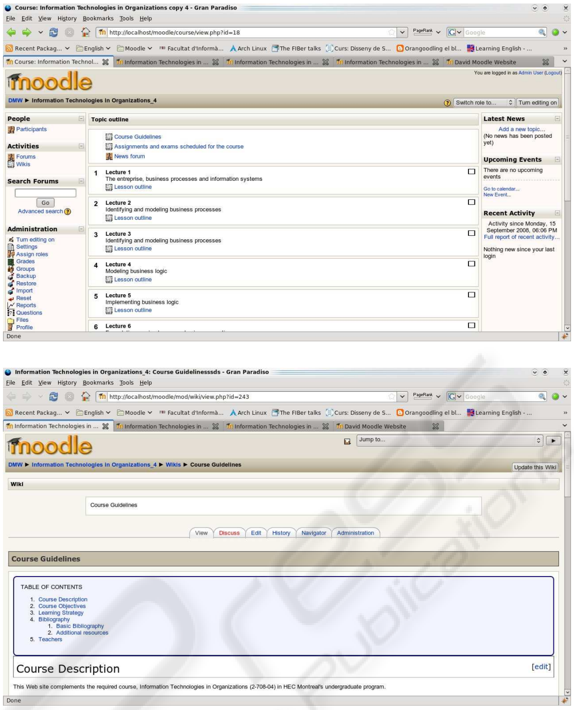

Figure 5: Moodle screen capture resulting from the import of a Open Syllabus XML file.

is based on the capability to compare and assess the quality of the training given at different universities. Besides gathering specific and precise information such as the number of hours given to some particular subject matter, the ability to extract the formulation of learning objectives and outcomes of individual courses as well as of entire education programs is now becoming increasingly important. In addition, this would assist instructors in mapping course design to measurable outcomes, thus helping the process of institutional program offering. This requires to complete work on reporting standards currently in progress (Course Description Metada).

5.3 A Course Information System

A syllabus can be viewed as a collection of metadata for a course, which leads to an integrated workflow approach to manage all the course information. The development of Open Syllabus as a specialized Syllabus Editor to help create and edit syllabi will encourage adoption of such schema as the CDIO or similar codifications adapted to other fields.

Of strategic importance is the integration of a syllabus editor into course management systems so that the use of such specialized tools will be seamless from the instructor's point of view. Furthermore, linking these parallel administrative systems to course syllabi as shown in Figure 6, could aleviate repetitive work

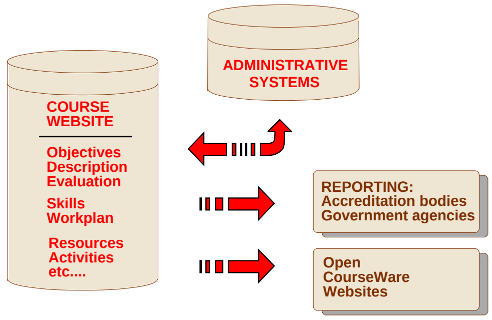

Figure 6: Overall System Architecture.

for both faculty and administrative staff, as well as avoiding many sources of errors resulting from the lack of synchronization of updates.

This integration of administrative and pedagogical systems will result in:

- faster and more accurate content search;
- enhanced sharing of syllabi and learning resources by providing export/import possibilities;
- provide a completely open gateway to the public content of all course syllabi in the spirit of the MIT OpenCourseWare project.

6 DEVELOPMENT OBJECTIVES

The lack of standardization of a semantic description of syllabi has resulted in a wide variety of formats with serious drawbacks. This paper has sketched the development of an electronic syllabus based on a XML model which semantically tags the set of detailed elements of the CDIO Syllabus. This approach is quite flexible and each university could easily parametrize the various elements of the model as well as the vocabulary used, while keeping a semantic suitable for sharing.

The authors believe that, for OpenSyllabus to have its intended impact, it needs to be integrated into the current pedagogical practices of educators and administrators. In particular, the adoption of a model-based course outline approach requires some integration with the course management systems used in different universities. In some cases, a generic GWT OpenSyllabus tool or some kind of advanced XML editor can be used directly by instructors and support staff to create, edit and publish structured syllabi. In other cases, taking advantage of its light or minimal web services architecture, the GWT OpenSyllabus client can be linked much more closely to a course management system as it is the case for an alpha prototype version of OpenSyllabus/Sakai. In other cases, the course management system itself could provide XML editing capabilities or could be modified to handle XML tagging. Whichever path chosen, an XML based approach to course outline like Open Syllabus could foster a much better integration of pedagogical information with administrative systems, providing a cost effective solution to the increasing reporting requirements from international agreements or accreditation agencies.

It can be envisionned that the availability of electronic syllabi, using an XML-based approach will make the content of a course much more reusable and open up possibilities for publication in a wide range of formats.

REFERENCES

ABET (2000). *Criteria for Accrediting Engineering Programs: Effective for Evaluations During the 2000-2001 Accreditation Cycle, Revised March 2000*. Accreditation Board of Engineering and Technology.

Bankel J., Berggren K.F., C. E. E. M. E. G. K. Ö. S. S. D. W. I. (2005). Benchmarking engineering curricula with the cdio syllabus. *International Journal of Engineering Education, Vol. 21, No. 1. Also at http://www.cdio.org*.

Crawley, E. F. (2001). *The CDIO Syllabus: a statement of goals for engineering education*. Department of Aeronautics and Astronautics, MIT, USA.

Crawley E.F., Malmqvist J., O. S. B. D. (2007). *Rethinking Engineering Education: The CDIO Approach*. Springer, New York.

D. R. Brodeur, E. F. Crawley, I. I. J. M. and Ostlund, S. (2002). International collaboration in the reform of engineering education. In *Proc. 2002 ASEE Conference and Exposition, Montreal, Canada, June 2002*. www.asae.org/conferences/proceedings/search.cfm.

Gerbé, O. and Raynauld, J. (2005). A management system for model-oriented course outlines. In *Ed-Media 2005 World Conference on Educational Multimedia, Hypermedia & Telecommunications*. ED. Piet Kommers et Griff Richards.

Gerbé, O. and Raynauld, J. (2009). *The OpenSyllabus Model*. Mati Montréal Technical report, http://www.matimtl.ca.

Gunnarsson S., Wiklund I., S. T. K. A. G. S. (2007). Large scale use of the cdio syllabus in formulation of program and course goals. In *Proceedings of the 3rd International CDIO Conference, MIT, Cambridge, Massachusetts, June 11-14, 2007*. http://www.cdio.org.

Marcis, J. G. and Carr, D. R. (2004). The course syllabus in the principles of economics: A national survey. *Atlantic Economic Journal, 32, 259*.

Parkes, J. and Harris, M. B. (2002). The purposes of a syllabus. *College Teaching, 50, 55-61*.

Tungare, M. and al (2006). Towards a standardized representation of syllabi to facilitate sharing and personalization of digital library content. In *International Workshop on Applications of Semantic Web technologies for E-Learning*.