

An Open Syllabus Model

Olivier Gerbé and Jacques Raynauld
HEC Montréal – MATI
Québec, Canada
olivier.gerbe@hec.ca, jacques.raynauld@hec.ca

Abstract: Syllabi are universal gateways to teaching and learning in universities. Building on this metaphor, this paper presents Open Syllabus, a model-based XML approach to Web-based course outlines. Open Syllabus organizes all the resources (files, citations, assignments, etc.), activities (quiz, forum, etc.) and course information (description, objectives, etc.) in a format providing guidance to the instructors, navigation clarity for the students and export/import facilities for sharing content across platforms. Open Syllabus is currently being developed as a Sakai 2.6 tool.

Introduction

Syllabi or course outlines are often considered as the cornerstone of university teaching. In most cases, syllabi take the form of a paper or electronic document (Word or PDF) where key aspects of the course are detailed: instructor information, course material, assessment and grading, course calendar, etc. In other cases, syllabi are embedded in a course website and comprise numerous electronic resources (PowerPoint files, URLs, etc.). For students, syllabi are the most important source of course information. For university administrators, syllabi are at the center of program design and content covered.

Not surprisingly, university teaching and learning centers provide guidelines, suggestions and even templates on proper use of a paper course outline. For Parkes and Harris (2002), syllabi serve three major roles: they can be seen as a contract, as a permanent record and as a learning tool. They provide a simple mapping of the purposes of a syllabus and its associated elements. In a recent paper, Marcis, Carr et al. (2004) report on a student survey about the relative importance of 23 distinct elements usually present in a course outline. These papers are examples of a rich literature which indicates that it is quite easy to propose a model or a template that would incorporate most if not all the basic elements of a proper paper course syllabus.

Adopting a Web perspective, the question of a course outline model is not as straightforward. Although PDF versions of syllabi are often available directly in HTML pages or imbedded in a Course Management System (CMS), the access to electronic resources (Power Point files, URLs, etc.), assignments and activities is often decided by the instructor with little guidance. As a result, students are exposed to quite different Web interfaces and some key elements like assignments due dates are displayed in locations, not necessarily optimal from a usability perspective. Moreover, important information is regularly duplicated leading to possible errors and unnecessary work by instructors or staff. Finally, current Web syllabi cannot be shared easily across platforms. Web based syllabi are thus quite different from their paper counterpart and need more appropriate models, templates and interfaces to leverage the possibilities of the Web. To our knowledge, few papers have addressed this problem (see Tungare et al., 2006 for a clear advocacy of standardized syllabi). Open Syllabus offers a possible avenue to address this important question albeit from a model perspective.

In 2003 the Object Management Group (OMG) presented a new approach to developing information systems. It is called the Model Driven Architecture (MDA) (OMG, 2003) (Bézivin & al., 2001). According to the OMG, the MDA provides a scalable approach that is platform-independent to address the challenges linked to rapid changing world of business and technology. According to the standards already stipulated by OMG, the MDA allows users to separate the business or applicative logic from the underlying platforms.

Based on a model-driven approach initiated in Gerbe and Raynauld (2005), Open Syllabus aims to replace both simple traditional syllabi and more elaborate Web syllabi by providing a structured approach using templates that can be parameterized. As for paper syllabi templates, adopting a model for Web syllabi could provide guidance to the instructors and simplify the navigation by the students. Furthermore, a model approach could permit syllabi export/import across platforms and thus support the sharing objectives of Open Education

initiatives like OpenCourseware Consortium. Finally, model-based Web syllabi could greatly facilitate reporting duties by universities (for example, collecting all the course descriptions or objectives) and streamline links to university catalogs.

This paper is organized as follows: we first present an illustrative syllabus and its associated structure. We then explain the abstract structure of a syllabus suitable for computer implementation. We then discuss a detailed presentation of our proposed model. The last section provides a quick summary of our approach and discusses its current Sakai 2.6 implementation.

An Illustrative Syllabus

Parkes and Harris (2002) list twelve important elements that should be present in a paper course syllabus: instructor information, office hours, course material, general course information, course objectives, course procedure, assessment and grading, expectations and professionalism, other course policies, how to succeed in this course, additional resources and tentative course calendar. As an illustration, Figure 1 displays some of these key elements albeit adopting a Web perspective. Right after some statutory course information (semester, number and title), we have the overview or the description of the course. The Exams section describes the assessment approach of the course, lists the percentage weights and provides links to the necessary documentation or information. The important Lecture section lists the three lectures included in our illustrative syllabus. As it is often the case, the lectures are organized in two different parts. For each lecture, we find a short description and either a reference or a link to relevant documents (slides).

Winter 2009
53-753-02 Information Technologies and Process Re-engineering

Overview
 The introduction of new technologies in organisations requires to reconsider the organisational operating modes in order to determine how the new technologies can support the business.
 The objectives of this course are:

- To clarify the BPR concept;
- To study existing theories on BPR;

Exams
Team assignment (30%)
 The team assignment has to handed in at session 6 and consists of a thorough analysis of the following cases:

- Section A01: Pharmacy Service Improvement at CVS (A), HBS case #606015
http://harvardbusinessonline.hbsp.harvard.edu/b02/en/common/item_detail.html?id=606015

Final exam (70%)
 Open book.

Lectures
Information Technologies
Lecture 1 - From BPR to BPM
 Course objectives and expectations, evaluation methods
 BPR definition and concepts
 Michael Hammer's vision
 Business processes - what are they?

- [Slides](#)

Lecture 2 - Understanding and Analyzing the As-Is Process
 Rivard, S. & Talbot, J. Transforming Business Processes and Developing Information Systems
Read chapter 3

Process Re-engineering
Lecture 3 -Designing the To-Be Process
 This session focuses on the design of the new process.

...

Figure 1 : A syllabus (partial)

The Syllabus Structure

Across universities, syllabi may be organised in different ways. However, as indicated by Parkes and Harris (2002), a set of key elements are most often found. Figure 2 shows an example of a syllabus structure encompassing many syllabi and especially the one illustrated in Figure 1.

The syllabus for the winter 2009 6-610 course is represented by the top square and is organized in three parts (white squares): an overview of the course content, the exams list and the lectures list. Lectures have been sub-organized in two themes; Theme 1 and Theme 2, and the content of Lecture 1 (black square) has been itself divided in two parts: Part 1 and Part 2 (gray squares). Overview, exams or lectures contains resources (white

circles). The use of a resource is dependent of the place where the resource appears. The reading of a document may be mandatory in a course but optional in another. To represent such situations, we created contexts (black circles) for the resources. A context is the particular use of a resource. Contexts are grouped into containers (black diamonds). These containers group resources into reusable modules. A container may reference a reusable module as shown on the right end of Figure 2. To represent all possible syllabus structures for computer implementation, we developed an abstract structure model that we detailed in next section.

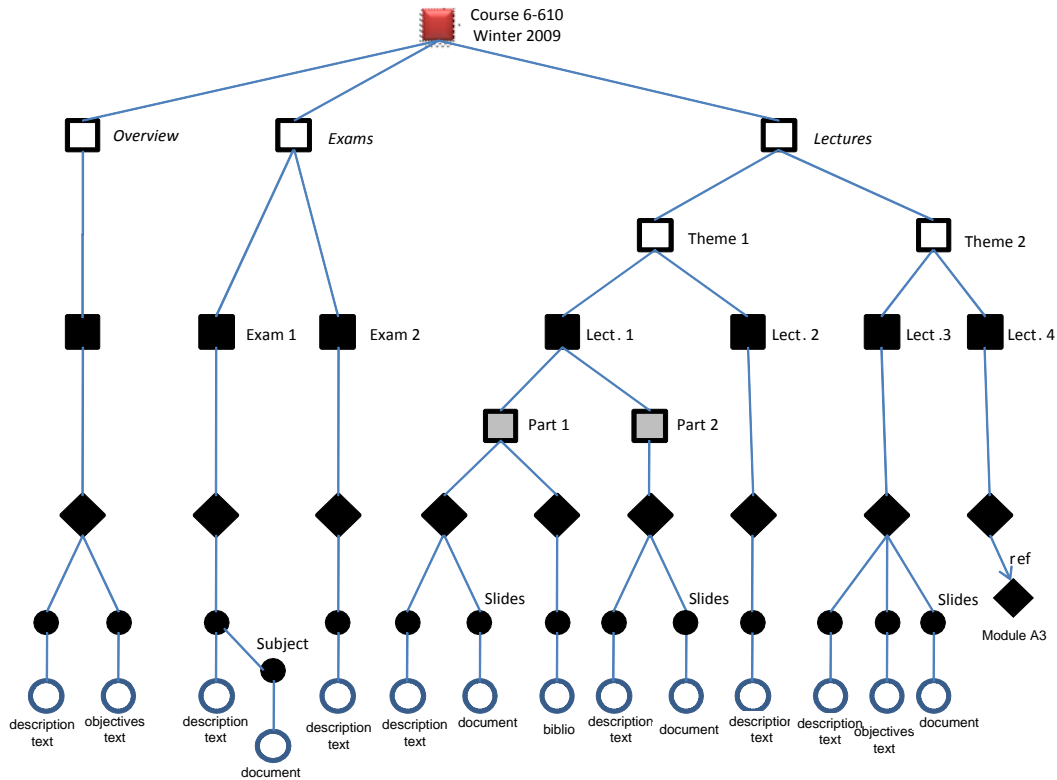


Figure 2: A syllabus structure example

The Abstract Structure Model

In the spirit of the Model Driven Architecture (OMG, 2003), this section presents the abstract structure model that may implement different syllabus organizations. Figure 3 illustrates the hierarchy of the model elements. *ASMOBJect* is an object of the structure. *ASMRoot* is the root of the structure and *ASMUnit* is the core of the structure. *ASMStructure* and *ASMUnitStructure* allow super and sub-organizations of *ASMUnits*. *ASMUnitContent* is a container for *ASMContext* of *ASMResources*. A same resource may be used in multiple units and *ASMContexts* capture the contextual characteristics.

ASMOBJect is characterized by three attributes: *id*, *audience* and *merge*. *Audience* controls the display of the element. For a syllabus, we propose the following hierarchical nomenclature:

- 0 – public
- 1 – University
- 2 – Faculty
- 3 – Department
- 4 – Attendees
- 5 – Course Team
- 6 – Owner.

This nomenclature could be extended by a code following the number that could be processed by an ad-hoc application. For example 3/a may represent a particular group “a” of the faculty.

ASMRoot is the top element of the structure and represents the course syllabus. Its attributes are: lang, type, template, date-modified and date-published.

ASMStructure is a structural element. These elements organize units. It is characterized by a label and a visibility element that allows the instructor to hide or show all the structure.

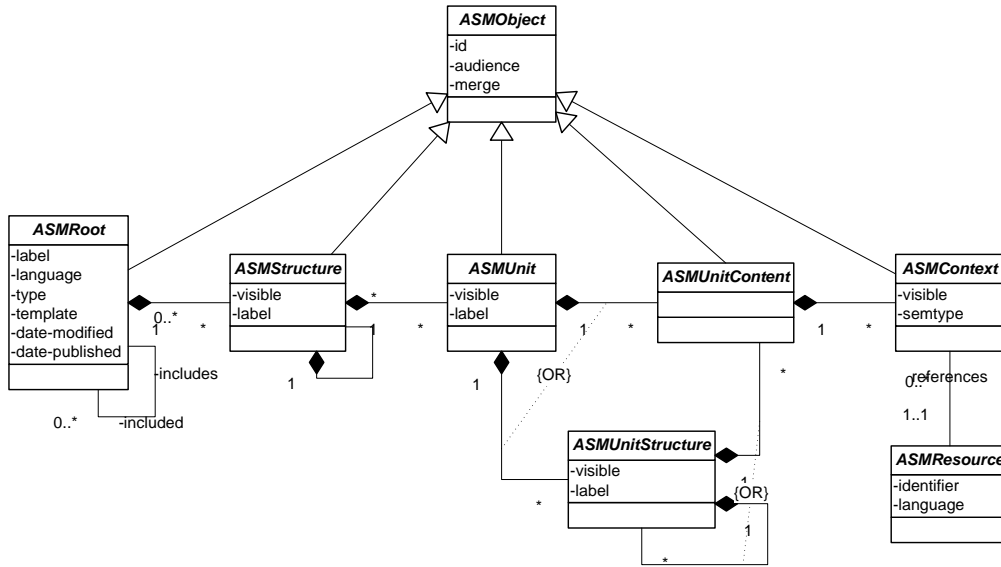


Figure 3: The Abstract Structure Model (UML Diagram).

ASMUnit is the most important element in the structure. As suggested by its name, it represents the semantical element of the structure. A syllabus is an assembly of units that are themselves an assembly of use of resources. Units are organized using *ASMStructure* and their content is organized using *ASMUnitStructure*. As we can have structure above and below, it is important to define clearly what a unit is. For example, a lecture is a unit for a course delivered in class, but a week is a unit for an on-line course.

ASMUnitStructure is a sub-structural element. These elements organize contents of a unit. Label and visible are the attributes.

ASMUnitContent is a container element. It groups together elements that have a common semantic. For example, an *ASMUnitContent* groups all the resources of a specific lecture. Contents may be reusable. They can be assembled into units. These containers may be moved easily from a lecture to another one. They have no attributes.

ASMContext is an element that represents the use of a resource in a unit. The abstract *ASMContext* has two attributes: visible and semtype. Semtype is the semantic type of the represented resource. Based on the semtype, the display engine organized resources into rubrics (description, objective, work, etc.). Depending on the represented resource a context may have specific attributes. In the next section, we provide some examples of specific attributes in the context of a proposed model.

ASMResource is the resource itself. Documents, hyperlinks, description and bibliographic references are examples of resources.

The Syllabus Structure Model

Figure 4 shows our syllabus structure model proposition. Based on an analysis of numerous syllabi from different universities, we propose a generic way to structure syllabus resources. This model is derived from the

abstract structure model. It is extensible. Following Parkes and Harris (2002), this model has the five components generally found in a syllabus:

- Staff information
- Course overview
- Assessment
- Learning material
- Lectures list.

Staff information

The staff information component generally contains the names of the instructors, assistants, and all the persons involved in the delivery of the course. At the lowest level, a *StaffUnitContent* contains *PeopleContext* elements that represent a person involved in the course and have two specific attributes: the role played by the person and optionally its availability.

Course overview

The course overview component contains generally a description of the course. It may also present the course objectives, the pedagogical approach used and all other information that have no place in other components. At the lowest level, an *OverviewUnitContent* may include a general *ResourceContext*.

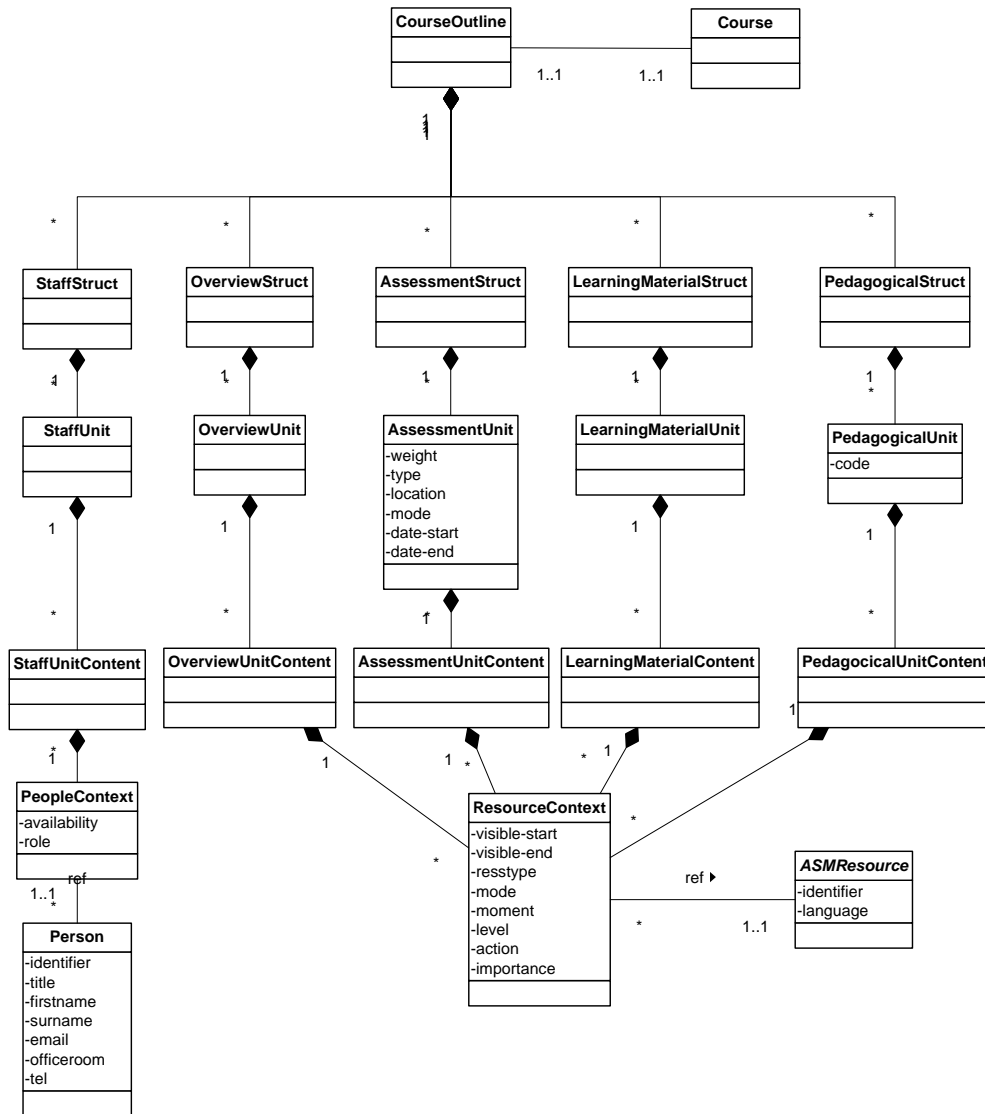


Figure 4: The syllabus structure model (UML Diagram).

Assessment

The assessment component is the list of the assessments (homework, examination, participation, or other type of assessment). An assessment is characterized by the attributes: weight (points or percentage), type (midterm exam, final exam, etc), location (home or in class), mode (individual or team) date-start and date-end of the assessment. An assessment may have resources, for example, the home work instructions. At the lowest level, an *AssessmentUnitContent* include *ResourceContext*.

Learning Material

The learning material component groups all types of resources needed during the course. They can be bibliography, tools, documents, URLs, past exams, and other resources. At the lowest level, a *LearningMaterialUnitContent* include *ResourceContext*.

Pedagogical Structure

The pedagogical structure component groups pedagogical units of the course. This component can contain lectures but also recitations or labs which can be referenced by their type. Each pedagogical unit may have a code (for example, L1 for lecture 1). At the lowest level, a *PedagogicalUnitContent* include *ResourceContext*.

The Resource Structure

This section presents the lowest part of the syllabus structure model. Figure 5 illustrates the resource structure. A *ResourceContext* represents the use of a resource in a specific *PedagogicalUnit*. We make a distinction between the following three elements: *BiblioContext* references a bibliographic reference; *InformationContext* references a text and *ReferenceContext* references all other kind of resources like hypertext managed by the system (*IntDocument*), exercises, documents or files, hyperlinks, quizzes, assignments and forums. We briefly detail some of these components with their attributes.

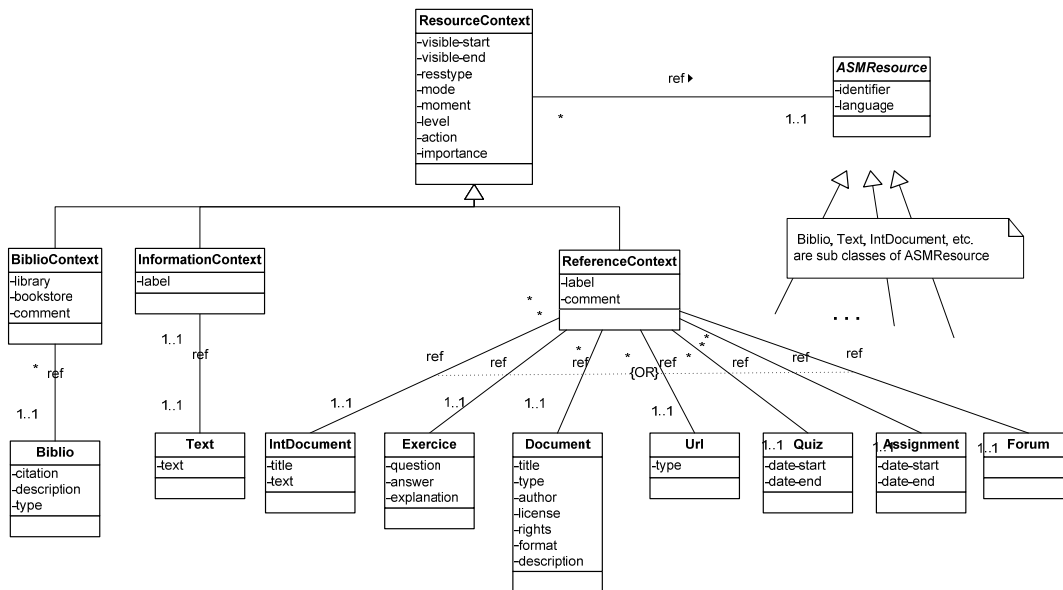


Figure 5: The resource structure (UML Diagram).

ResourceContext

A *ResourceContext* references a resource. All contexts (*BiblioContext*, *InformationContext* and *ReferenceContext*) have the following attributes:

- *visible-start* is the date when the resource becomes visible in the syllabus. The date may be absolute (2009/01/12 at 12:00) or relative (3 hours before the beginning of the lecture 2);
- *visible-end* is the date when the resource stops to be visible;
- *importance* is a flag to highlight the resource and show its importance (Don't forget your laptop!);
- *resstype* is the type of the referenced resource (biblio, information, document, etc.).

If students have an action to do with the resource, the following attributes applies:

- *mode* specifies if the action is individual or team based;
- *moment* specifies when the action takes place: before, during or after the pedagogical unit;
- *level* specifies if the action is mandatory, optional or targeted to learn more.
- *action* is the type of action: read, write (produce something) or execute (download a file).

BiblioContext / Biblio

BiblioContext is used to insert a bibliographic reference. We can specify if the reference is available or not at the library or at the bookstore and we can add specific comments about the reference. *Biblio* represents the reference. Specific information about this reference are the citation, a description and the type of the reference (book, article, review, journal, etc.).

InformationContext / Text

InformationContext introduces a text that gives information. A *Text* is not reusable (but an *IntDocument* is – see below) and is used locally. Description and objectives are described using *InformationContext* and *Text*.

ReferenceContext / IntDocument, Exercice, Document, URL, Quiz, ...

ReferenceContext references all other kinds of resources. The label and comment of a *ReferenceContext* are contextual. For an *IntDocument* (an hypertext) or an *Exercice*, the label is the title of the resource. For a document, URL, Quiz, Assignment and Forum, the label is used to display the hyperlink that gives access to the resource. Each particular type of resource has its own attributes. The proposed model is extensible and one can easily add a new resource type or specialise an existing one.

Conclusion

OpenSyllabus constitutes a model of syllabus organization supported by an XML representation (See Appendix for an XML example). Our abstract structure model is quite flexible and each university could easily parameterize the various elements of the model as well as the vocabulary used while keeping a semantic suitable for sharing across platforms. Improved Web syllabi organisation will benefit students, instructors and support staff. It could also become a nice tool for creating a unique open gateway to all the course syllabi of a university, an initiative advocated by the OpenCourseware consortium. OpenSyllabus is currently developed as a GWT/Sakai 2.6 tool so instructors and support staff will soon be able to create, edit, share and publish structured syllabi.

References

- Bézivin, J., & Gerbé, O. (2001). Towards a Precise Definition of the OMG/MDA Framework. *Conference on Automated Software Engineering, 2001*, IEEE Computer Society Press, San Diego, CA. 273–280.
- Gerbé, O., & Raynauld, J. (2005). A Management System for Model-Oriented Course Outlines, *World Conference on Educational Multimedia, Hypermedia & Telecommunications, 2005*, Montréal, Canada. 4601-4611. Available at http://www.editlib.org/index.cfm?fuseaction=Reader.ViewAbstract&paper_id=20804.
- Marcis, J., & Carr, D. (2004). The course syllabus in the principles of economics: A national survey. *Atlantic Economic Journal*, International Atlantic Economic Society, 32 (3), 259-259.
- OMG. (2003a). MDA Guide Version 1.0.1. June 2003. Document number: omg/2003-06-01.
- OpenCourseWare, Consortium, <http://www.ocwconsortium.org/>.
- Parkes, J., & Harris, M. B. (2002). The purposes of a syllabus. *College Teaching*, 50 (2), 55-61.
- Sakai Project, <http://bugs.sakaiproject.org/confluence/display/OSYL/OpenSyllabus+Home>.
- Tungare, M., Xiaoyan, Y., Perez-Quinones, M., Fox, A. E., Fan, W., & Cassel L. (2006), Towards a Standardized Representation of Syllabi to Facilitate Sharing and Personalization of Digital Library Content, *International Workshop on Applications of Semantic Web technologies for E-Learning, 2006*. Available at http://www.win.tue.nl/SW-EL/2006/camera-ready/14-tungare_2006_last.pdf.

Acknowledgements

The authors wish to thank the Open syllabus Team: Pascale Blanc, Robert Bolduc, Mathieu Cantin, Claude Coulombe, Laurent Danet, Mame Awa Diop, Robert Gérin-Lajoie, Tom Landry, Yvette Lapa Dessap, Sacha Leprêtre, Remi Saïas, Vincent Siveton and Emmanuel Vigne for their collaboration and the fruitful discussions we had. We would like to address a special thanks to Paul Mireault who challenged each element of this proposition. All remaining errors are our own.


```

</COUnit>
<!-- ===== -->
<COUnit type='evaluation' audience='0'>
  <properties>
    <label><![CDATA[Final exam]]></label>
    <weight>70</weight>
    <type>end-term</type>
    <location>inclass</location>
    <mode>individual</mode>
  </properties>
  <COUnitContent type='evaluation' audience='0'>
    <!-- ===== -->
    <COResourceContext type='information' audience='0'>
      <properties>
        <semtype>information</semtype>
        <visible>true</visible>
      </properties>
      <COResource type='text' audience='attendee'>
        <properties>
          <content>
            <![CDATA[Open book.]]>
          </content>
        </properties>
      </COResource>
    </COResourceContext>
  </COUnitContent>
</COUnit>
</COStructure>
<!-- ++++++ LECTURES ++++++ -->
<!-- ++++++ LECTURES ++++++ -->
<!-- ++++++ LECTURES ++++++ -->
<COStructure type='lectures' audience='0'>
  <COStructure type='lectures' audience='0'>
    <properties>
      <label><![CDATA[Information Technologies]]></label>
    </properties>
    <COUnit type='lecture' audience='0'>
      <properties>
        <code>L1</code>
        <label><![CDATA[From BPR to BPM]]></label>
      </properties>
      <COUnitContent type='lecture' audience='0'>
        <!-- ===== -->
        <COResourceContext type='information' audience='0'>
          <properties>
            <semtype>description</semtype>
            <visible>true</visible>
          </properties>
          <COResource type='text' audience='0'>
            <properties>
              <content>
                <![CDATA[Course objectives and expectations, evaluation methods
                  BPR definition and concepts
                  Michael Hammer's vision
                  Business processes - what are they?]]>
              </content>
            </properties>
          </COResource>
        </COResourceContext>
      </COUnitContent>
    </COUnit>
  </COStructure>
  .
  .
  .
</COStructure>
</CO>

```