AC 2008-2935: UNISYLLABUS: A TOOL TO MANAGE YOUR PROGRAM'S SYLLABI

Othoniel Rodriguez-Jimenez, Polytechnic University of Puerto Rico

Othoniel Rodriguez-Jimenez is Associate Director for the Computer Engineering program at Polytechnic University of Puerto Rico. He holds a PhD in Computer Engineering and Computer Science from the University of Missouri-Columbia. His main research areas are eLearning, computers in education, and reconfigurable hardware.

Carlos Pacheco, Polytechnic University of Puerto Rico

Nelson Pacheco graduated magna cum laude from the Computer Engineering Program at Polytechnic University of Puerto Rico. He is currently a Senior Engineering Consultant with Abacus,Inc.

Nelson Reyes-Aviles, Polytechnic University of Puerto Rico

Nelson is a graduate from the Computer Engineering Program at Polytechnic University of Puerto Rico. He is now the IT Manager for the Puerto Rico National Guard in San Juan PR.

Marisol Mercado, Polytechnic University of Puerto Rico

Marisol graduated magna cum laude from the Computer Science program at Interamerican University of Puerto Rico. She is currently a masters' student at Polytechnic University of Puerto Rico.

Unisyllabus: A Tool to Manage Your Program's Syllabi

Abstract

The course syllabus is a tool for teaching and a kind of contract with the students and the accreditation bodies. Our experience with accreditations at the institutional or program level, by national, regional, and state accrediting bodies indicates that a common cause for findings/concerns by these agencies is the syllabus. Different accreditation agencies will require your program's syllabi in different formats, making it critical to keep these diverse formats in synch with each other. Also important is that information on your syllabi and the school's catalog and website is in synch. The syllabus could also support your ABET accreditation goals by being an ideal place to state the skills students are expected to acquire. These skills are phrased in terms of performance criteria for different outcomes allowing faculty to clearly identify what is expected from them in terms of outcomes assessment within a specific course. From the syllabi for the whole program one can extract lists of textbooks for the bookstore, or lists of bibliographic references for adquisition by the library, or the short course descriptions for the catalog. All these issues point to the need for a flexible tool to support the creation, editing, maintenance, review, and publication of a program's syllabi in a uniform way.

Unisyllabus is a tool originally developed as a Capstone project which incorporates all the above features and some more. It is a web application which allows the capture of all the information contained in the syllabus formats used by the Accreditation Board for Engineering and Technology (ABET), and the Middle-States Association of Colleges and Schools (MSACS). The application uses secure, role-based access control for users, who are assigned a user-name, password and role, and allowed to log-in using a web browser. Roles include Viewer, Editor, and Publisher. The information captured in the process of editing a syllabus is stored in a relational database. A standard report writer is used to produce a syllabus for a particular course in the format required by a particular accrediting agency, and in specific document formats such as .doc or .pdf. Additional reports are defined and new ones can be easily introduced. Because a large part of the information required by the various accrediting agencies is common, this process is highly efficient. It reuses editable reference tables and avoids the confusion resulting from separately maintained documents. Other features are related to simplifying the assessment tasks by associating outcomes and their performance criteria to describe the skills that should be learned in the course. The application supports a practically unlimited number of departments, programs, course codes and syllabi, and new reports on the stored data can be easily introduced. Finally our experience with the use of Unisyllabus will be summarized.

Introduction

Unisyllabus is a web application that grew out of a need for a way to handle all course syllabi in a uniform and flexible manner. When one considers the practical uses of the course syllabus one is confronted with a variety of alternative uses which are not readily met through reliance on a single document file. Perhaps the most insidious is identifying the latest valid version of the syllabus file, particularly if there are more than one professor teaching a particular course. In addition, documents such as your school's catalog need to publish a short description of the course which should be kept in synch with that on the syllabus. The syllabus, as a kind of

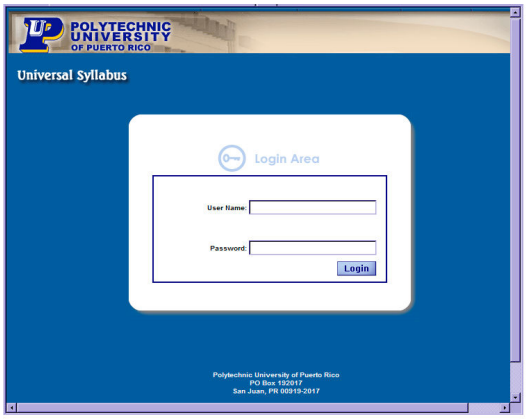
contract with the students, modulates their expectations on the course content, and has been known to led to protests and claims when what is discussed in the course fails to meet what is advertised in the syllabus and in the university catalog course description.

Program-level accrediting agencies such as the Accreditation Board for Engineering and Technology (ABET) have particular minimum requirements for syllabus content[1]. Regional institutional-level accreditation bodies such as the Middle-States Association of Colleges and Schools (MSACS) and the Middle States Commission on Higher Education (MSCHE) expect that "goals, including learning outcomes, (are) clearly articulated at every level" including the course level. Thus the syllabus is an ideal place to articulate these learning outcomes goals at the course level. State-wide program licensing bodies (i.e. Council on Higher Education) [2] also place some minimum constraints on syllabi.

Thus, to avoid duplication and the risk for proliferation of out-of-synch document versions for these different agencies, it is desirable to capture in a single repository the latest version of each of the sections of the document, and to facilitate the editing, review, and release of an official version for the syllabus. The use of editable reference tables for course codes, meeting hours, department and program names, and similar fields avoids errors of omission or inconsistencies. Using standard report generation tools, a syllabus for each course can be generated in standard formats suitable for various accreditation or licensing agencies. In addition storage of the syllabus sections in a database allows for the generation of reports or documents across the whole set of syllabi. Some of the reports or documents generated can include actualized list of the short course descriptions suitable for incorporation into the catalog and/or school website, list of textbooks for the campus bookstore, list of bibliographic references for the campus library, a mapping of where are the program outcomes being reflected throughout the curriculum, and a list of faculty responsible for the various courses.

The Universal Syllabus Application

The above list of problems and the advantages noted for automating the syllabus authoring process, prompted the decision to develop a dedicated web application.

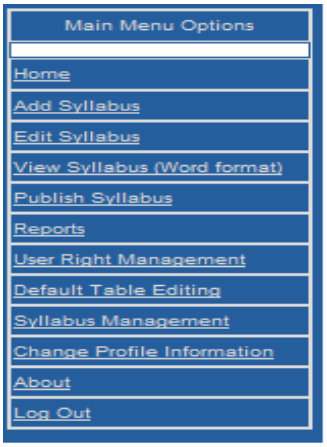


The Universal Syllabus application was originally assigned to a team of two (2) students as a Capstone project. They worked on a version for a single department, this was later extended for multiple departments and debugged and extended by a third student.

User Roles and Use-cases

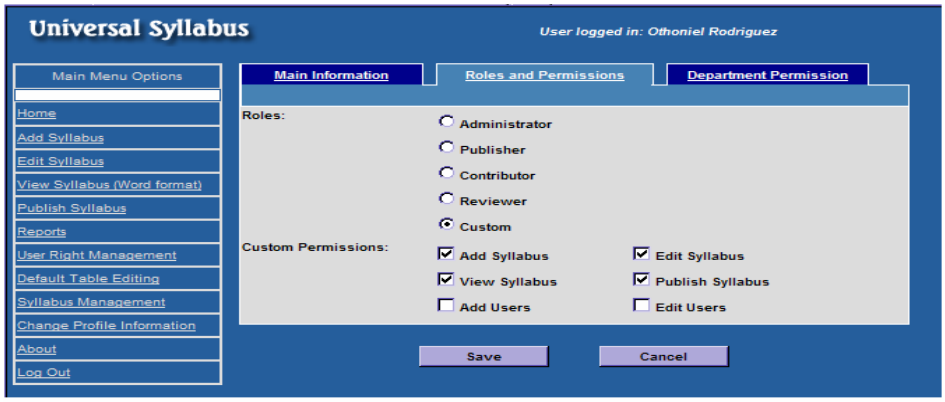
The system defines four (4) user roles: administrator, publisher, contributor, and reviewer. Each role has a defined subset of use-case permissions, from; add syllabus, view syllabus, edit syllabus, publish syllabus, add users, and edit users. There are also authoring permissions to

courses on a per-department basis and per course code department prefix. The main menu provides access to these various use cases unless otherwise disallowed by the assigned privileges. See main menu options below.



The Administrator

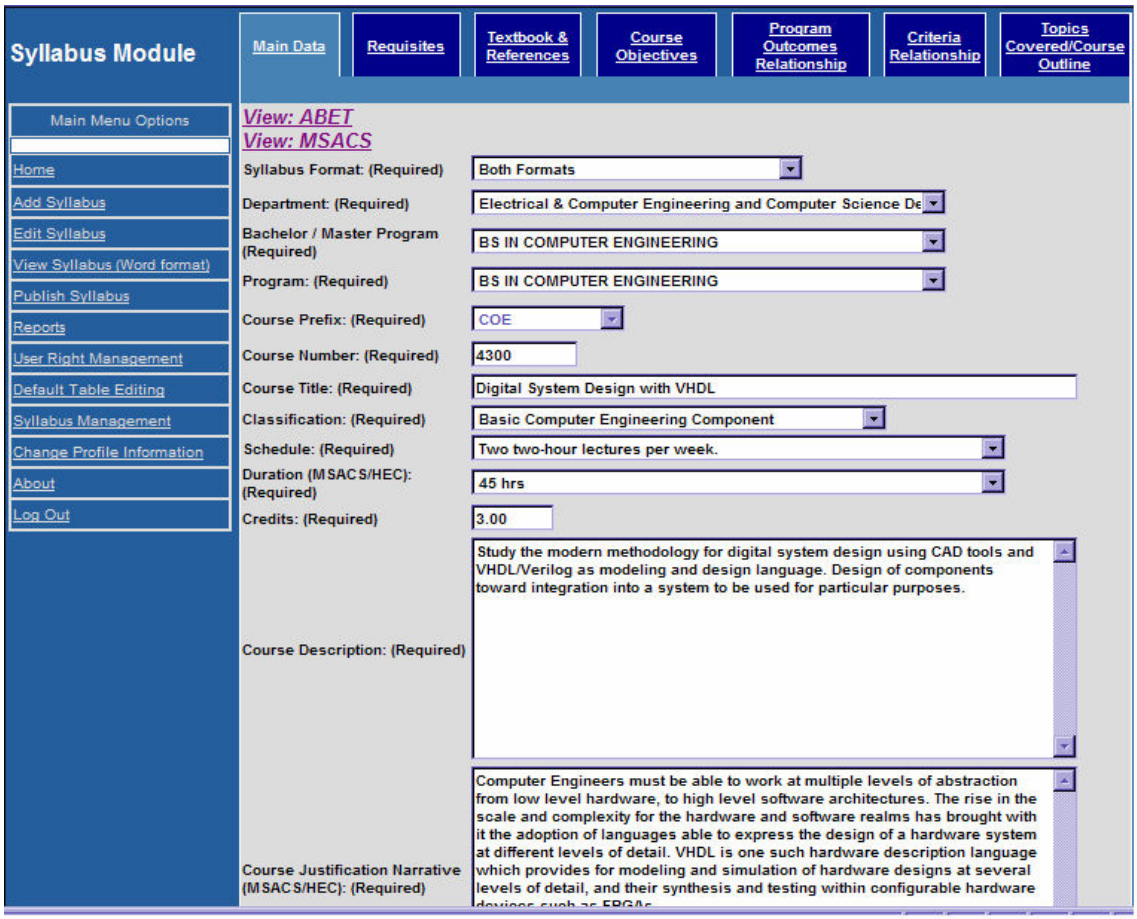
The administrator will assign roles to new user accounts, but can also create customized roles by assigning a customized set of use cases to a user. S/he can also assign the new user to a department and course code prefix. A less frequent activity of the administrator is to edit the reference tables which are used to fill the department names, program names, program-department associations, course code department prefix, list of program outcomes, and others. When deploying a new installation of Unisyllabus the administrator will spend some time setting-up the reference tables but afterwards his/her role is mostly limited to adding/removing users or resetting their passwords.



The Contributor

The Contributor is usually a faculty member who is entering (add) and/or maintaining (edit) the syllabi for which s/he is responsible. When performing this use case s/he will be entering into a multi-tab form the various sections of the syllabus. Some of these entry options will be provided as pull-down list items previously entered into reference tables by the Administrator. Those

sections that are syllabus-specific such as description, objectives, topics, etc. are entered into editable text-areas. Before saving the form into database tables those entries marked as required must receive at least a dummy entry in order to allow saving the whole form with the partially filled sections. The Contributor can save, logout, and return as many times as necessary to continue his capture of the syllabus sections content.

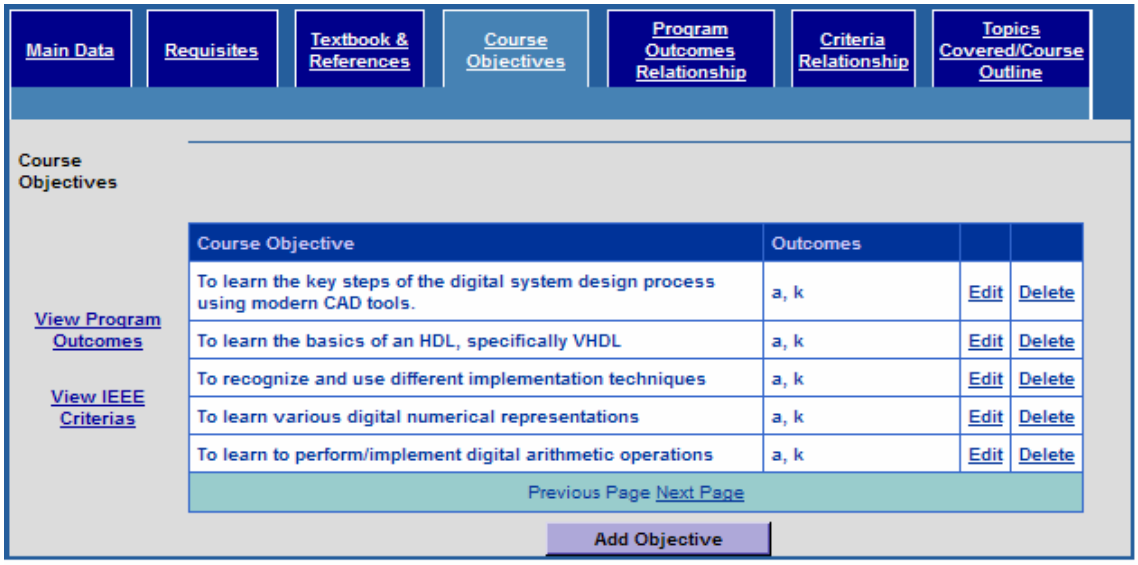


If the user needs to have an idea of how the syllabus is taking shape, s/he can click on either of two links (View: ABET, View: MSACS) which will result in the generation of the syllabus as a MS-Word document in a separate IFrame in the selected format, without the author losing his current form context.

While entering a syllabus some interactions are unavoidable, for example if the user is filling the pre-requisites or co-requisites s/he might have to pause, create (without any contents) those other courses, and then return to the original syllabus being edited and select the pre/requisite courses which will show-up in the pull-down list.

Because the system should work for non-ABET-format programs, currently it is allowing entry of the syllabus sections for either the ABET or the MSACS/CHE or BOTH format requirements.

For ABET-compatible syllabus format, additional tabs are provided that allow entry of objectives (course Objectives tab) and topics (Topics covered/Course Outline) and specifying the program ABET-outcomes[3] (a-k for engineering) that are associated with these.



To facilitate indexing to the ABET outcomes (few of the users will be able to list them by heart) a View-Program-Outcomes link is provided that will open an IFrame and list all (a-k) outcomes and their index letter for the particular program desired, this, without navigating away from the form used for filling the course objectives.

Currently, however, only the IEEE required criteria are being listed, we expect to modify the system to allow similar professional-society-related criteria for each engineering program. Because of this limitation the relationship to any professional society criteria is being directly typed into the section for Contribution to Program Outcomes and IEEE Criteria (ABET). This will be fixed in a future version of the application.

The Reviewer

This role has the same privileges as for the Contributor however its main task is to technically evaluate the syllabus contents and question or object to any issue deemed inaccurate or questionable.

Main Data | Requisites | Textbook & References | Course Objectives | Program Outcomes Relationship | Criteria Relationship | Topics Covered/Course Outline

Topics Covered

Topics Covered	Outcomes	Lesson		
General model for a synchronous digital system		Synchronous systems separate combinational logic for output generation and next state computation from the state holding storage elements. This imposes constraints on the maximum clock speed for the system.	Edit	Delete
Modeling and simulation of digital systems at various levels of abstraction.		Modeling can occur at several levels of abstraction. Each one is appropriate to particular purpose.	Edit	Delete
The VLSI Hardware Description Language (VHDL): structural, RTL, dataflow and behavioral modeling.	k	Aplicability for each type of model, cost of developing model versus level of detail being modeled. Tradeoff between designer feature control, automatic synthesis and actual chips resource contraints	Edit	Delete

The Publisher

The user in this role assures that the syllabus meets specific requirements such as no missing sections or course coding assigned, and assures that there is consistency between the syllabus as it is and the process and standard nomenclature established by the program. After making sure these standards are met, the publisher will generate a syllabus document in the available format (MS-Word) and convert it to a read-only format (Adobe-PDF) for publishing into a student-accessible repository of active/approved/published syllabi. The repository allows access through a web application or to use a url to refer to the document. This allows the professor using a course website or a Learner Management System to link directly to the published syllabus, avoiding stale versions of the syllabus.

The Student

The students have access to the repository through a web application where they can use a form to search for the desired syllabus, and view or download it.

A sample course syllabus generated by Unisyllabus in ABET format is shown in Appendix A, and sample in MSACS/CHE format is shown in Appendix B.

Reports

Currently there are two kinds of reports available, each on a per-department basis. The Textbook report lists all the textbooks included on the syllabi associated to a department.

Select Tool

CECS3210
Advanced Programming

Title	Publishing House	Edition	Author(s)	ISBN
C# How to Program	Prentice Hall	1st (2002)	Deitel and Deitel	0130622214

Last revision date: 06/20/2007
Last revision by: Felix J. Nevarez Ayala

CECS3212
Data Structures

Title	Publishing House	Edition	Author(s)	ISBN
Data Structures Using C And C++	Prentice Hall	2nd (1997)	Langsam, Augenstein, Tenenbaum	0137514050

Last revision date: 06/10/2007
Last revision by: Ottoniel Rodriguez Jimenez, PhD

CECS3214
Internet Programming I

Title	Publishing House	Edition	Author(s)	ISBN
Internet and World Wide Web: How to Program with CD-ROM BK&CD ROM	Prentice Hall	2nd (1999)	Harvey M. Deitel, Paul Deitel, T. Nieto	0130161438

Last revision date: 12/27/2003
Last revision by: Ottoniel Rodriguez Jimenez, PhD

CECS3220
Human Computer Interaction

The References report lists all the references included in the syllabi associated to a specific department. The first report facilitates providing the textbook information on a timely basis to the campus bookstore, while the second is used to assure that the Library collection is up to date, and is particularly useful during accreditation visits.

COE3321
Microprocessors Laboratory

Title	Publishing House	Edition	Author(s)	ISBN
Microcontrollers and Microcomputers	Oxford University Press	1997	Fredrick M Cady	
Microprocessor System Design	PWS Publishing Company	1997	Alan Clements	
Introduction to Microprocessor	Mc-Graw-Hill, Inc.		Ralph Horvath	
68Hc11 Reference Manual Motorola	Motorola Inc	CPU12RM/AD Rev 1.0 (1997)		
HC11 Advance Information	Motorola Inc	MC68HC912B32/D Rev. 2.0 (1999)		
68Hc12 Perfect Balance Course Notes	Motorola Inc	MTTHC12CN Rev. 2.0 (1998)		
68Hc12 Perfect Balance Lab Book	n/a	MTTHC12/LB REV 1.1		

Last revision date:
Last revision by:

COE4300
Digital System Design with VHDL

Title	Publishing House	Edition	Author(s)	ISBN
Digital Design: Principles and Practices	Prentice Hall	4th	John F. Wakerly	
Introduction to Logic Design	McGraw-Hill	2002	Alan B. Marcovitz	

Last revision date: 04/19/2007
Last revision by: Ottoniel Rodriguez Jimenez, PhD

With the metadata already available for the syllabi we expect to be able to generate additional reports. For example, a report mapping the program outcomes into the courses (and topics) which claim to be associated to those outcomes is usually used by programs seeking ABET accreditation to assure that all outcomes are being properly addressed throughout the curriculum. A report listing the professors responsible for particular courses identifies who is responsible for what and/or number of preparations. A list of departmental course syllabi sorted by date of

revision identifies those that are due for revision on a yearly basis, and of course before an accreditation visit.

Other Features

By appropriate configuration of the domain name servers the application can be made available through the campus intranet or made externally accessible. Besides password-based access control, the application was designed with robustness in mind, such as rejection of SQL injection attacks. Other features like search for available courses to view or edit by using partial course code entry, or data validation, are not discussed here as these are considered standard usability features.

Conclusion

The Unisyllabus application was designed to address some of the problems that plague the management of syllabi for a program, a department or a campus. Besides tackling the syllabi consolidation problem, it also seeks to make syllabi users more aware of ABET and similar accreditation bodies issues. The developers have started to populate the system with the syllabi for their Electrical and Computer Engineering Department. Once all the syllabi are in place and we do not find any major concerns we plan on deploying Unisyllabus campus-wide and to make it available to other institutions. Higher education institutions face problems similar to those of any other enterprise. These have to do with adopting information technology to make processes more efficient, and extracting business intelligence that will help make strategic decisions. The Unisyllabus application is an example of an application that begins to address some of these issues.

Bibliography

[1] Accreditation Board for Engineering and Technology, see section "Criterion 5 Curriculum" for syllabi requirements. Available Online: http://www.abet.org/Linked%20Documents-UPDATE/Program%20Docs/T003%20Self-Study%20Questionnaire%206-28-07.doc

[2] Consejo de Educacion Superior de Puerto Rico (Puerto Rico Council on Higher Education), Reglamento para el Otorgamiento de Licencias a Instituciones de Educacion Superior en Puerto Rico Núm. 6543, del 31 de octubre de 2002. Available Online: http://www.gobierno.pr/NR/rdonlyres/DAE616AC-C05C-4653-A0A3-DB61E4CCD2AC/0/Reglamento_OLA_Sept_2002.pdf

[3] Accreditation Board for Engineering and Technology, see Criteria for Accrediting Engineering Programs. http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/E001%2008-09%20EAC%20Criteria%2011-30-07.pdf

Appendix A: ABET-Format Syllabus

POLYTECHNIC UNIVERSITY OF PUERTO RICO
Electrical & Computer Engineering and Computer Science Dept.
COURSE SYLLABUS

Course Code:	Course Title:	Classification:	Credits:
COE4300	Digital System Design with VHDL	Basic Computer Engineering Component	3.00

PreRequisites:	Co-Requisites:	Schedule:
COE3300,COE3301		Two two-hour lectures per week.

Course Description:

Study the modern methodology for digital system design using CAD tools and VHDL/Verilog as modeling and design language. Design of components toward integration into a system to be used for particular purposes.

Textbook:

1. Fundamentals of Digital Logic with VHDL Design, Stephen Brown, Zvonko Vranesik, McGraw-Hill, 2, McGraw-Hill,

References:

1. Digital Design: Principles and Practices, John F. Wakerly, 4th, Prentice Hall,
2. Introduction to Logic Design, Alan B. Marcovitz, 2002, McGraw-Hill,

Contribution to Professional Component:

Provide computer engineering students with the ability to design complex digital systems as well as the methodology of CAD design using a Hardware Description Language (HDL) as design tool.

Contribution of Program Outcomes and IEEE Program Criteria:

This is a key course to measure outcomes a, k, and IEEE criteria 5 & 6 the course instructor has a commitment to measure the key parts off these outcomes on a regular basis, and to show documented evidence to prove their achievement.

Outcome a: Ability to apply knowledge of mathematics (a1 key), science (a2) and engineering (a3 key).
Outcome k: Ability to use techniques (k1 key), skills (k2 key), and modern engineering tools (k3 key), necessary for engineering practice.
Knowledge of Linear Algebra and Discrete Mathematics.

Course Objectives (Related to Program Outcomes and the IEEE Program Criteria):

1. To learn the key steps of the digital system design process using modern CAD tools. (a, k)
2. To learn the basics of an HDL, specifically VHDL (a, k)
3. To recognize and use different implementation techniques (a, k)
4. To learn various digital numerical representations (a, k)
5. To learn to perform/implement digital arithmetic operations (a, k)
6. To recognize and use the building blocks of a Digital System (a, k)

Topics Covered:

1. General model for a synchronous digital system
2. Modeling and simulation of digital systems at various levels of abstraction.
3. The VLSI Hardware Description Language (VHDL): structural, RTL, dataflow and behavioral modeling.
4. Structural modeling constructs in VHDL.
5. Behavioral modeling constructs in VHDL.
6. RTL modeling in VHDL
7. Dataflow modeling in VHDL.
8. Synthesis of VHDL models.
9. Simulation.

Evaluation Criteria:

Instruments for course evaluation will be used to measure established course objectives. This instruments will be associated with the Program Outcomes and the IEEE Program Criteria that they can support. These are:

Four (4) class projects of increasing level of complexity.

The course instructor will be responsible to show the course coordinator how these instruments will determine the final grade. The same person will also choose the most adequate instrument(s) from the above list to gather data, analyze it, and display valuable evidence of the Program Outcomes Assessment for which this is a key course.

Prepared by: Angel Gonzalez Lizardo, PhD	**Date:** 04/02/2007
Revised by: Othoniel Rodriguez Jimenez, PhD	**Date:** 04/19/2007

Appendix B: CHE/MSACS-Format Syllabus

POLYTECHNIC UNIVERSITY OF PUERTO RICO
Electrical & Computer Engineering and Computer Science Dept.
BS IN COMPUTER ENGINEERING
Course Syllabus

Course Title: Digital System Design with VHDL

Course Code: COE4300

Credits: 3.00

Duration: 45 hrs

Schedule: Two two-hour lectures per week.

PreRequisite: COE3300,COE3301

Co-Requisite:

Course Description:
Study the modern methodology for digital system design using CAD tools and VHDL/Verilog as modeling and design language. Design of components toward integration into a system to be used for particular purposes.

Justification:
Computer Engineers must be able to work at multiple levels of abstraction from low level hardware, to high level software architectures. The rise in the scale and complexity for the hardware and software realms has brought with it the adoption of languages able to express the design of a hardware system at different levels of detail. VHDL is one such hardware description language which provides for modeling and simulation of hardware designs at several levels of detail, and their synthesis and testing within configurable hardware devices such as FPGAs.

Objectives:
The course objectives should be aligned with the program outcomes.

1. To learn the key steps of the digital system design process using modern CAD tools.
2. To learn the basics of an HDL, specifically VHDL
3. To recognize and use different implementation techniques
4. To learn various digital numerical representations
5. To learn to perform/implement digital arithmetic operations
6. To recognize and use the building blocks of a Digital System

Textbook:
Fundamentals of Digital Logic with VHDL Design
by Stephen Brown, Zvonko Vranesik, McGraw-Hill
McGraw-Hill
2
ISBN:

Course Outline:
Some of the course topics will span multiple meeting sessions while some sections might include more than one topic.

1. General model for a synchronous digital system