IEEE Common Nomenclature for Computing Related Programs in Latin America
IEEE Common Nomenclature for Computing Related Programs in Latin America

Edited by

Teófilo J. Ramos
Osvaldo M. Micheloud
Richard Painter
Moshe Kam
# TABLE OF CONTENTS

EXECUTIVE SUMMARY .................................................................................................................................................. 80

CONTRIBUTORS ............................................................................................................................................................. 82
  Organizing Committee and Editors .......................................................................................................................... 82
  Working Group Participants ................................................................................................................................... 82
  Universities Participating in the Pilot Test of the Mapping Guidelines .............................................................. 83

I. INTRODUCTION ......................................................................................................................................................... 85

II. INTENDED USERS ..................................................................................................................................................... 85

III. CURRENT LANDSCAPE ........................................................................................................................................... 86

IV. WORKING METHODOLOGY .................................................................................................................................. 86

V. CATEGORIES DEFINED BY THE WORKGROUP ..................................................................................................... 86

VI. COMPETENCIES .................................................................................................................................................... 87
  VI.1. Common Competencies for all Categories ....................................................................................................... 87
  VI.2. Computer Science Competencies .................................................................................................................... 88
  VI.3. Information Systems Competencies ................................................................................................................ 89
  VI.4. Software Engineering Competencies .............................................................................................................. 90
  VI.5. Computer Engineering Competencies ........................................................................................................... 91
  VI.6 Information Technology Competencies .......................................................................................................... 91

VII: CONCLUSIONS ......................................................................................................................................................... 93

REFERENCES ................................................................................................................................................................. 93

APPENDIX I .................................................................................................................................................................... 94
  Guidelines for Mapping a Computing-Related Degree Program to the Defined Categories ..................................... 95

APPENDIX II Self-Assessment of the Systems Engineering Undergraduate Program
  Summary of the Experience of Ort Uruguay .......................................................................................................... 97
EXECUTIVE SUMMARY

Academic degree programs in computing-related fields have developed under different traditions and philosophies in Latin America and around the world. While the terms and phrases used to describe these programs within nations are generally understood by local practitioners and academic officials, these descriptive titles often mask significant differences in content when applied across regions. It is common to find programs with the same name but with different content and programs with different names and similar content. As a result, employers hesitate to hire graduates from academic programs with which they are not familiar, and the professional mobility of both new and existing graduates is unnecessarily limited.

In 2010, a small group of volunteers, herein called Organizing Committee, led by Dr. Moshe Kam, IEEE Past President, and Dr. Teófilo J. Ramos, IEEE, Educational Activities Board (EAB) Past Chair, decided to conduct a study of the computing engineering undergraduate programs in Latin America, Spain, USA and UK. The objective was to identify graduate competencies and main categories in computing in Latin American countries with a large number of students in these fields. The concept was to develop a tool to help each program interested in becoming internationally recognizable, to easily identify an internationally recognized name for international job searches and student exchange. The goal of this document is to enhance the consistency of program naming in order to increase professional mobility. It does not intend to be a new accreditation system.

The starting point of this task was to identify a few well-known specialists in the field from the countries with the largest number of computing-related engineering programs, and invite them to participate in the “working group.” They were then asked to produce a document called the “country summary”, which described the situation in their respective countries. All of this information was then shared among the working group for further review and discussion.

After one year of hard work, the working group” participated in a two-day meeting held in Lima, Peru, in April 2011. The purpose of the session was to formulate a document that defined the internationally recognized categories of computing-related university programs adapted to the Latin America context. This document is intended to serve as a first measure of approximation to which a given program conforms to one or various categories of computing-related programs defined by the working group.

The guidelines developed in Lima were then tested by several programs as part of a pilot phase. The feedback and suggestions received during this pilot phase are included in this document, and applied in two programs, one of them included as an example in this document.

This document is made available to government and accreditation agencies, professional associations in Latin America and to the international community through IEEE. It is intended to help these organizations in determining the category of their program that is recognized by the largest international organizations according to the categories specified in this document.
CONTRIBUTORS

Organizing Committee and Editors
Moshe Kam, IEEE Past President
Teófilo J. Ramos, IEEE EAB Past Vice President and Program Chair
Osvaldo Micheloud, Energy Research Chair Professor, Tecnológico de Monterrey, México
Richard Painter, IEEE, EAB Accreditation Policy Council (APC) Past Chair

Working Group Participants

Argentina
Daniel Morano, Asesor Secretaría de Políticas Universitarias, Ministerio de Educación
Guillermo Ricardo Simari, Universidad Nacional del Sur

Brazil
Daltro Nunes, Universidade Federal do Rio Grande do Sul
Jorge Luis Nicolas Audy, Pró-Reitor de Pesquisa e Pós-Graduação da PUCRS

Chile
Luis Salinas, Informatics Department, Universidad Federico Santa María
Hector Kaschel, Dean of Graduate Studies, Universidad de Santiago de Chile

Colombia
Germán A. Chavarro F. Universidad Javeriana Bogotá
José Ismael Peña Reyes, Academic Director of Systems and Industrial Engineering Programs, Univ. Nacional de Colombia

Costa Rica
Ignacio Trejos Zelaya, Centro de Formacion en Tecnologias de Informacion
Lilliana Sancho Chavarria Instituto Tecnologico de Costa Rica

Mexico
Rodolfo Castello Zetina, Director of the Division of Mechatronics and Information Technologies, Tecnologico de Monterrey
Guillermo Rodriguez Abitia, Director of Research, Development and Innovation, General Direction of Computing, Information Technologies and Communication, UNAM

Peru
Ernesto Cuadros-Vargas, Sociedad Peruana de Computación and Universidad Catolica San Pablo
César Luza Montero, Universidad Nacional Mayor de San Marcos, Lima
Carlos Javier Solano Salinas Universidad Nacional de Ingeniería, Lima
Spain
Javier Segovia, Dean of the School of Informatics, Universidad Politécnica de Madrid

United Kingdom
Leslie Smith, Chair, Department of Computing Science and Mathematics, University of Stirling

United States
Stephen Seidman, Dean College of Science, Texas State University-San Marcos

Uruguay
Julio Fernández, Dean of Academic Development, Universidad ORT Uruguay
Ariel Sabiguero Yawelak Facultad de Ingeniería, Universidad de la República

Universities Participating in the Pilot Test of the Mapping Guidelines

Universidad ORT, Uruguay
Universidad Javeriana Bogotá, Colombia
Tecnológico de Monterrey, México
Universidad Católica San Pablo, Perú
Universidad Politécnica de Madrid, Spain
Universidad de la República, Uruguay
IEEE Common Nomenclature for Computing Related Programs in Latin America

I. INTRODUCTION

The IEEE held a two-day workshop in April 2011 involving leading academic administrators and faculty from across Latin America to examine the variation of computing program names versus program contents and develop responses. Because of this variation, a need exists, without mandating unified names, for an agreed set of competencies that embody the main categories in computing. The event identified specific content differences in undergraduate computing programs in Argentina, Brazil, Chile, Colombia, Costa Rica, Mexico, Peru, Uruguay, Spain, UK and the USA. Workshop participants were asked to collaborate in developing a roadmap, with specific steps aimed at the issues of: 1) establishing consensus on the knowledge and skill sets that should be common to graduates of computing-related baccalaureate programs located across these regions; and 2) developing a shared nomenclature to describe relevant academic programs that can be used with confidence by employers, graduate admissions personnel, prospective students and other constituents. Solving this problem will improve the employability of graduates across borders without the need to emigrate. The ACM/AIS/IEEE Computing Curricula 2005 [1], Spain’s Ministry of Education Bulletin No 187 [2] and other related works were guiding references for this report.

II. INTENDED USERS

Employers and personnel recruiters:
- Describing needs for technical personnel
- Defining background for positions wanted
- Creating job descriptions for technical functions

Universities and governmental educational agencies:
- Creating new study programs in the field
- Classifying of existing programs
- Helping students to select elective content in order to conform to a given orientation

Accreditation agencies:
- Defining program specialties for accreditation review
- Reviewing contents /competencies associated with program profiles
- Integrating evaluation teams to match the profile of reviewed programs

Professional associations in the field:
- Supporting work on definition of professional competencies
- Organizing internal activities according to the different disciplines
III. CURRENT LANDSCAPE

Technology and the computing field are changing rapidly. New knowledge areas are emerging faster than program names can be changed. Varying names are given to programs for historical reasons or in response to market pressures to attract students. Some countries have unified program names but in others it is difficult to change existing titles. The Working Group participants were to varying degrees in agreement with The ACM/AIS/IEEE Computing Curricula 2005, Spain’s Ministry of Education Bulletin No 187 and other related works. All of which were guiding references for this report. There are many accrediting bodies with different standards for computing programs which influence variation in program names and content. It is not realistic to believe that already established programs would be willing to change their name.

IV. WORKING METHODOLOGY

The IEEE charged the working group to explore solutions to addressing the challenge. Each produced, following a guideline, a summary of the computing education and employment situation in their respective countries. These summaries were made available prior to the workshop for all participants to review. Presentations were made and discussed the first morning of the workshop. The first afternoon was devoted to a panel discussion and additional presentations to outline how to proceed.

It was concluded that it would be unreasonable to expect everyone to use a common set of program names. However, it is possible, without coercing unified names for undergraduate programs, to use a common set of categories for computing that express a common set of competencies or professional capabilities. The group formulated the approach embodied here and worked to develop sets of competencies for the computing areas.

In an attempt to address, this working group began by reviewing the ACM/AIS/IEEE Computing Curricula 2005 and updating the professional capabilities here to account for regional needs and contemporary changes in the field. They analyzed the competencies of each area listed in the ACM/AIS/IEEE Computing Curricula 2005, modified and agreed on a set of revised competencies contained herein. In the context of this document “competencies” means “professional capabilities”.

V. CATEGORIES DEFINED BY THE WORKING GROUP

After much debate and careful consideration of each country’s/region’s need, it was consensually decided to select the following categories defined by the ACM/AIS/IEEE Computing Curricula 2005 in addition to common requirements:

- computer science
- information systems
- software engineering
- computer engineering
- information technology
- emerging disciplines and new hybrid programs
VI. COMPETENCIES

This section contains common and category-specific competencies for the identified categories to facilitate program leaders and others to assess the degree of conformance of their programs.

This document does not define any category-specific competencies for emerging disciplines and new hybrid programs.

VI.1. Common Competencies for all Categories

This section is based on the ACM/AIS/IEEE Computing Curricula 2005.

Each of the major computing disciplines has its own character. Each one is somewhat different from its siblings in the emphasis, goals, and capabilities of its graduates. Yet they have much in common. Any reputable computing degree program should include each of the following elements.

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory
C2. Ability to have a critical and creative perspective in identifying and solving problems using computational thinking.
C3. An intellectual understanding of, and an appreciation for, the central role of algorithms and data structures.
C4. An understanding of computer hardware from a software perspective, for example, use of the processor, memory, disk drives, display, etc.
C5. Ability to implement algorithms and data structures in software.
C6. Ability to design and implement larger structural units that utilize algorithms and data structures and the interfaces through which these units communicate.
C7. Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.
C8. Understanding of what current technologies can and cannot accomplish.
C9. Understanding of computing’s limitations, including the difference between what computing is inherently incapable of doing vs. what may be accomplished via future science and technology.
C10. Understanding of the impact on individuals, organizations, and society of deploying technological solutions and interventions.
C11. Understanding of the concept of the lifecycle, including the significance of its phases (planning, development, deployment, and evolution).
C12. Understanding the lifecycle implications for the development of all aspects of computer-related systems (including software, hardware, and human computer interface).
C13. Understanding the relationship between quality and lifecycle management.
C14. Understanding of the essential concept of process as it relates to computing especially program execution and system operation.
C15. Understanding of the essential concept of process as it relates to professional activity, especially the relationship between product quality and the deployment of appropriate human processes during product development.
C16. Ability to identify advanced computing topics and understanding the frontiers of the discipline.
C17. Ability to properly express in oral and written media as expected from a university graduate.
C18. Ability to participate actively and as a member of a team.
C19. Ability to effectively identify the goals and priorities of their project, stating the action, the time and resources required.
C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.
C21. Understanding the professional, legal, security, political, humanistic, environmental, cultural and ethical issues.
C22. Ability to demonstrate attitudes and priorities that honor, protect, and enhance the profession’s ethical stature and standing.
C23. Ability to undertake, complete, and present a capstone project.
C24. Understanding the need for lifelong learning and improving skills and abilities.
C25. Ability to communicate in a second language.

VI.2. Computer Science Competencies

This section is based on the ACM/IEEE Computer Science Curriculum 2008: An Interim Revision of CS 2001 [3].

Computer science spans a wide range, from its theoretical and algorithmic foundations, to cutting-edge developments in robotics, computer vision, intelligent systems, bioinformatics, and other exciting areas.

The work of computer scientists falls into three categories:

- Design and implement software. Computer scientists take on challenging programming jobs. They also supervise other programmers, keeping them aware of new approaches.
- Devise new ways to use computers. Progress in the CS areas of networking, database, and human-computer-interface, enabled the development of the World Wide Web. Now CS researchers are working with scientists from other fields to make robots become practical and intelligent aides, to use databases to create new knowledge, and to use computers to help decipher the secrets of our DNA.
- Develop effective ways to solve computing problems. For example, computer scientists develop the best possible ways to store information in databases, send data over networks, and display complex images. Their theoretical background allows them to determine the best performance possible, and their study of algorithms helps them to develop new approaches that provide better performance.

Computer science spans the range from theory through programming. While other disciplines may produce graduates with more immediately relevant job-related skills, computer science offers a comprehensive foundation that permits graduates to adapt to new technologies and new ideas.

Competencies and skills relating to computer science are abilities to:

CS1. Model and design computer-based systems in a way that demonstrates comprehension of the tradeoff involved in design choices.
CS2. Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.

CS3. Analyze the extent to which a computer-based system meets the criteria defined for its current use and future development.

CS4. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.

CS5. Specify, design, and implement computer-based systems.

CS6. Evaluate systems in terms of general quality attributes and possible tradeoffs presented within the given problem.

CS7. Apply the principles of effective information management, information organization, and information-retrieval skills to information of various kinds, including text, images, sound, and video. This must include managing any security issues.

CS8. Apply the principles of human-computer interaction to the evaluation and construction of a wide range of materials including user interfaces, web pages, multimedia systems and mobile systems.

CS9. Identify any risks (and this includes any safety or security aspects) that may be involved in the operation of computing equipment within a given context.

CS10. Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems. This should include tools for software control including version control and configuration management.

CS11. Be aware of the existence of publicly available software and understanding the potential of open-source projects.

CS12. Operate computing equipment and software systems effectively.

VI.3. Information Systems Competencies

This section is based on the ACM/AIS IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems [4].

Information systems specialists focus on integrating information technology solutions and business processes to meet the information needs of businesses and other enterprises, enabling them to achieve their objectives in an effective, efficient way. This discipline’s perspective on information technology emphasizes information, and views technology as an instrument for generating, processing, and distributing information. Professionals in the discipline are primarily concerned with the information that computer systems can provide to aid an enterprise in defining and achieving its goals, and the processes that an enterprise can implement or improve using information technology. They must understand both technical and organizational factors. They must be able to help an organization determine how information and technology-enabled business processes can provide a competitive advantage.

The information systems specialist plays a key role in determining the requirements for an organization’s information systems and is active in their specification, design, and implementation. As a result, such professionals require a sound understanding of organizational principles and practices so that they can serve as an effective bridge between the technical and management communities within an organization, enabling them to work in harmony to ensure that the organization has the information and the systems it needs to support its operations. Information systems professionals are also involved in designing
technology-based organizational communication and collaboration systems.

Information systems specialists should be able to analyze information requirements and business processes and be able to specify and design systems that are aligned with organizational goals.

Competencies and skills relating to information systems include the abilities to:

IS1. Identify, understand and document information systems requirements.
IS2. Account for human-computer interfaces and intercultural differences, in order to deliver a quality user experience.
IS3. Design, implement, integrate and manage IT systems, enterprise, data and application architectures.
IS4. Manage information systems projects, including risk analysis, financial studies, budgeting, procurement and development, and to appreciate the challenges of information systems maintenance.
IS5. Identify, analyze and communicate problems, options and solution alternatives, including feasibility studies.
IS6. Identify and understand opportunities created by technological innovations.
IS7. Appreciate the relationships between business strategy and information systems, architecture and infrastructure.
IS8. Understand business processes and the application of IT to them, including change management, control and risk issues.
IS9. Understand and implement secure systems, infrastructures and architectures.
IS10. Understand performance and scalability issues.
IS11. Manage existing information systems including resources, maintenance, procurement and business continuity issues.

VI.4. Software Engineering Competencies


Software engineering has evolved in response to factors such as the growing impact of large and expensive software systems in a wide range of situations and the increased importance of software in safety-critical applications. Software engineering is different in character from other engineering disciplines due to both the intangible nature of software and the discontinuous nature of software operation. It seeks to integrate the principles of mathematics and computer science with the engineering practices developed for tangible, physical artifacts.

Mastery of the software engineering knowledge and skills, and professional issues necessary to begin practice as a software engineer include the abilities to:

SE1. Develop, maintain and evaluate software systems and services to meet all user requirements and behave reliably and efficiently, are affordable to develop and maintain and meet quality standards, applying the theories, principles, methods and best practices of Software Engineering.
SE2. Assess customer needs and specify the software requirements to meet those needs, reconciling
conflicting goals by finding acceptable compromises within the constraints arising from the cost, time, the existence of systems already developed and the organizations themselves.

SE3. Solve integration problems in terms of strategies, standards and technologies available.

SE4. Work as an individual and as part of a team to develop and deliver quality software artifacts. Understand diverse processes (activities, standards and lifecycle configurations, formality as distinguished from agility) and roles. Perform measurements and analysis (basic) in projects, processes and product dimensions.

SE5. Reconcile conflicting project objectives, finding acceptable compromises within limitations of cost, time, knowledge, existing systems, organizations, engineering economics, finance and the fundamentals of risk analysis and management in a software context.

SE6. Design appropriate solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal, and economic concerns.

SE7. Demonstrate an understanding of, and apply current theories to, models, and techniques that provide a basis for problem identification and analysis, software design, development, construction and implementation, verification and validation, documentation and quantitative analysis of design elements and software architectures.

SE8. Demonstrate an understanding of software reuse and adaptation, perform maintenance, integration, migration of software products and components, prepare software elements for potential reuse and create technical interfaces to components and services.

SE9. Demonstrate an understanding of systems of software and their environment (business models, regulations).

VI.5. Computer Engineering Competencies

This section is based on the IEEE/ACM Computer Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering [7]-

Computer engineering is concerned with the design and construction of computers and computer-based systems. It involves the study of hardware, software, communications, and the interaction among them.

Its curriculum focuses on the theories, principles, and practices of traditional electrical engineering and mathematics and applies them to the problems of designing computers and computer-based devices.

Computer engineering students study the design of digital hardware systems including communications systems, computers, and devices that contain computers. They study software development, focusing on software for digital devices and their interfaces with users and other devices. CE study may emphasize hardware more than software or there may be a balanced emphasis. CE has a strong engineering flavor.

Currently, a dominant area within computing engineering is embedded systems, the development of devices that have software and hardware embedded in them. For example, devices such as cell phones, digital audio players, digital video recorders, alarm systems, x-ray machines, and laser surgical tools all require integration of hardware and embedded software and are all the result of computer engineering.

Competencies of Computer Engineering Graduates include the abilities to:

CE1. Specify, design, build, test, verify and validate digital systems, including computers, microprocessor-based systems and communications systems.

CE2. Develop specific processors, embedded systems, software development, and optimization of such
CE3. Analyze and evaluate computer architectures, including parallel and distributed platforms, as well as developing and optimizing software for them.
CE4. Design and implement software for communications system.
CE5. Analyze, evaluate and select hardware and software platforms suitable for application support and real-time embedded systems.
CE6. Understand, implement and manage the security and safety systems.
CE7. Analyze, evaluate, select and configure hardware platforms for the development and implementation of software applications and services.

**VI.6 Information Technology Competencies**

This section is based on the ACM/IEEE Information Technology 2008 Curriculum Guidelines for Undergraduate Degree Programs in Information Technology [8].

IT professionals are in charge of assuring that IT systems work properly, are available, secure, safe, upgraded, maintained, and replaced as appropriate. They are more concerned with the technology itself than on the information it conveys. The IT professional should have the abilities to:

**IT1.** Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs within an organizational and societal context.
**IT2.** Identify and analyze user needs and incorporate them in the selection, creation, evaluation and administration of computer-based systems.
**IT3.** Integrate effectively IT based solutions, including the user environment.
**IT4.** Function as a user advocate. Explain and apply appropriate information technologies and employ best practices standards and appropriate methodologies to help an individual or organization achieve its goals and objectives.
**IT5.** Assist in the creation of an effective project plan.
**IT6.** Manage the information technology resources of an individual or organization.
**IT7.** Anticipate the changing direction of information technology and evaluate and communicate the likely utility of new technologies to an individual or organization.
VII: CONCLUSIONS

This document is made available to government and accreditation agencies, professional associations in Latin America and to the international community. This document defines the categories to classify computational undergraduate programs in Latin America on the basis of the competencies that graduates must have, in accordance with the recommendations of the key professional associations in the field. The Working Group revised and adapted the selected competencies to the Latin American context.

Appendix I describes a methodology for an institution or agency to conduct an assessment of the competencies for programs in computer related areas. Based on the results, a determination can be made on which of the defined categories corresponds to its program, independent of the name of the program.

With the increasing need for program accreditation, institutions using this document will have the ability to define the IEEE category for the program and employ a common naming convention that corresponds to the professional competencies of graduation.

REFERENCES

APPENDIX I

Guidelines for Mapping a Computing-Related Degree Program to the Defined Categories

These guidelines describe a self-assessment process for the exploratory mapping of degree programs to the previously defined categories. The objective of the process is to determine the degree to which a given program conforms to a selected category of computing programs.

The university team should prepare a table, a competencies-course matrix, in which a row is dedicated to each competency specified in the category definition, and a column is assigned to each course of the program.

The courses included are the ones that are listed in the official study plan. They are regular courses, but also courses consisting of seminars, projects, industry apprenticeships, or any other activity described in the general study plan including the elective courses which must be chosen to complete the program.

A. Each course in the program is then analyzed and compared systematically to the competencies listed in the rows of Competencies, assessing specific contributions of the course to each competency.

B. Based on the degree of contribution to the development of the competency, the table cells in the competencies-course matrix, are filled out using the following metrics:
   - A number 2 – (D, Developed) is inserted in the cell if the competency is fully or partially developed, assessed in the course and evidence is available.
   - A number 1 – (PD, Partially Developed) is inserted in the cell if the course contributes in some way to the development of the competency, but the competency is not assessed.
   - A number 0/blank – (ND, Not Developed) is inserted in the cell if the course does not have any contribution to the development of the competency.

From the numeric evaluation assigned by the evaluator in the competency-course matrix, the “Coverage of Competency” and the “Intensity of Coverage” for each competency and for each category can be calculated, using the procedure described in this document.

The Intensity of Coverage indicates the percentage of courses in which the competencies from each category are developed and evaluated, wholly or partly and is expressed as a percentage of the maximum possible. The categories with the highest values of Intensity of Coverage indicate the core professional competencies or strengths of the program that define the IEEE category of the same. This indicator also provides information about the intensity of coverage of each competency within a category.
The **Coverage of Competency** is an indicator of the percentage of competencies that are assessed (partially or fully) within a category.

Appendix 2 describes a detailed example of the application of the methodology to determine the IEEE category for computing-related academic programs based on the assessment of the professional competencies acquired by graduates. This example corresponds to the Systems Engineering undergraduate program at the University ORT Uruguay.

The self-assessment results are displayed in a Table 1 (below). Values obtained for the **Intensity of Coverage and the Coverage Competency** for the competency categories defined in the document IEEE Common Nomenclature for Computing-Related Programs in Latin America for the program being assessed.

<table>
<thead>
<tr>
<th>Category</th>
<th>Intensity of Coverage</th>
<th>Coverage of Competency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Competencies</td>
<td>20%</td>
<td>92.00%</td>
</tr>
<tr>
<td>Computer Science Competencies</td>
<td>17%</td>
<td>91.67%</td>
</tr>
<tr>
<td>Information Systems Competencies</td>
<td>7%</td>
<td>100%</td>
</tr>
<tr>
<td>Software Engineering Competencies</td>
<td>16%</td>
<td>100%</td>
</tr>
<tr>
<td>Computer Engineering Competencies</td>
<td>4%</td>
<td>100%</td>
</tr>
<tr>
<td>Information Technology Competencies</td>
<td>6%</td>
<td>85.71%</td>
</tr>
</tbody>
</table>

The assessment detected that the program covers more than 85% of the competencies in each category. The categories with the greatest intensity correspond to the Common Competencies, followed by the Computer Science and Software Engineering competencies. Therefore, this program corresponds to the Computer Science and Software Engineering categories, matching the design objectives of the program known as Systems Engineering.

**CONCLUSIONS**

This methodology allows an educational institution to determine the alignment of its program with one or several of the categories/names defined by IEEE for computing-related programs and their associated professional competencies. Employers and other organizations will recognize that graduates have demonstrated their professional competencies, independent from the name of their academic program, in a format documented and approved by IEEE.
APPENDIX II
Self-Assessment of the Systems Engineering Undergraduate Program
Summary of the Experience of Ort Uruguay

TABLE OF CONTENTS

INTRODUCTION ..................................................................................................................................................98
DESCRIPTION OF THE SYSTEMS ENGINEERING PROGRAM ........................................................................98
SELF-ASSESSMENT RESULTS .......................................................................................................................98
ASSESSMENT OF THE SET OF COMPULSORY COURSES (WITHOUT THE ELECTIVE COURSES) .................99
ASSESSMENT OF THE SET OF COMPULSORY COURSES (WITH THE ELECTIVE COURSES) ......................100
SELF-ASSESSMENT PROCESS .....................................................................................................................100
Curriculum Objectives ...................................................................................................................................107
CONTRIBUTORS TO APPENDIX II ...............................................................................................................107
REFERENCES ....................................................................................................................................................108
INTRODUCTION

The aim of this report is to illustrate the process and the results of the self-assessment of the Systems Engineering program at ORT University in Uruguay within the context of the IEEE Common Nomenclature for Computing-Related Programs in Latin America project.

After complying with the self-assessment criteria defined in the document “Guidelines for Mapping a Computing-Related Degree Program to the Defined Categories”, the result obtained matches the intent originally defined for the program. The evaluation of the Systems Engineering program, in relation to the degree and Intensity of Coverage of the competency categories, reveals that the program’s strengths lie in the Common Competencies, followed by Computer Science and Software Engineering Competencies.

In order to contextualize the analysis of the results obtained, the following section describes the objectives and characteristics of the program, the recommendations used to design the curriculum, and also compares the self-assessment results. Finally, the process used to assess the degree program is specified.

DESCRIPTION OF THE SYSTEMS ENGINEERING PROGRAM

This section provides contextual information on the assessed program to facilitate the interpretation of the results obtained from the self-evaluation exercise.

The general objectives of the program’s curriculum indicate that one of its main characteristics is emphasis on the area of software engineering, accompanied by a solid theoretical/practical preparation that offers graduates, among other skills, the capacity for: self-education, detecting needs, problem solving and applying the knowledge acquired. It is also important to note that the curriculum designers seriously considered the IEEE/ACM curricular recommendations for Computer Science [2] preliminary versions of the recommendation for Software Engineering [3] and the body of knowledge for Software Engineering—SWEAK [1].

The program has a core of thirty six compulsory courses, seven elective courses and a one year-long capstone project. The aim of the elective courses is to provide a set of in-depth profiles in diverse application domains, such as: Information Systems, Systems Architecture and Development, Management and Business, etc. Students choose a profile and select four courses from the corresponding set of electives, and the remaining three are free-choice electives.

SELF-ASSESSMENT RESULTS

The results of the self-assessment of the Systems Engineering program are described in this section. The assessment of the program’s set of compulsory subjects is presented first. It is then followed, as an example, by the elective courses corresponding to the Information Systems profile.
ASSESSMENT OF THE SET OF COMPULSORY COURSES (WITHOUT THE ELECTIVE COURSES)

The following table details the values obtained for the Competency Coverage and Intensity of Coverage for the competency categories defined in the document “Guidelines For Mapping a Computing-Related Degree Program to the Defined Categories”.

<table>
<thead>
<tr>
<th>Category</th>
<th>Intensity of Coverage</th>
<th>Competency Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Competencies</td>
<td>20%</td>
<td>92.00%</td>
</tr>
<tr>
<td>Computer Science Competencies</td>
<td>17%</td>
<td>91.67%</td>
</tr>
<tr>
<td>Information Systems Competencies</td>
<td>7%</td>
<td>100%</td>
</tr>
<tr>
<td>Software Engineering Competencies</td>
<td>16%</td>
<td>100%</td>
</tr>
<tr>
<td>Computer Engineering Competencies</td>
<td>4%</td>
<td>100%</td>
</tr>
<tr>
<td>Information Technology Competencies</td>
<td>6%</td>
<td>85.71%</td>
</tr>
</tbody>
</table>

As can be seen in the table, the assessment detected that the degree program covers over 85% of the competencies from each category. The categories with the greatest intensity are the Common Competencies, followed by the Computer Science and Software Engineering competencies. This order is consistent with the curriculum’s design objectives.
ASSESSMENT OF THE SET OF COMPULSORY COURSES (WITH THE ELECTIVE COURSES)

The assessment of the **Intensity of Coverage** also contemplates the elective courses corresponding to the Information Systems profile. This profile includes four elective courses particular to this area of knowledge and, when considering these courses, the total number of subjects evaluated in the degree program increases from thirty seven to forty one.

The following table shows that, when including these subjects, the Intensity of Coverage for the “Information Systems” competency category increases and the impact on the intensity of the remaining categories is slight.

<table>
<thead>
<tr>
<th>Category</th>
<th>Intensity of Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Competencies</td>
<td>21%</td>
</tr>
<tr>
<td>Computer Science Competencies</td>
<td>17%</td>
</tr>
<tr>
<td>Information Systems Competencies</td>
<td>11%</td>
</tr>
<tr>
<td>Software Engineering Competencies</td>
<td>18%</td>
</tr>
<tr>
<td>Computer Engineering Competencies</td>
<td>4%</td>
</tr>
<tr>
<td>Information Technology Competencies</td>
<td>12%</td>
</tr>
</tbody>
</table>

**SELF-ASSESSMENT PROCESS**

The self-assessment process consisted of the following stages:

1. **Creation of the work team.** The self-assessment was coordinated by two professors appointed as project leaders. They worked together with the lecturers and professors responsible for teaching the program’s subjects to collect the information.

2. **Preparation of the competency-course matrix.** In order to standardize and facilitate the collection of information on the development of the competencies for each subject, a matrix was formulated with the following fields (see example):
   - Subject name
   - List of competencies grouped by category
   - The fields established for each competency indicate:
     - The type of test conducted to assess the competency.
     - Examples or justifications for the competency assessment.
     - Level of compliance with the established criteria (**PD**, **D**).
     - Justification for the value of the criterion selected.
3. For each of the program’s subjects, the coordinators completed the corresponding matrix together with the professors responsible for teaching the course. This way of working minimized any competency interpretation issues and ensured the correct application of the established evaluation criteria. Filling in the data for one subject took approximately 50 minutes on average.

The matrix data was completed as follows:

In the columns for **Evidence of the competency assessment**, the coordinators were asked to check the box of the evaluated competencies for which evidence exists.

In the column **Justification**, the professors were asked to describe briefly the reason they thought the competency was being assessed.

The column **Level of development** was used to indicate the corresponding value according to the assessment criterion:

- 0 - (ND, Not developed). The competency is not assessed or developed.
- 1 - (PD, Partially Developed). Develops or contributes to the development of the competency, but does not assess it. Alternatively, the competency is assessed but not developed.
- 2 - (D, Developed). Develops or contributes to the development of the competency, evaluates it and

<table>
<thead>
<tr>
<th>Evidence of the competency assessment – Check the forms of competency assessment.</th>
<th>Development criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial</td>
<td>Compulsory</td>
</tr>
<tr>
<td>Common</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>Cn</td>
<td></td>
</tr>
<tr>
<td>Computer Science</td>
<td></td>
</tr>
<tr>
<td>CS1</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
</tr>
<tr>
<td>CSn</td>
<td></td>
</tr>
</tbody>
</table>
evidence exists that can be used by third parties for verification.

4. **Data integration.** Once the data-collection process was completed, the data was integrated into the *Course and Competency Matrix*.

The *Competency-course* matrix is a double-entry table in which the rows represent the competencies for each category, the columns represent the subjects in the curriculum and the cells represent the assessment values for the competencies of each subject. The values in the columns correspond to the values included in the data-collection template (see Figure – 1).

![Figure 2- Competency – Course Matrix](image)

5. **Calculation of metrics.** The *Percentage of Coverage* and *Intensity of Coverage* for each category were calculated based on the information entered in the *Competencies-courses* sheet.

**Calculation of Coverage of Competency**

1. For each competency in a category, in the intersection of the column “PD” (*Col. 1*) a (1) is entered if at least one subject is assessed as PD for this competency. In the intersection of the column “D” (*Col. 2*) a (1) is entered if at least one subject is assessed as D for this competency.
2. The number of ones (1) existing in the column (Col. 1) are calculated in the cell (Row 0, Col. 1), indicating the number of competencies for the categories assessed as PD.

The number of ones (1) existing in the column (Col. 2) are calculated in the cell (Row 0, Col. 2), indicating the number of competencies for the categories assessed as D.

In this example, the values are 2 and 23 respectively for the category Common Competencies.

3. The number of competencies in the category is entered into the column (Col. 3). In this example, the value is 25.

4. The Competency coverage based on the previous values is calculated in the column (Col. 4). In this case, coverage is 92% corresponding to (23/25*100).
Calculation of the **Intensity of Coverage**

1. For each competency of the category in the intersection with the column *(Col. 5)*, the number of subjects that assess the competency as **D** is entered. In this case, for competency C1, there are 19 subjects that are assessed as **D**.

2. The cell *(Row 0, Col.6)* calculates the product between the number of subjects in the program and the number of competencies within the category (total number of cells in relation to the number of competencies of the category and the number of subjects in the program). In this example, there are 925 cells (25 competencies * 37 subjects).
3. The cell *(Row 0, Col.5)* calculates the sum of the values obtained in step 1. In this example, there are 183 cells of the 925 possible cells with a value of (D).

<table>
<thead>
<tr>
<th>Row 0</th>
<th>Common Competencies</th>
<th>Course 1</th>
<th>Course 2</th>
<th>Course N</th>
<th>Total assessments (3) per category</th>
<th>Maximum number of assessments possible (2) per category</th>
<th>Intensity of coverage of the category</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>2</td>
<td>...</td>
<td>183</td>
</tr>
<tr>
<td>...</td>
<td>C2</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>2</td>
<td>...</td>
<td>25</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>2</td>
</tr>
</tbody>
</table>

4. Finally, the **intensity of coverage** is calculated on the basis of two previous values in column *(Col. 7)*. In this case, 20% (183/925*100).
<table>
<thead>
<tr>
<th>Row 0</th>
<th>Common Competencies</th>
<th>Course 1</th>
<th>Course 2</th>
<th>...</th>
<th>Course N</th>
<th>...</th>
<th>...</th>
<th>Col. 4</th>
<th>Col. 5</th>
<th>Col. 6</th>
<th>Col. 7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total assessments (2)</td>
<td>Maximum number of assessments possible (2)</td>
<td>Intensity of coverage of the category</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C1</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>2</td>
<td>...</td>
<td>1</td>
<td>19</td>
<td>925</td>
<td>20%</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>C2</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>0</td>
<td>2</td>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>C25</td>
<td>2</td>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
<td>2</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Curriculum Objectives

The general objective is to prepare systems engineers who recognize the need for ongoing education and are committed to ethical and professional responsibilities, with effective communication and multidisciplinary teamwork skills. Their comprehensive education will allow them to propose technological solutions that are consistent with their environment and society.

The objectives of this degree program include:
A. Training professionals who are capable of detecting their clients’ needs and of developing solutions that meet these needs.
B. Permitting the development of diverse technical and managerial professional profiles.
C. Providing a solid education to ensure that the graduate will have the capacity for self-education and adaptation to technological change, and the necessary knowledge to study a professional or academic postgraduate degree.
D. Developing skills that add value to graduates’ professional practice: rigor, teamwork, leadership, communication, trust, service and commitment.
E. Training professionals with the skills and knowledge needed to find a job rapidly.

The current curriculum maintains the characteristics of the former curriculum, focusing in particular on: *Maintaining and detailing the distinctive profile of the academic program in which software engineering is the unifying concept.*

1. Providing knowledge of the most common software application dominions.
2. Allowing students to develop their personal preferences.
3. Developing a solid theoretical-practical preparation that balances the practical application of knowledge, theory and the use of technology.

CONTRIBUTORS TO APPENDIX II

The following people collaborated on this report.

**General Coordinator**
Julio Fernández Dean of Academic Development

**Project Coordinators**
Gastón Mousqués Software Engineering Professor
Alejandro Adorjan Systems Engineering Academic Coordination Assistant

**Working Team**
Nora Szasz Academic Coordinator of the BSc in Systems Engineering
Alvaro Tasistro Theory of Computation Professor
Carlos Luna Theory of Computation Associate Professor
Luis Silva Business Administration Professor
REFERENCES

