Model for Characterizing and Creating Certificate Programs in Technology-Specific Areas

Carol L. Hoover
May 1996
CMU-CS-96-128

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Author's Internet Address: clh@cs.cmu.edu

Abstract

Rapid changes in technology have resulted in an increasing need to transfer state-of-the-art technical concepts to practicing technologists. These technical professionals and their employers seek in-depth educational opportunities to acquire advanced knowledge of emerging technologies; but often they cannot afford the time and money to pursue advanced degree programs. To satisfy industry's need to assimilate state-of-the-art concepts in shortened time spans, educational institutions have developed certificate programs. A certificate program is a set of one or more courses related to a particular technical area. Certificate programs are less extensive than degree programs. In this paper, we present a model for analyzing and creating certificate programs. We apply the model to the analysis of certificate programs offered by four schools in areas related to software engineering. Then we use our "meta-model" to generate a generic model for the creation of graduate certificate programs in the School of Computer Science at Carnegie Mellon University. Lastly, we discuss issues related to the creation of certificate programs.

This research was supported by the Master of Software Engineering (MSE) Program in the School of Computer Science at Carnegie Mellon University.

Keywords: certificate programs, graduate software engineering programs, professional education, software engineering education, technology-specific educational programs.

Table of Contents

1	**Introduction**	**1**
2	**Model Attributes and Attribute Values**	**1**
3	**Modeling Constraint Relations**	**5**
4	**Case Studies**	**8**
4.1	Georgia Tech College of Computing Certificate Programs	8
4.2	The University of British Columbia Certificate in Software Engineering	9
4.3	George Mason University Graduate Certificate in Software Systems Engineering	10
4.4	The Santa Clara Advanced Studies in Software Engineering Certificate Program	11
5	**Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University**	**12**
6	**Further Discussion of Issues**	**17**
7	**Summary**	**17**

Acknowledgments 17
References 18

Appendix A:	Georgia Tech College of Computing Certificate Programs	19
Appendix B:	The University of British Columbia Certificate in Software Engineering	20
Appendix C:	George Mason Graduate Certificate in Software Systems Engineering	21
Appendix D:	The Santa Clara University Advanced Studies in Software Engineering Certificate Program	22

List of Tables

Table 1:	TypeOfProgram Related to CreditAwarded	5
Table 2:	TypeOfProgram Related to Fees	5
Table 3:	TypeOfProgram Related to StudentRegistration	6
Table 4:	TypeOfProgram Related to StudentType	6
Table 5:	TypeOfProgram Related to TypeOfPrerequisites	7
Table 6:	RangeOfCertificate Related to RangeOfPrerequisites	7
Table 7:	TypeOfEvaluation Related to CreditAwarded	7
Table 8:	Georgia Tech College of Computing Certificate Programs	8
Table 9:	The University of British Columbia Certificate in Software Engineering	9
Table 10:	George Mason University Graduate Certificate in Software Systems Engineering	10
Table 11:	The Santa Clara Advanced Studies in Software Engineering Certificate Program	11
Table 12:	Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University	12
Table 13:	5-Semester Schedule of Classes for Graduate Certificate in Software Engineering	16
Table 14:	3-Semester Schedule of Classes for Graduate Certificate in Software Engineering	16

Model for Characterizing and Creating Certificate Programs in Technology-Specific Areas

1 Introduction

With rapid changes in technology, there is a need to transition state-of-the-art technical concepts to practicing technologists.[1] Some technical professionals seek short-term courses that brief them on specific concepts. Others want more technically in-depth training or technical education at an advanced level. Many professionals (and their employers) cannot take the time and money to pursue advanced degree programs. Hence, we have seen the development of certificate programs.

A technical certificate program consists of a set of one or more courses that relate to a particular technical area.[2] Certificate programs come in many flavors: some are continuing education programs with courses not awarded academic credit; and others are academic programs with courses taken for credit. Through our study of existing certificate programs, we have identified a set of distinguishing features that can be used to characterize certificate programs. We have specified a model that structures these features and their variations as attributes and attribute values.

This report describes our model and shows its application to the characterization of certificate programs offered by four different schools in areas related to software engineering. We then instantiate this "meta-model" to produce a generic model for the creation of graduate certificate programs within the School of Computer Science at Carnegie Mellon University. We present an example program, *The Carnegie Mellon University Graduate Certificate in Software Engineering Program*, that will be the first test case. Lastly, we expand our discussion of issues in creating certificate programs and present summary remarks.

2 Model Attributes and Attribute Values

The proposed model for characterizing certificate programs uses *attributes* and *attribute values* to identify the distinguishing features of a certificate program. *Instantiation* of the model is the process of specifying the values for each attribute. One can instantiate the model to characterize an existing certificate program or to create a new one. One can also generate a generic model used to create programs with common attribute values. *Table 12* outlines such a generic model in *Section 5: Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University*.

Attributes are instantiated as non-empty subsets of their attribute values sets. For instance, the attribute *CertificateRequirements* may be instantiated as {Required}, {Elective}, or {Required, Elective}. This means that certificate programs may include required, elective, or both required and elective courses. The attributes followed by asterisks should be instantiated only as singleton sets. An example is the *ProgramManagement** attribute which should be instantiated as {ContinuingEducationDepartment} or {DegreeGrantingDepartment}.[3] From a practical point of view, a certificate program is probably better administrated by one department! As one would expect, the attribute value *None* precludes the appearance of other attribute values in the instantiation of an attribute.

[1] Interested readers should note a recent issue of *Communications of the ACM* featuring articles that discuss the importance of computer science research in solving manufacturing problems. Transition of this research to industrial products and practices will likely involve the education of technologists [COM].

[2] Certificate programs differ from specialization tracks within degree programs which enable students to acquire domain-specific expertise via their elective courses [HOOVER].

[3] According to the guidelines of the International Association for Continuing Education and Training (IACET), the designations continuing education and degree-granting are mutually exclusive. Continuing education departments do not offer courses for academic credit nor do they grant degrees [IACET].

The subsection *Specification of Attributes and Attribute Values* presents and discusses each model attribute and its attribute values. The creator of a certificate program should become familiar with the contents of this subsection. Readers are encouraged to use the proposed model to define sample certificate programs or to characterize existing certificate programs. *Section 3: Modeling Constraint Relations* discusses realistic concerns which should be considered when creating certificate programs. Readers should check that the attribute values specified for each attribute are reasonable according to the modeling constraint relations discussed in *Section 3.*

Specification of Attributes and Attribute Values:

- TypeOfProgram* =
  {ContinuingEducation, AcademicCreditTerminal, AcademicCreditApplicable}

  ContinuingEducation -> Student does not receive academic credit for certificate courses.

  AcademicCreditTerminal -> Student receives academic credit for certificate courses but cannot apply them towards a degree program.

  AcademicCreditApplicable -> Student receives academic credit for certificate courses and can apply them towards a related degree program.

  Additional discussion of the differences between continuing education and academic credit certificate programs can be found in *Section 6: Further Discussion of Issues*.

- ProgramManagement*={ContinuingEducationDepartment, DegreeGrantingDepartment}

  This attribute characterizes the type of department that administrates the certificate program.

- RangeOfCertificate* =
  {SingleCourse, SetOfCourses}

  This attribute specifies whether a certificate is awarded for the completion of a single course or for the completion of a set of related courses.

- CertificateRequirements =
  {Required, Elective}

  Certificate programs can include required and or elective courses.

- AdmissionsCriteria =
  {None, CompletedApplicationForm, LetterOfIntent, Recommendations, References, Resume, Transcript(GPA), StandardizedTestScores, ParticularDegree, AcademicStatus, *<TypeOfPrerequisites>*}

  Admission decisions may be based upon completed application forms, letters of intent, recommendation forms or letters, references, resumes describing work experience, transcripts (and GPA's) describing academic background, standardized test scores, completion of particular degree programs, current academic status, and prerequisites. In the case of the *George Mason Graduate Certificate in Software Systems Engineering* described in *Subsection 4.3*, entering students are expected to have a bachelor's degree in a scientific or technical discipline or to have graduate status in a scientific or technical graduate program.

- TypeOfPrerequisites =
  {None, KnowledgeSkillsPractice, AcademicCoursework}

Prerequisites for certificate courses may be academic coursework in a particular area. Other courses may require knowledge of specific concepts, particular technical skills, or practical experience. Continuing education programs tend to specify prerequisites in the *KnowledgeSkillsPractice* format.

- RangeOfPrerequisites =

  {None, PerCourse, PerSetOfCourses}

  Prerequisites for a set of certificate courses may be part of the admission criteria. This may be reasonable if the student is expected to have requisite background before pursuing the certificate program. Each course in a certificate program may also have its own prerequisites. This would certainly be the case for sequential courses which would require that prior courses in a sequence be prerequisites.

- CreditAwarded* =

  {None, CEU's, AcademicCredits}

  IACET defines one CEU as ten contact hours of participation in an organized continuing education experience under responsible sponsorship, capable direction, and qualified instruction. The primary purpose of the CEU is to provide a permanent record of the educational accomplishments of an individual who has completed one or more significant non-credit educational experiences. IACET provides guidelines for administrating and awarding CEU's.

- TypeOfEvaluation =

  {None, PassFail, GradeLevels}

  PassFail -> criteria for distinquishing between Pass and Fail performances.

  GradeLevels -> (criteria for assigning grades) and (specification of which grade levels are passing for the certificate program).

- DesignationOfCompletion =

  {None, Certificate, OnTranscript}

  The attribute value *Certificate* refers to a paper document which indicates that the student has completed a certificate program of study. This document is distinct from an official transcript. Completed CEU's can be shown on a certificate as well as on an official transcript. Academic credits earned for completing certificate courses are shown on an official transcript. An official transcript can also include a designation that a certificate program has been successfully completed.

- CourseTimeSpan* =

  {Hours, Days, AcademicSchedule}

  Continuing education certificate courses are typically packaged as a variable number of hours or days of in-class activities. Academic courses which are part of a certificate program follow the normal academic schedule for classes with equivalent numbers of credits.

- Fees* =

  {PerCredit, PerCourse, PerSetOfCourses}

Fees for certificate courses may be charged per academic credit (academic credit certificate programs), per course, or as one price for the entire certificate program. It may be beneficial to give continuing education students an incentive in the form of a cost savings for completing an entire track of courses.

See the constraint relation *TypeOfProgram Related to Fees* for a discussion of the relationship between the type of program and fees.

- ProgramTimeLimit* =

  {None, Years}

  Students pursuing a certificate may or may not be required to finish their certificates within a specified number of years.

- Faculty =

  {WithinUniversity, OutsideUniversity}

  Faculty for certificate courses may be recruited from within or outside the university. Some schools employ qualified experts from industry to teach certificate courses.

- Location =

  {OnCampus, OffCampus, Remote}

  Certificate courses are typically held on campus and use academic departmental resources. Continuing education departments may provide their own training facilities on and off campus. Remote refers to course delivery via satellite broadcasts, networked communications, video, or other media which enables the student to learn about the material without direct contact to the instructor. In her article about trends in technical education, Mead states that there is an increasing shift from instructor-led classes to other delivery mechanisms. She predicts an increase in remote course delivery [MEAD].

- Course Availability =

  {Weekdays, Evenings, Weekends}

  Certificate courses may be offered during weekdays, evenings, or weekends.

- StudentType =

  {ShortTerm, PartTime, FullTime}

  Short-term students take courses that span several hours or days. Students may take courses whose time frames span a normal academic schedule (e.g. quarters or semesters) on a part-time or full-time basis.

- StudentRegistration =

  {NonRegistered, RegisteredNonDegree, InDegreeProgram}

  NonRegistered -> Student is taking courses which will not appear on an official transcript. The student does not complete the normal university registration forms.

  RegisteredNonDegree -> Student is taking courses which will appear on an official transcript but is not pursuing a degree program. The student (or an agent) must complete the normal university registration forms.

InDegreeProgram -> Student is taking courses which will appear on an official transcript and is pursuing a degree program. The student (or an agent) must complete the normal university registration forms.

3 Modeling Constraint Relations

When selecting attribute values for the model attributes, one should consider the relationship between attributes. Some attribute values are not compatible. This means that they should not be used in conjunction with each other. For instance, by definition continuing education certificate program do not offer academic credit for completed courses. Likewise, academic credit programs do not offer CEU's. The constraint relation can be expressed as shown in the following table.

Table 1: TypeOfProgram Related to CreditAwarded

TypeOfProgram x CreditAwarded	**None**	**CEU's**	**AcademicCredit**
ContinuingEducation	X	X	
AcademicCreditTerminal			X
AcademicCreditApplicable			X

Below are some suggested constraints that can be applied to specific attribute pairs. Each table displays attribute values that are compatible (can be reasonably used in conjunction). Attribute pairs that are fully compatible (represented by completely filled tables) have no constraints and therefore need not be listed. Readers are encouraged to suggest other attribute pairs that have attributes values that should not be used in conjunction.

- TypeOfProgram Related to Fees: $\mathrm{TypeOfProgram} \times \mathrm{Fees}$

  By definition, continuing education programs do not award credits and therefore do not charge per the number of credits taken. Continuing education programs generally charge per course or per set of related courses (possibly with a discount for taking an entire set of certificate courses).

  Academic credit programs charge per credit (or per some other basic unit associated with a credit). A student can determine a cost per course using the number of credits associated with each target course, but this calculation may not hold if the fee structure includes an upper-limit for full-time students. In other words, full-time students may pay at most the upper limit regardless of how many credits they take beyond the required number to maintain their full-time status. Likewise, this analogy applies to determining the fee for a set of courses.

  It might be worthwhile to study the issue of whether or not students would be more likely to complete a certificate program if there is a fee incentive for doing so.

Table 2: TypeOfProgram Related to Fees

TypeOfProgram x Fees	**PerCredit**	**PerCourse**	**PerSetOfCourses**
ContinuingEducation		X	X

Table 2: TypeOfProgram Related to Fees

TypeOfProgram x Fees	PerCredit	PerCourse	PerSetOfCourses
AcademicCreditTerminal	X	X (*possibly applicable only to part-time students*	X (*possibly applicable only to part-time students*
AcademicCreditApplicable	X	X (*possibly applicable only to part-time students*	X (*possibly applicable only to part-time students*

- TypeOfProgram Related to StudentRegistration: $\mathrm{TypeOfProgram} \times \mathrm{StudentRegistration}$

  Credit for continuing education courses in the form of CEU's may or may not be recorded on an official transcript. If the credits will appear on an official transcript, the student is registered; otherwise, the student need not be registered. Students earning academic credits for certificate courses obviously must be registered. Students working on a terminal certificate program may also be concurrently pursuing a degree program and therefore be registered as a student in a degree program.

Table 3: TypeOfProgram Related to StudentRegistration

TypeOfProgram x StudentRegistration	NonRegistered	RegisteredNonDegree	InDegreeProgram
ContinuingEducation	X	X	
AcademicCreditTerminal		X	X
AcademicCreditApplicable		X	X

- TypeOfProgram Related to StudentType: $\mathrm{TypeOfProgram} \times \mathrm{StudentType}$

  Continuing education courses typically are offered on a short-term basis. Certificate courses taken for academic credit generally follow the normal academic schedule for classes. Usually, students can take these courses on a part-time or full-time basis.

Table 4: TypeOfProgram Related to StudentType

TypeOfProgram x StudentType	ShortTerm	PartTime	FullTime
ContinuingEducation	X	X	
AcademicCreditTerminal		X	X
AcademicCreditApplicable		X	X

- TypeOfProgram Related to TypeOfPrerequisites: $\text{TypeOfProgram} \times \text{TypeOfPrerequisites}$

  Continuing education programs generally specify prerequisites in the form of knowledge, skills, and practice. Courses listed as prerequisites may be other continuing education courses. Certificate programs which offer courses for academic credit often require background related to the content of specific academic courses. Some programs may even require that the student hold a particular degree (such as a degree in a scientific or technical field).

Table 5: TypeOfProgram Related to TypeOfPrerequisites

TypeOfProgram x TypeOfPrerequisites	**None**	**Knowledge SkillsPractice**	**Academic Coursework**
ContinuingEducation	X	X	
AcademicCreditTerminal		X	X
AcademicCreditApplicable		X	X

- RangeOfCertificate Related to RangeOfPrerequisites: $\text{RangeOfCertificate} \times \text{RangeOfPrerequisites}$

  It should be obvious that prerequisites for a single-course certificate are for that course.

Table 6: RangeOfCertificate Related to RangeOfPrerequisites

RangeOfCertificate x RangeOfPrerequisites	**PerCourse**	**PerSetOfCourses**
SingleCourse	X	
SetOfCourses	X	X

- TypeOfEvaluation Related to CreditAwarded: $\text{TypeOfEvaluation} \times \text{CreditAwarded}$

  It is likewise obvious that no credit should be awarded if there is no evaluation of a student's work or participation in a certificate course. Likewise, continuing education courses with CEU credit are usually evaluated on a PassFail basis. Sometimes, Pass simply means that the student participated in some way in every session.

Table 7: TypeOfEvaluation Related to CreditAwarded

TypeOfEvaluation x CreditAwarded	**None**	**CEU's**	**AcademicCredit**
None	X		
PassFail		X	X
GradeLevels			X

4 Case Studies

This section includes four case studies discussed in the following order: (1) Georgia Tech College of Computing Certificate Programs, (2) The University of British Columbia Certificate in Software Engineering, (3) George Mason University Graduate Certificate in Software Systems Engineering, and (4) The Santa Clara University Advanced Studies in Software Engineering (ASSE) Certificate Program. The first two schools listed above offer continuing education programs in computing specialty areas, and the following two offer graduate certificate programs in software engineering.

4.1 Georgia Tech College of Computing Certificate Programs

The Georgia Tech College of Computing offers certificate programs in software engineering, object-oriented development, and graphical user interface (GUI) development. Each program consists of short-term courses for working professionals. The goal of these certificate programs is to provide professional development in computing for executives, managers, end users, and programmers in business and industry.

All three certificate programs are structurally the same with respect to the model presented in *Section 2*. *Table 8* shows the model attributes and corresponding values that characterize these programs. No specific information is given about admissions criteria for the certificate programs. Prerequisite knowledge and skills are specified for the certificate courses to guide the student in course selection. Readers should refer to *Appendix A: Georgia Tech College of Computing Certificate Programs* for further information [GIT] [WGIT]. The Georgia Tech Department of Continuing Education also supports an electronic mailing list that provides members announcements about new courses. Readers should write to *conted-info@cc.gatech.edu* for more information.

Table 8: Georgia Tech College of Computing Certificate Programs

Certificate Program Attribute	Value
Type of Program	Continuing Education *(Software Engineering Certificate,* *Object-Oriented Development Certificate,* *and Graphical User Interface Certificate)*
Program Management	Continuing Education Department *(in cooperation with the College of Computing)*
Range of Certificate	Set of Courses
Certificate Requirements	Required and Elective Courses
Admissions Criteria	None Listed
Type of Prerequisites	Knowledge Skills Practice
Range of Prerequisites	Per Course
Credit Awarded	CEU's
Type of Evaluation	Pass Fail
Designation of Completion	Certificate and On Transcript
Course Time Span	Days
Fees	Per Course
Program Time Limit	None
Faculty	Within University and Outside University

Table 8: Georgia Tech College of Computing Certificate Programs

Certificate Program Attribute	Value
Location	On Campus and Off Campus
Course Availability	Weekdays
Student Type	Short Term
Student Registration	Registered Non Degree *(Official transcript available upon request.)*

4.2 The University of British Columbia Certificate in Software Engineering

The University of British Columbia Certificate in Software Engineering focuses on the software engineering process. Developed for people involved in planning, developing, testing, and managing software systems, the program presents a practical, multi-disciplinary systems approach to software engineering. The program was developed to meet the needs of a diverse audience of software professionals as well as the needs of the growing and important software industry in Canada. As such, courses are offered on a part-time basis and, in some cases, will be offered via remote access. *Table 9* shows the model attributes and corresponding values that characterize this program. Readers should refer to *Appendix B: The University of British Columbia Certificate in Software Engineering* for further information [UBC] [WUBC].

Table 9: The University of British Columbia Certificate in Software Engineering

Certificate Program Attribute	Value
Type of Program	Continuing Education
Program Management	Continuing Education Department *(UBC Continuing Studies in Computer Science Programs and UBC Continuing Studies in Engineering and Architecture)*
Range of Certificate	Set of Courses
Certificate Requirements	Required and Elective Courses
Admissions Criteria	Completed Application Form, Letter Of Intent, References, Resume, Transcript (GPA), and Academic Coursework *(Submission of a technical writing example is optional.)*
Type of Prerequisites	Knowledge Skills Practice or Academic Coursework
Range of Prerequisites	Per Set of Courses
Credit Awarded	CEU’s
Type of Evaluation	Pass Fail
Designation of Completion	Certificate
Course Time Span	Hours
Fees	Per Course

Table 9: The University of British Columbia Certificate in Software Engineering

Certificate Program Attribute	Value
Program Time Limit	Years *(All requirements must be completed within four years of admission.)*
Faculty	Within University and Outside University
Location	On Campus, Off Campus, and Remote
Course Availability	Weekdays, Evenings, and Weekends
Student Type	Part Time
Student Registration	Non Registered

4.3 George Mason University Graduate Certificate in Software Systems Engineering

The George Mason University Graduate Certificate in Software Systems Engineering is designed for persons who are involved in the development and modification of larger complex computer software systems. Originally, the program was designed for people holding master's degrees in a scientific or technical field and for students currently enrolled in a scientific or technical master's program. The Department of Information and Software Systems Engineering reports that there has been significant interest in the certificate program by people who hold bachelor's degrees and who do not want to pursue master's degrees. As a result, the department has changed its admission criteria to admit appropriate scientific or technical bachelor's degree holders with GPA's of 3.0/4.0 or better and adequate GRE/GMAT scores. Students are permitted to acquire course prerequisites upon admission but prior to taking the certificate courses. *Table 10* shows the model attributes and corresponding values that characterize this program. Readers should refer to *Appendix C: George Mason University Graduate Certificate in Software Systems Engineering* for further information [GMU] [WGMU].

Table 10: George Mason University Graduate Certificate in Software Systems Engineering

Certificate Program Attribute	Value
Type of Program	Academic Credit Applicable
Program Management	Degree Granting Department *(Department of Information and Software Systems Engineering)*
Range of Certificate	Set Of Courses
Certificate Requirements	Required and Elective Courses
Admissions Criteria	(Transcript (GPA) and Standardized Test Scores) or Academic Status
Type of Prerequisites	Academic Coursework or Knowledge Skills Practice
Range of Prerequisites	Per Set of Courses
Credit Awarded	Academic Credits
Type of Evaluation	Grade Levels

Table 10: George Mason University Graduate Certificate in Software Systems Engineering

Certificate Program Attribute	Value
Designation of Completion	Certificate and On Transcript
Course Time Span	Academic Schedule
Fees	Per Credit
Program Time Limit	Years *(All requirements must be completed within six years of admission.)*
Faculty	Within University and Outside University
Location	On Campus and Remote
Course Availability	Weekdays
Student Type	Part Time and Full Time
Student Registration	Registered Non Degree and In Degree Program

4.4 The Santa Clara University Advanced Studies in Software Engineering (ASSE) Certificate Program

The Santa Clara Advanced Studies in Software Engineering Certificate Program emphasizes methodologies used during the development of large and complex software systems. The program is designed for industrial people who are developing such software systems or who are leading software development projects. People who have had little or no experience with recent advances in software development methodology can upgrade their knowledge and skills via this program. Courses are offered at the graduate level and can be applied towards a master's degree. Some courses are offered at industrial sites. *Table 11* shows the model attributes and corresponding values that characterize this program. Readers should refer to *Appendix D: The Santa Clara Advanced Studies in Software Engineering Certificate Program* for further information [WUSC].

Table 11: The Santa Clara Advanced Studies in Software Engineering Certificate Program

Certificate Program Attribute	Value
Type of Program	Academic Credit Applicable
Program Management	Degree Granting Department *(Computer Engineering Department)*
Range of Certificate	Set of Courses
Certificate Requirements	Required and Elective Courses
Admissions Criteria	Transcript (GPA), Knowledge Skills Practice, and Academic Coursework
Type of Prerequisites	Academic Coursework
Range of Prerequisites	Per Set of Courses and Per Course
Credit Awarded	Academic Credits

Table 11: The Santa Clara Advanced Studies in Software Engineering Certificate Program

Certificate Program Attribute	Value
Type of Evaluation	Grade Levels
Designation of Completion	Certificate and On Transcript
Course Time Span	Academic Schedule
Fees	Per Credit
Program Time Limit	None
Faculty	Within University and Outside University
Location	On Campus, Off Campus, and Remote *(Remote access is planned for the future.)*
Course Availability	Weekdays
Student Type	Part Time and Full Time
Student Registration	Registered Non Degree *(non-matriculating)*

5 Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University

The generic model described below is a prototype model for creating graduate certificate programs within the School of Computer Science at Carnegie Mellon University.

A graduate certificate program would be geared towards professionals who want to pursue academic studies in particular technical areas at an advanced level. Certificate programs would be designed for professionals pursuing part-time, graduate-level education; but students with appropriate academic backgrounds who are currently enrolled in degree programs at Carnegie Mellon University could also pursue these programs. Since graduate courses require students to have requisite academic background, it is reasonable that prerequisites for graduate certificate programs would be undergraduate coursework or equivalent knowledge. Students entering these programs would be expected to satisfy the prerequisites before entering a certificate program. Enabling students to apply completed certificate courses to master's programs in the same technical areas could be an incentive for qualified students to pursue the master's degree.

Table 12 describes the proposed generic model. The example instantiation of the model, *The Carnegie Mellon University Graduate Certificate in Software Engineering*, will be offered in Fall 1996.

Table 12: Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University

Certificate Program Attribute	Value
Type of Program	Academic Credit Applicable or Academic Credit Terminal *(The Carnegie Mellon University Graduate Certificate in <name of technical area>)*
Program Management	Degree Granting Department *(School of Computer Science)*

Table 12: Generic Model for Graduate Certificate Programs Within the School of Computer Science at Carnegie Mellon University

Certificate Program Attribute	**Value**
Range of Certificate	Set of Courses
Certificate Requirements	Required Courses
Admissions Criteria	Completed Application, Letter of Intent, Recommendations, Resume, Transcript (GPA), Particular Degree or Academic Status, and *<prerequisites for technical area of study>*
Type of Prerequisites	Knowledge Skills Practice or Academic Coursework
Range of Prerequisites	Per Set of Courses and Per Course
Credit Awarded	Academic Credits
Type of Evaluation	Grade Levels
Designation of Completion	Certificate and On Transcript
Course Time Span	Academic Schedule
Fees	Per Credit
Program Time Limit	None
Faculty	Within University and Outside University
Location	On Campus and Remote *(Remote to be offered in the future.)*
Course Availability	Weekdays
Student Type	Part Time and Full Time
Student Registration	Registered Non Degree and In Degree Program

Basic Format: Graduate courses.

Intended Audience:

- Professionals who want to pursue academic studies in particular technical areas at an advanced level.
- Students with appropriate academic backgrounds who are currently enrolled in degree programs at Carnegie Mellon University (e.g. upper-level undergraduate students in CS).

Purpose of the Graduate Certificate Program: List of educational goal(s) for the graduate certificate program.

Example: Purpose for the Carnegie Mellon University Graduate Certificate in Software Engineering

To help the student acquire the foundation for learning and practicing the discipline of state-of-the-art software engineering with respect to four intellectual themes:

- Engineering design and system architecture.
- Mathematical modeling, abstraction, and reasoning.

- Technical management and evolution of large, long-lived software.
- Human resource management.

Required courses for the Graduate Certificate in Software Engineering are the core courses for the Master of Software Engineering Program offered in the School of Computer Science at Carnegie Mellon University. The four intellectual themes described here are referenced from the master's program brochure [MSE]. Additional information regarding the core courses can be found in [GARLAN].

Contact Points:

- ...@cs.cmu.edu
- http://www.cs.cmu.edu/...
- Address:
  Carnegie Mellon University
  School of Computer Science
  Graduate Certificate Programs or
  Graduate Certificate Program in *<name of technical area>*
  5000 Forbes Avenue
  Pittsburgh, Pennsylvania 15213-3890 USA
- Telephone: 412-268-*XXXX*
- Fax: 412-268-*XXXX*

<name of technical area> ::= Software Engineering
| Speech Recognition
| Distributed and Multi-media Systems
| other technical area

Admission Criteria:

- Bachelor's degree or upper-level undergraduate status in Carnegie Mellon University degree program of study related to *<name of technical area>* with a GPA equivalent to 3.0/4.0 or better.
- English competency as demonstrated by the TOEFL for applicants whose first language is not English.
- GRE general and subject test.
- Demonstrated competency in *<prerequisites for technical area of study>*. (See the *Prerequisites Specification Grammar* for generating text that describes the prerequisites.)

Application including:

- Letter of intent
- Official transcripts (academic records)
- Letters of recommendation
- Resume

Prerequisites Specification Grammar:

<prerequisites for technical area of study> ::= Prerequisites for the Carnegie Mellon University
Graduate Certificate in *<name of technical area>*:
<bullet> <coursework for technical area of study>
<bullet> <knowledge>
<bullet> <skills>
<bullet> <practice>
| Prerequisites for the Carnegie Mellon University
Graduate Certificate in *<name of technical area>*: None

<bullet>::= carriage return
•

| *empty string*
<coursework for technical area of study> ::= list of prerequisite courses
| Coursework in: *<subject areas>*
| *empty string*
<subject areas> ::= list of subject areas
<knowledge> ::= Knowledge of : *<list of prerequisite concepts>*
| *empty string*
<list of prerequisite concepts> ::= list of prerequisite concepts
<skills> ::= Skill in: *<list of prerequisite skills>*
| *empty string*
<list of prerequisite skills> ::= list of prerequisite skills
<practice> ::= Experience in: *<practical experience>*
| *empty string*
<practical experience> ::= list of practical experience areas

Example: Prerequisites for the Carnegie Mellon University Graduate Certificate in Software Engineering:

- Coursework in: data structures and algorithms, discrete mathematics, and programming languages.
- Knowledge of: high-level programming language, software development life cycle, and logic.
- Experience in: developing software systems.

Course Requirements Specification Grammar: (Used to generate text describing course requirements.)
<course requirements> ::=
Course Requirements for the Carnegie Mellon University Graduate Certificate in *<name of technical area>*: NoTotalUnits total units
<bullet> *<required>* units of required courses including : *<list of required courses>*
<bullet> *<elective>* units of elective courses from: *<list of elective courses>*
| Course Requirements for the Carnegie Mellon University Graduate Certificate in *<name of technical area>*: NoTotalUnits total units
<bullet> *<required>* units of required courses including : *<list of required courses>*
| Course Requirements for the Carnegie Mellon University Graduate Certificate in *<name of technical area>*: NoTotalUnits total units
<bullet> *<elective>* units of elective courses from: *<list of elective courses>*

<bullet> *<required>* units of required courses including : *<list of required courses>*
<bullet> *<elective>* units of elective courses from: *<list of elective courses>*
<bullet> ::= carriage return
•
| *empty string*
<required> ::= {x : (x is an integer) and (x + *<elective>* = NoTotalUnits)}
<elective> ::= {y : (y is an integer) and (*<required>* + y = NoTotalUnits)}
<list of required courses> ::= bulleted list of regularly offered graduate courses
<list of elective courses> ::= bulleted list of frequently offered graduate courses

Example: Course Requirements for the Carnegie Mellon University Graduate Certificate in Software Engineering: 60 total units

- 60 units of required courses including:
  - Models of Software Systems (Fall 1 - 12 units)
  - Methods of Software Development (Fall 1 - 12 units)
  - Management of Software Development (Fall 2 - 12 units)
  - Analysis of Software Artifacts (Spring - 12 units)
  - Architectures of Software Systems (Spring - 12 units)

Course Availability and Facilities:

- On-campus facilities used for regularly offered graduate courses.
- Remote access to graduate courses in the future.

Graduation Criteria: (B- or better in all certificate courses) and a (GPA of 3.0/4.0 or better for all certificate coursework). This permits the student to balance a B- in a particular certificate course with a B+ or better in another certificate course.

Length of Program:

- $\lceil (\mathrm{NoTotalUnits})/X \rceil$ semesters (X units per semester) assuming that there are X units of properly sequenced courses available per semester.
- Students should check course prerequisites to determine the sequence in which courses can be taken.

Example: Length of Program for the Carnegie Mellon University Graduate Certificate in Software Engineering

Course Dependencies:

- Models of Software Systems: If possible, Methods of Software Development should be taken concurrently.
- Methods of Software Development: If possible, this course should be taken concurrently or after Models of Software Systems and Management of Software Development.
- Analysis of Software Artifacts: Models of Software Systems is a prerequisite.

Two example schedules are shown below. Other schedules are feasible.

$\lceil 60/12 \rceil = 5$ semesters as a part-time student (12 units per semester).

Table 13: 5-Semester Schedule of Classes for Graduate Certificate in Software Engineering

Fall 1	Spring 1	Fall 2	Spring 2	Fall 3
Models	Architectures	Management	Analysis	Methods

$\lceil 60/24 \rceil = 3$ semesters as a part-time student (24 units per semester).

Table 14: 3-Semester Schedule of Classes for Graduate Certificate in Software Engineering

Fall 1	Spring	Fall 2
Models	Architectures	Methods
Management	Analysis	

Coordination of Certificate and Degree Programs:

- Successfully completed certificate courses may be applied towards a master's degree in *<name of technical area>* if such a master's program exists (e.g. Successfully completed certificate courses in the Carnegie Mellon University Graduate Certificate in Software Engineering may be applied to the Master of Software Engineering degree).
- Applicant must follow normal admission procedures for the master's program.

6 Further Discussion of Issues

The creation of an academic program of study involves resolution of issues. Foremost is setting the educational goals and target audience for the program. When an academic department plans to offer both a certificate program and a degree program in the same technical area, the objectives of both programs must be clearly differentiated and coordinated. The obvious differentiator is the extent to which the technical area is covered. A degree program usually offers more breadth of coverage and possibly more depth as well. Another differentiator might be the admissions criteria. The admissions criteria for a certificate program may or may not be less extensive than that for a degree program.

Offering a certificate program for academic credit as well as a certificate program for CEU credit in the same technical area may be confusing to prospective applicants and employers. We recommend that educational objectives clearly specify and delineate the target knowledge and skills for each program. Educational objectives for either program should be behavioral in the sense that they describe observable behavior to be exhibited by students who successfully complete the certificate program. Educational objectives for continuing education programs are typically less extensive and require fewer prerequisites than those for programs offering academic credit. But both types of programs should include mechanisms for evaluating and reporting whether or not a student has successively achieved the educational objectives. Pass/fail evaluations may be most appropriate for continuing education courses.

Clarifying the educational objectives as well as the evaluation and reporting mechanisms would help employers better understand the value of continuing education courses. Likewise, if both academic and CEU credit are to be reported on an official transcript, the distinction between the two must be clear. Registration is a related issue. Awarding CEU's or academic credits for certificate courses requires that the student be registered with the university's registrar. Since certificate programs are non-degree programs, there needs to be a special status for students pursuing certificate programs that will be awarded some type of recorded credit.

Another issue is the coordination between certificate programs for academic credit and degree programs. Should a student who completes a certificate program for academic credit be able to apply certificate courses that overlap with degree program courses to a degree program? If so, should the certificate student be given any special consideration when applying to the degree program? Students with GPA's of 3.5/4.0 or better in the George Mason Software Systems Engineering Certificate Program need not take the GRE's when applying for the master's program in the same area. Likewise, can the student apply for admission to a degree program while completing a certificate program; or must the application be done after the certificate program is completed? Readers are encouraged to identify other issues that relate to the type of certificate programs being considered.

7 Summary

In this paper, we discussed a model for characterizing the distinguishing features of a certificate program. We applied the model to the analysis of certificate programs offered by four schools in areas related to software engineering. We also used our "meta-model" to generate a generic model for graduate certificate programs in the School of Computer Science at Carnegie Mellon University. We applied this generic model to the definition of a new graduate certificate program in software engineering to be offered at Carnegie Mellon University in fall 1996. We did not show that our "meta- model" is complete. In fact, readers might use the ideas presented in this paper to construct their own "meta-level" models. Certainly readers should tailor constraint relations to match the objectives of their target certificate programs. Most importantly, we hope that we challenged readers to think critically about why and how they construct academic programs of study, in particular certificate programs.

Acknowledgments:

The author would like to acknowledge Professor Mary Shaw (Associate Dean for Professional Programs) for her idea to create certificate programs in the School of Computer Science at Carnegie Mellon University. Dr. James Tomayko (a Principal Lecturer in the School of Computer Science) contributed his viewpoints regarding how these programs could be organized; and Phyllis Lewis (Administrator for the Master of Software Engineering Program in the School of Computer Science) helped with reference materials. The author would also like to thank the following people for the information that they provided about certificate programs being offered at their schools: Sobia Fayyaz from the George Mason Department of Information and Software Systems Engineering, Jane Hutton (Division Director, Career and Corporate Programs) of the University of British Columbia Continuing Studies Division, Professor Samiha Mourad (Associate Dean) from the University of Santa Clara Electrical Engineering Department in conjunction with the Computer Engineering Department, and Diane Turner from the Georgia Tech Continuing Education Department.

References:

[COM] Communications of the ACM (*Computer Science in Manufacturing Issue*). Feb. 1996, Vol. 39, No. 2, ACM, New York, NY.

[GARLAN] David Garlan, Alan Brown, Daniel Jackson, Jim Tomayko, and Jeannette Wing. The CMU Master of Software Engineering Core Curriculum. In Proc. of the 8th SEI CSEE Conf. at New Orleans, LA, Mar. 29-Apr. 1, 1995, Springer-Verlag, Berlin, pp. 65-86.

[GIT] Georgia Tech College of Computing Certificate Programs and Individual Courses brochure. Georgia Institute of Technology, Department of Continuing Education, Atlanta, GA, 1995.

[GMU] The Graduate Certificate in Software Systems Engineering program brochure. George Mason University, Department of Information and Software Systems Engineering, Fairfax, VA, 1996.

[HOOVER] Carol L. Hoover, Mary Shaw, and Nancy R. Mead. The Carnegie Mellon University Master of Software Engineering Specialization Tracks. In Proc. of the 9th SEI CSEE Conf. at Daytona Beach, FA, April 22-24, 1996, IEEE, New York, NY, pp. 100-118.

[IACET] The Continuing Education Unit Criteria and Guidelines. International Association of Continuing Education and Training, Washington, DC, 1991.

[MEAD] Nancy R. Mead. Education Trends and Their Impact on Management of Software Engineering Education. In Proc. of the 8th SEI CSEE Conf. at New Orleans, LA, Mar. 29-Apr. 1, 1995, Springer-Verlag, Berlin, pp. 419-428.

[MSE] Master of Software Engineering program brochure. Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, 1995.

[UBC] UBC Certificate in Software Engineering program handbook. The University of British Columbia Continuing Studies, Vancouver, B.C. Canada, 1996.

[WGIT] De Duff (de@cc.gatech.edu). Contents of http://www.cc.gatech.edu/conted/ and linked web pages relating to the Georgia Tech College of Computing Certificate Programs and Individual Courses. Georgia Institute of Technology, Department of Continuing Education, Atlanta, GA, 1995.

[WGMU] Jeff Offutt (webmaster@isse.gmu.edu). Contents of http://isse.gmu.edu/ and linked web pages relating to the Graduate Certificate in Software Systems Engineering program. George Mason University, Department of Information and Software Systems Engineering, Fairfax, VA, 1996.

[WUBC] Contents of http://www.cstudies.ubc.ca/softeng and linked web pages relating to the UBC Certificate Program in Software Engineering. The University of British Columbia Continuing Studies, Vancouver, B.C. Canada, 1996.

[WUSC] dlewis@scu.edu. Contents of http://pcsel10.scu.edu/coen/softeng.html and linked web pages relating to the Advanced Studies in Software Engineering (ASSE) Certificate Program. Santa Clara University, Department of Computing Engineering, Santa Clara, CA, 1996.

Appendix A Georgia Tech College of Computing Certificate Programs

Basic Format: Short-term courses for working professionals.

Contact Points:
- conted-info@cc.gatech.edu
- http://www.cc.gatech.edu/conted/
- Address:
  Georgia Institute of Technology
  Department of Continuing Education
  P.O. Box 93686
  Atlanta, Georgia 30332-0385
- Telephone: 404-894-2400
- Fax: 404-894-8925

Prerequisites:
The basic prerequisite for the required courses in the Software Engineering Certificate is experience developing software systems and knowledge about the software development process. The primary prerequisite for the Object-Oriented Development Certificate is familiarity with some high-level language.

Course Requirements: Software Engineering Certificate
- Improving Software Quality (3 days)
- Project Management (3 days)
- Software Testing (2 days)
- 6 days of other College of Computing continuing education courses

Course Requirements: Object-Oriented Development Certificate
- Introduction to Object-Oriented Programming (2 days)
- Object-Oriented Analysis and Design (3 days)
- Object-Oriented Programming Using C++ (4 days)
- 6 days of other College of Computing continuing education courses

Course Requirements: Graphical User Interface Development Certificate Program
- Human Computer Interaction (2 days)
- User-Centered Interface Engineering (2 days)
- Using the X Window System (1 day)
- Programming With the X Window System and Motif (2 days)
- 6 days of other College of Computing education courses

Other Related Continuing Education Courses:
- Computer-Aided Software Engineering (2 days)
- Designing User Interfaces (2 days)
- Database Design (4 days)
- Introduction to C Language (3 days)
- Intermediate C Language (3 days)
- Introduction to OpenGL Programming (2 days)
- Visual Basic (3 days)

Course Availability and Facilities:
- Weekdays
- 2-4 day courses
- Department of Continuing Education Training Facility
- Sheraton Colony Square Complex

Faculty: (Faculty have Ph.D. and/or master's degrees. Some faculty have significant industrial experience.)
- Faculty from College of Computing
- Research scientists from College of Computing
- Principals from Carnegie Technology Group Inc. in Atlanta

Fees (as of April 1996):
- 2 day course - $695
- 3 day course - $995
- 4 day course - $1295

Appendix B The University of British Columbia Certificate in Software Engineering

Basic Format: Short-term courses for working professionals.

Contact Points:
- compureg@cce.ubc.ca
- http://www.cstudies.ubc.ca/softeng
- Address:
  UBC Certificate in Software Engineering
  The University of British Columbia
  Continuing Studies, 5997 Iona Drive
  Vancouver, B.C. V6T 1Z1
- Telephone: 604-822-1420
- Fax: 604-822-1499

Admission Criteria: Completed application including the following details.
- Resume indicating educational achievements and work/volunteer experience
- Letter of intent
- Three references
- Technical report (optional)

Prerequisites:
- Reasonable communication skills and English proficiency
- Introductory knowledge of computer programming, computer systems, and discrete mathematics

Course Requirements:

99 classroom hours of core courses:
- An Introduction to Software Engineering (6 hours)
- The Software Engineering Process (18 hours)
- Requirements Analysis and Specification (15 hours/3 opt. lab hours)
- Software Architecture and Design (12 hours)
- System/Software Testing (12 hours)
- Software Project Management (12 hours)
- Software Engineering Team Project (24 hours)

Minimum of 51 classroom hours of elective courses:
- Software Quality Assurance (12 hours)
- Software Configuration Management (6 hours)
- Computer-Human Interfaces and GUIs (9 hours/3 opt. lab hours)
- Object-Oriented Techniques (12 hours)
- Data Engineering and Databases (12 hours)
- Comparative Programming Languages (9 hours)
- Real-Time Systems (6 hours)
- Software Engineering Tools (6 hours/6 opt. lab hours)
- Advanced Topics (6 hours)

Course Availability and Facilities:
- Weekdays, Evenings, and Weekends
- 2-6 day or parts-of-day courses
- Off-campus locations
- Remote access to courses is planned for the future.

Faculty:
- Faculty have Ph.D. and/or master’s degrees.
- Some faculty have significant industrial experience.

Graduation Criteria:
- Pass or completion in elective and required courses
- 60% correct in all courses graded numerically

Appendix C George Mason Graduate Certificate in Software Systems Engineering

Basic Format: Graduate courses

Contact Points:

- http://isse.gmu.edu/
- Address:
  George Mason University
  Department of Information and Software Systems Engineering
  4400 University Drive
  Fairfax, Virginia 22030-4444
- Telephone: 703-993-1640

Admission Criteria:

- Bachelor's degree with GPA of 3.0/4.0 or higher and (GRE's or GMAT's) OR
- Graduate status in a scientific or technical graduate program

Prerequisites: Knowledge equivalent to undergraduate courses in the following areas.

- Block-structured programming languages
- Data structures and algorithms
- Machine organization
- Discrete mathematics

Course Requirements: 15 semester credits of graduate study from the following set of courses

- Software Construction
- Software Requirements and Prototyping
- Software Design
- Formal Methods and Models in Software Engineering
- Software Project Management OR Software Testing and Quality Assurance

Course Availability and Facilities:

Regularly offered graduate courses and departmental facilities available for these courses.

Faculty:

- Departmental faculty and adjunct faculty from industry.
- All faculty have Ph.D. degrees.
- Some faculty have significant industrial experience.

Graduation Criteria:

B or better in all courses

Coordination of Certificate and Degree Programs:

- Certificate may be pursued concurrently with a graduate degree program in School of Information Technology and Engineering.
- Certificate courses may be applied towards the master's degree program in software systems engineering.
- Certificate awarded:
  - Upon completion of certificate if enrolled in no other program
  - Upon completion of degree program if pursued concurrently

Appendix D The Santa Clara University Advanced Studies in Software Engineering Certificate Program

Basic Format: Graduate courses with an emphasis on methodologies used during the development of large and complex software systems.

Contact Points:
- comp-eng@scu.edu
- http://pcsel10.scu.edu/coen/softeng.html
- Address:
  Computer Engineering Department
  Santa Clara University
  Santa Clara, CA 95053
- Telephone: 408-554-4483
- FAX: 408-554-5474

Admission Criteria:
- Bachelor's degree with a GPA of at least 2.75
- English competency as demonstrated by TOEFL
- Competence in data structures
- Introductory level course in software engineering

Course Requirements:

12 units of required courses:
- Formal Methods in Software Engineering
- Software Architecture
- Software Quality Assurance and Testing
- Software Analysis, Maintenance, and Re-Engineering
- Software Development Process Management
- Software Engineering Project

4 units of electives:
- Object-Oriented Analysis and Design
- Object-Oriented Programming
- Software Tools Design
- Graphical User Interface Design and Programming
- Structure and Interpretation of Computer Programs
- Principles of Computer-Aided Engineering Design
- Software Engineering Economics
- Quality Issues in Managing Software
- Managing Software Projects
- Software Project Issues

Course Availability and Facilities:

Regularly offered graduate courses and departmental facilities available for these courses. Some courses are offered on-site at industrial locations. Remote access to courses via television and videos is planned for the future.

Faculty:
- Departmental faculty and adjunct faculty from Silicon Valley technology companies.
- All faculty have Ph.D. degrees.
- Some faculty have significant industrial experience.

Length of Program: Four quarters on part-time basis (4 units per quarter)

Coordination of Certificate and Degree Programs:
- Certificate may be applied towards a graduate degree in Computer Engineering.
- Applicant must follow normal admission procedures for graduate programs except that the GRE is waived for students with a GPA of 3.5 or better in certificate program.