CSE 422 - Introduction to Software Engineering
CSE 522 - Software Engineering I
CSE 622 - Software Engineering II

Undergraduate Curriculum and Course Development
CSE 422, 522, 622
Final Report
for
BAA 95-01
Grant No. DCA100-96-1-0005

Technical Report 96-16
November 8, 1996

Department of Computer Science and Engineering
Auburn University

James H. Cross II, Ph.D.
Principal Investigator
Computer Science and Engineering
107 Dunstan Hall
Auburn University, AL 36849
(334) 844-6315  cross@eng.auburn.edu
Undergraduate Curriculum and Course Development

CSE 422 - Introduction to Software Engineering
CSE 522 - Software Engineering I
CSE 622 - Software Engineering II

James H. Cross II
Auburn University

Abstract

A sequence of three software engineering courses at Auburn University were migrated to Ada 95. The first course provides a survey of the general field of software engineering with detailed coverage of analysis and design. The second course emphasizes verification, implementation, and testing. The third course is essentially the graduate version of the first course. Software projects in these courses had previously focused on the use of general purpose CASE tools with implementation in C and Ada 83. While some of the overall structure of these courses remained intact, the content of each course was significantly revised as it was migrated to Ada 95. In addition, this project introduced the use of GRASP, a development environment which provides for the automatic generation of Control Structure Diagrams (CSDs) for Ada 95. GRASP, which was used as a seamless GUI front-end for GNAT, includes (1) CSD generation, editing, and printing, (2) source code compilation with error messages highlighted in the CSD, and (3) program execution. GRASP, which was also supported by a grant, is described more fully in a separate report.
Table of Contents

1. Introduction ........................................................................................................................................ 1

2. CSE 422 - Introduction to Software Engineering ............................................................................... 2

3. CSE 522 - Software Engineering I ..................................................................................................... 4

4. CSE 622 - Software Engineering II .................................................................................................. 5

5. Deliverables ....................................................................................................................................... 6

6. Proprietary Claims ............................................................................................................................ 7

Acknowledgements .................................................................................................................................... 7

References ................................................................................................................................................ 7

Attachments ............................................................................................................................................. 8

CSE 422 - Introduction to Software Engineering
  Syllabus
  Labs
  Lecture Notes
CSE 522 - Software Engineering I
  Syllabus
  Labs
  Lecture Notes
CSE 622 - Software Engineering II
  Same as CSE 422 (see items above)
1. Introduction

The Department of Computer Science and Engineering offers two bachelor's degrees: (1) the Bachelor of Science in Computer Engineering with its curriculum accredited by the Accreditation Board for Engineering and Technology (ABET) and (2) the Bachelor of Science in Computer Science with its curriculum accredited by the Computer Science Accreditation Board (CSAB). The Master of Science and Doctor of Philosophy degrees are offered at the graduate level. The department offers numerous undergraduate and graduate software engineering courses. Three of these courses are the focus of this report:

- CSE 422 Introduction to Software Engineering
- CSE 522 Software Engineering I
- CSE 622 Software Engineering II

CSE 422 Introduction to Software Engineering is required of all students in both the computer science and computer engineering degree programs. CSE 522 Software Engineering is a dual level elective course taken by both undergraduate and graduate students. These two courses are described in detail below in Sections 2 and 3. It should be noted that most undergraduates will have had CSE 200 and 220 (our CS1 and CS2) which use Ada 95.

CSE 622 is essentially the graduate version of CSE 422. However, since many of the CSE graduate students at Auburn may have little or no background in Ada 95, a more time may be spent on the details of the language itself in CSE 622 than in CSE 422. See Section 4 below for additional details.

While not the focus of this report, it should be noted that CSE 571-572 Senior Design Project (a two quarter sequence) is required by all computer engineering majors. The Senior Design Project course is essentially a non-lecture course which provides a capstone experience for seniors. Students work in teams of three to five on separate projects provided by outside sponsors. During the first quarter, teams meet with their respective sponsors, develop and present their proposals to class, write and present a requirements specifications, and develop an architectural designs. The second quarter is devoted to detailed design, prototyping, and testing, and culminates with installation and demonstration for the sponsor. Implementation languages are determined by a combination of the nature of the project and the students' previous courses and experience. Pascal, C and FORTRAN have dominated recently since Ada has not be emphasized in their previous software engineering courses. Now that CSE 422/522 introduces Ada oriented CASE tools and utilizes Ada 95 for prototyping and testing, the use of Ada 95 is expected to become a viable and desirable choice for the Senior Projects courses.
1. CSE 422 - Introduction to Software Engineering

CSE 422 is a three quarter credit course which has a combination lecture and lab format. The lecture meets twice a week, one hour each time, and the lab meet once a week for three hours over a period of ten weeks. This course was taught in Fall 1996, and is scheduled to be taught in Spring 1997 and twice per year thereafter. In this section, the current version of CSE 422 is described. The deliverables resulting from the migration to Ada are described in Section 4.

Course Description

CSE 422 introduces the student to the broad field of software engineering with concentration on the analysis and design phases of the software life cycle. Traditional structured analysis and design is treated only briefly. The major emphasis is on object-oriented analysis and design. It is the first in a sequence of two traditional software engineering courses (CSE 422, CSE 522) available to undergraduates. The lectures provide a general survey of the material and the labs provide practical experiences.

Textbook

Previously, Pressman's *Software Engineering: A Practitioner's Approach* (3rd edition, 1992) was used prior to the move to Ada. This text is non-specific with respect to an orientation toward a particular computer language although the PDL is somewhat Ada-oriented. It was specifically selected for its potential as a useful reference beyond the scope of CSE 422. Currently, much of the generic software engineering material covered in Pressman's text is now contained in the CSE 422 Lecture Notes provided to the students. Booch and Bryan's *Software Engineering with Ada* was used in CSE 422 last year, but the authors have not updated it to Ada 95.

Currently, *Object-Oriented Software in Ada 95* by Michael A. Smith is used. As its title suggests, this text concentrates on object-oriented programming with Ada 95. Since most of the general software engineering material for the course is included in the Lecture Notes which are provided to the students, this text has been a nice fit. However, other texts will be reviewed for their applicability as general software engineering texts with an underlying orientation of Ada 95.

GRASP/Ada 95

The GRASP/Ada 95 software tool is used to produce *control structure diagrams* during detailed design and implementation [CRO92, CRO96]. GRASP (Graphical Representation of Algorithms, Structures, and Process) is an ongoing project at Auburn University that has been funded as a related Category 3 project, entitled "GRASP/Ada 95 Modification."
The control structure diagram (CSD) is intended to improve the comprehensibility of Ada by providing simple graphical constructs for each Ada control construct without interrupting the traditional "look and feel" of well-structured source code. Our experience has indicated that these diagrams can be generally beneficial for students at all levels, with special benefits for those with limited experience using some of Ada's more complex control constructs such as tasking and rendezvous. To illustrate the latter, a simple example is presented below.

Figure 1 contains an Ada task body CONTROLLER adapted from [BAR84], which loops through a priority list attempting to accept selectively a REQUEST with priority P. Upon on acceptance, some action is taken, followed by an exit from the priority list loop to restart the loop with the first priority. In typical Ada task fashion, the priority list loop is contained in an outer infinite loop. This short example contains two threads of control: the rendezvous, which enters and exists at the accept statement, and the thread within the task body. In addition, the priority list loop contains two exits: the normal exit at the beginning of the loop when the priority list has been exhausted, and an explicit exit invoked within the select statement. While the concurrency and multiple exits are useful in modeling the solution, they do increase the effort required of the reader to comprehend the code.

The CSD in Figure 2 uses intuitive graphical constructs to depict the point of rendezvous, the two nested loops, the select statement guarding the accept statement for the task, the unconditional exit from the inner loop, and the overall control flow of the task. When reading the code without the diagram, as shown in Figure 1, the control constructs and control paths are much less visible although the same structural and control information is available. With additional levels of nesting and increased physical separation of sequential components, the visibility of control constructs and control paths becomes increasingly obscure, and the effort required of the reader dramatically increases in the absence of the CSD.

Figure 2. Task Controller

Figure 1. CSD for Controller
Prerequisites

The current prerequisite for CSE 422 is one of the following: CSE 360 Fundamental Algorithm Design and Analysis or consent of instructor. CSE 360 has a prerequisite of CSE 200/220 (our CS1 and CS2) which both use Ada 95.

Lecture Topics and Labs

The overall structure of the current course is indicated in the Syllabus, Topics, Lecture Notes, and Labs that accompany this report. Although some of the major topics remained, the move to Ada 95 resulted in a major revision of the course content. In particular, the lab exercises were restructured to facilitate object-oriented analysis and design of a term project with detailed design and implementation done in Ada 95 using GRASP.

2. CSE 522 - Software Engineering I

CSE 522 is a four quarter credit course which has a combination lecture and lab format. The lecture meets three times a week, one hour each time, and the lab meet once a week for three hours over a period of ten weeks. Although CSE 522 is an elective course (CSE 422 is required) it is very popular among undergraduates. Many graduate students enroll as well since it is a dual level course. This course was taught in Winter 1996, and is scheduled to be taught once per year in the future. In this section, the current version of CSE 522 is described. The deliverables resulting from the migration to Ada are described in Section 4.

Course Description

CSE 522 introduces the student to the broad field of software engineering with concentration on software quality assurance and testing. Optional topics include proof of correctness, maintenance, and reverse engineering. It is the second in a sequence of two traditional software engineering courses (CSE 422, CSE 522) available to undergraduates. The lectures provide a general survey of the material and the labs provide practical experiences. The general topics are indicated below.

Textbook

Pressman's *Software Engineering: A Practitioner's Approach* (4rd edition, 1997) is the current text for CSE 522. This text is non-specific with respect to an orientation toward a particular computer language although the PDL is somewhat Ada-oriented. Other texts will be reviewed and assessed for their applicability as general software engineering texts with an underlying orientation of Ada.

In addition to the textbook, several technical papers are generally assigned and reviewed in class to supplement to the text, especially in the area of testing. The following are of particular

**Lab Environment and Materials**

The GRASP/Ada software tool described above is used to produce control structure diagrams during detailed design, implementation, and testing. These diagrams have proved particularly useful during (1) Cleanroom correctness verification, (2) technical reviews, and (3) testing. In testing, the diagrams are used in determining test cases and levels of coverage.

**Prerequisites**

The current prerequisite for CSE 522 is one of the following: CSE 422 Introduction to Software Engineering, graduate standing, or consent of instructor.

**Lecture Topics and Labs**

The overall structure of the current course is indicated in the Syllabus, Topics, Lecture Notes, and Labs that accompany this report. Although some of the major topics remained, the move to Ada resulted in a major revision of the course content. In particular, the lab exercises were restructured to facilitate technical reviews, implementation, and testing done in Ada 95.

**3. CSE 622 - Software Engineering II**

CSE 622 is a three quarter credit course which has a three hour per week lecture format with no formal lab. CSE 622 is in many respects the graduate version of CSE 422, but with more emphasis on the details of Ada 95. The lecture meets three times a week, one hour each time, over a period of ten weeks. This course was taught in Fall 1995 as CSE 625. The new version, CSE 622, is scheduled to be taught once per year beginning in AY 97-98. In this section, the new version of CSE 622 is described. The deliverables resulting from the revision and migration to Ada 95 are the same as those for CSE 422. When the new version is taught during AY 97-98, it is anticipated that the CSE 422 Lecture Notes and Labs (homework assignments) will be used in CSE 622. At that time the CSE 422 Syllabus will be adjusted from two hours per lecture and three hours per week lab to a CSE 622 Syllabus of three hours per week lecture.

**Course Description**

CSE 622 reviews the broad field of software engineering, and then concentrates on the analysis and design phases of the software life cycle. Traditional structured analysis and design is treated only briefly. The major emphasis is on object-oriented analysis and design. It is the
second in a sequence of two traditional software engineering courses (CSE 522, CSE 622) available to graduate students. The lectures provide a general survey of the material and the labs provide practical experiences.

Textbook

In Fall 1995, Booch and Bryan's Software Engineering with Ada was used in this. Although this is a very fine textbook, the authors have not updated it to Ada 95.

Currently, Object-Oriented Software in Ada 95 by Michael A. Smith is the front runner for the text to be used the next time the course is taught. As its title suggests, this text concentrates on object-oriented programming with Ada 95. Most of the general software engineering material for the course is included in the Lecture Notes which are provided to the students, so this text should work out well. However, as the course continues to evolve, other texts will be reviewed and assessed for their applicability as general software engineering texts with an underlying orientation of Ada 95.

4. Deliverables

The following deliverables were produced from this project. A hardcopy of each is provided in conjunction with this report (see attachments). Electronic versions are available via anonymous FTP from the following address: ftp.eng.auburn.edu (in directories: pub/cross/cse422 and cse522).

Report - Final Report for Project (this document)

CSE 422 - Introduction to Software Engineering
   Syllabus
   Labs
   Lecture Notes

CSE 522 - Software Engineering I
   Syllabus
   Labs
   Lecture Notes

CSE 622 - Software Engineering II
   Same as CSE 422 (see items above)
5. Proprietary Claims

The principal investigator, and Auburn University retain all intellectual property rights and copyrights. However, the principal investigator and the University seek the widest possible distribution and utilization of the materials developed in this project. Thus, the materials developed within the scope of this project will be made available without charge via electronic distribution. Distribution via other means may require charges to offset shipping costs.

Acknowledgements

Research on the GRASP/Ada project was supported, in part, by a grant from Marshall Space Flight Center, MSFC, AL 35812.

References


Attachments

The following deliverables were produced from this project. A hardcopy of each is provided in conjunction with this report. Electronic versions are available via anonymous FTP from the following address: ftp.eng.auburn.edu in directories: pub/cross/cse422 and cse522.

**CSE 422 - Introduction to Software Engineering**

- Syllabus
- Labs
- Lecture Notes

**CSE 522 - Software Engineering I**

- Syllabus
- Labs
- Lecture Notes

**CSE 622 - Software Engineering II**

- Same as CSE 422 (see items above)
A sequence of three software engineering courses at Auburn University were migrated to Ada 95. The first course provides a survey of the general field of software engineering with detailed coverage of analysis and design. The second course emphasizes verification, implementation, and testing. The third course is essentially the graduate version of the first course. Software projects in these courses had previously focused on the use of general purpose CASE tools with implementation in C and Ada 83. While some of the overall structure of these courses remained intact, the content of each course was significantly revised as it was migrated to Ada 95. In addition, this project introduced the use of GRASP, a development environment which provides for the automatic generation of Control Structure Diagrams (CSDs) for Ada 95. GRASP, which was used as a seamless GUI front-end for GNAT, includes (1) CSD generation, editing, and printing, (2) source code compilation with error messages highlighted in the CSD, and (3) program execution. GRASP, which was also supported by a grant, is described more fully in a separate report.