



MONASH University
Information Technology

FIT3077
Software engineering: architecture and design

Unit Guide

Semester 1, 2011

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated: 27 Feb 2011

Table of Contents

<u>FIT3077 Software engineering: architecture and design - Semester 1, 2011</u>	1
<u>Mode of Delivery</u>	1
<u>Contact Hours</u>	1
<u>Workload</u>	1
<u>Unit Relationships</u>	1
<u>Prohibitions</u>	1
<u>Prerequisites</u>	1
<u>Chief Examiner</u>	1
<u>Campus Lecturer</u>	1
<u>Clayton</u>	2
<u>Learning Objectives</u>	2
<u>Graduate Attributes</u>	2
<u>Assessment Summary</u>	3
<u>Teaching Approach</u>	3
<u>Feedback</u>	3
<u>Our feedback to You</u>	3
<u>Your feedback to Us</u>	3
<u>Previous Student Evaluations of this unit</u>	3
<u>Required Resources</u>	4
<u>Unit Schedule</u>	4
<u>Assessment Policy</u>	5
<u>Assessment Tasks</u>	5
<u>Participation</u>	5
<u>Examinations</u>	6
<u>Examination 1</u>	6
<u>Assignment submission</u>	6
<u>Extensions and penalties</u>	6
<u>Returning assignments</u>	6
<u>Policies</u>	6
<u>Student services</u>	7

FIT3077 Software engineering: architecture and design - Semester 1, 2011

This unit builds on introductory units to analysis and design. It provides the professional software engineer with advanced knowledge and skills in high-level architectural design, its theoretical foundations, industrial best practice, and relevant application context. In the software life-cycle, software architecture sits between analysis/specification and design/implementation. The field of software architecture has come of age with a thriving research community and numerous high-level models, methods, tools and practices widely used in industry.

Mode of Delivery

Clayton (Day)

Contact Hours

2 hrs lectures/wk, 1 hr laboratory/wk

Workload

Workload commitments are:

- two-hour lecture and
- one-hour practice class conducted in a laboratory.
- a minimum of 3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

Unit Relationships

Prohibitions

CSE3308

Prerequisites

FIT2001 (or CSE2305) and FIT2004 (or CSE2304)

Chief Examiner

David Squire

Campus Lecturer

Clayton

David Squire

Contact hours: Consultation hours will be announced on the MUSO website. Making an appointment via email is strongly advised.

Learning Objectives

At the completion of this unit students will have -
A knowledge and understanding of:

- modelling and design of flexible software at the architectural level. Basics of model-driven architecture;
- Architectural styles and patterns, Middleware & application frameworks;
- product lines. Design using COTs software;
- configurations and configuration management;
- in-depth look at software design, design patterns;
- design of distributed systems using middleware;
- design for qualities such as performance, safety, reusability etc;
- evaluation and evolution of designs, reverse engineering.

Developed attitudes that enable them to:

- apply variety of design pattern;
- appreciate analysis fundamentals;
- analyse well-formedness (completeness, consistency, robustness, etc);
- analyse correctness (eg. static analysis, simulation etc.);
- analyse quality requirements (eg. root cause analysis, safety, usability, security, etc.).

Developed the skills to:

- take requirements for simple systems and develop software architectures and designs at a high level;
- use configuration management tools effectively;
- apply a variety of frameworks and architectures in designing software.

Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:
 - a. engage in an internationalised world
 - b. exhibit cross-cultural competence
 - c. demonstrate ethical values

critical and creative scholars who:

- a. produce innovative solutions to problems
- b. apply research skills to a range of challenges
- c. communicate perceptively and effectively

Assessment Summary

Examination (3 hours): 40%; In-semester assessment: 60%

Assessment Task	Value	Due Date
UML Design Assignment	20%	Week 4, Friday 25 March 2011
Team Software Architecture Assignment	40%	Week 9, Friday 6 May 2011 (Stage 1); Week 12, Friday 27 May 2011 (Stage 2)
Examination 1	40%	To be advised

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

Feedback

Our feedback to You

Types of feedback you can expect to receive in this unit are:

- Informal feedback on progress in labs/tutes
- Graded assignments with comments
- Interviews

Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through SETU, Student Evaluation of Teacher and Unit. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, and on student evaluations, see:

<http://www.monash.edu.au/about/monash-directions/directions.html>

<http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html>

Previous Student Evaluations of this unit

If you wish to view how previous students rated this unit, please go to

<https://emuapps.monash.edu.au/unitevaluations/index.jsp>

Required Resources

You will need access to:

- A tool for creating UML diagrams, such as Rational Rose, Visual Paradigm, etc.
- An object-oriented programming language, with the capacity to create a graphical user interface.

On-campus students may use the software that is installed in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook.

Unit Schedule

Week	Date*	Activities	Assessment
0	21/02/11		No formal assessment or activities are undertaken in week 0
1	28/02/11	Introduction to FIT3077; What is Software Architecture?; Object-Oriented Analysis using UML	
2	07/03/11	Object-Oriented Analysis using UML	Assignment 1 specification available
3	14/03/11	Principles of Object-Oriented Analysis and Design; Design Patterns	
4	21/03/11	Principles of Object-Oriented Design	Assignment 1 due Fri 25 March 2011
5	28/03/11	Principles of Object-Oriented Design	Assignment 2, Stage 1 specification available
6	04/04/11	Design Principles and Design Patterns	
7	11/04/11	Analysis Patterns; Refactoring	
8	18/04/11	Software Architecture; Architectural Structures	
Mid semester break			
9	02/05/11	Documenting Software Architectures; The Model-View-Controller Architectural Pattern	Assignment 2, Stage 1 due Fri 6 May 2011; Assignment 2, Stage 2 specification available
10	09/05/11	Architecture and Design with COTS components	
11	16/05/11	Software Product Lines: Re-using Architectural Assets	
12	23/05/11	Service Orientation; Service-Oriented Architecture	Assignment 2, Stage 2 due Fri 27 May 2011
	30/05/11	SWOT VAC	No formal assessment is undertaken SWOT VAC

*Please note that these dates may only apply to Australian campuses of Monash University. Off-shore students need to check the dates with their unit leader.

Assessment Policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 50% then a mark of no greater than 49-N will be recorded for the unit

Assessment Tasks

Participation

• Assessment task 1

Title:

UML Design Assignment

Description:

Students will be provided with software requirements scenarios for which they must produce a conceptual design for an object-oriented solution. This design must be described using UML diagrams. The purpose of this assignment is to reinforce UML knowledge and ensure that students are ready for the next assignment.

Weighting:

20%

Criteria for assessment:

Designs will be assessed for completeness with respect to the specification provided, correct use of UML notation, and design quality.

Due date:

Week 4, Friday 25 March 2011

• Assessment task 2

Title:

Team Software Architecture Assignment

Description:

This is a team assignment. Students will work on it in pairs. This assignment will have two stages, each requiring students to analyse a problem, design a solution, and implement it in an object-oriented programming language. The second stage will build on the first, and the ease with which the students' stage one design can be extended for stage two will depend on appropriate use of architecture and design knowledge presented in this unit.

The main focus of the assignment is on the application of the design patterns and principles presented in lectures. Simply meeting the external functional requirements for the system is not sufficient to pass this assignment.

Weighting:

40%

Criteria for assessment:

Teams will be assessed on the basis of their submitted design documentation and code, as well as an interview. Both team members will be required to demonstrate knowledge

and understanding of all parts of their design.

Designs and code will be assessed for quality, including extensibility and appropriate use of design patterns. Completeness of code functionality with respect to the problem requirements will also be assessed.

Due date:

Week 9, Friday 6 May 2011 (Stage 1); Week 12, Friday 27 May 2011 (Stage 2)

Examinations

• Examination 1

Weighting:

40%

Length:

3 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

None

Assignment submission

Assignment coversheets are available via "Student Forms" on the Faculty website:

<http://www.infotech.monash.edu.au/resources/student/forms/>

You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:

<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>.

Returning assignments

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at:

<http://policy.monash.edu.au/policy-bank/academic/education/index.html>

Key educational policies include:

- Plagiarism (<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html>)
- Assessment (<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-p>)
- Special Consideration (<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h>)
- Grading Scale (<http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html>)
- Discipline: Student Policy (<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html>)
- Academic Calendar and Semesters (<http://www.monash.edu.au/students/key-dates/>);
- Orientation and Transition (<http://www.infotech.monash.edu.au/resources/student/orientation/>); and
- Academic and Administrative Complaints and Grievances Policy (<http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy>)

Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at www.monash.edu.au/students The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to <http://www.lib.monash.edu.au> or the library tab in my.monash portal for more information. Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis

- Website: <http://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html>;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

Reading List

- Fowler M., UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 3rd ed., 2003.
- Gamma E., Helm R., Johnson R., Vlissides J. M., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
- Martin R. C., Design Principles and Design Patterns, 2000.
 - ◆ This and other Robert C. Martin articles are made available through the MUSO site for the unit.
- Fowler M., Analysis Patterns: Reusable Object Models, Addison-Wesley, 1996.
- Fowler M., Beck K., Brant J., Opdyke W., Roberts D., Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999.
- Bass L., Clements P. and Kazman R., Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003.
 - ◆ Full text available electronically through the Monash library
- Allen, P and Schlamann, H., Service Orientation: Winning Strategies and Best Practices, Cambridge University, 2006.
 - ◆ Full text available electronically through the Monash library