



MONASH University

FIT3077
Software engineering: architecture and design

Unit guide

Semester 1, 2009

Last updated : 13 Feb 2009

Table of Contents

<u>FIT3077 Software engineering: architecture and design - Semester 1, 2009</u>	1
<u>Unit leader</u> :.....	1
<u>Lecturer(s)</u> :.....	1
<u>Clayton</u>	1
<u>Malaysia</u>	1
<u>Introduction</u>	1
<u>Unit synopsis</u>	1
<u>Learning outcomes</u>	1
<u>Workload</u>	2
<u>Unit relationships</u>	2
<u>Prerequisites</u>	2
<u>Relationships</u>	2
<u>Continuous improvement</u>	3
<u>Student Evaluations</u>	3
<u>Improvements to this unit</u>	3
<u>Unit staff - contact details</u>	3
<u>Unit leader</u>	3
<u>Lecturer(s)</u> :.....	3
<u>Additional communication information</u>	4
<u>Teaching and learning method</u>	4
<u>Tutorial allocation</u>	4
<u>Communication, participation and feedback</u>	4
<u>Unit Schedule</u>	4
<u>Unit Resources</u>	6
<u>Prescribed text(s) and readings</u>	6
<u>Recommended text(s) and readings</u>	6
<u>Required software and/or hardware</u>	6
<u>Equipment and consumables required or provided</u>	7
<u>Study resources</u>	7
<u>Library access</u>	7
<u>Monash University Studies Online (MUSO)</u>	7
<u>Assessment</u>	8
<u>Unit assessment policy</u>	8
<u>Assignment tasks</u>	8
<u>Examinations</u>	9
<u>Assignment submission</u>	9
<u>University and Faculty policy on assessment</u>	9
<u>Due dates and extensions</u>	9
<u>Late assignment</u>	9
<u>Return dates</u>	10
<u>Plagiarism, cheating and collusion</u>	10
<u>Register of counselling about plagiarism</u>	10
<u>Non-discriminatory language</u>	11
<u>Students with disabilities</u>	11
<u>Deferred assessment and special consideration</u>	11

FIT3077 Software engineering: architecture and design - Semester 1, 2009

Unit leader :

David Squire

Lecturer(s) :

Clayton

- David Squire

Malaysia

- Saadat Alhashmi

Introduction

Welcome to FIT3077 'Software Engineering: Architecture and Design' for semester 1, 2009. This 6 point unit is core to both the Bachelor of Software Engineering and the Bachelor of Computer Science. In this unit you will learn about both large and small scale software architecture and design, including both design and analysis patterns, architectural patterns such as Model-View-Controller, and emerging topics such as service-oriented architectures and commercial-off-the-shelf components (COTS). You will learn about incremental design improvement through refactoring. These designs, patterns, and architectures will be described using the Unified Modeling Language (UML).

Unit synopsis

Modeling and analysis is first applied for the analysis, specification & validation of requirements. Software design is focused on strategies, techniques, patterns and representations used to decide how to implement a component or a system.

Major topics include: object-oriented analysis using UML, principles of object-oriented design, design patterns, analysis patterns, refactoring, software architectures, documenting software architecture, architecture and design with COTS components, software product lines, and service-oriented architectures.

Learning outcomes

Knowledge and Understanding

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

- In-depth look at software design principles, analysis and design patterns
- Modeling and design of flexible software at the architectural level
- Architectural styles and patterns
- Product lines

- Design using COTS software
- Service Orientation

Attitudes, Values and Beliefs

At the completion of this unit students will have developed attitudes that enable them to:

- apply a variety of analysis and design patterns
- appreciate design fundamentals and subsystem and architectural levels
- analyse design quality (completeness, extensibility, performance)
- analyse correctness (e.g. static analysis, simulation etc.)

Practical Skills

At the completion of this unit students will have the skills to:

- take requirements for simple systems and develop software architectures and designs at a high level
- apply a variety of patterns and architectures in designing software
- create software that realizes these design patterns and principles

Relationships, Communication and Teamwork

Objectives in this domain cover skills for building relationships and working collaboratively. They include communication skills, teamwork skills and leadership and management skills. This domain is closely linked to the affective domain, but involves objectives that develop skills related to group work.

Workload

Workload commitments are:

- two-hour lecture and
- one-hour practice class
- A minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

Unit relationships

Prerequisites

Before attempting this unit you must have satisfactorily completed FIT2001 or CSE2305, and FIT2004 or CSE2304, or equivalent..

Relationships

FIT3077 is a core unit in the Bachelor of Software Engineering and the Bachelor of Computer Science.

Before attempting this unit you must have satisfactorily completed FIT2001 or CSE2305, and FIT2004 or CSE2304, or equivalent.

You may not study both this unit and CSE3308 in your degree.

Continuous improvement

Monash is committed to 'Excellence in education' (Monash Directions 2025 - <http://www.monash.edu.au/about/monash-directions/directions.html>) and strives for the highest possible quality in teaching and learning.

To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. The University's Unit Evaluation policy (<http://www.policy.monash.edu/policy-bank/academic/education/quality/unit-evaluation-policy.html>) requires that every unit offered is evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to "have their say". The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

Faculties have the option of administering the Unit Evaluation survey online through the my.monash portal or in class. Lecturers will inform students of the method being used for this unit towards the end of the semester.

Student Evaluations

If you wish to view how previous students rated this unit, please go to <http://www.monash.edu.au/unit-evaluation-reports/>

Improvements to this unit

Student feedback on the first-ever offering of this unit in first semester 2008 was uniformly positive (if limited so far). Some lectures might be improved by the addition of more practical examples of the principles presented.

Feedback for the second semester offering was not so good. The major problem appears to have been timely feedback on assignment work. Consideration will be given to assignment design to attempt to reduce the marking load for the lecturer (there are no tutors for this unit). An approach will also be made to the HoS to consider providing marking assistance for this unit.

Unit staff - contact details

Unit leader

Dr David Squire

Senior Lecturer

Phone +61 3 990 59013

Fax +61 3 990 55157

Lecturer(s) :

Dr David Squire

Senior Lecturer

Phone +61 3 990 59013

Fax +61 3 990 55157

Dr Saadat Alhashmi

Additional communication information

It is always a good idea to email the lecturer in advance if you wish to meet outside designated consultation hours. David Squire is also teaching at the Caulfield campus in first semester, and will thus often be away from his Clayton office.

Teaching and learning method

There will be two one-hour lectures per week in this unit. During lectures, the lecturer will present the theory underlying the topics in the unit, and illustrate this with concrete examples.

There will also be a one-hour 'practice class' for the unit. The lecturer and tutors (if any) will be present for the practice class. Students are expected to work on example problems or their assignments during the practice class, and ask the teaching staff for assistance when necessary. Teaching staff may occasionally choose to explain a common problem to the whole class.

Note that a practice class is *not a tutorial*. It is time for you to work on the example problems provided, with a staff member present to answer any questions you might have. To get value from the practice class, it is vital that you attempt the example problems provided.

The practice classes are intended to be the primary mechanism for consultation in this unit.

Tutorial allocation

Students should register for a practice class using Allocate+. It will most likely be listed as a 'laboratory'.

Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should discuss it with your lecturer or a Community Service counsellor as soon as possible.

Unit Schedule

Week	Topic	References/Readings	Key dates
1	Introduction to FIT3077; What is Software Architecture?; Object-Oriented Analysis using UML	Bass L., Clements P. and Kazman R. Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003, Ch. 2.; Fowler, Martin, UML Distilled, Addison-Wesley, 1997, 2000, or 2003.	
2	Object-Oriented Analysis using UML	Fowler, Martin, UML Distilled, Addison-Wesley,	Assignment One specification

		1997, 2000, or 2003.	available
3	Principles of Object-Oriented Analysis and Design; Design Patterns	Page-Jones, Meilir, Fundamentals of Object-Oriented Design in UML, Addison-Wesley, 2000 (Ch. 8, 9); Gamma, Erich; Helm, Richard; Johnson, Ralph; and Vlissides, John, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995, Chs. 1, 3, 4, 5.	
4	Principles of Object-Oriented Design	Martin, Robert C., Design Principles and Design Patterns, 2000 (available via Blackboard).	Assignment One due
5	Principles of Object-Oriented Design	Martin, Robert C., Design Principles and Design Patterns, 2000 (available via Blackboard).; Martin, Robert C., Granularity, 1997 (available via Blackboard).	Assignment Two, Stage One specification available
6	Design Principles and Design Patterns	Martin, Robert C., Design Principles and Design Patterns, 2000 (available via Blackboard).	
Mid semester break			
7	Analysis Patterns; Refactoring	Fowler, Martin, Analysis Patterns: Reusable Object Models, Addison-Wesley, 1997 (Chs. 1, 2); Fowler, Martin, Refactoring: Improving the Design of Existing Code, Addison-Wesley, 1999, Chs. 1, 2, 7.	
8	Software Architecture; Architectural Structures	Bass L., Clements P. and Kazman R. Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003, Ch. 2.	
9	Documenting Software Architectures; The Model-View-Controller Architectural Pattern	Bass L., Clements P. and Kazman R. Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003, Ch. 9.; Microsoft Corporation, Model-View-Controller, Microsoft Patterns & Practices Developer Center,	Assignment Two, Stage One due; Assignment Two, Stage Two specification available

		2008.	
10	Architecture and Design with COTS components	Bass L., Clements P. and Kazman R. Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003, Ch. 18.	
11	Software Product Lines: Re-using Architectural Assets	Bass L., Clements P. and Kazman R. Software Architecture in Practice, Addison-Wesley, 2nd ed., 2003, Ch. 14.	
12	Service Orientation; Service-Oriented Architecture	Allen, Paul and Schlamann, Hermann, Service Orientation: Winning Strategies and Best Practices, Cambridge University Press, 2006, Chs. 1, 3, 7	Assignment Two, Stage Two due
13	Revision		

Unit Resources

Prescribed text(s) and readings

There is no set text for this unit.

Recommended text(s) and readings

- 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 Blackboard 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

Required software and/or hardware

You will need access to:

- A tool for creating UML diagrams, such as Rational Rose, Visual Paradigm, etc.

- An object-oriented programming language

On-campus students may use this software which 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.

Equipment and consumables required or provided

Students studying off-campus are required to have the minimum system configuration specified by the Faculty as a condition of accepting admission, and regular Internet access. On-campus students, and those studying at supported study locations may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to 9 hours per week for use of a computer, including time for newsgroups/discussion groups.

Study resources

Study resources we will provide for your study are:

- Weekly detailed lecture notes outlining the learning objectives, discussion of the content, and required readings;
- Regular practice class exercises, with feedback and solutions provided *in the practice classes*;
- Assignment specifications;
- A sample examination and suggested solutions presented *in the week 13 lectures*;
- Discussion groups;
- This Unit Guide outlining the administrative information for the unit;
- The unit web site on MUSO, where resources outlined above will be made available.

Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to <http://www.lib.monash.edu.au>.

The Educational Library and Media Resources (LMR) is also a very resourceful place to visit at <http://www.education.monash.edu.au/library/>

Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (<http://moodle.monash.edu.au>) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

You can access MUSO and Blackboard via the portal: <http://my.monash.edu.au>

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: <http://www.monash.edu.au/muso/support/students/downloadables-student.html>

You can contact the MUSO Support by phone : (+61 3) 9903 1268

For further contact information including operational hours, please visit:
<http://www.monash.edu.au/muso/support/students/contact.html>

Further information can be obtained from the MUSO support site:
<http://www.monash.edu.au/muso/support/index.html>

Assessment

Unit assessment policy

The unit is assessed with two assignments and a three hour closed book examination. To pass this unit, a student must obtain:

- 40% or more in the unit's examination and
- 40% or more in the unit's 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 assessment then a mark of no greater than 44-N will be recorded for the unit.

Assignment tasks

• Assignment Task

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 large 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

• Assignment Task

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.

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 12

Examinations

- **Examination 1**

Weighting : 40%

Length : 3 hours

Type (open/closed book) : Closed book

Assignment submission

Assignment submission details will be specified in the assignment specifications.

University and Faculty policy on assessment

Due dates and extensions

The due dates for the submission of assignments are given in the previous section. Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Requests for extensions must be made to the unit lecturer at your campus **before the due date**.

You will be asked to forward original medical certificates in cases of illness, and may be asked to provide other forms of documentation where necessary. A copy of the email or other written communication of an extension must be attached to the assignment submission.

Late assignment

Assignments received after the due date will receive a mark of $M \cdot (0.9)^n$, where M is the raw mark for the assignment, and n is the number of days that the assignment is late (including weekends).

Return dates

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

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at <http://www.policy.monash.edu/policy-bank/academic/education/assessment/>

We will aim to have assignment results made available to you within two weeks after assignment receipt.

Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with Student Rights and Responsibilities (<http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html>) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

Register of counselling about plagiarism

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

Non-discriminatory language

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

Students with disabilities

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- [Disabilities Liaison Unit](#)

Deferred assessment and special consideration

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at <http://www.monash.edu.au/exams/special-consideration.html>. Contact the Faculty's Student Services staff at your campus for further information and advice.