COMP5348
Enterprise-Scale Software Architecture
6 Credit Points
Unit of Study Outline & Assessment Details
Semester 1, 2010
IMPORTANT: Policy relating to Academic Dishonesty and Plagiarism.

The School of Information Technologies views all forms of academic dishonesty, including plagiarism and recycling, very seriously.

Plagiarism means presenting another person’s ideas, findings or work as one’s own by copying or reproducing them without due acknowledgement of the source.

Recycling means the submission for assessment of one’s own work, or of work which is substantially the same, which has previously been counted towards the satisfactory completion of another unit of study, and credited towards a university degree, and where the examiner has not been informed that the student has already received credit for that work.

Students who submit work containing significant portions that have been copied from other sources, including published works, the internet, existing programs, work previously submitted for other awards or assessments, or the work of other students, without proper acknowledgement will be penalised. Decisions as to the penalty may include:

(a) counselling the student;
(b) issuing a written warning;
(c) requiring the student to resubmit the work for assessment; or to undertake other remedial work;
(d) requiring the student to undertake another form of assessment in lieu of the assignment in question, such as an unseen examination;
(e) applying a fail grade to the work, or part thereof, submitted for assessment;
(f) applying a fail grade overall in the unit of study; or
(g) referring the matter to the Registrar if the head of school considers there has been a breach of the University’s standards of academic honesty and the student continues in a denial, or, following the interview, the head of school considers that failing the unit of study is insufficient to deal with the matter.

Where there is doubt about which portions of work are contributed by a particular student he/she may be required to demonstrate knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

COMP5348 Enterprise-Scale Software Architecture

1. Introduction

This unit covers topics on software architecture for large-scale enterprises. Computer systems for large-scale enterprises handle critical business processes, interact with computer systems of other organisations, and have to be highly reliable, available and scalable. This class of systems are built up from several application components, incorporating existing "legacy" code and data stores as well as linking these through middleware technologies, such as distributed transaction monitors, remote objects, message-queueing, publish-subscribe, and clustered application servers. The choice of middleware can decide whether the system achieves essential non-functional requirements such as performance and availability. The objective of this unit of study is to educate students for their later professional career and it covers Software Architecture topics from the ACM/IEEE Software Engineering curriculum.

2. Objectives

This subject will cover both fundamental concepts that apply across many technologies, and practical work with specific technologies. Students who successfully complete this unit will:

1. Understand the role of a software architect; Respect what he/she is doing, and why; Know when to involve him/her. (Graduate Attribute Type: Professional Practice)

2. Demonstrate broad knowledge of common architectural approaches for enterprise software, as well as detailed skills with some technologies available to implement those approaches. (Graduate Attribute Type: Discipline Specific Expertise)

3. Produce written evaluation of different architectures and/or of different technologies. (Graduate Attribute Type: Communication)

4. Know the fundamentals of performance analysis, performance measurement principles, and state management in face of concurrency and distribution. (Graduate Attribute Type: Science and Engineering Fundamentals)

5. Work in small teams of people with diverse skills and backgrounds, to complete demanding tasks which are loosely defined and require rapid learning of new concepts. (Graduate Attribute Type: Teamwork and Project Management)

3. Entry Requirements

Programming competence in Java, C# or a similar OO language. Capacity to master novel technologies (especially to program against novel APIs) using manuals, tutorial examples, etc.

4. Unit of Study Delivery

A variety of learning situations will be employed during the unit of study, including lectures, on-line demos, prescribed reading, tutorials, pen and paper exercises, and directed computer laboratory exercises. To benefit fully from this unit it is necessary to participate fully in all aspects of the unit of study.

5. Expectations

1. Students are expected to attend all scheduled lectures, and laboratory classes. You should expect to spend a minimum of nine hours per week including scheduled lectures and laboratory times.

2. Students are expected to undertake prescribed reading, to carry out exercises and laboratory tasks and to submit selected work for assessment as directed. It should be realised that laboratory exercises are expected to take longer than just the time scheduled for classes.

3. Students are expected to self-dependently prepare the prescribed research paper readings and conduct additional literature and system research as necessary.

4. Students are expected to be able to work independently and to make effective use of a range of resources.
including the library, the Internet and relevant on-line help facilities.

5. Students are expected to check their progressive results regularly. Results will be published through Sydney eLearning. Any errors or omissions must be reported to the unit coordinator, with appropriate evidence, as soon as possible. **Please note:** Marks are considered to have been confirmed ten days after being published and will not subsequently be altered.

6. **Assessment Package**

   The unit of study will be assessed by means of the following components:

   ![table]

<table>
<thead>
<tr>
<th>Component</th>
<th>Week</th>
<th>Learning Outcomes Assessed</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1: Group Questions</td>
<td>7</td>
<td>1,2,4</td>
<td>10</td>
</tr>
<tr>
<td>Group Project</td>
<td>11</td>
<td>2,3,5</td>
<td>20</td>
</tr>
<tr>
<td>Assignment 2: Group Questions</td>
<td>12</td>
<td>2,3,4</td>
<td>10</td>
</tr>
<tr>
<td>Final examination, 2 hour paper</td>
<td></td>
<td>1,2,3,4</td>
<td>60</td>
</tr>
</tbody>
</table>

   **It is a policy of the School of Information Technologies that in order to pass this unit, a student must achieve at least 40% in the written examination as well as in the other components of assessment together. A student must also achieve an overall final mark of 50 or more. Any student not meeting these requirements can achieve a maximum mark of no more than 45.**

   **Late work:** In the interests of fairness to all students, the School of Information Technologies policy states that late work cannot be accepted. In exceptional cases late work must be submitted directly to the unit of study coordinator accompanied by an application for Special Consideration as outlined on page 16 of the School of Information Technologies Postgraduate Enrolment Guide.

   Assessment results will be published on WebCT or the course web page. Students are required to check their results.

   Any errors or omissions must be reported to the unit coordinator, with appropriate evidence, within ten (10) days of being published. Ten days after being published, marks are considered to have been confirmed and will not subsequently be altered.

   Deadlines for assignments are set on the assumption that students may experience minor setbacks caused by sickness, computer breakdown etc. In this context, ‘minor’ means ‘causing a delay of up to three working days’. Extensions will not be granted for minor setbacks. It is important to work steadily on assignments as soon as they are given.

7. **Details of Assessment Components**

   Note that each group activity will be done in a different group, typically consisting of 3 students.

7.1 **Assignment 1: Group Questions**

   In this paper-based assignment, students work in groups to answer questions on material of weeks 1-6. These tasks are similar to questions on the final exam. Due in week 7, worth 10%.

7.2 **Group Project**

   In this practical assignment, students work in groups using various middleware technologies, including distributed transactions, asynchronous messaging, and web services. Due in week 11, worth 20%. Groups are asked to hand in a draft, with a design and some early iterations of the coding, in week 6; this draft receives informative feedback, but is not worth any marks.

7.3 **Assignment 2: Group Questions**

   In this paper-based assignment, students work in groups to answer questions on material of weeks 7-11. These tasks are similar to questions on the final exam. Due in week 12, worth 10%.

7.4 **Written Examination**

   The written examination will be drawn from all aspects of the unit of study. It will test the candidates’ ability to discuss issues critically and to apply the knowledge learnt during the course them to specific situations. Duration of the examination will be two hours.
8. **Teaching team**

<table>
<thead>
<tr>
<th>Name</th>
<th>Room</th>
<th>Phone</th>
<th>Consultation time</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof Alan Fekete</td>
<td>SIT 4E-447</td>
<td>93514287</td>
<td>By appt</td>
<td><a href="mailto:alan.fekete@sydney.edu.au">alan.fekete@sydney.edu.au</a></td>
</tr>
<tr>
<td>(coordinator and some lectures)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mr Paul Greenfield</td>
<td></td>
<td></td>
<td></td>
<td><a href="mailto:p.greenfield@computer.org">p.greenfield@computer.org</a></td>
</tr>
<tr>
<td>(some lectures)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9. **Textbook and Readings**

The main reference for much of the course will be:

**Essential software architecture / Ian Gorton, 005.12 61**

Other books with useful material for parts of the course include:

**Web services : concepts, architectures and applications / Gustavo Alonso ... [et al.]. 005.72 143**

**Performance by design : computer capacity planning by example / Daniel A. Menascé, Virgilio A. F. Almeida, Lawrence W. Dowdy. 004.24 22**

All these reference books are available at the SciTech library on reserve, for short-period use by comp5348 students.

9.1 **Course Web Page**

For most recent information on this course, lecture slides, exercises etc. visit WebCT or the course’s web page at [http://sydney.edu.au/it/~comp5348](http://sydney.edu.au/it/~comp5348)

10. **Lecture and Assessment Schedule**

This schedule is indicative only - the list of topics and dates may change. Lecture slides are usually posted on the unit website a few days after the lecture.

<table>
<thead>
<tr>
<th>Wk</th>
<th>Lecture</th>
<th>Topic</th>
<th>Assessment due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Wed 3 March</td>
<td>Introduction; Definition of the field; Nature of Software Architect's Role; Types of non-functional requirements (“ilities”)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Wed 10 March</td>
<td>State and Concurrency</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Wed 17 March</td>
<td>Distributed Computing; RPC, Distributed Objects, and Component Models</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Wed 24 March</td>
<td>Distributed Transactions</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Wed 31 March</td>
<td>Performance Measures</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wed 14 April</td>
<td>Predicting performance</td>
<td>Draft project [informative]</td>
</tr>
<tr>
<td>7</td>
<td>Wed 21 April</td>
<td>Asynchronous Messaging and related technologies</td>
<td>Assignment 1</td>
</tr>
<tr>
<td>8</td>
<td>Wed 28 April</td>
<td>Service Oriented Architectures I</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Wed 5 May</td>
<td>Service Oriented Architectures II</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Wed 12 May</td>
<td>Fault-tolerance concepts</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Wed 19 May</td>
<td>Replication for scalability and availability</td>
<td>Project</td>
</tr>
<tr>
<td>12</td>
<td>Wed 26 May</td>
<td>Guest lecturer from industry</td>
<td>Assignment 2</td>
</tr>
<tr>
<td>13</td>
<td>Wed 2 June</td>
<td>Unit Review</td>
<td></td>
</tr>
</tbody>
</table>