Copyright warning

COMP5348 Enterprise Scale Software Architecture

Lecture 1: Introduction
Thursday March 5, 2009

Agenda

- Administrivia
  - People and places
  - Course objectives and outline
  - Assessment
  - Policies

Administrative (Places)

- Lectures
  - Time: Thursday 6-8 pm
  - **Carslaw 251**
- Labs
  - Thursdays 8-9 SIT building rms 114 and 115
- Course Website:
  - http://www.cs.usyd.edu.au/~comp5348/

Administrative (people)

- Instructors
  - Unit coordinator: A/Prof Alan Fekete (fekete@it.usyd.edu.au)
    - Phone: 93514287, Office: SIT 447
  - Dr Uwe Roehm (roehm@it.usyd.edu.au)
    - Phone: 90365305, Office: SIT 446

Communication Channels

- Consultation
  - Fekete: Mon 5-6, call from kiosk on SIT 4th floor;
  - Roehm: Mon 2-3, call from kiosk on SIT 4th floor;
  - Or email for appointment
- Emails: {fekete, roehm}@it.usyd.edu.au
- USyd eLearning site
  - Especially discussion board
- Unit web site: www.it.usyd.edu.au/~comp5348

Outcomes

- Understand the role of a software architect
- Know the main concepts and technologies used in architectures for large-scale enterprise software
- Produce written evaluation of different architectures and/or of different technologies
- Knowledge of fundamentals of performance analysis, performance measurement principles, and state management in face of concurrency and distribution
- 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
- Not an outcome: ready to work as software architect!

Enrolment advice

- Who should take this course?
  - Are you comfortable in an OO language: Java/C#/C++?
  - Can you learn to use a new language or technology quickly, from the API and on-line documentation?
  - Are you planning a career path in software development, especially for large-scale business applications? Or perhaps in consulting for such enterprises?
- Who should probably back away from this course?
  - Are you completely new to OO programming?
  - Do you hate reading/writing code?
  - Do you hate working with others?

Expectations

- Assumed Knowledge
  - Java or other similar OO languages (C#, C++)
  - Many examples will be in Java, others may use C#
- Know the uni/school policies
  See www.it.usyd.edu.au/current_students/postgrad_coursework/policies/communication.shtml
- Read the tutorial instructions before the lecture
- Make sure you keep up with the progress
  - Consult course staff **EARLY** enough for difficulties and problems!
  - Spend at least 9 hrs per week on this unit

Course Outline

- Two major aspects of the contents, interspersed
  - microarchitecture issues
    - state management
    - fault-tolerance
    - performance
    - capacity planning
  - enterprise-level architecture concepts
    - middleware and integration technologies
    - asynchrony (messaging, events, and pub/sub)
    - replication and scalability
    - service-oriented architecture
    - security, admin and deployment

Assessment Components

- 4 Group Assignments (10% each, total 40%)
  - Ass1 (due wk 6); Ass 3 (due wk 11) medium-scale programme modifications with written discussion
  - Ass2 (due wk 8) and Ass4 (due wk 12): paper-based, short answers, calculations and discussions (like the exam)
  - Groups will change each assignment!
- Results will be published on USyd eLearning two weeks after the due date, please check to make sure that your results are correctly recorded. Any discrepancies should be resolved within one week after the result being published!
- Final Exam (60%)
  - 2 hours
- Note: by School of IT policy, in order to pass the unit, you must obtain at least 40% on exam, at least 40% on assignments, *and* at least 50% overall!

Special Consideration

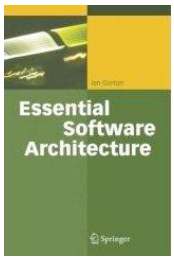
- If your performance on assessments is affected by illness or misadventure
- Follow proper bureaucratic procedures
  - Have professional practitioner sign special USyd form
  - Submit application for special consideration to Faculty office
- Also, notify coordinator by email immediately

Special Consideration (Exam)

- If you have sat for the exam, you won't be allowed to sit it again, nor will your marks be adjusted
- If you don't feel well enough to show your knowledge/skill, *do not sit the exam*
- Follow procedures and apply for special consideration for missing the exam because of illness/misadventure
- If you become ill during the exam, speak to the invigilators who can arrange Health Service access

Our Reference book

- Ian Gorton, *Essential Software Architecture,* 1st edition, Springer, 2006.
- You will need to consult this regularly
- A copy is in SciTech library reserve
- Other books and readings will be provided via the unit website



Software

- Microsoft Visio, Word etc
  - Licensed software
  - Installed on the lab PCs
  - Can be used for writing reports, drawing diagrams, etc
- JEE 5
  - Free download from java.sun.com
  - Installed on lab PCs
  - Used for running sample code or assignment
- Other Java-based technologies for integration
- Use of Microsoft .NET technologies

Academic Honesty

- Okay to discuss ideas and problem approaches with people in other groups, or outside the class
- All work must be the creation of the members of the group
  - Except as acknowledged
- Plagiarism will not be tolerated under any circumstance!
- Academic honesty form needs to be signed for assignment and handed in.
- Know the official School and University policy

Group dynamics

- Group must organize itself
  - Arrange internal communication and meetings
  - Set internal deadlines
  - Follow up if deadlines are missed
  - Have fall-back mechanism
  - Have dispute resolution mechanism
- Let unit coordinator know if there are problems (unresolvable disagreements, member who doesn't contribute, etc)
- All members get the group mark, for the assignment
- Remember: groups will change each assignment

Questions?