Software Engineering with Objects and Components

Stuart Anderson

Room 1610, JCMB, KB

0131 650 5191, soa@inf.ed.ac.uk

Software Engineering

- Software Engineering Institute motto:
  - *The right software. Delivered defect free, on time and on cost, every time.*
- Software Engineering studies:
  - How to make software that is fit for purpose.
  - "fit for purpose": good enough – functionally, non-functionally, meets constraints of the environment, law, ethics and work practice.
  - How to meet time and financial constraints on delivery.
- We still fail too often, see:
  - http://wwwzenger.informatik.tu-muenchen.de/persons/huckle/bugse.html

Administration

- Look at the webpage:
  - http://www.inf.ed.ac.uk/teaching/modules/seoc1
- Assessment: 25% coursework, 75% degree examination.
- Coursework: in small teams (approx 4 people), two deliverables equally weighted.
- Software: Java and Argo/UML

An Example: Patriot Missile

- On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to track and intercept an incoming Iraqi Scud missile.
- The Scud struck an American Army barracks, killing 28 soldiers and injuring around 100 other people.
- The system's internal clock was multiplied by 1/10 to produce the time in seconds.
- The binary expansion of 1/10 is 0.0001100110011001100110011001100.... It is not representable precisely as a decimal.

Patriot Continued …

- The 24 bit register in the Patriot stored instead 0.00011001100110011001100
- An error of 0.00000000000000000000000011001100... binary, or about 0.000000095 decimal.
- The unit had been running for approximately 100 hours so the internal clock was about 0.34 of a second out (a scud covers around 500 metres in that time).
- The disagreement between the Patriot's clock and the radar clock was sufficient that the missile could not be detected.
- The problem was exacerbated by poor maintenance.

Models supporting SE

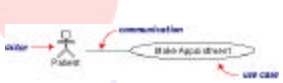
- UML provides a range of graphical notations that capture various aspects of the engineering process.
- Provides a common notation for various different facets of systems.
- Provides the basis for a range of consistency checks, validation and verification procedures.
- Provides a common set of languages and notations that are the basis for creating tools.

Software Engineering

- We will study the following areas:
  - Software Requirements: the activities involved in gaining an accurate idea of what the users of the system want it to do.
  - Software Design: the design of a system to meet the requirements.
  - Software Construction: the realisation of the design as a program.
  - Software Testing: the process of checking the code meets the design …
  - Software Configuration, Operation and Maintenance: major cost in the lifetime of systems
- These are the essential activities – how we deploy effort and arrange these activities is part of Software Engineering Process.

UML: Use Case Diagrams

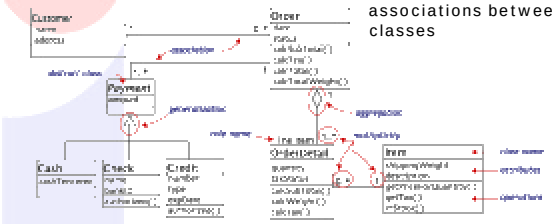
Used to support requirements Capture and analysis.



Show the actors' Involvement in System activities.

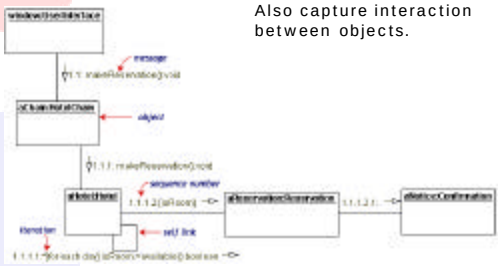
UML: Class Diagrams

Capture the static structure of systems associations between classes



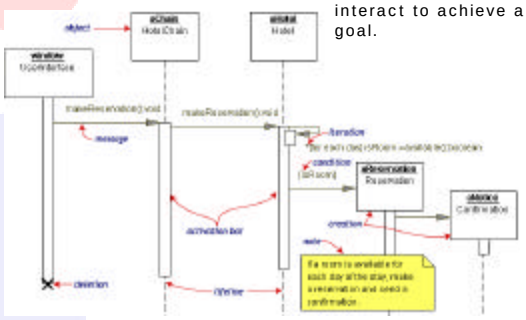
UML: Collaboration Diagrams

Also capture interaction between objects.



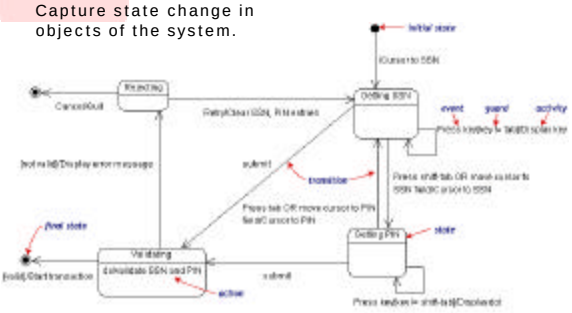
UML: Sequence Diagrams

Capture how objects interact to achieve a goal.



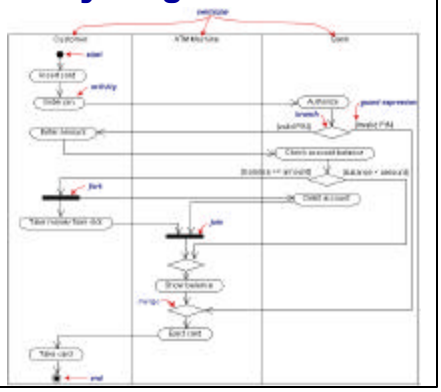
UML: Statechart Diagrams

Capture state change in objects of the system.



UML: Activity Diagrams

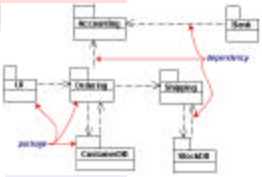
Capture the workflow in a situation.



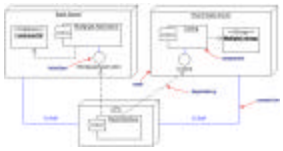
Things to Do

- Read the introduction to UML referenced off the course resource page.
- Buy the book: UML, Schaum's Outline Series, Simon Bennett, John Skelton and Ken Lunn, McGraw-Hill, London, ISBN 0-07-709673-8.
- Read chapters 1 and 2 of the UML book.
- Look at the practical page off the course page.

UML: Other Diagrams



Package



Component and Deployment

Object