CS6321
Model-Based Software Development

Dr. John Herbert

1.78 Western Gateway Building
j.herbert@cs.ucc.ie

Overview

Model-based software development (model driven engineering/ model driven development)

- Software development making systematic use of models as the primary engineering artefacts throughout the software lifecycle

Best known notation for models: UML (Unified Modelling Language)

- An open, single standard
- Replacing many diverse modelling notations

Motivation

Code as the main artefact

- Does not convey the important higher level aspects of software, such as
  - analysis
  - architectural design

Models

- Document these other aspects of design
- Allow designs to be communicated, analyzed, debugged prior to coding

UML Overview

A modelling notation (like a programming language) is not a solution, just a medium to express a solution

One can, however, have a process or method based on a modelling notation, such as the UML

The process can try to provide best practice method(s) for software development …addressing stages from requirements to design, implementation, testing, deployment, maintenance, evolution etc.

UML Overview

The Rational Unified Process is original ‘big’ UML-based process (devised by several object-oriented development gurus)

Several attempts at smaller, more agile versions of this process

Larman’s lightweight process will be our choice

UML Overview

The notation (UML) and lightweight agile process are used as the framework for modelling best practice

The use of a modelling notation is much less important in the world of agile

The important aspect is not the notation but in the way of thinking about software, the abstract design of software (i.e. models not focused on code)

UML Overview

- The main goal of modelling is communication

- Different ways to use models include (Fowler):
  - To sketch a system
  - Forward engineering (cf. model driven engineering)
  - Reverse engineering (e.g. to document a design)

UML Overview

The notation (UML) and lightweight agile process are used as the framework for modelling best practice

We also look at best practice encapsulated in

- Software design patterns
- Software Architectures

We also look at modelling non-functional properties

- Important aspects of systems
- Not covered adequately in conventional UML based methods

Initial Lectures Overview

Initial introduction to

- Software process
- Agile
- System modelling

(based on standard Software Engineering textbook by Sommerville)