Software Engineering Mathematics and Specification *or*

Cartoon from NASA Larc collection of FM humour

Topics

This part of CSCU9P6 will include:

- Motivation: why use formal specification?
- Introduction to the Alloy notation and tool
- Alloy basics: sets, relations, logic
- Creating static and dynamic models of software systems and exploring their properties using the Alloy Analyser tool.
- Assessment: practical checkpoints, exam question

Textbooks and Other Resources

- Daniel Jackson, *Software Abstractions: Logic, Language and Analysis*, revised ed, MIT Press, 2012 (textbook on Alloy)
- http://alloy.mit.edu/alloy/ The Alloy website at MIT, where you can download Alloy for free.
- http://www.doc.ic.ac.uk/project/examples/2007/271j/suprema_on_alloy/Web/index.php An excellent, short, online tutorial on Alloy created by students at Imperial College
- http://en.wikipedia.org/wiki/Formal_methods - a useful, high-level overview of formal methods in general

What is Formal Specification?

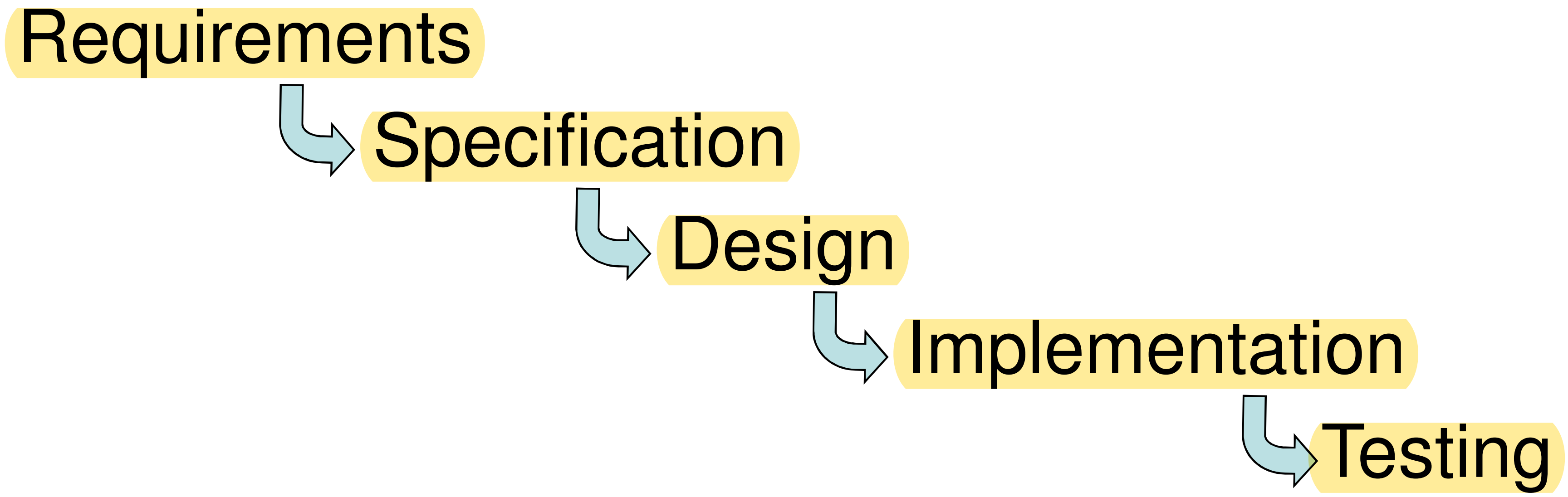
- **Formal specification** belongs to a group of software development techniques called "*formal methods*"
- The word "***formal***" means that these techniques are precise and unambiguous. They use notations based on mathematics and logic.
- "Formal ***specification***" means using a formal notation to create an abstract model of a system and describe its required properties.

Why use Formal Specification?

- Formal specification is not (at present) widely used for general software development, as it is seen as too costly and time-consuming.
- However, its use is increasing in the area of “critical” software, where the cost of failure is very high.
- Examples: air traffic control, railway signalling, spacecraft systems, medical control systems, smartcard programming…
- Formal methods are also used in **hardware** development (not covered in this module)

Why use Formal Specification(2)

- Here is a very simplified view of software development :



Requirements

Specification

Design

Implementation

Testing

- Errors introduced at an early stage cost the most to fix.
- A formal specification helps provide early detection and correction of errors.

Advantages of “formal” notation

- Formal specification uses a notation that is based on **mathematical formulas**. This has many advantages:
- **Precision:** The meaning of a formula is unambiguous.
- **Conciseness:** Formulas are much more compact than prose.
- **Abstraction:** We can deal with the complexity of large systems by hiding details, and focussing only on essentials.
- **Language independence:** Mathematical language is not tied to any particular programming language.
- **Allows logical analysis:** Modern formal methods provide automated / interactive analysis of the logical properties specification via tools such as model-checkers and theorem-provers.

Pause for thought

- Do we really need special notations for specification? Isn’t English good enough? Consider the following problem:
- [Lightfoot] An annual weekend event begins on the Friday evening and finishes on the Sunday afternoon. The date of the event is specified as “the last weekend in September”. What is the date of the Friday on which the event begins if the last day of September (30th) in that year is:
  - a Monday?
  - a Sunday?
  - a Saturday?
  - a Friday?
- Suggest a better specification of the date of the event.

The Alloy Specification Language

- Alloy is part of a new generation of “lightweight” formal methods supported by easy-to-use automated tools.
- The design of Alloy was heavily influenced by an earlier, well-established language called Z.
- Z has been widely used both in academia and in industry (by companies including IBM, Logica, INMOS, PraxisHIS, Adelard…) for formal modeling of safety-critical software.
- The Alloy Analyser tool was inspired by the SMV model-checker, a tool for automatically analysing and verifying models of computer hardware.

Alloy Overview

- The Alloy language uses an ASCII-based syntax (unlike Z).
- The basic concepts used are **sets**, **relations**, and **first-order logic**.
- Alloy is used to build an abstract description of a system and explore its properties.
- Alloy specifications are not executable. They describe WHAT the system must do, but not HOW to do it.
- The Alloy Analyser tool provides graphical visualizations of the specified system and can be used to check if the system has certain desired properties.

Alloy Example: Boarding an Aircraft

- We are given the following situation:
  - Passengers are boarding an aircraft
  - There is a maximum capacity of passengers that can fit into the aircraft
  - For safety reasons, this capacity must **not** be exceeded
- [Start up Alloy and build the Aircraft Boarding example].

Points About Alloy

- Alloy is a **model-based** notation.
  - A system is modelled by using mathematical concepts to represent its state and its behaviour

- Alloy is a language and tool, not a rigid methodology:
  - Alloy is flexible, and can be used in different ways for different purposes.
  - Alloy provides abstraction: we can choose whether to include or omit details.
  - All this flexibility can be confusing at first. We shall learn by looking at examples.

- Important: Alloy is **not** a programming language.
  - Alloy specifications are usually much more abstract than code. They cannot be executed like a program.

What shall we do next?

- Review the mathematical concepts that underlie Alloy
  - sets (today’s lecture)
  - relations and functions
  - mathematical logic

- Using examples, learn how to use the Alloy language to describe simple systems.

- Learn how to use the Alloy Analyzer to explore the logical properties of the systems we have described in Alloy.