Subject Knowledge

Aims This module aims to give students a hands-on experience with methods, languages and software tools used in industry for the specification and design of a wide range of systems (software, hardware, business process/workflow, embedded systems, etc.).

Learning Outcomes We will study two different type of specification languages that are often used in conjunction for describing different aspects of systems: the notation Z, which is suitable for describing transformational aspects, i.e. how the actions of a system operate on data; and the process language CSP, which allows to describe the reactive behaviour of systems, i.e. the processes as part of which actions are executed and the communication channels through which processes communicate.

Methods Class sessions, tutorials and practical sessions together with course notes, recommended reading, worksheets, printed solutions, and some additional hand-outs.

Assessment Assessed coursework, traditional written examination.

Skills

Aims To teach students to write system specifications and problem solving skills.

Learning Outcomes Students will be able to: write system specifications at various levels of abstraction; to solve abstract and concrete problems (both routine seen, and simple unseen).

Methods Class and practical sessions together with worksheets

Explanation of Prerequisites No specific knowledge is required, but some familiarity with the notation of Sets and Predicate Logic as well as general program design skills will be helpful.

Course Description We will study and practice with two different type of specification languages that are often used in conjunction for describing different aspects of systems: the notation Z, which is suitable for describing transformational aspects, i.e. how the actions of a system operate on data; and the process language CSP, which allows to describe the reactive behaviour of systems, i.e. the processes as part of which actions are executed and the communication channels through which processes communicate.

Both languages come with various tools to verify properties of the specifications and to help in the refinement process from specification to program.

Detailed Syllabus

Background: Lack of formalism in software development can lead to a loss of precision and correctness in the resulting software. Formal specification languages and their tools can improve this situation.

Z is a state based language for specifying systems. We will study: Notation of proposition and predicate logic, sets, relations and schemas; simple refinement; object-Z; producing Z code with LaTeX; type checking Z code with the tool FUZZ.
**Communicating Sequential Processes:** Notation, semantics, refinement, verification; tool support ProBe and FDR.

**Reading List and Tools**

**Material on Z**

- J. M. Spivey. *The Z Notation: a reference manual*, Prentice Hall, 2001:  
  http://spivey.oriel.ox.ac.uk/mike/zrm/


**Material on CSP**

- C.A.R. Hoare, *Communicating Sequential Processes (CSP)*, Prentice Hall, 1985:  


**Tools**

- **Fuzz:** http://spivey.oriel.ox.ac.uk/mike/zrm/

- **Probe and FDR2:** http://www.fsel.com/software.html

**Resources** Study guide, slides, website, worksheets, lecture and tutorial rooms with data projector, whiteboard and OHP computer laboratory access.

**Module Evaluation** Course questionnaires, course review.