SWEN30006
Software Modelling and Design

Subject Overview

Teaching Staff

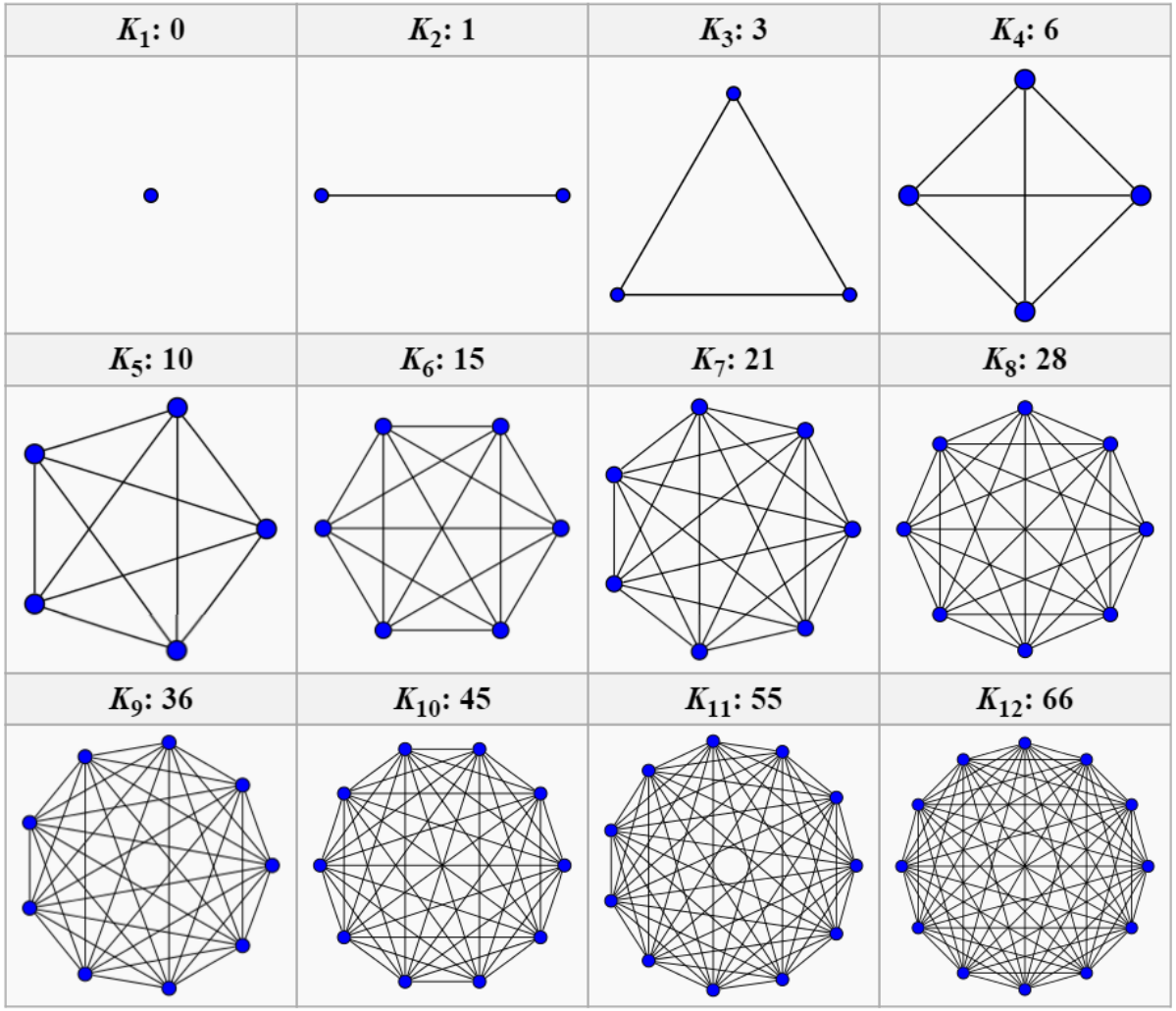
- *Coordinator/Lecturer*
  - Philip Dart (philip.dart@unimelb.edu.au)
- *Tutor in Charge*
  - William Tio (william.tio@unimelb.edu.au)
- *Tutors*
  - Jessica Tobagus
    jessica.tobagus@unimelb.edu.au
  - Sehrish Kanwal
    kanwal.s@unimelb.edu.au
  - Eileen O'Callaghan
    eileen.ocallaghan@unimelb.edu.au
  - Eduardo Oliveira
    eduardo.oliveira@unimelb.edu.au

Aims and Objectives

- The aim of the subject is to teach you about ‘Software Modelling’ and ‘Software Design’.
- Software Design is all about *purposefully* choosing the structure and behaviour of your software system.
  - The behaviour is all about how your systems responds to inputs and events, and choosing how parts of the system collaborate to achieve the goals of the system.
- Software Modelling is the creation of tangible, but abstract, representations of a system so that you can communicate your design ideas, critique them and explore viable alternatives

Potential Complexity of Software

Software is abstract: relationships are not limited by physical proximity.



K_N: N*(N – 1)/2 undirected edges.

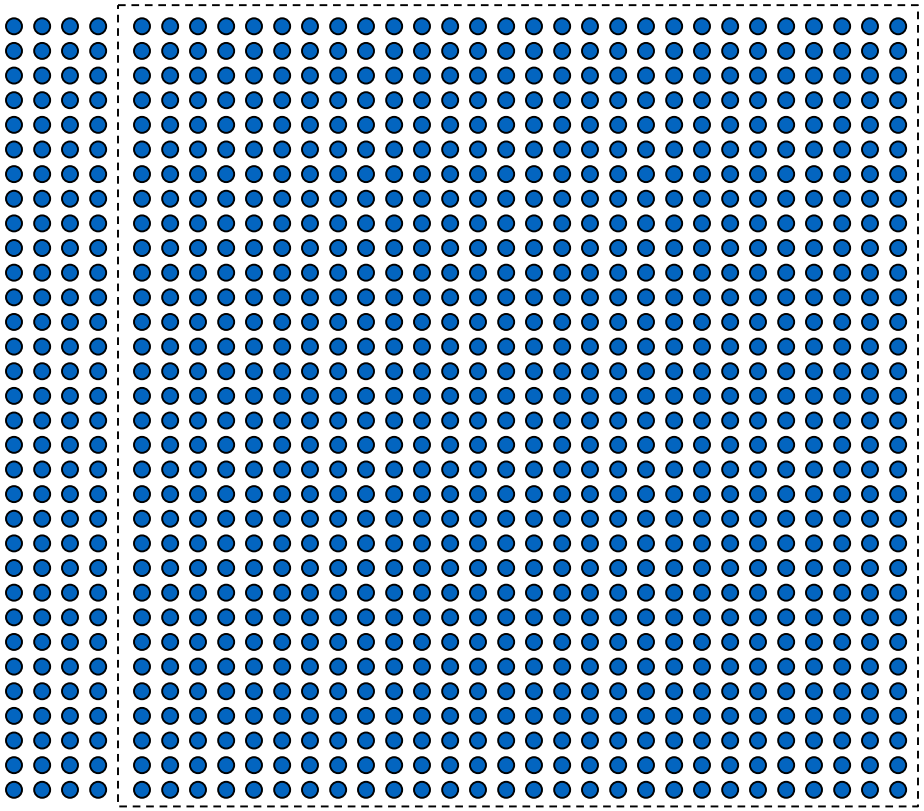
https://en.wikipedia.org/wiki/Complete_graph

Case Study: Helicopter Flight Control System

Military Helicopter On-board Software:

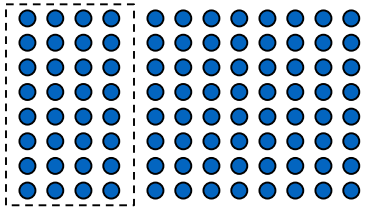
- Navigation System
- Weapons Control System
- Communications System
- …
- Flight Control System (500 KLOC)
  - Build1 (150 KLOC)
    - Subsystem (15 KLOC - 10% of Build 1, 100 Ada packages)
      - Cluster (32 Ada packages)

Case Study: Helicopter Flight Control System



Build 1

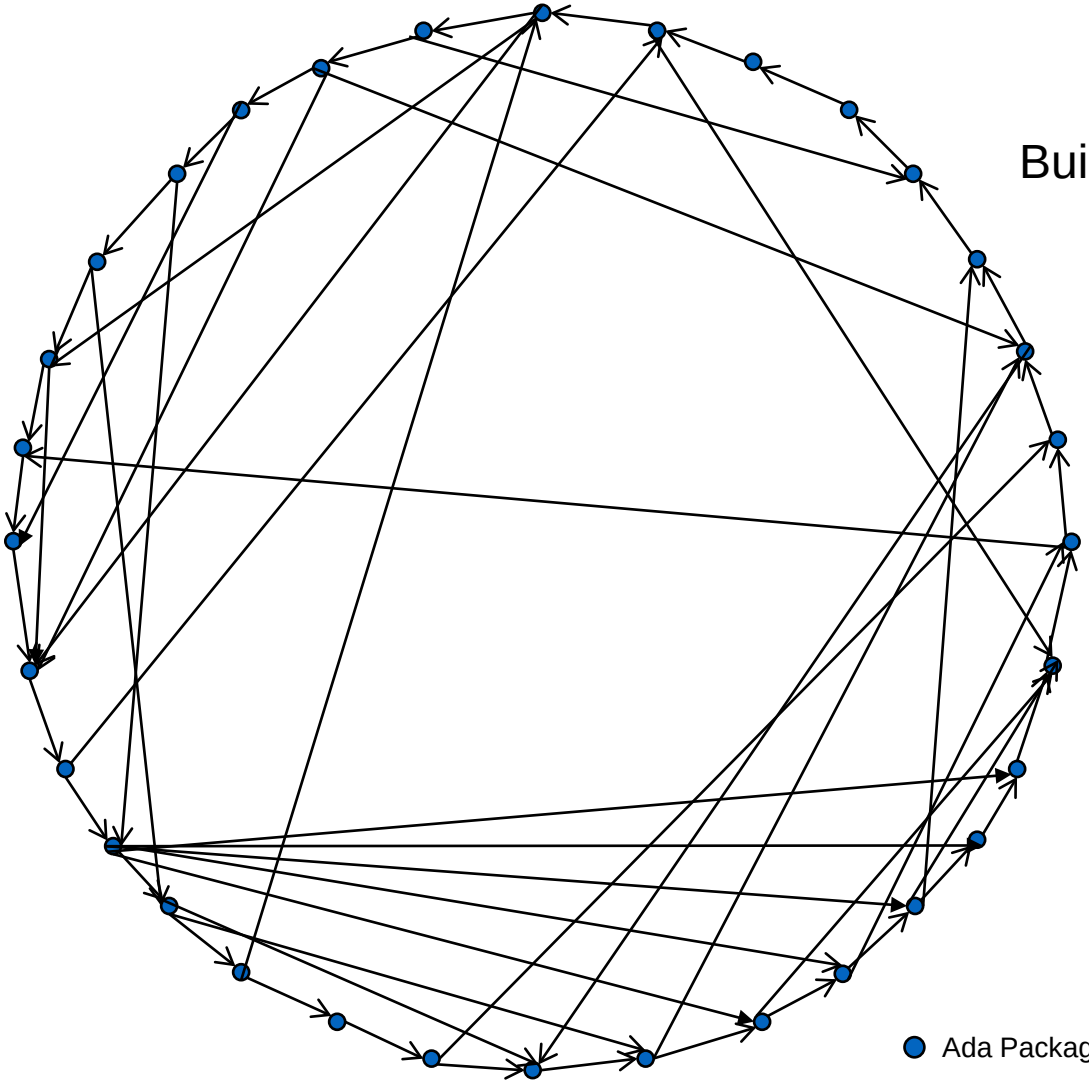
Build 1 Subsystem cluster



Build 1 Subsystem

Ada Package ~150LOC

Case Study: Helicopter Flight Control System



Build 1 Subsystem cluster

Each package in the cluster depends on every other package in the cluster.

High coupling →
hard to understand →
poor design.

Ada Package ~150LOC

Case Study: Submarine CDS and Destroyer CDS

CDS: Combat Data System

1. Major defence company experienced in SE.
2. Awarded contract for Submarine CDS
3. Work on Submarine CDS commences
4. Awarded contract for Destroyer CDS
5. Work on Submarine CDS going very poorly
   - developers not sticking to design
6. Destroyer CDS solution
   - request design changes in writing

Lessons from the Case Studies

- Software complexity is a problem in practice, not just in principle
- Software developers tend to add to complexity unnecessarily if unconstrained
- Requiring developers to think through the implications of a structure/design change is important in maintaining a good design
- These issues are particularly critical in large projects

Software Modelling and Design

- The subject will focus on the object-oriented design method, with object-oriented modelling and modelling heuristics.
- UML (Unified Modelling Language) will be used as the primary modelling notation.
- Java will serve as the programming language to explore and validate the important design ideas.

Assessment

- **Project Work (40%)**
  - Workshops (10%)
  - Project in 3 parts (30%)
- **Final Examination (60%)**
  - A 2-hour end-of-semester written examination
- **Hurdles**
  - To pass the subject, students must obtain at least 50% overall, 20/40 in project work, and 30/60 in the final exam.
- **Plagiarism and Conduct**
  - Direct copying, cutting and pasting from websites or submitting other people's work as your own without referencing their input is plagiarism and not tolerated by the University. Plagiarism is considered academic misconduct and can result in severe penalties.

Workshops (10%)

- The workshops are key to your learning in this subject.
- Workshops start in week 2
- You must attend your assigned workshop.
- Workshops will present content as well as providing hands-on exercises.
- Attendees will get 1% for showing their tutor a credible attempt at the exercises at the end of the workshop, or by the start of the next one at the latest. (10%)
- Initial workshops (by week):
  2. Introduction to tools and the development environment
  3. Class modelling (in UML)
  4. Sequence/communication modelling (in UML)

Project (30%)

- Part A: Robot Mail Delivery (5%)
  - *Individual* submission
  - Demonstrate competence with Java and the basic toolset
- Part B: Metro Rail System (10%)
  - *Team* submission
  - Critique, improve and extend a more complex design
- Part C: Autonomous Car (15%)
  - *Team* submission with two components
    1. Proposed design and rationale
    2. Refined design, implementation, and reflection

Teams will be self selected through the LMS.

Getting help

- Textbook
- LMS
  - Lectures (slides and recordings)
  - Workshops and Project
  - Forums – staff will be monitoring and responding
- Tutors
  - Participate in workshop discussions
  - Ask questions during the workshops
  - Email workshop-related questions to your tutor
- Lecturer
  - Ask questions during or after the lectures
  - Meet me during the consultation times

Larman: Required Textbook

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition, by Craig Larman, Pearson Education Inc., 2005.

- You need access to the textbook (link on LMS).
- We will be following the textbook closely.
- Sections for reading from the textbook will appear week by week.

Additional references have been provided on the LMS.