



The move to object-oriented approaches

Dr. Jon Holt
Brass Bullet Ltd



Overview

- 1: Introduction**
- 2: OO concepts**
- 3: OO modelling**
- 4: Conclusions**



1: Introduction

- When people hear OO, they think of
 - programming
 - Java
- People may also think of
 - analysis
 - design
- However, the scope is wider than just software engineering
 - target system could be almost anything
 - forget “OO application”



Definitions

- **Object**

- “physical thing”

- “focus of thoughts or action”

- **Orientation**

- “position oneself according to one’s surroundings”

Object orientation is simply a way of thinking about things, in order to solve problems



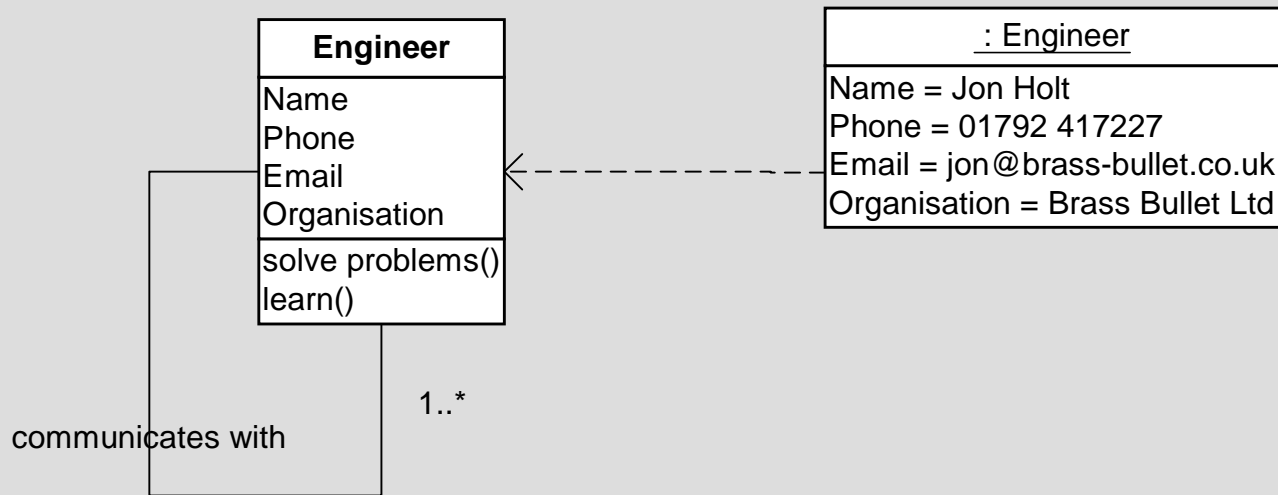
Origins of OO

- **Practical use derived from software**
 - Simula 67, Lisp, KEE, etc
- **Many concepts derived from “frames”**
 - properties and behaviour encapsulated into cognitive frames
- **Frames are derived from AI concepts**
 - psychology
 - cognitive science
- **AI tries to emulate human intelligence**
 - approach to problem solving
 - much derived from Descartes



2: OO concepts - classes, objects and encapsulation

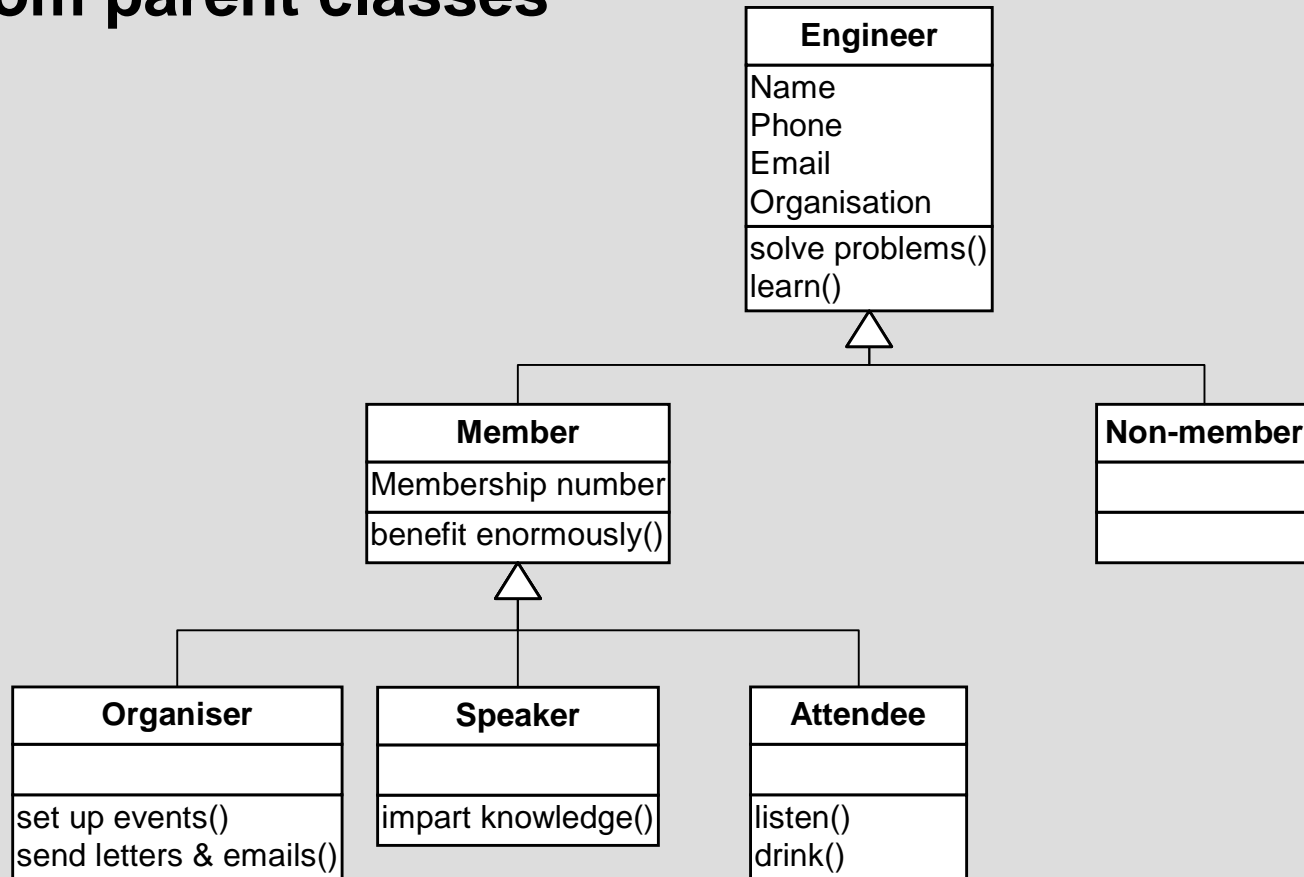
- Encapsulate data (its properties) and process (its behaviour)





OO concepts - inheritance

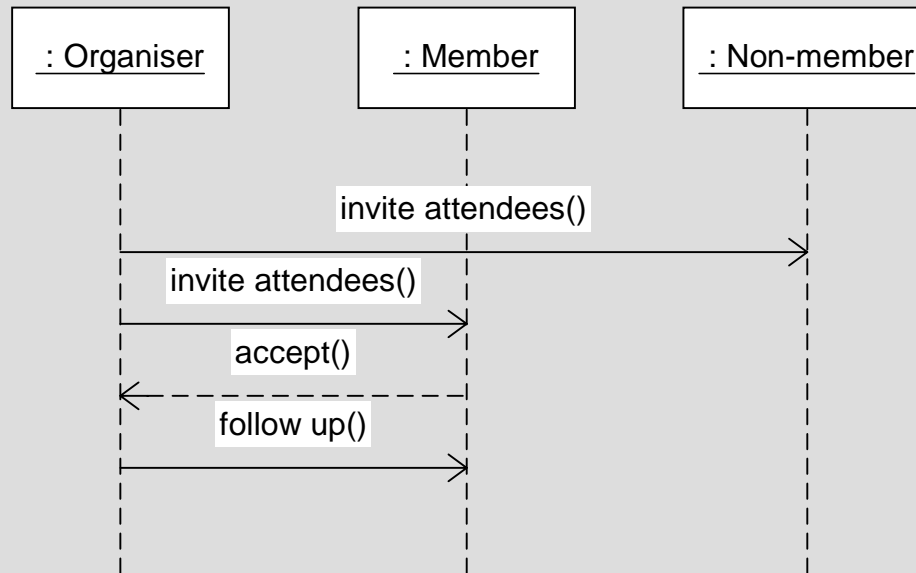
- Classes inherit properties and behaviour from parent classes





OO concepts - message passing

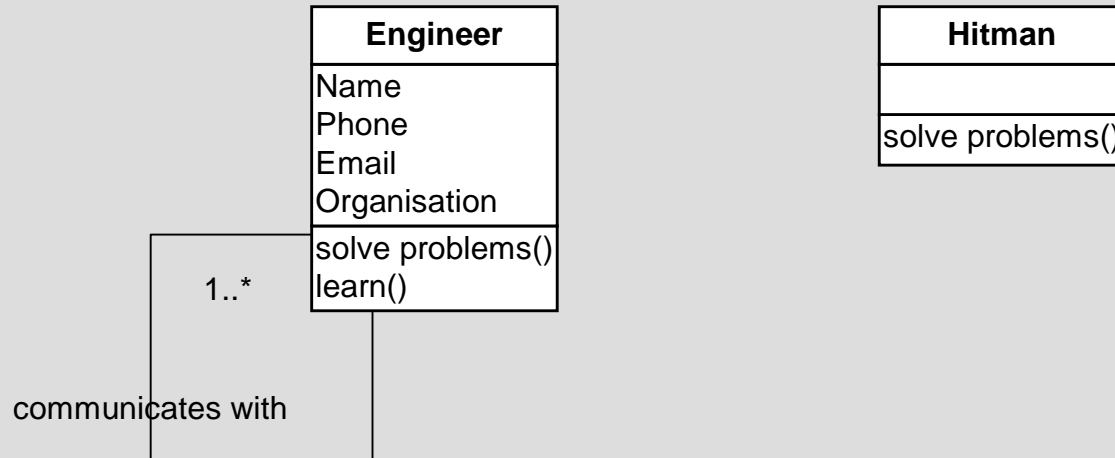
- Objects communicate by passing messages





OO concepts - polymorphism

- Use same expression to represent different operations





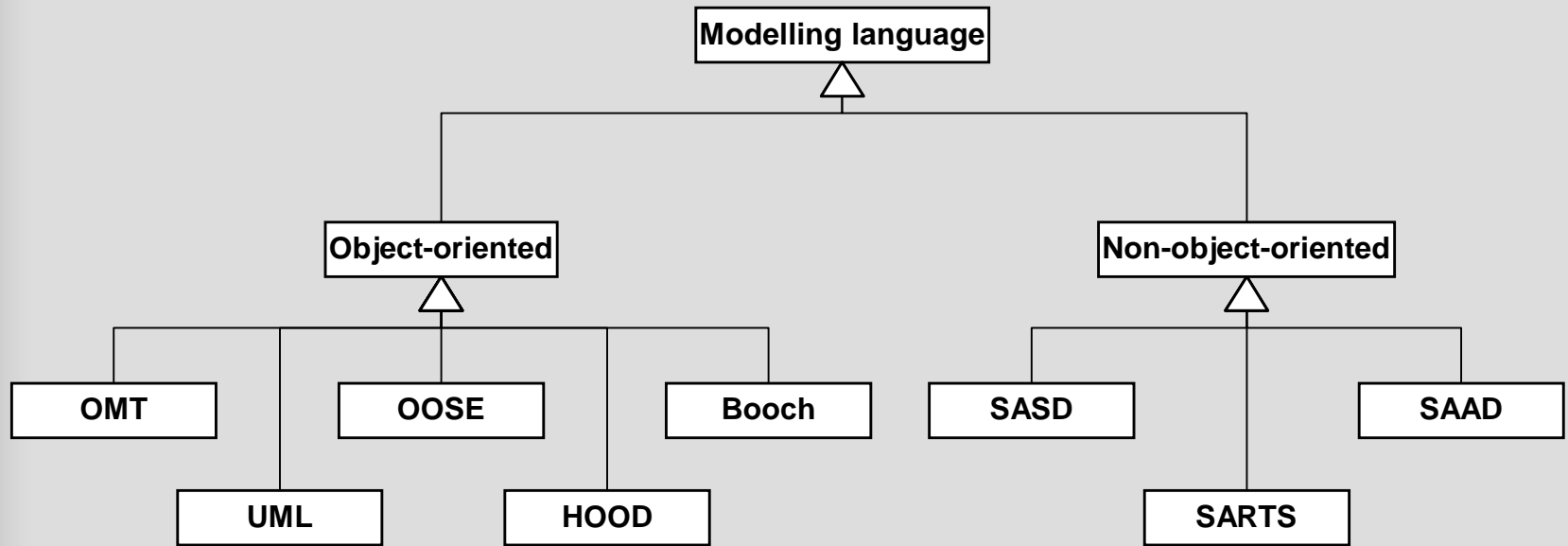
Why use OO?

- **Re-usability**
 - assemble new systems based on existing components
 - enforces modularity
- **Extensibility**
 - new classes/objects based on existing ones
 - saves re-inventing the wheel every time
- **Intuitiveness**
 - more intuitive than traditional techniques



3: OO modelling

INCOSE UK Autumn Assembly 2001





The unified modelling language (UML)

- **General purpose modelling language**
 - effective communication
 - get ideas out of your head and onto paper
- **Well supported**
 - tools
 - texts
- **The UML will:**
 - help effective communication
- **The UML will not:**
 - solve all your problems

becoming too prevalent to be ignored

Example systems applications



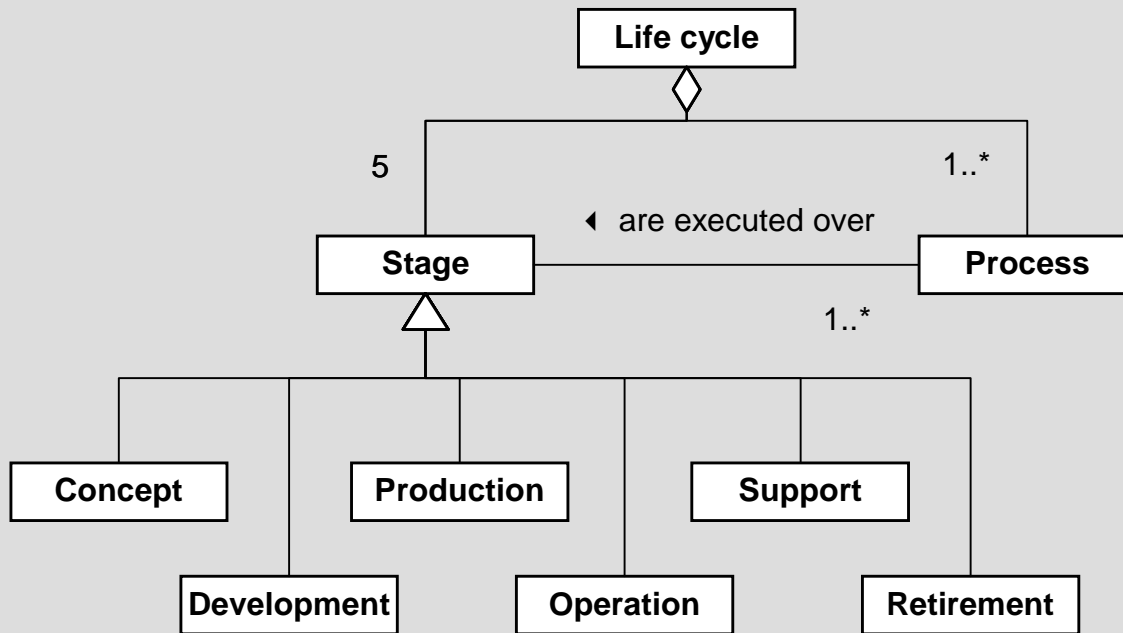
- **Typical lifecycle activities**
 - requirements engineering
 - analysis and design
 - testing (V&V)
- **Process modelling**
 - defining new processes
 - analysing standards
- **Business modelling**
- **Project management**
 - planning tasks and resources
- **Teaching and training**

In fact, anything where clear, unambiguous communication is required



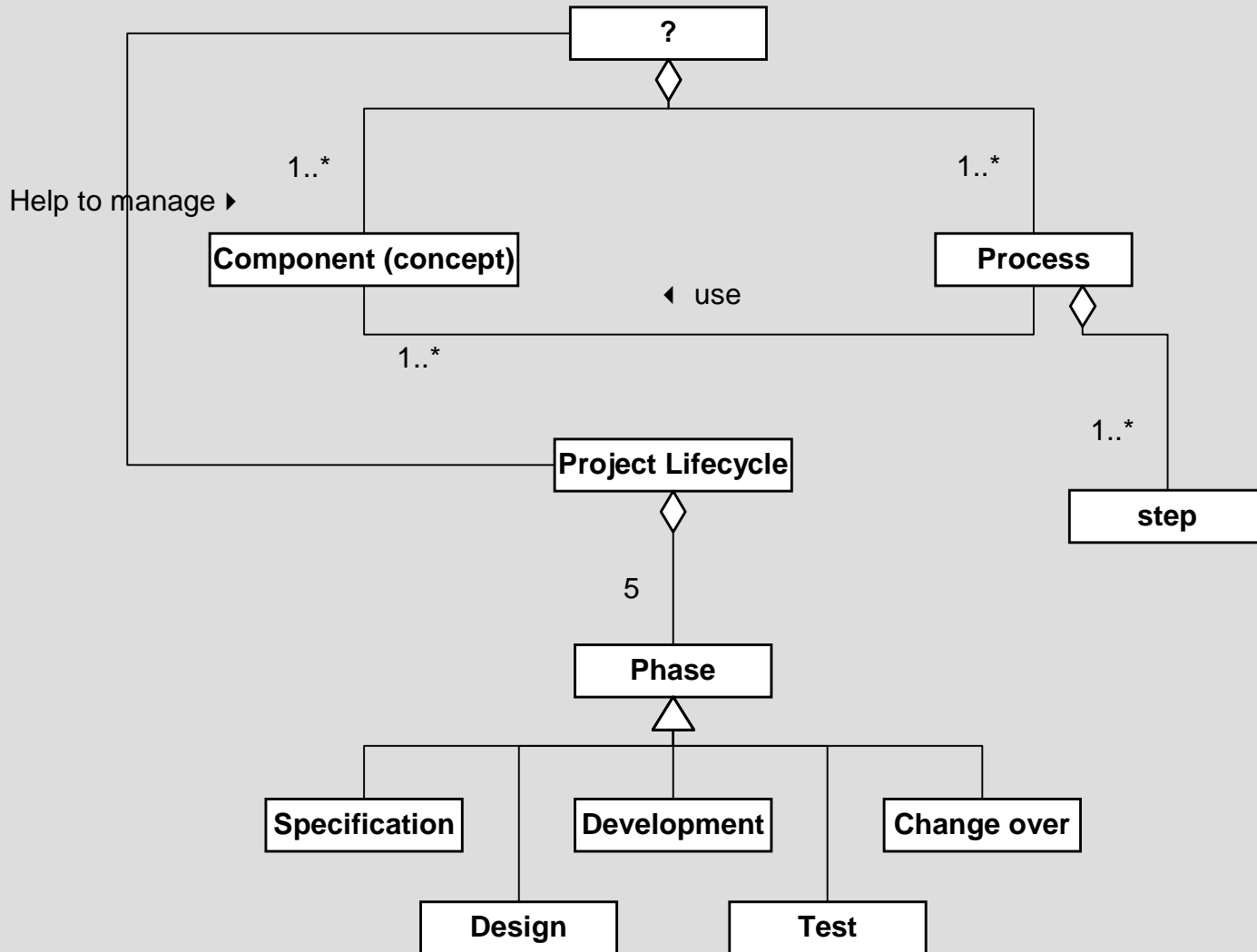
Example application

INCOSE UK Autumn Assembly 2001





Example application



INCOSE UK Autumn Assembly 2001

4: Conclusions



- **OO is:**
 - a way of thinking
 - intuitive
 - effective
 - useful for many applications
 - realisable using existing techniques and tools
- **OO is not:**
 - magic
 - evil
 - just a way of programming