UML Case Study – 1
Use Cases

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

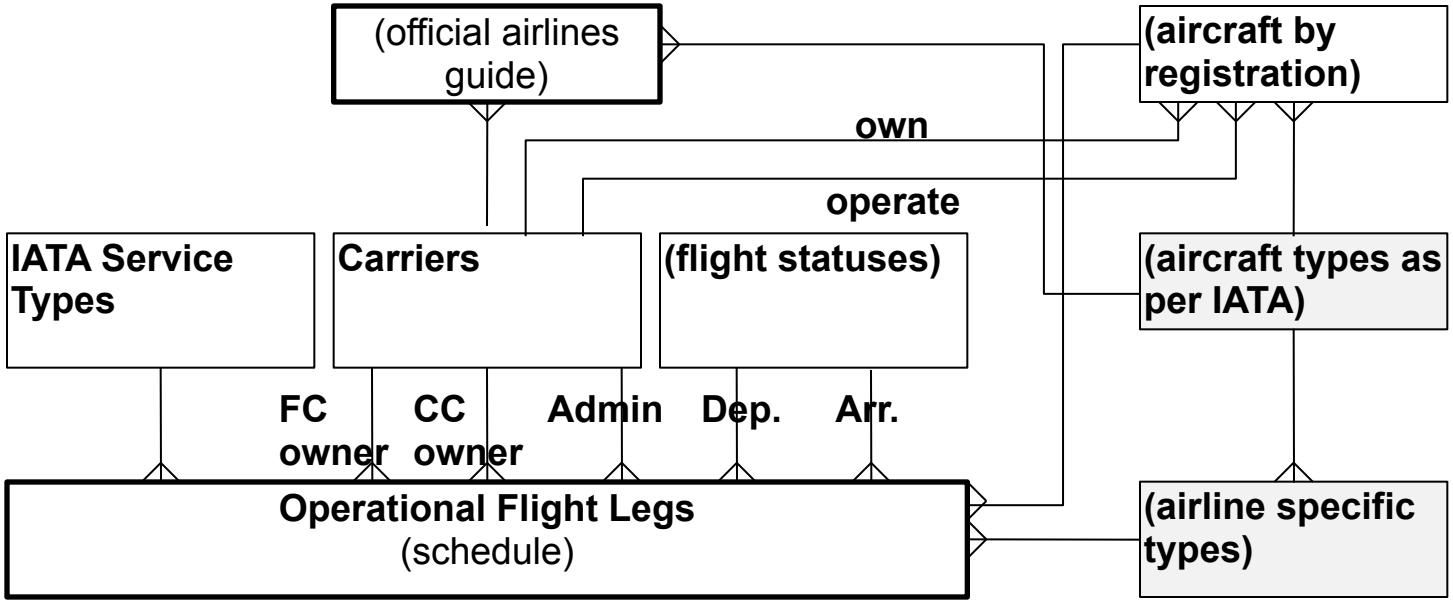
Modelling Objects

- Aim is to identify the main elements of data that the system must support
- Two main approaches
  - Entity Relationship Modelling (data modelling)
  - Object Modelling (producing class diagrams)
- Both techniques can be good for communication
- Both work best with tool support

Entity Relationship Modelling

- ER Diagrams used to develop relational databases
- Describe entities, their attributes and relationships between them
- Broadly speaking:
  - Entities become relational database tables
  - Attributes become columns in those tables
  - Relations are links between rows in tables
- Example Tools – Oracle Case Designer / Designer 2000

Example ER Diagram



Pros & Cons of ER Diagrams

- Good Thing – They are closely related to database tables
  - Can automatically generate table & index definitions
  - Automatic creation of validation routines
- Bad Thing – They are closely related to database tables
  - Sometimes need to create entities (e.g. for many-to-many)
  - Not good at hierarchies
  - Doesn't address functionality

Object Modelling

- Modelling objects using class diagrams
  - Often in parallel with other types of diagram
- A combination of :
  - Data modelling (as for a database)
  - Functional analysis (what functions are needed)
- Good for communication
- Best with tool support (e.g. Rational Rose)
- Not limited to OO design

First Cut Flight Planning Objects

Weather
latitude longitude flightLevel windSpeed windDirection temperature

Navigation
…
…

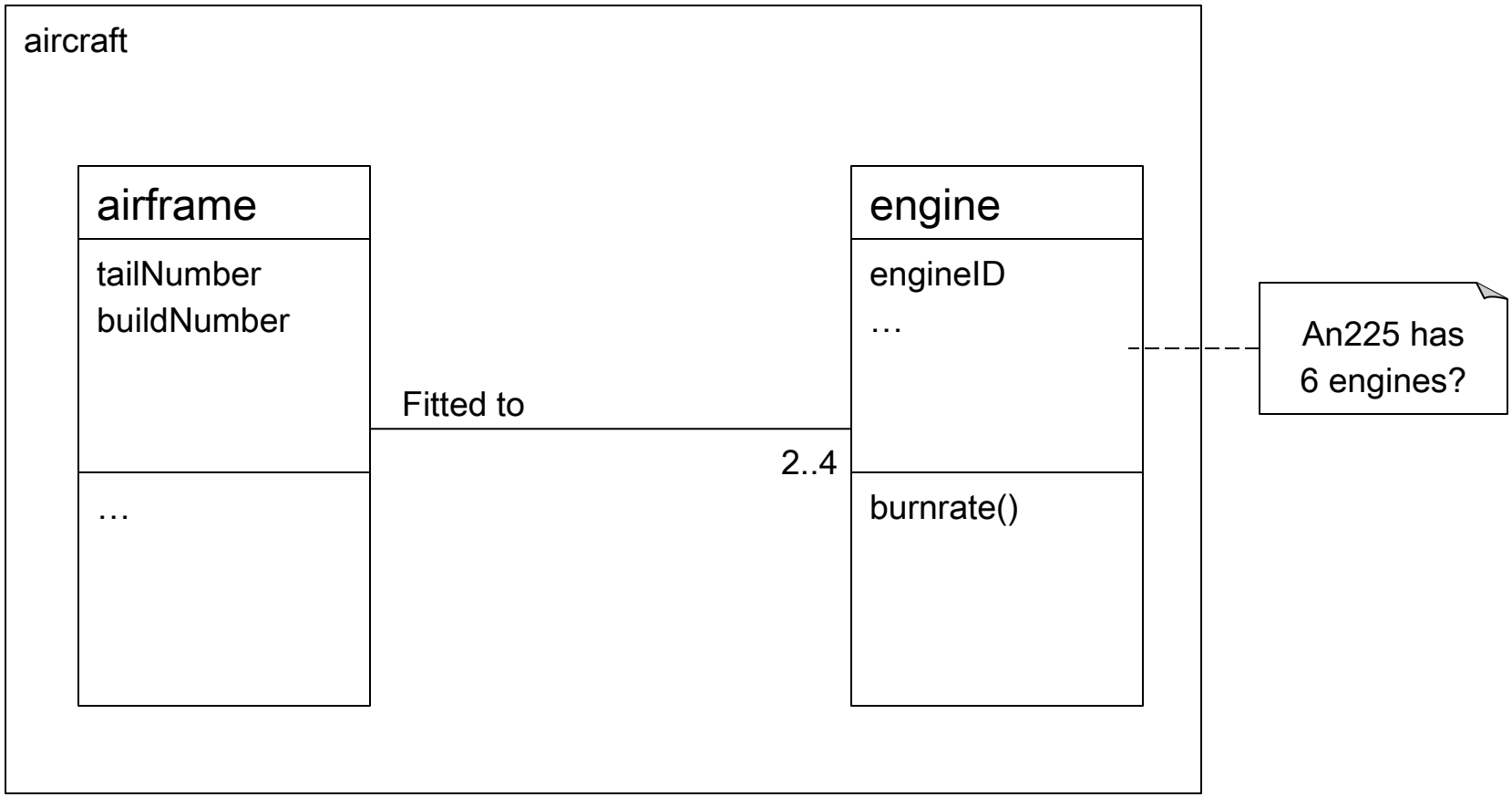
Schedules
…
…

Aircraft
…
…

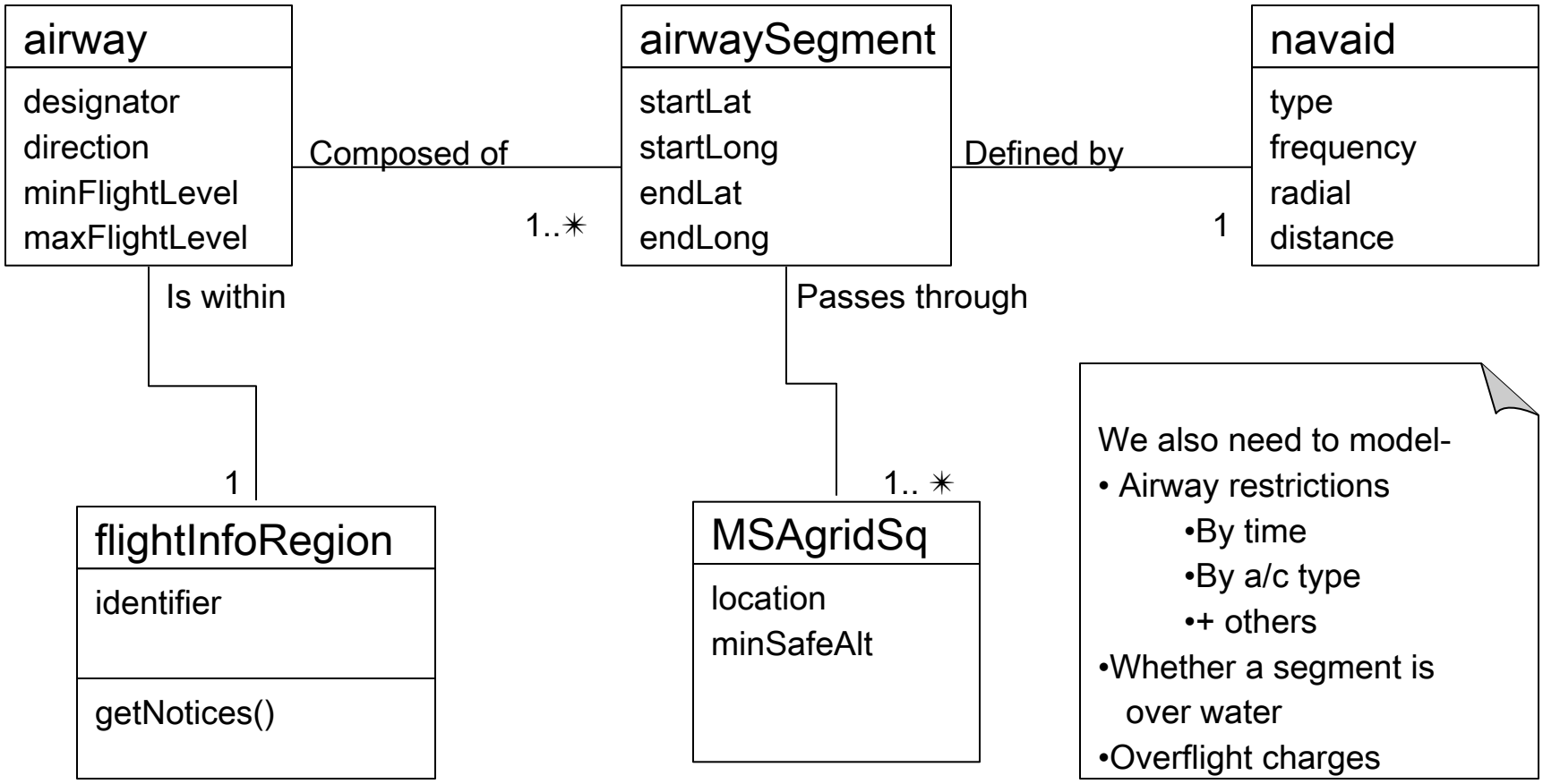
Admin
userName permissions …
setPermission() addUser() …

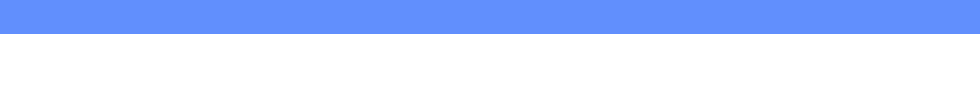
Stats
…
…

Aircraft Composition

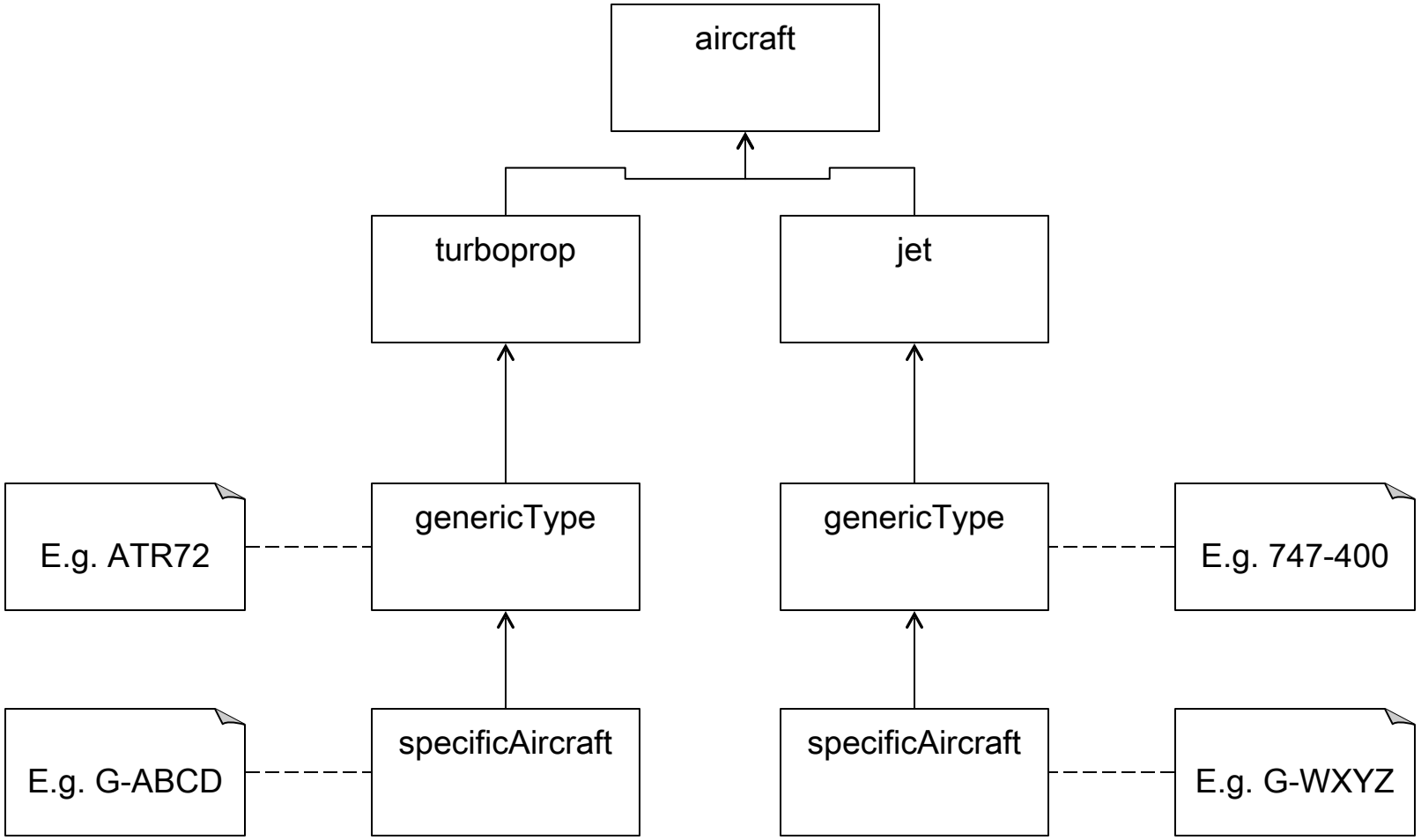


Navigation Data





Aircraft Inheritance



Are Rules Data or Function?

- **Recall last week – Discussion of constraints**
  - **Physical constraints (geography)**
  - **Legal constraints (laws, regulations)**
  - **Operational constraints ("soft" rules, preferences)**
- **Neither ER or Class diagrams make these explicit**
- **Why is this a problem?**
  - **q.v "Semi-constant" data, e.g. airport location**
  - **User (administrator) expects to change this data**
  - **Would also expect to change laws and soft rules**
  - **Especially for evaluating the effect of proposed changes**

Approaches to Rules

- **Model "expressions" in the database**
  - **Cumbersome, & not usually rich in expressive power**
- **Stored procedures (functions stored in database)**
  - **This is just code, but in a different place**
- **Code rules in a language closer to the user domain**
  - **E.g. declarative languages – LISP, Prolog**

- **Latest initiatives on Business Process Modelling**
  - **XML based languages to define business processes**
  - **Execution engines to carry out that logic**
  - **Typically for web services, e.g. EB-XML, BPML, WSDL…**

Conclusions

- **Class Diagrams *can* be useful for communication**
  - **Good for identifying related data and functionality**
  - **Object identification and decomposition still needs skill and intelligence**
- **There are some areas (rules / business logic) that are NOT addressed by any UML diagram**
  - **This is not just a failing of OO methodologies**

Next week – Collaboration Diagrams