Lecture 2

What is a Database and a Database Management System?

Week 1

Overview

- Application areas of Databases
- Why use Databases?
- Properties of Database systems
- Structure of a typical Database Management System
- Database People

Application areas of Databases

- Manufacturing, engineering, banking, education, government, health services, tax office, personnel, tourism and travel, information services (libraries, reference material), geographic information systems

Why use Databases?

Properties of traditional office information systems:

- bulky
- slow access to the data
- organised for one purpose and hard to use data in them for any other purpose,
- prone to inconsistencies, inflexible (hard to change the way data were stored)

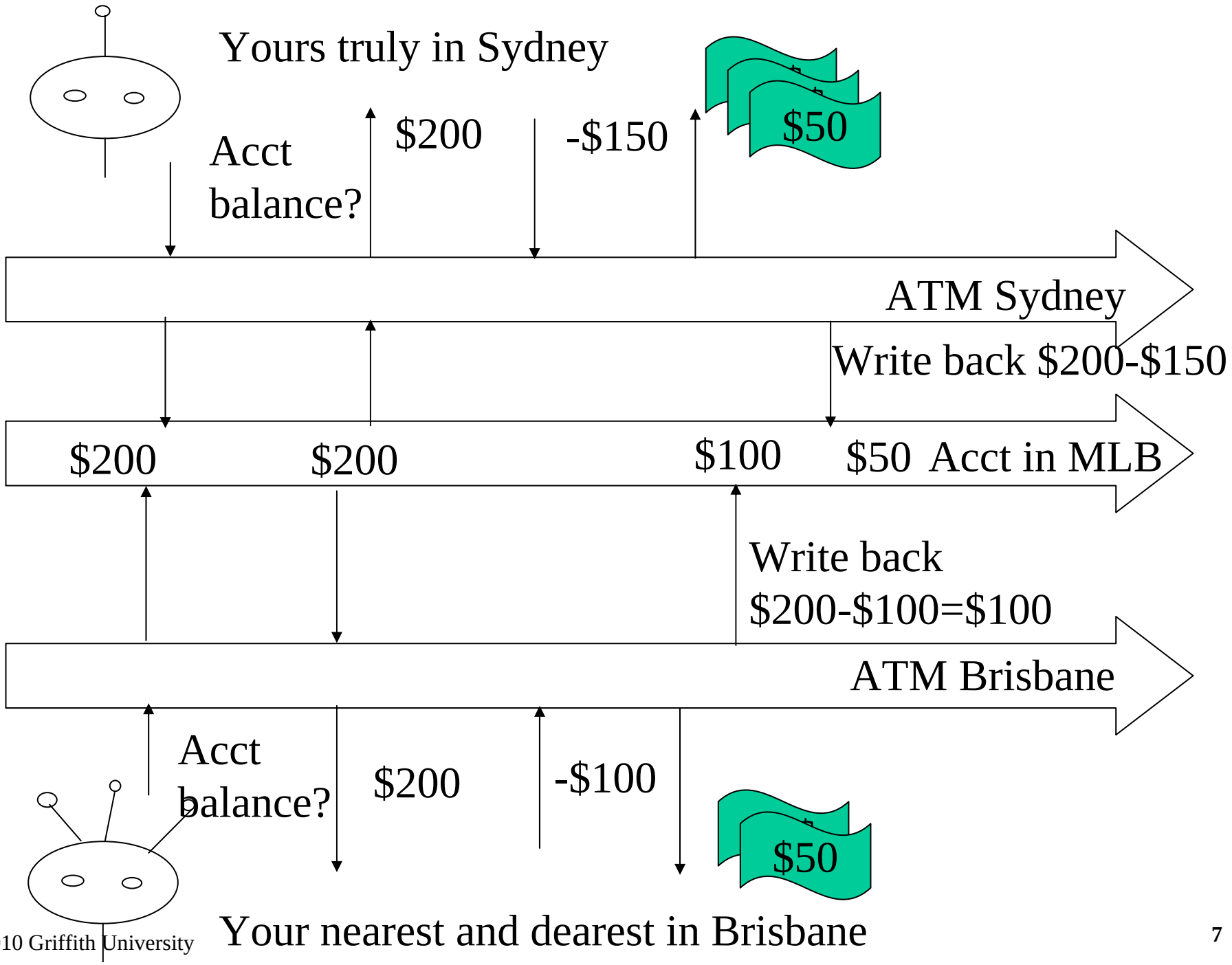
Why use Databases? (cont'd)

Properties of computer file system

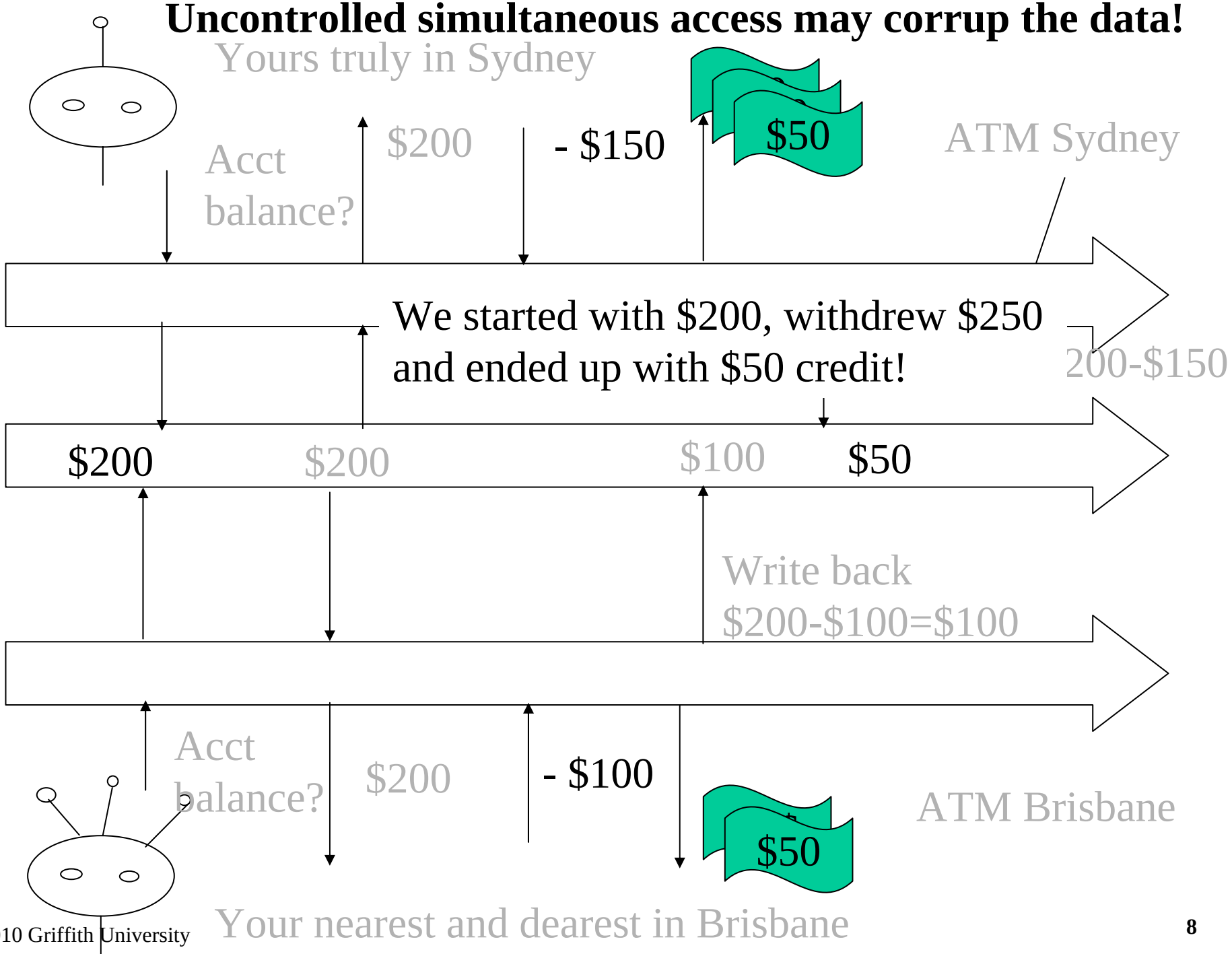
- collection of files in various formats
- for each application using the data, a separate storage format was designed
- each new business need saw new data file formats causing redundancy (and inconsistency)

Undesired Effects

- Data *redundancy* and inconsistency
- *Inaccessibility* (hard to access data to answer arbitrary queries)
- Data *isolation* (difficult to write new applications)
- *Unreliability* (simultaneous access may corrupt data)
- No data *security* (no or insufficient control of who can access which data)
- *Inflexibility* (hard to change storage structure without changing application program)



Uncontrolled simultaneous access may corrup the data!



Definitions

- A Database (DB):
  - is a logically *coherent* collection of data
  - *uniformly* represents data using a given *data model*
- A Database Management System (DBMS):
  - is a controlled collection of programs (software)
  - uses a given data model to define, store, retrieve, modify the data stored in it

Example DBMSes

- **IMS** (IBM, *hierarchical* data model)
- **IDMS2** (IBM, *network* [Codasyl] data model)
- **System R** (IBM, *relational* data model, research prototype, 1970s)
- **DB2** (IBM, *relational* data model)
- **Ingres, Tandem, Oracle, Access** (independent software vendors, *relational* data model)
- **Ontos, O2** (independent software vendors, *object-oriented* data model)

Some characteristics...

- Millions of lines of code
- Amount of data managed: gigabytes (~10^9) to terabytes (~10^{12})
- Enormous overhead
- Cost: mainframe ~$100,000 per year

Properties of Database systems

- Control of redundancy
- Uniform access to data (using standard data model)
- Use of query languages
- Efficient data access (optimization)
- Reliable concurrent access
- Integrity maintenance
- Multiple views of data available

ACID properties...

- Atomicity: any change on the data is either performed or not performed
- Consistency: the database always changes from consistent state to consistent state as a result of a database transaction

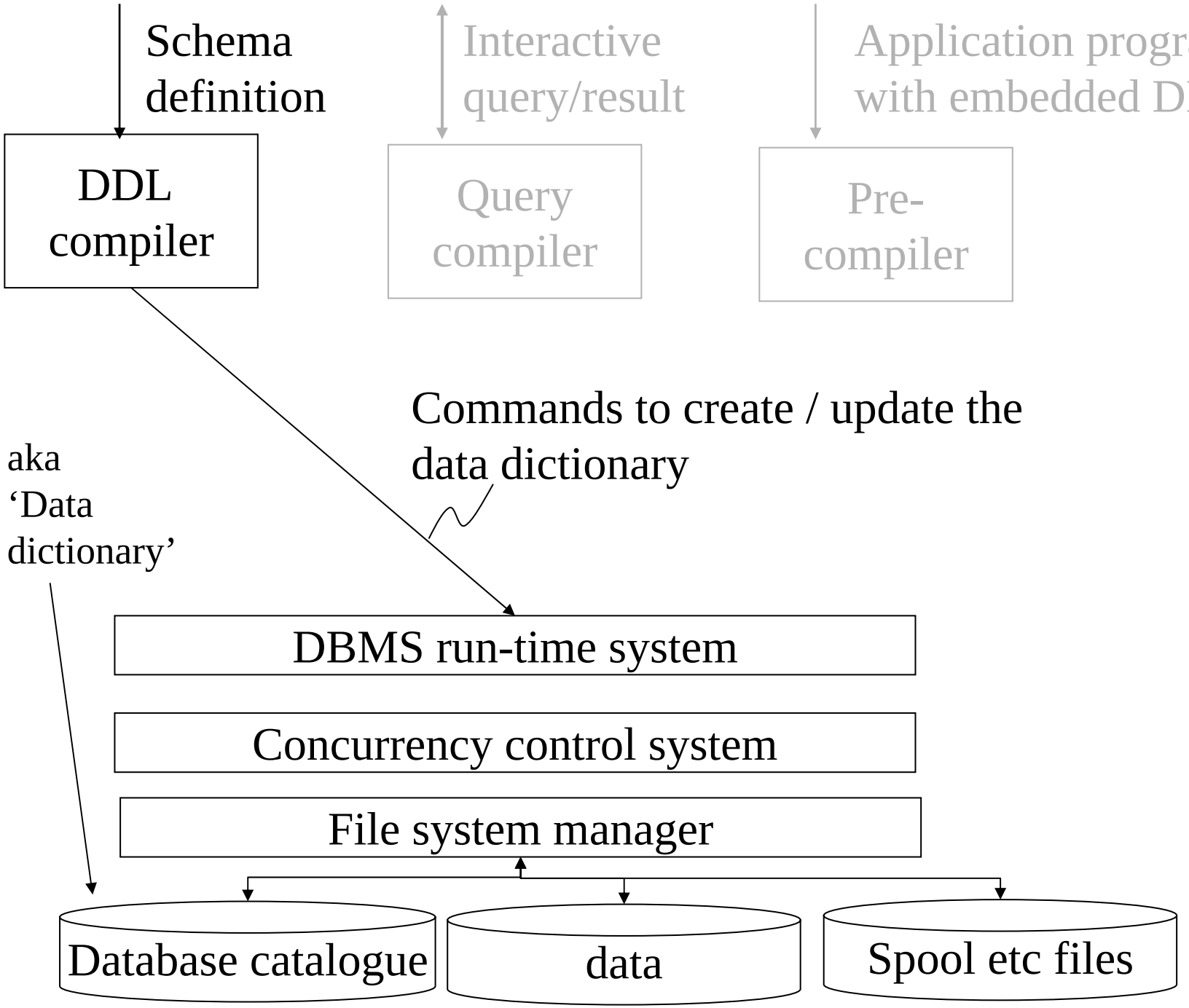
ACID properties… (cont'd)

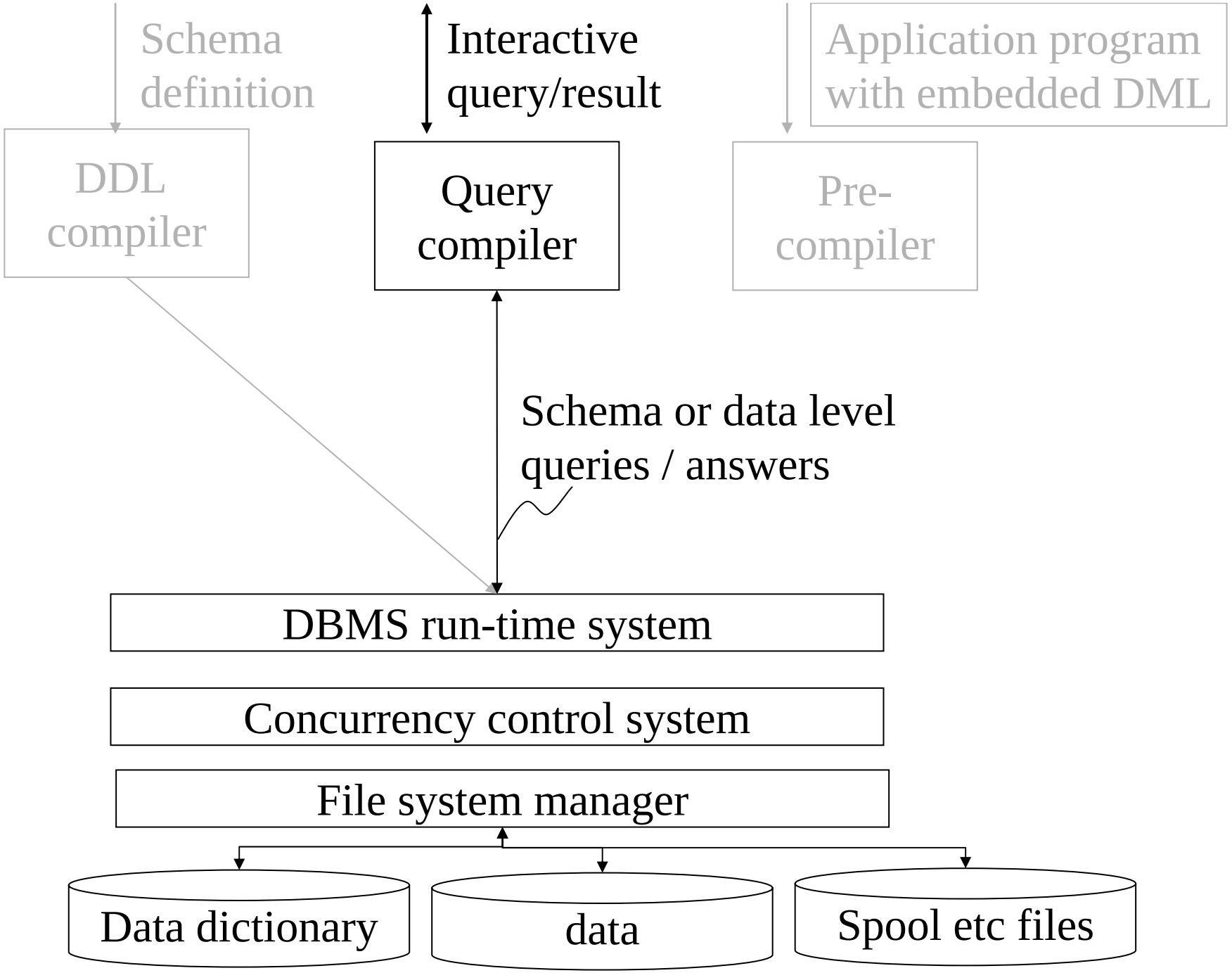
- Isolation: a transaction does not see the intermediary state of database caused by another transaction. If two transactions want to use the same data item, they are *queued up* (or they are scheduled in a way that the effect is *as if* they had been queued up) or *serialized*
- Durability: if a transaction has been performed and committed then the result remains there 'indefinitely'

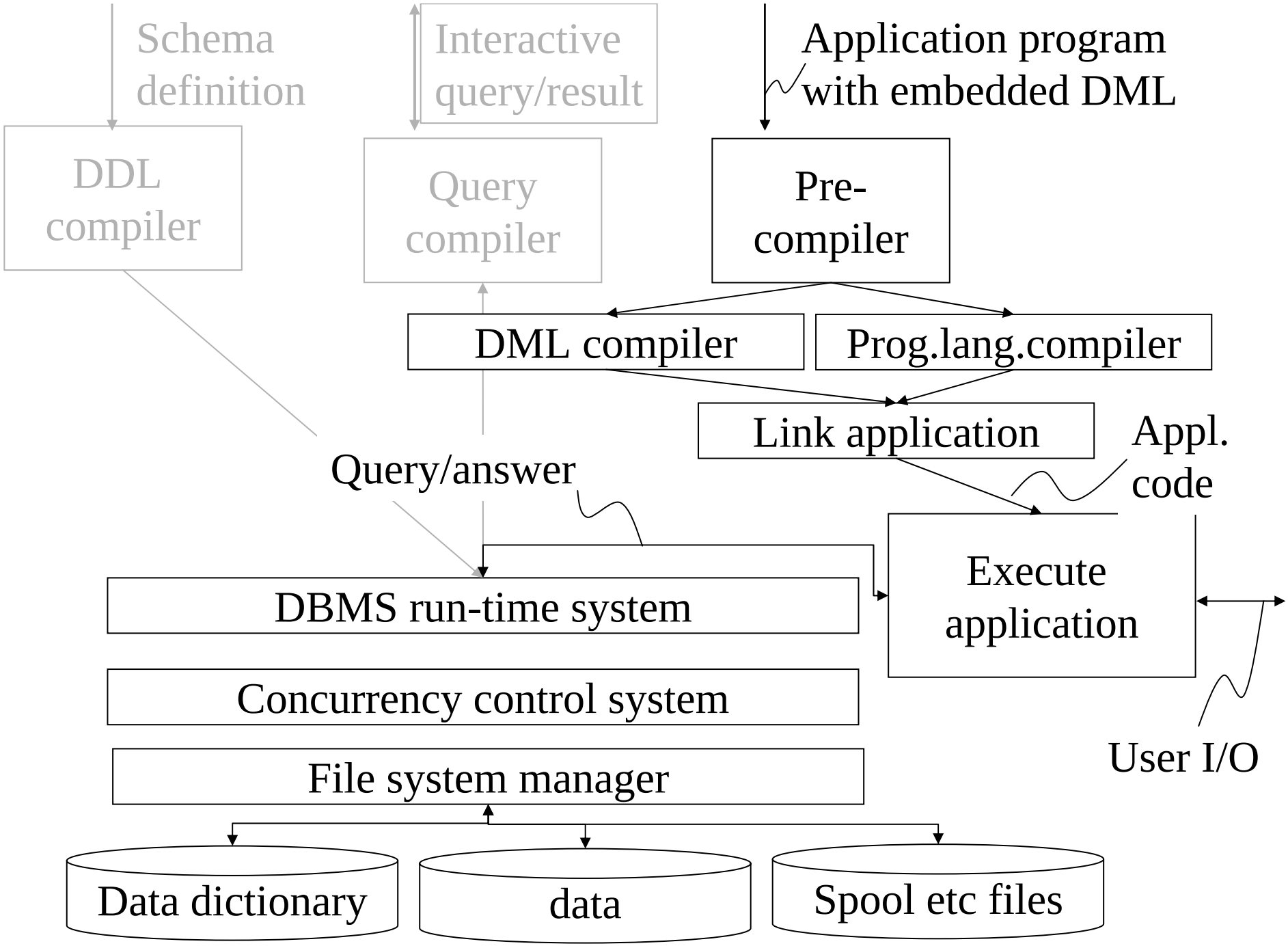
Concepts

- DDL: data definition language (used to describe the *database schema* (i.e. the structure of the data in the database)
- DML: data manipulation language used to insert, retrieve and update *data* in the database
- Database Catalogue: the part of the database which stores the database schema description

Structure of a typical Database Management System







Database People

- Database administrator (a well paid job!)
  - [consults users about their needs,]
  - [select what information will be stored]
  - [chooses logical / physical data structures for best performance]
  - [chooses integrity constraints to maintain automatically]
  - monitors performance
  - changes data structures as needed
  - arranges backup and recovery
  - determines access rights

- Database Designer (in large organisations a separate function):
  - consults users about their needs,
  - select what information will be stored
  - chooses logical / physical data structures for best performance
  - chooses integrity constraints to maintain automatically

- Programmers (‘Software engineers’):
  - implement/ test / debug application programs
  - configure software systems
- End users:
  - casual users (occasionally use the DB)
  - naïve (parametric) users (no database knowledge, use DB through application programs only)
  - sophisticated users (directly use database)
  - stand-alone users: personal databases

The End