Data Encapsulation in Software Components

Kung-Kiu Lau and Faris M. Taweel

School of Computer Science

The University of Manchester, UK

{kung-kiu, faris.taweel}@cs.man.ac.uk

Overview

- Data encapsulation in OOP facilitates reuse (multiple instances)
- Is data encapsulation possible in CBSE?
- Need to combine data encapsulation with composition

Current Software Component Models

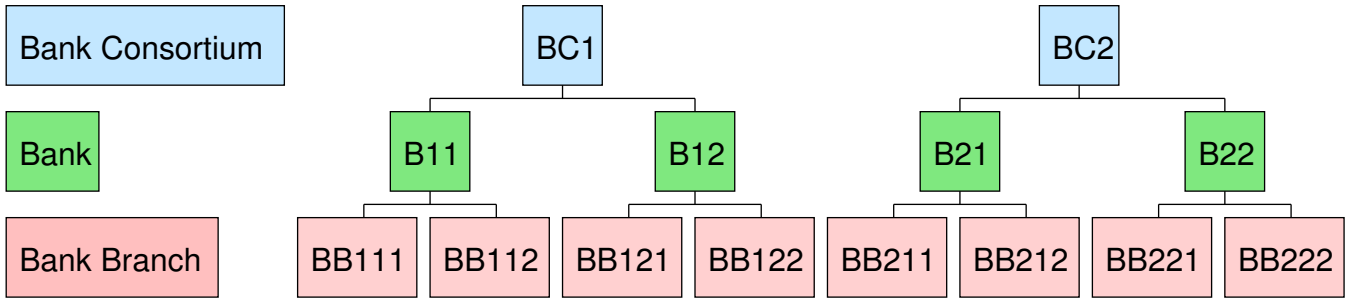
Current component models fall into 2 categories:

- components are objects, e.g. EJB
- components are architectural units, e.g. ADLs

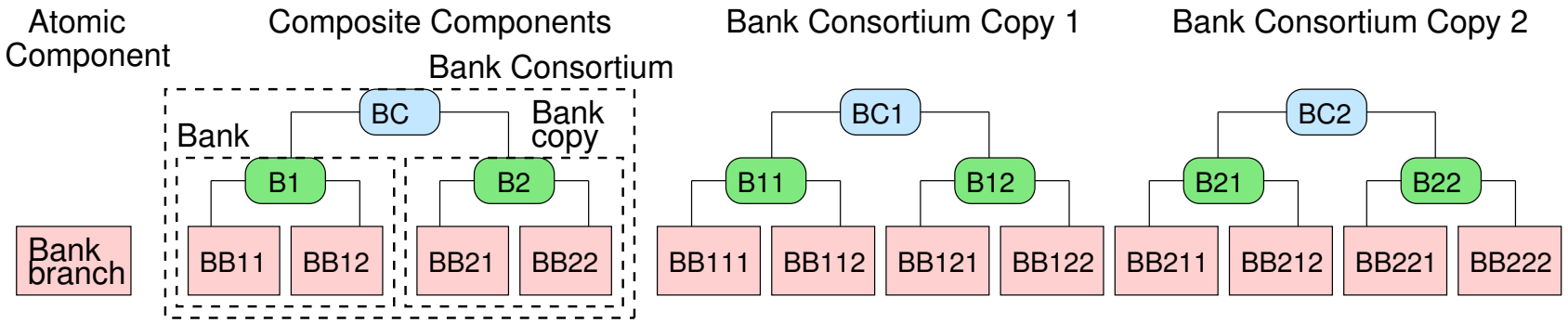
Components	Data encapsulation	Composition
Objects	Yes	No
Architectural units	?	Yes

Composition with Data Encapsulation

Bank system description:



Component-based bank system implementation:



Our Component Model

Two kinds of basic entities:

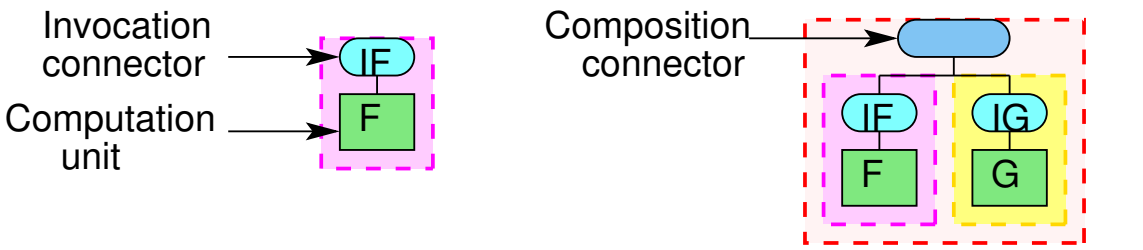
- computation unit
  - provides a set of methods (or services)
  - methods do not call methods in other computation units

  (encapsulates computation)
- connector
  - invocation connector
    - connected to a computation unit, provides access to its methods
  - composition connector
    - defines and coordinates the control for a set of components
      e.g. sequencer, selector, pipe

  (encapsulates control)

Two kinds of components:

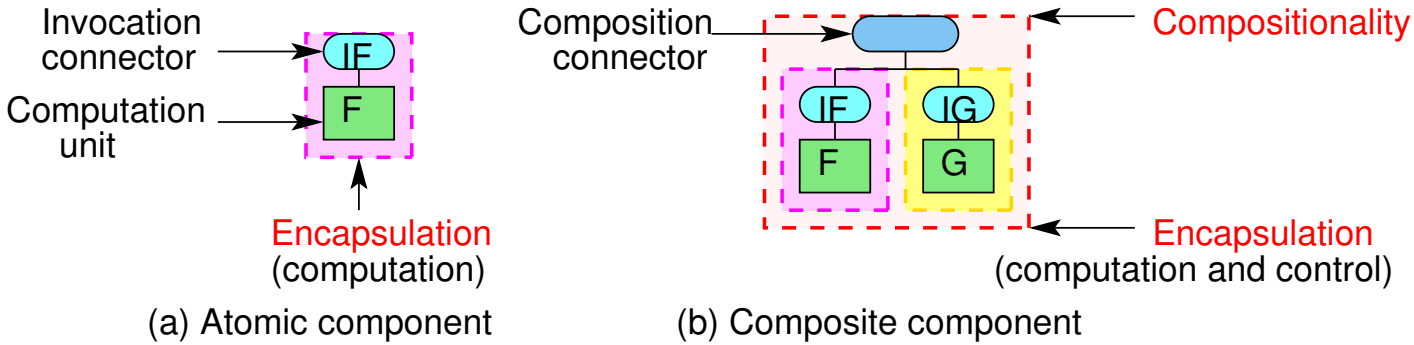
- atomic component
  - invocation connector + computation unit
- composite component
  - composition connector + components (atomic or composite)



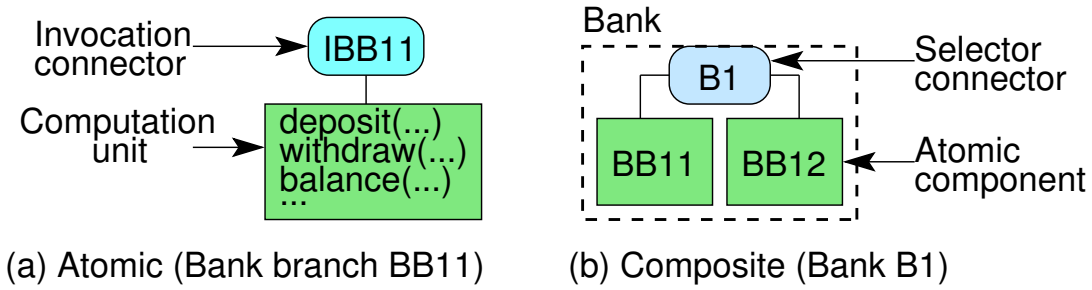
(a) Atomic component (b) Composite component

Our Component Model: Encapsulation of Control and Computation

Atomic and composite components:

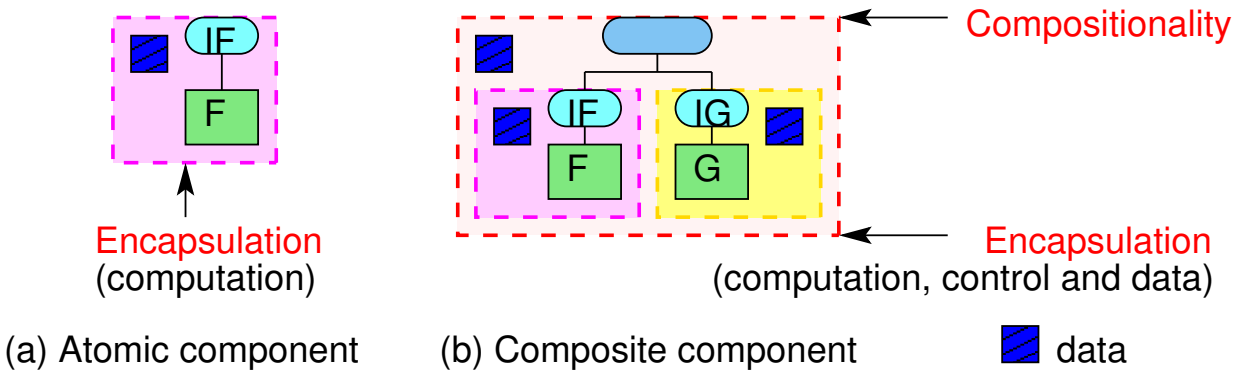


Bank example:

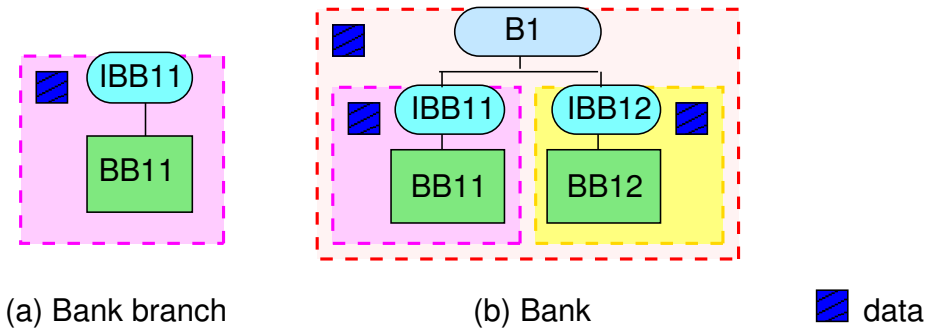


Our Component Model: Encapsulation of Data

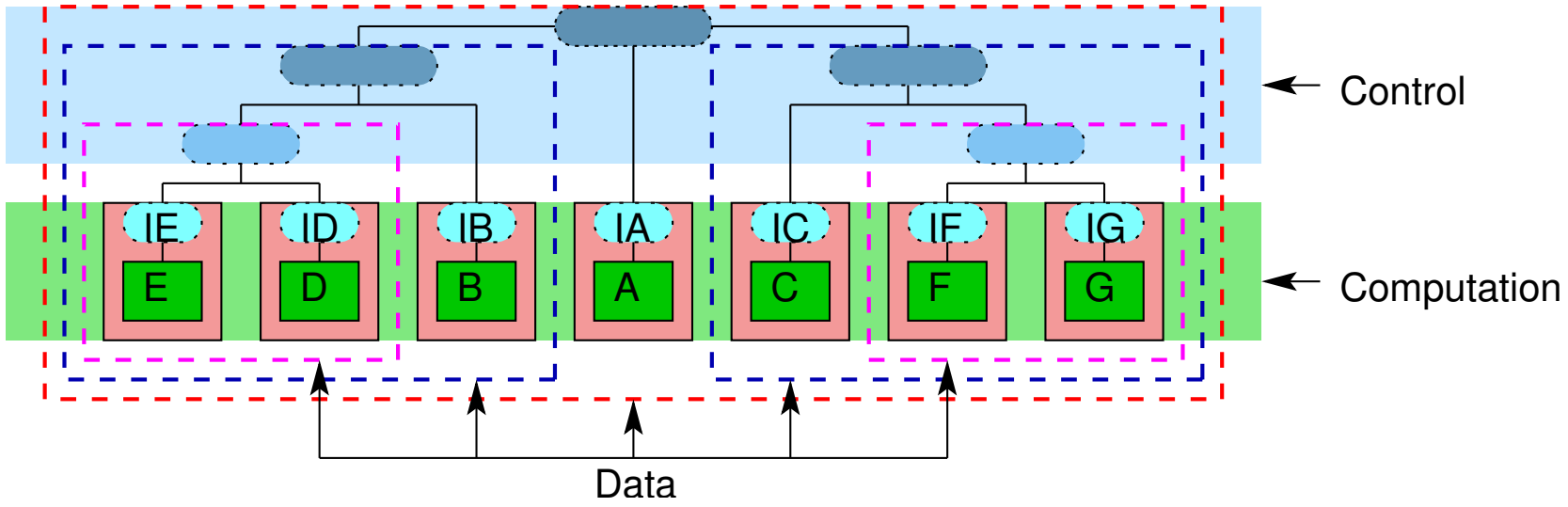
Data encapsulation in every (composite) component:



Bank example:



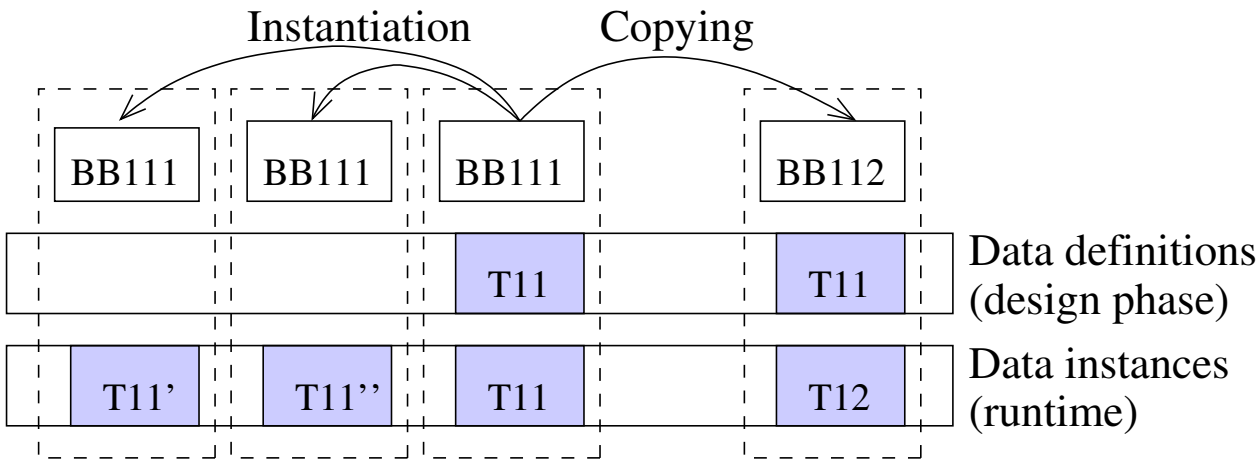
Our Component Model: Encapsulation



('*Encapsulation: Enclosure in a capsule*', OED)

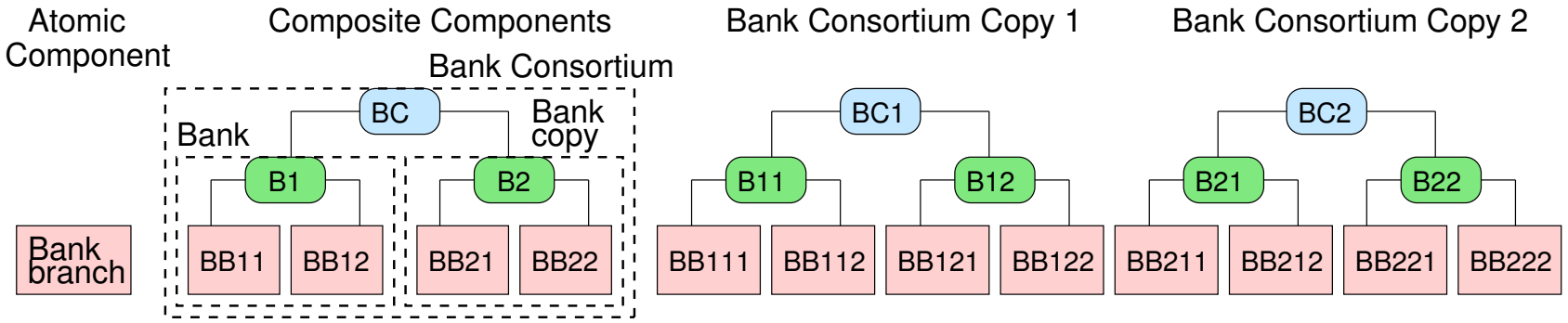
Implementation of Data Encapsulation

- Centred on the constructor of a component
  - copies of components at design time
    (with place-holders for data)
  - instances at run-time
    (with initialised private data)



- Implementation in PL/SQL
  - computation units are Oracle packages
  - connectors are Oracle object types
  - data operations use data connectors

Bank example:



Implemented with

- 1 atomic component (bank branch)
- 1 composition connector (selector)

Conclusion

- Component model with data encapsulation
- Combines encapsulation with composition
- Facilitates reuse:
  - multiple copies at design time
    (unlike OO classes, which cannot have copies)
  - multiple instances at run-time
    (like OO objects)
- Encapsulation at the level of component models, not at the level of programming languages