

Towards Composing Software Components in both Design and Deployment Phases

Kung-Kiu Lau, Ling Ling, Perla Velasco Elizondo

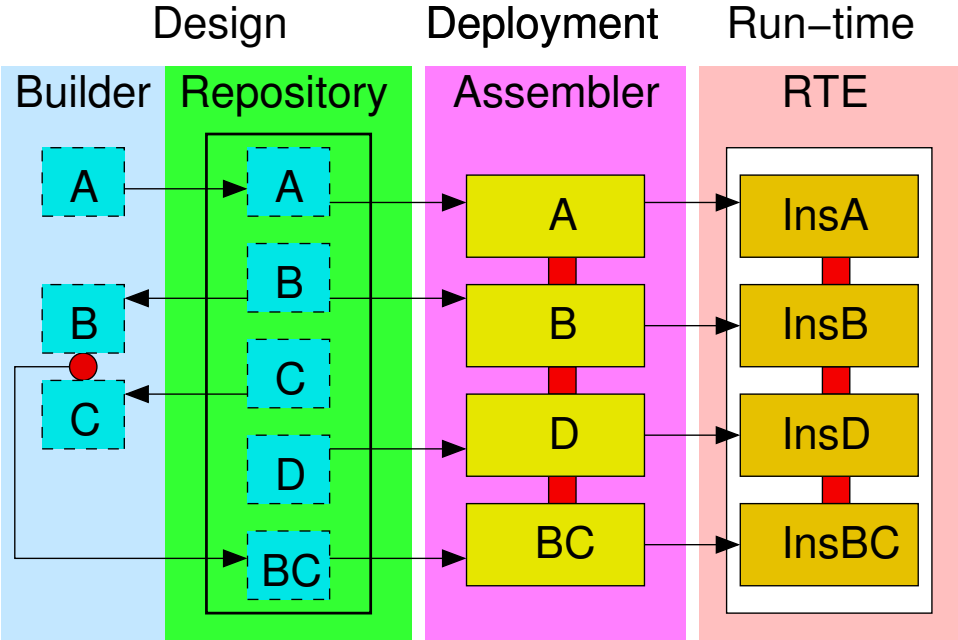
**School of Computer Science
The University of Manchester, UK**

{kung-kiu, lling, pvelasco}@cs.man.ac.uk

Overview

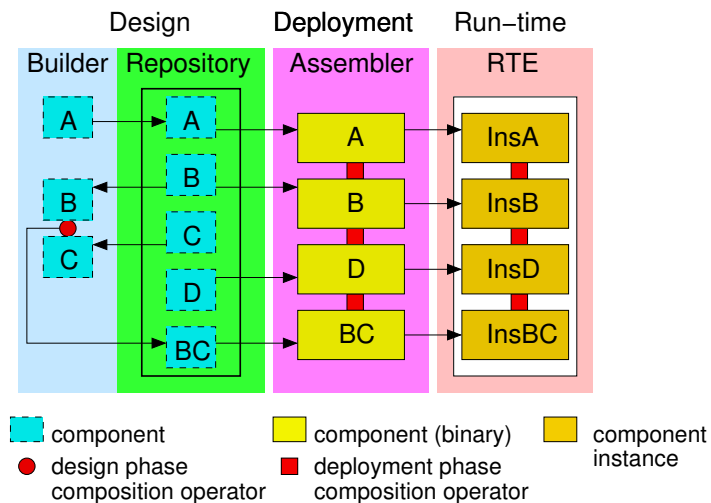
- **Composition** is central to CBSE
- **Desiderata** for CBSE
- **Two-phase** composition in our component model
- Preliminary implementation

Motivation: An Idealised Component Life Cycle



- component
- component (binary)
- component instance
- design phase composition operator
- deployment phase composition operator

Motivation: An Idealised Component Life Cycle (Continued)



This life cycle reflects CBSE desiderata:

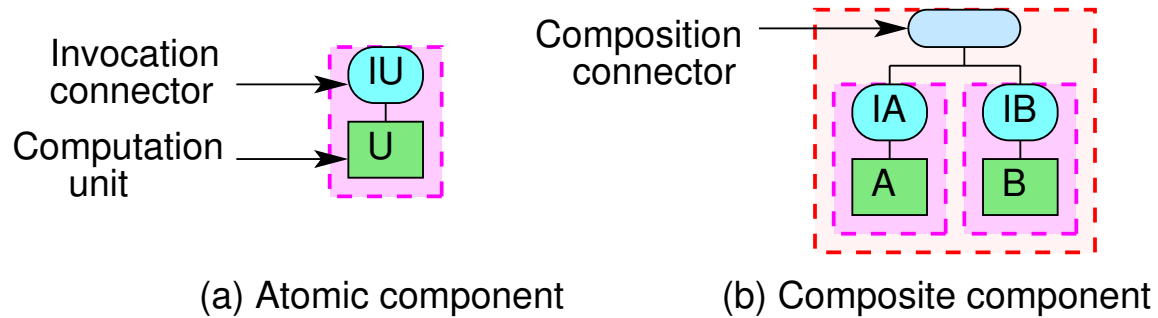
Desideratum	Design Phase	Deployment Phase	Run-time Phase
components produced independently	builder	--	--
components pre-exist	repository	repository available	--
components deployed independently	--	assembler	--
components can be copied and instantiated	copies possible	copies possible	instances possible
composites can be made	composition possible	composition possible	--
composites can be stored	repository	--	--

Two-Phase Composition

	Design Phase	Deployment Phase	Run-time Phase
Role	Component designer (producer/vendor)	System developer	System user
Environment	Application independent	System specific	System execution environment
Component type	Template + deployment contracts	Deployed subsystem	Executable subsystem
Data in components	Place-holders	Place-holders + configuration data	All data initialised
Component format	Source or binary	Binary	Binary instance
Composition operators	Pre-defined	Pre-defined	--
Interface generation	Yes	Yes	--

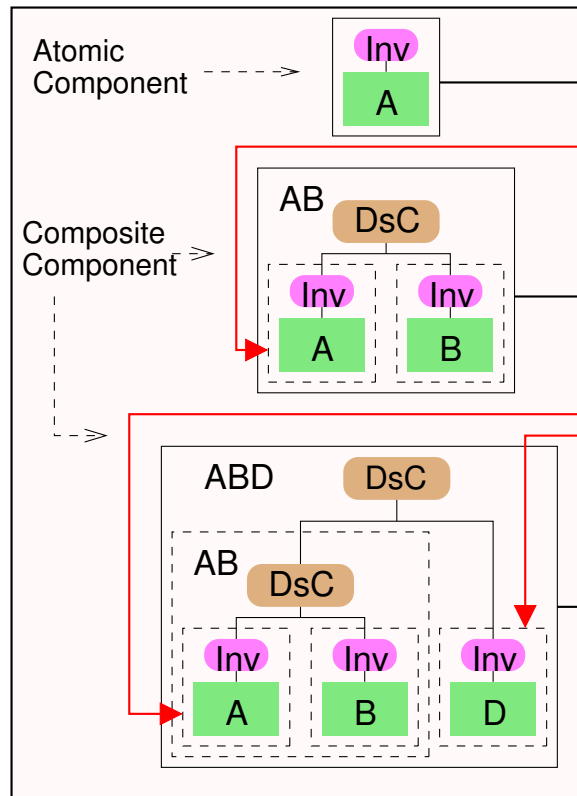
Our Approach

Based on our **component model**:

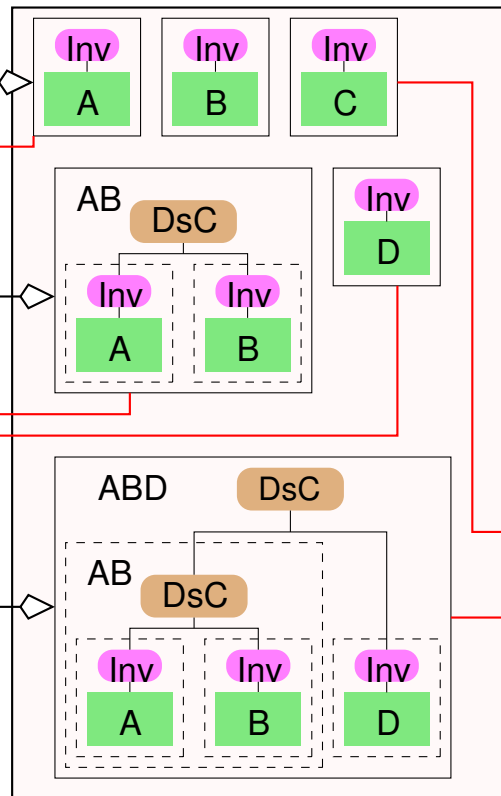


Our Approach (Continued)

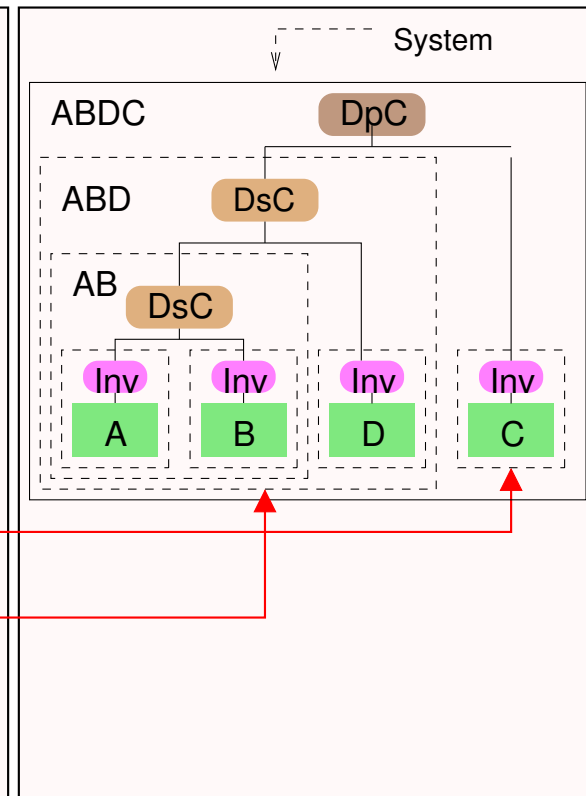
(a) Design Phase Composition



Repository



(b) Deployment Phase Composition



Inv Invocation Connector
 Computation Unit

DsC Design Phase Connector
 DpC Deployment Phase Connector

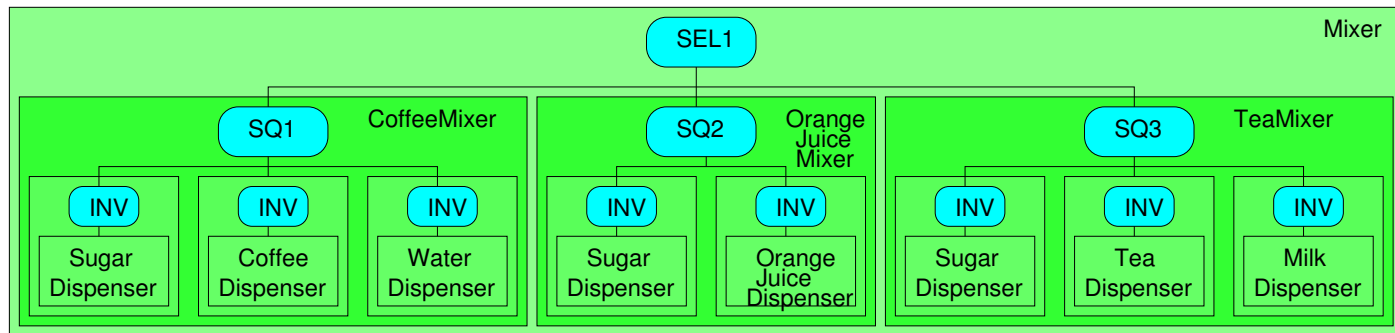
—◇ Store into Repository
 —▶ Retrieve from Repository

Preliminary Implementation

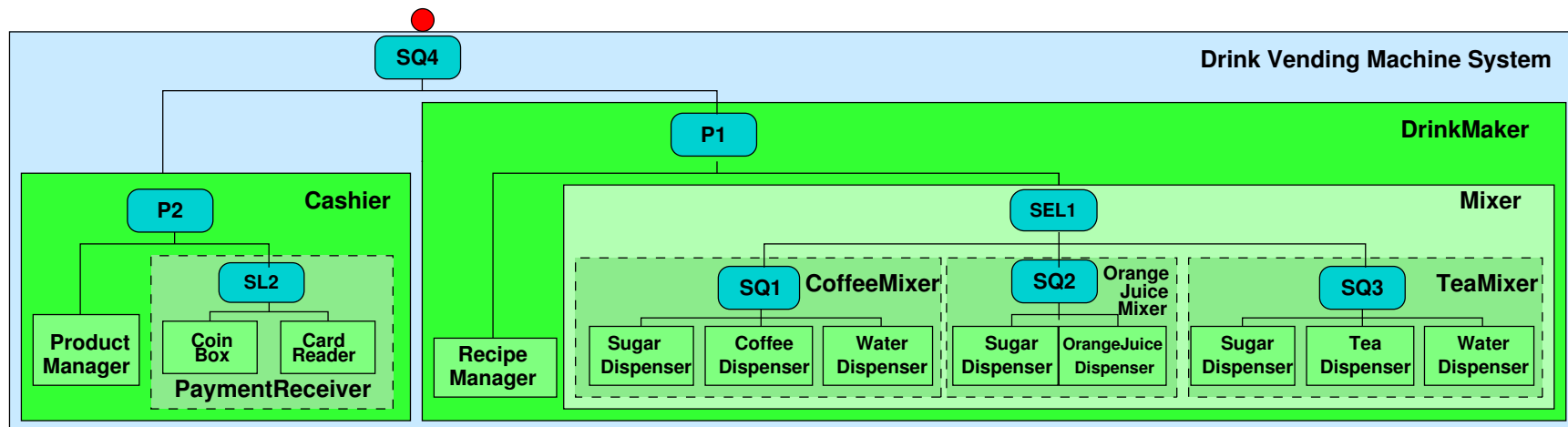
- We have implemented components and composition connectors in both design and deployment phases
- Java as programming language
- JVM as execution environment
- Simple repository
- Full-blown tools for builder and assembler as future work

Example

Design phase composition: a composite component Mixer



Deployment phase composition: a complete system Drink Vending Machine



Discussion

- Two-phase composition **maximises**:
 - design flexibility
 - component reuse
- Two-phase composition is **not** possible in **current component models**:

Component Model	Design Phase Composition	Deployment Phase Composition
ADLs, UML2.0, PECOS, Pin, Fractal, EJB, COM, CCM, Koala, SOFA, KobrA	Yes	No
JavaBeans	No	Yes
Our Approach	Yes	Yes

Future Work

- Improve the implementation of the
 - builder
 - assemblertools
- Reference semantics
- Deployment contracts
- Deployment tools