Improving Software Reliability by Enforcing Composition Time Constraints

Lieven Desmet – Frank Piessens – Wouter Joosen – Pierre Verbaeten

DistriNet Research Group, Katholieke Universiteit Leuven, Belgium

Lieven.Desmet@cs.kuleuven.ac.be

Overview

- Composition time constraints
- Dataflow dependencies
- Support for enforcing composition time constraints
- Summary

Composition time constraints

- Modern software systems:
  - quite complex
  - composed of reusable components
  - highly reconfigurable

  but also:
  - manageable
  - dependable
    - expected behavior
    - availability
    - robustness
    - …

Composition time constraints

- Design constraints & invariants:
  - architectural constraints
  - Inheritance
  - encapsulation and data typing
  - Component contracts
  - …

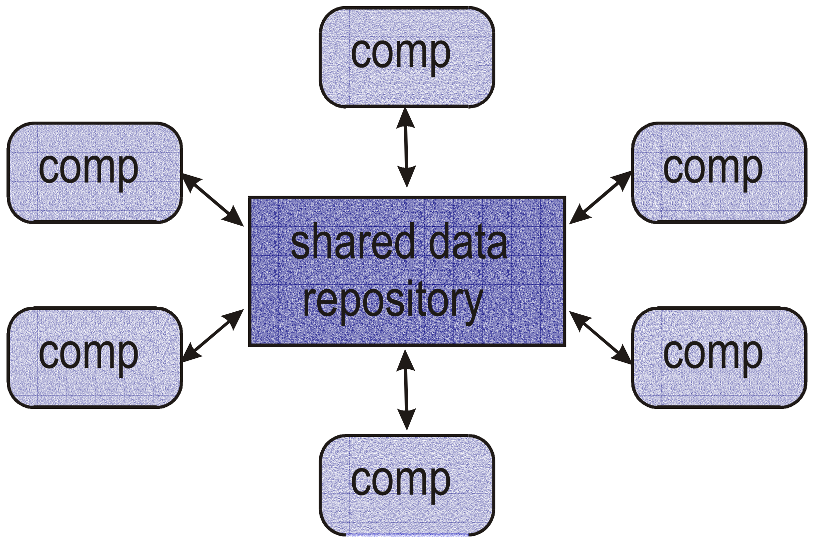
- Although, constraints and dependencies are not only introduced at design time…

Composition time constraints

- ➢ *Composition time constraints:*
  - ➢ Implicit constraints and dependencies introduced by composing software components into an application
  - ➢ Examples:
    - implicit invocation in event based communication
    - indirect data sharing through data-centered repositories
    - …

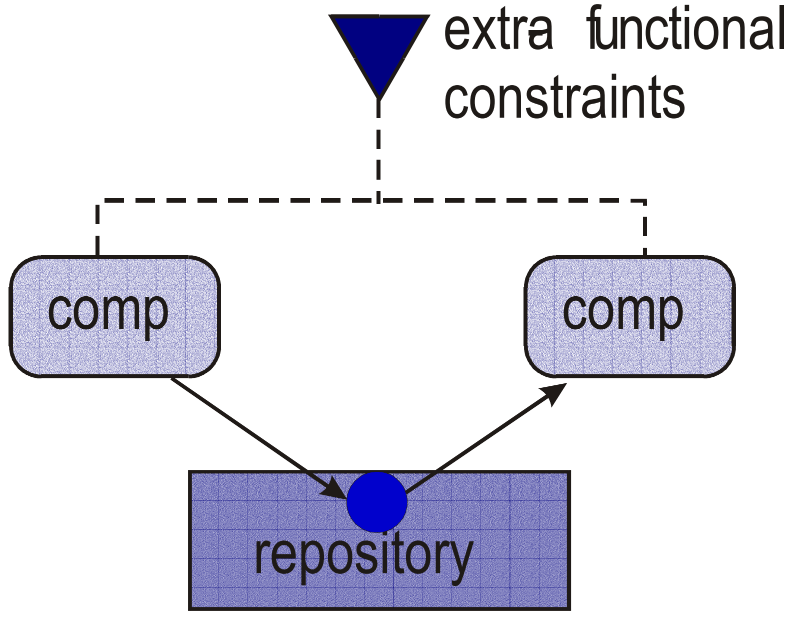
Dataflow dependencies

- Data-centered systems:
  - Central data repository
  - Components can read and write data to the repository
  - Components share data through the shared data repository



Dataflow dependencies

- Composing a data-centered application:
  - Introduces dataflow dependencies between components
  - May require extra-functional constraints on the dataflows



Dataflow dependencies

- Dataflow dependencies are modeled implicitly within a software composition
- Without explicit modeling:
  - Dataflow depencies can break
  - Especially in run-time reconfigurable systems
- Explicit support is needed!

Dataflow dependencies

- Two application domains:
  - Component based protocol stack development with DiPS
  - Dynamic webapplications with Java Servlets
- Experiences:
  - A data providing component is missing in the composition or is swapped out at run-time
  - Synchronization problems on shared data
  - A newly added component breaks existing flows
  - ...

Support for enforcing composition time constraints

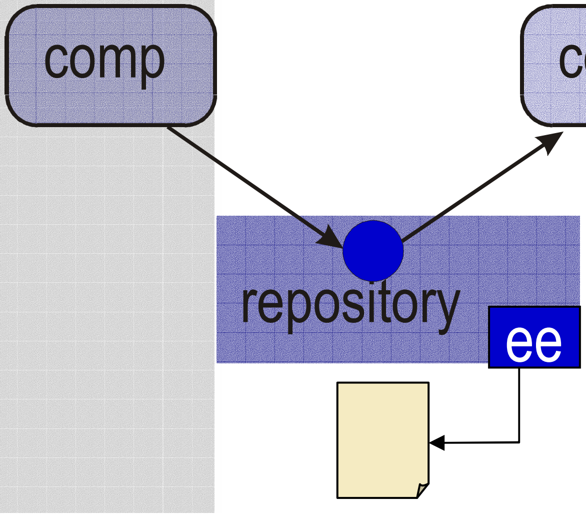
- Support for enforcing composition time constraints:
  - Extended specification
  - Declarative dataflow policy
  - Policy enforcement

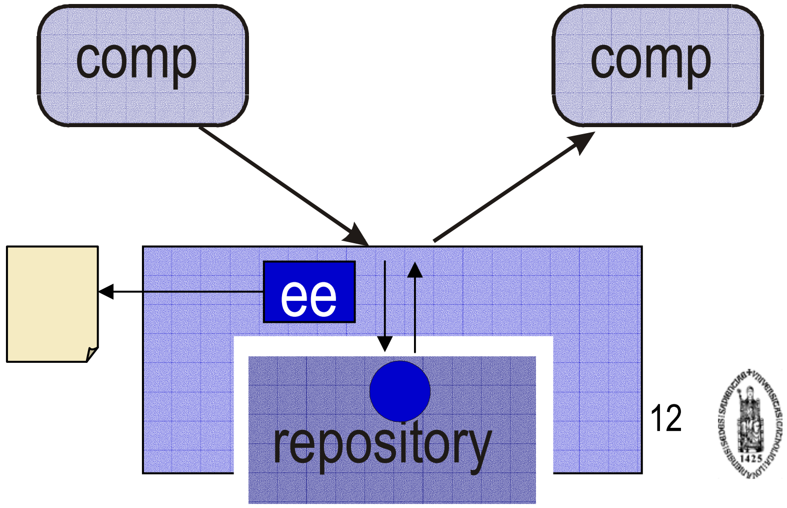
Support for enforcing composition time constraints

- Extended specification:
  - Every component describes interactions with the shared repository
  - Specification can be added manually or generated by tool support
- Declarative dataflow policy:
  - The composer explicitly expresses the dataflows and the extra-functional constraints
  - This can be an extension to the current ADL

Support for enforcing composition time constraints

- Policy enforcement:
  - Enforcement of dataflow policy at run-time
  - Controlling access to the shared repository:
    - Extending shared repository with an enforcement engine
    - Adding a wrapper with built-in enforcement engine





Summary

- Composing software introduces extra constraints and dependencies
- Expressing and enforcing composition time constraints improves software reliability

- Current state:
  - Working prototype in DiPS (component oriented protocol stack framework)
  - conceptual proof of concept with Java Servlets
  - working on prototype in Tomcat webcontainer

Questions

? ? ?

? ?

Lieven Desmet – Frank Piessens – Wouter Joosen – Pierre Verbaeten

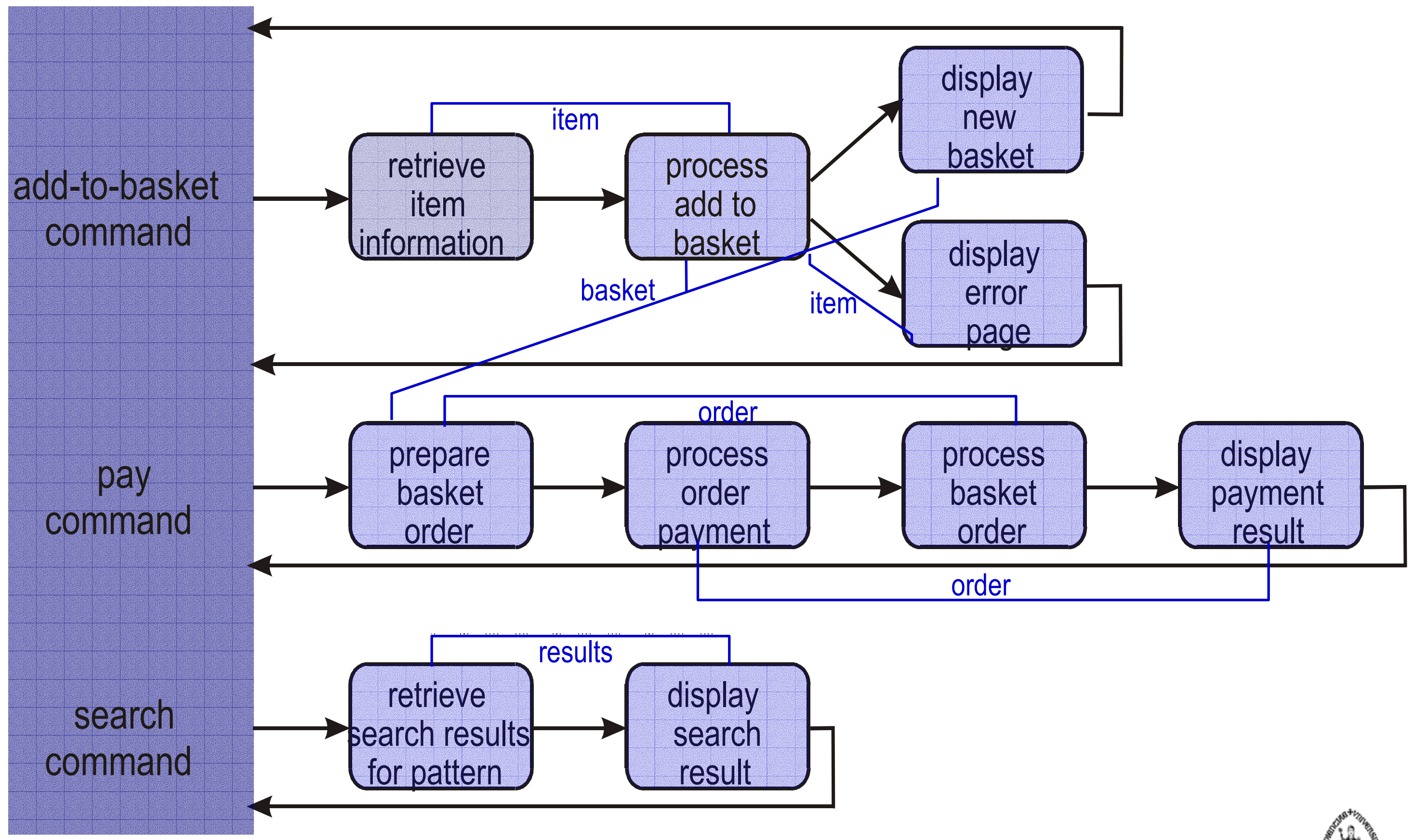
DistriNet Research Group, Katholieke Universiteit Leuven, Belgium

Lieven.Desmet@cs.kuleuven.ac.be

Case study

- A small e-commerce site:
  - Dynamic web application
  - Composed of Java Servlets (part of J2EE)
  - Servlets communicate through shared data repository
  - Set of 3 functionalities:
    - Adding a product to the shopping basket
    - Payment of the shopping basket order
    - A search engine for the website

Case study


add-to-basket command
retrieve item information
item
process add to basket
display new basket
display error page
basket
item
pay command
prepare basket order
order
process order payment
process basket order
display payment result
order
search command
retrieve search results for pattern
results
display search result

Support for enforcing composition time constraints

```
<servlet>
  <servlet-name>itemRetriever</servlet-name>
  <servlet-class>be.compRetrieveItem</servlet-class>
  <data-provision>
    <data-name>item</data-name>
    <data-class>be.comp.Item</data-class>
  </data-provision>
</servlet>

<dependency>
  <provider>
    <Servlet-name>ecommerceInitializer</servlet-name>
    <data-name>basket</data-name>
  </provider>
  <consumer>
    <servlet-name>addToBasket</servlet-name>
    <data-name>basket</data-name>
  </consumer>
</dependency>
```