Dedal-CDL: Modeling First-class Architectural Changes in Dedal

Huaxi (Yulin) Zhang, Christelle Urtado, Sylvain Vauttier, Lei Zhang, Marianne Huchard and Bernard Coulette

❶ Dept. Math / Info, Université Toulouse 2, Toulouse, France

❷ LGI2P / Ecole des Mines d'Alès, Nîmes, France

❸ LIRMM, UMR 5506, CNRS and Univ. Montpellier 2, Montpellier, France

❹ Research Center of Automation, Northeastern University, Shenyang, China

Outline

- **Context**

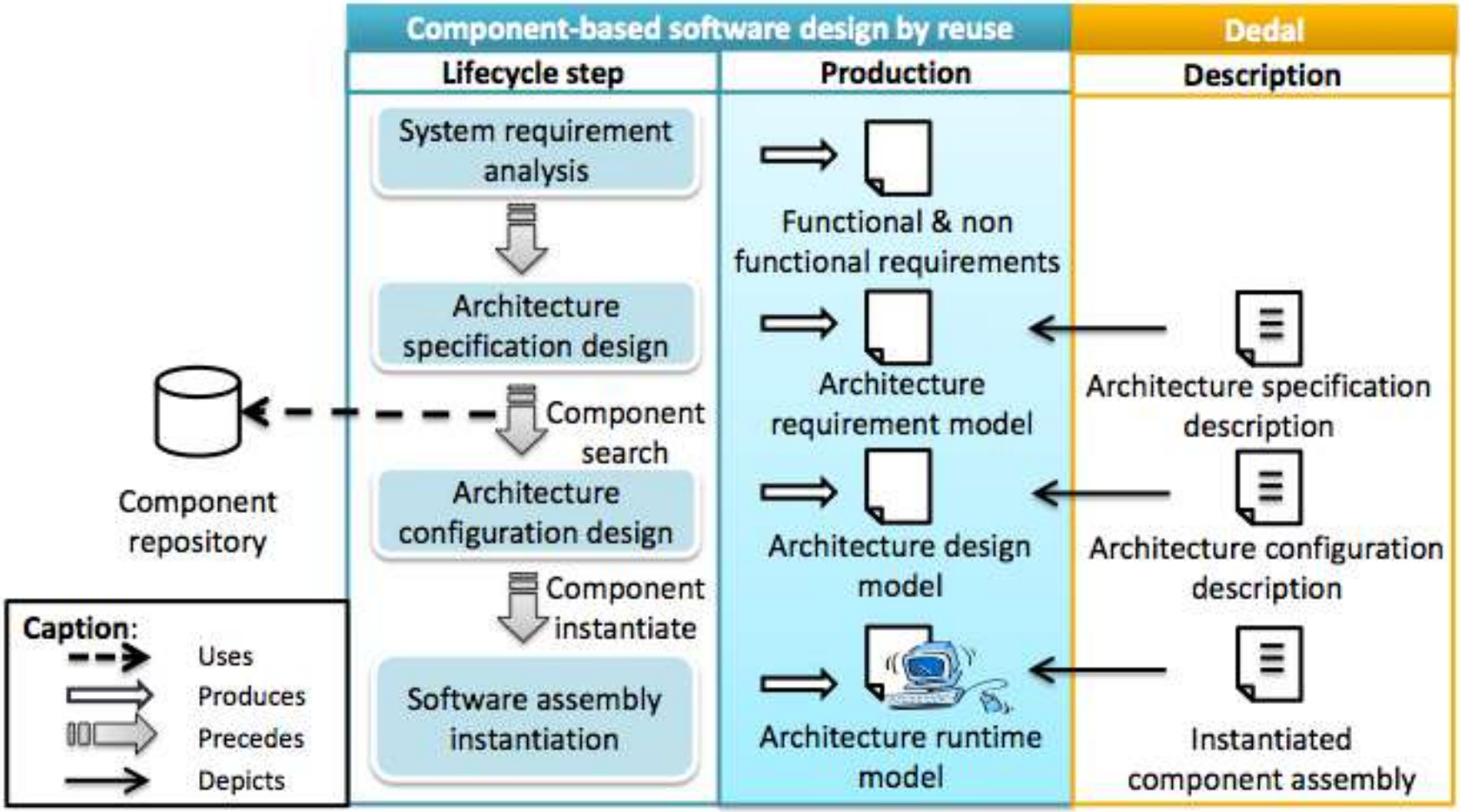
  Architecture-centric component-based development needs a three-level ADL
  Huaxi (Yulin) Zhang, Christelle Urtado, and Sylvain Vauttier
  @ ECSA 2010

  - **Component-based software design by reuse**
  - **The three level Dedal ADL**

- **Motivation : architecture evolution management**
- **Modeling changes**
- **Versioning architectures**
- **Comparison to state-of-the-art ADLs**
- **Conclusion and perspectives**

Component-based software design by reuse

The Dedal three levels ADL

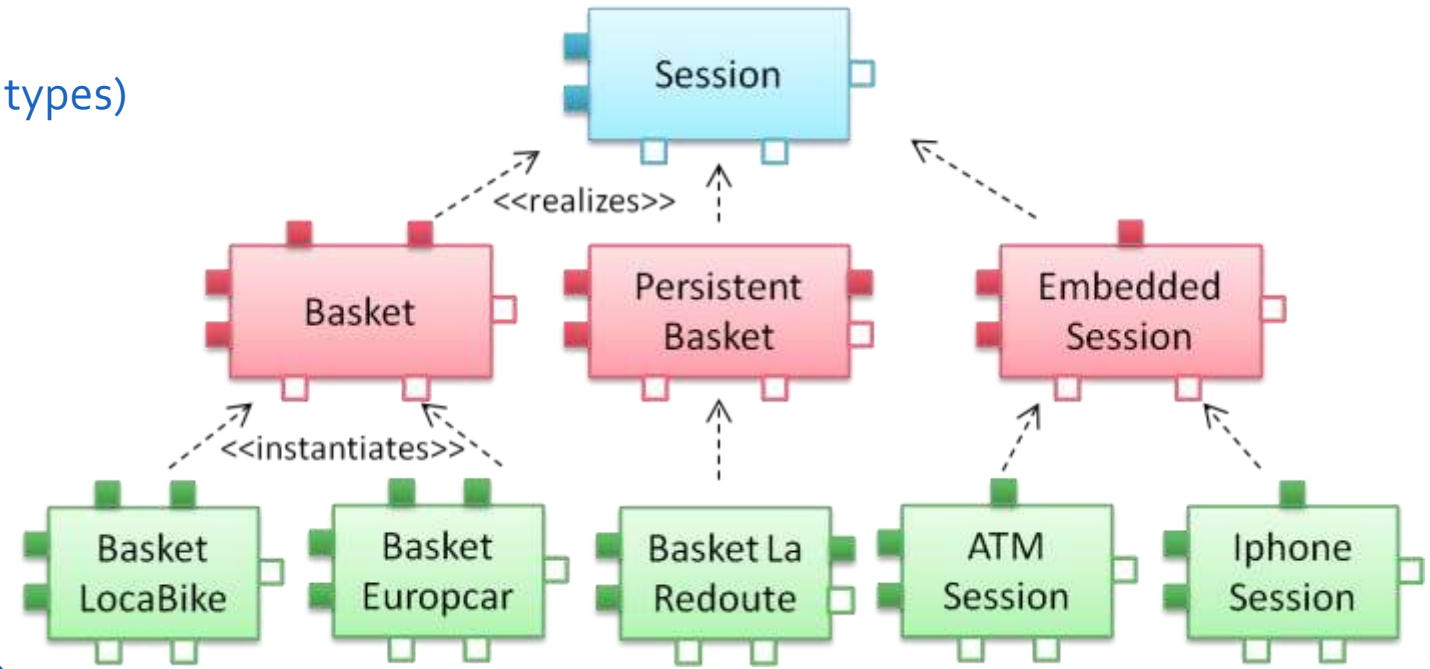


The three levels for component description in Dedal

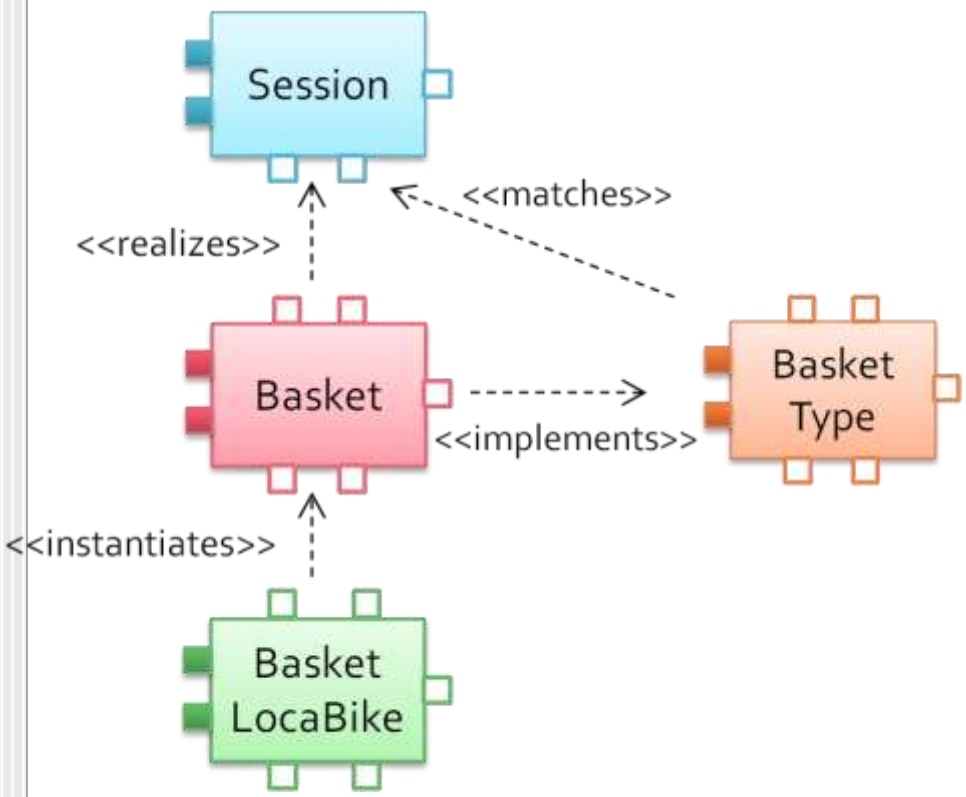
Specification
(abstract & partial types)

Configuration
(concrete classes)

Assembly
(specific instances)



Insights on Dedal's syntax

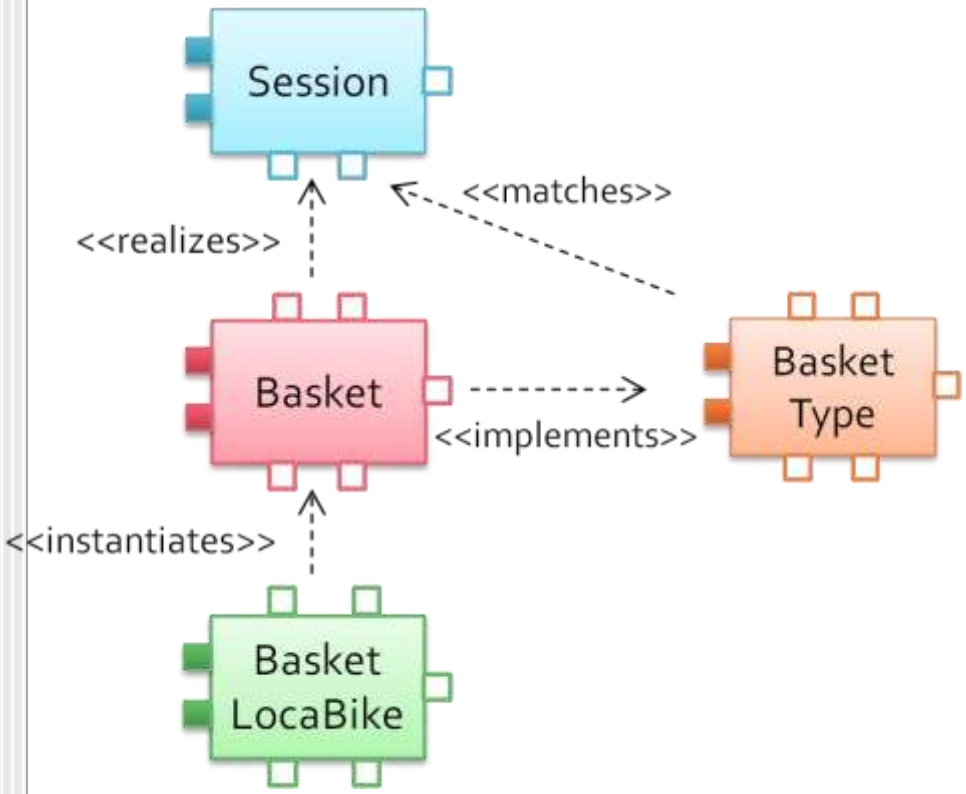


component_role Session
required_interfaces BikeOprs; CourseOprs; AccountOprs
provided_interfaces Account; Bike
component_behavior . . .

component_type BasketType
required_interfaces BikeOprs ; CourseOprs; AccountOprs; CampusOprs; AccessoryOprs
provided_interfaces Account; Bike
component_behavior . . .

component_class Basket
implements BasketType
content fr.ema.locaBike.Basket
attributes string company; string currency
versionID 1.0

Insights on Dedal's syntax

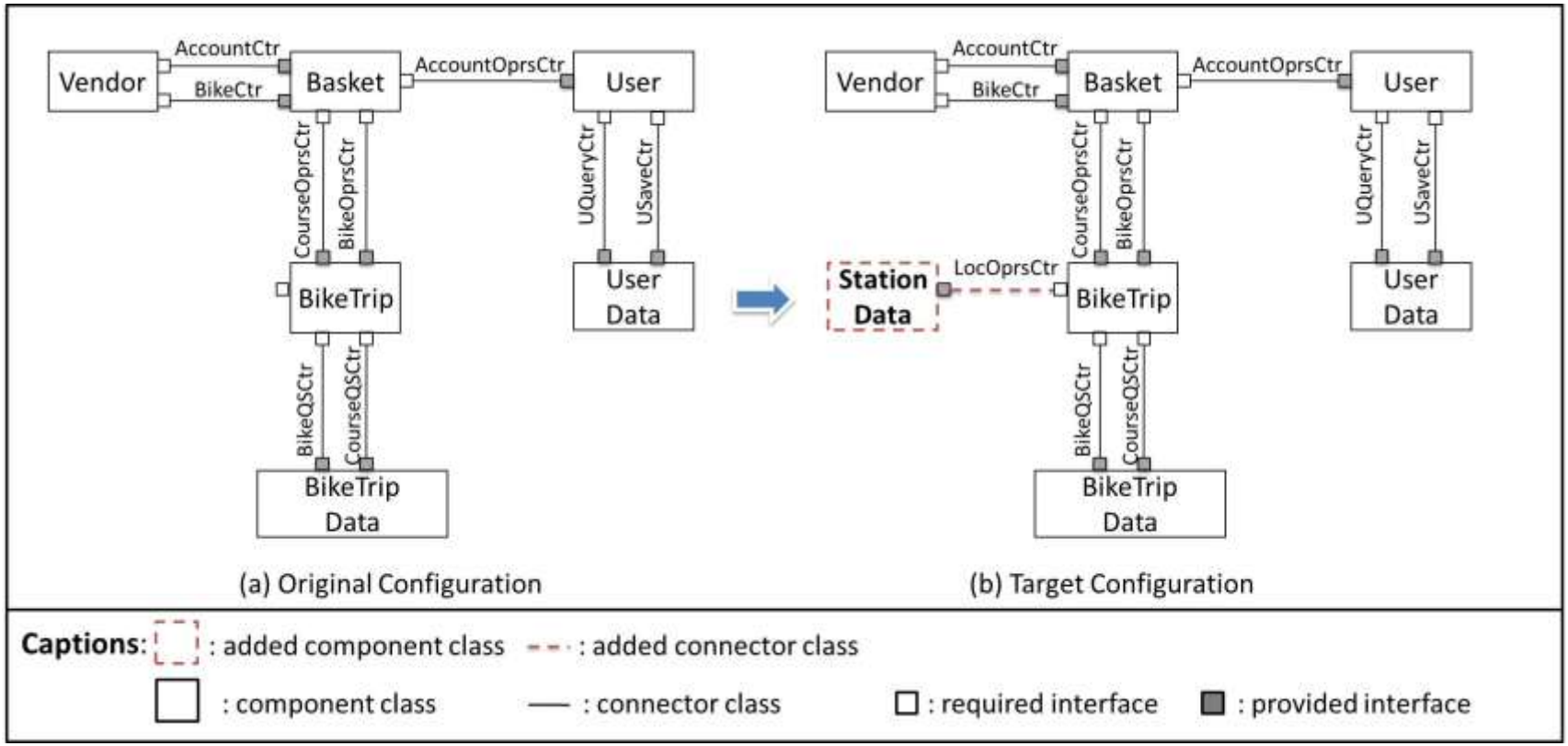


component_role Session
required_interfaces BikeOprs; CourseOprs; AccountOprs
provided_interfaces Account; Bike
component_behavior . . .

component_class Basket
implements BasketType
content fr.ema.locaBike.Basket
attributes string company; string currency
versionID 1.0

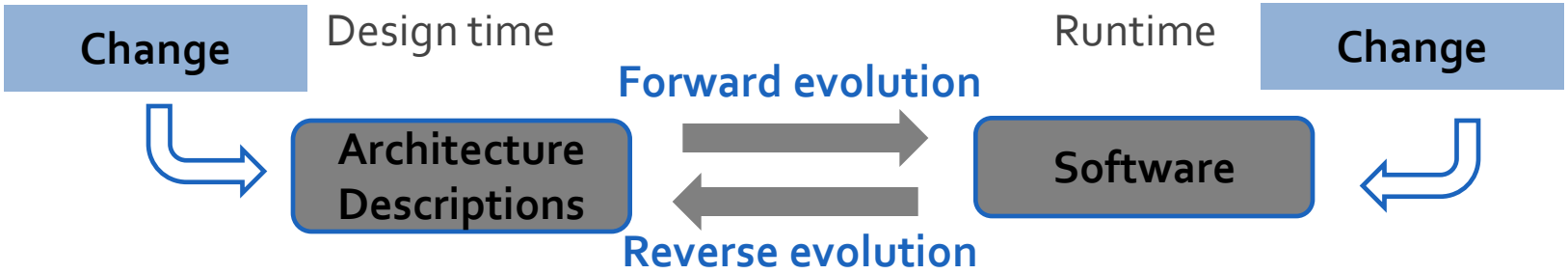
component_instance BasketLocaBike
instance_of Basket **(**1.0**)**
initiation_state company="LocaBike"; concurrency="euro"

Motivation : architecture evolution management 1/2



Motivation : architecture evolution management 2/2

- Changes can be **initiated elsewhere**.



- **Avoid mismatches** : architecture pendency, architecture erosion and architecture drift.
- Change impact must be **propagated appropriately** to maintain all architecture descriptions and the software coherent at each time (co-evolution).
- **Versioning** should be proposed to **track architectural changes** and soundly integrated with change management.

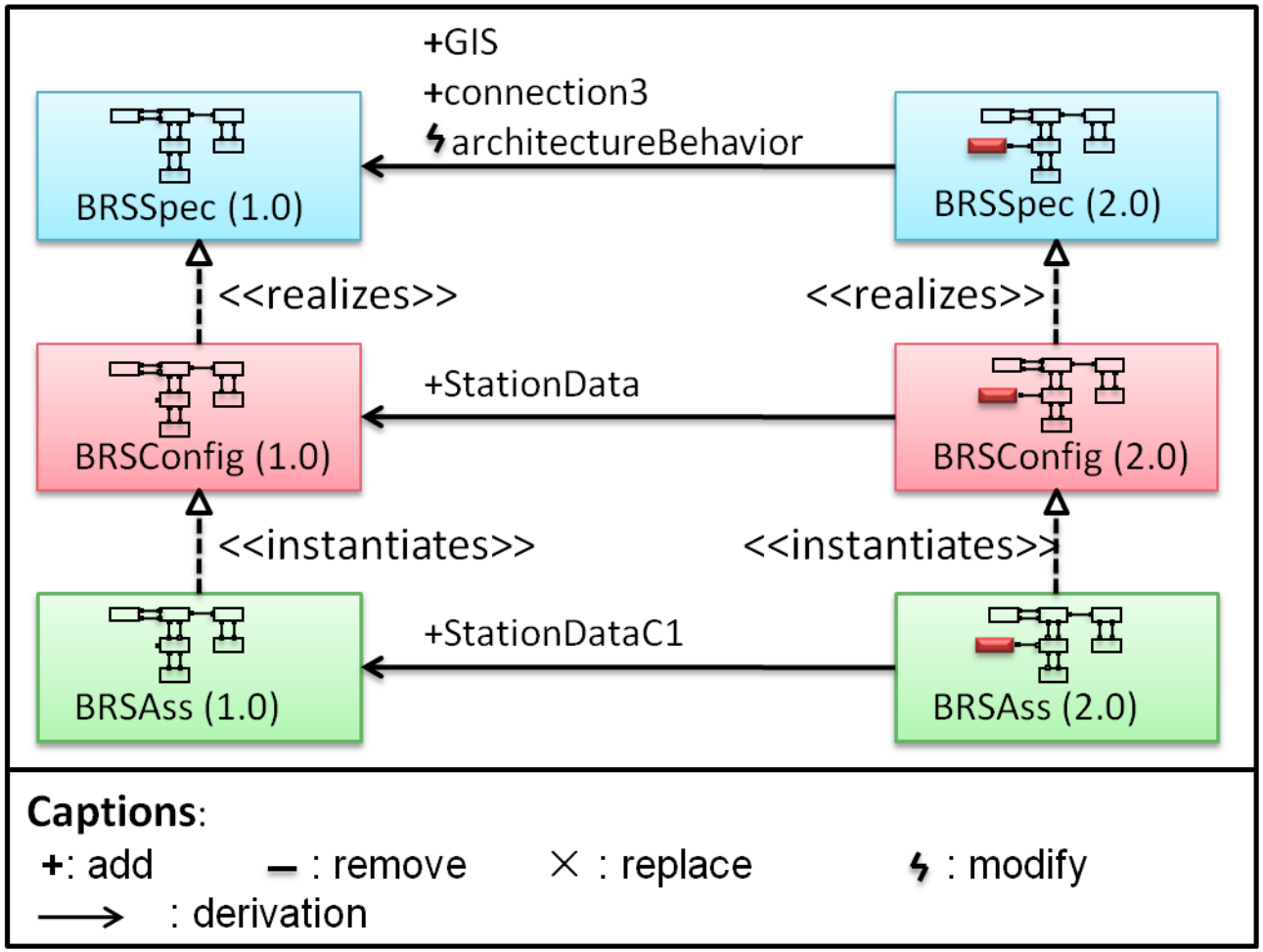
Proposal : model changes as first class elements.

Change characteristics and a Dedal-CDL change description example

Change Characteristics	Values
time of change	static, dynamic
anticipation	anticipated, unanticipated
affected level	specification, configuration, assembly
change operation	addition, removal, substitution, modification
subject of change	elements of architectures in three levels
change purpose	corrective, perfective, adaptive
nature of change	given, generated, propagated

change additionStationData
time dynamic
anticipation unanticipated
level configuration
operation addition
subject component_class **is** StationData
purpose perfective
nature given

Versioning architecture descriptions



Comparison 1/2

Characteristics of change	C2	Darwin	Dynamic Wright	SOFA2.0
Time of change	Dynamic	Dynamic	Dynamic	Dynamic
Anticipation	Unanticipated	Anticipated, Unanticipated	Anticipated	Anticipated
Change type	Structural	Structural	Structural	Structural
Change purpose	—	—	—	—
Level of change	Configuration	Configuration	Configuration	Configuration
Change operation	Addition, removal	Addition, removal	Addition, removal	Addition, removal
Subject of change	Components, connectors, connections	Components, connections	Components, connections	Components, connections, interfaces of composite components
Version model	—	—	—	State-based

...

Comparison 2/2

Characteristics of change	...	xADL2.0	MAE	Dedal
Time of change		Dynamic	Dynamic	Dynamic
Anticipation		Unanticipated	Unanticipated	Anticipated, Unanticipated
Change type		Structural	Semantical	Structural, Semantical
Change purpose		—	Perfective	Perfective, corrective
Level of change		Configuration	Configuration	Specification, configuration, assembly
Change operation		Addition, removal	Substitution	Addition, removal, substitution
Subject of change		Components, connectors, connections	Components	Components, connectors in three levels
Version model		—	Change-based	Change-based

Conclusion: contribution

- **Contribution**
  - **Changes are first class**
    - described by the caracteristics we found in state-of-the-art works
  - **Architecture description / running software co-evolution principle is described**
    - consistency checking, change propagation, evolution test, versioning
  - **Software prototype: (small) ad-hoc proof of concept**
    - over the Fractal component model

Conclusion : perspectives

- **Further investigations**
  - **Produce the change information (automatically / manually)**
    - learn from / reuse previous changes
  - **Use the change information to adequately / rigorously manage change**
    - formalization of relations among architecture descriptions (various semantics can be explored)
    - formalization of coherence maintenance rules (no more ad hoc implementation but transformations)
- **Need for an improved state of the art**
  - **deeper / more recent**
- **Develop an eclipse plugin**
  - **Integration of the architectural change management tool to the system development tool for an improved control**

Dedal-CDL: Modeling First-class Architectural Changes in Dedal

Huaxi (Yulin) Zhang, Christelle Urtado, Sylvain Vauttier, Lei Zhang, Marianne Huchard and Bernard Coulette

We welcome any suggestion.