A Metacomponent Model to Support the Extensibility and Evolvability of Networked Applications

Gledson Elias da Silveira
Silvio Lemos Meira

Federal University of Pernambuco, Brazil
Federal University of Rio Grande do Norte, Brazil

Or “How to give your talk the proper spin using keywords …”
Metacomponents

- Wrap a component of any existing model and provide …

- Identification and versioning
  
  Unique identification along the lines of “vendor.product.component.version”

- Access to embedded component

- Binding to other metacomponents
  
  “Dependencies are directly and automatically derived … whenever X sends messages or refers to Y, the model assumes that X is composed of Y.”

Not clear if they are a runtime or a packaging concept …
Probably because they did not want to bother with CDL …
**Metacomponent Types**

- **Basic**
  
  Components that have no other components embedded in them.
  
  ```
  getID(), getType(), getComponent(), getEnvironment()
  ```

- **Composite**
  
  Implements its facilities by employing services of other components.
  
  ```
  getComposition()
  ```

- **Application**
  
  Such composite component that represents an entire application.
  
  ```
  getName(), getIcon(), getInstaller()
  getUpdate(), setUpdate(), getLoad(), setLoad()
  ```

*Feels pretty naive for any practical use ...*
Metacomponents Identification

- String identifiers in a form

Name "/" Version
  Name = Authority "." Product "." Artifact
  Authority = Sector "." Institution
  Product = Level "." ... "." Level
  Version = Number "." ... "." Number

No semantics on the identification, just uniqueness ...
Architectural Style

- Layered architecture
- Bidirectional links allowed on same layer
- Downwards links allowed between layers

That’s probably because they wanted to look architectural …
Architecture Framework

- Just an elaborate ruse to say there is some infrastructure
- Metacomponents must be registered by vendors
- Replication must be implemented
- Multiple bindings must be supported
- Distributed repository service must be available

- Execution environments somehow do demand loading …

Yeah, wouldn’t we all want that …
Conclusion

“The metacomponent model may cause a serious impact on current approaches, solving some hard problems and overcoming significant limitations. …”

We are the only ones to provide “support for location transparency, referential integrity, reusability, scalability, resilience, extensibility, evolvability, security.”

- CoDelivery prototype (probably Java toy)
- SOS prototype (probably nonexistent)