

## An Experience Report on Using UML 2.0 to Document Software Architectures

SEI Software Architecture Technology User Network  
April 6 -7, 2005 Pittsburgh, PA

Art Culbertson  
*Principal Architect*  
Lockheed Martin Information Technology

### Why UML for Architecture Documentation?

- “Lingua franca” of software engineering
  - Leverages current knowledge
  - ADLs unlikely to achieve widespread acceptance
- More formal alternative to ad hoc approaches
  - Semantic metamodel
  - Accessible to a wide spectrum stakeholders
- OMG standardization infrastructure
  - Defined adoption process to evolve UML
  - Broad industry representation
- Tool vendor and training support

## UML as a Documentation Medium

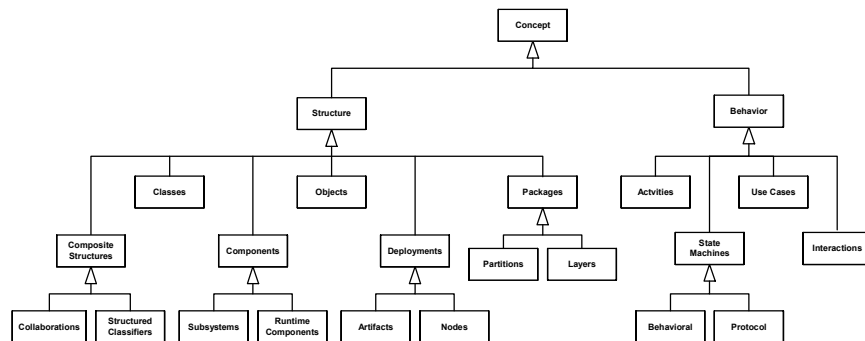
- ❑ The model construct captures system documentation wrt a particular viewpoint
- ❑ Characteristics of UML models
  - Comprised of model elements organized into containment hierarchies
  - Self-contained
  - May be nested
  - May have mapping dependencies between them
- ❑ Model elements can be grouped into several concept areas

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

3

## UML Concept Areas



4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

4

## Choosing UML Concepts for Architecture

- There is more than one way of using UML to model software architecture
- Criteria for choosing UML concepts [Ivers 2004]
  - Semantic match
  - Visual clarity
  - Completeness
  - Tool support

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

5

## A “3+1” View Model of Architecture

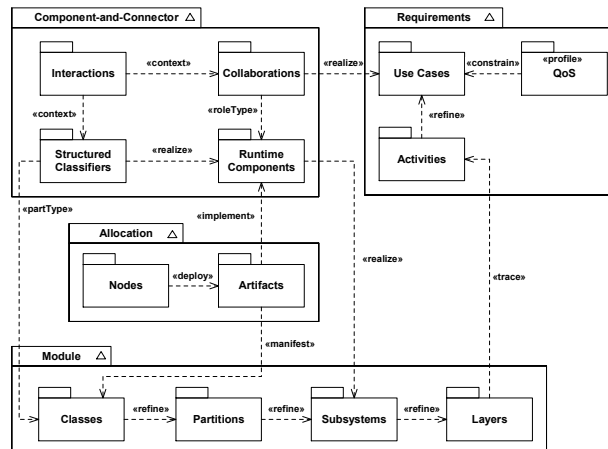
- Based on the three “Views and Beyond” viewtypes [Clements 2002]
  - Module – concerned with how functionality maps to logical implementation units of software
  - Component-and-Connector (C&C) – concerned with the runtime mapping of functionality to components and connectors
  - Allocation – concerned with how software elements are mapped to environmental elements
- Architecturally significant requirements
  - Functional (i.e., use cases)
  - Quality of Service (QoS)

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

6

## Mapping UML Concepts to the “3+1” Models

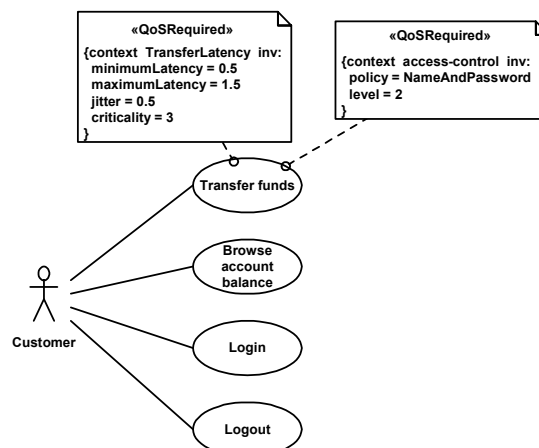


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

7

## Requirements Example – Use Cases

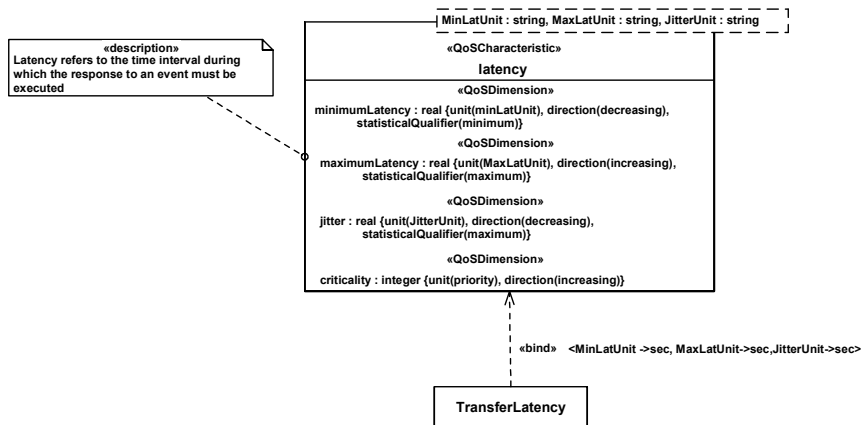


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

8

## Requirements Example – QoS

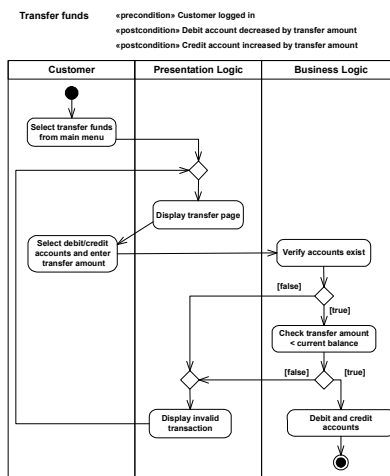


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

9

## Requirements Example – Activities

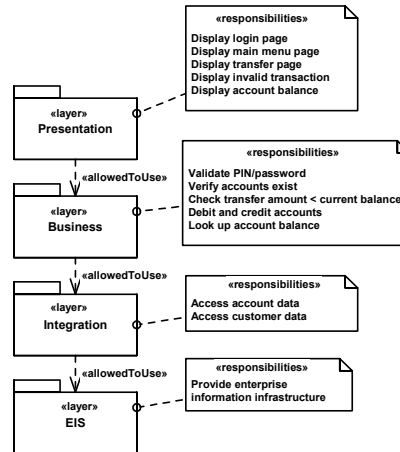


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

10

## Module Example – Layers

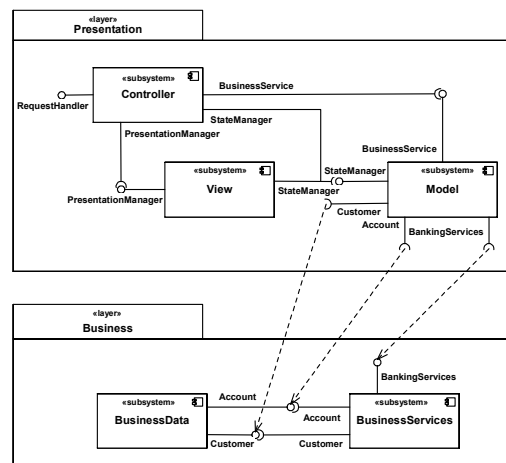


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

11

## Module Example – Subsystems

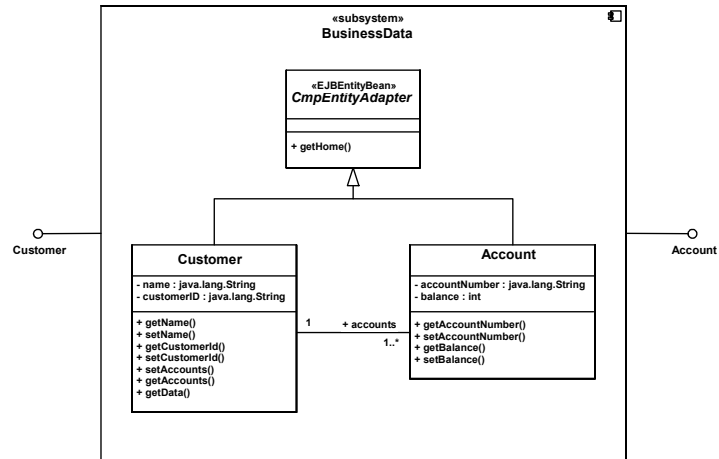


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

12

## Module Example – Classes

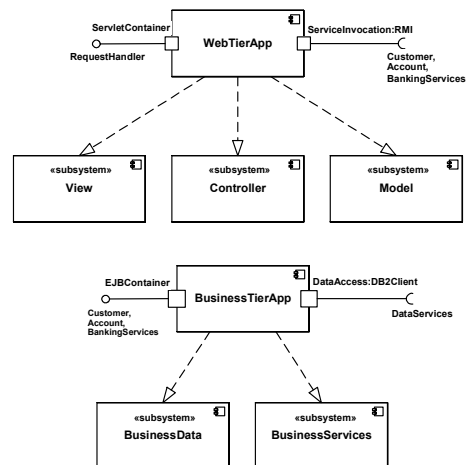


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

13

## C&C Example – Runtime Components

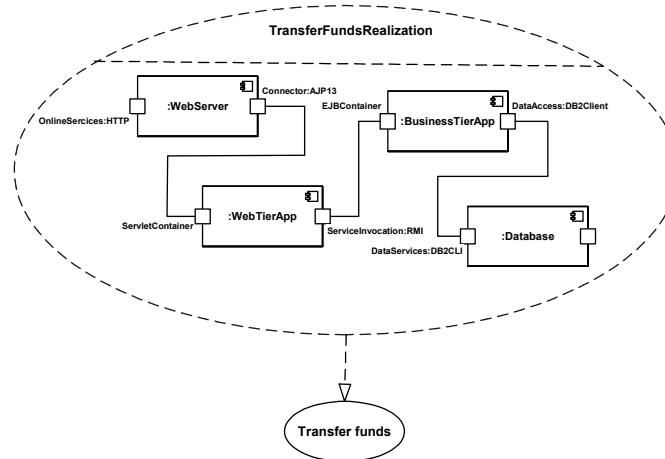


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

14

## C&C Example – Collaborations

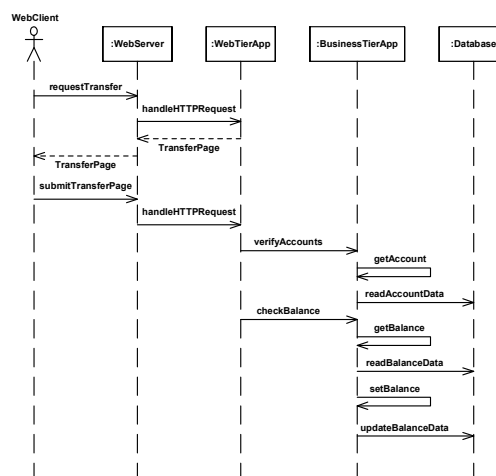


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

15

## C&C Example – Interactions



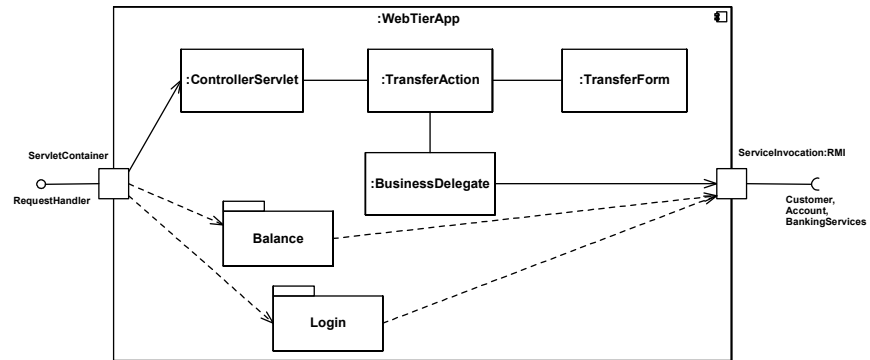
4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

16



## C&C Example – Structured Classifiers

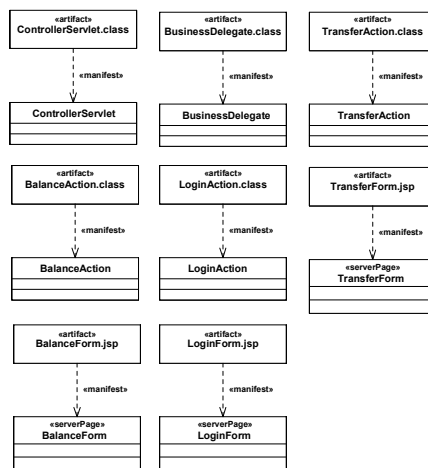


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

17

## Allocation Example – Artifacts (1)

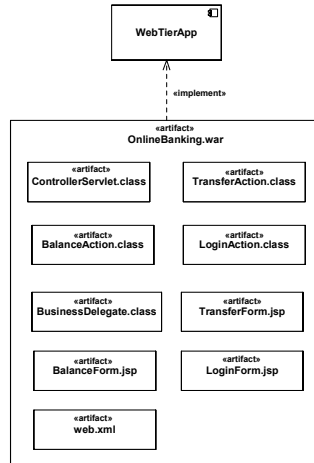


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

18

## Allocation Example – Artifacts (2)

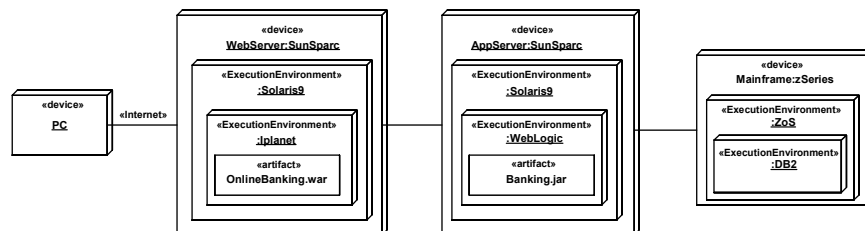


4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

19

## Allocation Example – Nodes



4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

20

## Summary

---

- ▣ New or changed concept areas that impact architecture documentation
  - Composite structures
  - Components
  - Deployments
- ▣ Gauging the effectiveness of UML documentation
  - Completeness
    - ▣ Minimal ambiguity
    - ▣ Minimal redundancy
  - Integration
    - ▣ Explicit mapping between views
    - ▣ Explicit requirements traceability
    - ▣ Broad lifecycle support

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

21

## Summary – contd.

---

- Stakeholder communication
  - ▣ Excellent for technical stakeholders
  - ▣ Non-technical stakeholders prefer cartoons

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

22

## Q&A

---

4/6/2005

SEI Software Architecture Technology User Network  
April 6-7, 2005 – Pittsburgh, PA

23