Modeling with StarUML

by Xinjiang Lu
Outline

- Introduction
- 4+1 View Approach
- An Example: Bank ATM System
Introduction

- Background
- UML Diagrams
- StarUML
Unified Modeling Language (UML) has been available since 1997, and UML 2 was released in 2004, building on an already successful UML 1.x standard. UML 2 comes with 13 basic diagram types to support Model Driven Architecture (MDA) and Model Driven Development (MDD).

UML is continuously expanding standard managed by OMG(Object Management Group)
Philippe Kruchten originally presented the 4+1 View Model to describe the architecture of software-intensive systems. This approach uses multiple views to separate stakeholders’ concerns. The 4+1 View Approach is widely accepted by the software industry to represent application architecture blueprints.
UML diagrams (UML2)

- Package Diagram
- Class Diagram
- Object Diagram
- Composite Structure Diagram
- Component Diagram
- Deployment Diagram

- Use Case Diagrams
- Activity Diagrams
- State Machine Diagrams
- Communication Diagrams
- Sequence Diagrams
- Timing Diagrams
- Interaction Overview Diagrams

Structural Diagrams
Behavioral Diagrams
StarUML is an open source project to develop fast, flexible, extensible, featureful, and freely-available UML/MDA platform running on Win32 platform.

GOAL: build a software modeling tool and also platform that is a compelling replacement of commercial UML tools such as Rational Rose, Borland Together and so on.
Go to http://staruml.sourceforge.net/en/download.php
What can you do using StarUML?

- StarUML supports the following UML diagrams
  1. Use Case Diagram
  2. Class Diagram
  3. Sequence Diagram
  4. Collaboration Diagram
  5. Statechart Diagram
  6. Activity Diagram
  7. Component Diagram
  8. Deployment Diagram
  9. Composite Structure Diagram (UML 2.0)

- Code generation can be done for Java, C++ and C# as well as Reverse Engineering

- Documentation can be generated for all of the Microsoft Office suite programs such as Word, Excel and PowerPoint.
Create a NEW PROJECT

□ Select the [File] -> [New Project By Approach] menu.
Select the [4+1 View Model]
Now there are 4 views and 1 Scenarios in the Model Explorer
To draw a Use Case Diagram, double click the **Scenarios**

You can start drawing in the default ‘**Main**’ or [Add Diagram] -> [Use Case Diagram] by right-clicking on the Scenarios.
The following elements are available in a usecase diagram.

- Actor
- UseCase
- Association
- Directed Association
- Generalization
- Dependency
- Include
- Extend
- System Boundary
- Package
Create Actor
Create Use Case
Create Association

[Image of a UML diagram showing an association between a Customer and a Balance inquiry]
Example
4+1 View Approach
The fundamental organization of a system can be represented by:

- Structural elements and their interfaces that comprise or form a system
- Behavior represented by collaboration among the structural elements
- Composition of Structural and Behavioral elements into larger subsystems

Such compositions are guided by desired abilities

Also, there are cross-cutting concerns
Architecture also means different things to different stakeholders

- A Network Engineer would only be interested in the hardware and network configuration of the system
- A Project Manager in the key components to be developed and their timelines
- A Developer in classes that make up a component
- A Tester in scenarios

We need multiple viewpoints for distinct stakeholders’ needs, showing what is relevant while masking the details that are irrelevant
4+1 View

CONCEPTUAL

Logical View
Functionality

Use Case View
Scenarios

Process View
Performance
Scalability
Throughput

PHYSICAL

Implementation View
Configuration Management

Deployment View
Modeling Logical View with UML2

- Package Diagrams
- Composite Structure Diagrams
- Class Diagrams
  - Object Diagrams
  - State Diagrams
Modeling Process View with UML2
Modeling With UML 2 (Cont.)

CONCEPTUAL

Logical View
- Class, Object, Package, Composite Structure, State Machine

Use Case View
- Use Case, Activity

Process View
- Sequence, Communication, Activity, Timing, Interaction Overview

PHYSICAL

Implementation View
- Component

Deployment View
- Deployment
An Example: ATM SubSystem
System level
ATM transaction and customer authentication
ATM maintenance and diagnostics
Bank account taxonomy

Class Diagram

- Liability Type
- Bank Account
  - Business Account
  - Personal Account
- Checking Account
  - Checking with Interest
  - Certificates of Deposit
- Saving Account
  - Children's Saving Account
  - Health Saving Account
- Account Type
  - No Frills Checking
  - Account Interest
Behavioral state machine – Bank ATM
Summary

- Brief introduction on StarUML
- 4+1 view architecture methodology
- A practical example