Génie Logiciel Avancé

Brief Revision of UML

Burkhart Wolff
wolff@lri.fr
Plan of the Chapter

- The UML notation is used as document - core in SE processes (such as the V model)
- Syntax and semantics of class model elements and their visualization in diagrams
  - Class Invariants
  - Constraints
  - Operations
  - Pre- and Post-Conditions
- Syntax and semantics of state machines
  Specify system components for test and verification
The UML ...

- ... is the **Unified Modeling Language**
- ... is a normed data-structure, a „technical format“ of model-elements (that may contain other model-elements) with *consistent* naming for
  - various system descriptions
  - various code formats
- ... has various external representations
  - as **XMI** exchange format (tool-independent in theory ...)
  - as UML diagrams
The UML offers the advantage ...

- ... of being a basis for Integrated Development Environments (IDE’s like ArgoUML, Poseidon, Rational Rose, Prodigé, ...)
The Shapes Project - shapes class diagram - ArgoUML

Order By Type, Name
- Profile Configuration
  - shapesmodel
    - shapes class diagram
      - Use Case Diagram 1
        - unattachedCollaboration
          - double
          - int
          - void
          - (Unnamed Generalization)
          - (Unnamed Generalization)
          - (Unnamed Generalization)
          - (Unnamed Generalization)
          - create
          - TD transient
          - TD volatile
          - (Unnamed Association)
        - OneDimensional
          - +getLength() : double
          - +newOperation() : void
        - TwoDimensional
          - +getArea() : double
    - Polygon
      - <create> +Polygon() : void
      - +Vertices
    - Point
      - +x : int
      - +y : int

Polygon has multiple base classes, but Java does not support multiple inheritance. You must use interfaces instead.

This change is required before you can generate Java code.

To address this, use the "Next>" button, or manually (1)
The UML offers the advantage ...

- ... of being a basis for Integrated Development Environments (IDE's like ArgoUML, Poseidon, Rational Rose, ...)
- ... to offer „object-oriented“ specifications
- ... to offer a formal, mathematical semantics (well, at least to parts of the UML)
- ... to be fairly widely used in industry, even if not always supported entirely
- ... is the basis for a whole software-engineering paradigm called Model-Driven Engineering (MDE).
The UML 2.0 Diagrams (for corresp. models)

- UML, Version 1.1: 9 types of diagrams
- UML, Version 2.0 adds

  4 more types of diagrams

  - structure composition
  - communication
  - packaging
  - temporal constraints (timing)
The UML 2.0 Diagrams (for corresp. models)
Principal UML diagram types (1)

- Structure and Visualization
  - Use Case Models and Use Case Diagrams
  - Sequence Models and Sequence Diagrams
  - State Machines and State Charts
  - Class Models and Class Diagrams
  - Object Graphs and Object Diagrams

All these Model Elements are described in a UML-document itself, the „Meta-Object-Framework“ (MOF)
Main UML diagram type:

- **Class Diagrams** („Diagrammes de classes“): the static **structure** of the DATA of the system
  - the classes of interest to be represented in the system
  - the relations between classes
  - the attributes and the methods
  - the types, required/defined interfaces ...

  can be used for top-level views as specific interfaces for local code ...
Main UML diagram type:

- **Class Diagrams** („Diagrammes de classes“):
  - the static **structure** of the DATA of the system
  - the classes of interest to be represented in the system
  - the relations between classes
  - the attributes and the methods
  - the types, required/defined interfaces ...

This is what we will refine in the sequel!