

UML CASE Tools



Your friendly assistant
Chair of Programming Methodology

Spring Semester 2009

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Agenda for Today

1. Why need UML tools?
2. Main UML tools
3. StarUML
4. Introduction of Case Study

Agenda for next week

6. Continuing development of case study

Brief re-cap of UML

- Modeling language used in **analysis**, **design** and **implementation** phases
- **Textual** and **graphical** notation to document specification
- Main diagram types:
 - **structural:** **class**, component, deployment diagrams
 - **behavioral:** **use case**, **sequence**, **statechart** diagrams

Why need UML tools?

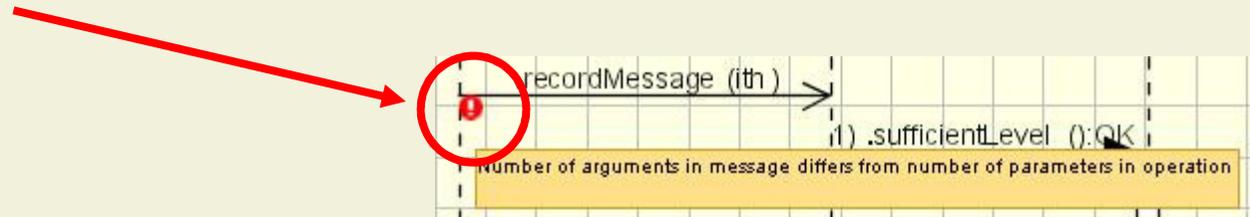
- Advantages of “e-design” over “paper-design”
 - modifications made easier
 - communication between designers easier
 - co-ordination and handling of large projects easier



Why need UML tools?

- Advantages of **UML-tools** over **drawing-tools**

- **uniform** notations
- certain **checks** provided



- UML-tools provide **add-ons**

Add-ons

- Document generation
 - HTML, PDF etc.
- Round-trip engineering
 - code generation from diagram (forward engineering)
 - diagram generation from code (reverse engineering)
- Test generation
- Simulation
- Model validation and verification



Level of sophistication

UML tools

- **Big competition** among UML tool vendors
 - plg.uwaterloo.ca/~migod/uml.html
 - objectsbydesign.com/tools/umltools_byCompany.html

- Many **free tools/editions** with limited capabilities

- **Compatibility** ensured by XML dialect
 - XMI – XML Metadata Interchange
 - may contain diagram layout info
(Diagram Interchange Standard)

Main UML tools

■ Commercial

- Rational Rose (IBM)
- Together Designer (Borland)
- Rhapsody (I-Logix)
- Poseidon (Gentleware)

■ Free

- UMLet
- EclipseUML
- Visual Paradigm (Visual Paradigm)

■ Open source

- ArgoUML (Tigris)
- StarUML (Sourceforge)



And many, many more...

Choosing a UML tool

- Platform support
- UML 2.0 and XMI support
- Support of all diagram types
- Intuitive and clear GUI
- Add-ons
- Actively maintained
- Document generation and printing support

StarUML

■ Pros

- Open-source
- UML 2.0 and XMI
- All diagram types
- Code generation (C++, C#, Java)
- Diagram exporting and printing



■ Cons

- **Windows only**
- Not maintained since 2005

<http://staruml.sourceforge.net>

Other tools you could use

- Any drawing tool (e.g., Paint or Gimp)
- NetBeans' or Eclipse's UML plug-in
- ArgoUML
 - Pros: Platform independent and small
 - Cons: no undo feature and sometimes tedious to use
- UMLet: use for small models only

Tool demo via Case Study

- Digital sound-recorder (Dictaphone)
- Based on:

Ivan Porres Paltor, Johan Lilius:

Digital Sound Recorder: A case study on designing embedded systems using the UML notation



For next week: download diagrams from website
(either StarUML or PDF file)

Main features

- Capacity for **10 different messages**, each max. of 2 minutes
- **Messages** can be **recorded**, **played back** and **deleted**
- **Messages** can be **locked/unlocked** to prevent unwanted deletion/overwriting
- LCD display

Problem Statement

1. The recorder stores up to 10 messages
2. Each message is max. 2 minutes long
3. The user can record message
4. Recording of a message ends after 2 minutes or when the user stops recording
5. Recording destroys the original message at chosen slot
6. Sufficient level of battery is checked before recording message
7. Message of a given slot can be replayed
8. Sufficient level of battery is checked before replaying message
9. Messages can be locked/unlocked
10. Locked messages cannot be deleted or over-written by recording to the same slot

11. User uses LCD display and buttons to interact with recorder

BACKUP

Problem Statement

1. The **recorder stores** up to 10 **messages**
2. Each message is max. 2 minutes long
3. The **user** can **record** message
4. Recording of a message **ends** after 2 minutes or when the user **stops** recording
5. Recording **destroys** the original message at chosen slot
6. Sufficient level of **battery** is **checked** before recording message
7. Message of a given slot can be **replayed**
8. Sufficient level of battery is checked before replaying message
9. Messages can be **locked/unlocked**
10. Locked messages cannot be **deleted** or **over-written** by recording to the same slot

11. User uses LCD **display** and **buttons** to **interact** with recorder