UML CASE Tools

- StarUML -
Agenda for today

Why need UML tools?
Main UML tools
StarUML

Case Study using StarUML
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
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
- Omondo
- Visual Paradigm (Visual Paradigm)

Open source
- ArgoUML (Tigris)
- StarUML

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

Free and open source

Supports
- Windows only
- UML 2.0 and XMI
- all diagram types
- diagram printing
- code generation (C++, C#, Java)
- EJB and GoF pattern application

Download: staruml.sourceforge.net

StarUML is the tool we suggest. You can use any other tool for the project.

We accept diagrams in PDF or in StarUML’s .uml format.
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
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

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

- User uses LCD display and buttons to interact with recorder
Your tasks (suggestion)

1. Develop use case diagram
   Write flow of event for one or two interesting use cases

2. Develop sequence diagrams for selected use cases
   Remember: need not model all possible scenarios

3. Develop class diagram

4. Refine sequence diagrams from step 2 according to the classes you have identified in step 3
   Remember: you might need to refine your class diagram

5. Draw state diagram for classes with interesting behavior
   Remember: typical candidates are control objects