

Software Engineering

Advances in Software Architecture From Objects First to Visual Programming

Lecture 21
2007/11/13



Last Lecture: UML

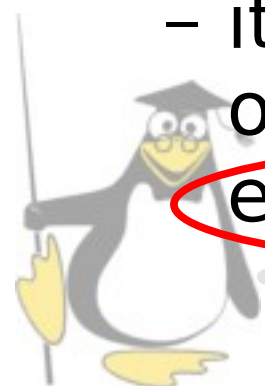
- usecase diagrams
- object diagrams
- class diagrams
- (activity diagrams)

- Announcement: <http://www.fujaba.de/>
 - allows MDA (more in my next lecture on 16th of November, Advances in Software Architecture)



Fujaba - <http://www.fujaba.de>

- From UML to Java And Back Again
- Fujaba is an “easy to extend UML Story Driven Modelling and Graph Transformation platform with the ability to add plug-ins.”
 - it “combines UML class diagrams and UML behaviour diagrams (Story Diagrams) to a powerful, easy to use, yet formal system design and specification language.”
 - it “supports the generation of Java sourcecode out of the whole design which results in an executable prototype.”



Fujaba

- UML
- Story Driven Modeling
- Graph Transformation
- Class Diagrams \leftrightarrow Story Diagrams
- java sourcecode \rightarrow executable prototype



Scenario: Towers of Hanoi



http://en.wikipedia.org/wiki/Towers_of_hanoi



Usecase 1 – Hanoi setup

- Pre: 4 discs, different sizes (1,2,3,4), empty table
- define three places on table (tower-place)
- put the four discs in the right order on the first place
- Post: 4 discs, ordered on first tower-place

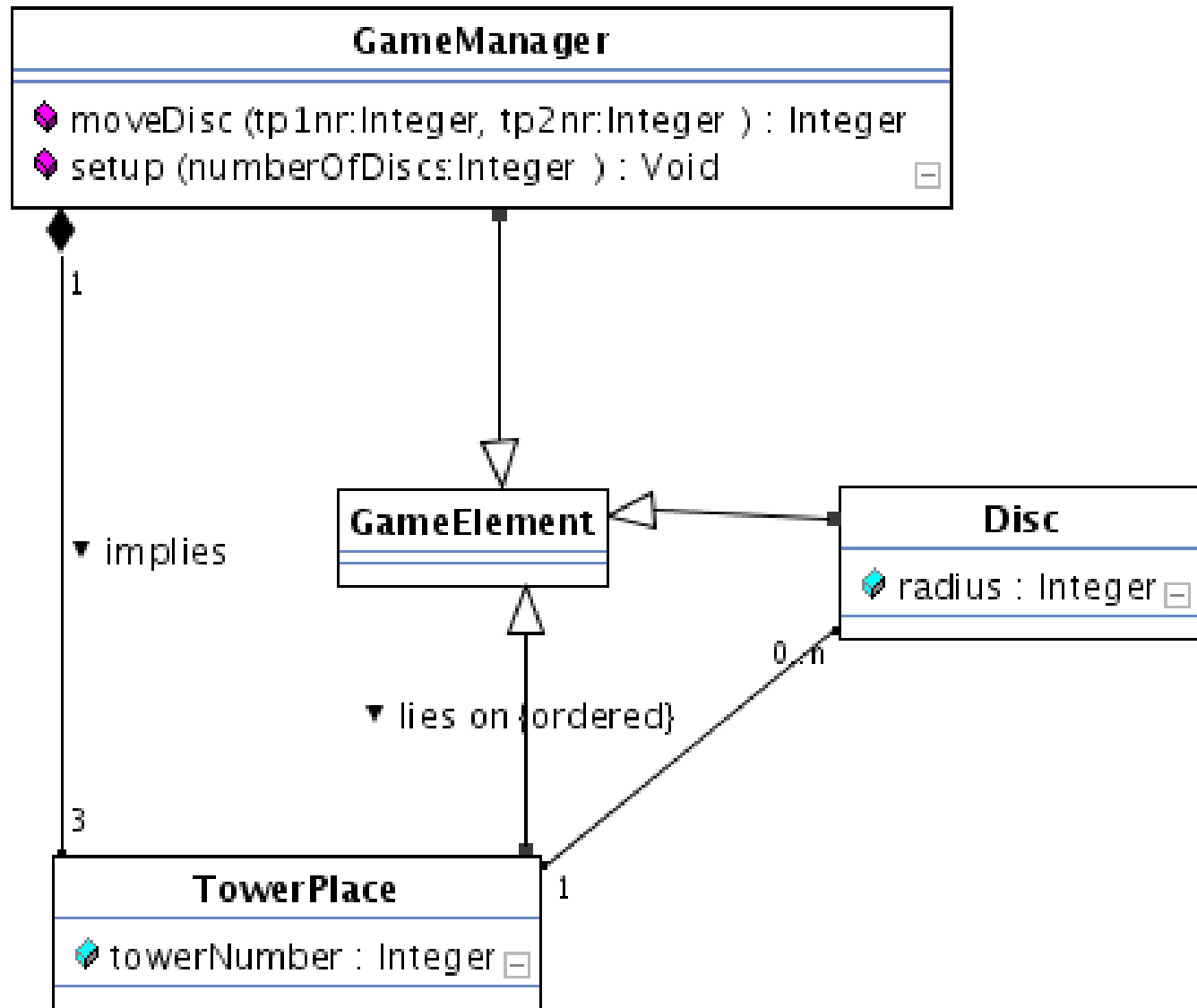


Usecase 2 – Initial move

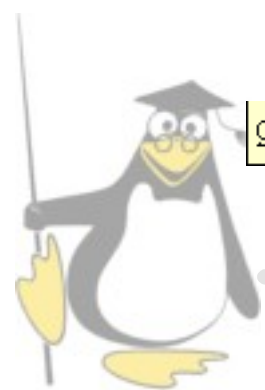
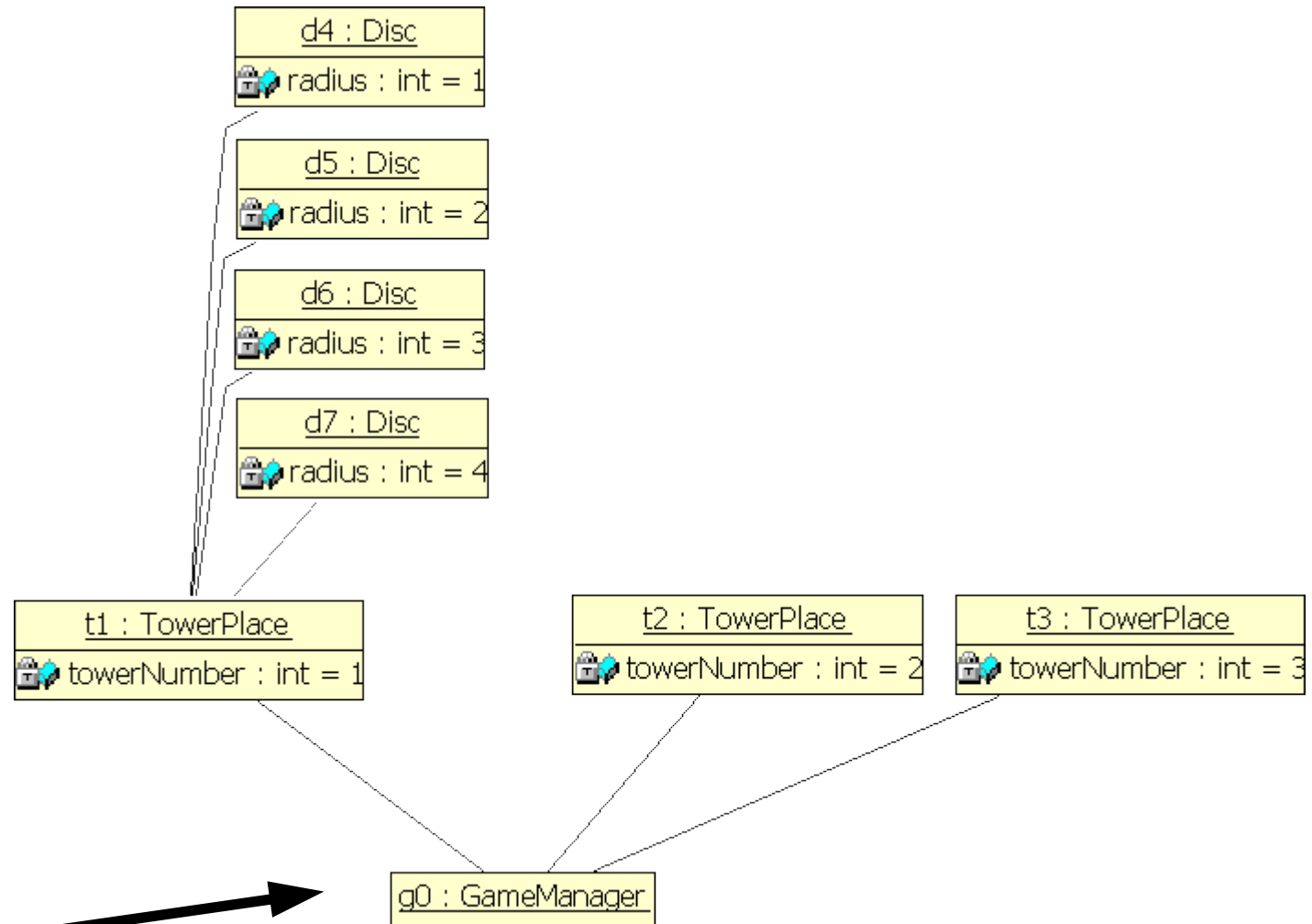
- Pre: 4 discs, ordered (4,3,2,1) on first tower-place
- Move disc with radius 1 to second tower-place
- Post: 3 discs (4,3,2) on first tower-place, 1 disc (radius 1) on second tower-place



Class Diagram - Hanoi



Graph Transformation



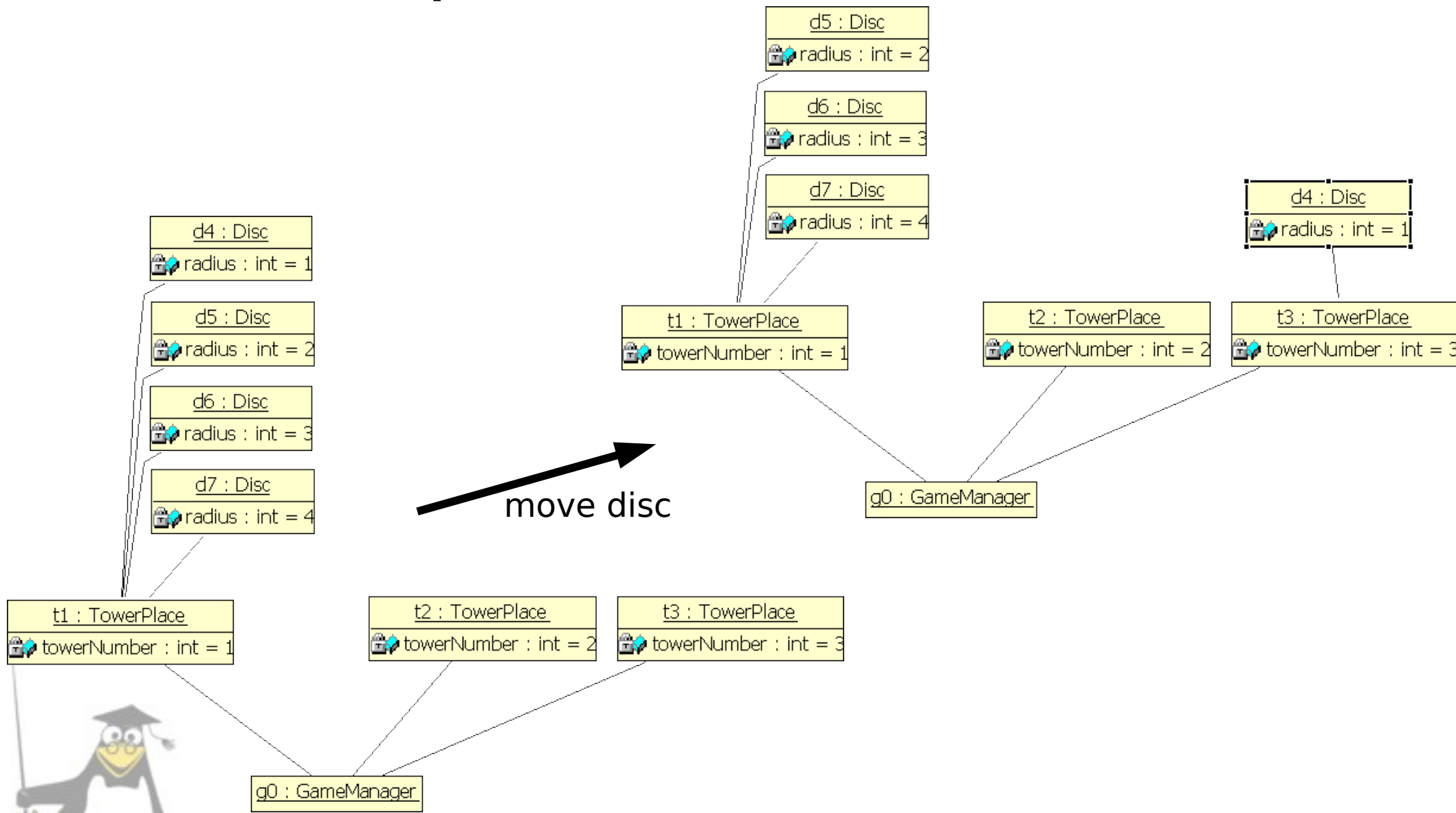
g0 : GameManager



setup

g0 : GameManager

Graph Transformation



Practical work

- (install Java-JDK)
- install Fujaba
- load Hanoi_initial-Project
- start Dobs
 - create objects
 - connect objects
 - play through usecases
 - record steps for method-implementation



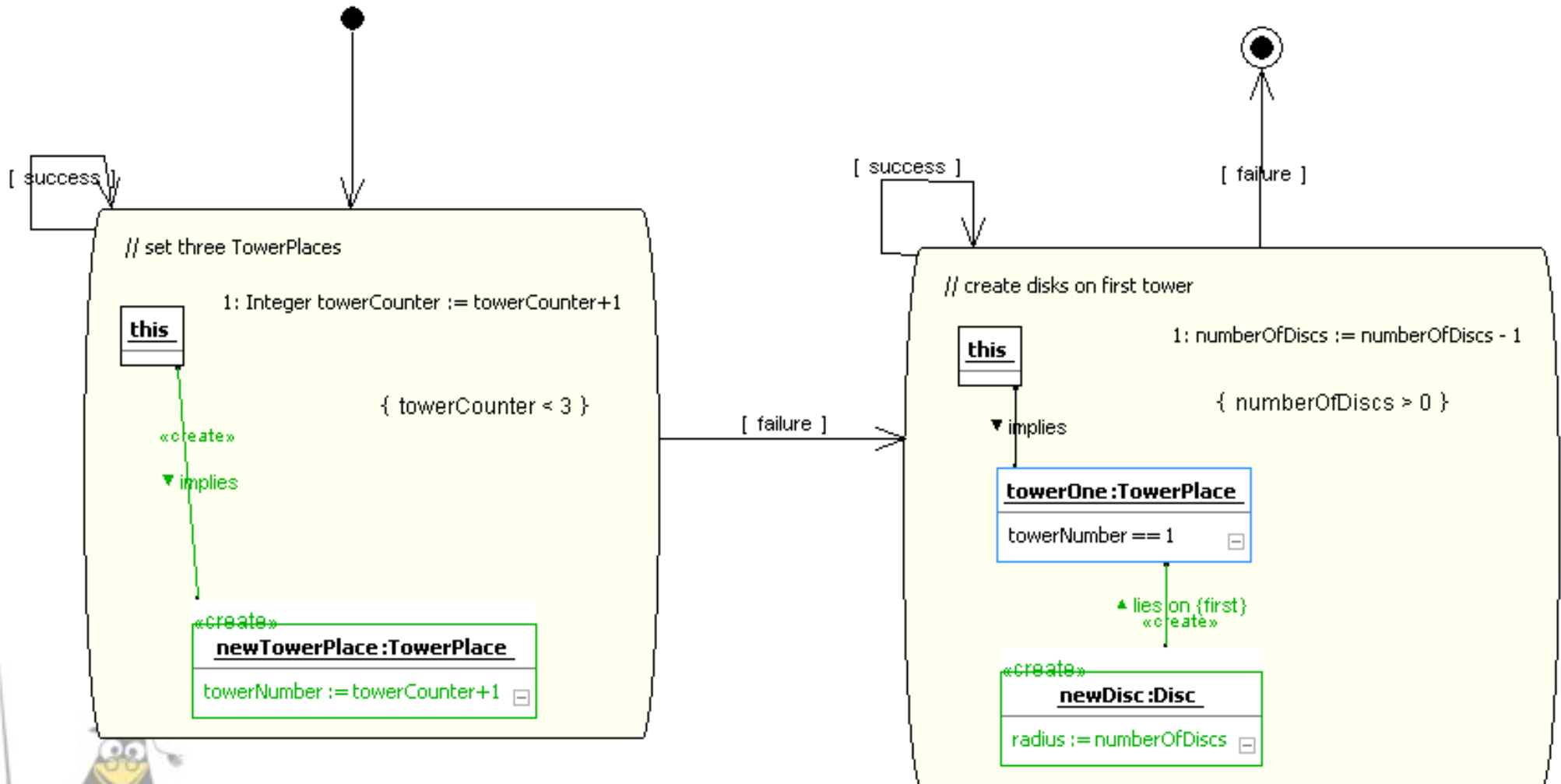
Create Methods

- Story driven modeling (shown in Fujaba)
- If you do not attend this class, compare documentation on <http://www.fujaba.de> (very limited) or drop into my office (311)



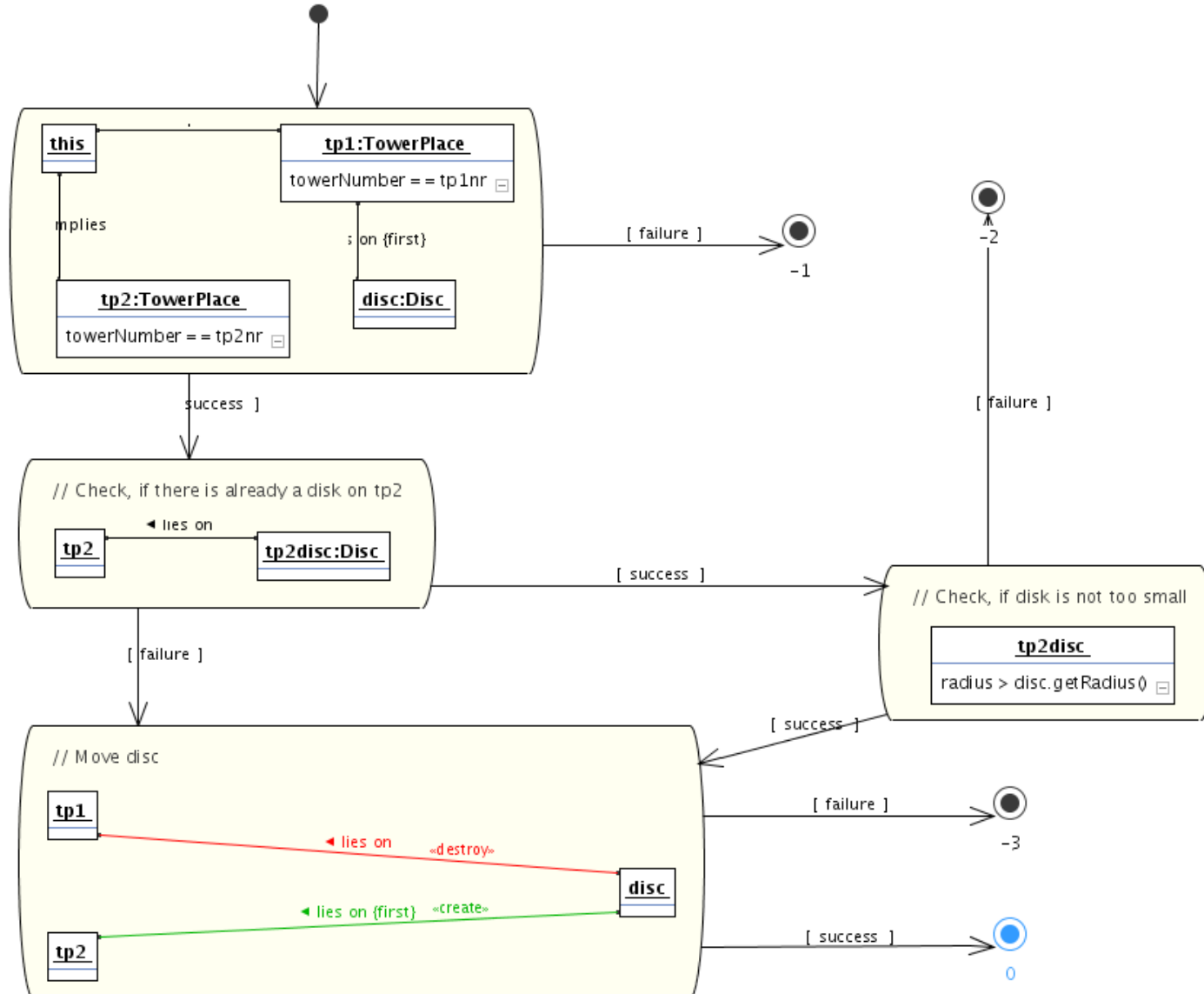
Setup (Example)

GameManager::setup (numberOfDiscs: Integer): Void



Move Disc (Example)

GameManager::moveDisc (tp1nr: Integer, tp2nr: Integer): Integer



A Complex Example: eHomeConfigurator

- The class diagram: eHomeModelComplete.pdf
- A story diagram: Activity_Service_install.pdf (resulting in about 1000 lines of java-code)
- Find more at: <http://phd.ulno.net>

The screenshot displays the eHomeConfigurator v2.0 interface. The main window shows a service selection wizard with a class diagram in the background. The wizard is titled "Wizard : Service Selection" and contains instructions for selecting services. A list of services is shown with checkboxes for installation and dropdown menus for location and device selection.

Wizard : Service Selection

Please select which services you want to install in your eHome, their default strategies concerning the room selection and the strategies concerning the installation of additional devices.

Furthermore, please note that you will be able to select and deselect each room individually later on in this wizard.

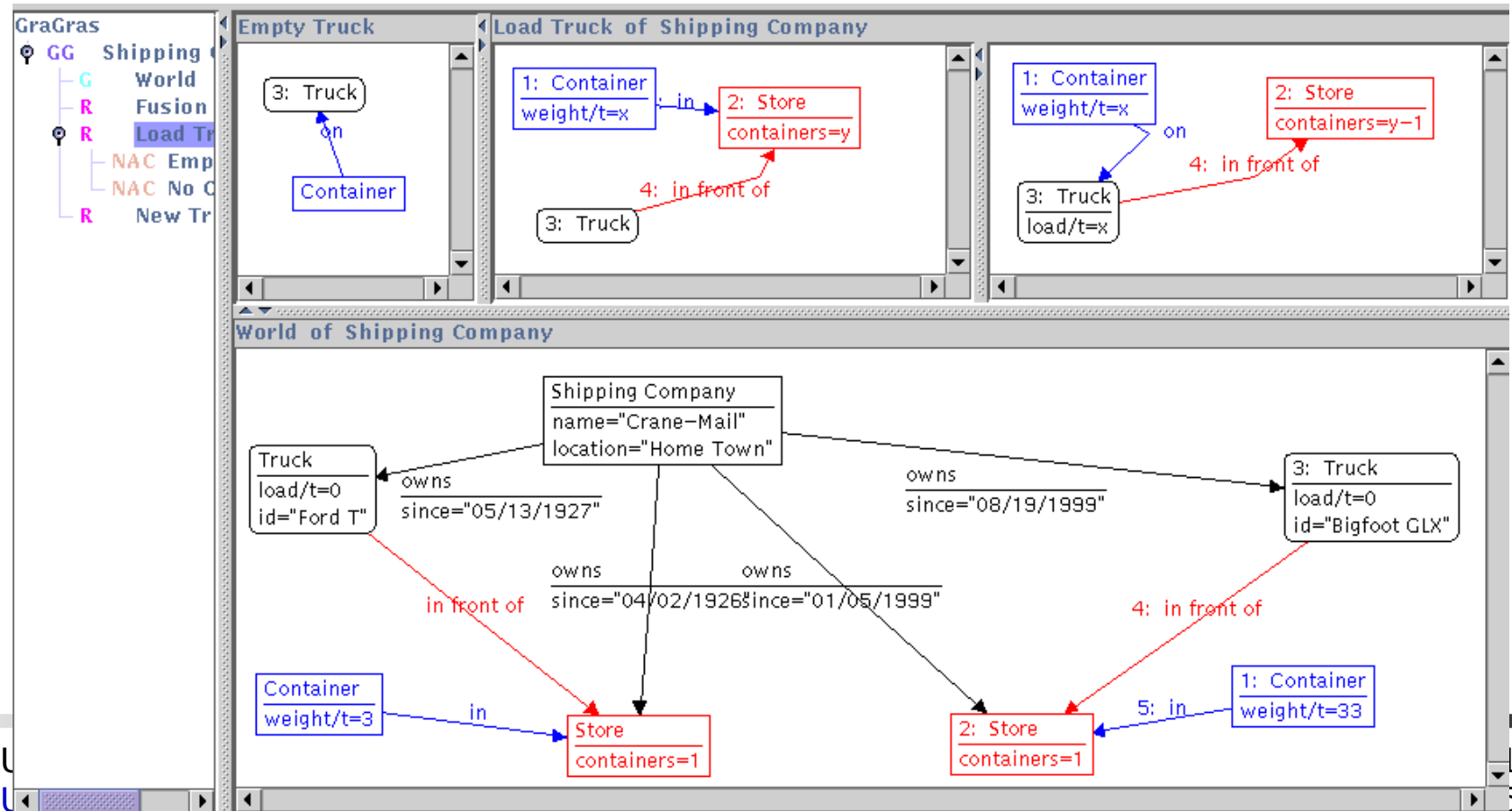
Dependent services will be installed automatically!

| | | | |
|----------------------|---|----------------------|------------------------|
| Lighting Service | <input checked="" type="checkbox"/> Install | Select All Locations | Necessary Devices O... |
| Security Service | <input checked="" type="checkbox"/> Install | Select All Locations | Necessary Devices O... |
| All Off Service | <input checked="" type="checkbox"/> Install | Select All Locations | Necessary Devices O... |
| All On Service | <input checked="" type="checkbox"/> Install | Select All Locations | Necessary Devices O... |
| Music Follows Person | <input checked="" type="checkbox"/> Install | Select All Locations | Necessary Devices O... |
| Light Motion Service | <input type="checkbox"/> Install | Select All Locations | Necessary Devices O... |

Buttons: Stop, Next

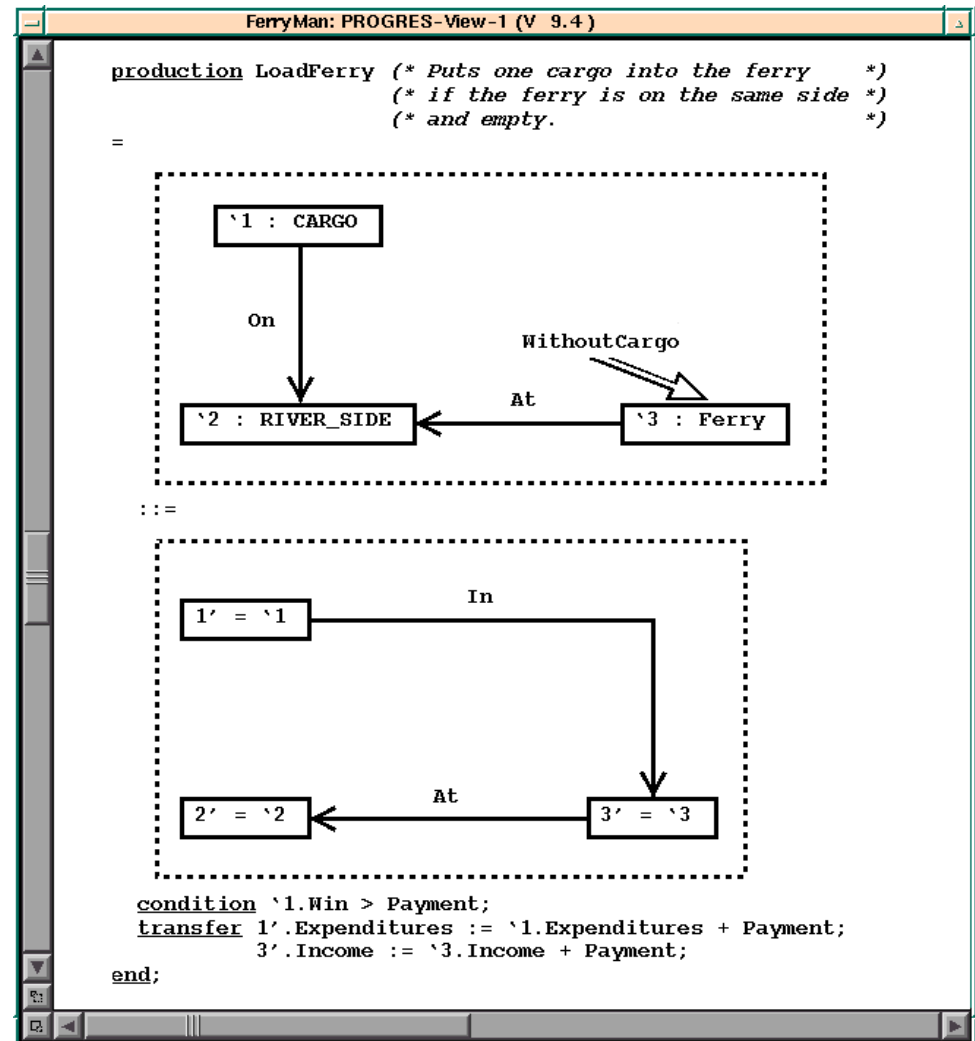
Other Graph Transformation Based Systems

- AGG:
The Attributed Graph Grammar System
<http://tfs.cs.tu-berlin.de/agg>



Other Graph Transformation Based Systems

- Progres:
An integrated environment and very high level language for PROgrammed Graph REwriting Systems



<http://www-i3.informatik.rwth-aachen.de/progres>



Lessons Learned

- Objects First Design
- Story Driven Modeling
- Visual Programming Language

