

Systems Modeling

Lecture
2009/02/19



Tiny Excursion: Distributed Revision Control

- Revision control + distribution
- Good for independent or small developer teams
- Working copy is branch
- No central copy of repository needs to exist
- Every participant has their own local branch
- behaves locally like central revision control system



Examples for DRCSSs

- **Bazaar**
 - GNU Mailman, MySQL, Gnash, Squid, GNUPdf
- **Darcs**
 - ghc (Glasgow Haskell Compiler), DokuWiki, Nitro
- **Git**
 - (by Linus Torvalds), used for the Linux Kernel, X.org, under investigation by KDE
- **Mercurial**
 - Mozilla project, NetBeans and OpenJDK



Mercurial

- Standard cli commands:
 - hg init
 - hg add
 - hg commit
 - hg pull/push
 - hg --help
- Good integration in various IDEs
 - MercurialEclipse
 - TortoiseHG (Windows explorer extension)
 - Netbeans, VisualHG, hg4idea, ...



Use of DRCS in this course

- Mercurial – on special agreement one of the others
- Use it in your team for revision management, not only for code but also for documentation
- Check in often, synchronize often
- Course project has to be delivered at the end as final source+.hg directory, homework, too (just zip or tgz source-folder)
- I will verify activity of team members



Towers of Hanoi



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

Discs should be moved interactively.



User Story

- Title: Move red disk to second place
- Precondition: red d. (size 1), yellow d. (size 2), blue d. (size 3), and orange disc (size 4) sorted on initial towerplace (of 3)
- Action: The player moves the red disc onto the second towerplace
- Postcondition: yellow, blue, and orange disc sorted on initial towerplace (of 3), red on second towerplace



Use Cases and Use Case Diagrams

- Read the provided material (10 minutes)
- Compile down with your neighbor a one page document, what the key elements of Use Cases and Use Case Diagrams are (10 minutes)
- Team up to teams with four and compile a one page document how you could relate user stories and use cases, create a use case summary for the Towers-of-Hanoi user story (10 minutes)



Object

- A thing, being or concept
- A language mechanism for binding data with methods that operate on that data
- A representation of a physical object
- Instance of a class
- Objects represent “things” from the real world, or from some problem domain

- How do objects and examples relate?



Distinguishing Objects

- How can these be distinguished?
 - all persons in the world for the purpose of sending mail
 - all persons in the world for the purpose of criminal investigations
 - all customers with safe deposit boxes in a given bank
 - all customers of a telephone company for billing purposes
 - all employees of a company to restrict access for security reasons



Modeling Objects

Red-Disc
size: 1



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

Yellow-Disc
size: 2



Modeling Objects Relations/Interaction



User Story (revisited), how to identify objects?

- Title: Move red disk to second place
- Precondition: red d. (size 1), yellow d. (size 2), blue d. (size 3), and orange disc (size 4) sorted on initial towerplace (of 3)
- Action: The player moves the red disc onto the second towerplace
- Postcondition: yellow, blue, and orange disc sorted on initial towerplace (of 3), red on second towerplace

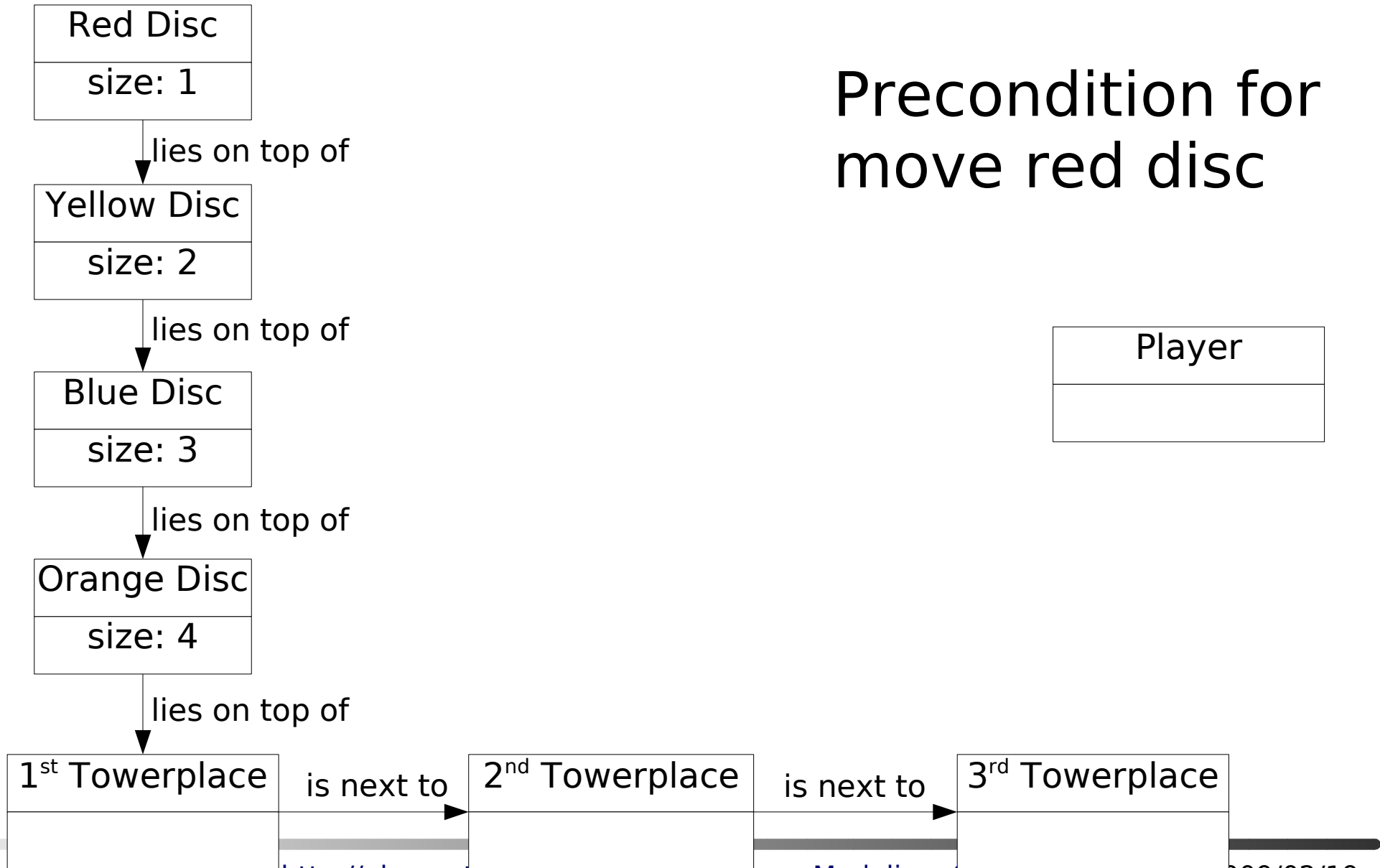


User Story, identify objects

- Title: Move red disk to second place
- Precondition: red d. (size 1), yellow d. (size 2), blue d. (size 3), and orange disc (size 4) sorted on initial towerplace (of 3)
- Action: The player moves the red disc onto the second towerplace
- Postcondition: yellow, blue, and orange disc sorted on initial towerplace (of 3), red on second towerplace



Object Diagram

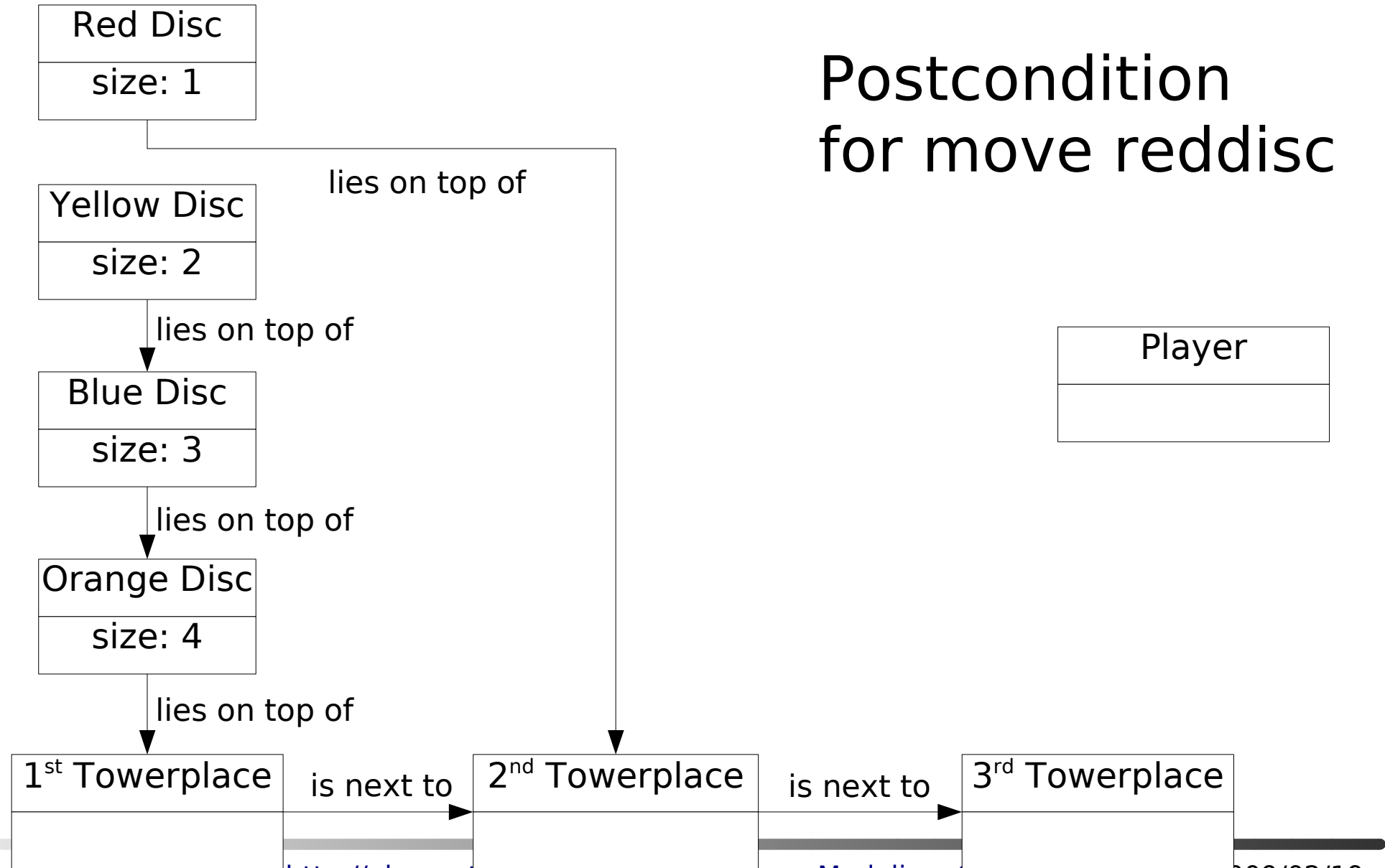


Precondition for
move red disc



Object Diagram

Postcondition
for move reddisc



Object Diagrams ArgoUML

- Not supported
- Hack
 - For very simple object diagrams use deployment diagram
 - Better: use class diagram
 - ignore multiplicity
 - use in different document from your project document
- Or use Visio or Openoffice
- Demo



Homework

- Get acquainted with Mercurial
- Check in the scenarios and user stories in your repository
- ≥ 4 Use cases summary + diagram
- Object diagrams for pre- and postcondition for all 30 user stories \rightarrow 60 diagrams
- Check in all documents
- Merge your repositories
- Send repository archive or link to me until March 1st

