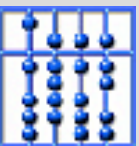


Automatic Layout Of Presentation Elements In Augmented Reality

Systementwicklungsprojekt

Michael Riedel
riedel@in.tum.de

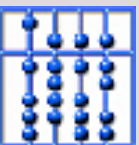
20.04.2004



Contents

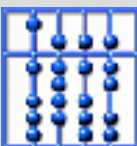
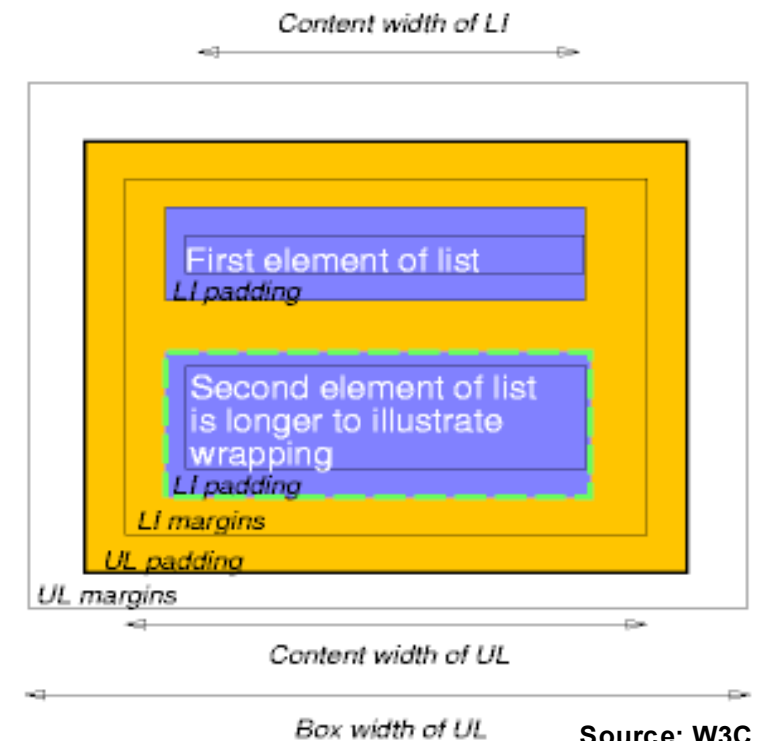
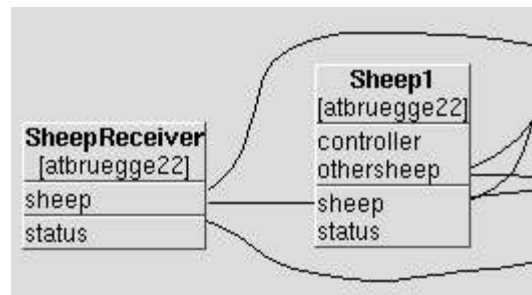
► Why Automatic Layout?

- Applications & Sample Layout Problems
- Requirements for Layout in DWARF
- System Design & Object Design
- Layout Demos
- How to do it better? Who Will do it Better?



What is Automatic Layout Anyway?

- Fixed manual layout in print, GUI, etc.
 - predefined by human layout experts
- Automatic typesetting (T_EX)
- Semi-automatic layout of GUIs (CSS, Swing, etc.)
- Automatic graph layout
- Automatic map labeling



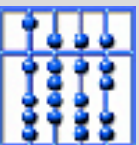
Why Automatic Layout in AR?

- Manual layout in advance is not possible
- Frequent change of layout problem
 - real world objects → layout
 - perspective → view → layout
- User can hardly layout manually
- Naive layout will result in undesired occlusion
- Laying out additional information *is* AR



Contents

- Why Automatic Layout
- ▶ **Applications & Sample Layout Problems**
- Requirements for Layout in DWARF
- System Design & Object Design
- Layout Demos
- How to do it better? Who Will do it Better?

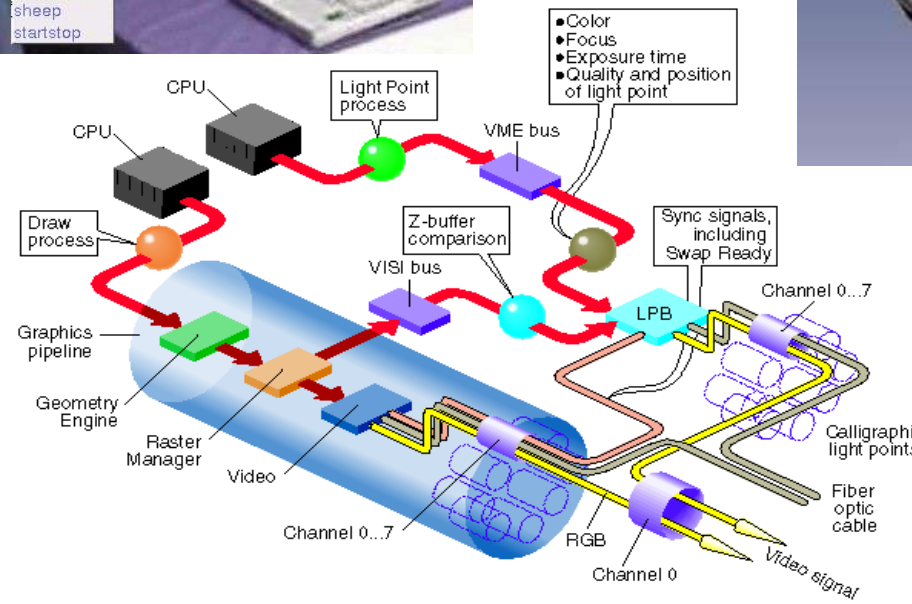
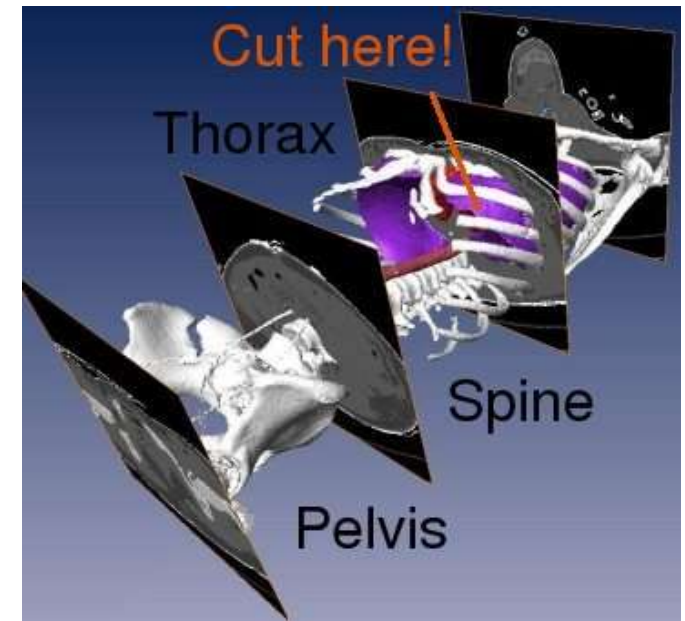


Proposed Applications

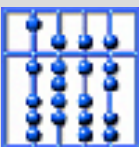


DIVE 3D

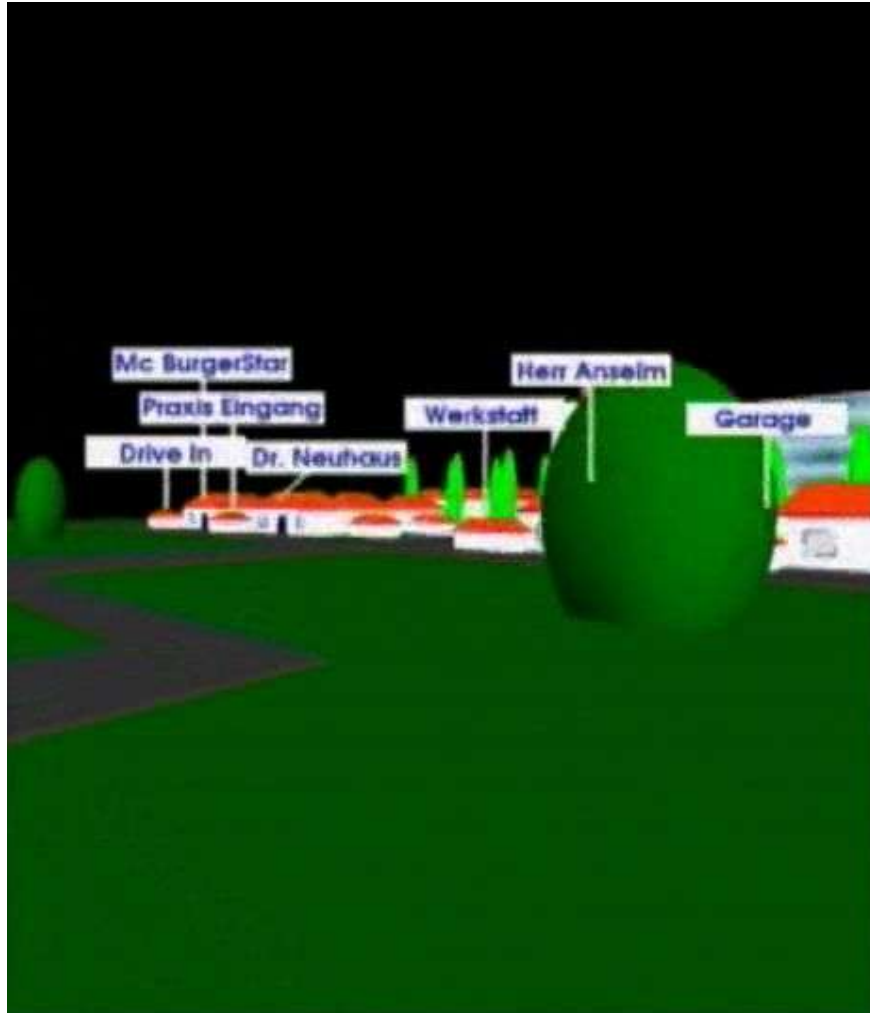
Medical Visualization



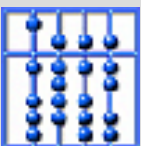
Interactive Technical Manuals



Automatic Layout in CAR Project

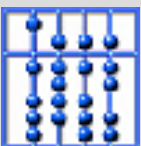
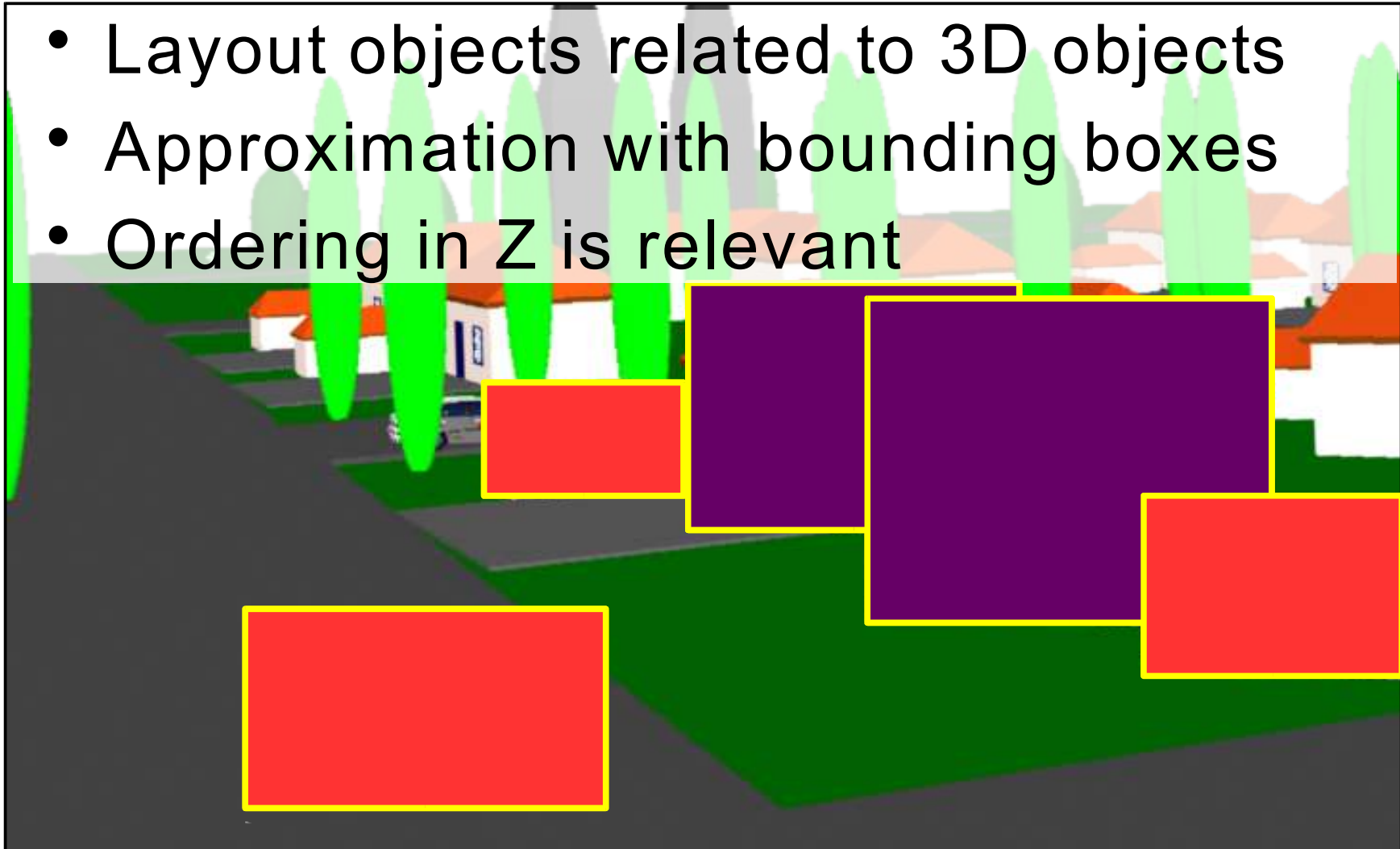


- Egocentric navigation
- City guide
- Tourist information
- Historical information



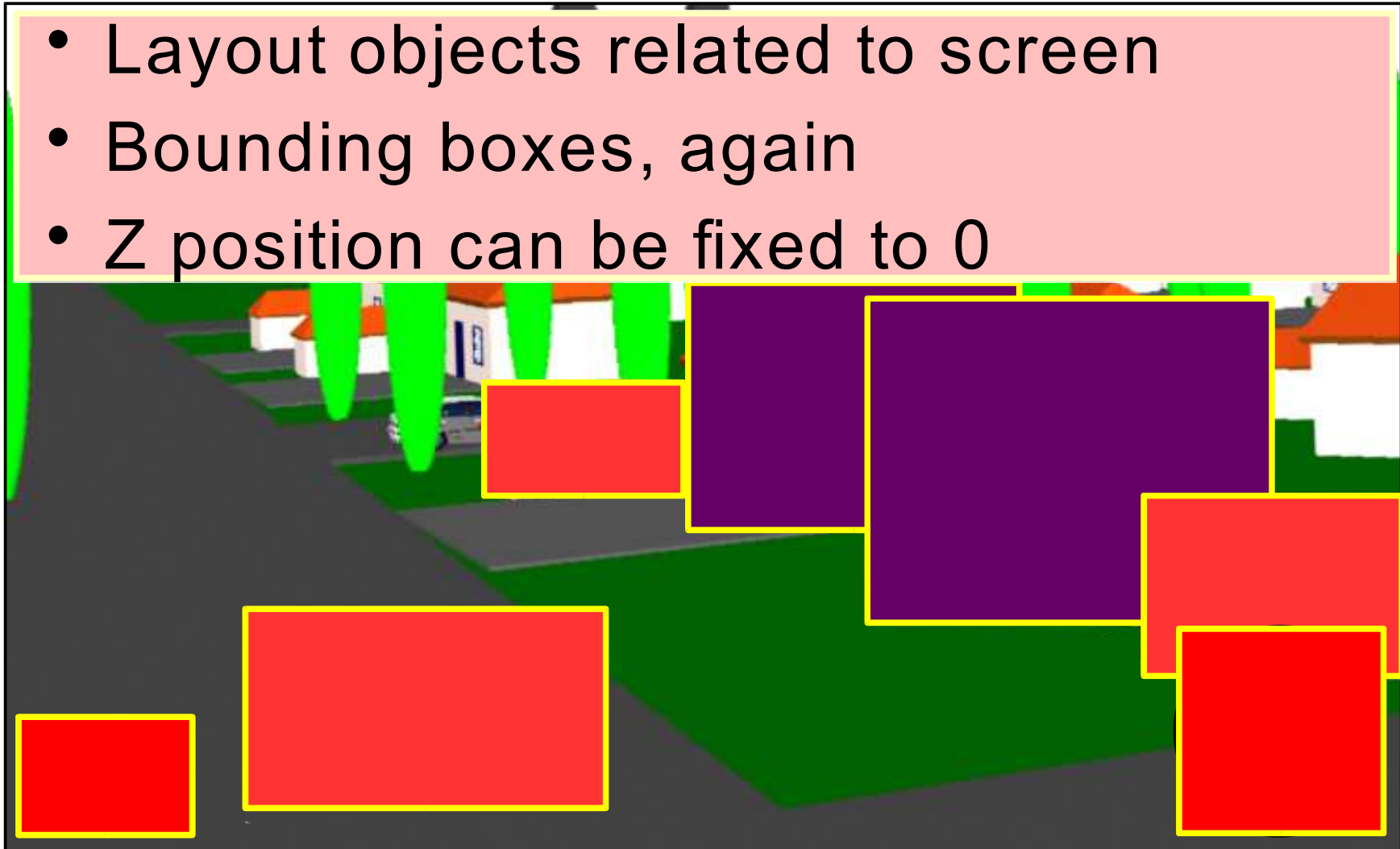
Common Layout Problems

- Layout objects related to 3D objects
- Approximation with bounding boxes
- Ordering in Z is relevant



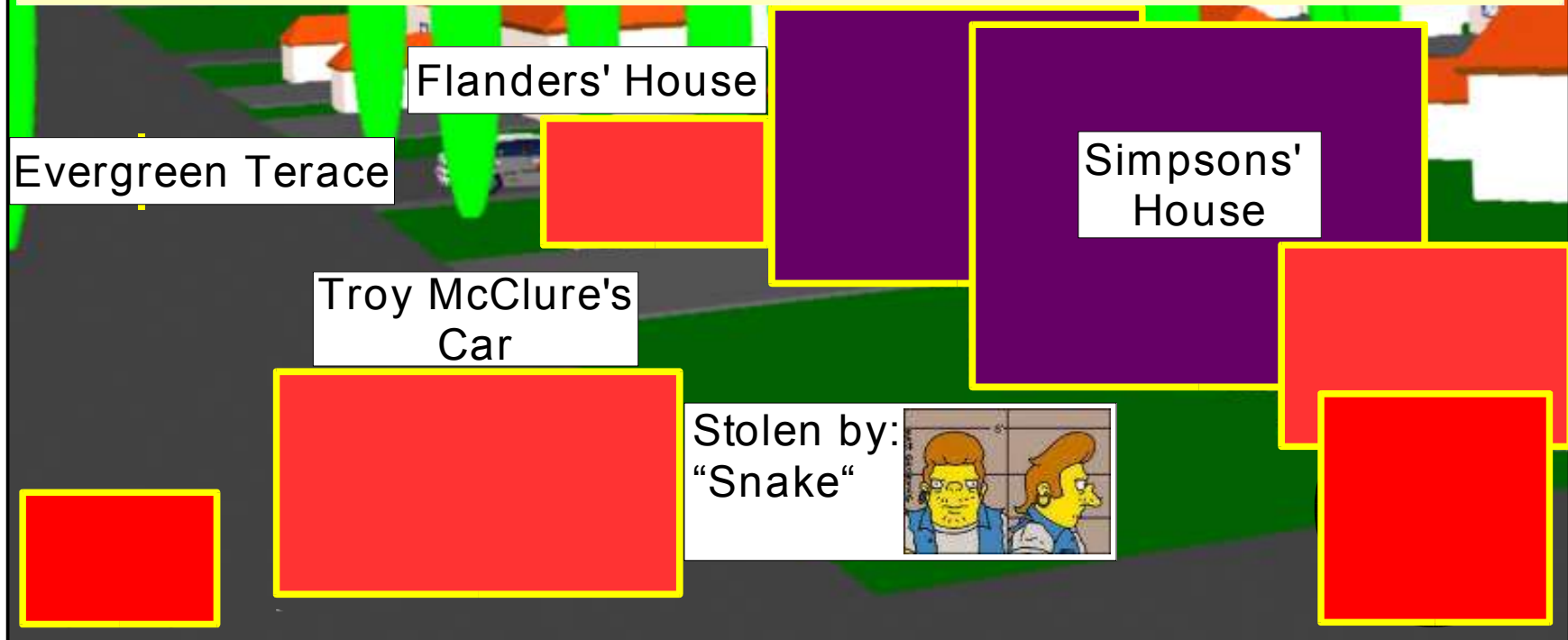
Common Layout Problems

- Layout objects related to screen
- Bounding boxes, again
- Z position can be fixed to 0



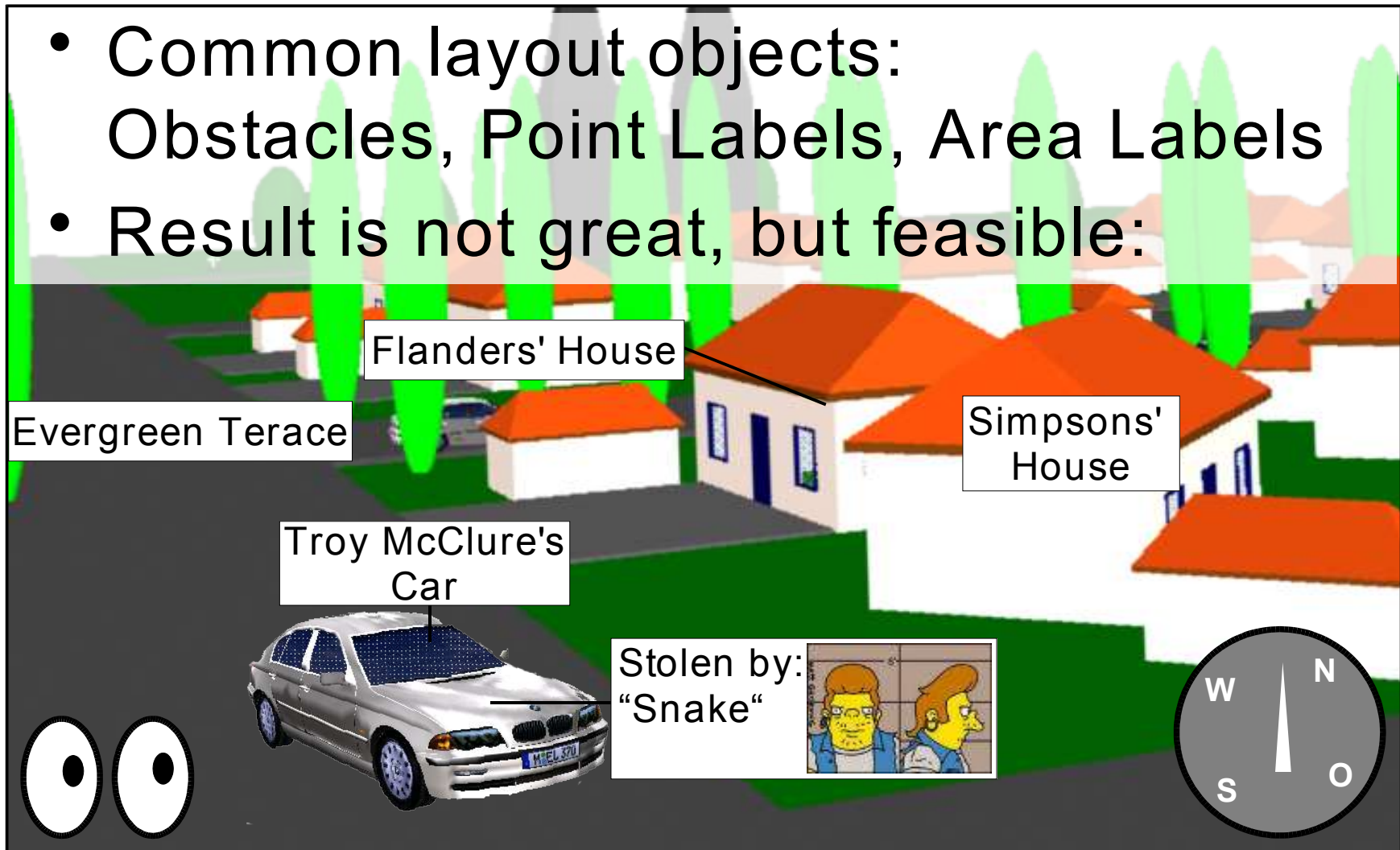
Common Layout Problems

- Labels have known size
- Appropriate positions are searched consecutively



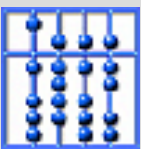
Common Layout Problems

- Common layout objects:
Obstacles, Point Labels, Area Labels
- Result is not great, but feasible:



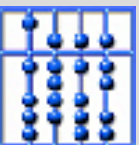
Possible Layout Constraints

- Local constraints on objects
 - Distance to target
 - Relative position to other objects (Inter-Object-Constraints)
 - Size (fixed or in a given range)
- Global settings of layout behavior
 - How to handle overlap of 3D objects
 - Minimize occlusion (how is that measured?)
 - Smoothness over time
 - What to do if layout fails?



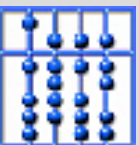
Contents

- Why Automatic Layout
- Applications & Sample Layout Problems
- ▶ **Requirements for Layout in DWARF**
- System Design & Object Design
- Layout Demos
- How to do it better? Who Will do it Better?



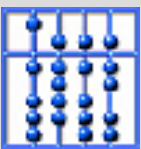
Pseudo Requirements

- Support the DWARF Viewer
 - Use the DWARF network infrastructure
 - Use C++
 - Use OpenInventor, COIN
- Use Columbia SpaceManager (2D layout)
 - Use Java
- Be useful for the CAR project



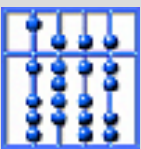
Functional Requirements

- Manage layout information for a 3D scene
 - Add, remove and modify layout objects
 - Specify relation of layout objects to 3D objects
 - Store and load layout information for a 3D scene
- Extract 2D layout problem from a 3D scene
- Calculate solution for a layout problem
- Display layout solution
 - in overlay
 - in 3D scene
- Change between different layout behaviors



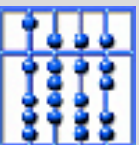
Non-Functional Requirements

- Realtime capability
 - Approximation by bounding box
 - No expensive calculations of the best layout
- Good quality of layout results
- Smoothness of layout results over time
- Easy authoring
- Good integration in DWARF applications
- Extensibility

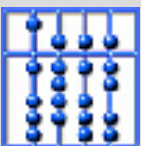
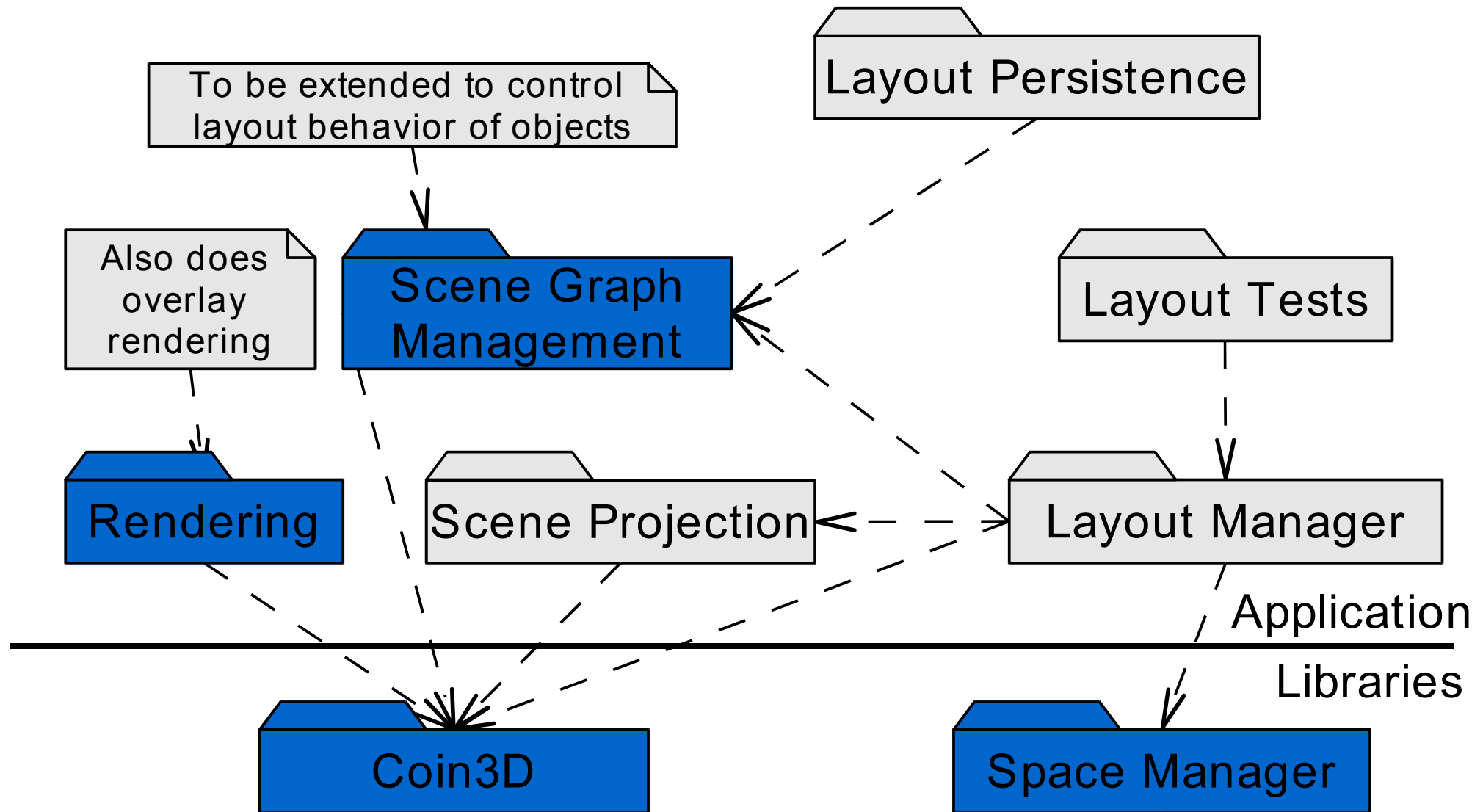


Contents

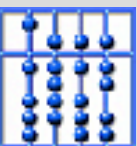
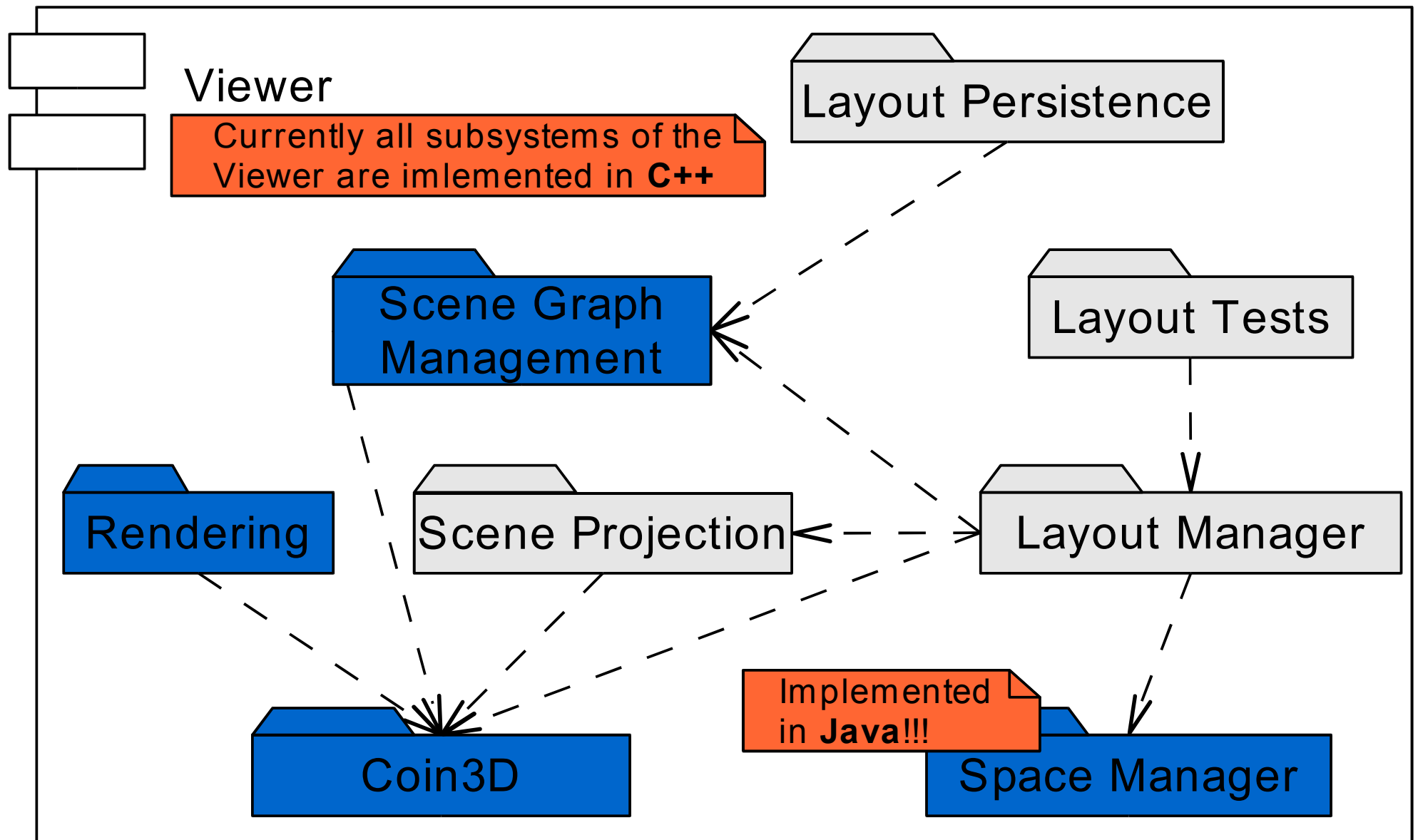
- Why Automatic Layout
- Applications & Sample Layout Problems
- Requirements for Layout in DWARF
- ▶ **System Design & Object Design**
- Layout Demos
- How to do it better? Who Will do it Better?



Subsystem Decomposition

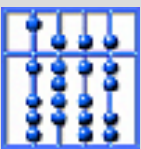
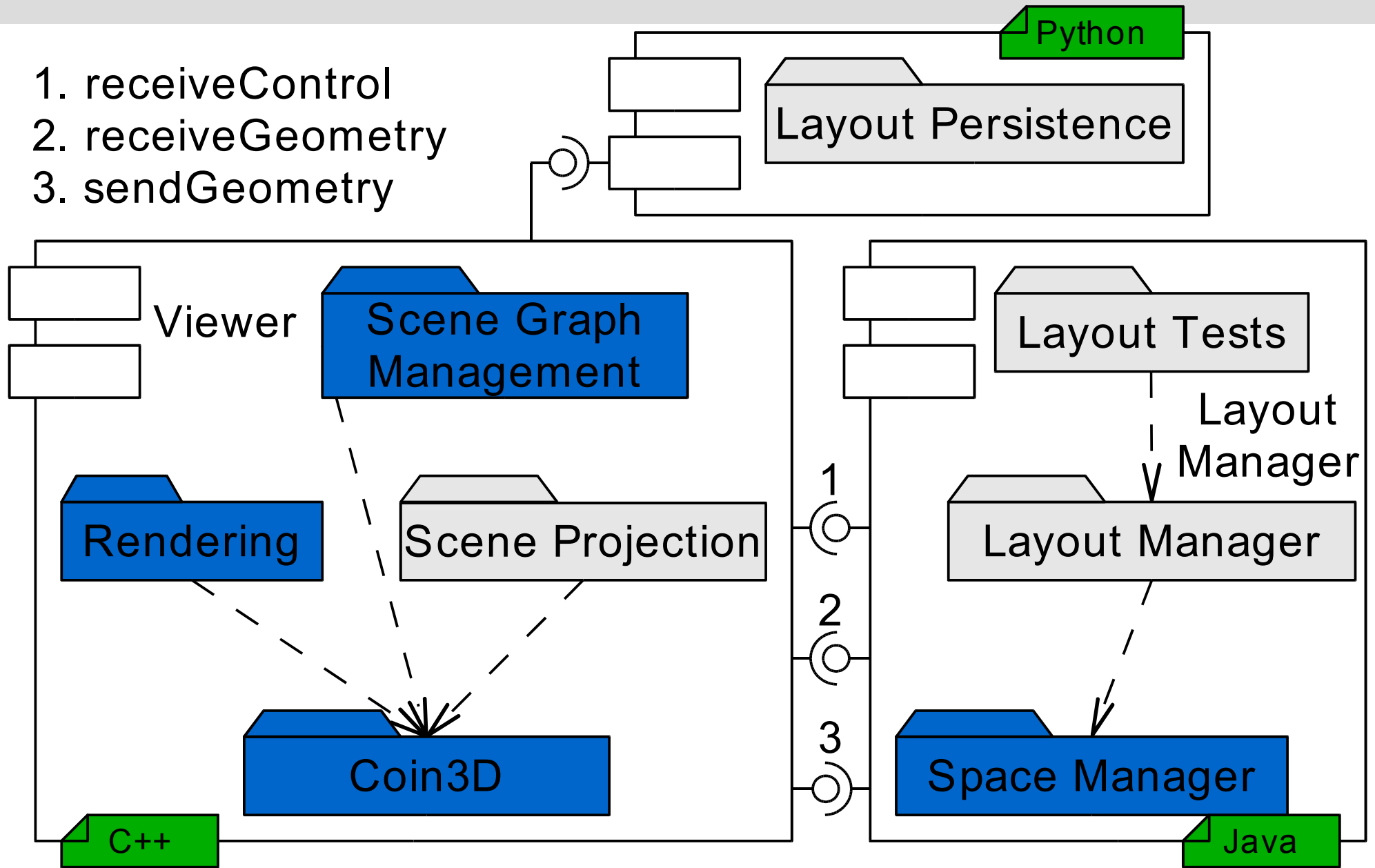


Monolithic Mapping to the DWARF Viewer

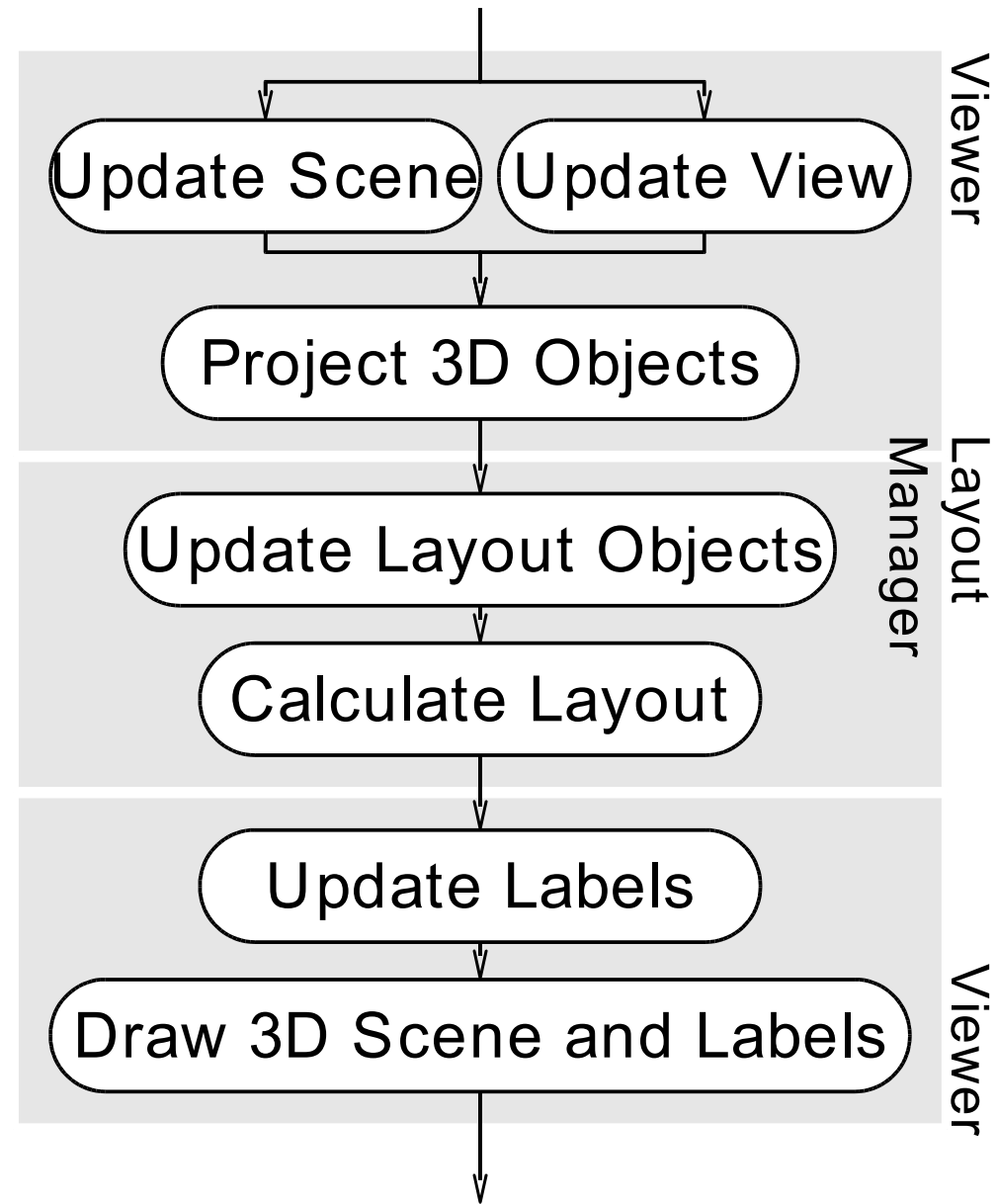
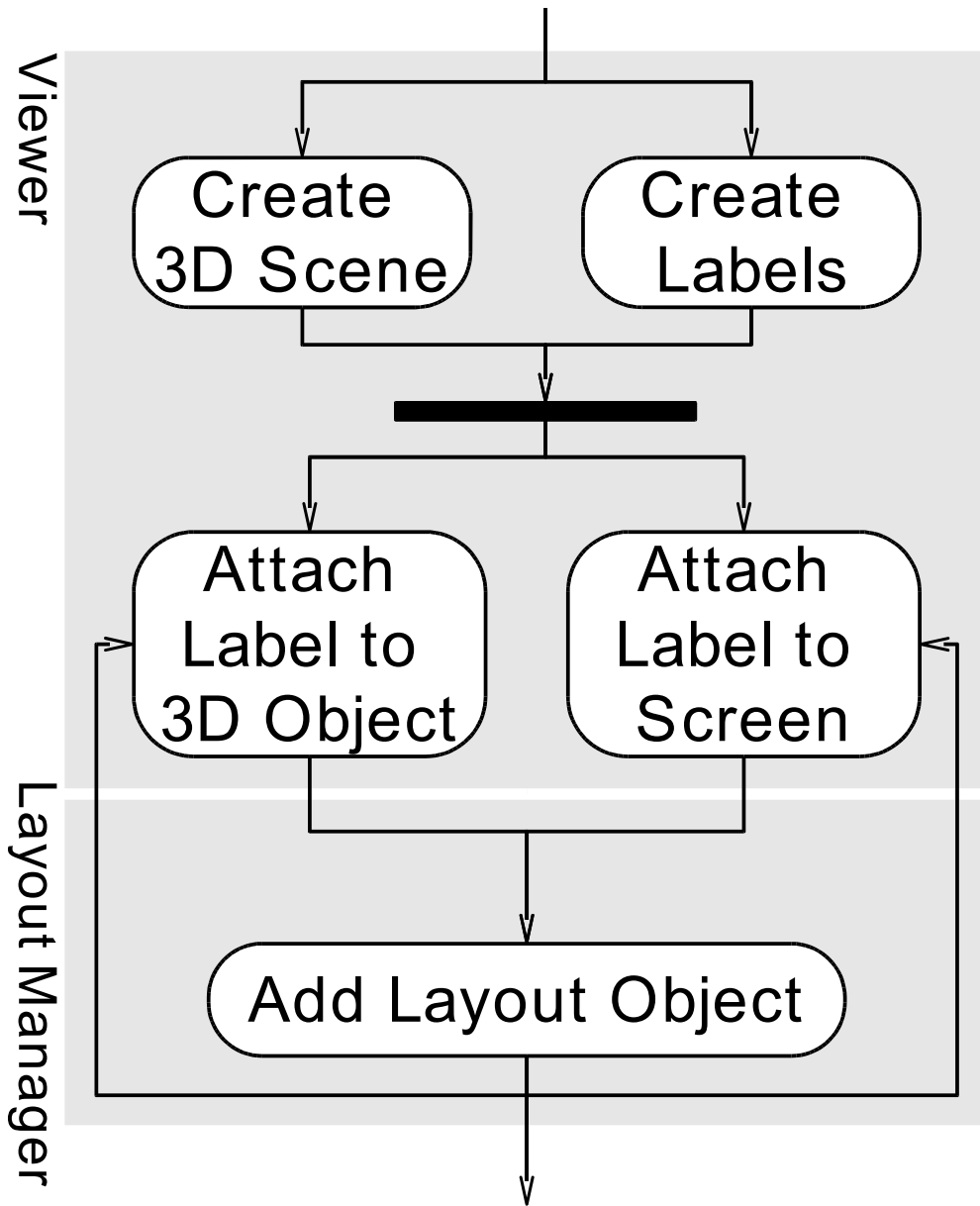


Distributed Mapping to DWARF Services

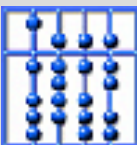
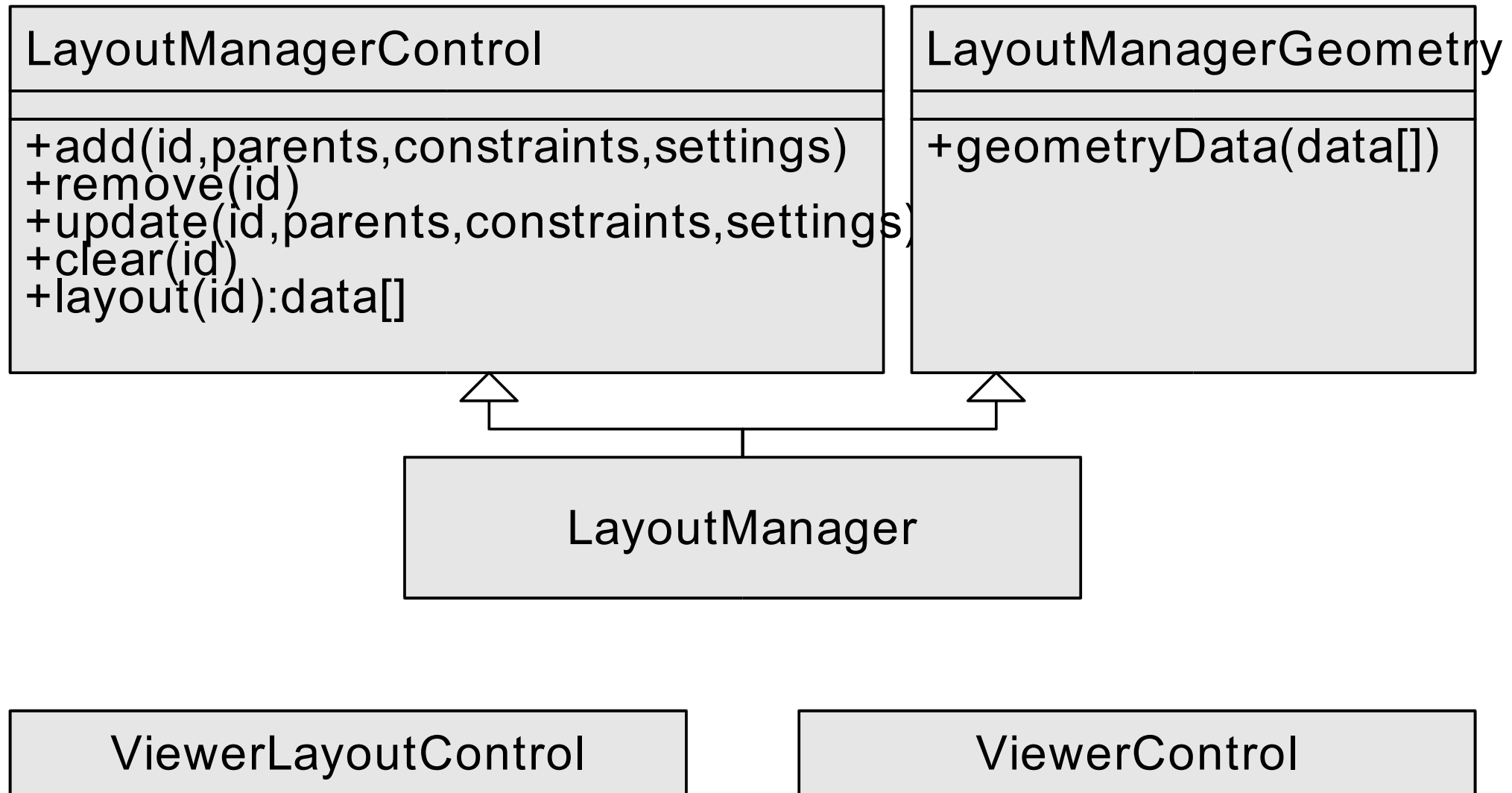
1. receiveControl
2. receiveGeometry
3. sendGeometry



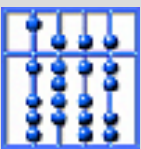
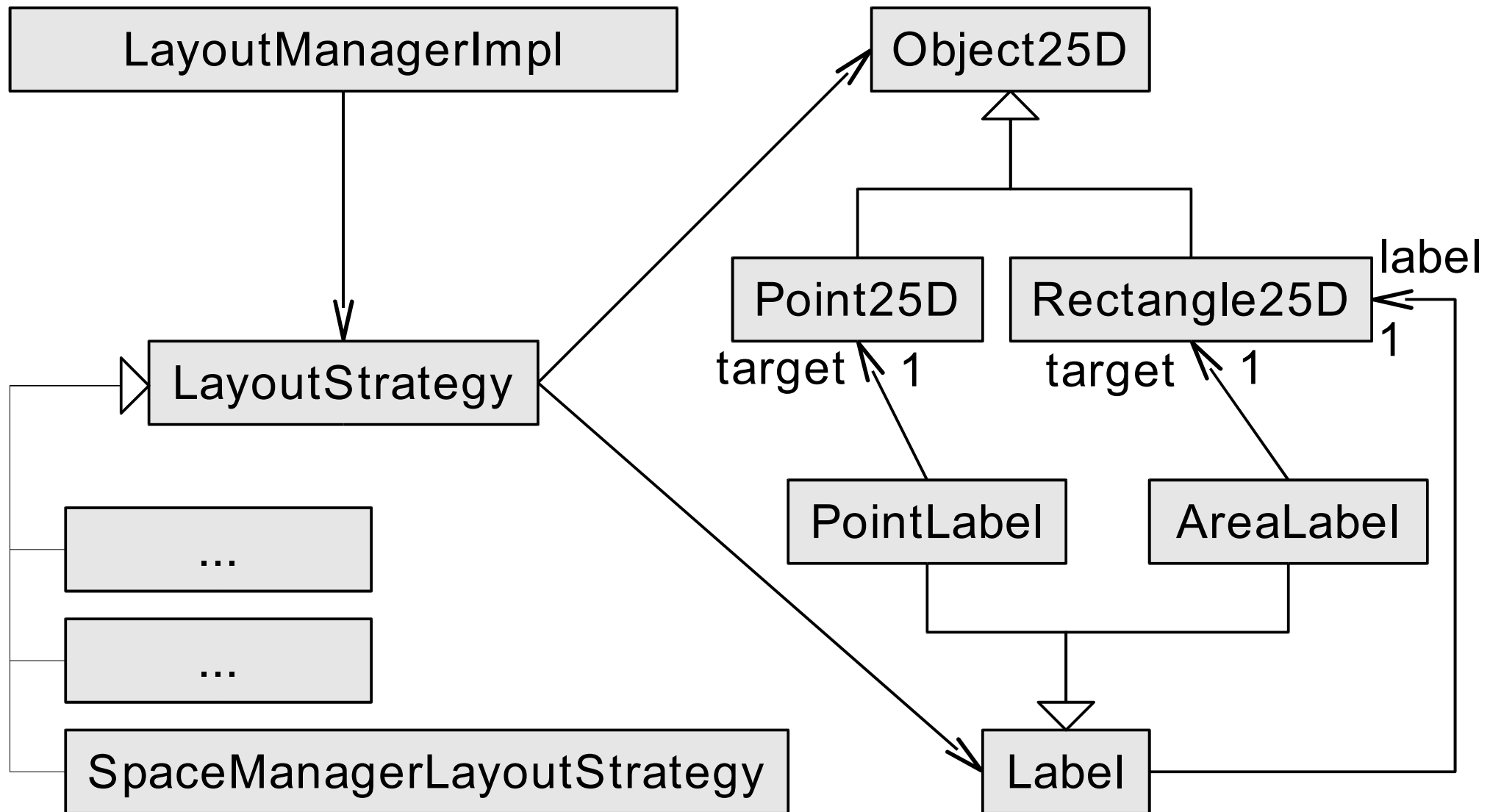
Flow of Control



Interfaces for Layout-Related Services

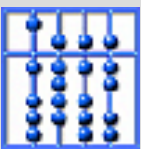


Classes of the LayoutManager Service



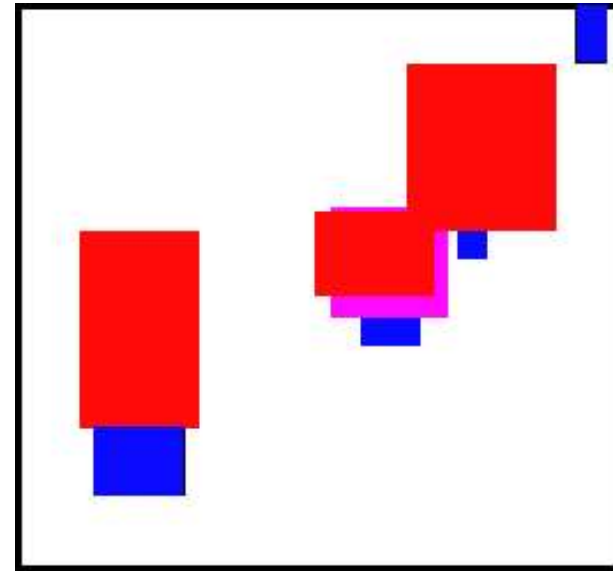
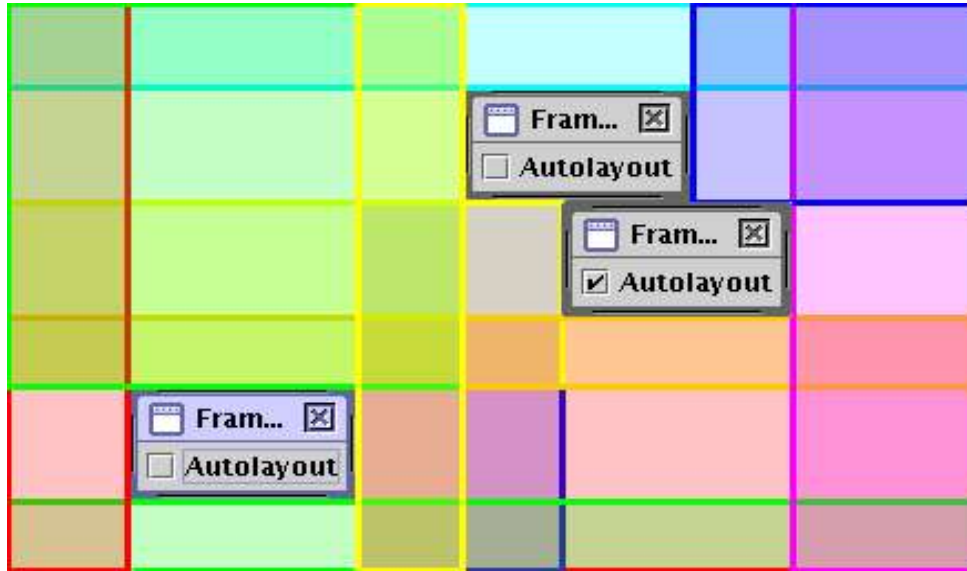
Contents

- Why Automatic Layout
- Applications & Sample Layout Problems
- Requirements for Layout in DWARF
- System Design & Object Design
- ▶ **Layout Demos**
- How to do it better? Who Will do it Better?



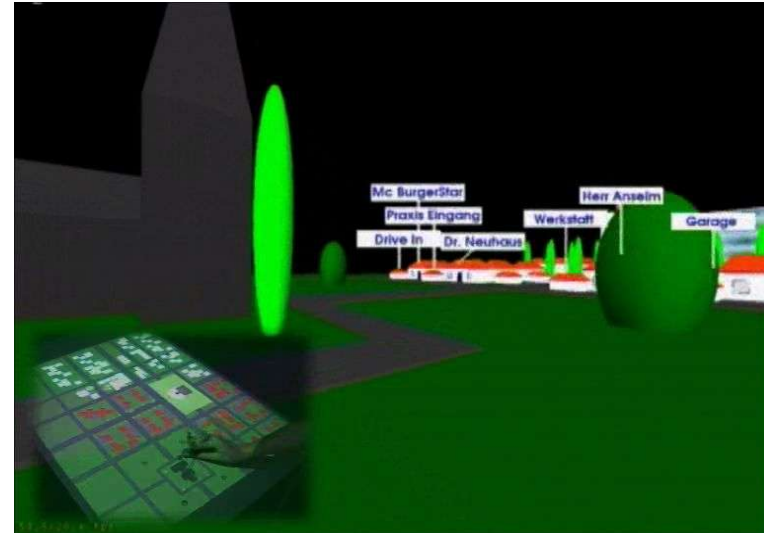
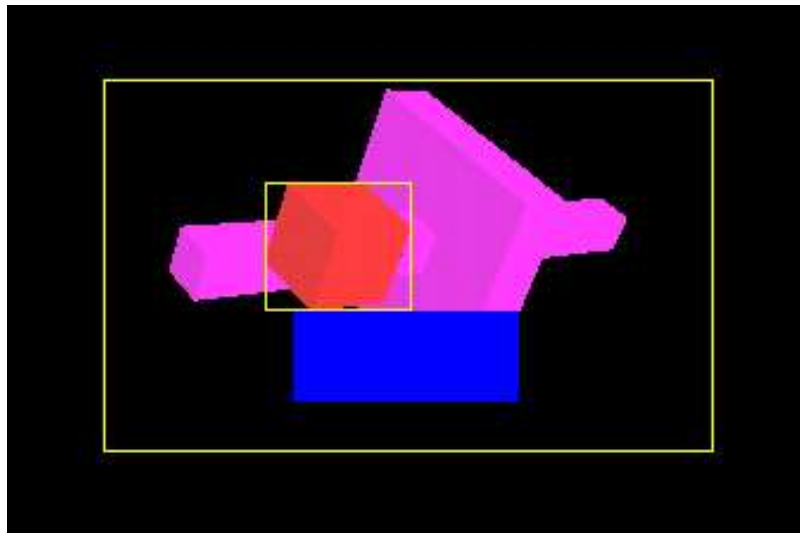
Layout Demos

Frame Test



2D Test

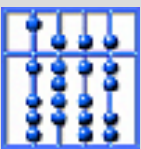
3D Test



CAR Demo

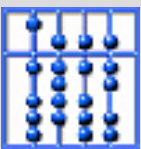
Contents

- Why Automatic Layout
- Applications & Sample Layout Problems
- Requirements for Layout in DWARF
- System Design & Object Design
- Layout Demos
- ▶ **How to do it better? Who Will do it Better?**



Yet to be Implemented

- Free hierarchical model of layout objects
- Support for all constraints specified
- More layout strategies
- Smoothnes over time
- More precise bounding boxes
- Improved performance and benchmarks
- Persistance format and authoring support

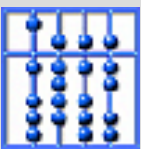


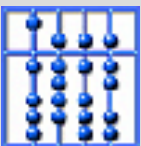
Further Options Related to Layout

- Productive use and usability studies
- More interaction between user and layout objects
- Layout of user interface widgets
- Layout of 3D objects
- Multiscreen layout

coming soon ...

Diplomarbeit of Fabian Sturm



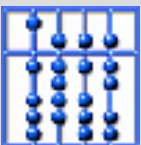


New Classes for the Viewer Service

DwarfViewer

ProjectionAction

LayoutKit



2.5D Geometry

