

Mapmaking

(with code and world)

Simon Denier - RMoD
SATTOSE'09

Porting Moose

Moose Core and Famix 3 ready

Now what?

Porting Moose

Now: Hismo, Hapax, SCG Algorithms...

Project by project

Package by package

Problem

Package organization of a project?

Where do I start?

How do I relate stuff within packages?

Software Maps

I want an overview of packages, classes,
and their relationships,

to understand the overall structure
and find master elements

I want maps

Behind the word?

Map

(from wikipedia)

a symbolic depiction

highlighting relationships

between elements

Orienteering Map



Symbols depict particular terrain elements

Elements can be related to each other spatially

Contours connect the whole thing

Maps (in real world)

Maps are goal-driven:
road navigation, water system

Goal dictates guidelines to select
interesting elements:
obstacle, natural ways, rivers, big relief

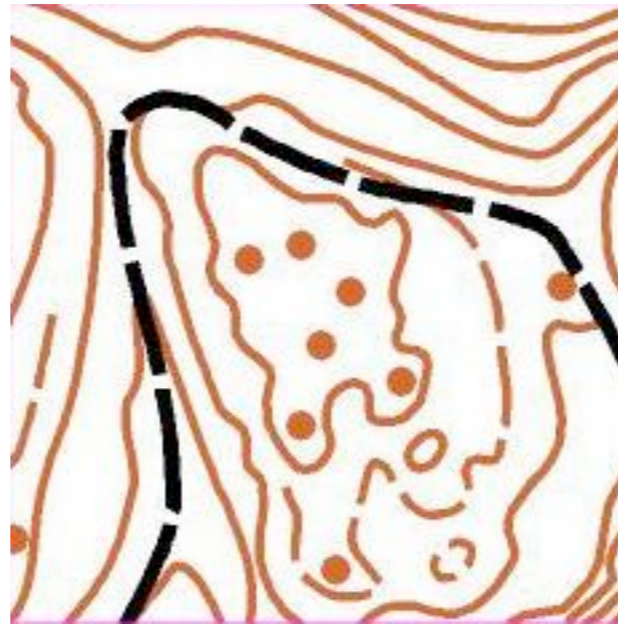
Mapmaking

1. Get the overall structure right:
road network, big contours
2. Refine within each delimited area
3. Proceed by layers of same type
elements

Mapmaking Process



Digital Terrain
Model



Photogrammetry



Orienteering

Maps for packages

1. System Map by packages
2. Inner Package Map (package blueprint)
3. 2-Packages Collaboration Map

System map

Goal = package organization

Guidelines =
dependencies between packages,
complexity,
symbolic information

Package Organization

Problem = layout graph of dependencies

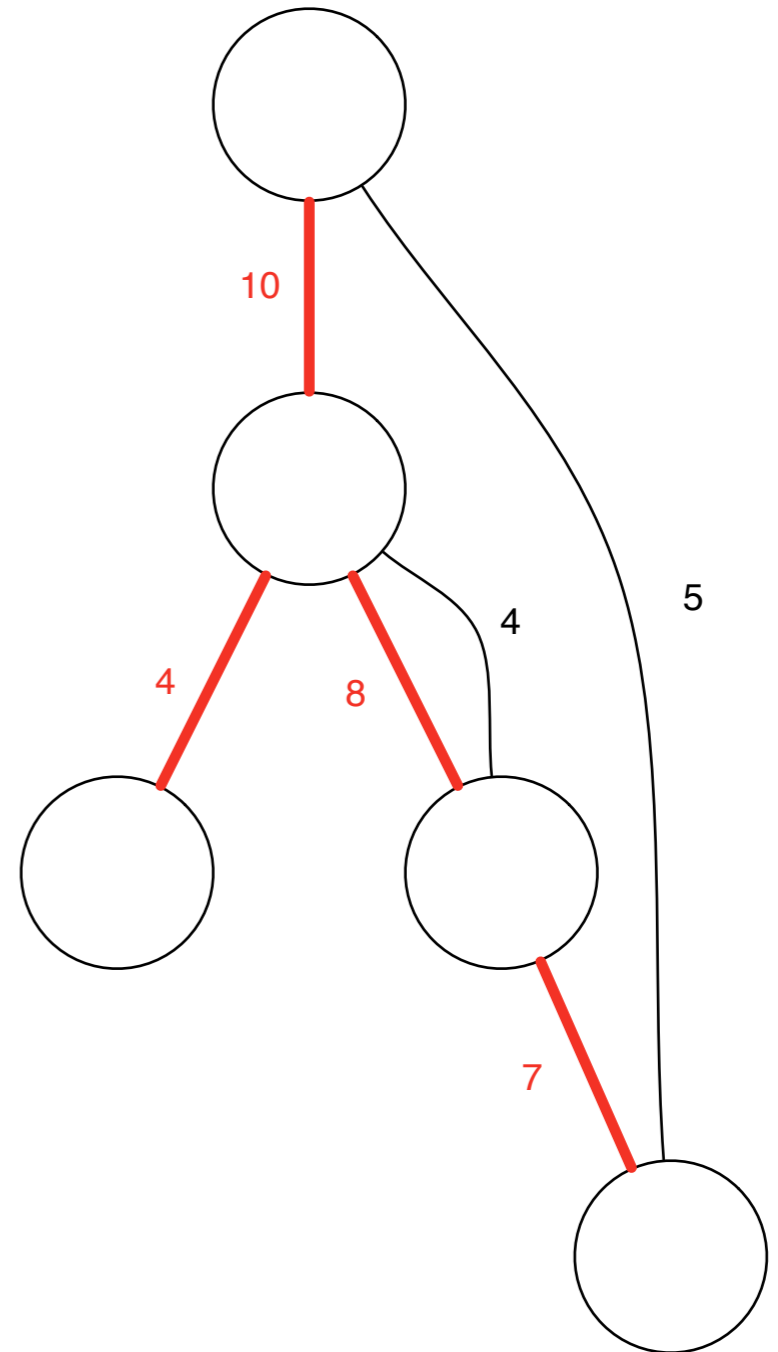
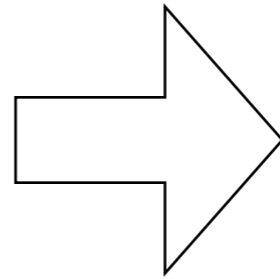
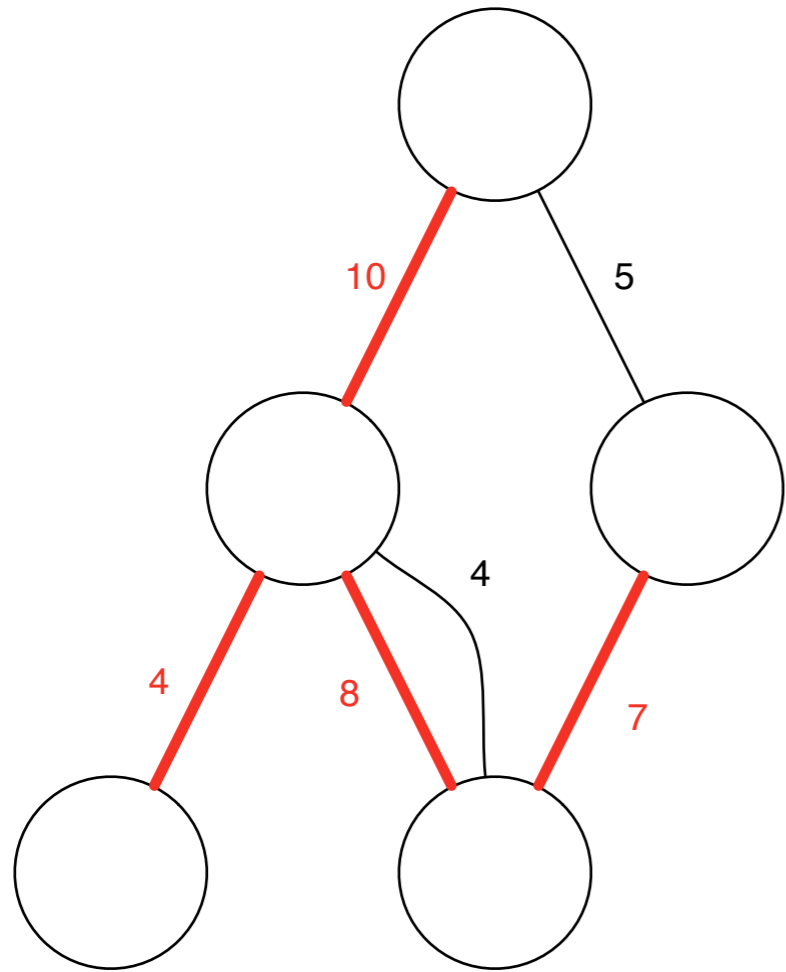
possibly complex

possibly with cycles

Layout with Cycles

1. Compute spanning tree
2. Layout spanning tree
3. Draw removed edges

Spanning Tree



Other Ideas

- Minimize edge removal to break cycles
- Strongly Connected Component Analysis
- Module Analysis

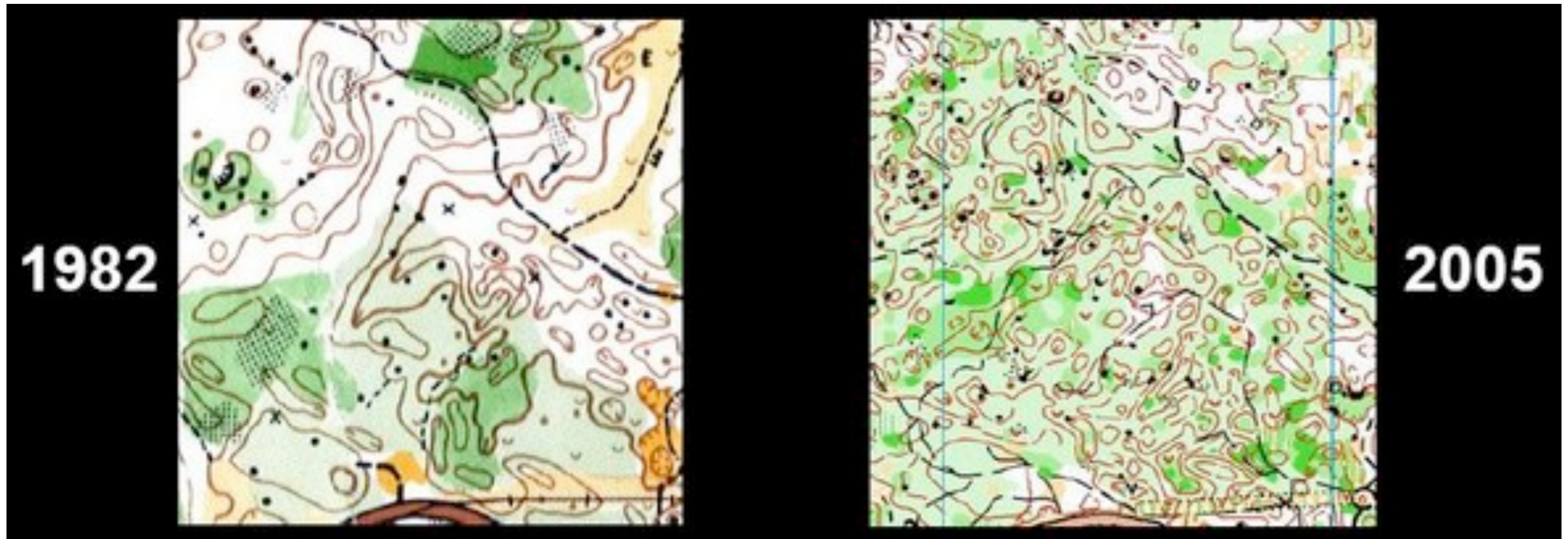
Map Summary

- Maps are designed with a goal and guidelines
- Maps display a selection of elements organized through relationships

Good maps
come from
good base structure

Common Pitfall 1

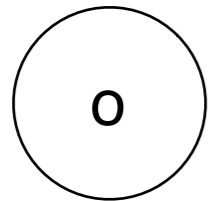
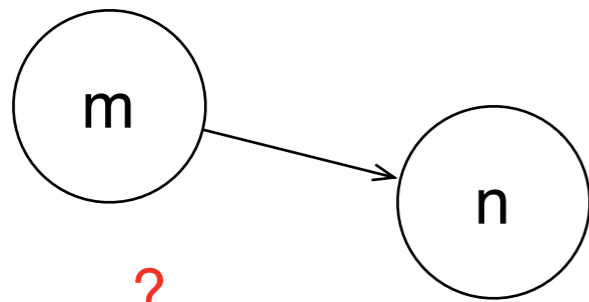
Over simplification



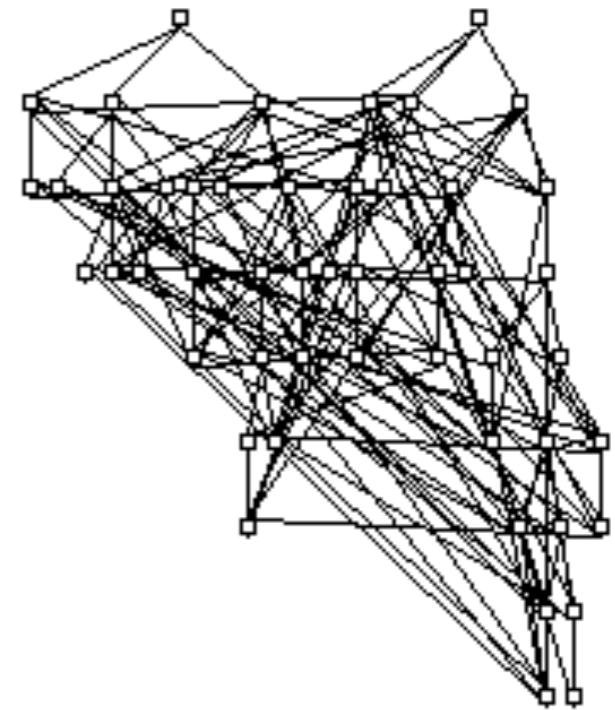
Too much details

Common Pitfall 2

Irregular selection



```
m() {  
  n();  
  o();  
}
```



No context filter

Common Pitfall 3

Failed basemap

will produce

bad map