

# Please don't make me draw!

Lesson learned during the development of a software to support early analysis of object-oriented systems.

**Andrea Valente**

Department of Electronic Systems, Automation & Control,  
Aalborg University - Esbjerg Institute of Technology  
Esbjerg, Denmark  
av@aaue.dk

**Emanuela Marchetti**

Mads Clausen Institute of Product Innovation,  
University of Southern Denmark  
Sønderborg, Denmark  
e.manum@gmail.com

# Rich pictures

Rich pictures represent knowledge about a domain  
Knowledge acquisition during OOA  
Help in construction of the system's early prototypes

A rich picture provides "a broad, high-grained view of the problem situation":

- **structures**
- **processes**
- **concerns**

Usually:

- rich pictures are created with low-tech support (whiteboards, pen and paper)
- students use non-specific diagram-drawing tools
- need for acquiring knowledge incrementally

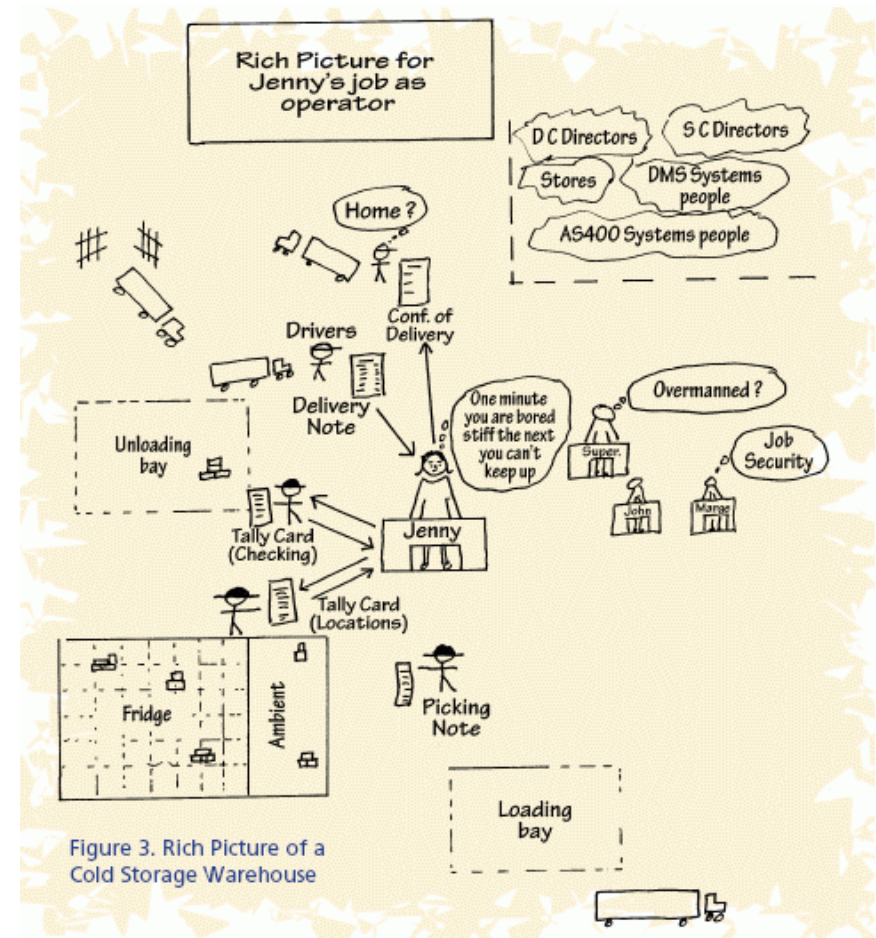


Figure 3. Rich Picture of a Cold Storage Warehouse

FROM: A. Monk and S. Howard, Methods & tools: the rich picture: a tool for reasoning about work context.

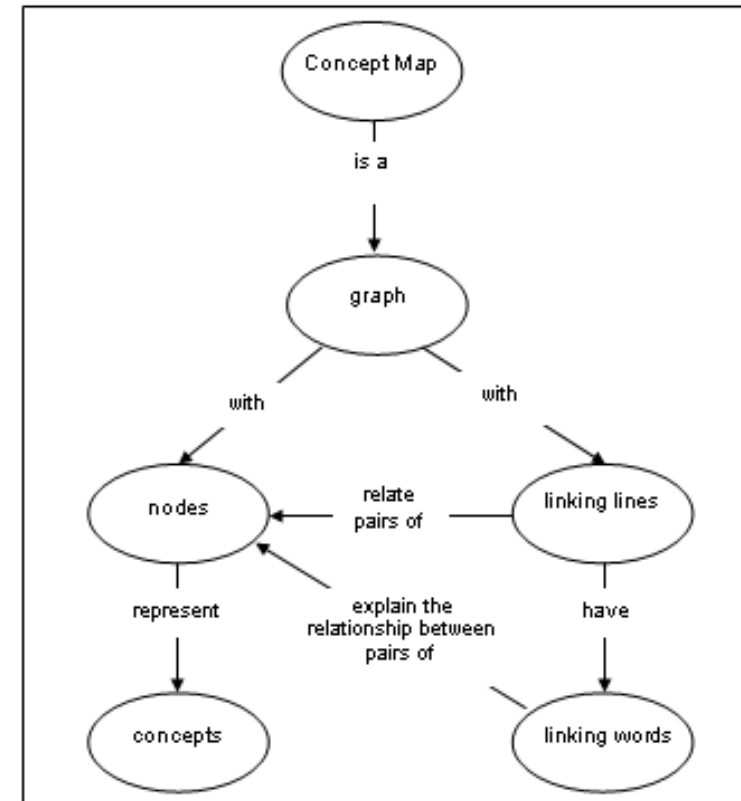
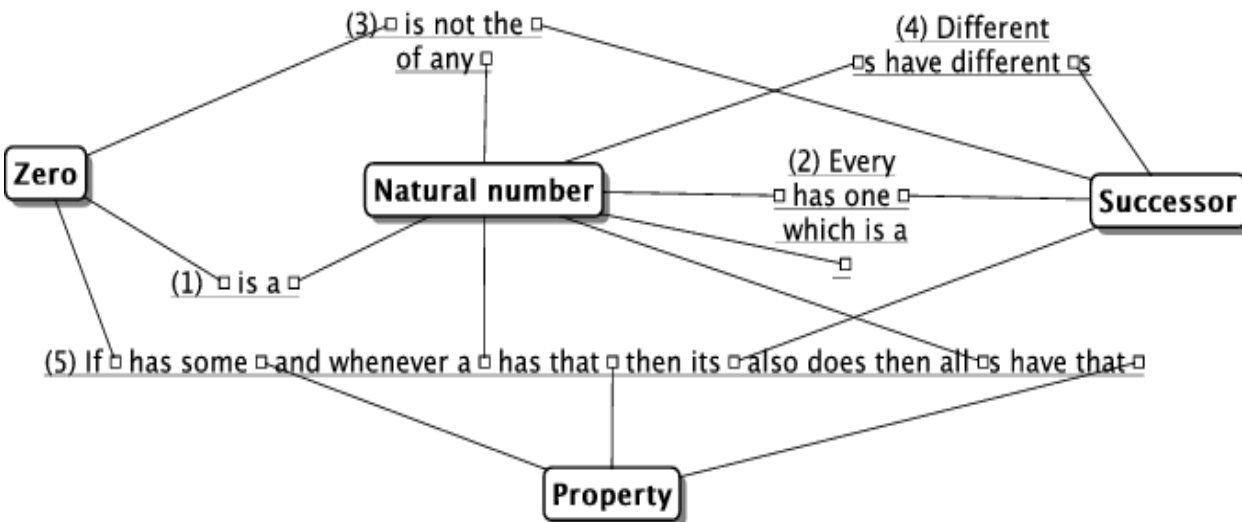
# Concept Maps and Text Graphs

Rich pictures are related to Concept Maps (Novak)

How can **concept maps** be made more *visual*?

What happens when text is replaced with pictures, in a **text graphs**?

How to represent: **structures** (*static aspects*) and **events** (*dynamic aspects*)



- A Text Graph -

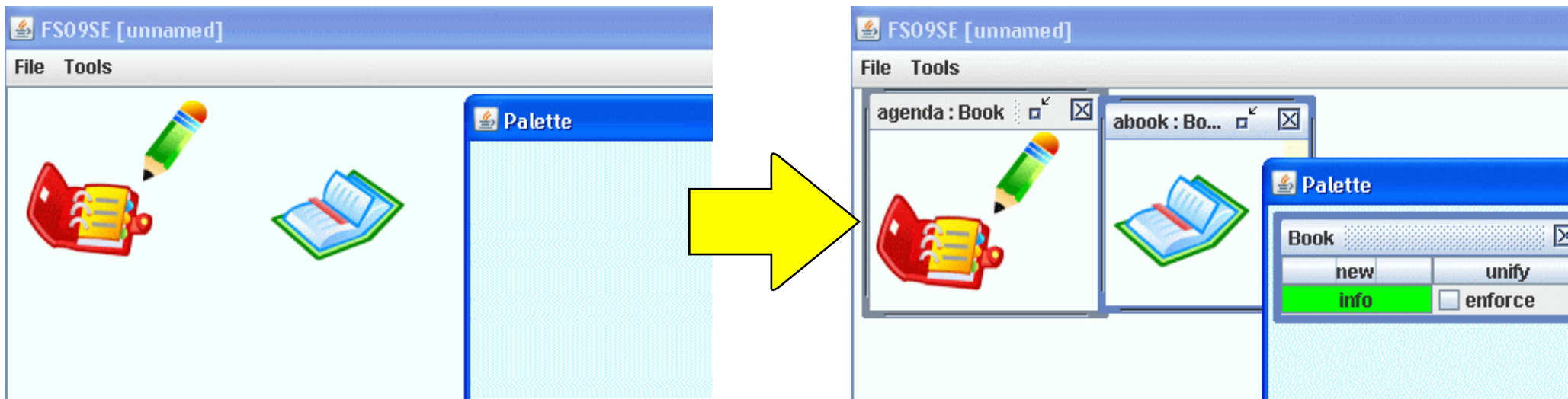
- A Concept Map -

# The (prototype) tool

The new tool is called **Free Sketch for Software Engineering (FSSE)**

- minimalistic prototype
- supports rich pictures construction
- knowledge acquisition during the OOA phase
  - incremental definition of both instances and types
- multiple hierarchy
- (provides code generation)

FrameName : List of Tags



**DEMO**

# The Test

## Problem:

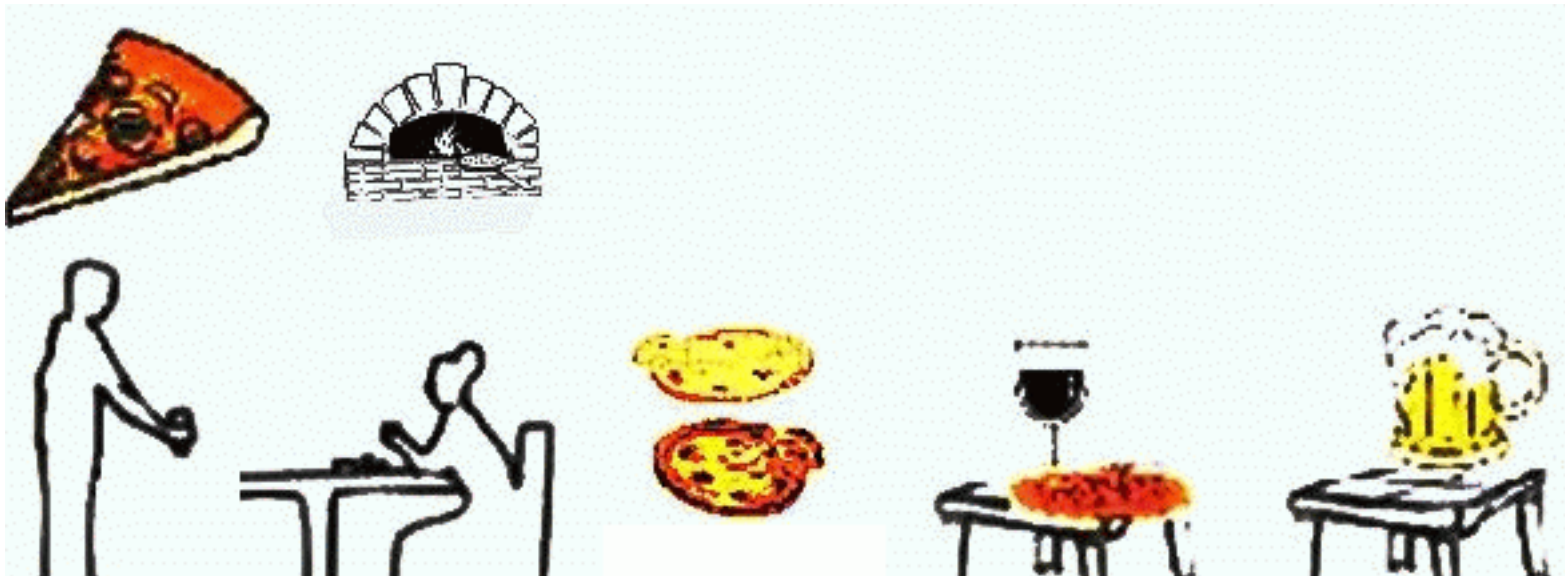
transposing the manual rich picture practice into a software tool was difficult

## Solution:

follow a user-centered approach -> involve students (EARLY) in a usability test

## Aim of test:

- evaluate how users may perceive a tool for rich pictures
- easy to use
- if it adequately supports work-flow (for individuals and groups)
- *collect suggestions for new features (to expand minimalistic prototype)*



Icons provided for the test

# (testing methods and ... ) Results

## Findings:

- don't make me draw!
  - especially on paper
  - provide pre-made icons
  - support for arrows
  - keep focus on knowledge modelling
- works like "think aloud" -> nice for team work
- software tool is good -> given iterative nature of OOA, paper requires (re)drawing
- tags VS types (?)
- tags VS frame names (?)
- *emergent usage pattern: prototypes and prototype clustering* -> EVENTS!

## *Standard qualitative methodologies from the Design field*

### Questions:

- method of **situated interviews** (interview users in their context of practice)
- start with open questions, then gradually focus on details of users' statements. Ask for examples.
- **Interviews** VS questionnaires:
  - give more possibility to express freely
  - show them that we cared for their contribution -> more direct feedback
- leverage on **reflection in action** (a process of critical thinking while performing a skilled practice)

Quick look at new  
features and GUI

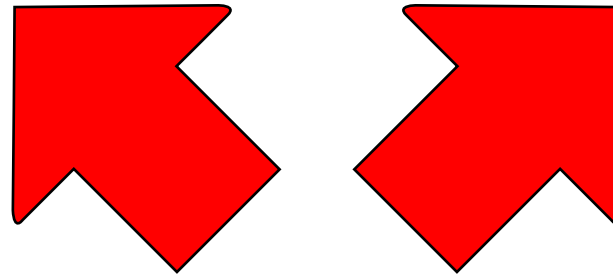
# EVENTS = arrows between prototypes

## Main window

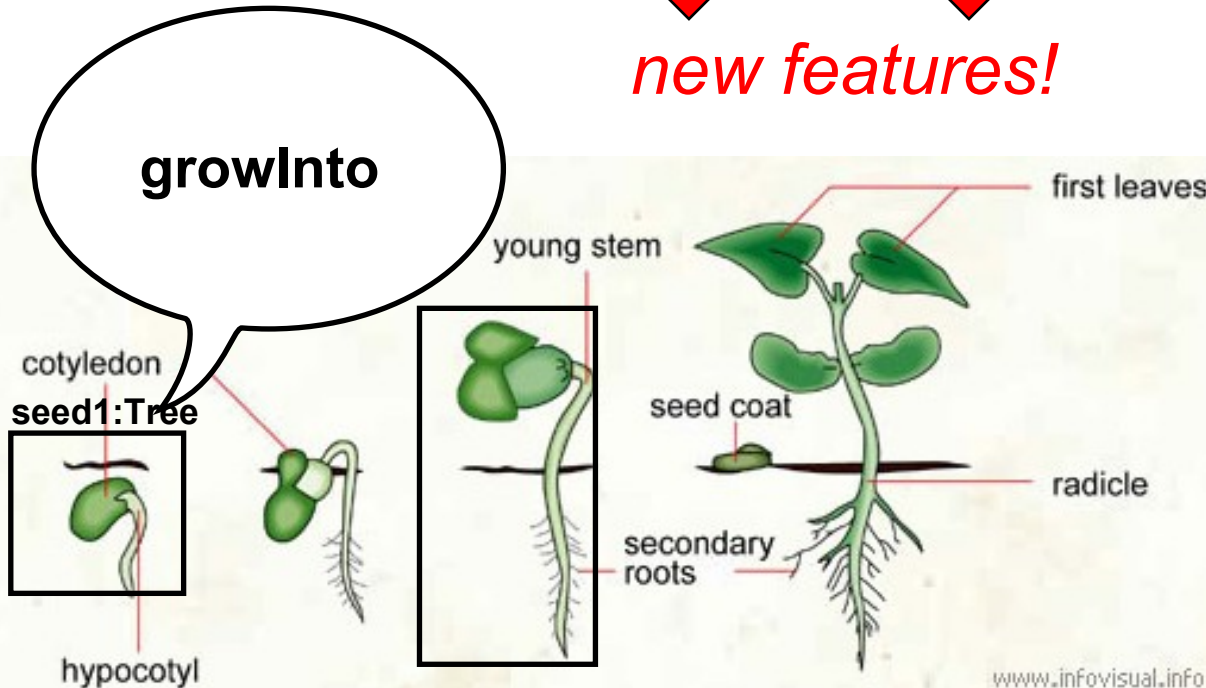
- Frames
- *Arrows between frames*

## Palette window

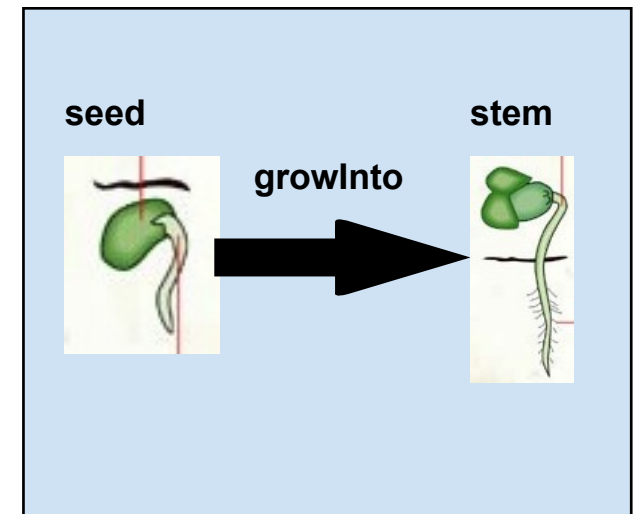
- Tags
  - *Prototypes = typical frames*
  - *Events = arrows between prototypes*



*new features!*



:Tree



# Conclusion / future work

- No tools to support rich pictures creation, for object-oriented analysis
- How to validate and complete the initial prototype of the tool?
  - involve users (i.e. students) -> usability test
  - small group is not a problem -> we got meaningful feedback for new iteration
- Emergent patterns of use suggested new/better features: e.g. events
- Discovery: programmers' values/skills do not include drawing!

## Long term plan:

- improve FSSE and test it further
- propose FSSE as the main tool for the OOA&D course

## Future work:

- Free external painter VS **internal painting capabilities**: provide a more uniform environment for drawing rich pictures
- Events will be used to **validate** one's understanding of a system
- Plug-in mechanism: users will define their own mapping from rich pictures to external formats/code
- *multiple (remote) users -> CSCW*