Inf-GraphDraw: Automatic Graph Drawing

Lecture 01: Modeling Pragmatics

Reinhard von Hanxleden
rvh@informatik.uni-kiel.de

We:

Prof. Reinhard von Hanxleden
rvh@informatik.uni-kiel.de, R1117

Ulf Rüegg
uru@informatik.uni-kiel.de, R1111

Christoph Daniel Schulze
cds@informatik.uni-kiel.de, R1112

Carsten Sprung
csp@informatik.uni-kiel.de

You?

Class Materials

- Roberto Tamassia, Editor
  Handbook of Graph Drawing and Visualization, CRC Press, 2013
  Chapters 1, 2, 5, 12, 13, 15
  On-line: https://cs.brown.edu/~rt/gdhandbook/

- Ioannis Tollis, Peter Eades, Giuseppe Di Battista, Roberto Tamassia
  Graph Drawing: Algorithms for the Visualization of Graphs
  Prentice Hall, 1998

- Plus original papers
  Most (if not all) should be on-line, at least within CAU

- https://ilearn.ps.informatik.uni-kiel.de/lecturer/courses/105

Your Grade

Your grade is given by final (probably written) exam + homework assignments

- Tentative exam date: Wed 20 July 2016, 10:00 – 12:00
- Need at least 50% in final exam to pass
- Grade = max (final exam, 85% final exam + 15% homeworks)
- In borderline cases, also consider participation in class

Allowed to take final exam if:

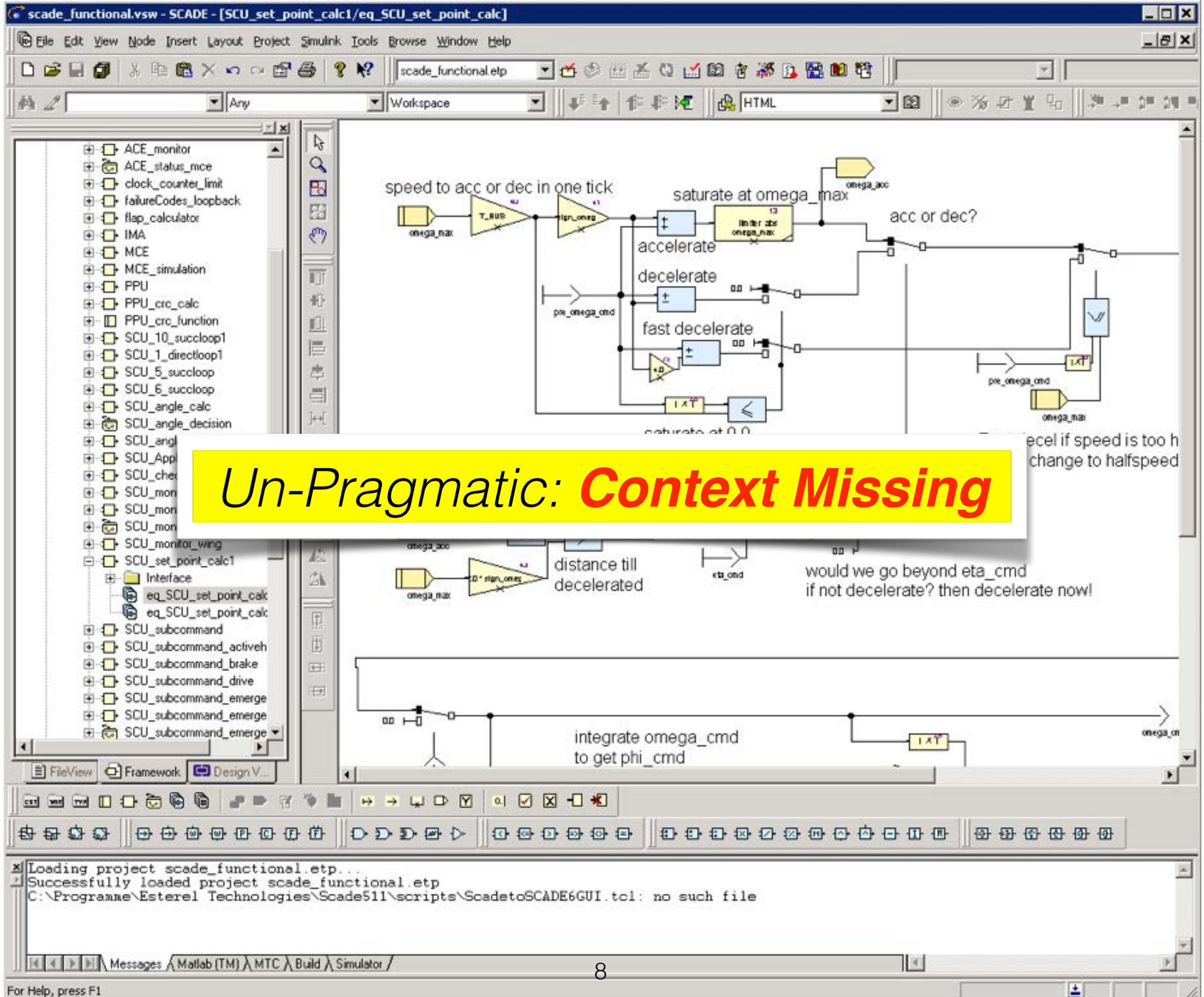
- Received at least 50% of homework assignment points

Outline (Tentative)

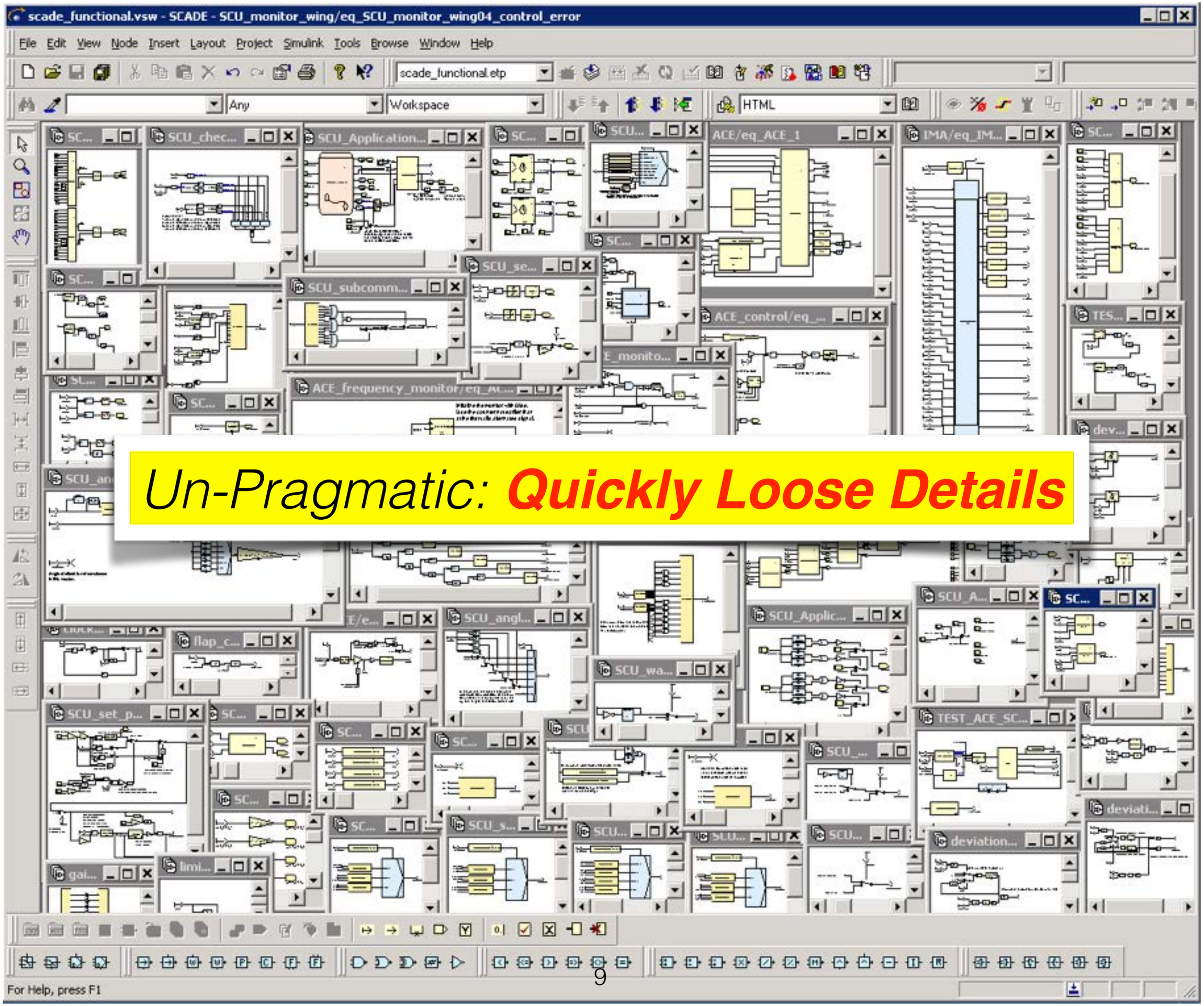
- Modeling Pragmatics, Terminology, Aesthetics
- Tooling, usage of ELK and KIELER
- Force-based drawing
- Hierarchical graph drawing
- Drawing trees
- Planarization-based graph drawing
- Labeling approaches

Modeling Pragmatics

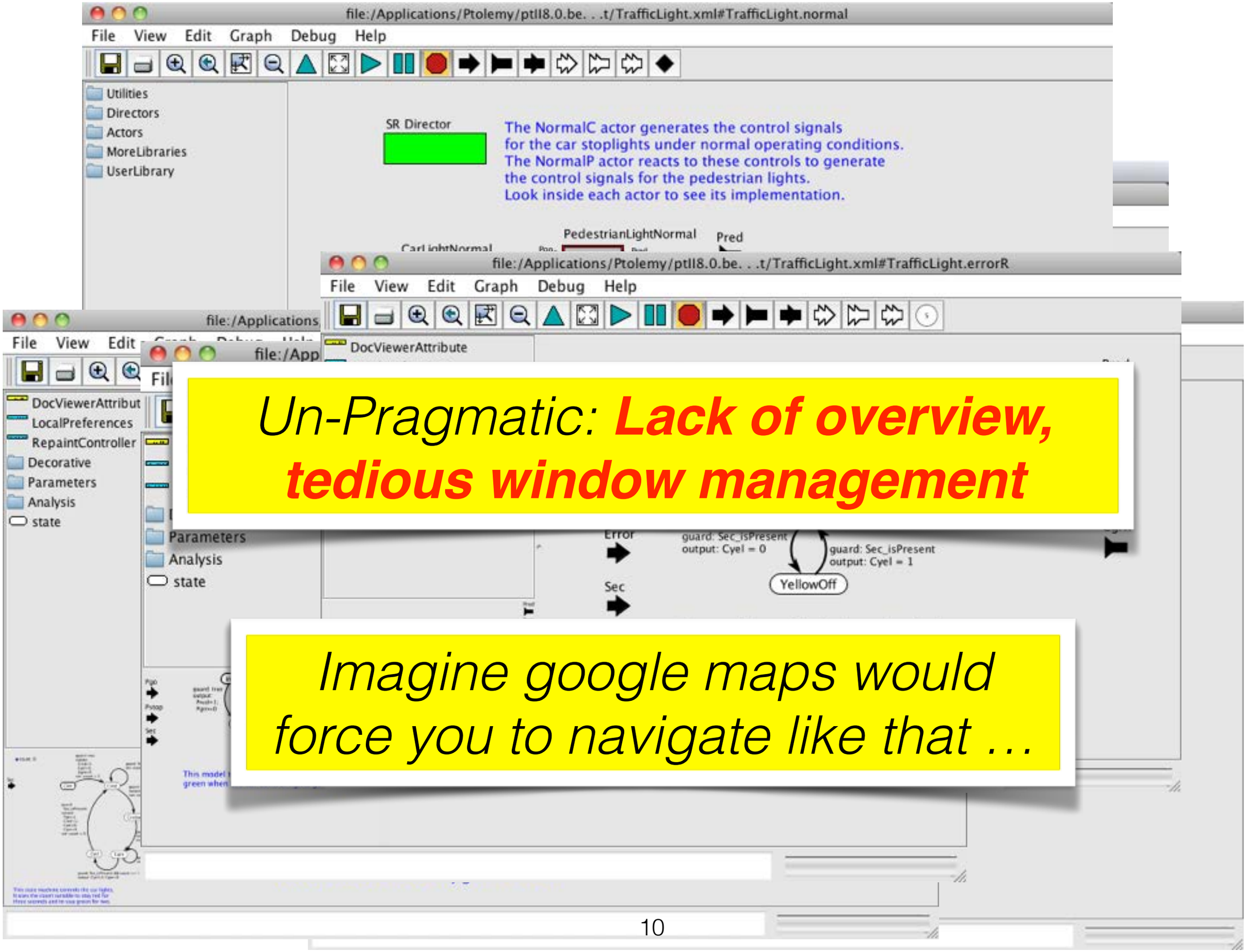
- **Goal:** Better use of 1) designer's time, 2) screen real estate
- **State of Practice:**
  1. Modeler draws view of a model
     (manual place & route – tedious!)
  2. Tool infers model
- **Our approach:**
  1. Modeler constructs model
  2. Tool continuously updates view of a model,
     with *automatic layout* and *filtering*



Un-Pragmatic: **Context Missing**

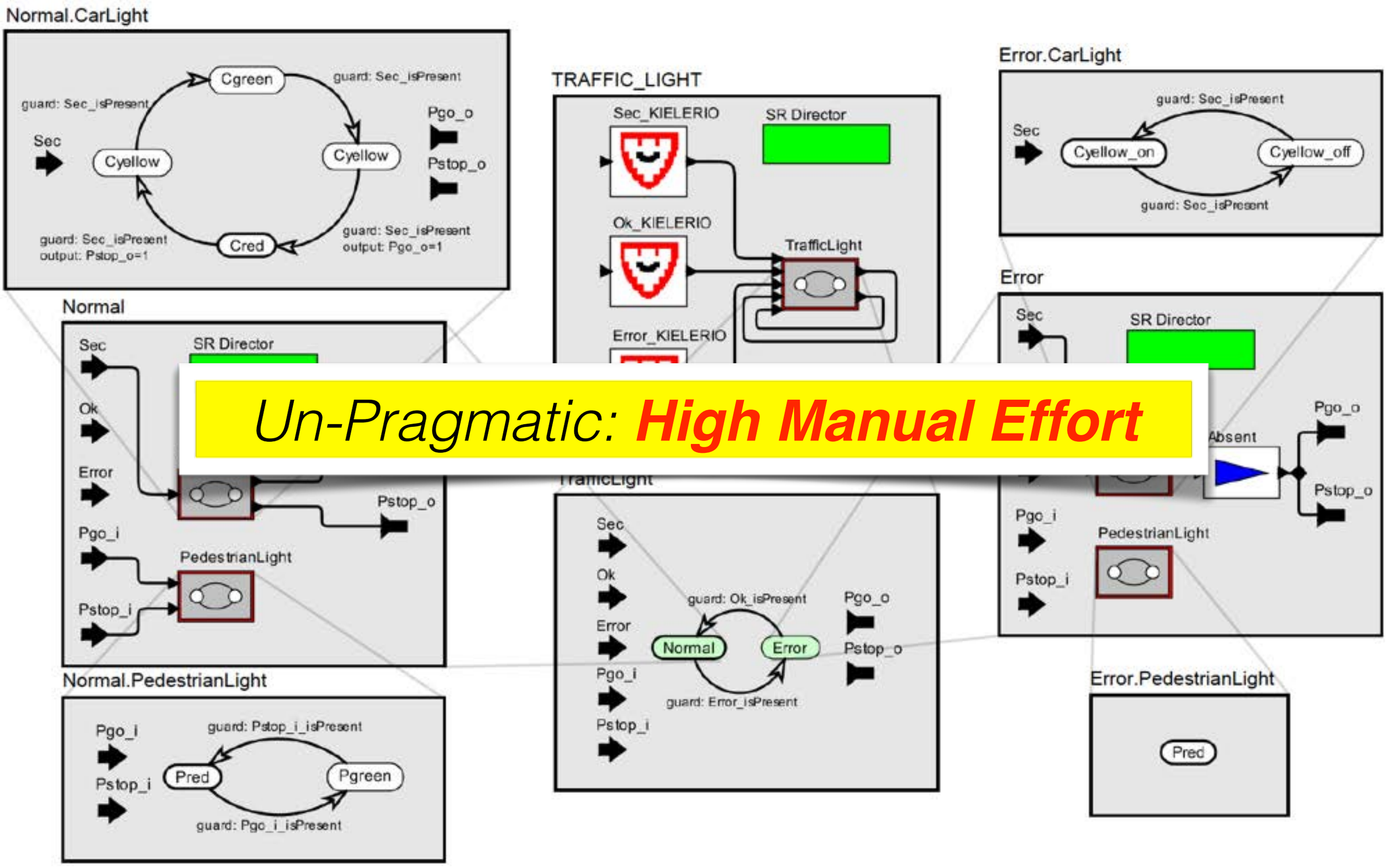


Un-Pragmatic: **Quickly Loose Details**

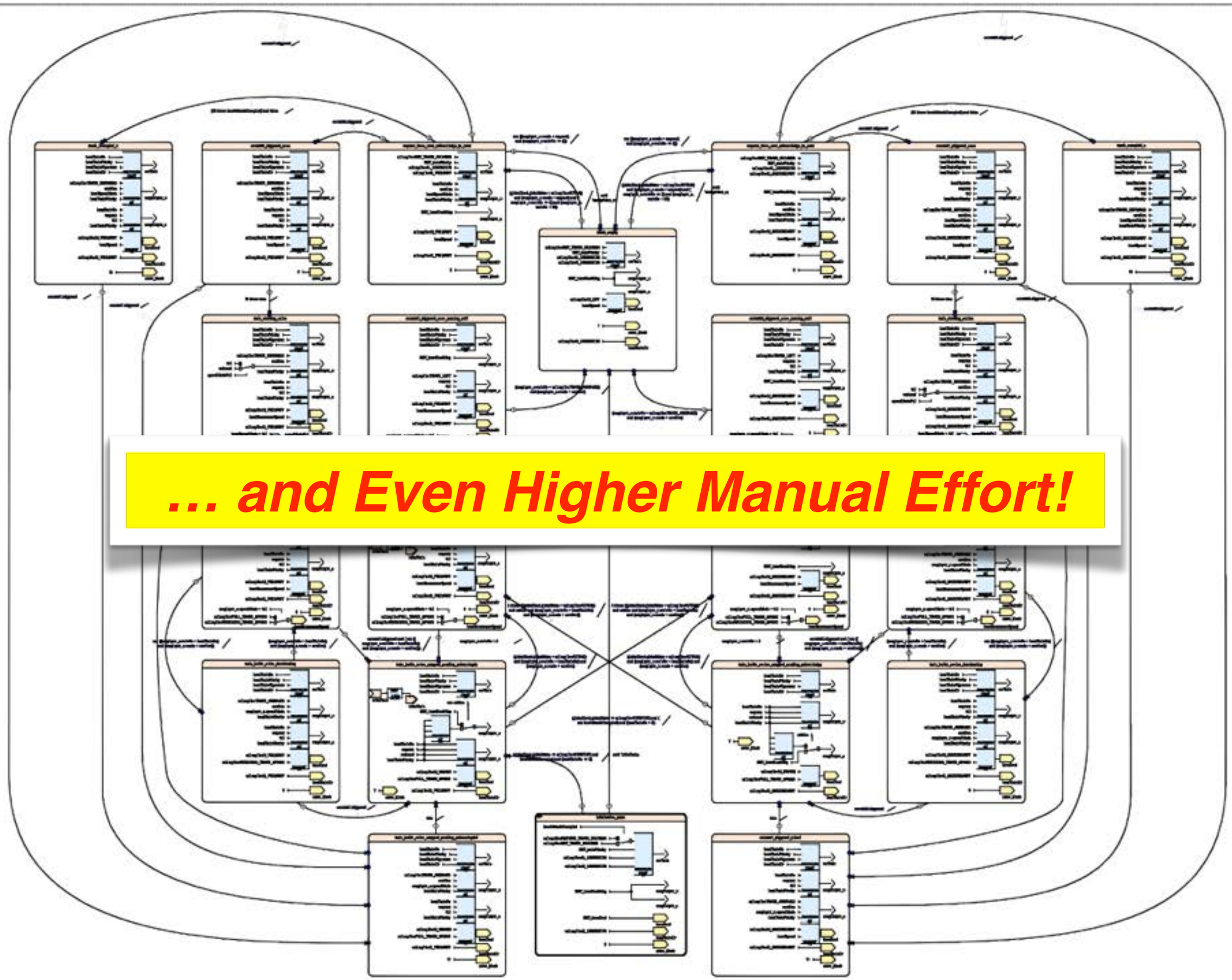


Un-Pragmatic: ***Lack of overview, tedious window management***

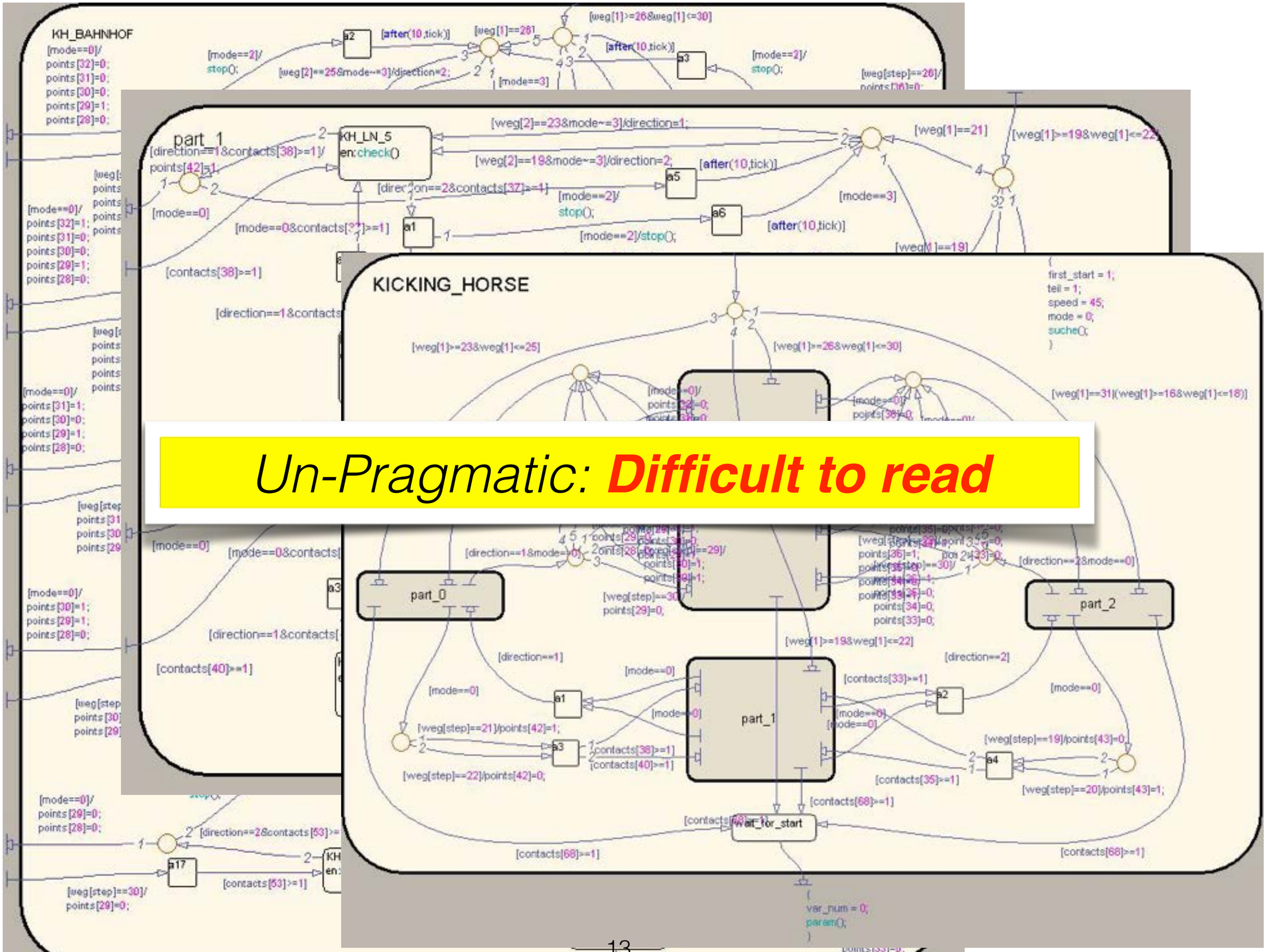
Imagine google maps would force you to navigate like that …



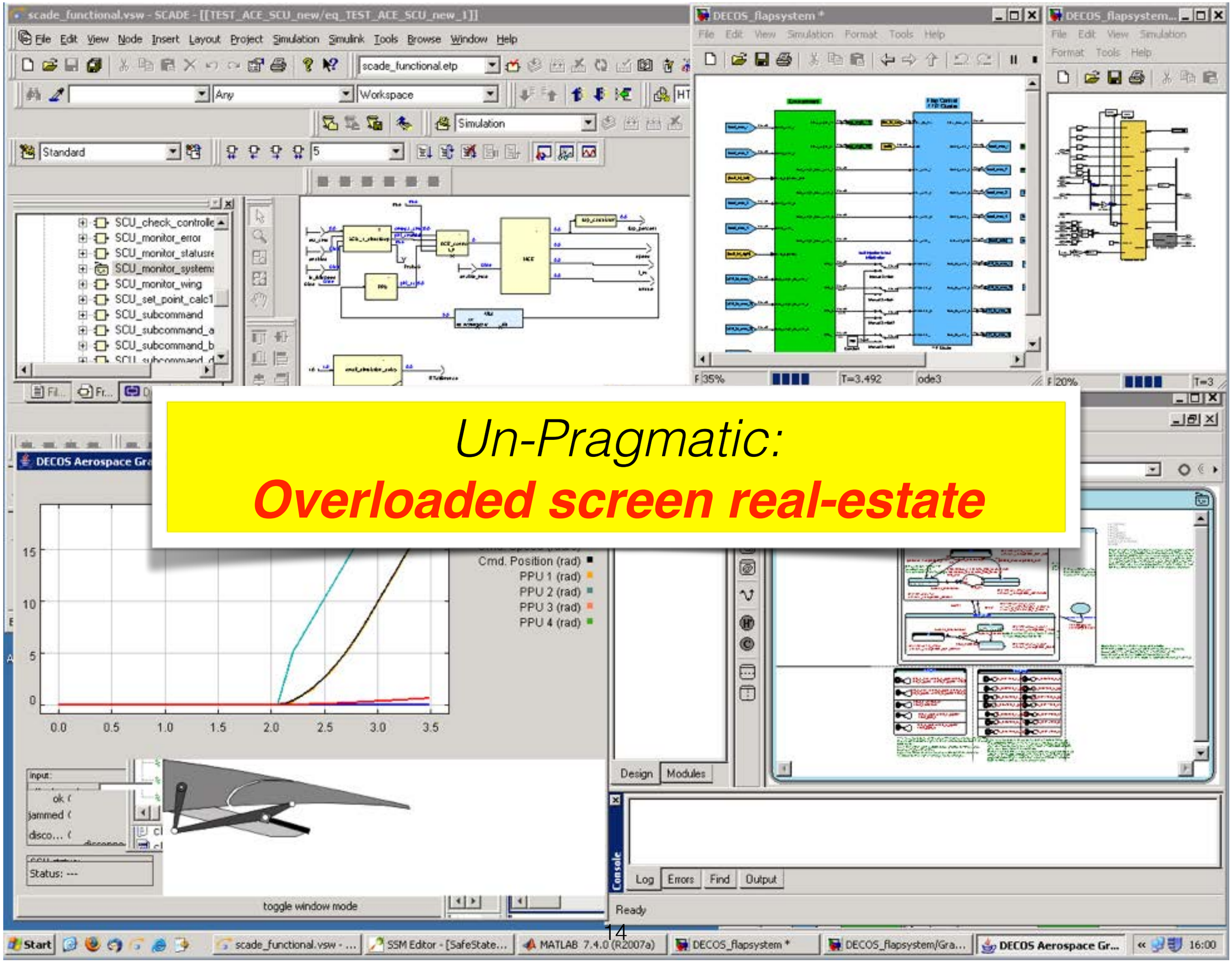
Un-Pragmatic: ***High Manual Effort***



… and Even Higher Manual Effort!

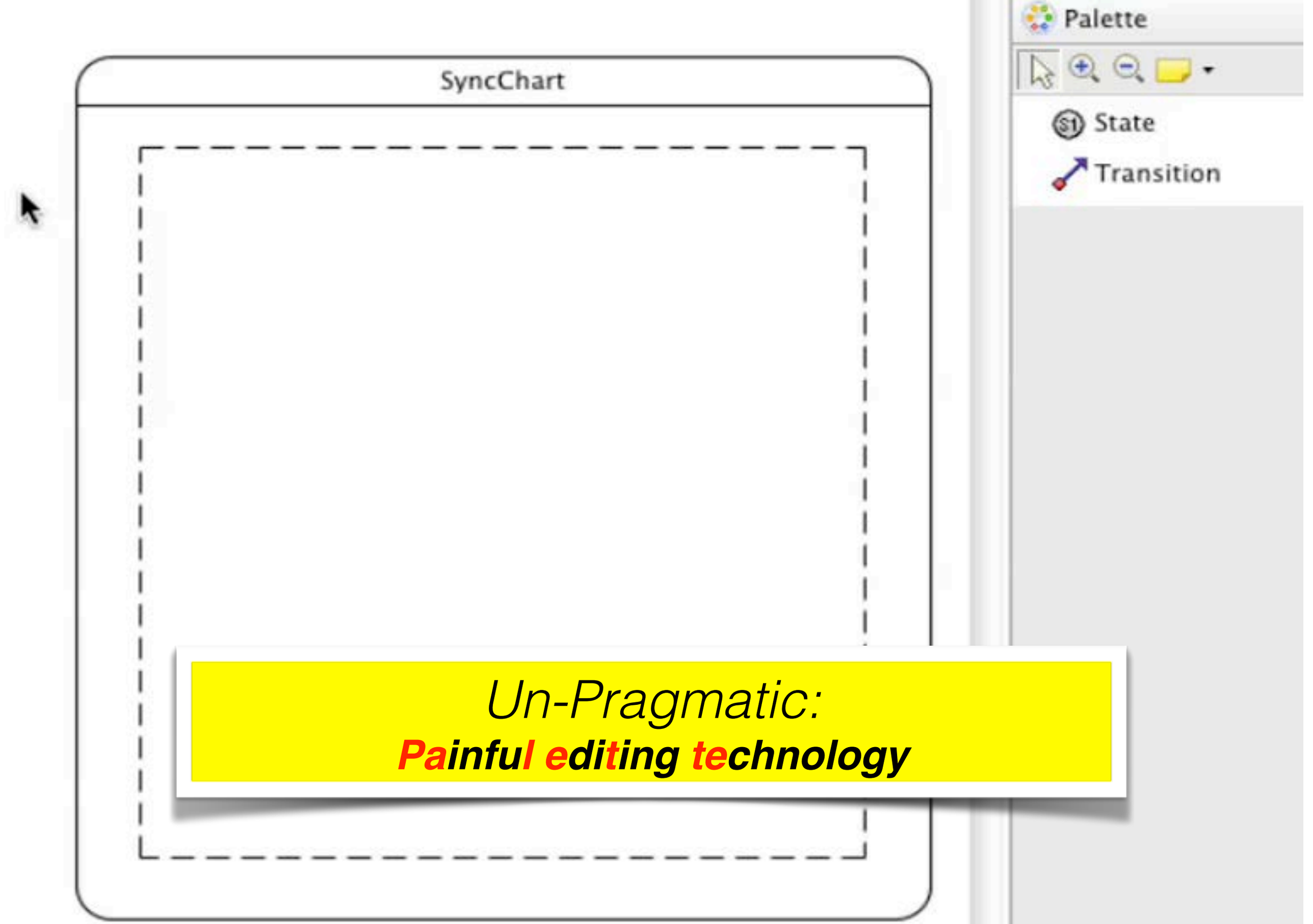


Un-Pragmatic: ***Difficult to read***



Un-Pragmatic:

Overloaded screen real-estate


SyncChart
Palette
State
Transition
Un-Pragmatic:
Painful editing technology

Are Textual DSLs the Answer?

Resource – VERA/GA/WeichenkontrolleUmstell12.logic – MENGES IDE – /Users/chsch/menges-workspace

*WeichenkontrolleUmstell12.logic

```
56    /** Ein berechtigtes Kommando zum Hilfsumstellen ist eingetroffen. */
57    from grund
58      if (hilfsUm()) then
59       change to hUml;
60
61    /** Ein berechtigtes Kommando zum Einzelsumstellen ist eingetroffen. */
62    from grund
63      if (!hilfsUm() && umEin()) then
64       change to uml;
65
66    /** Ein berechtigtes Kommando zum Stellen nach Rechts durch Befahrpunkt ist eingetroffen. */
67    from grund
68      if (hilfsUm() && !umEin() && umBpR() && !umBpL()) then
69       change to bPRe;
70
71    /** Ein berechtigtes Kommando zum Stellen nach Links durch Befahrpunkt ist eingetroffen. */
72    from grund
73      if (hilfsUm() && !umEin() && umBpL() && !umBpR()) then
74       change to bPLi;
75
76    /** Ein berechtigtes Kommando zum Stellen nach Rechts durch Fahrweg ist eingetroffen. */
77    from grund
78      if (!hilfsUm && !umEin && !umBpR && !umBpL && umFwR && !umFwL) then
79        change to fwLi;
```

Writable | Insert | 9 : 15

- Editing text much less tedious than editing graphics
- Revision control simpler
- Portable, tool independent

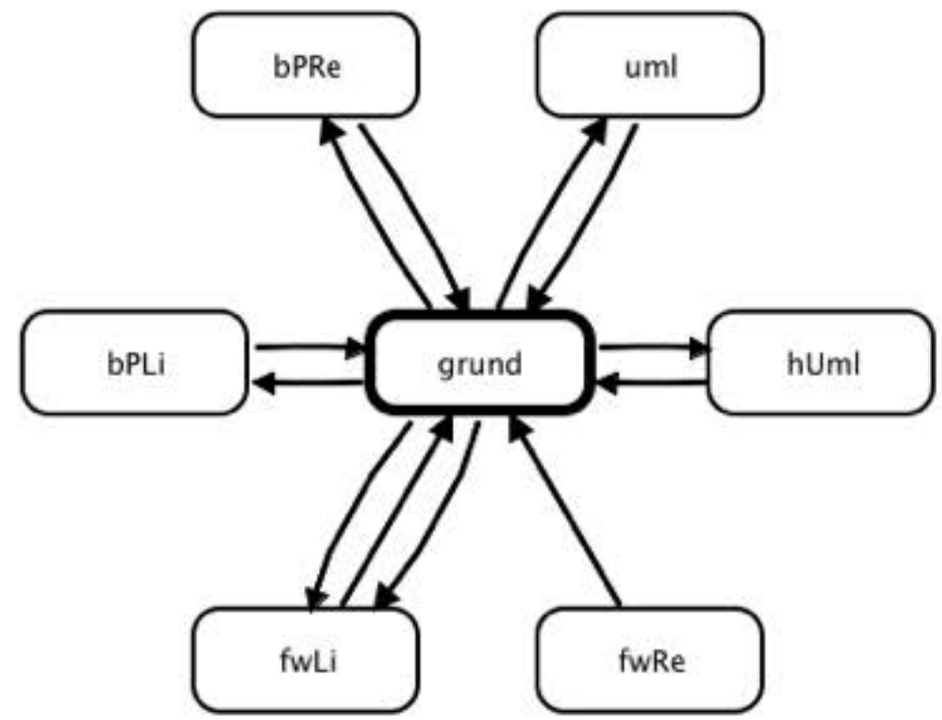
Still Want Graphical Views!

Resource – VERA/GA/WeichenkontrolleUmstell12.logic – MENGES IDE – /Users/chsch/menges-workspace

*WeichenkontrolleUmstell12.logic

```
56    /** Ein berechtigtes Kommando zum Hilfsumstellen ist ei
57    from grund
58      if (hilfsUm()) then
59       change to hUml;
60
61    /** Ein berechtigtes Kommando zum Einzelsumstellen ist
62    from grund
63      if (!hilfsUm() && umEin()) then
64       change to uml;
65
66    /** Ein berechtigtes Kommando zum Stellen nach Rechts d
67    from grund
68      if (hilfsUm() && !umEin() && umBpR() && !umBpL()) the
69       change to bPRe;
70
71    /** Ein berechtigtes Kommando zum Stellen nach Links du
72    from grund
73      if (hilfsUm() && !umEin() && umBpL() && !umBpR()) the
74       change to bPLi;
75
76    /** Ein berechtigtes Kommando zum Stellen nach Rechts d
77    from grund
78      if (!hilfsUm && !umEin && !umBpR && !umBpL && umFwR &
79       change to fwLi;
```

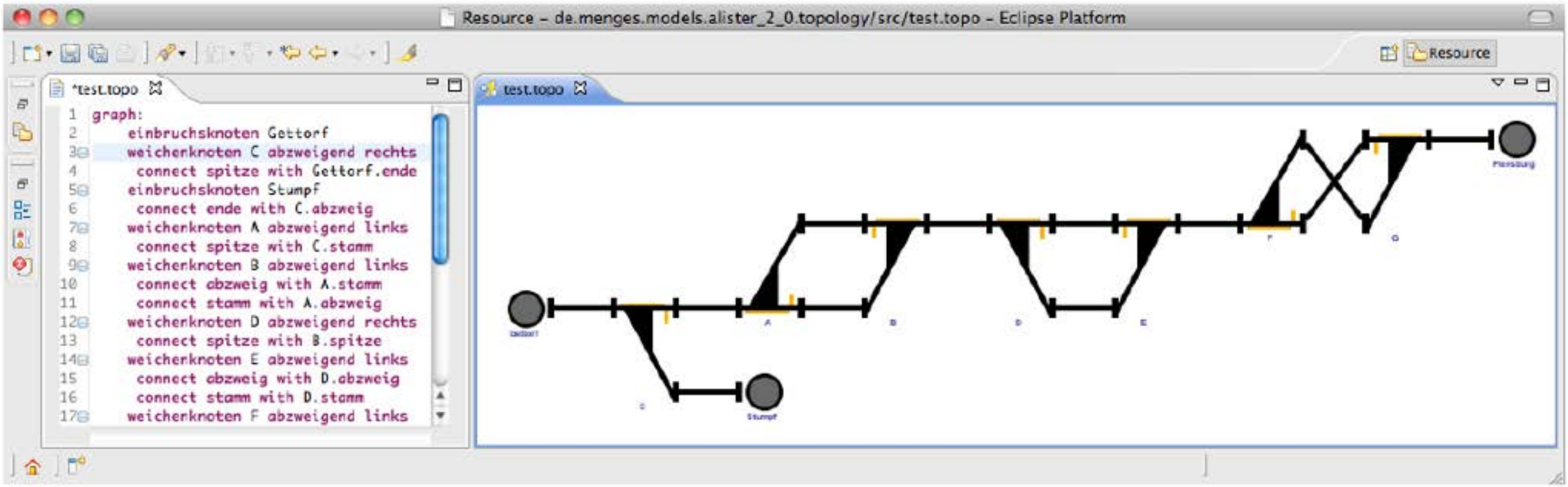
WeichenkontrolleUmstell12.logic



Writable | Insert | 9 : 15

- Text requires string matching to uncover structure
- Diagram makes structure obvious
- Unreachable state in model emerges

Still Want Graphical Views!



- Diagram uncovers specification flaw (upper right corner)

Pragmatics-Aware Modeling

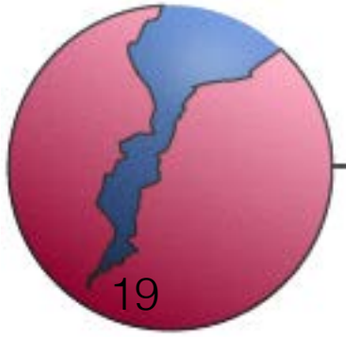
Free user of tedious mechanical work, such as . . .

- manual placing of graphical objects
- manual navigation in complex models

Focus on pragmatics:

- New interaction methodologies
- New analysis methodologies
- New ways to synthesize models

Our experimental platform:



KIELER

Kiel Integrated Environment for Layout Eclipse RichClient

Key to Pragmatics: The MVC Paradigm

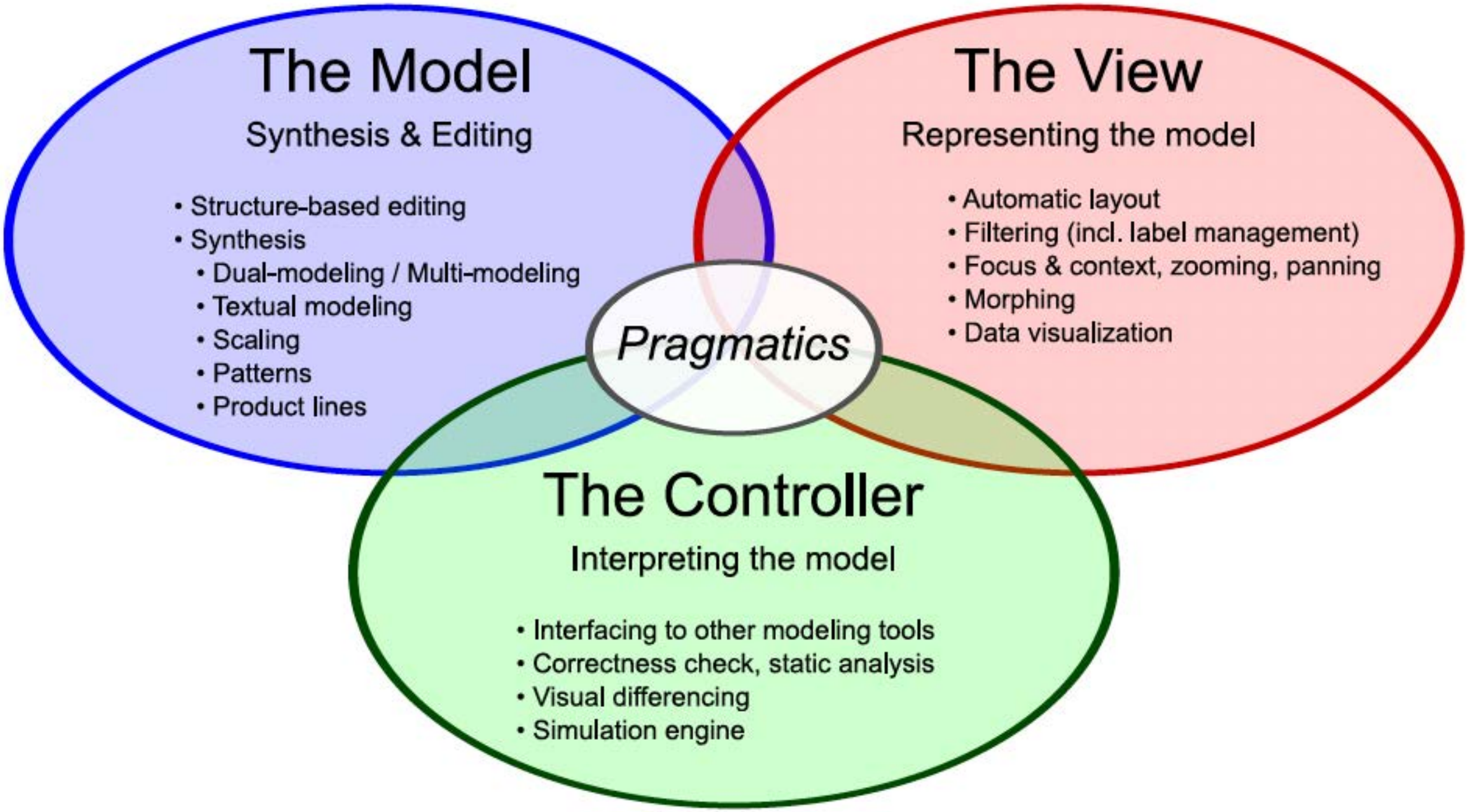
- A **model** represents knowledge.
  A model could be a single object (rather uninteresting),
  or it could be some structure of objects.

- A **view** is a (visual) representation of its model.
  It would ordinarily highlight certain attributes of the model
  and suppress others.
  It is thus acting as a presentation filter.

- A **controller** is the link between a user and the system.
  It provides the user with input by arranging for relevant views
  to present themselves in appropriate places on the screen.

Trygve Reenskaug
Models - Views - Controllers
Xerox PARC technical note, 1979

Key to Pragmatics: The MVC Paradigm



Fuhrmann, von Hanxleden
On the Pragmatics of Model-Based Design
15th Monterey Workshop 2008, LNCS 6028 (2010)

Overview

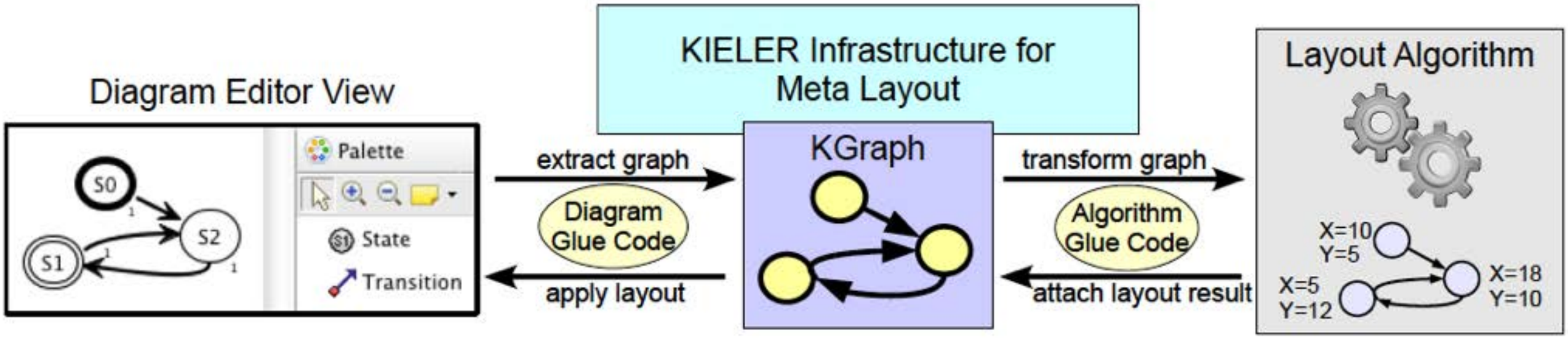
Pragmatics-Aware Modeling

- Definition and Motivation
- MVC a Key to Pragmatics!

Pragmatics/KIELER Spotlights

- **KIELER Infrastructure for Meta Layout (KIML)**
- **An Experiment**

Wrap-Up



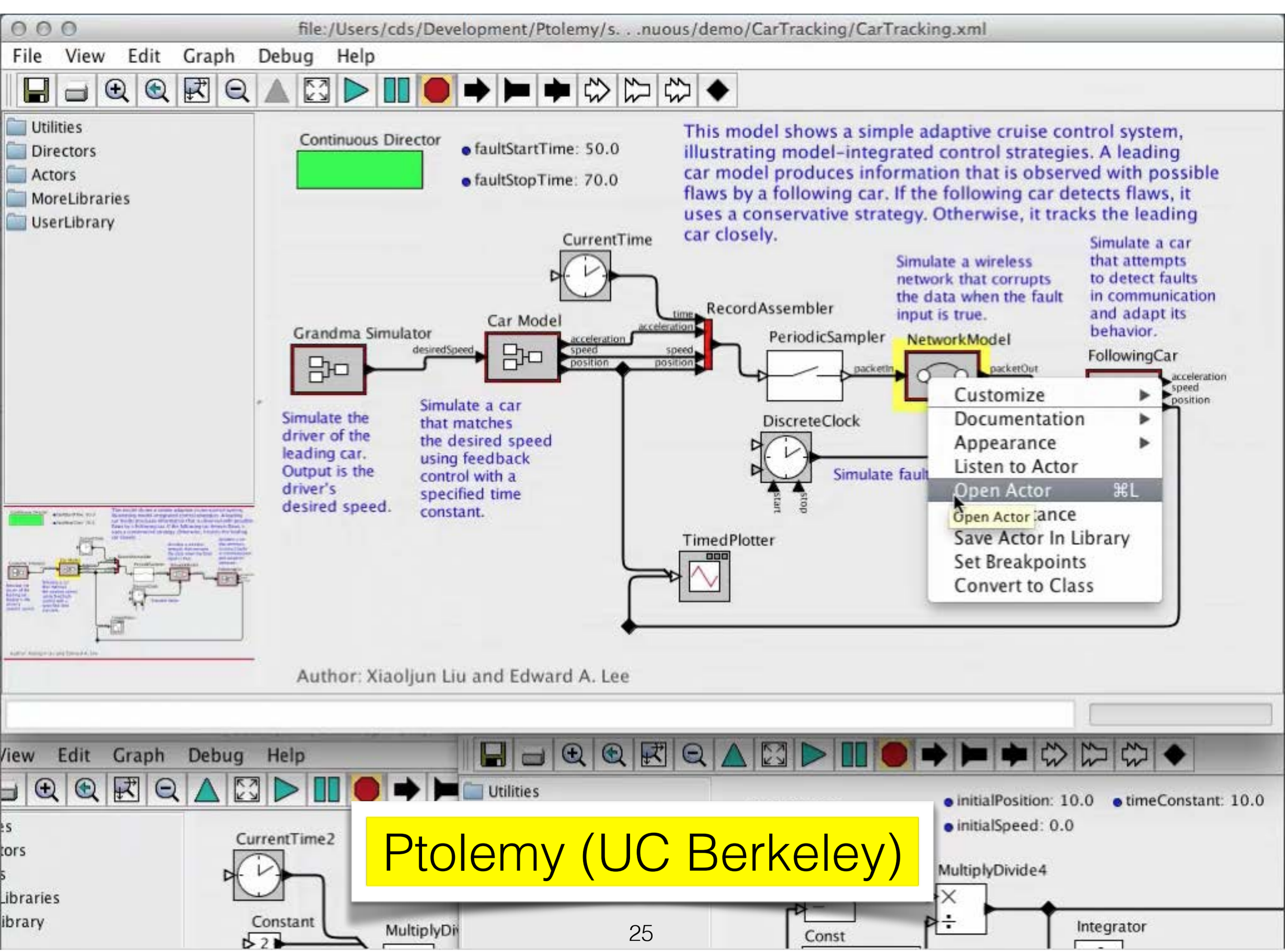
- GMF
- Graphiti
- Papyrus
- Yakindu

- Graphviz (Dot, Neato, FDP, Twopi, Circo)
- Open Graph Drawing Framework (OGDF) (Layer-based, Planarization, Force-directed)
- Own Implementations (Data flow diagrams)

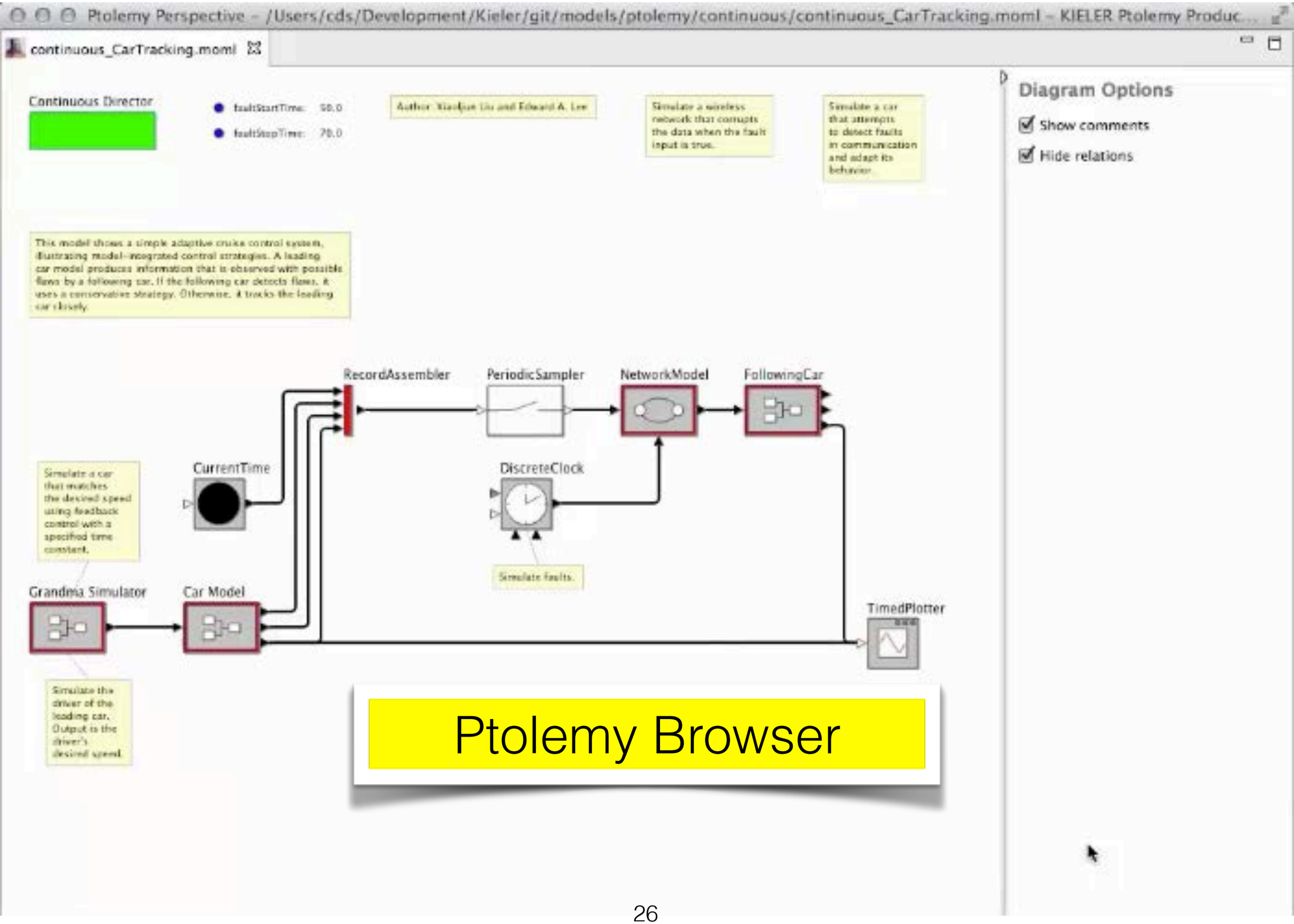
Schulze, Spönemann, v. Hanxleden
Drawing Layered Graphs with Port Constraints
JVLC'14

Eclipse File Edit Navigate Search Project Run Window Help

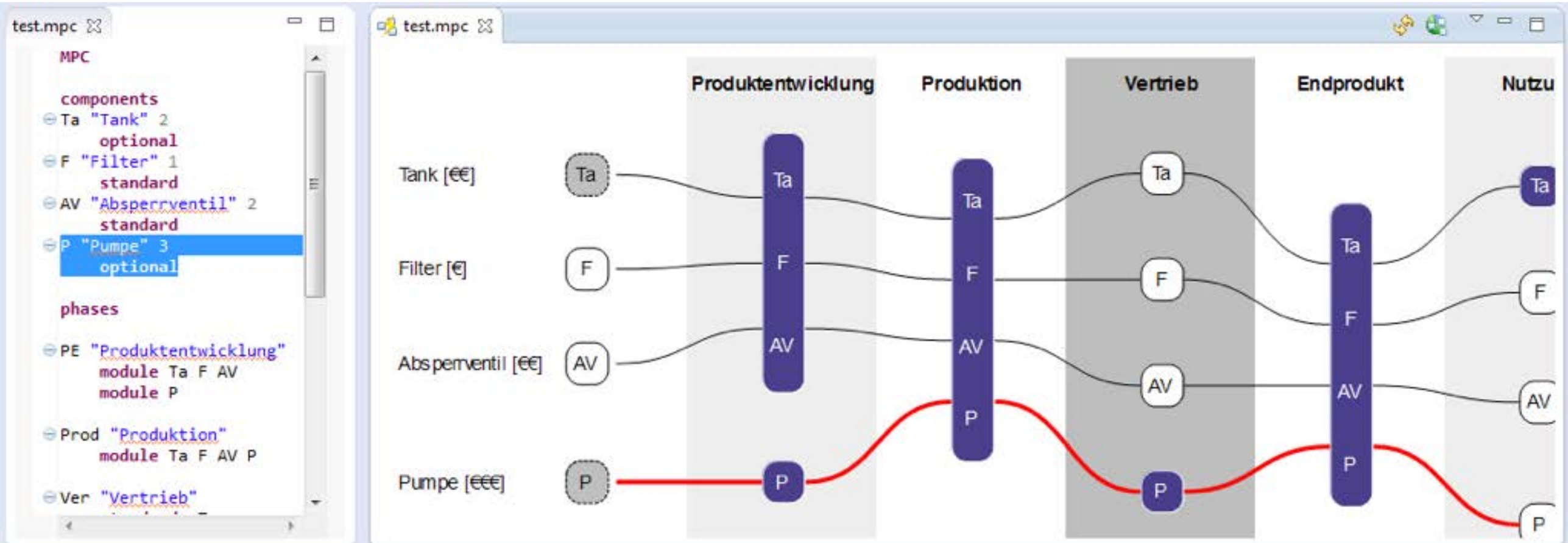
Resource - Eclipse Platform



Ptolemy (UC Berkeley)

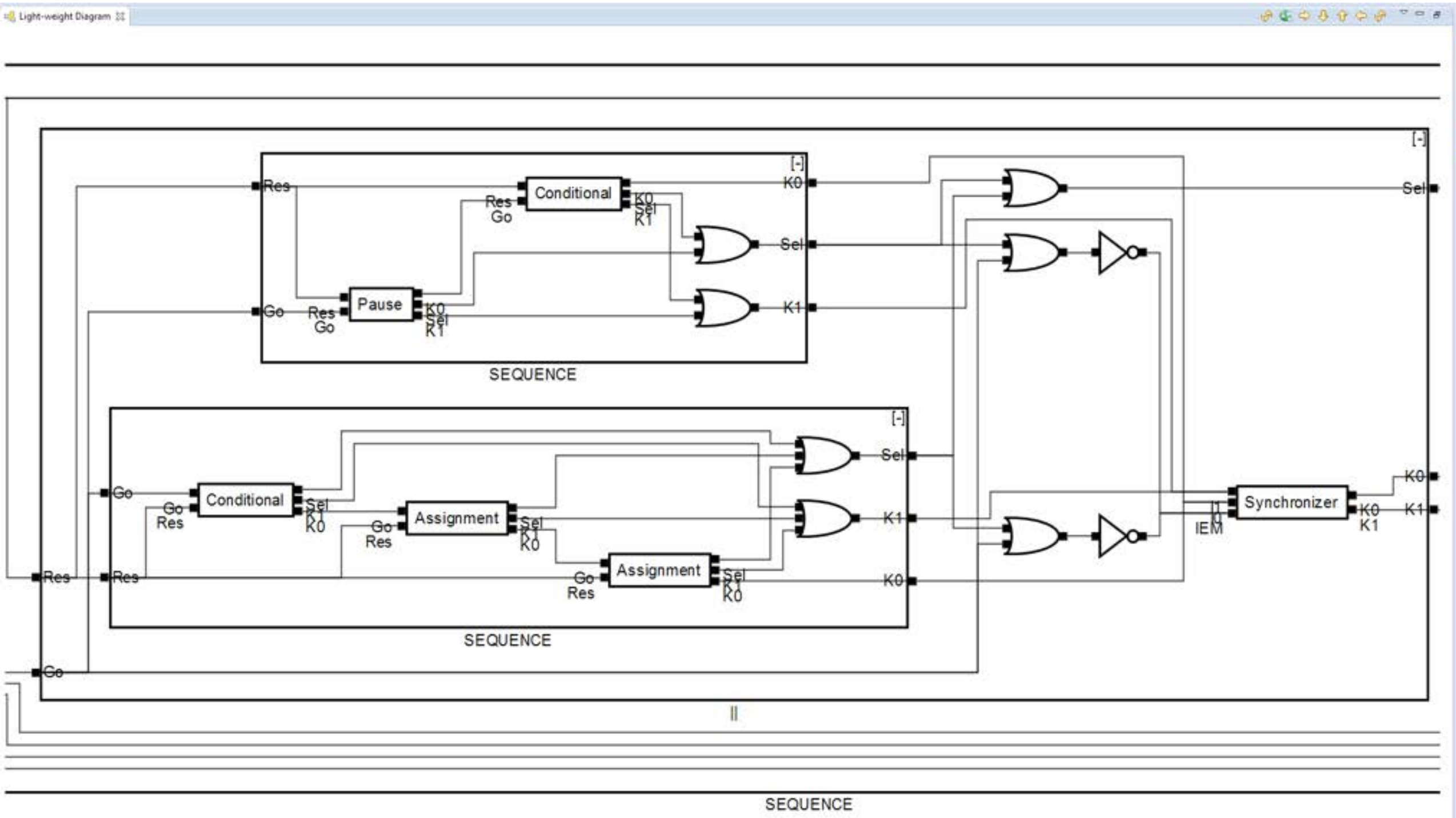

Ptolemy Perspective - /Users/cds/Development/Kieler/git/models/ptolemy/continuous/continuous_CarTracking.moml - KIELER Ptolemy Produc...
continuous_CarTracking.moml
Continuous Director
faultStartTime: 50.0
faultStopTime: 70.0
Author: Xiaojun Liu and Edward A. Lee
Simulate a wireless network that corrupts the data when the fault input is true.
Simulate a car that attempts to detect faults in communication and adapt its behavior.
This model shows a simple adaptive cruise control system, illustrating model-integrated control strategies. A leading car model produces information that is observed with possible flaws by a following car. If the following car detects flaws, it uses a conservative strategy. Otherwise, it tracks the leading car closely.
RecordAssembler
PeriodicSampler
NetworkModel
FollowingCar
CurrentTime
DiscreteClock
Simulate faults.
Simulate a car that matches the desired speed using feedback control with a specified time constant.
Grandma Simulator
Car Model
TimedPlotter
Simulate the driver of the leading car. Output is the driver's desired speed.
Ptolemy Browser
Diagram Options
Show comments
Hide relations

Example: Component Clustering

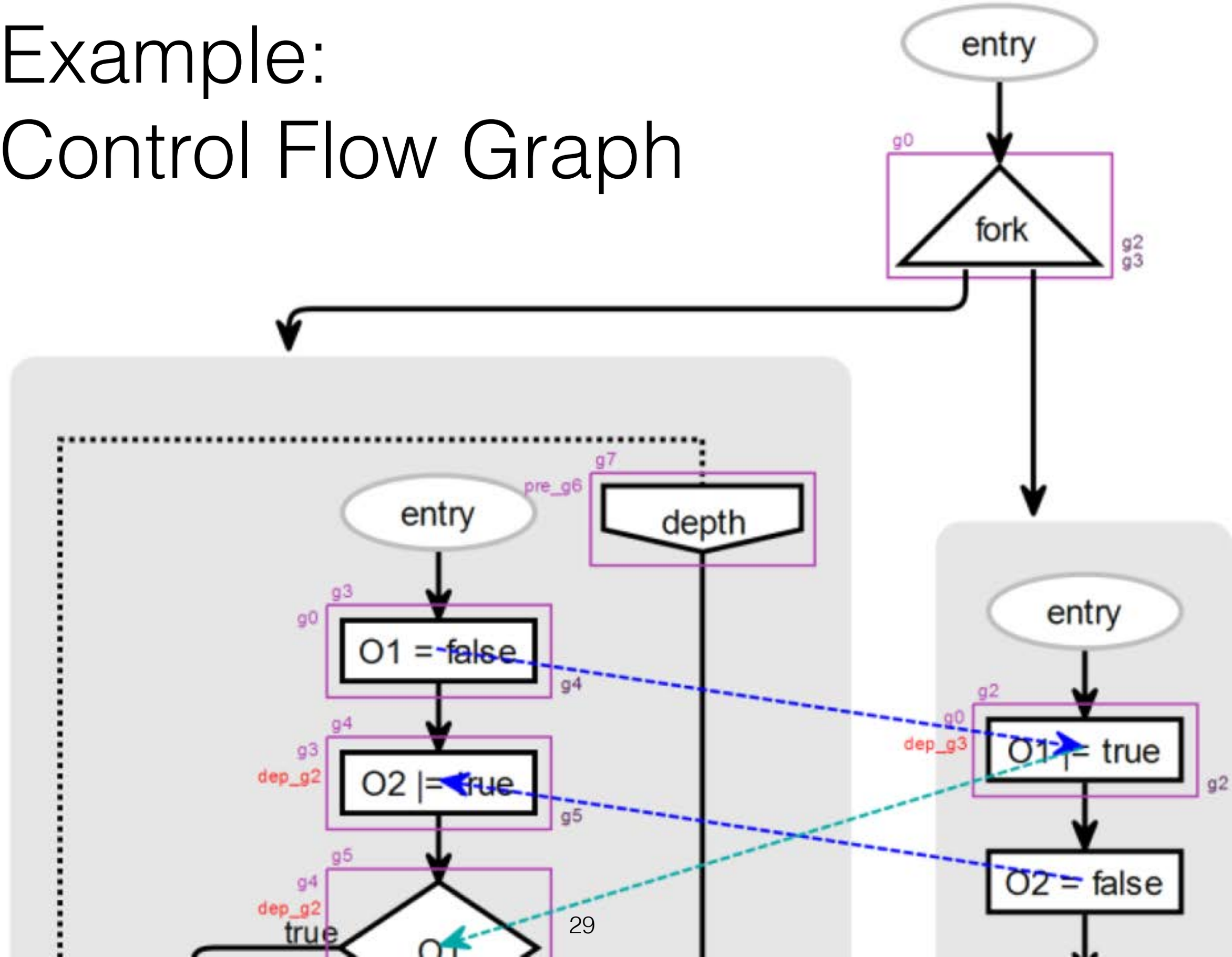


Christoph Blees
Eine Methode zur Entwicklung modularer Produktfamilien
Dissertation, TU Hamburg-Harburg, Schriftenreihe
Produktentwicklung und Konstruktionstechnik vol. 3, 2011

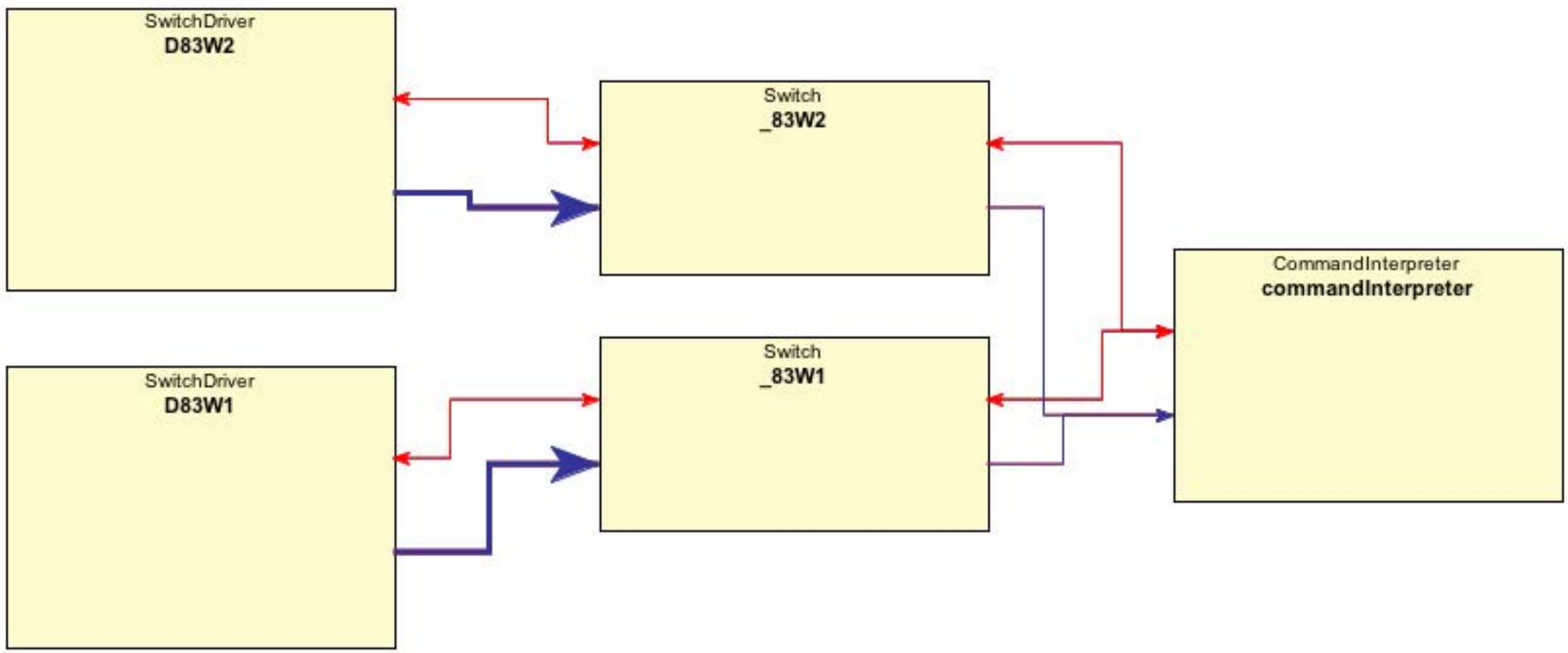
Example: Net Lists



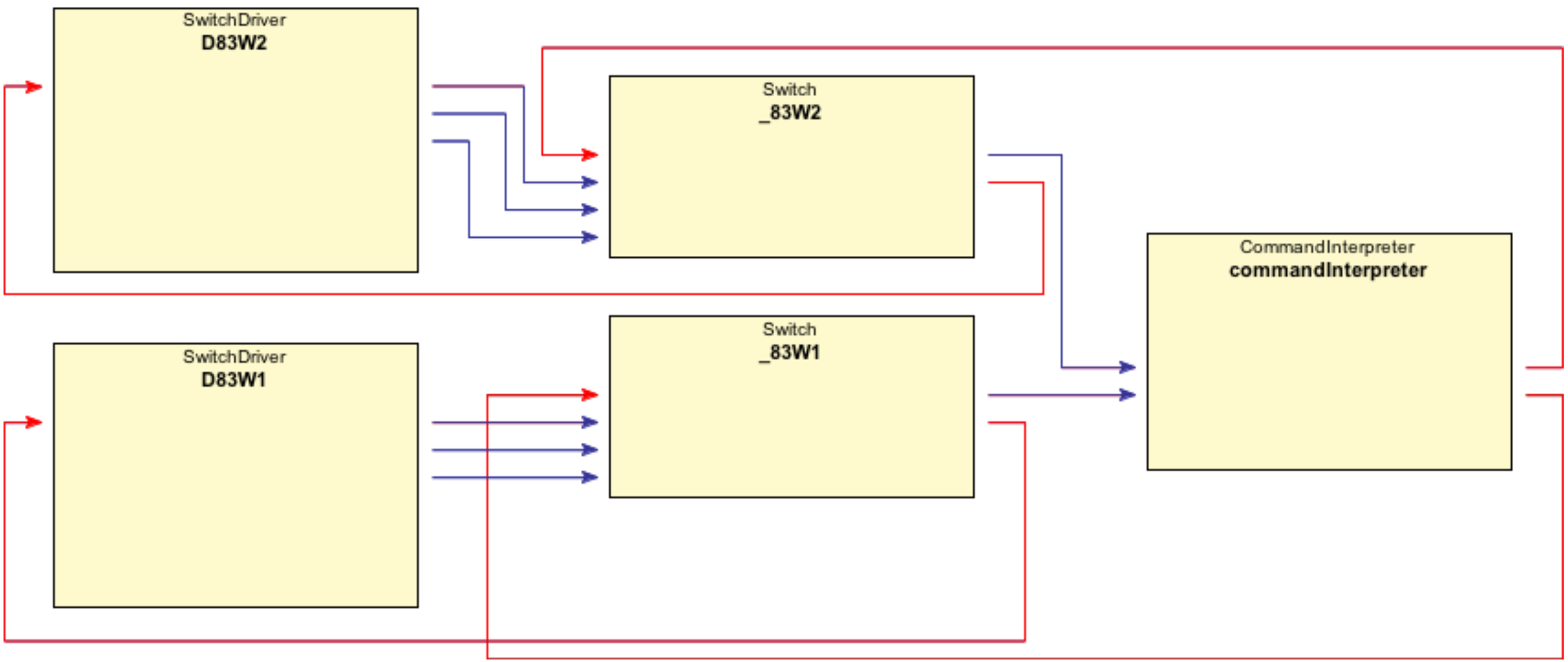
Example:
Control Flow Graph



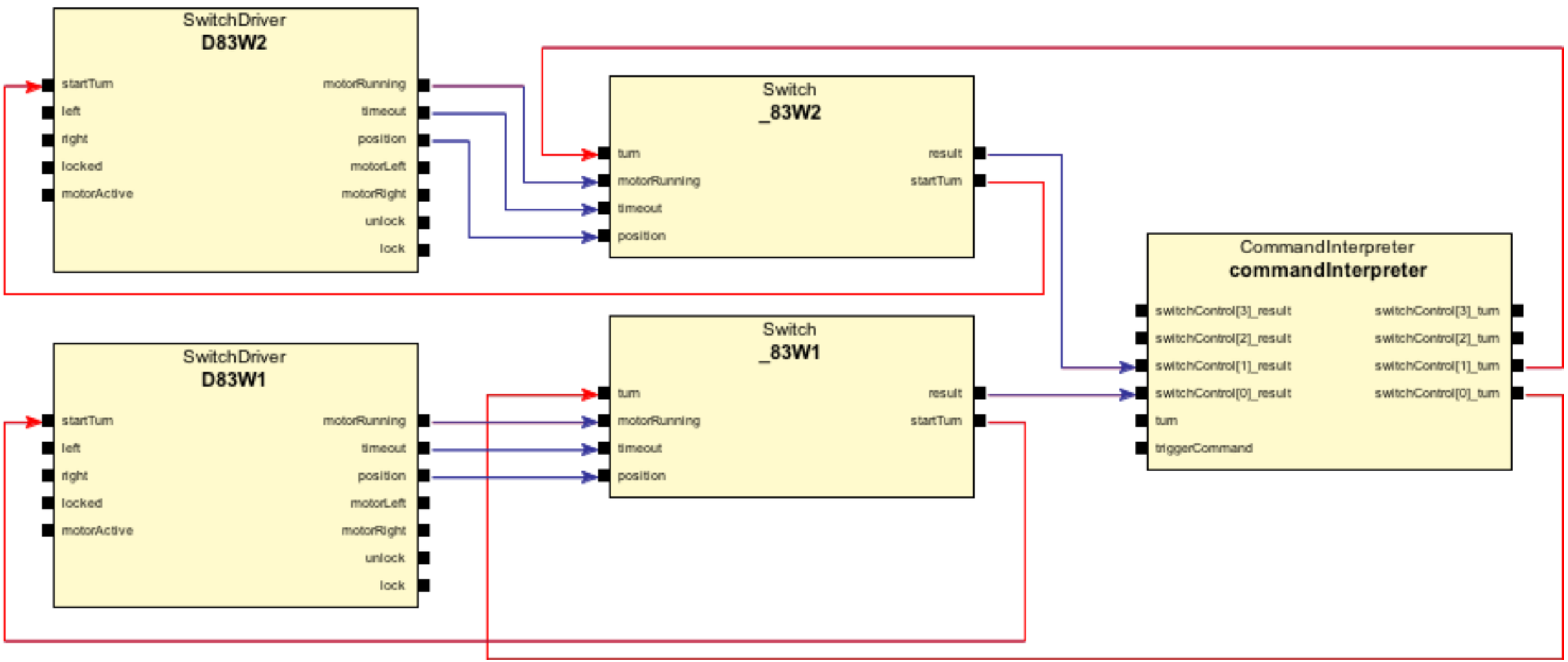
Example: Component Diagram



Example: Component Diagram

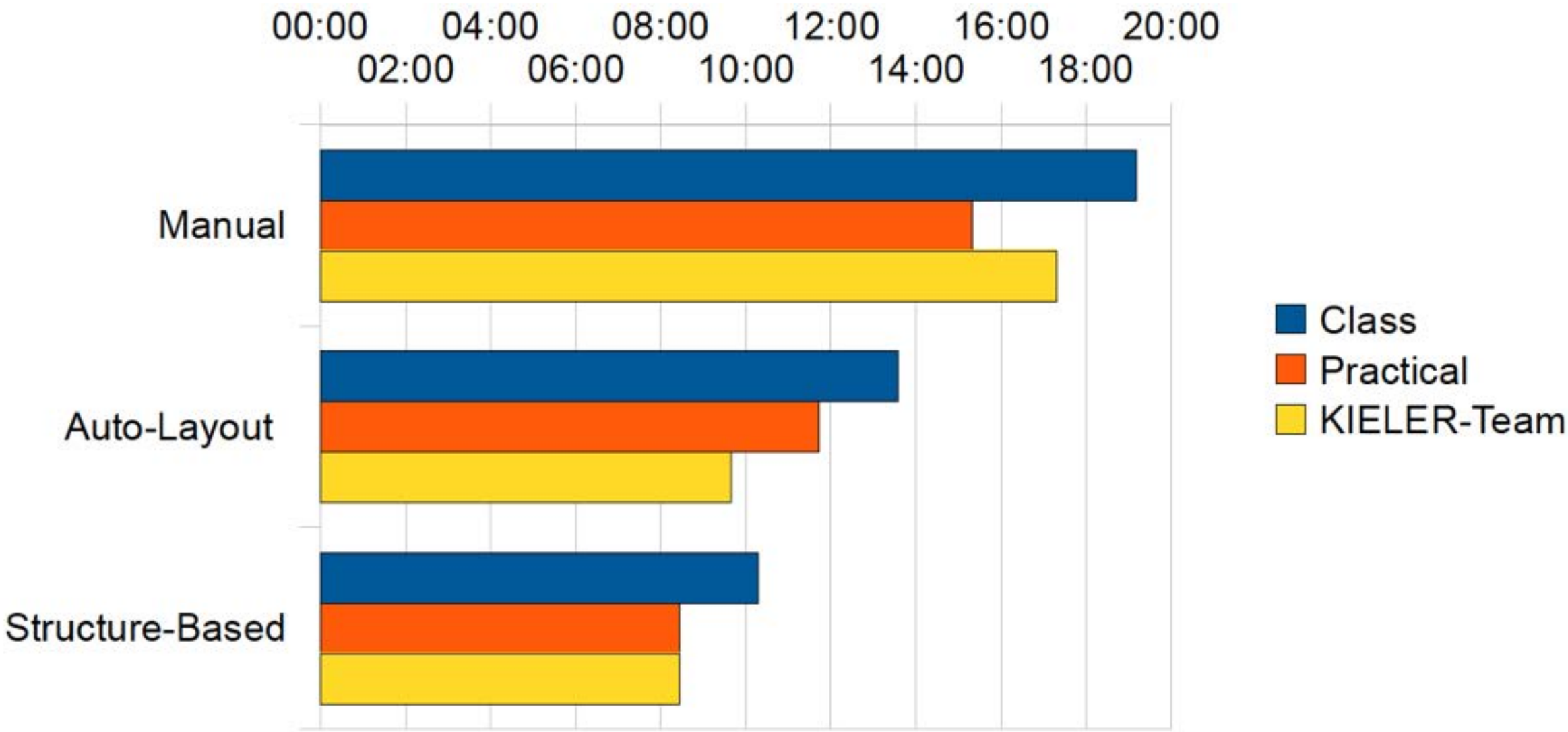


Example: Component Diagram



Editing Efficiency: An Experiment

Task: Create diagram from textual specification



Fuhrmann, v. Hanxleden
Taming Graphical Modeling
Proceedings of the ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MoDELS'10), volume 6394 of LNCS, page 196–210, October 2010

Wrap-Up Modeling Pragmatics

- The Problem: Lots of productivity wasted with drawing diagrams manually
- Our Approach: Pragmatics-aware modeling
  - Let designer concentrate on model, automatically synthesize views
  - Employ filtering to synthesize views customized to user stories
  - Key enabler: automatic layout

Problem Set 1

- Due: Wed, 20 April
- Generally, you may write in German or English

Problem 1: Study [Fuhrmann, von Hanxleden MODELS'10] and answer the following questions:

1. What does *focus & context* mean? Give an example of where you used this, or would have liked to use it.
2. Give an example application of when you have used *DND editing*. In that application, would alternatives have been useful/feasible?
3. What is *view management*? Give an example where you have used this. Give another example where you would have liked to use it but could not do so because of tool limitations.

Fuhrmann, v. Hanxleden
Taming Graphical Modeling
Proceedings of the ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MoDELS'10), volume 6394 of LNCS, page 196–210, October 2010

Problem Set 1

Problem 2: Study [Petre, CACM'95] and answer the following questions:

1. Name an example of where you used secondary notation in a visual language

2. What arguments do you see in the paper for/against the usage of automatic layout

Marian Petre
Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming
Communications of the ACM (CACM), June 1995, Vol. 38, No. 6, 33–44