Taming Graphical Modeling
On Pragmatics-Aware Model-Driven Engineering

KIELER
Kiel Integrated Environment for Layout Eclipse RichClient

Reinhard von Hanxleden

Real-Time Systems and Embedded Systems Group
Department of Computer Science
Faculty of Engineering
Christian-Albrechts-Universität zu Kiel
`www.informatik.uni-kiel.de/rtsys`

MDE Workshop
HPI, Potsdam, May 30, 2014

Pragmatics of Model-Based Design

Pragmatics: relation of signs to their users

Pragmatics of Model-Based Design

Pragmatics: relation of signs to their users

$+$

Syntax: relations between signs

Pragmatics of Model-Based Design

Pragmatics: relation of signs to their users

+

Syntax: relations between signs

+

Semantics: relations between signs and the things they refer to

Pragmatics of Model-Based Design

Pragmatics: relation of signs to their users

$+$

Syntax: relations between signs

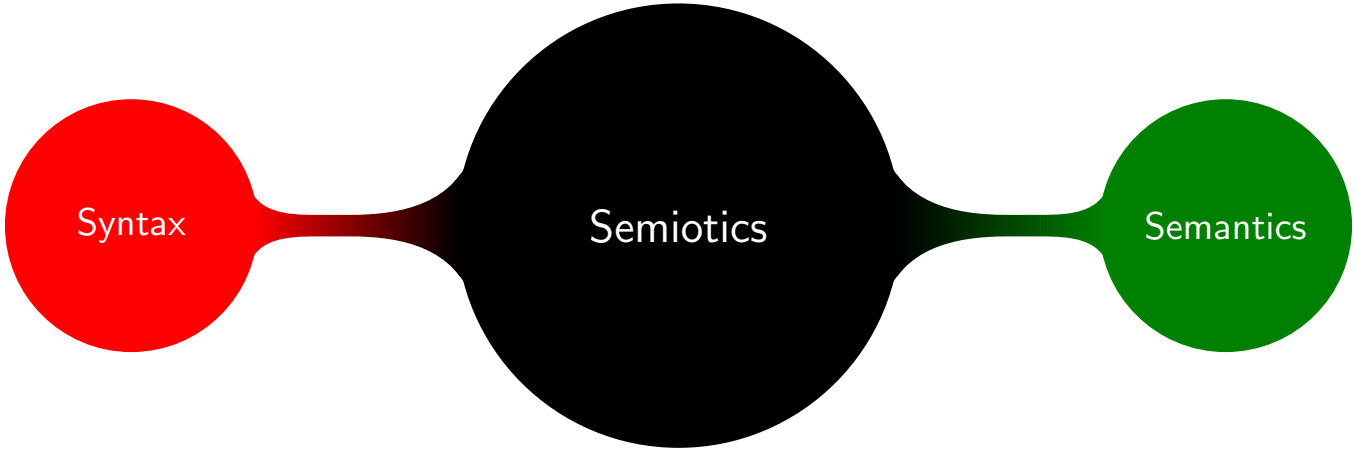
$+$

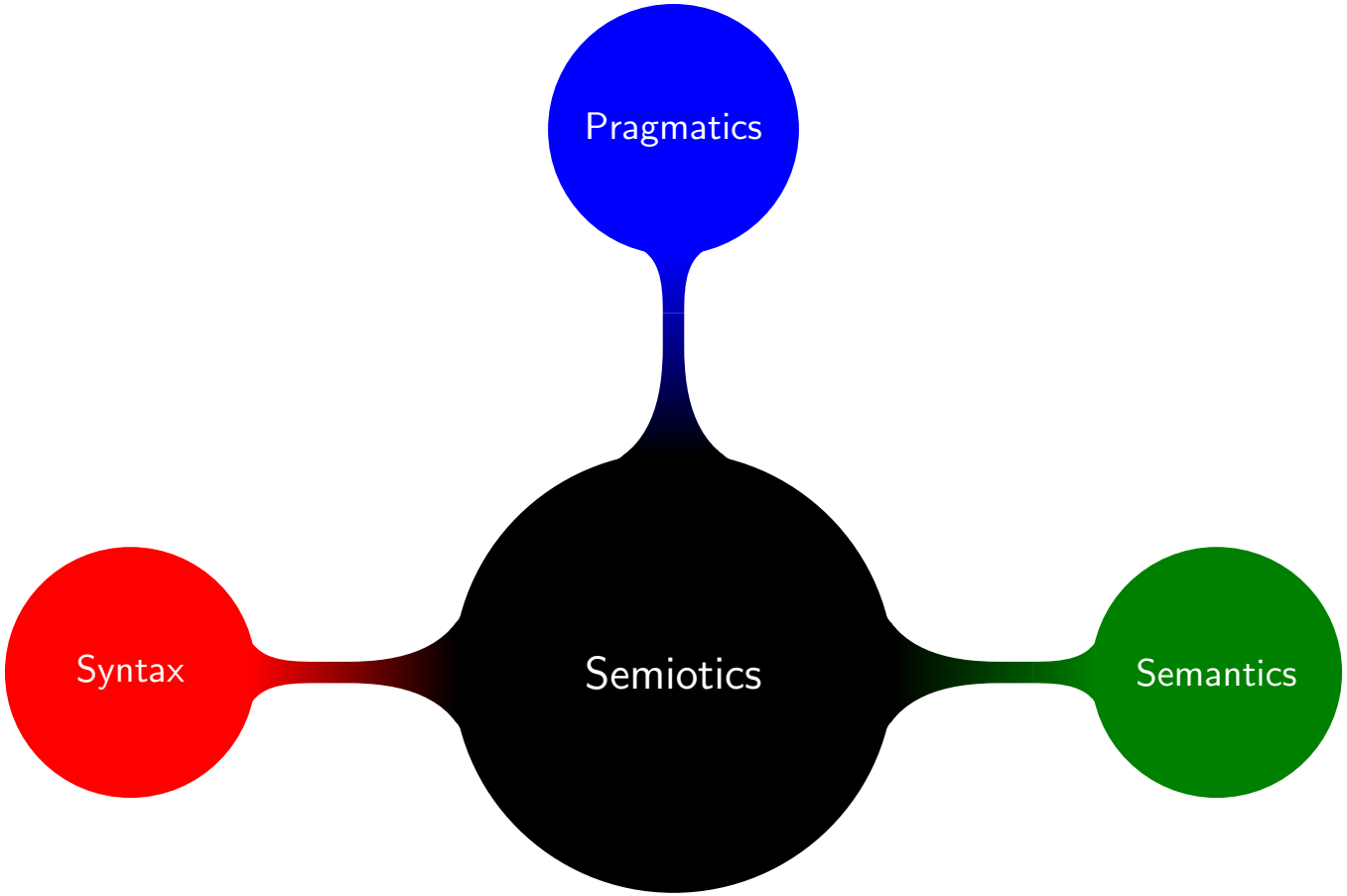
Semantics: relations between signs and the things they refer to

$=$

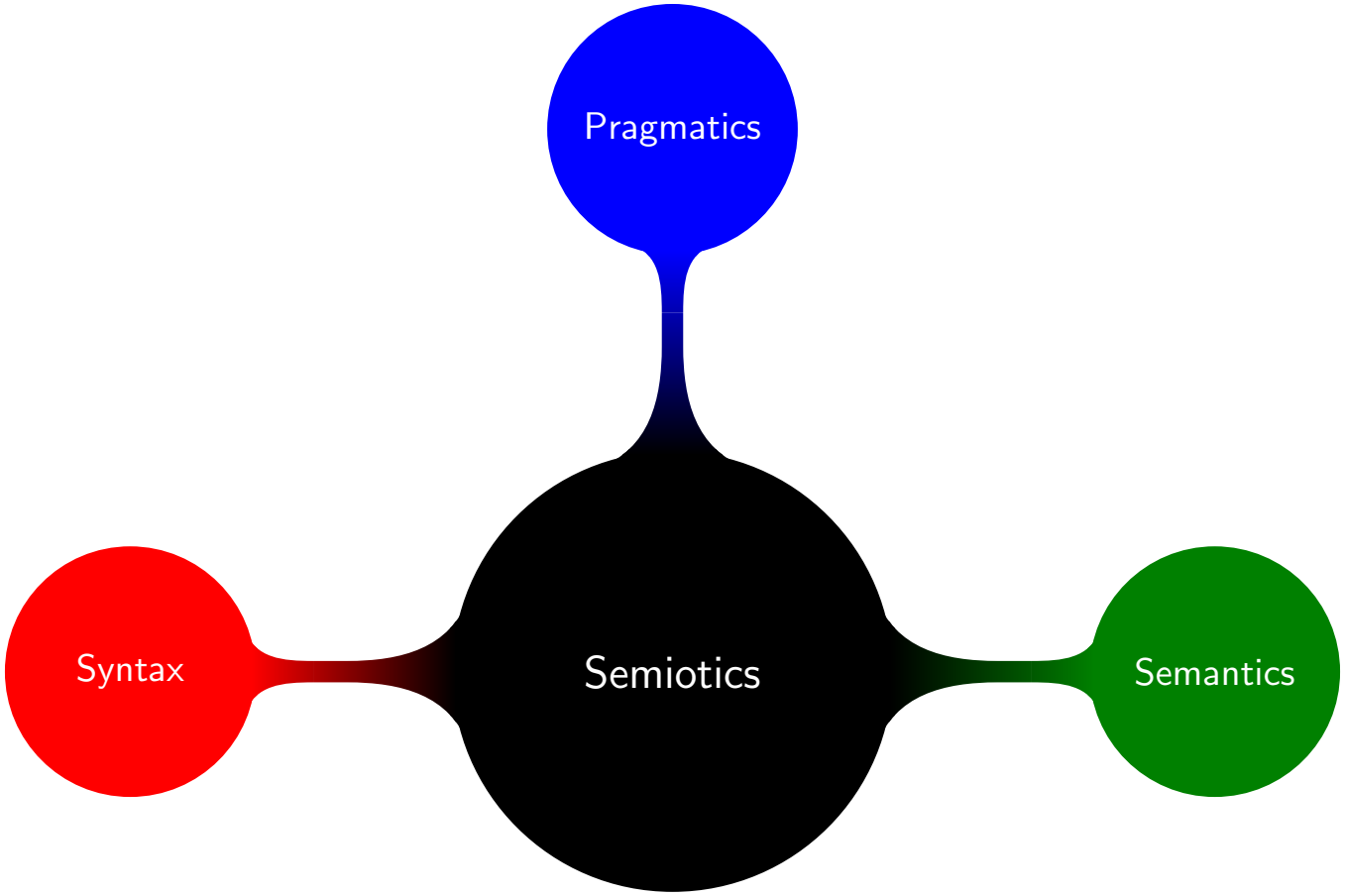
Semiotics: how meaning is constructed and understood

(Charles Morris, *Foundation of the Theory of Signs*, 1938)


Syntax
Semiotics
Semantics



Modeling pragmatics $=_{def}$ practical issues of model creation, editing, browsing, versioning, ...

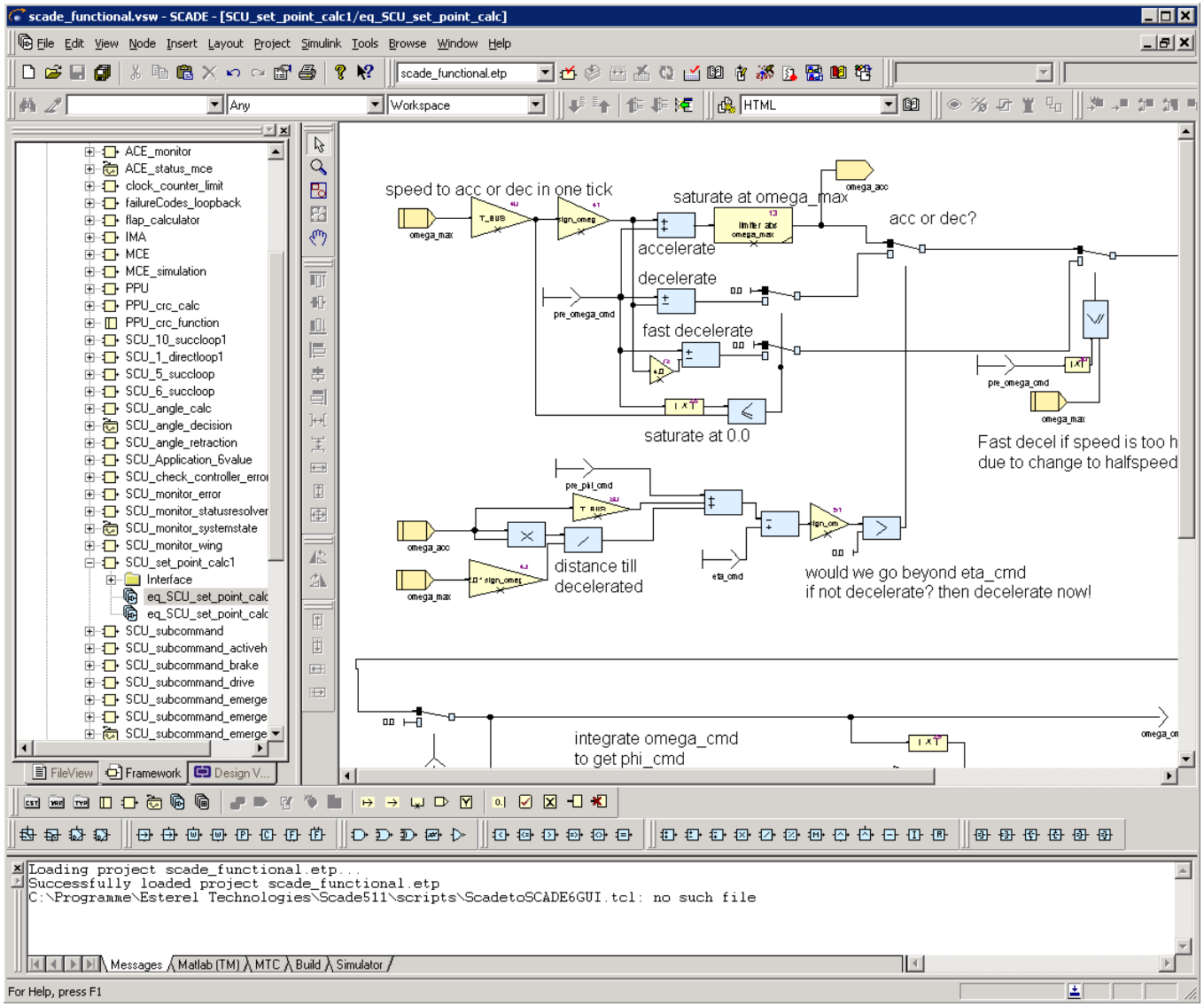


Modeling pragmatics $=_{def}$ practical issues of model creation, editing, browsing, versioning, ...

Pragmatics-aware modeling $=_{def}$ model-driven engineering with specific support for designer productivity

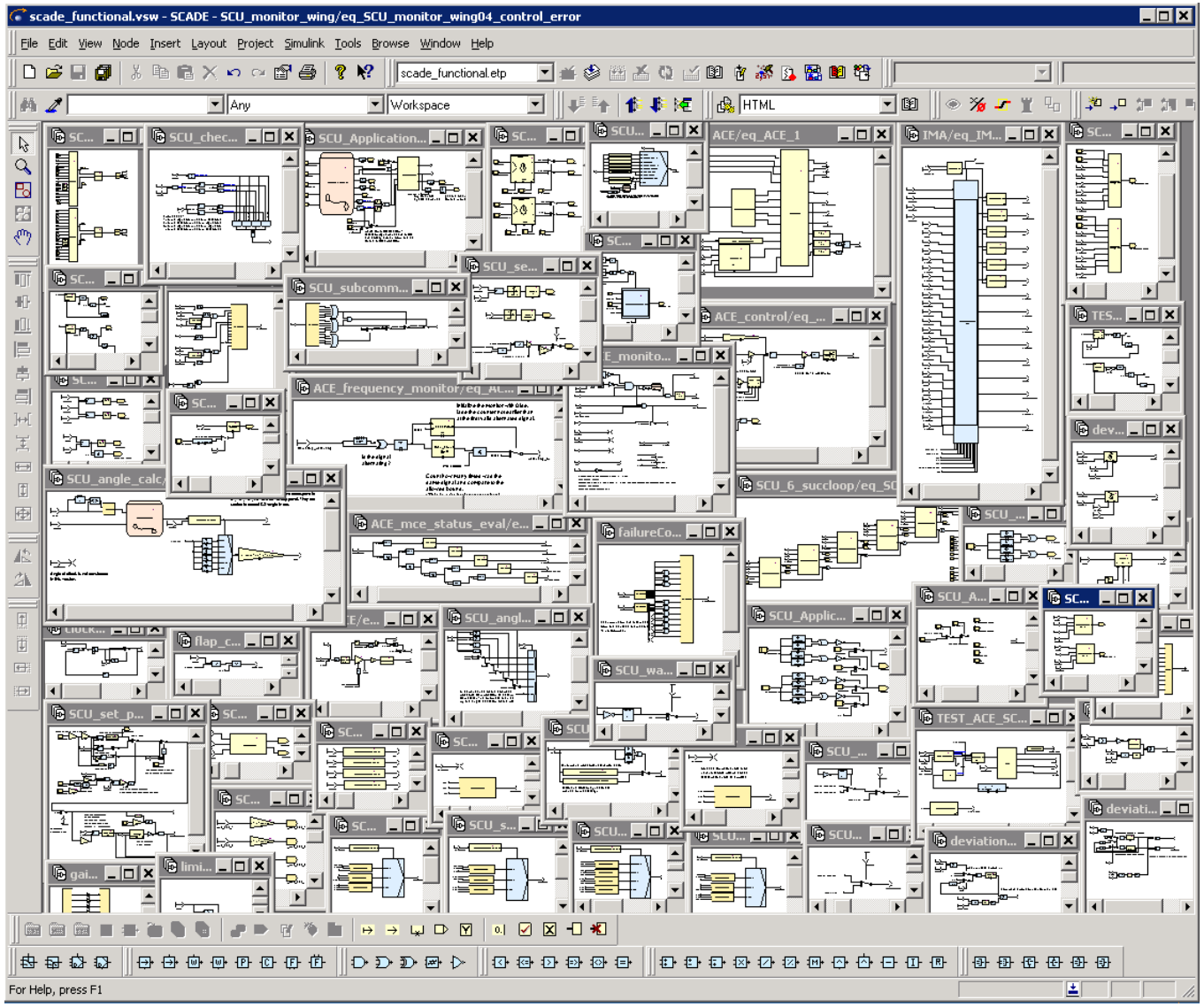
Pragmatics-Unaware Modeling

Context missing



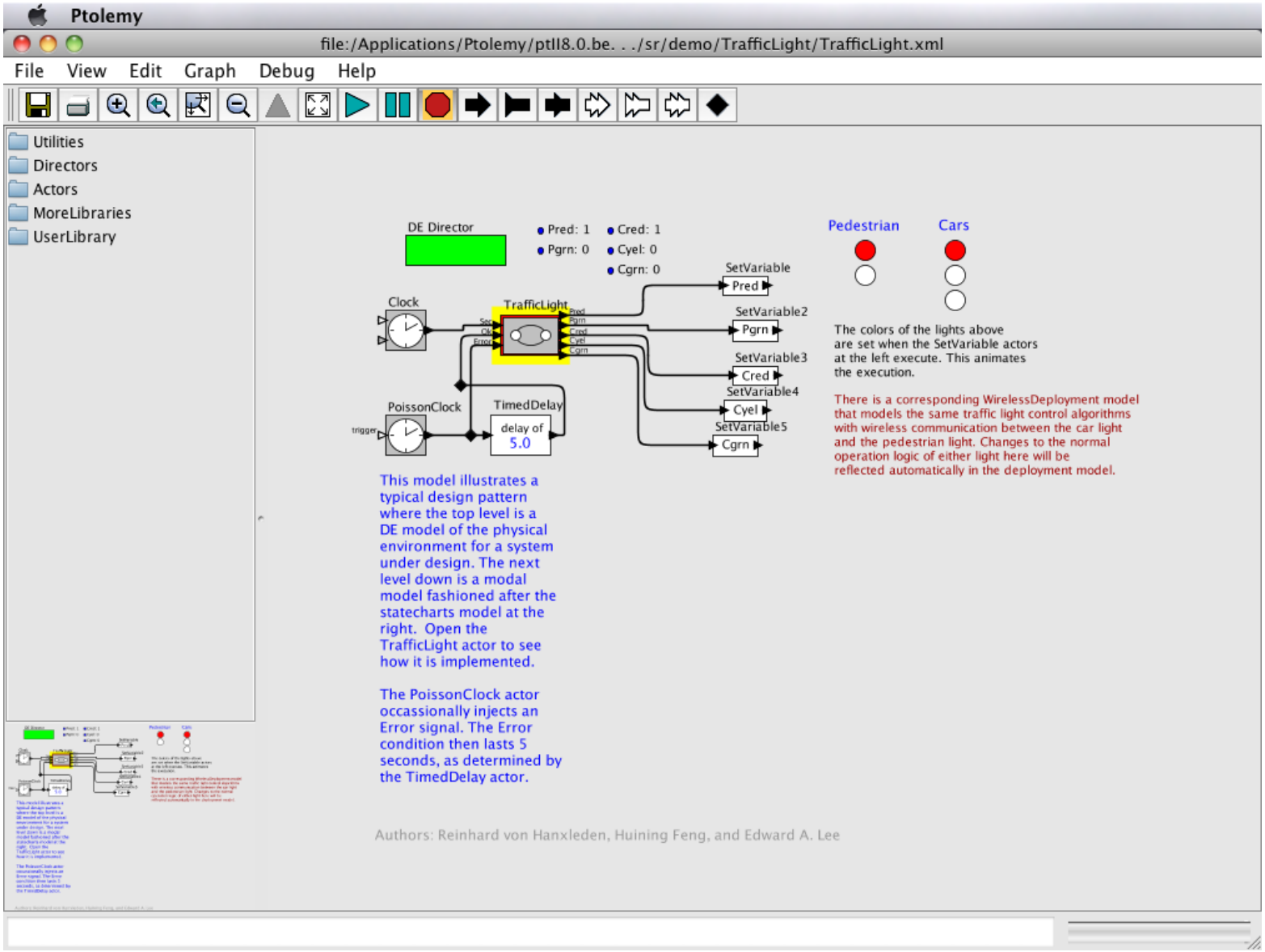
Pragmatics-Unaware Modeling

Quickly loose details



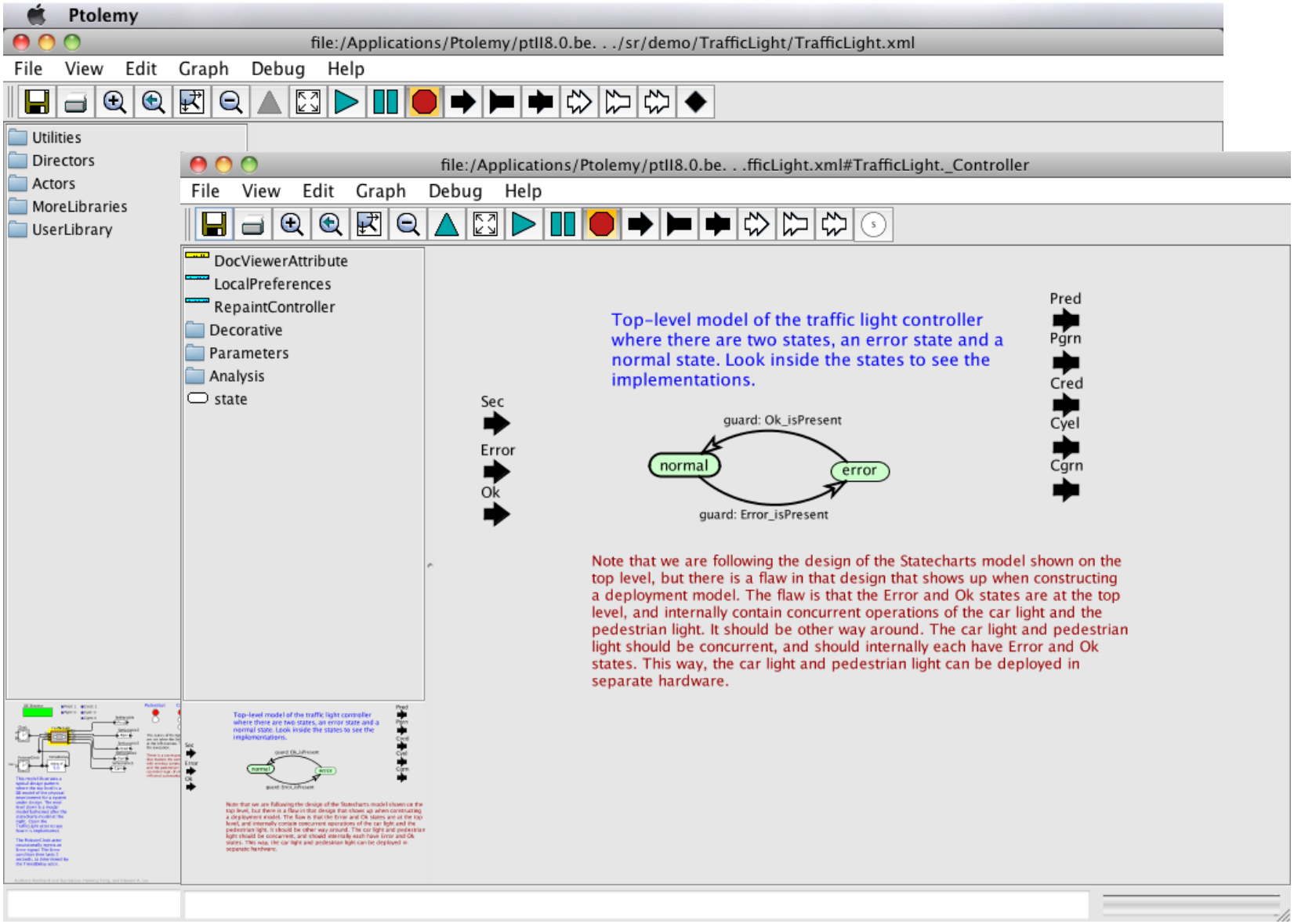
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



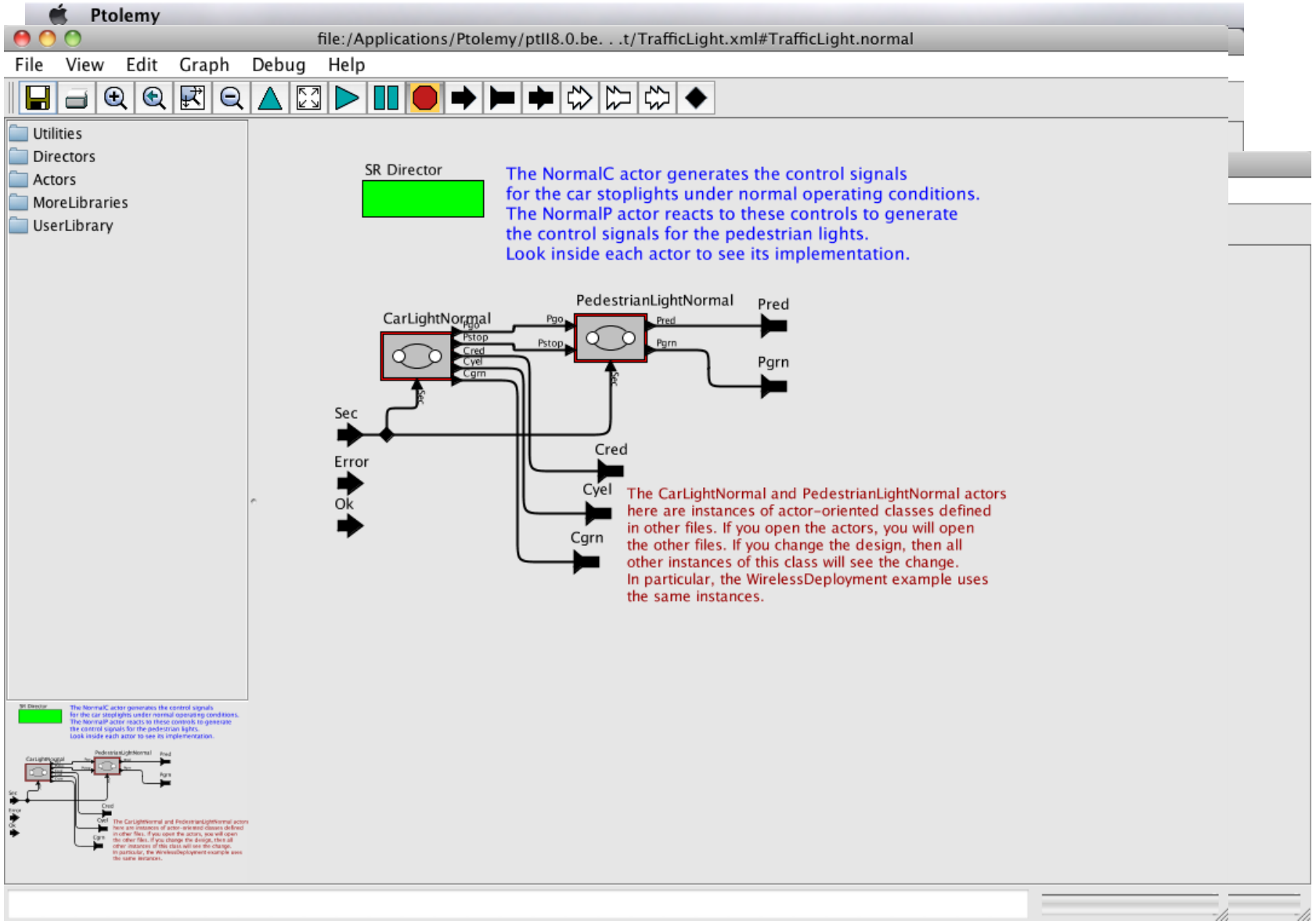
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



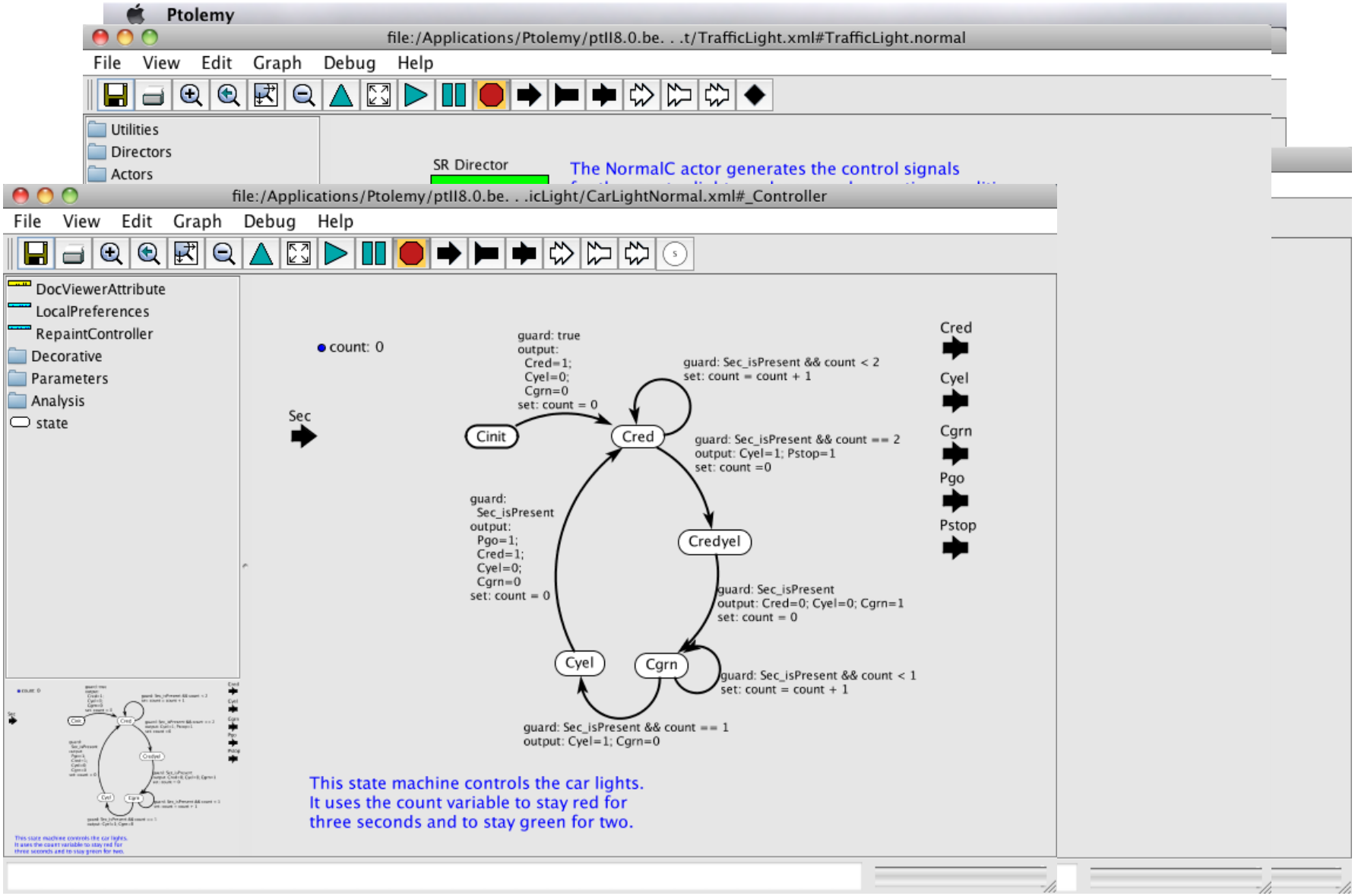
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



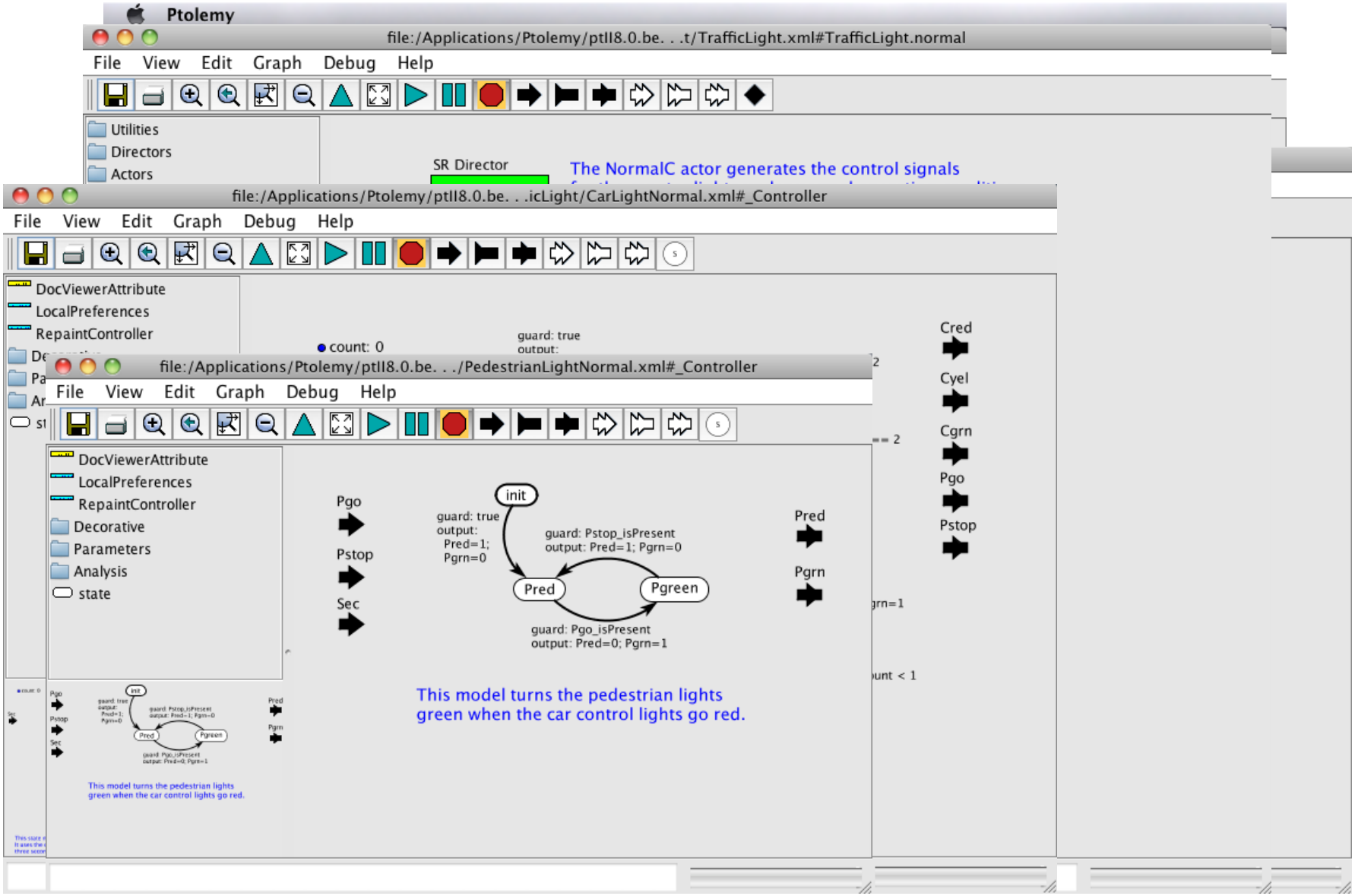
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



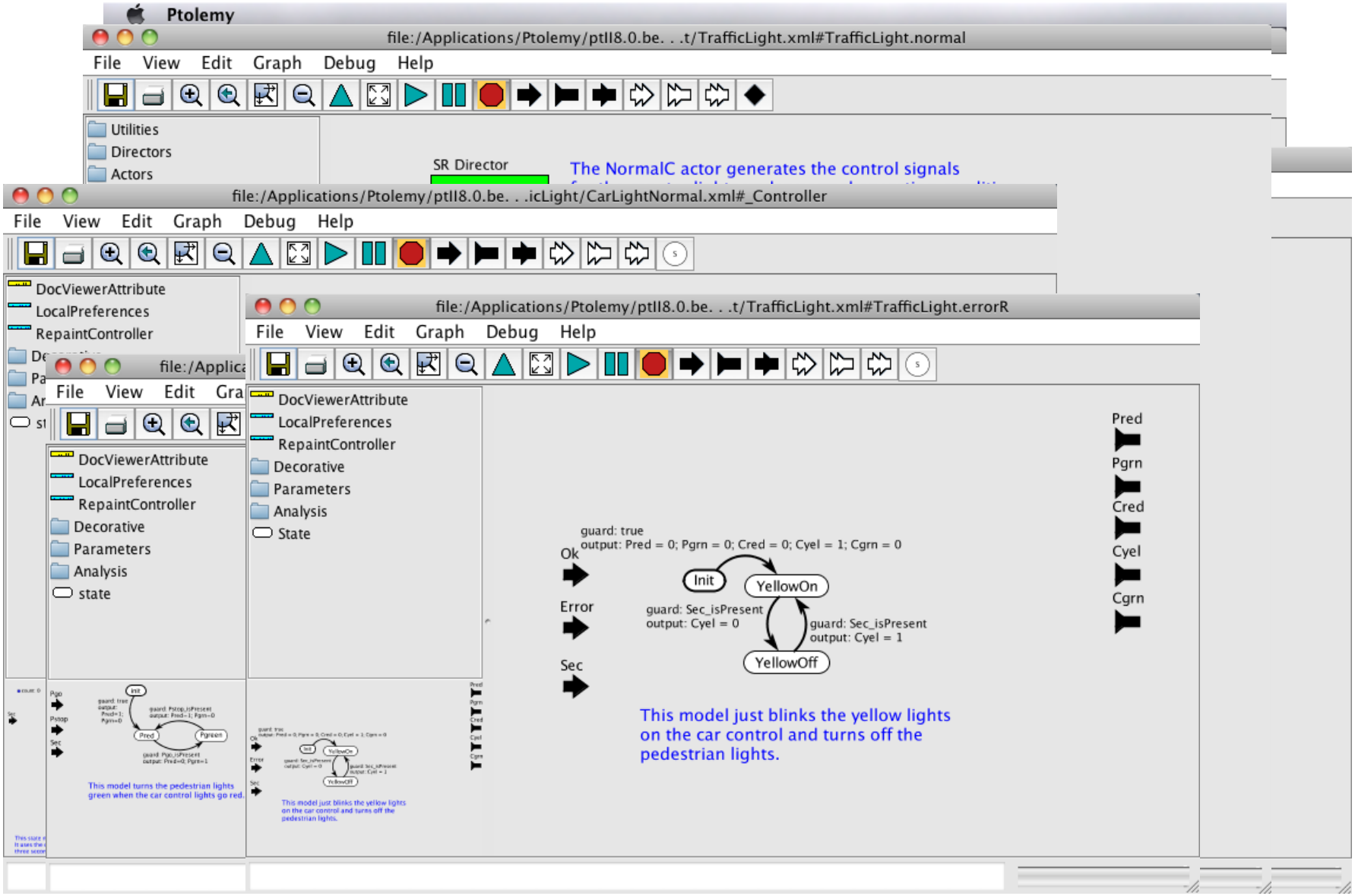
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



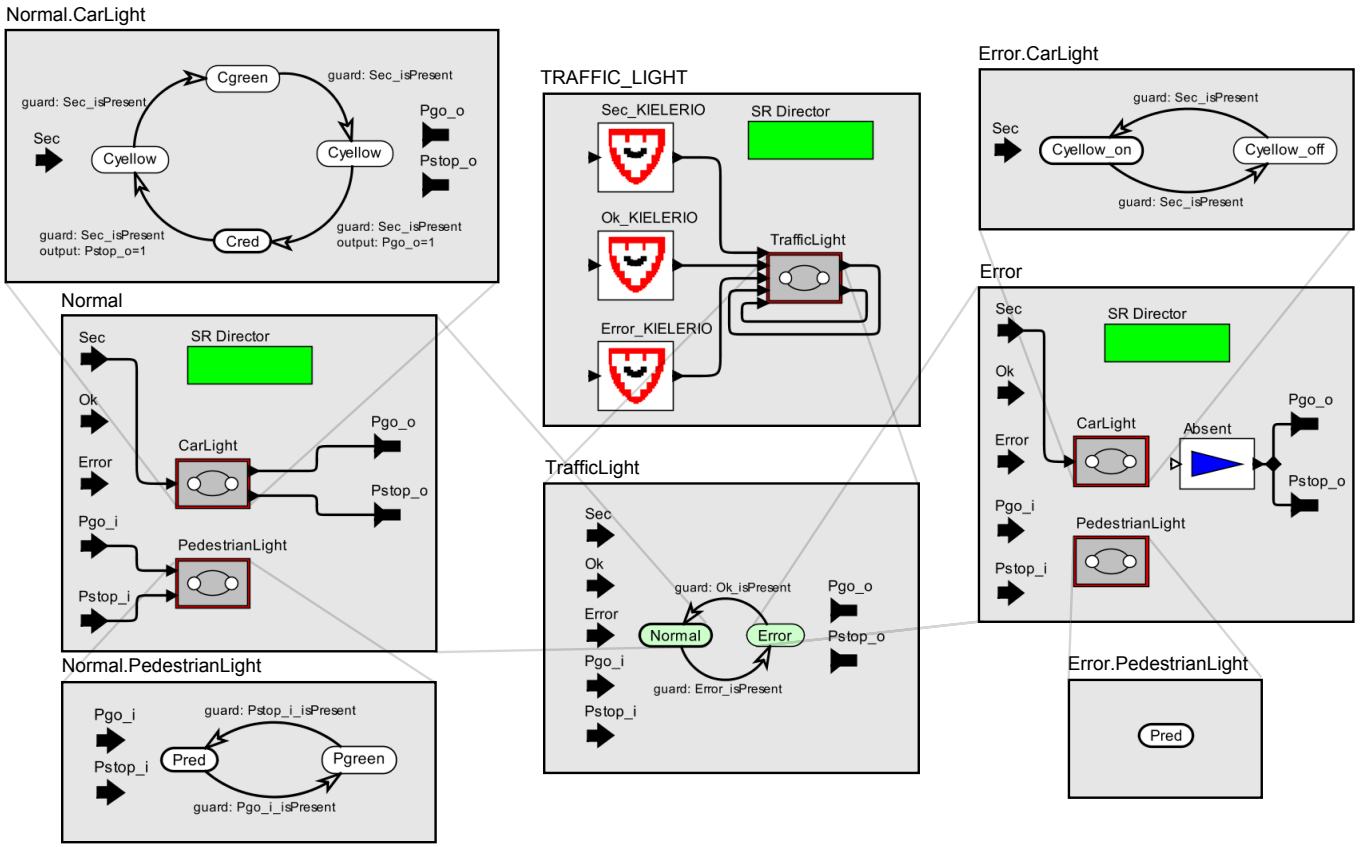
Pragmatics-Unaware Modeling

Lack of overview, tedious window management



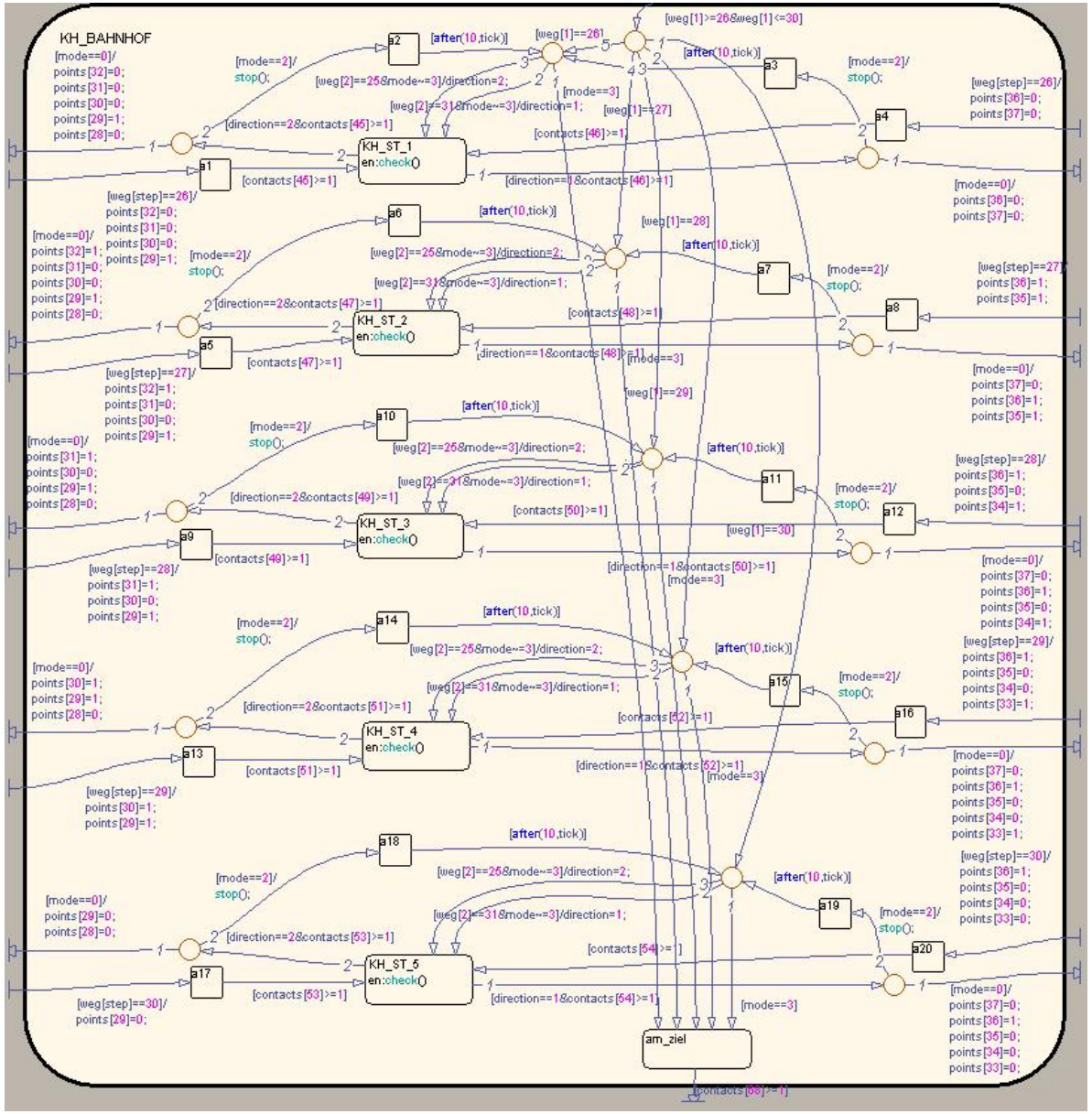
Pragmatics-Unaware Modeling

Better overview, but high manual effort



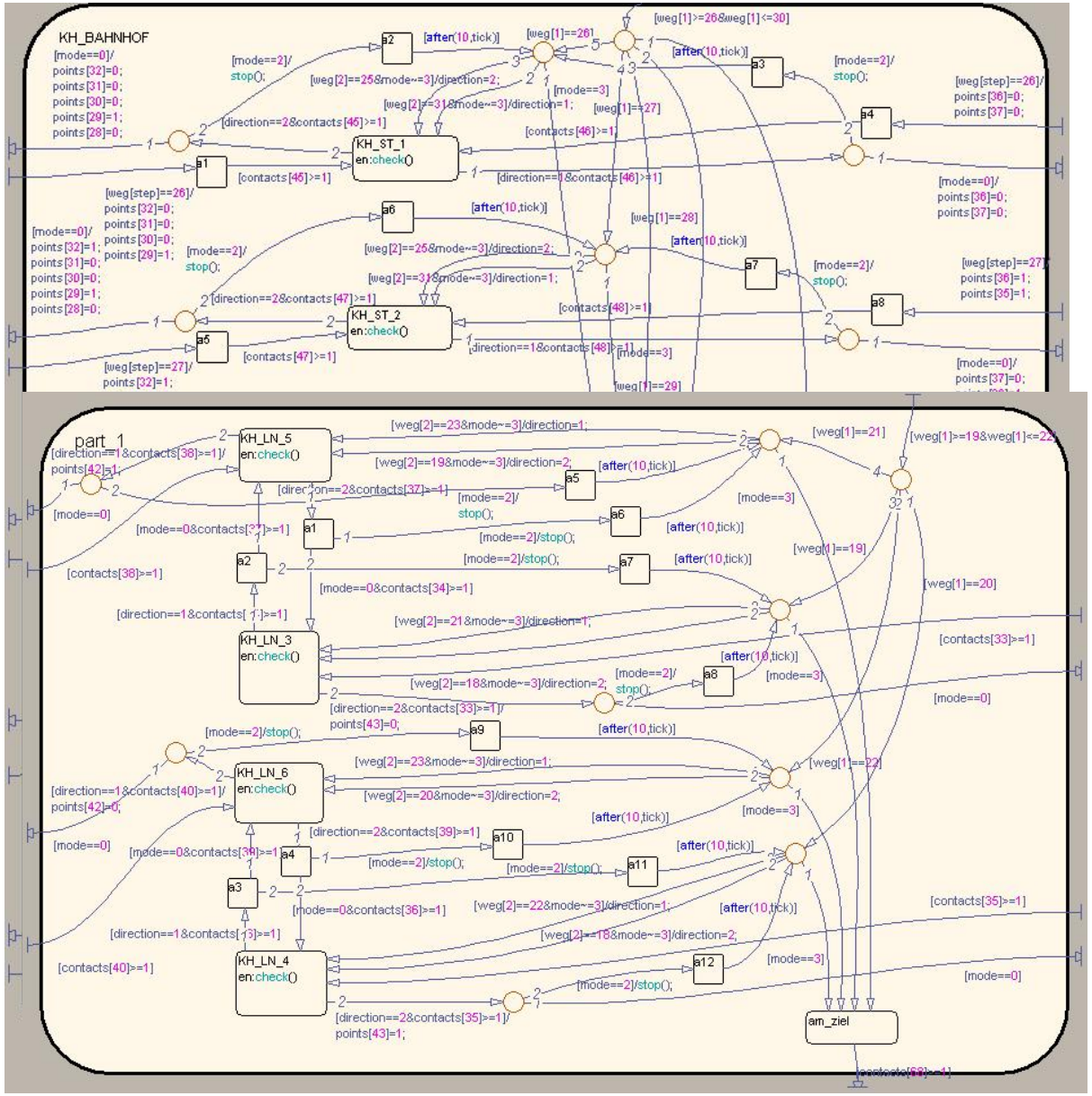
Pragmatics-Unaware Modeling

Diagrams difficult to read



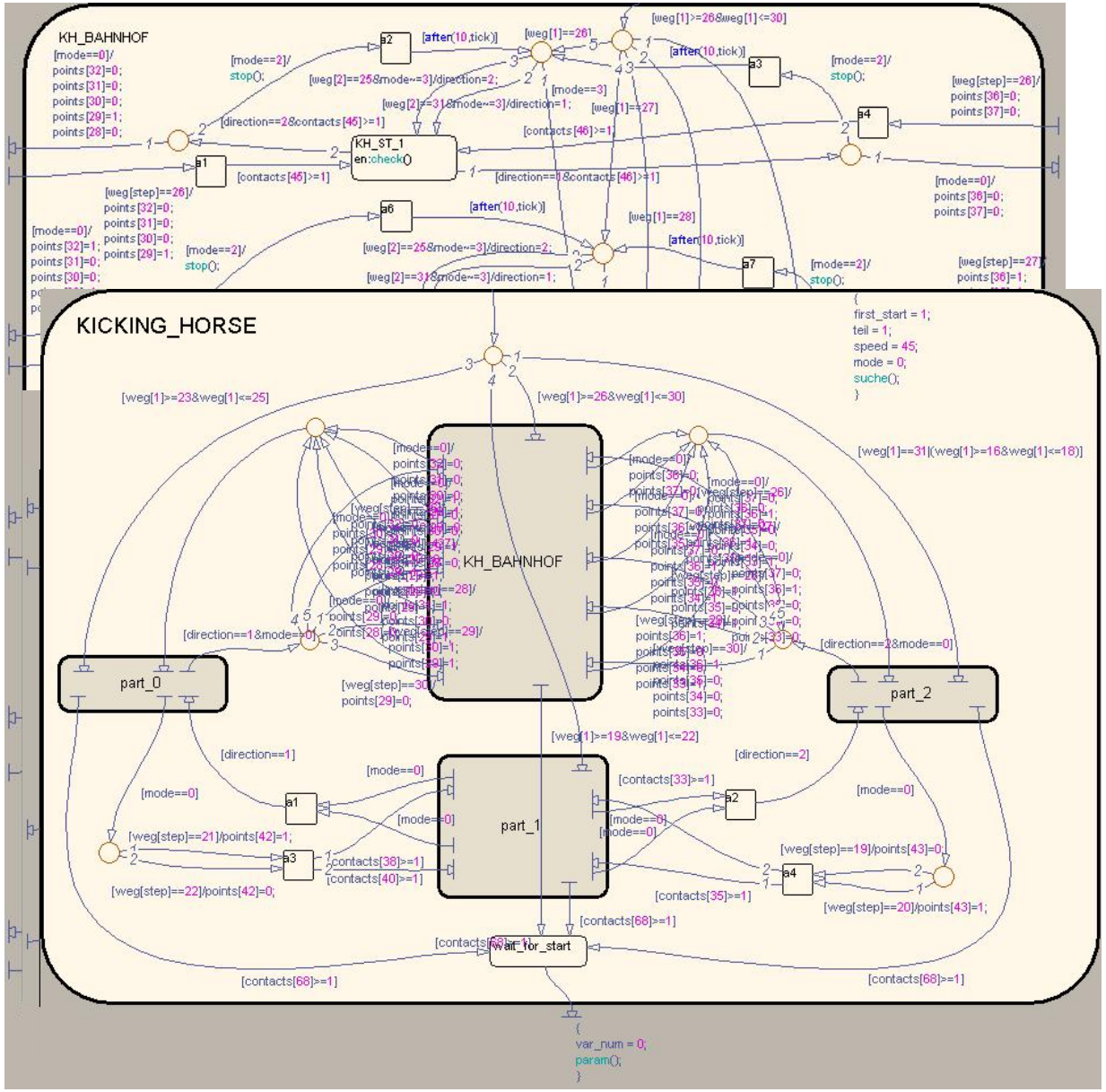
Pragmatics-Unaware Modeling

Diagrams difficult to read



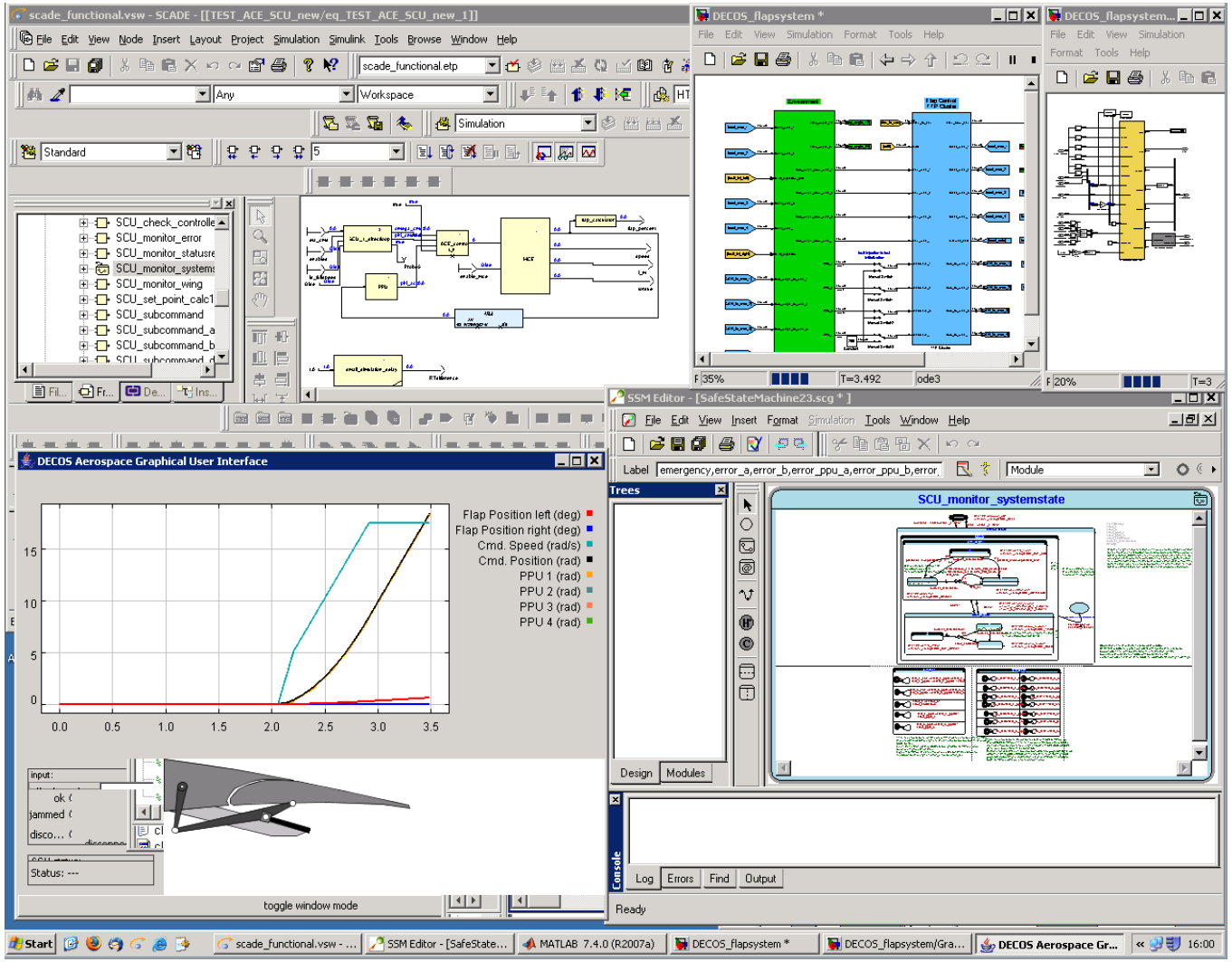
Pragmatics-Unaware Modeling

Diagrams difficult to read



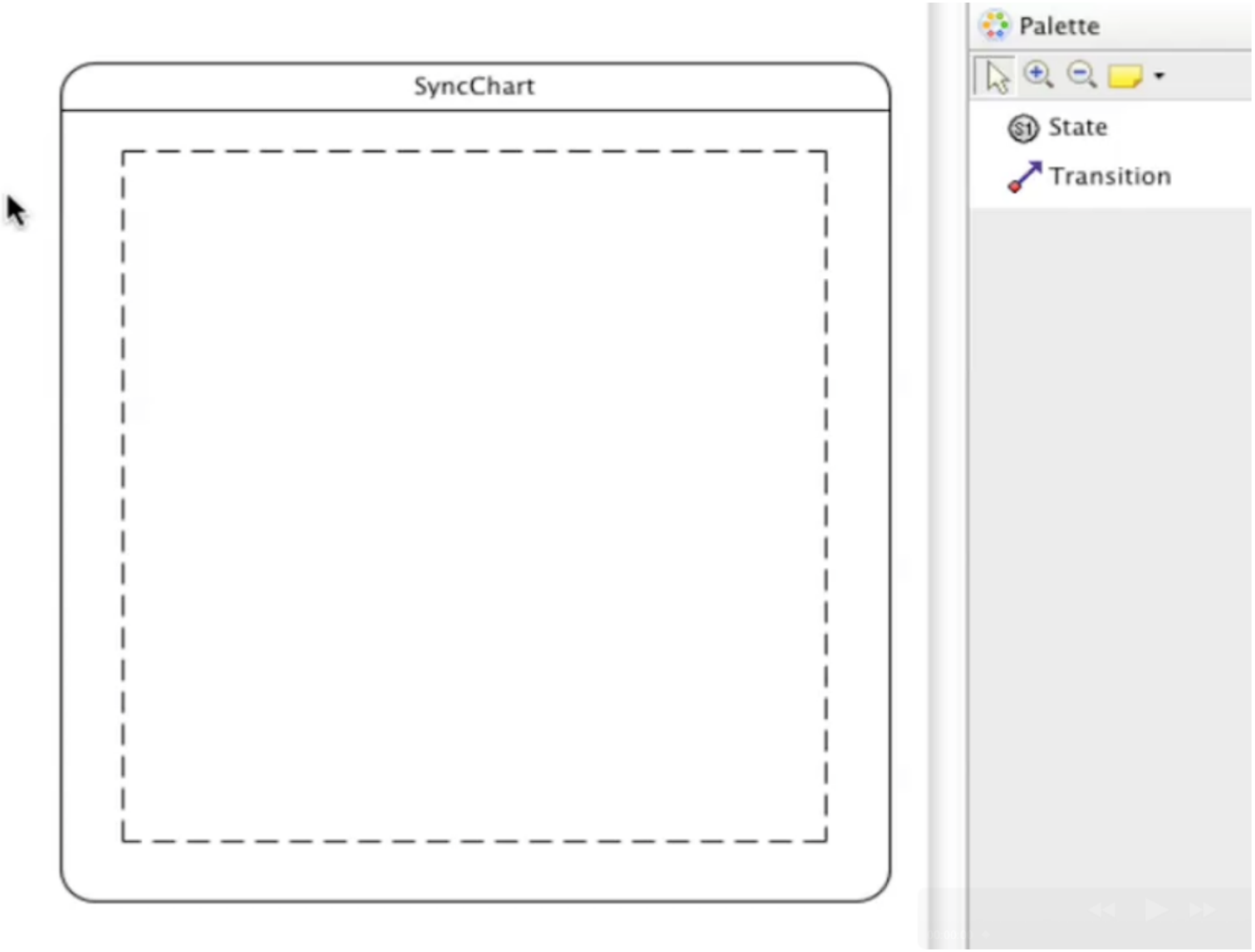
Pragmatics-Unaware Modeling

Overloaded screen real-estate



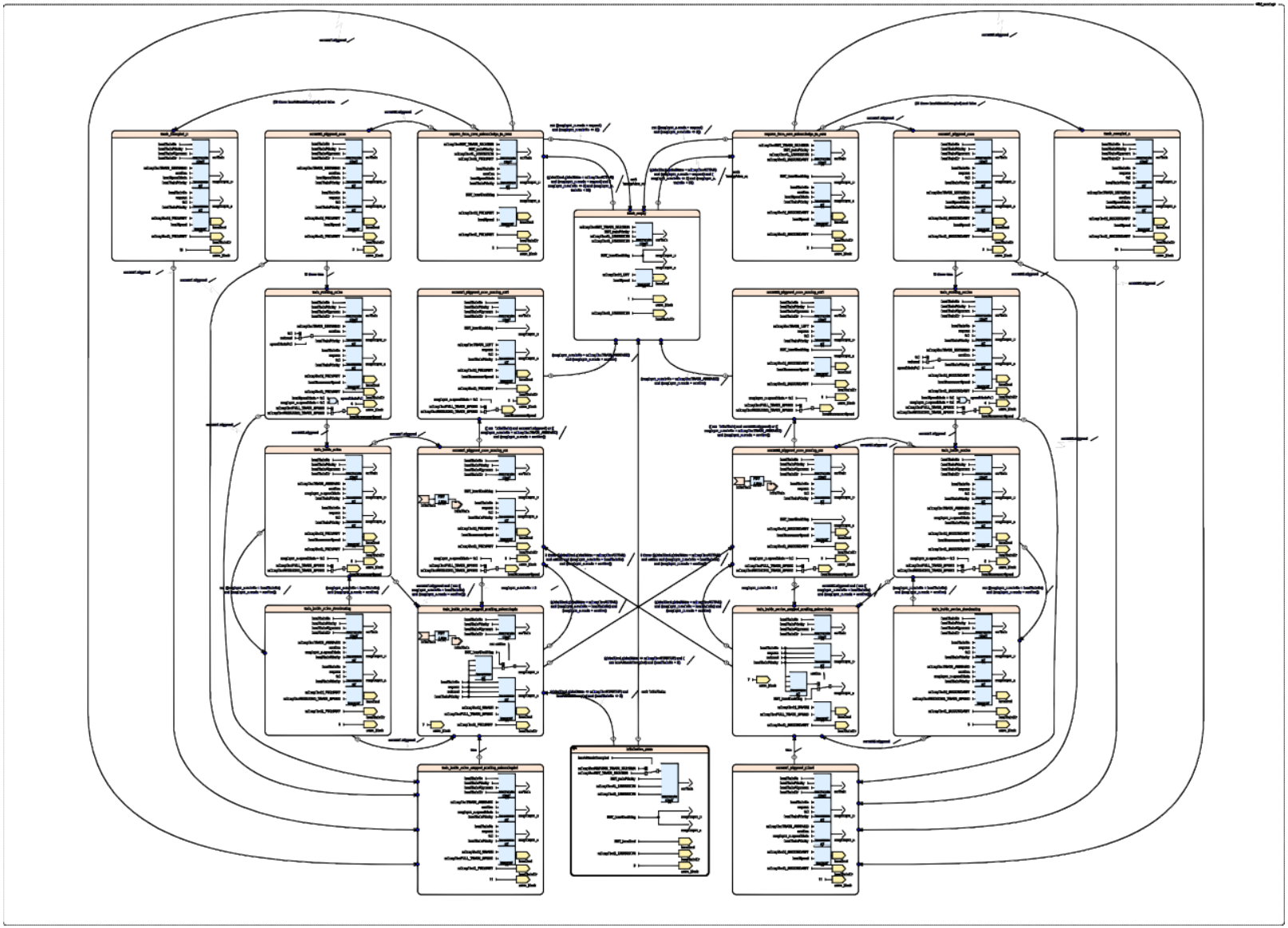
Pragmatics-Unaware Modeling

Painful editing technology



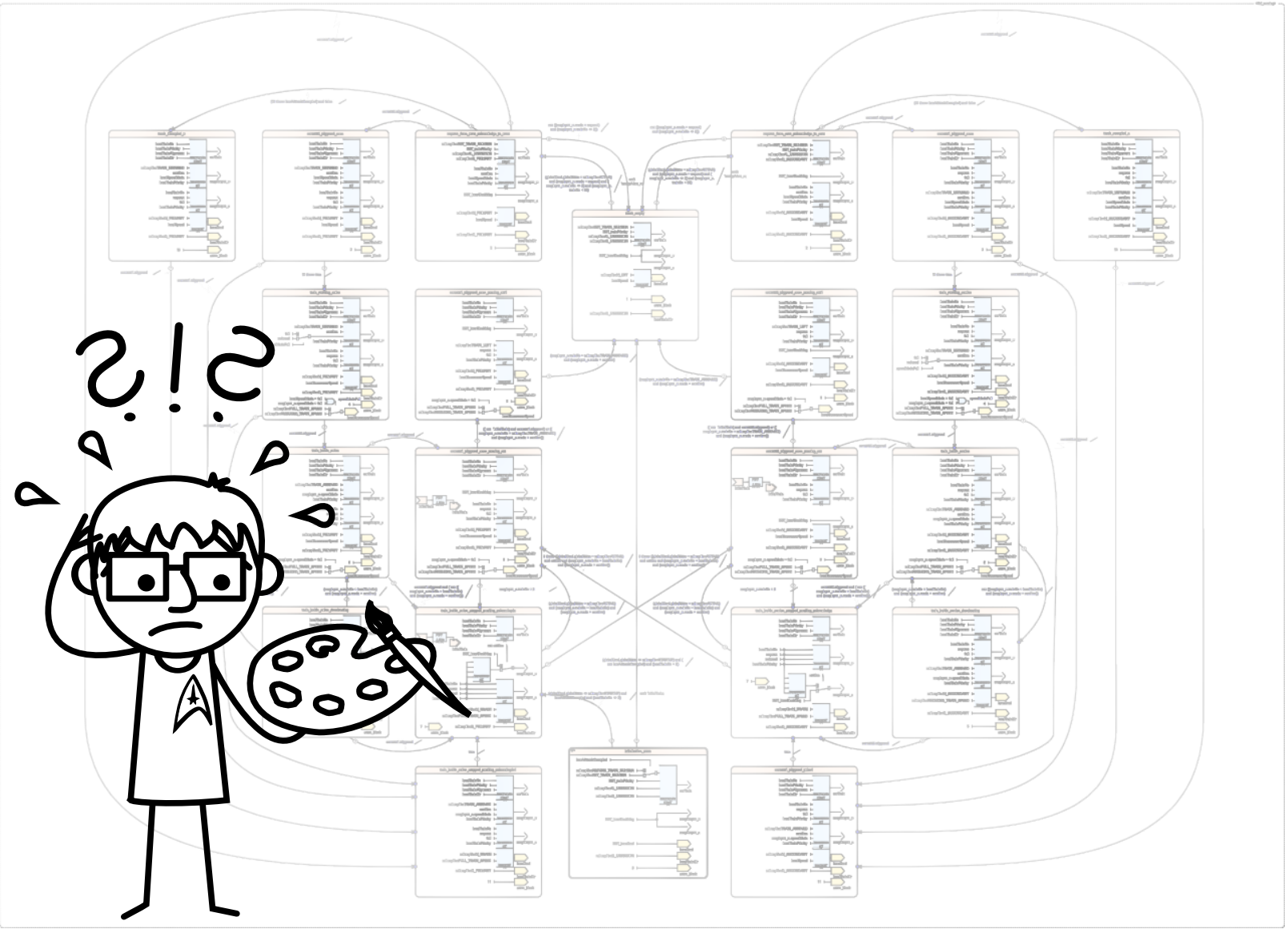
Pragmatics-Unaware Modeling

Manual effort increases significantly with model complexity



Pragmatics-Unaware Modeling

Manual effort increases significantly with model complexity



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

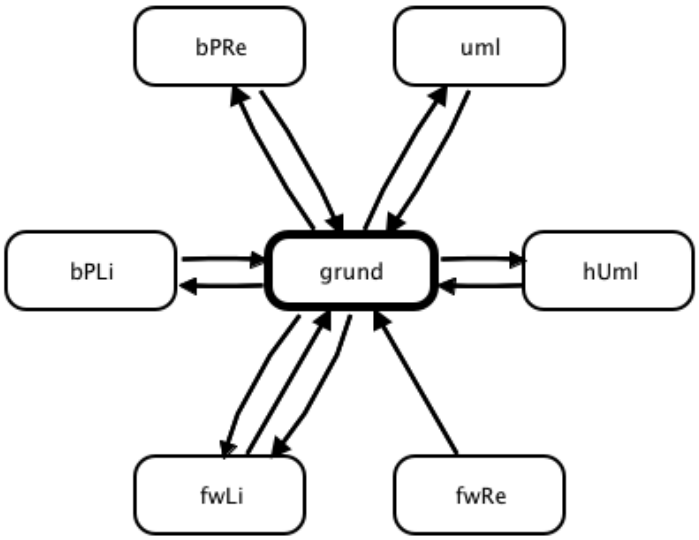
Still Want Graphical Views!

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;
```

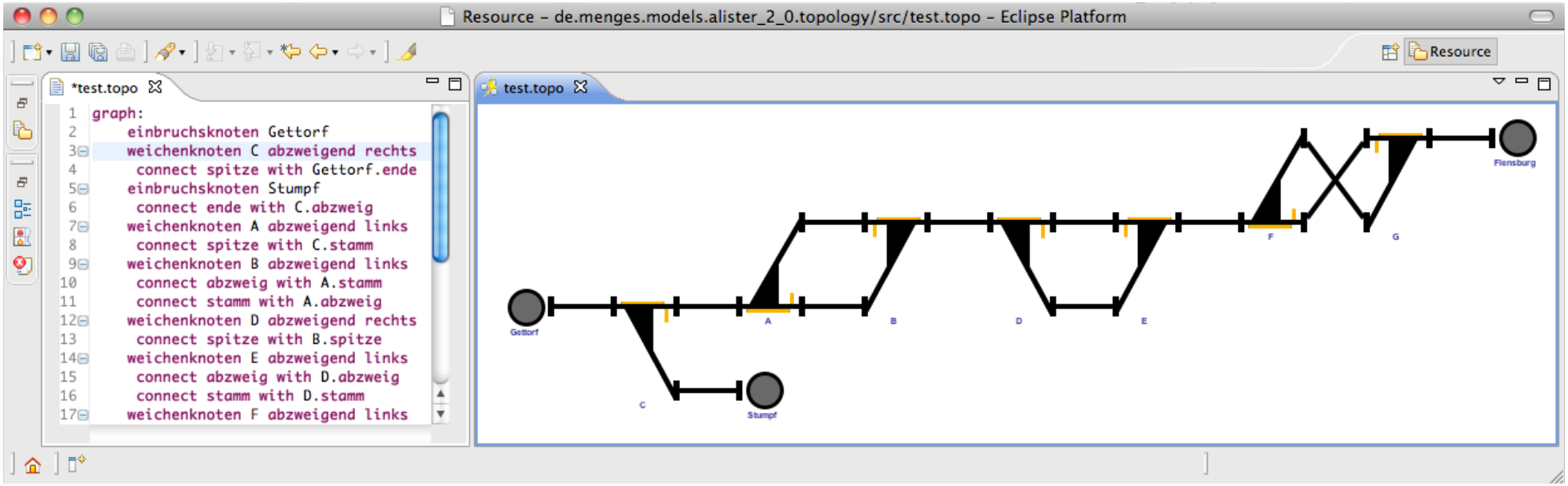
WeichenkontrolleUmstell12.logic



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

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.

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

The Model-View-Controller (MVC) Paradigm

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

The Model-View-Controller (MVC) Paradigm

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

Views 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*.

The Model-View-Controller (MVC) Paradigm

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

Views 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*.

Controllers 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.

The Model-View-Controller (MVC) Paradigm

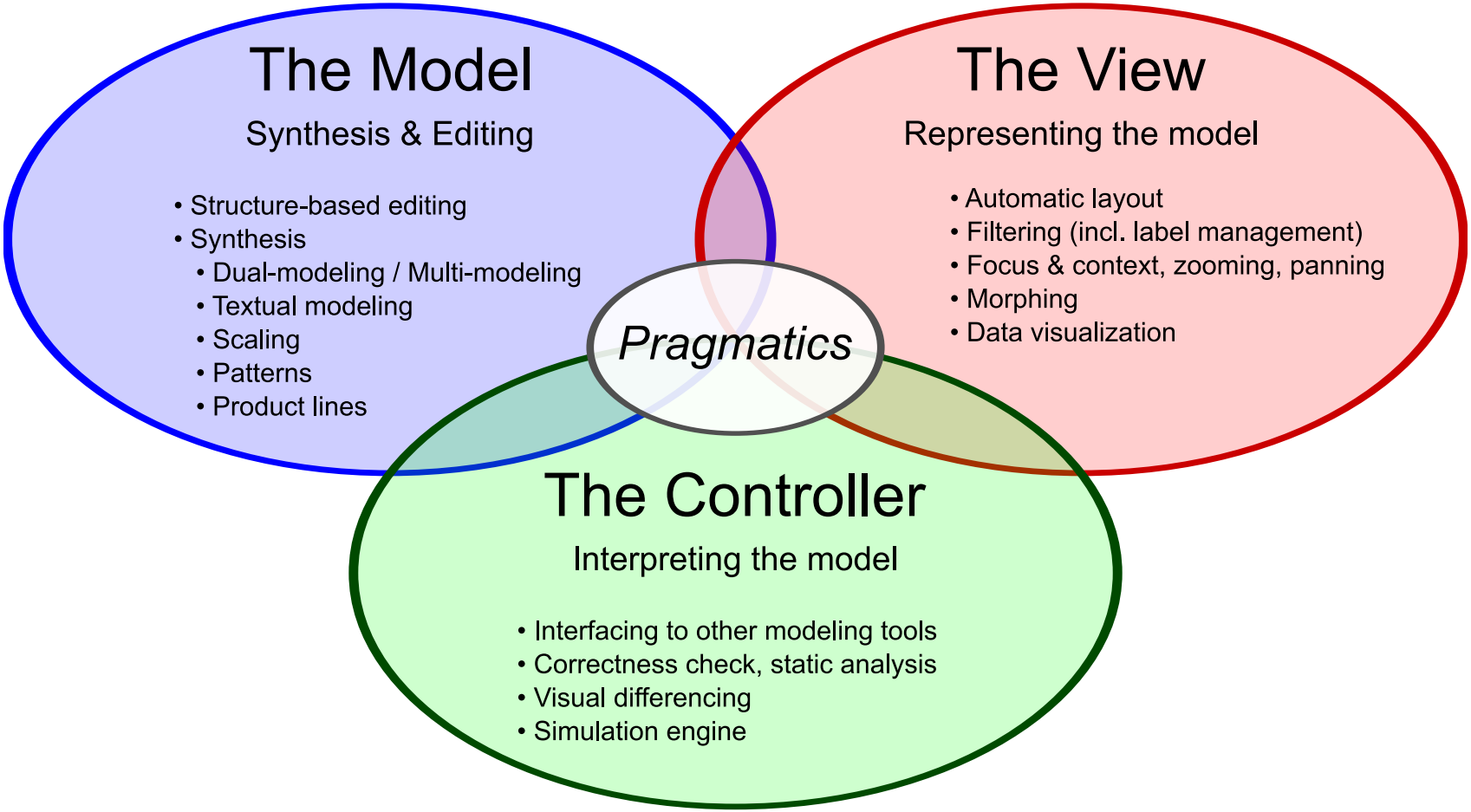
Models Models represent knowledge. A model could be a single object (rather uninteresting), or it could be some structure of objects.

Views 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*.

Controllers 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)

A Key to Pragmatics: MVC



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

Overview

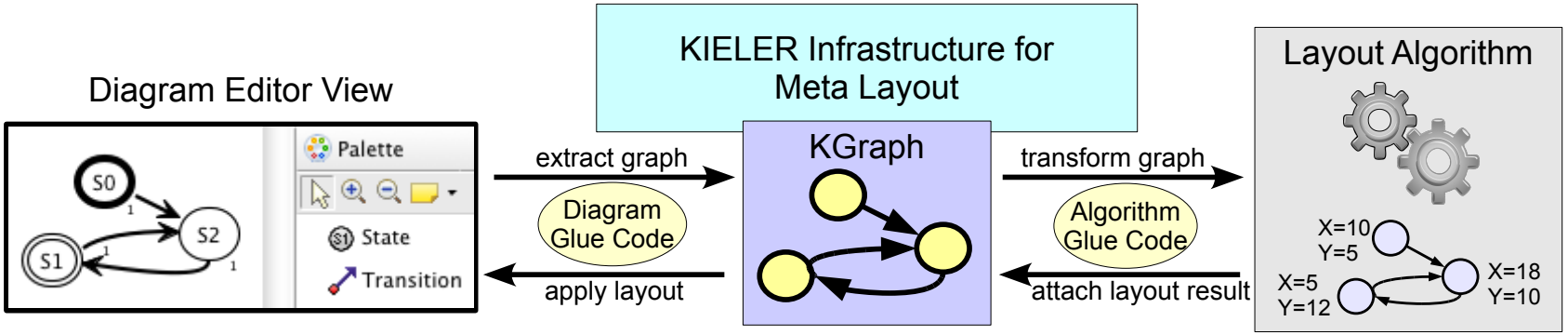
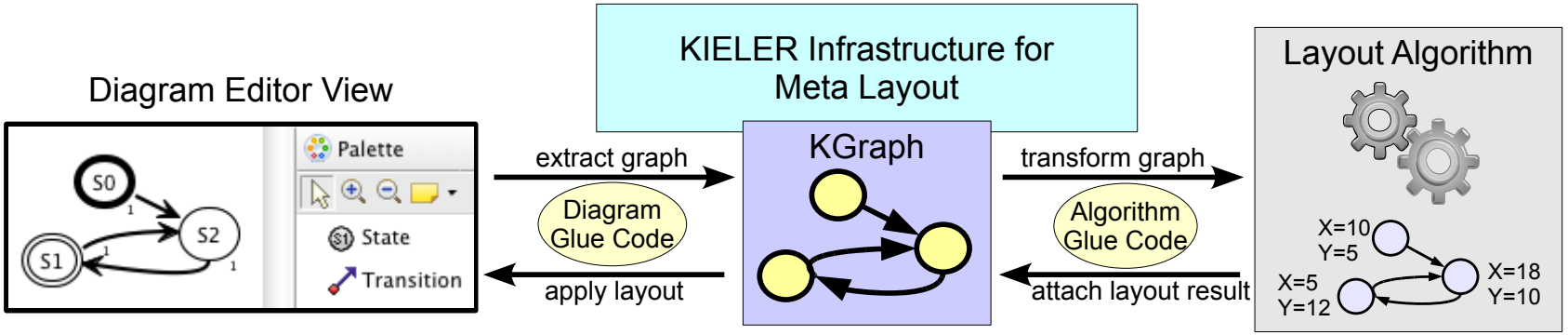
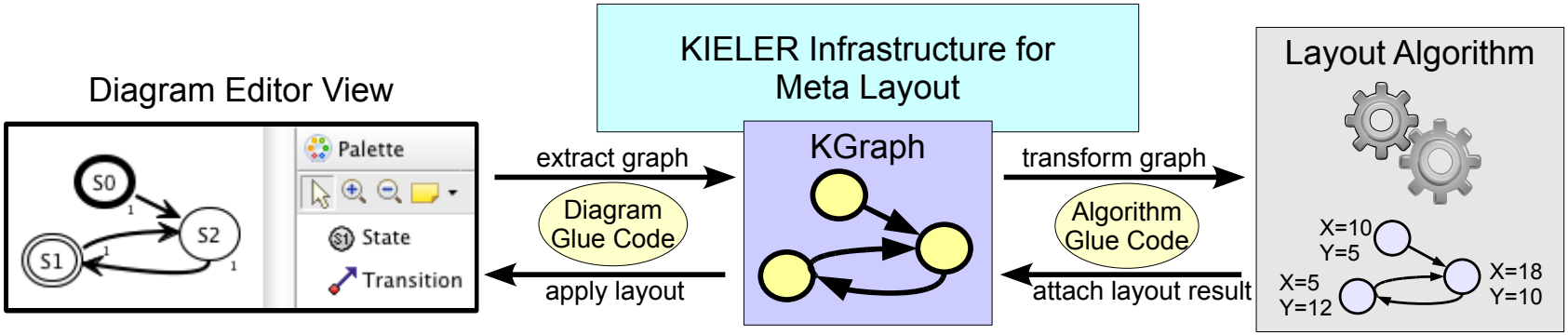


Diagram Editor View
S0
S1
S2
Palette
State
Transition
extract graph
Diagram Glue Code
apply layout
KIELER Infrastructure for Meta Layout
KGraph
transform graph
Algorithm Glue Code
attach layout result
Layout Algorithm
X=10
Y=5
X=5
Y=12
X=18
Y=10



- 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)



- 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)

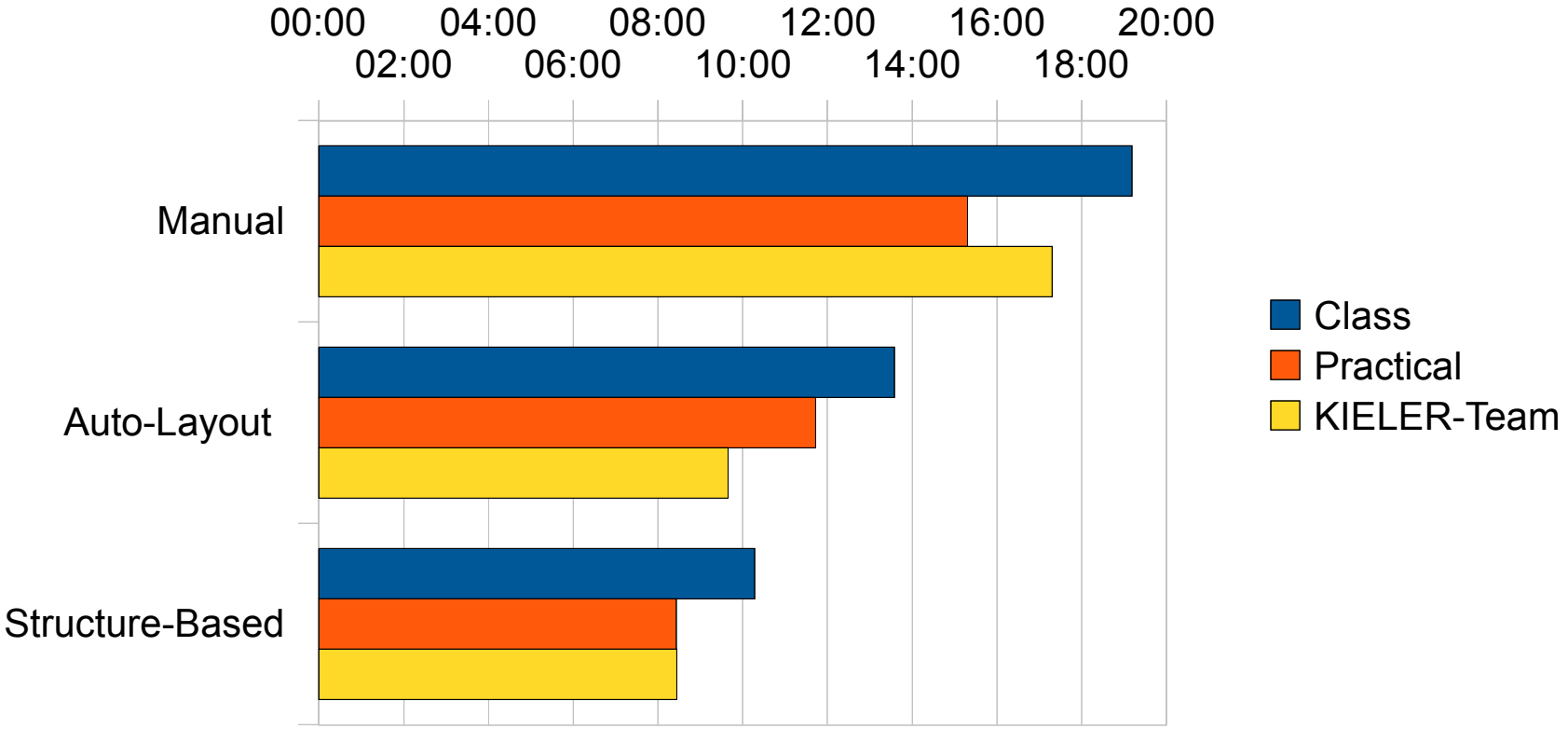
Schulze, Spönemann, v. Hanxleden
Drawing Layered Graphs with Port Constraints
JVLC'14

Editing Efficiency: An Experiment

Task: create diagram from textual specification

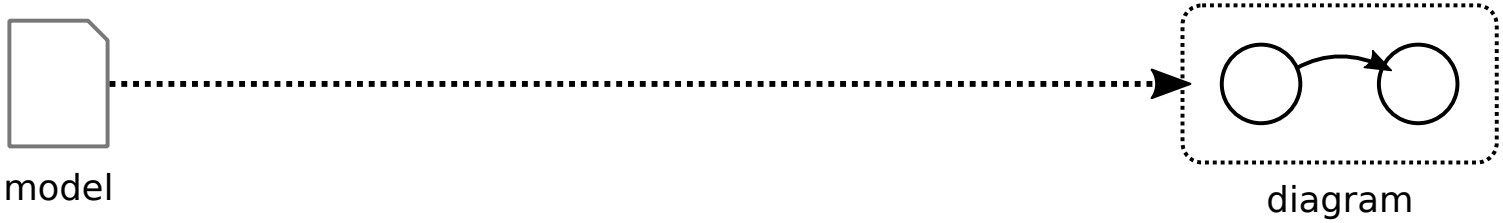
Editing Efficiency: An Experiment

Task: create diagram from textual specification



Fuhrmann, von Hanxleden
Taming Graphical Modeling
MODELS'10

KIELER Lightweight Diagrams (KLighD)

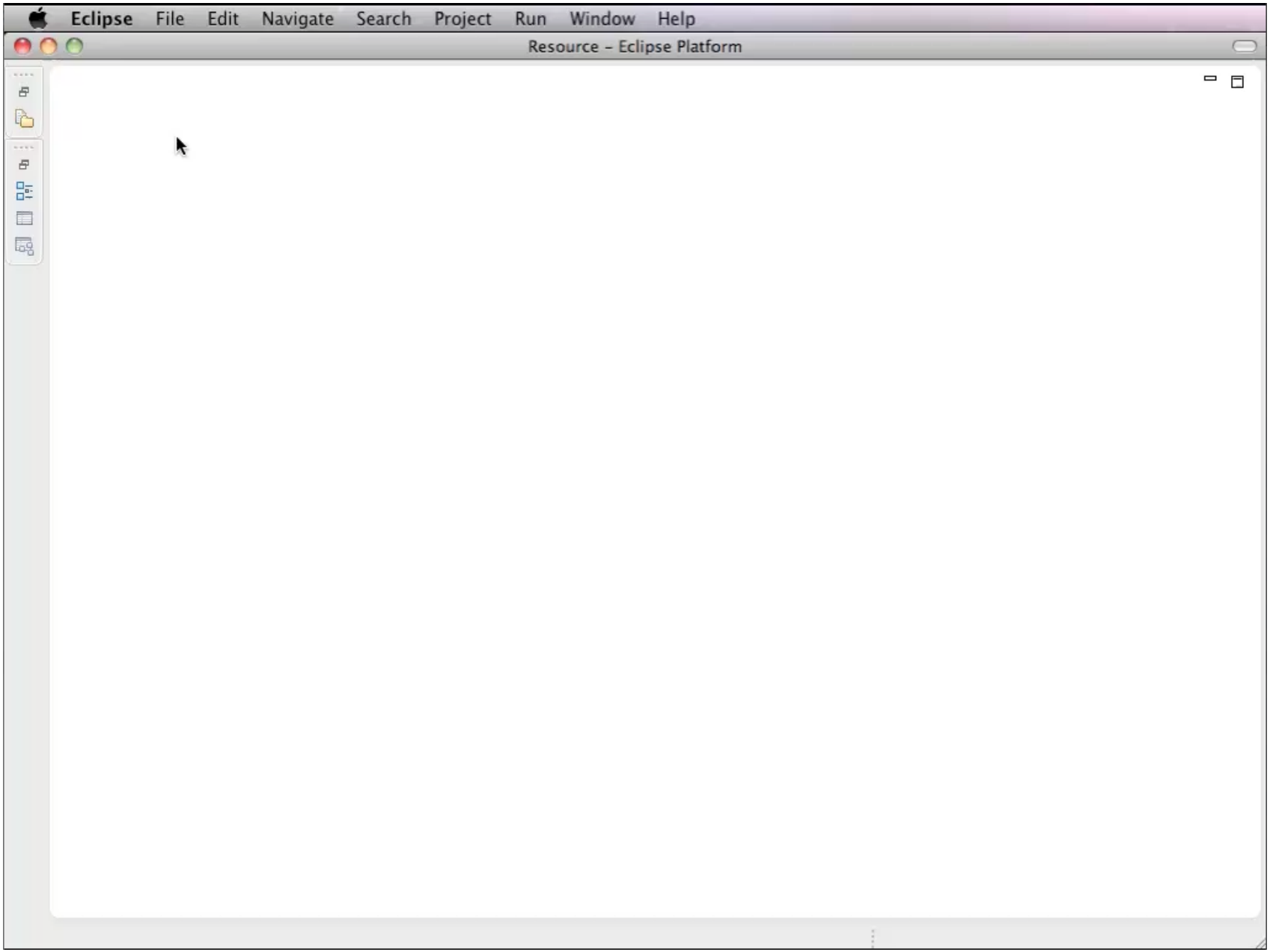


KIELER Lightweight Diagrams (KLighD)

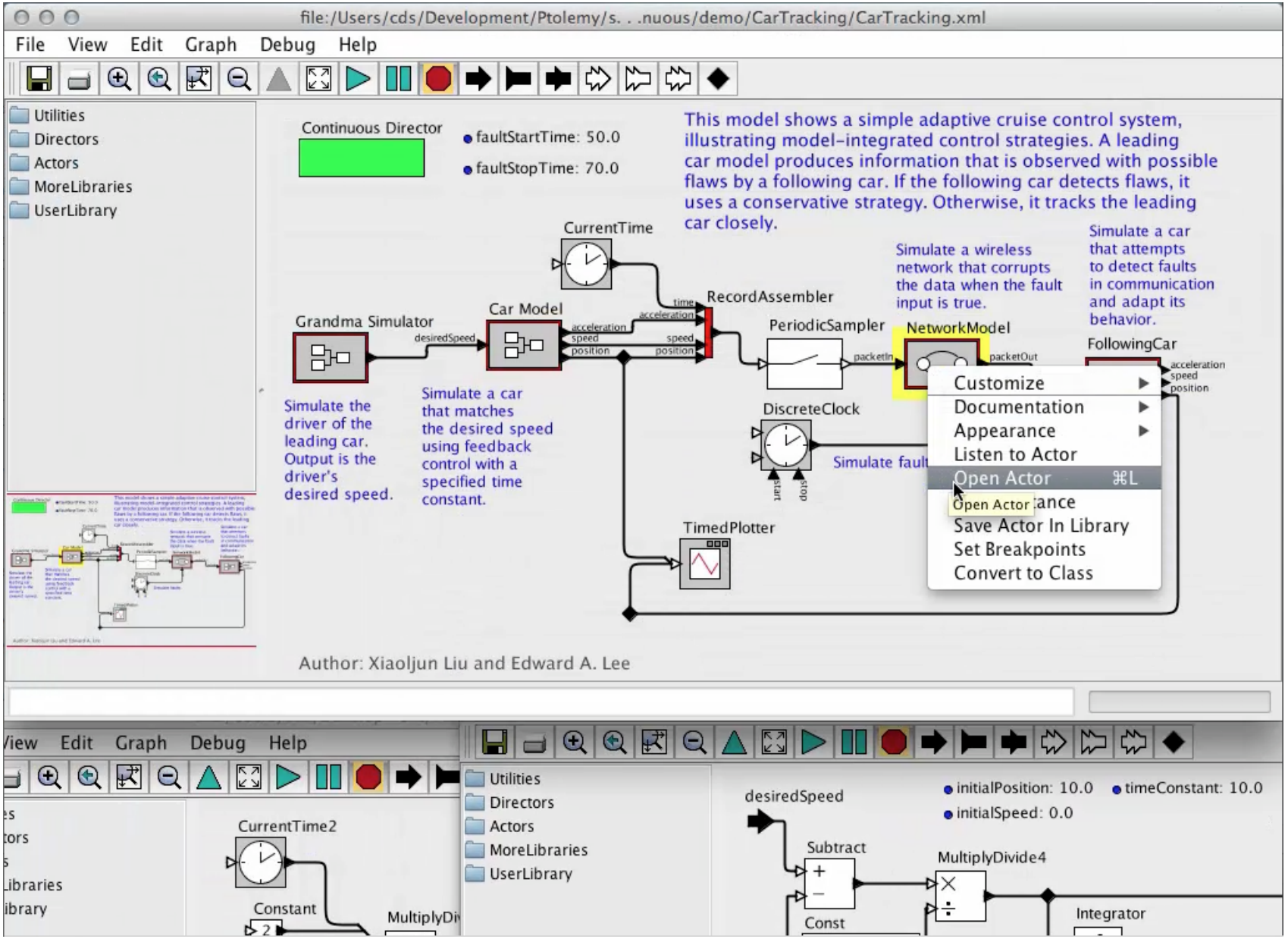
```
state machine {
  input signal POWER
  region:
    init state Off
    --> On with POWER;
    state On;
};
```

diagram

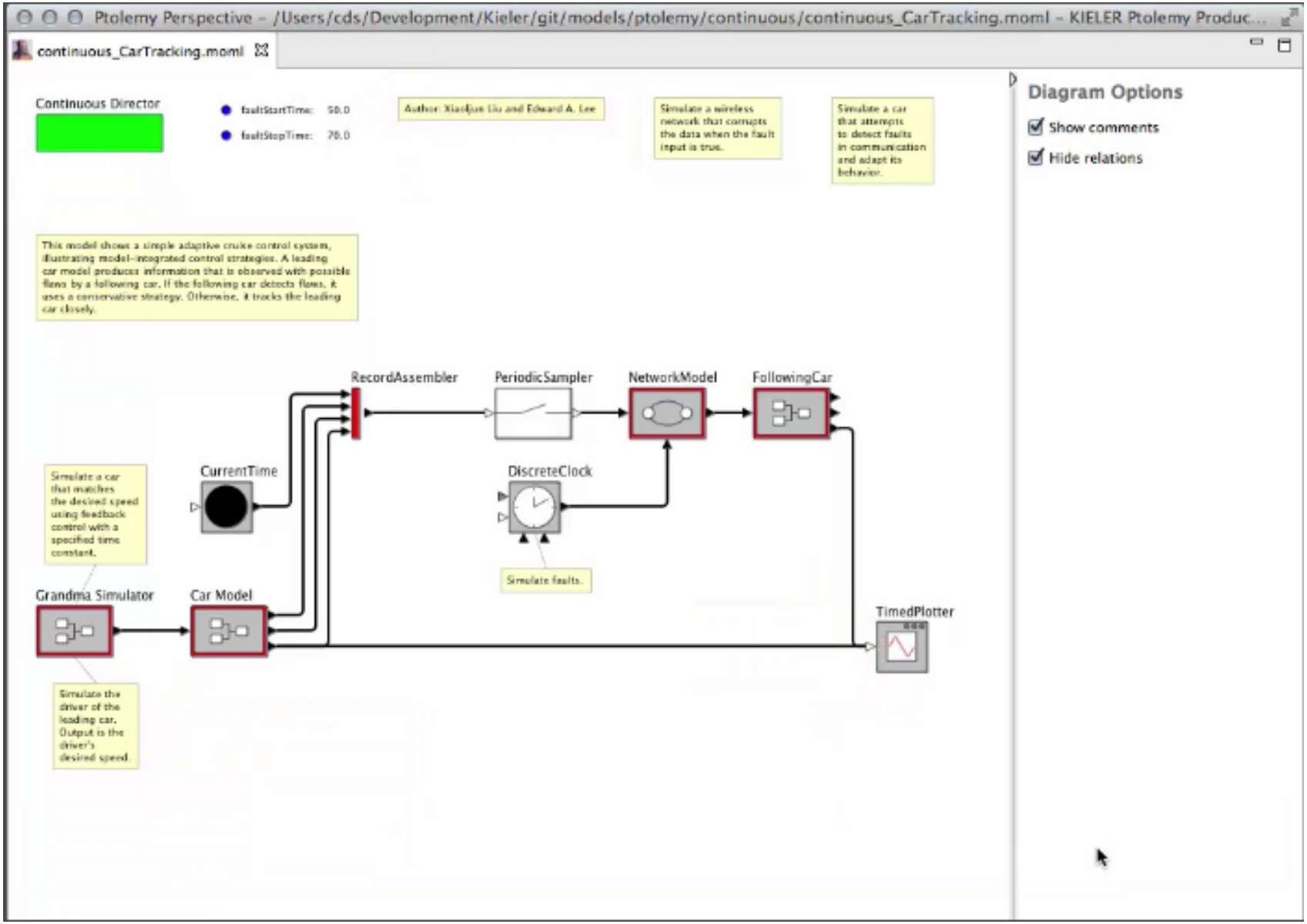
KIELER Lightweight Diagrams (KLighD)


Eclipse
File
Edit
Navigate
Search
Project
Run
Window
Help
Resource - Eclipse Platform

Ptolemy (UC Berkeley)



Ptolemy Browser (Based on KLighD)

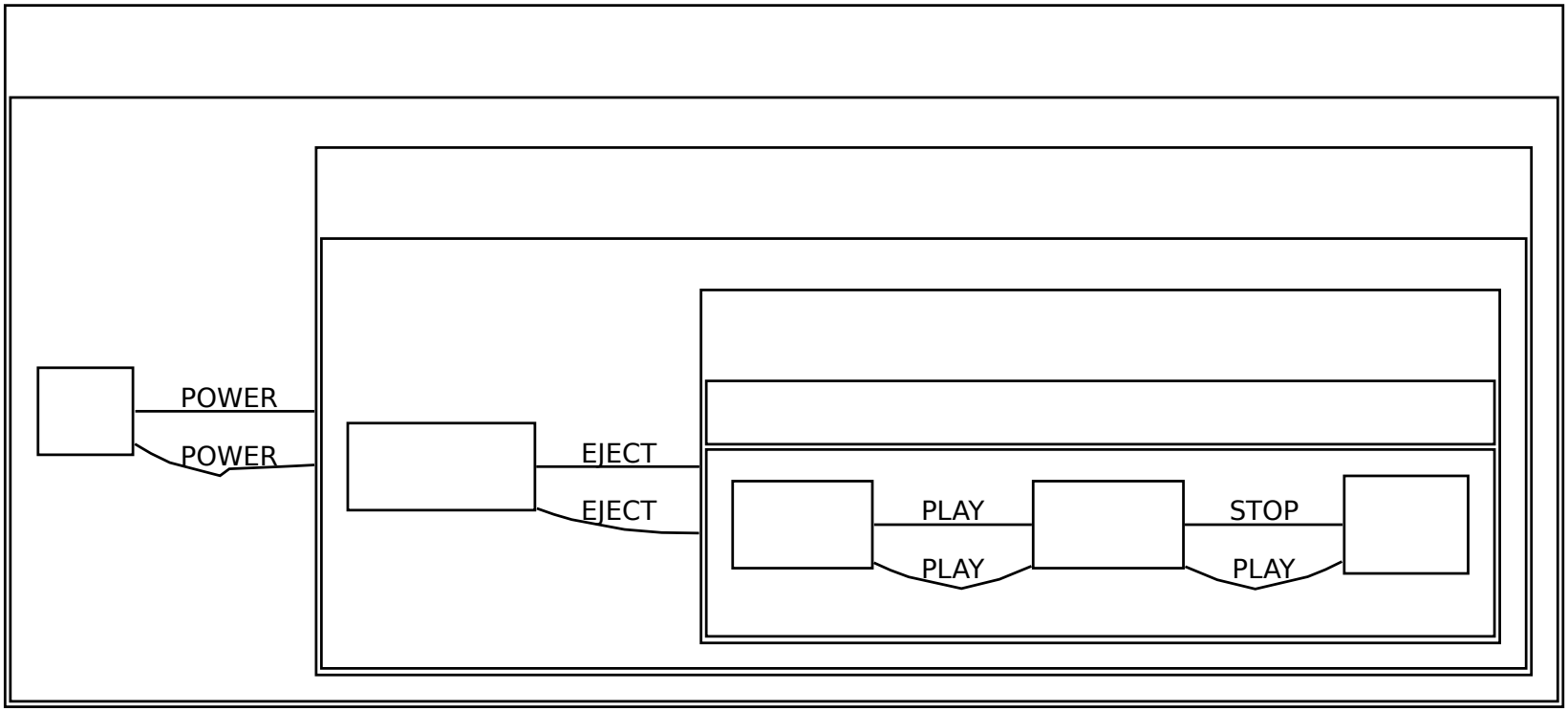


Lightweight Diagrams: Building blocks

- Automatic arrangement of depicted elements – macro layout
- Local arrangement of the figures' primitives – micro layout
- A description language to formulate diagrams smartly

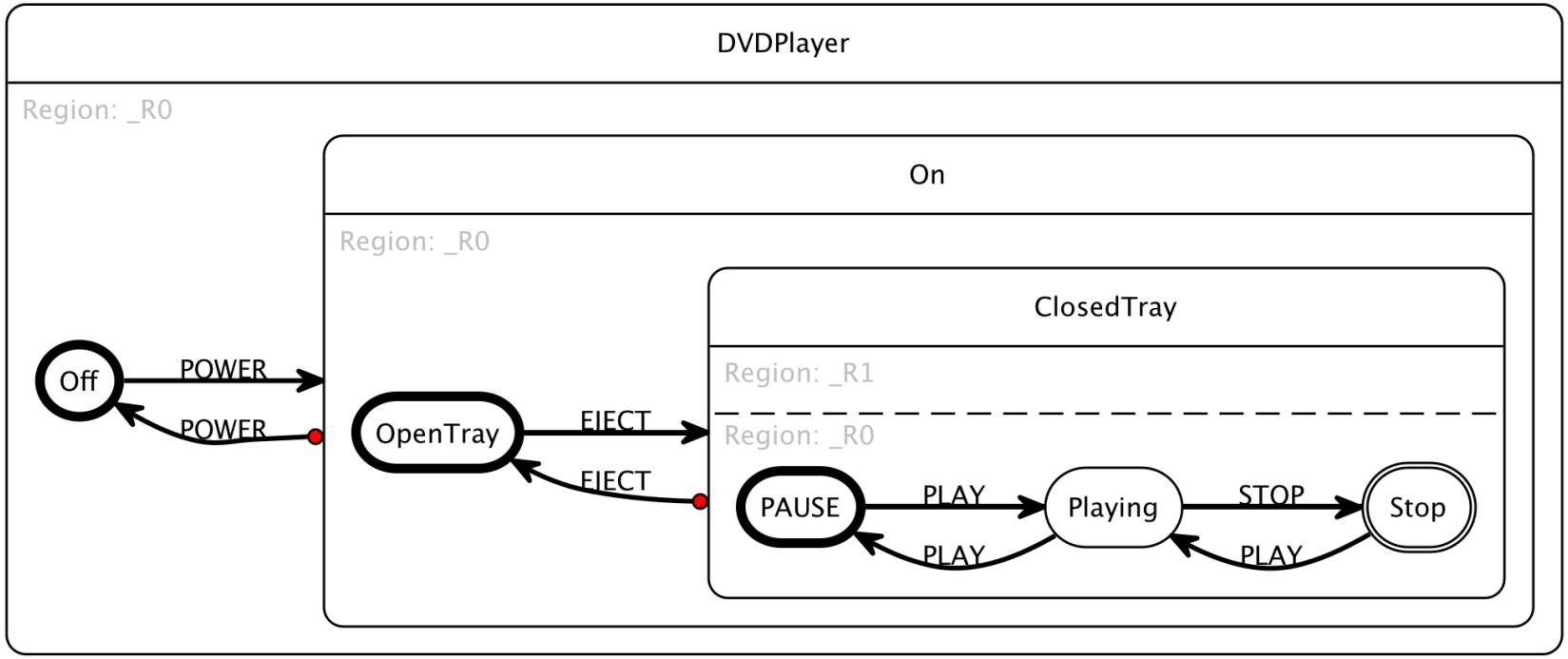
Lightweight Diagrams: Building blocks

- Automatic arrangement of depicted elements – macro layout
- Local arrangement of the figures' primitives – micro layout
- A description language to formulate diagrams smartly



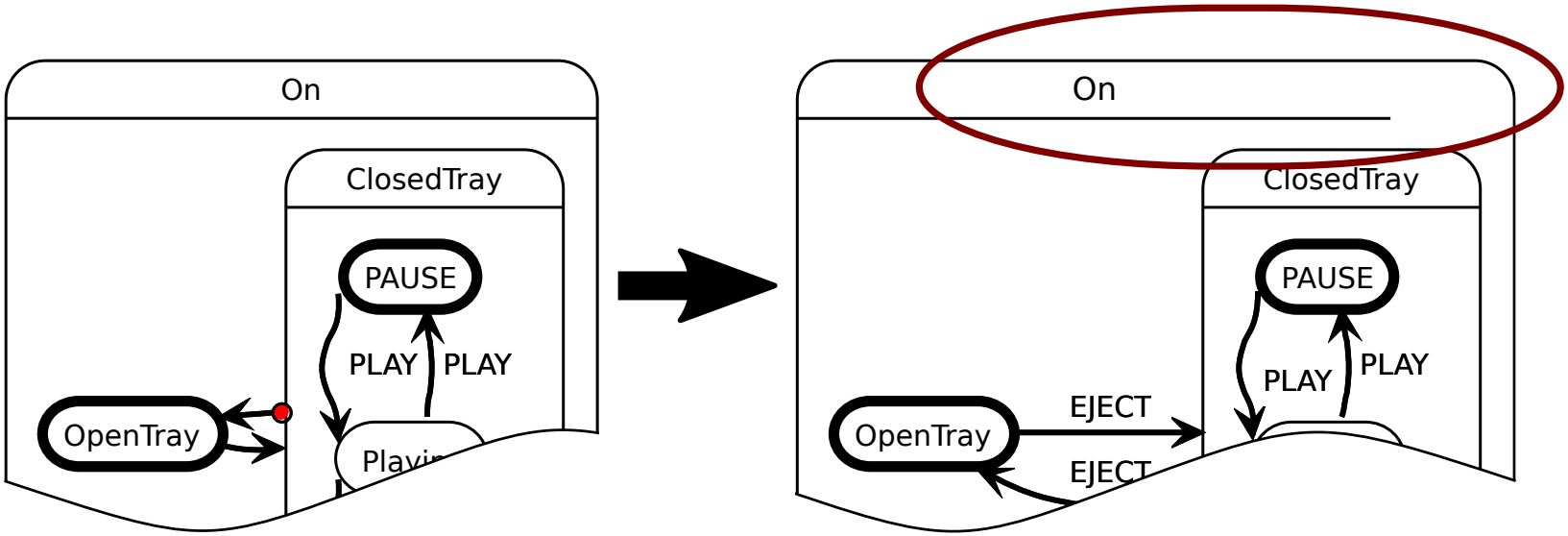
Lightweight Diagrams: Building blocks

- Automatic arrangement of depicted elements – macro layout
- Local arrangement of the figures' primitives – micro layout
- A description language to formulate diagrams smartly



Lightweight Diagrams: Building blocks

- Automatic arrangement of depicted elements – macro layout
- Local arrangement of the figures' primitives – micro layout
- A description language to formulate diagrams smartly



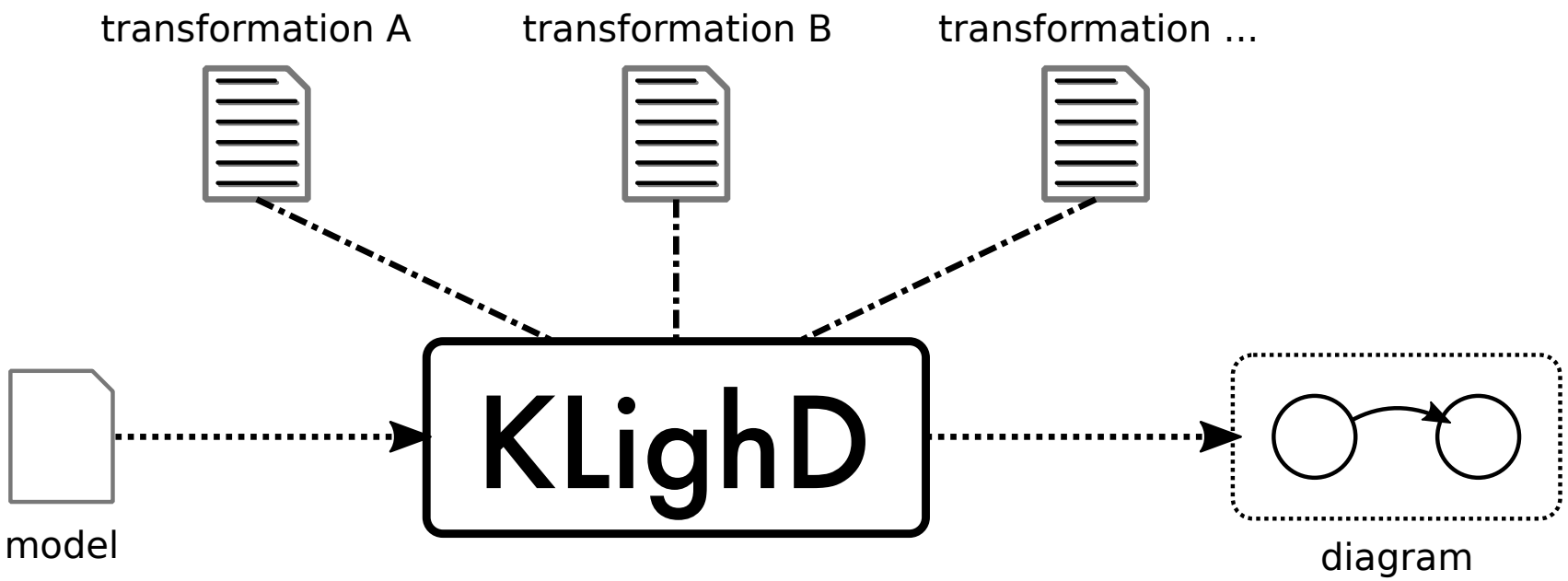
Lightweight Diagrams: Building blocks

- Automatic arrangement of depicted elements – macro layout
- Local arrangement of the figures' primitives – micro layout
- A description language to formulate diagrams smartly

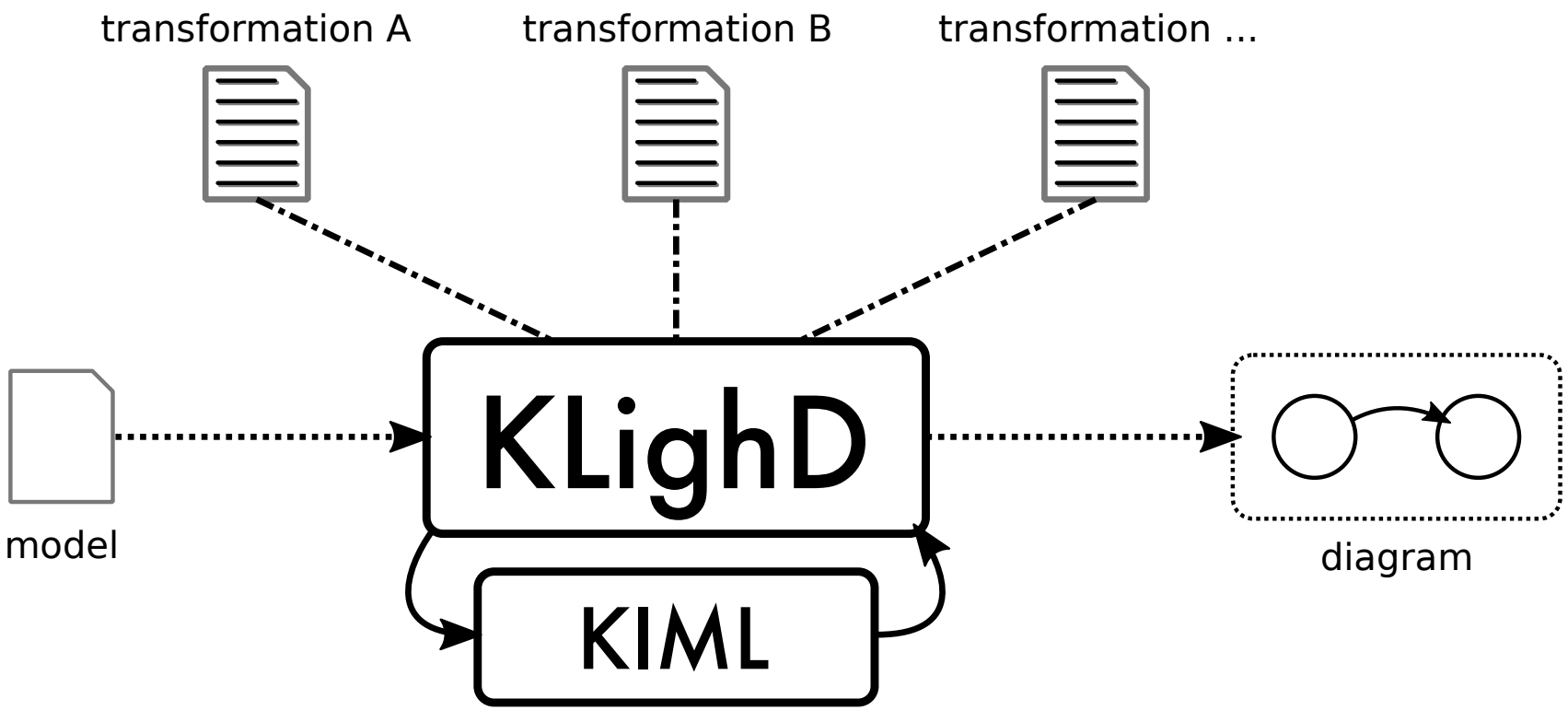
```
KNode {
 width=200 height=200
 KRoundedRectangle cornerWidth=10, cornerHeight=10 {
  KText "On" {
   KAreaPlacementData
    topLeft TOP abs=0px rel=0, LEFT abs=10px rel=0
    bottomRight TOP abs=20px rel=0, RIGHT abs=10px rel=0
  }
  KPolyline {
   points =
    (TOP abs=20px rel=0, LEFT abs=0px rel=0),
    (TOP abs=20px rel=0, RIGHT abs=0px rel=0)
  }
  ...
```

Architecture overview

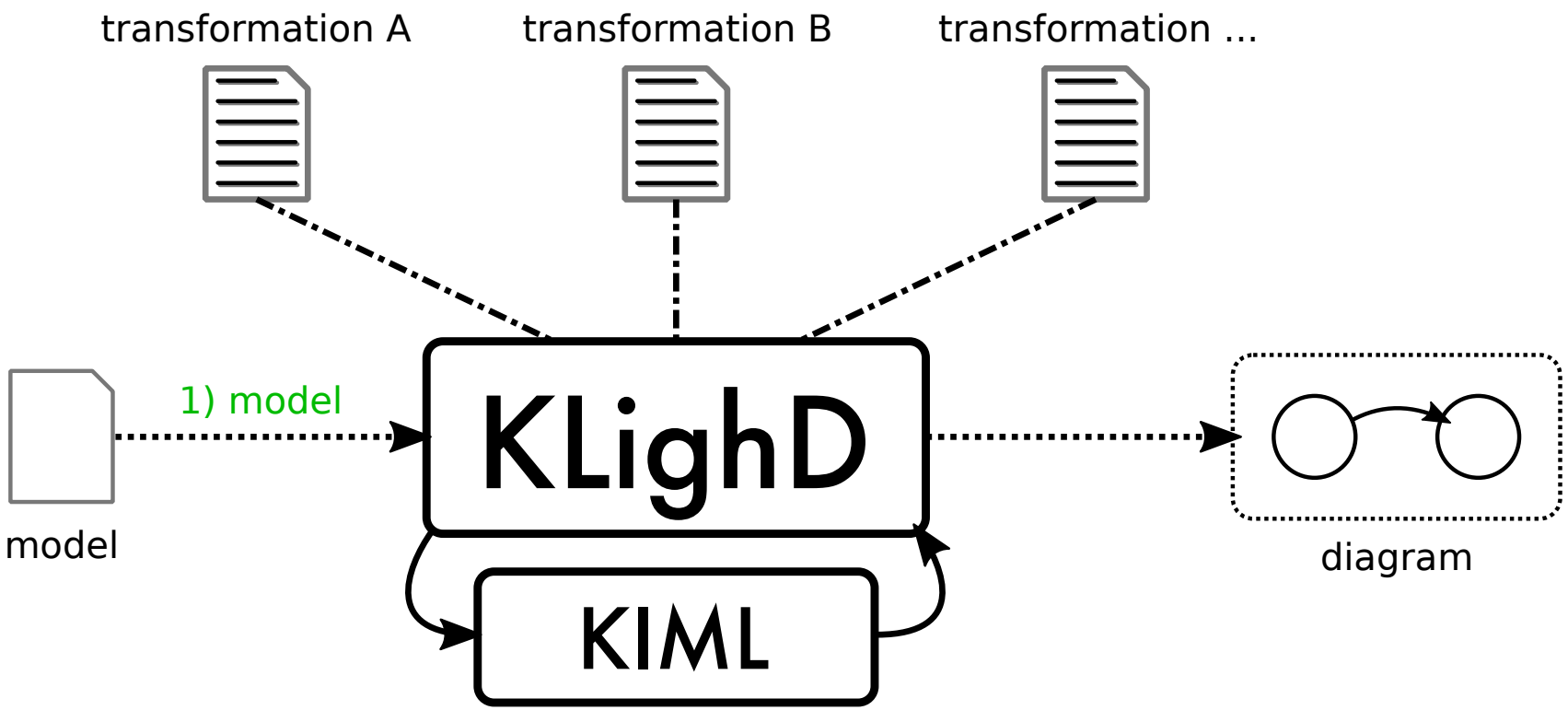
Architecture overview



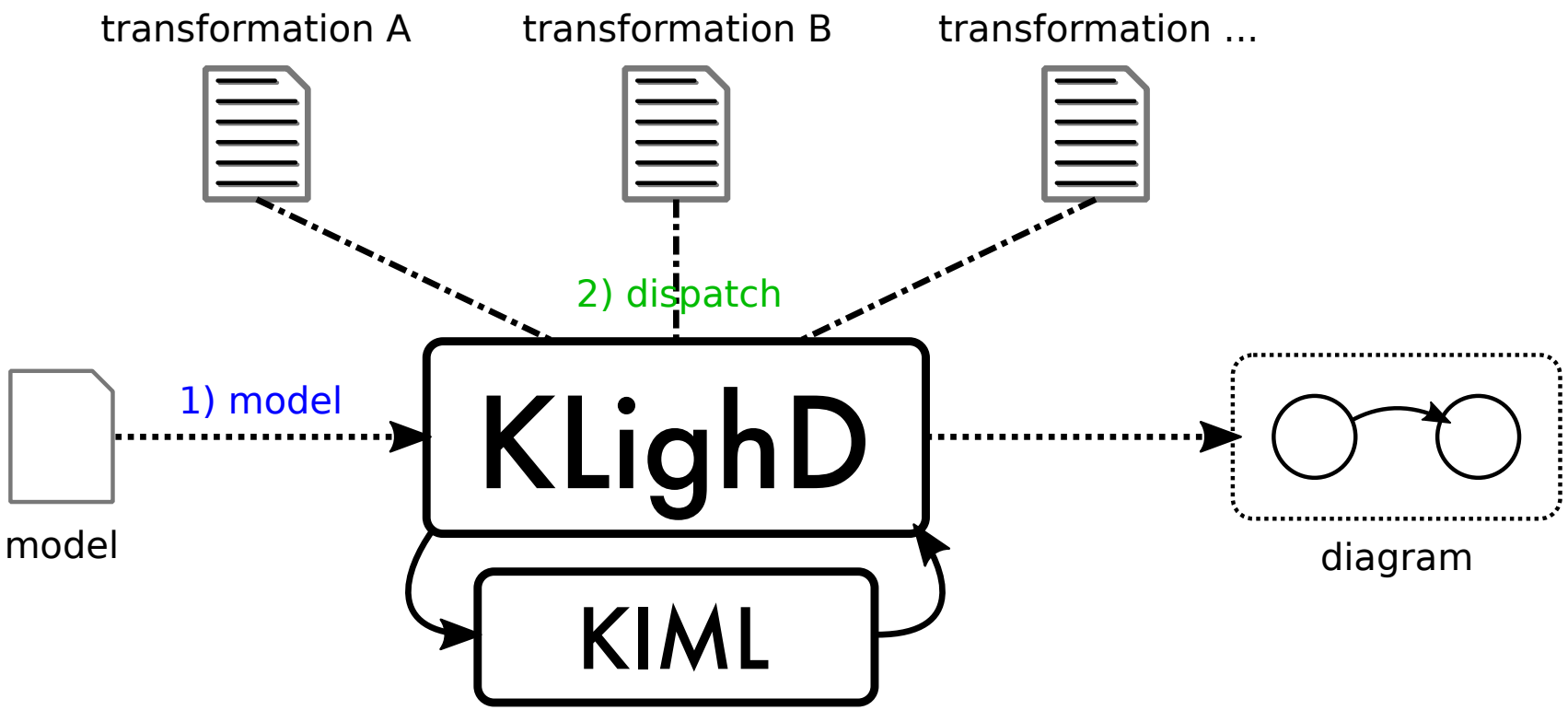
Architecture overview



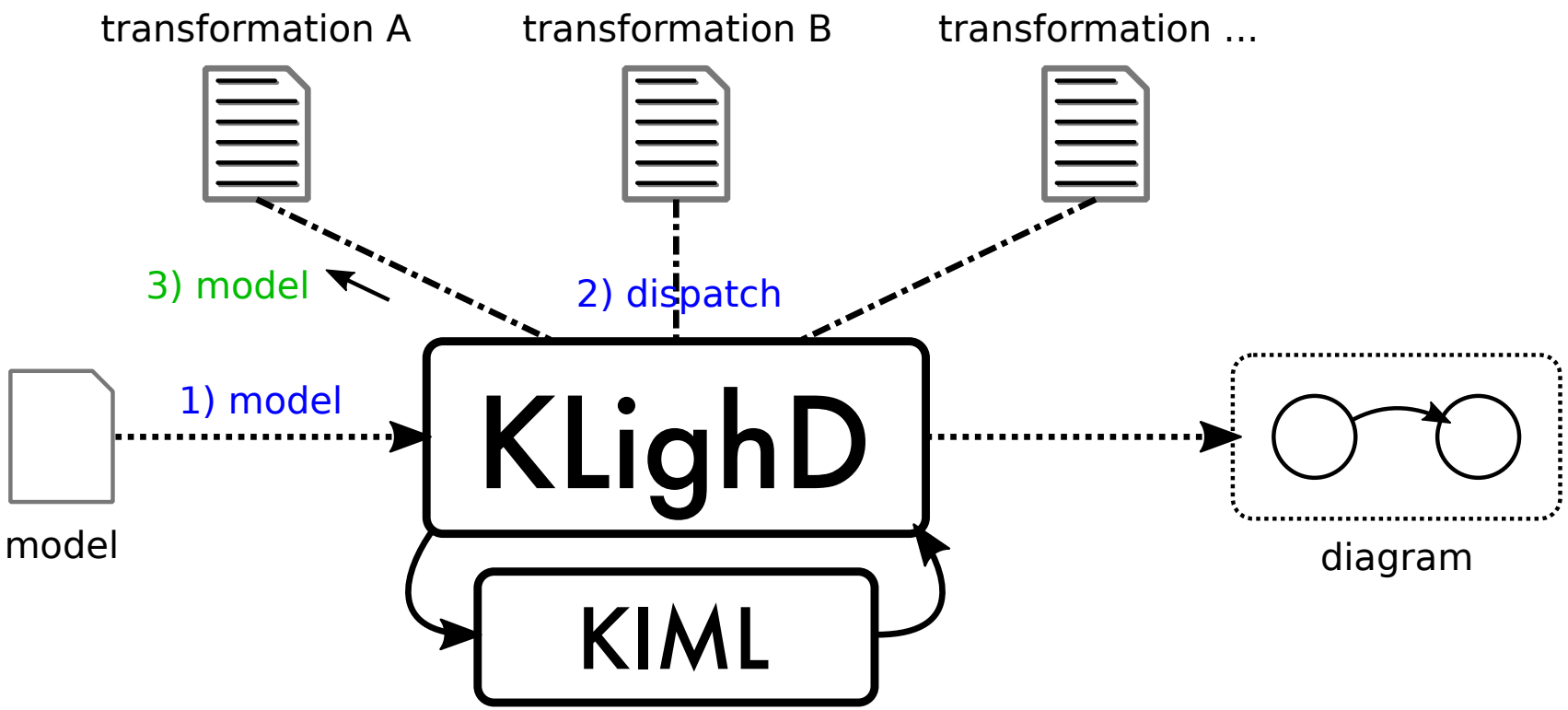
Architecture overview



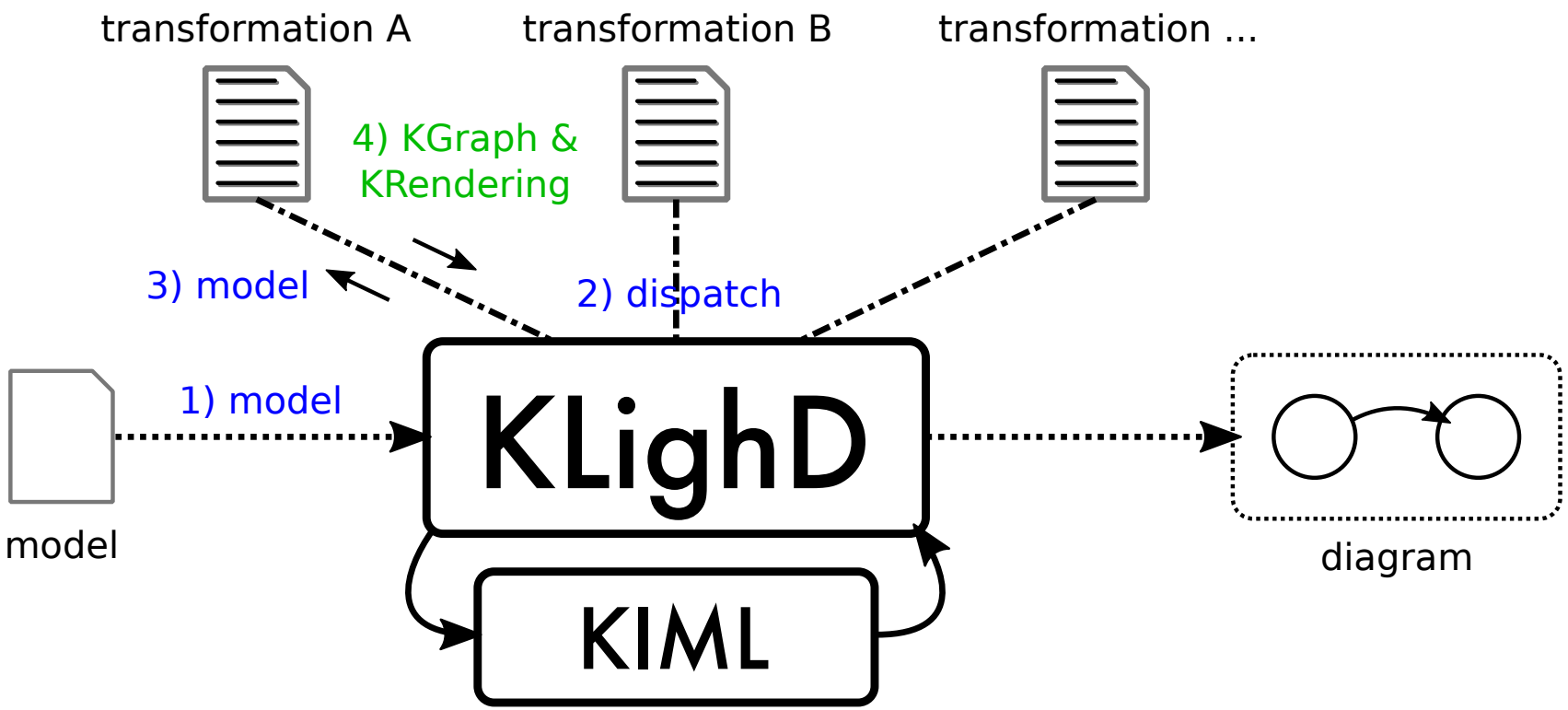
Architecture overview



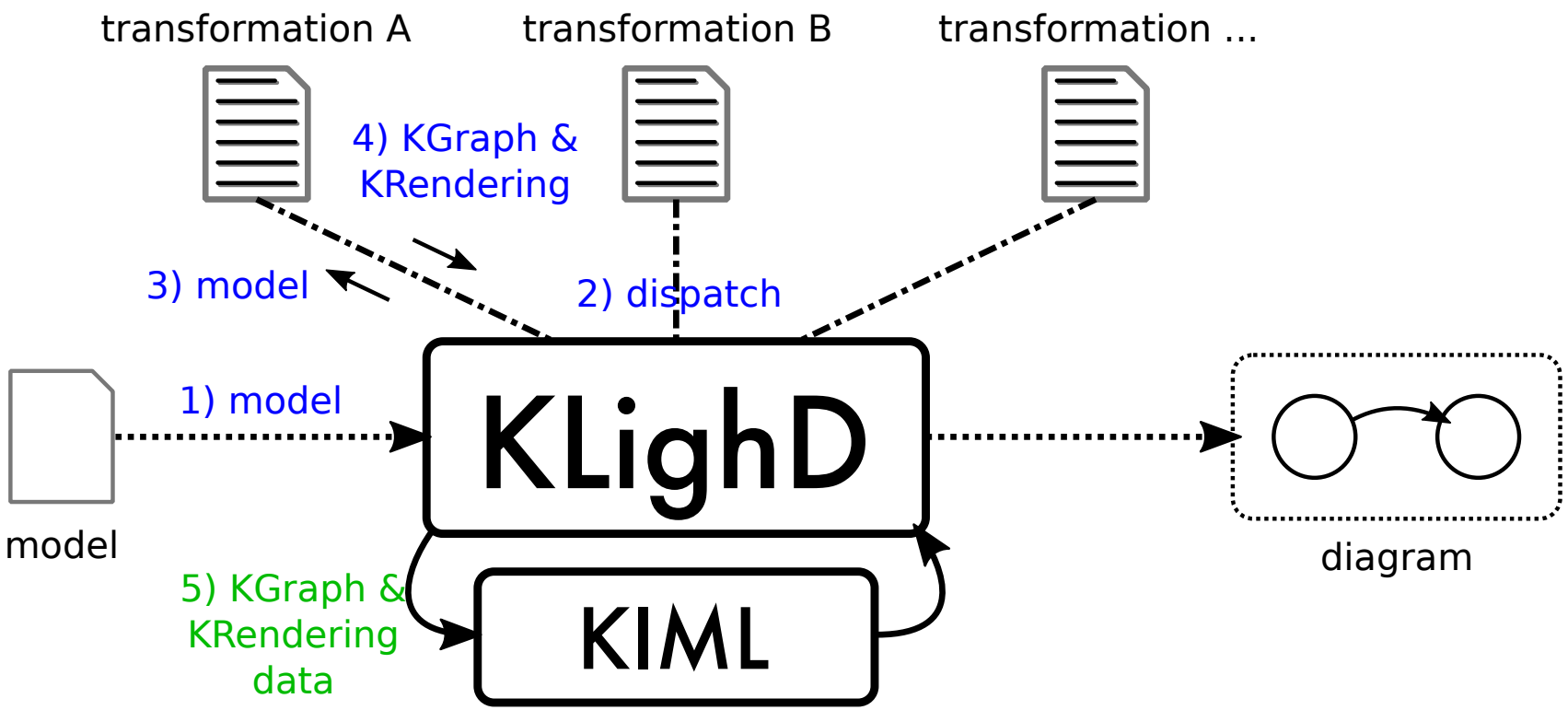
Architecture overview



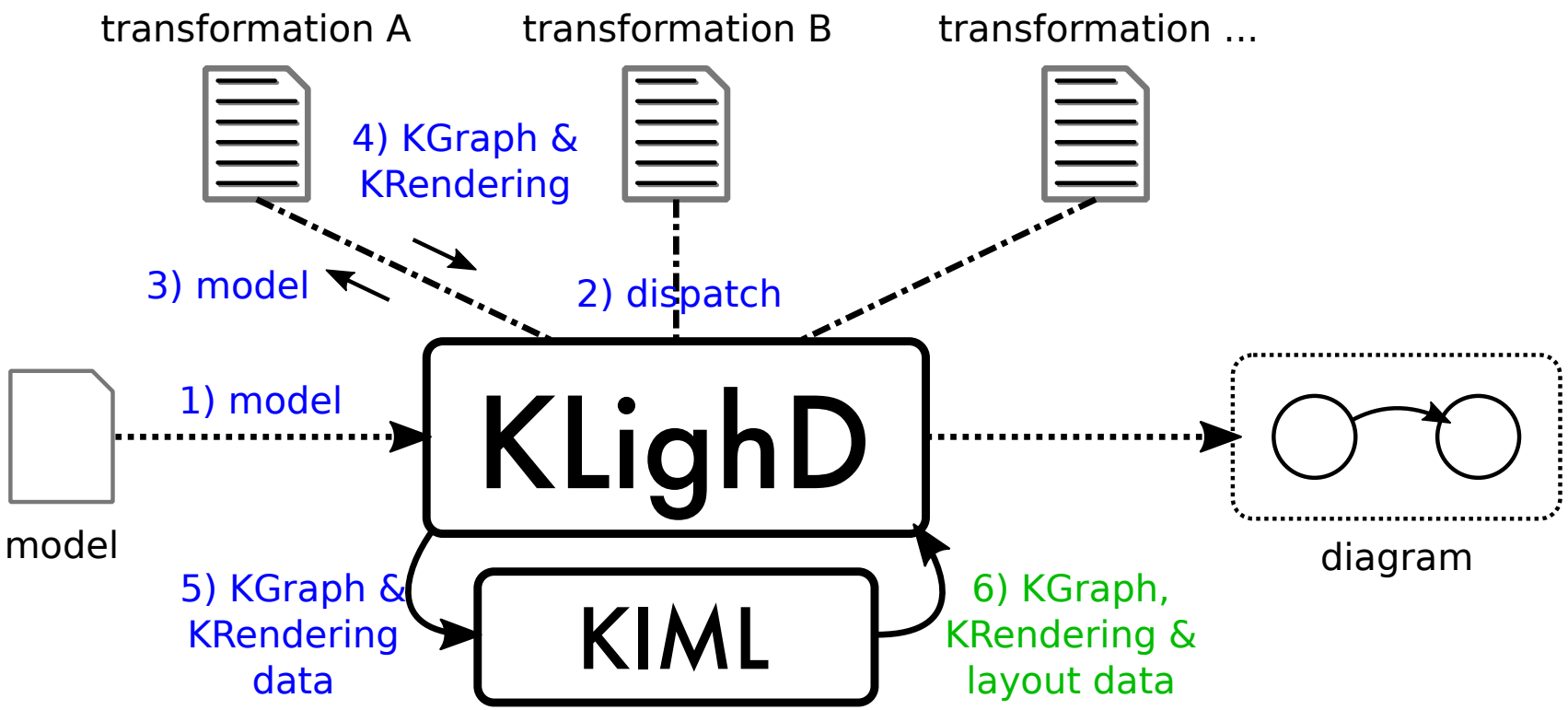
Architecture overview



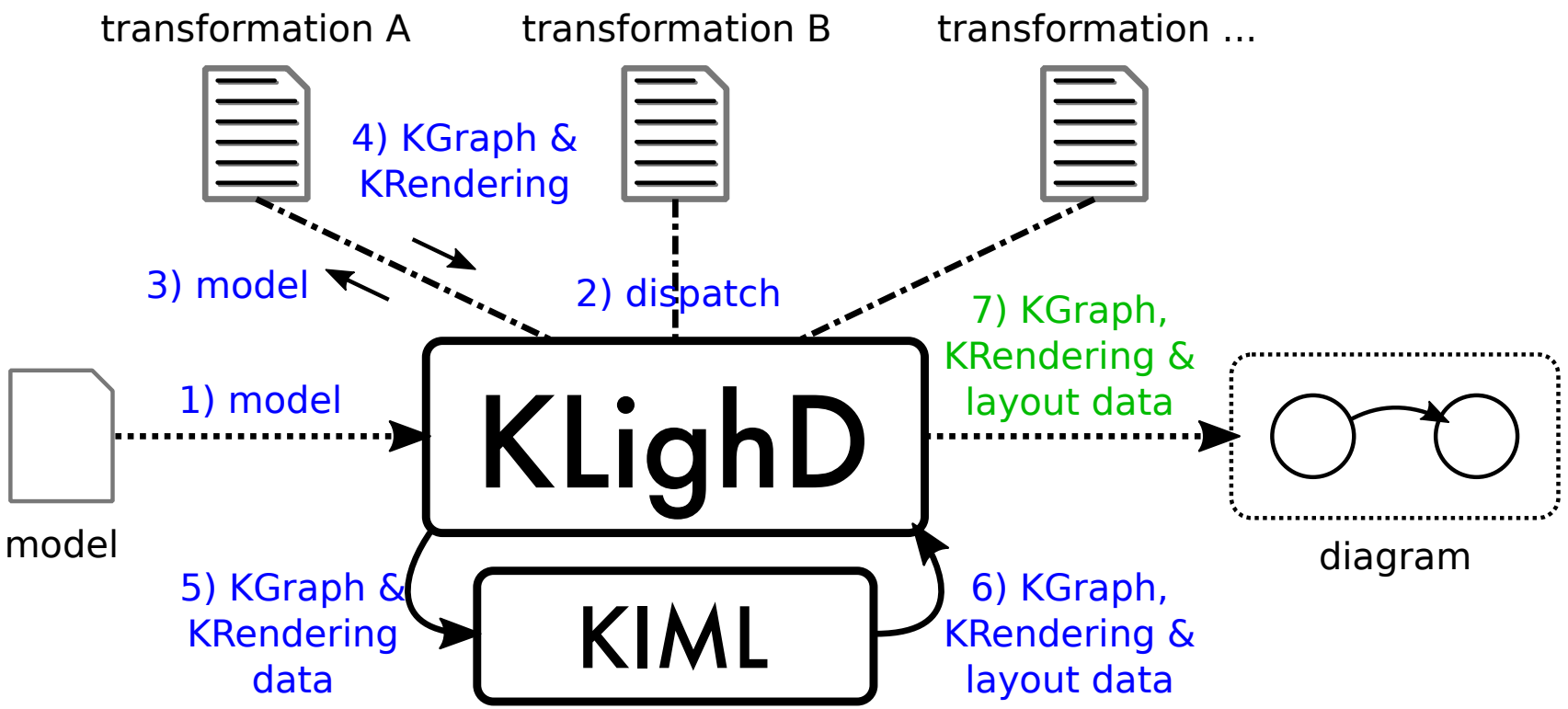
Architecture overview



Architecture overview



Architecture overview



KGraph/KRendering Diagram Notation Format

- Designed for describing concrete diagrams
  - Formulated in EMF's meta modeling language *Ecore*
  - Lifts diagram synthesis to modeling level of abstraction

KGraph/KRendering Diagram Notation Format

- Designed for describing concrete diagrams
  - Formulated in EMF's meta modeling language *Ecore*
  - Lifts diagram synthesis to modeling level of abstraction
- Clear separation of diagram structure (KGraph, coordinates) and appearance (KRendering)
- Enables declarative micro layout statements
- Aims at reducing concrete-coordinate-related statements as much as possible

Schneider, Spönemann, von Hanxleden
Just Model! – Putting Automatic Synthesis of Node-Link-Diagrams into Practice
VL/HCC'13

Outline

Wrap-Up

The Problem

- Lots of productivity wasted with drawing diagrams manually
- Sign for trouble: the palette

Wrap-Up

The Problem

- Lots of productivity wasted with drawing diagrams manually
- Sign for trouble: the palette

Our Approach

- Pragmatics-aware modeling

Wrap-Up

The Problem

- Lots of productivity wasted with drawing diagrams manually
- Sign for trouble: the palette

Our Approach

- Pragmatics-aware modeling
- Let designer concentrate on *model*, automatically synthesize *views*

Wrap-Up

The Problem

- Lots of productivity wasted with drawing diagrams manually
- Sign for trouble: the palette

Our Approach

- Pragmatics-aware modeling
- Let designer concentrate on *model*, automatically synthesize *views*
- Employ *filtering* to synthesize views customized to user stories

Wrap-Up

The Problem

- Lots of productivity wasted with drawing diagrams manually
- Sign for trouble: the palette

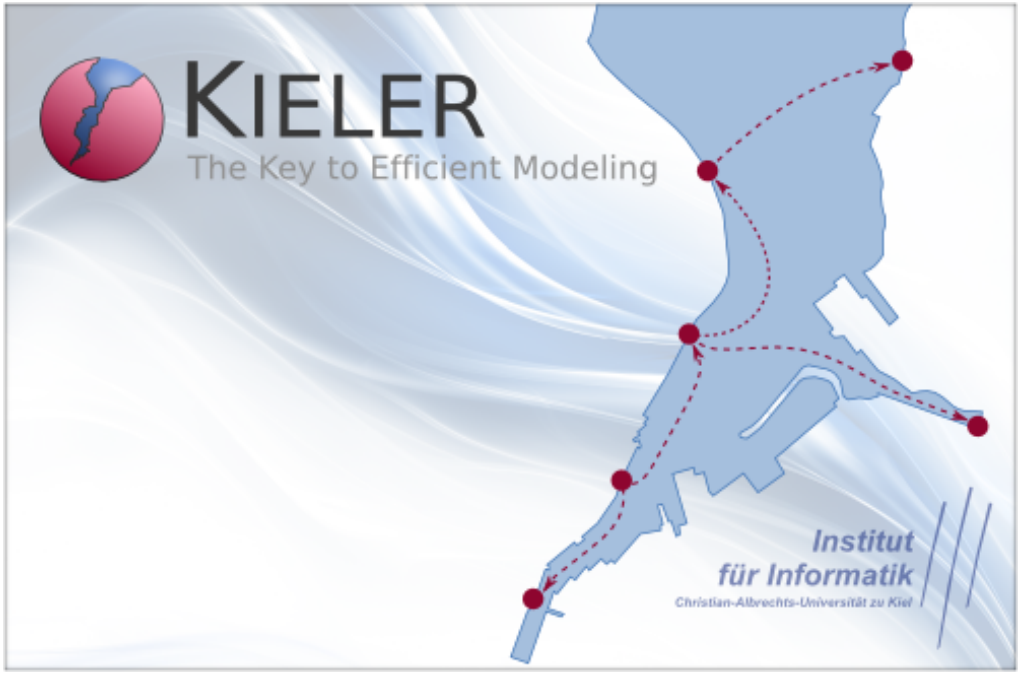
Our Approach

- Pragmatics-aware modeling
- Let designer concentrate on *model*, automatically synthesize *views*
- Employ *filtering* to synthesize views customized to user stories
- Key enabler: automatic layout

Outlook

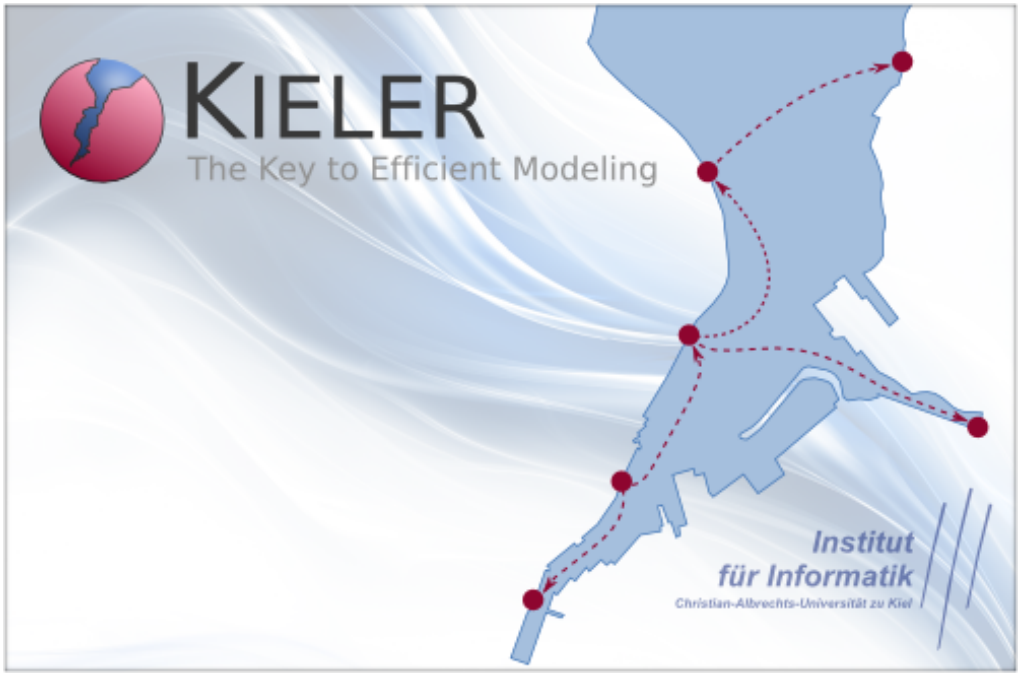
- DSL for synthesis of transient views
- Smart layout configuration
- Collaborative web-based model browsing
- Efficient browsing of very complex models (think google maps)

Get In Touch



- Open-source license (EPL) encourages commercial use
- `http://www.informatik.uni-kiel.de/rtsys/kieler/`

Get In Touch



- Open-source license (EPL) encourages commercial use
- `http://www.informatik.uni-kiel.de/rtsys/kieler/`

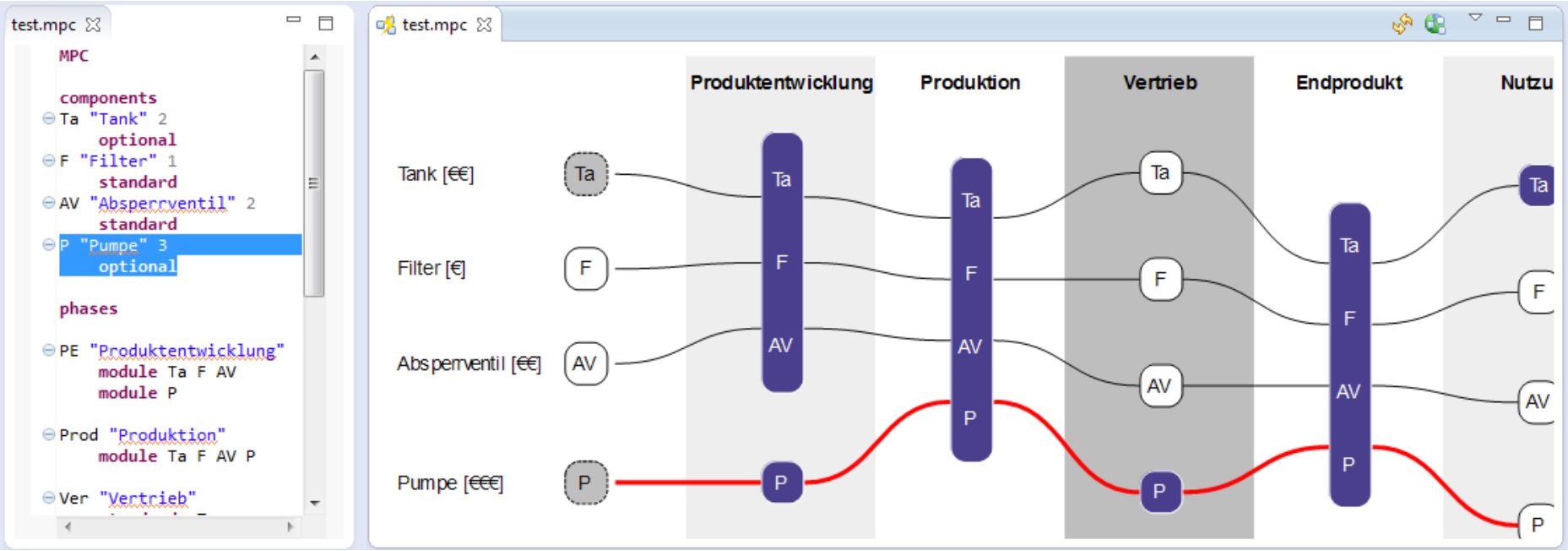
thanks!

questions or comments?

Appendix

Further examples

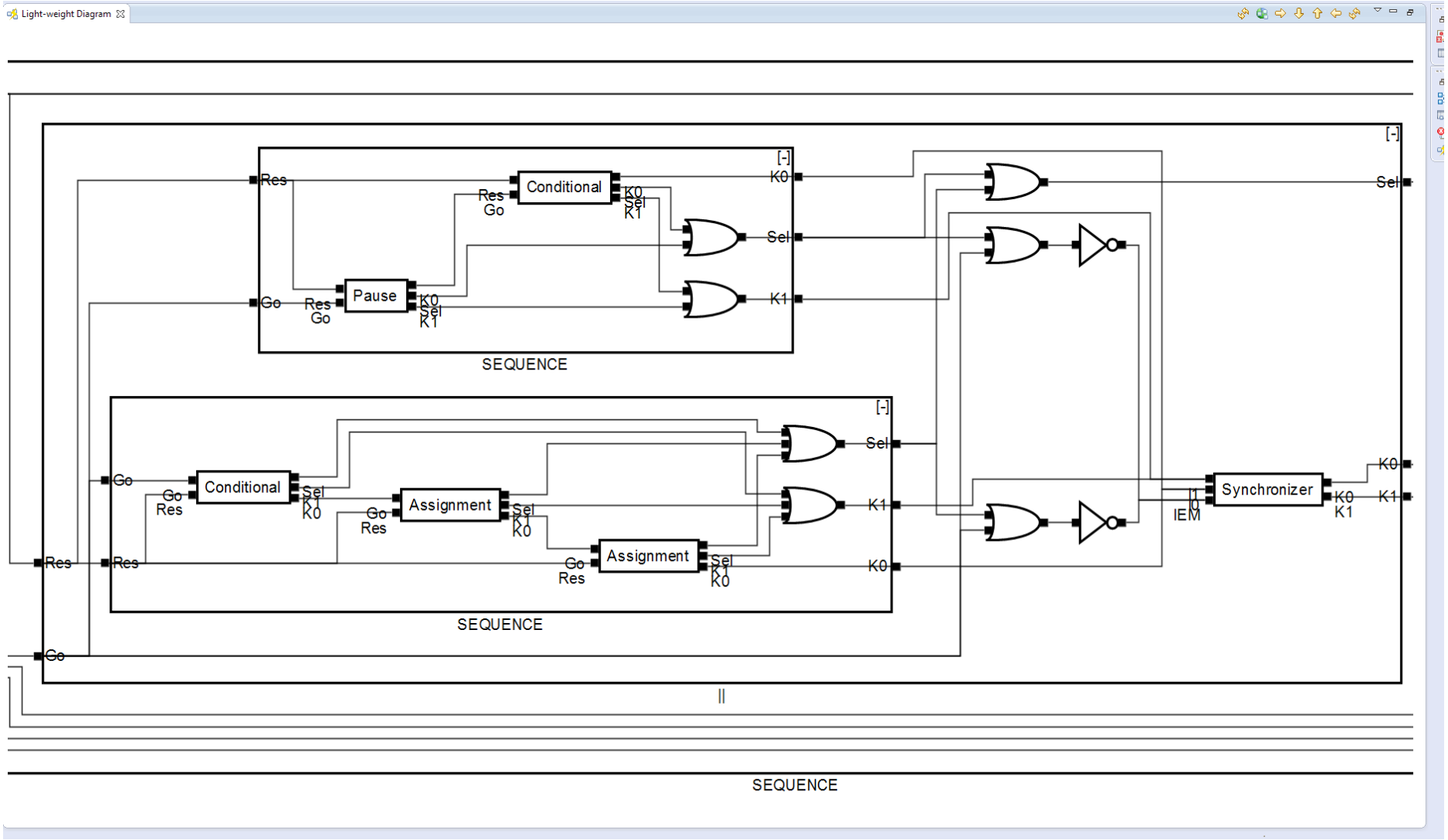
Component clustering



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

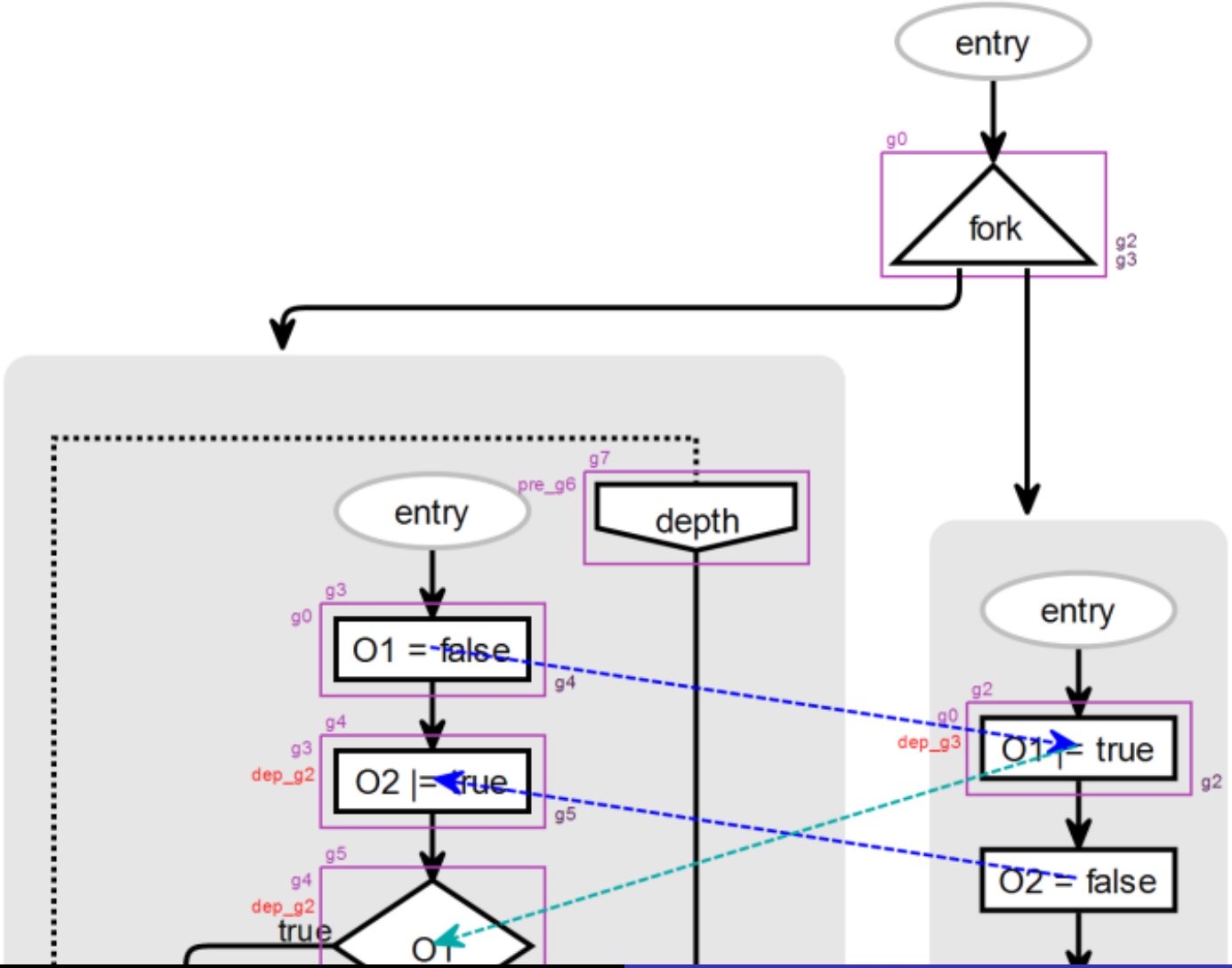
Further examples

Net lists



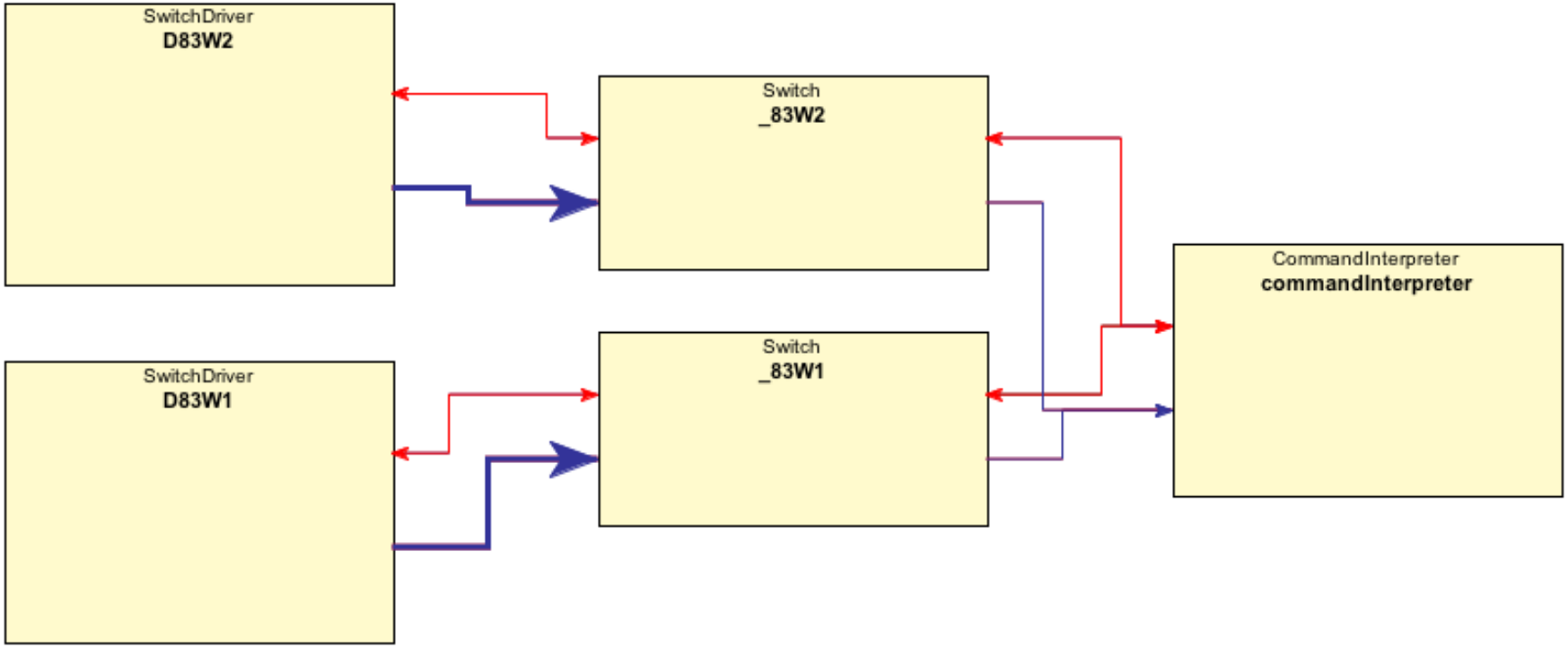
Further examples

Control flow graph



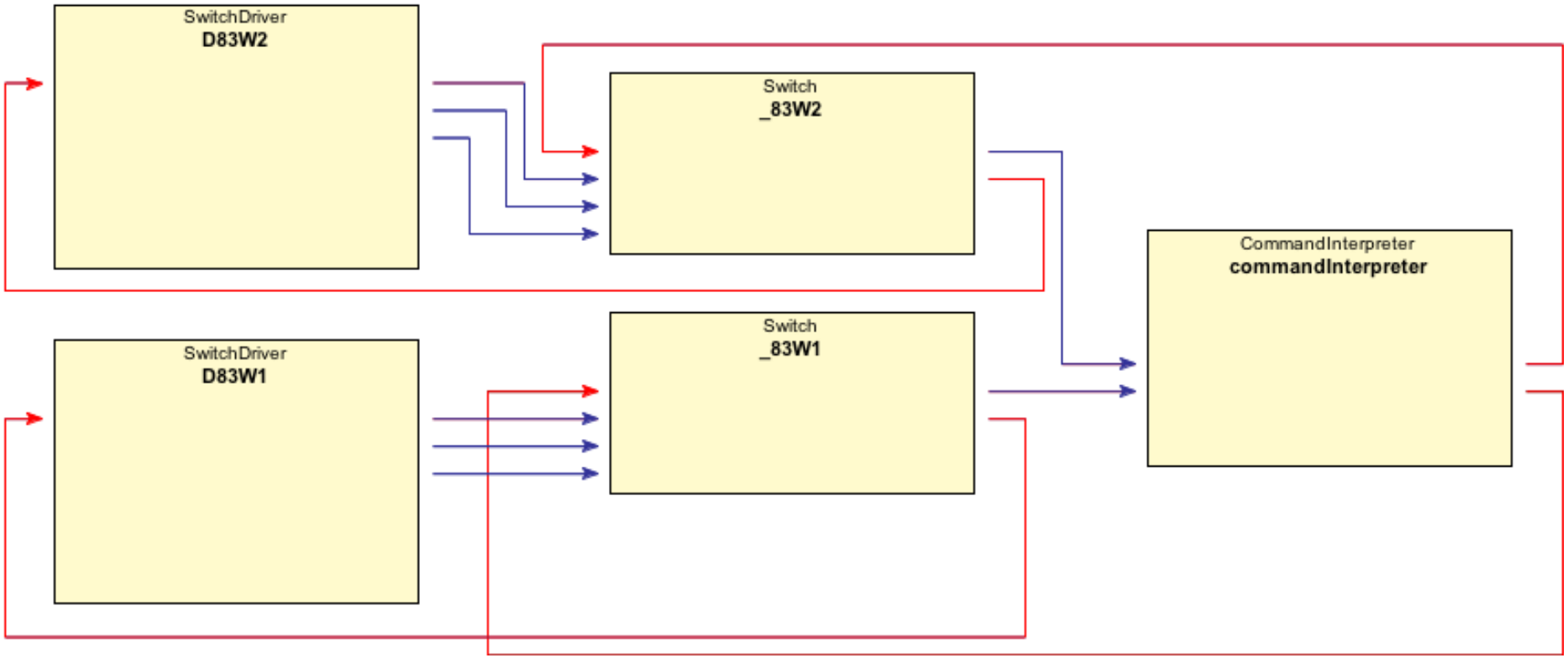
Further examples

Domain-specific components



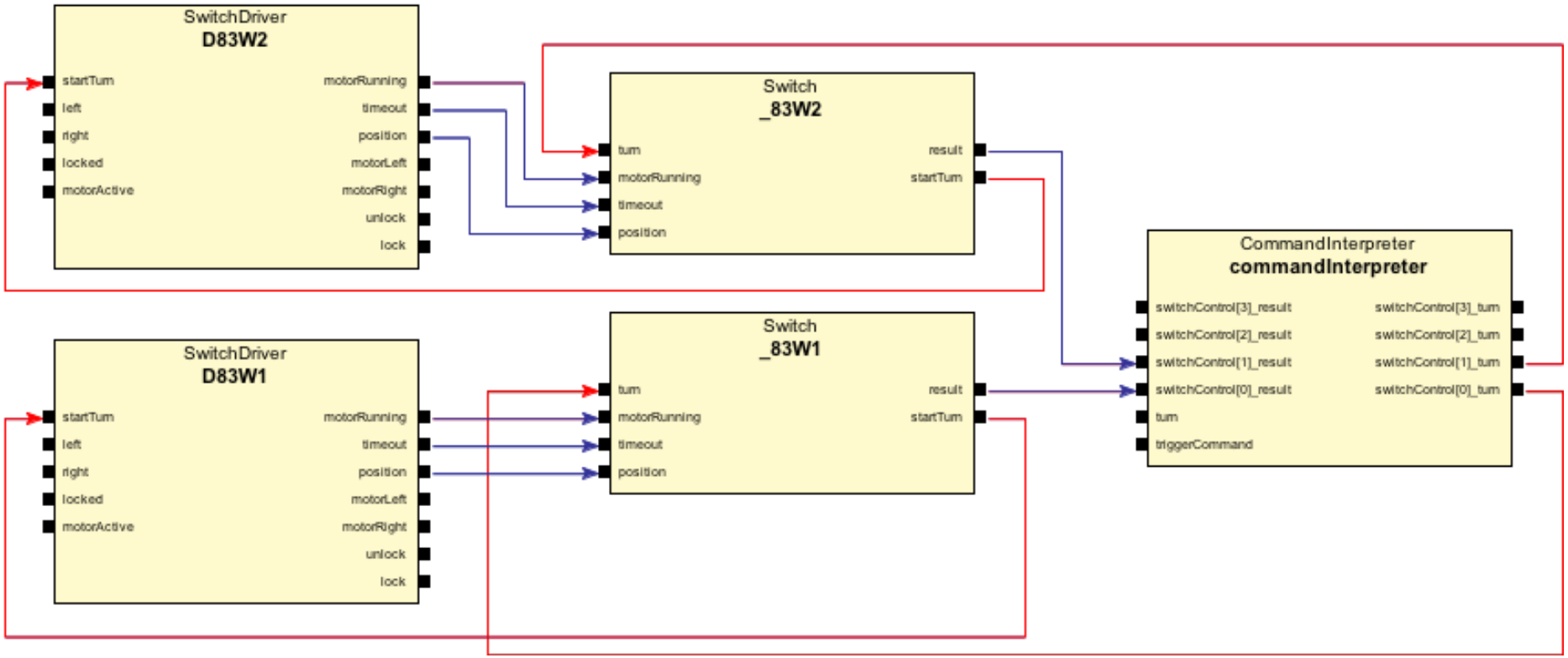
Further examples

Domain-specific components



Further examples

Domain-specific components



Related Work

Robert Ian Bull
Towards A Model Driven Engineering Approach For Information Visualization
Ph.D. thesis, University of Victoria, BC, Canada (2008)

Jan Koehnlein
Discovery Diagrams for the Generic Graphical View
http://koehnlein.blogspot.de/2012/01/discovery-diagrams-for-generic.html

Related Work

Robert Ian Bull
Towards A Model Driven Engineering Approach For Information Visualization
Ph.D. thesis, University of Victoria, BC, Canada (2008)

Jan Koehnlein
Discovery Diagrams for the Generic Graphical View
http://koehnlein.blogspot.de/2012/01/discovery-diagrams-for-generic.html

Martin Auer, T. Tschurtschenthaler, and Stefan Biffl,
A Flyweight UML Modelling Tool for Software Development in Heterogeneous Environments
29th EUROMICRO Conference "New Waves in System Architecture" (2003)
http://www.umlet.com

OMG Diagram Definition (DD), 2012)
http://www.omg.org/spec/DD/1.0/

More Related Work

Akos Ledeczi et al.
The Generic Modeling Environment
IEEE Workshop on Intelligent Signal Processing (WISP 2001)

Gergely Mezei et al.
Visual presentation solutions for domain specific languages
IASTED International Conference on Software Engineering (2006)

M. Minas
Generating meta-model-based freehand editors
3rd International Workshop on Graph Based Tools (GraBaTs'06)

Eclipse GMF (http://www.eclipse.org/modeling/gmp/)

Graphiti project (http://www.eclipse.org/graphiti/)

Benjamin B. Bederson et al.
Toolkit Design for Interactive Structured Graphics
IEEE Transactions on Software Engineering, Vol. 30, No. 8 (2004)