On Integration of Textual and Graphical Modeling

Pragmatics in MENGES

Christian Schneider

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

KoSSE-Workshop, November 10, 2010

Outline

The Context of MENGES

Setting

- Design of safety-critical controlling systems in the rail-bounded transportation field

Current state in the business

- Requirements analysis
  $\longrightarrow$ huge amount of text documents, informal
- Design specifications with a proprietary modeling language
  $\longrightarrow$ formalized description of the system logic

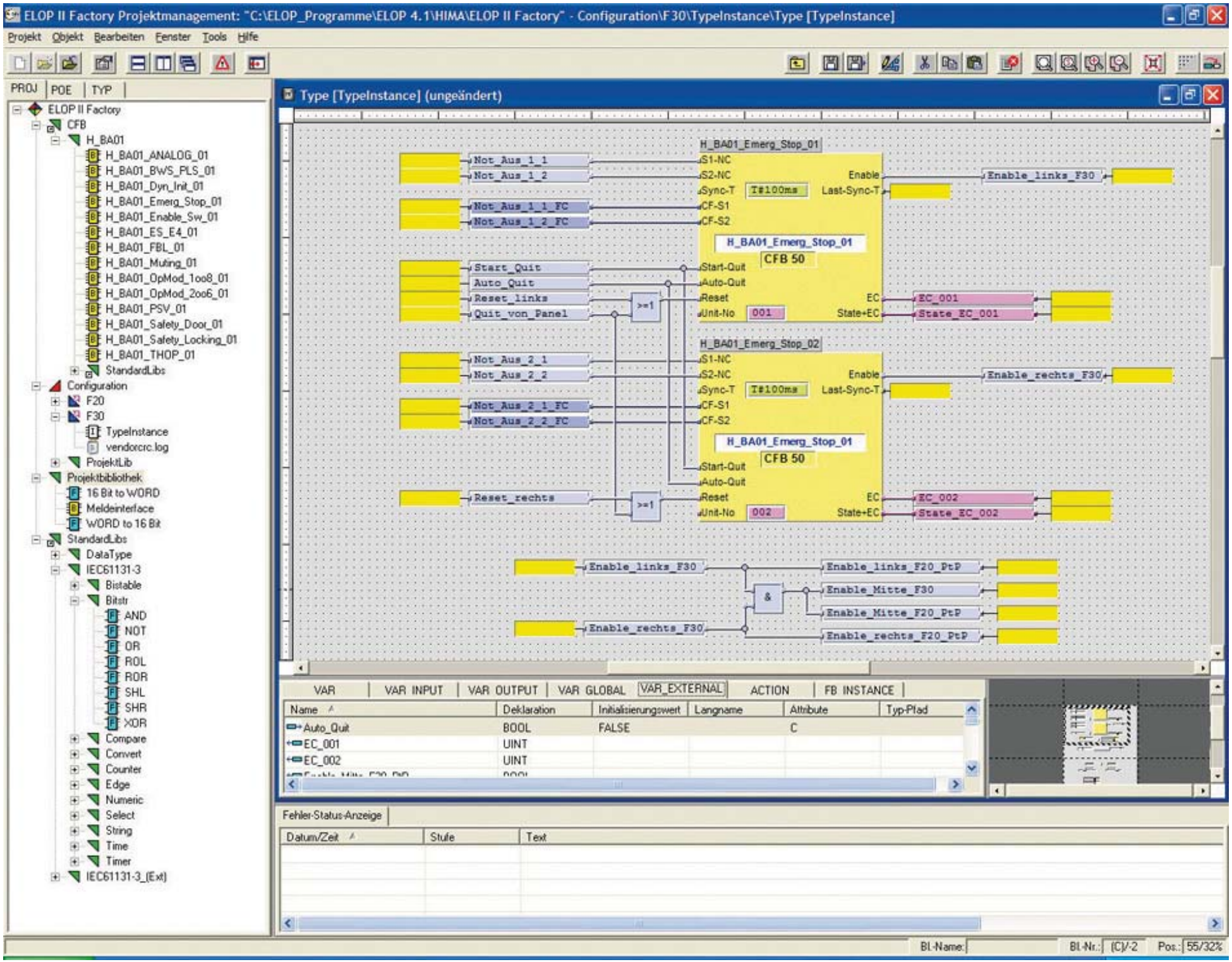
The Context of MENGES

Setting

- Design of safety-critical controlling systems in the rail-bounded transportation field

Current state in the business

- Requirements analysis
  $\longrightarrow$ huge amount of text documents, informal
- Design specifications with a proprietary modeling language
  $\longrightarrow$ formalized description of the system logic
- Implementation in a model-based way



http://www.hima.de/Produkte/ELOP_II_Factory_default.php

The Context of MENGES

Setting

- Design of safety-critical controlling systems in the rail-bounded transportation field

Current state in the business

- Requirements analysis
  $\longrightarrow$ huge amount of text documents, informal
- Design specifications with a proprietary modeling language
  $\longrightarrow$ formalized description of the system logic
- Implementation in a model-based way

The Context of MENGES

Setting

- Design of safety-critical controlling systems in the rail-bounded transportation field

Current state in the business

- Requirements analysis
  $\longrightarrow$ huge amount of text documents, informal
- Design specifications with a proprietary modeling language
  $\longrightarrow$ formalized description of the system logic
- Implementation in a model-based way

Most important issue

- No integration of those specifications (derivation, tracing, . . .)

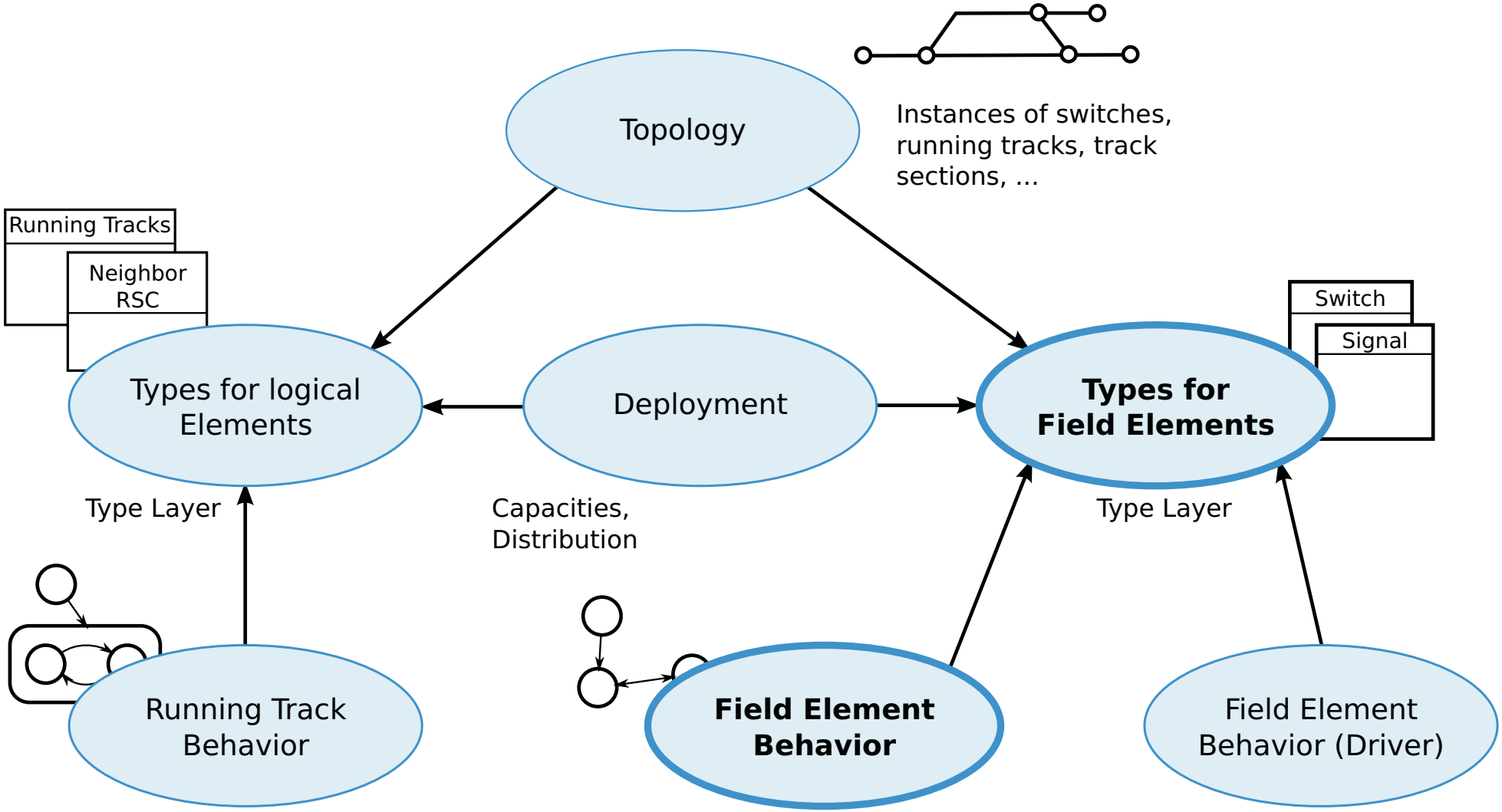
The Aim of MENGES

- Engineers shall be assisted in these tasks, i.e. while
  - analysing,
  - designing,
  - maintaining,
  - testing,
  - verifying,
  - documenting and
  - translating systems and their parts

The Aim of MENGES

- Engineers shall be assisted in these tasks, i.e. while
  - analysing,
  - designing,
  - maintaining,
  - testing,
  - verifying,
  - documenting and
  - translating systems and their parts
- Domain Specific Languages(DSLs)+tooling are to be created
  - intended to cover the necessary specification parts and
  - form a homogeneous development environment

Kinds of specifications



Field Element description

```
1  field element Gleisabschnitt extends Beanspruchbar {
2    statevars
3      reserviert:
4        ( ja,
5          nein );
6      beansprucht:
7        ( nicht,
8          DWeg,     // als Durchrutschweg-Element
9          FLR,      // im Flankenschutzraumn
10         FWR,      // im Fahrweg einer Rangierstrasse
11         FWZ,      // im Fahrweg einer Zugstrasse
12         DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR, FWZ_DWeg,
13         DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR    );
14   procedures
15     reservieren() = {
16         reserviert -> ja
17     };
18 }
```

Behavior specification - State Machines

```
1  state machine Gleisabschnitt_beansprucht_SM {
2      references beansprucht in Gleisabschnitt;
3      transitions
4      start nicht   --> DWeg, FLR, FWR, FWZ;
5      DWeg          --> nicht, DWeg_FLR, FWZ_DWeg;
6      FLR           --> nicht, DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR;
7      FWR           --> nicht, FWR_FLR;
8      FWZ           --> nicht, FWZ_FLR, FWZ_DWeg;
9      DWeg_FLR      --> DWeg, FLR, DWeg_FLR_FLR;
10     FLR_FLR       --> FLR, DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR;
11     FWR_FLR       --> FLR, FWR, FWR_FLR_FLR;
12     FWZ_DWeg      --> DWeg, FWZ;
13     FWZ_FLR       --> FLR, FWZ, FWZ_FLR_FLR;
14     DWeg_FLR_FLR  --> DWeg_FLR, FLR_FLR;
15     FWR_FLR_FLR   --> FWR_FLR, FLR_FLR;
16     FWZ_FLR_FLR   --> FWZ_FLR, FLR_FLR;
17 }
```

Behavior specification - Rule Blocks

```
1  rule block reserviere references Gleisabschnitt {
2      rule graph
3      --> [istReservierungsVorbedingung()] {
4          -> [beansprucht == nicht] / {
5              reservieren()
6          };
7          -> [beansprucht == DWeg] / {
8              reservieren()
9          };
10         -> [beansprucht == FLR] / {
11             reservieren()
12         };
13         -> [beansprucht == FLR_FLR] / {
14             reservieren()
15         };
16     }
17 }
```

Assessment

The textual languages are ...

- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easily and fast editable

Assessment

The textual languages are ...

- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easily and fast editable

but ...

- ☹ content may be hard to conceive
- ☹ they are inflexible:
  mostly text = document
- ☹ exploring is laborious
- ☹ the context is missing / get lost quickly

Assessment

The textual languages are ...

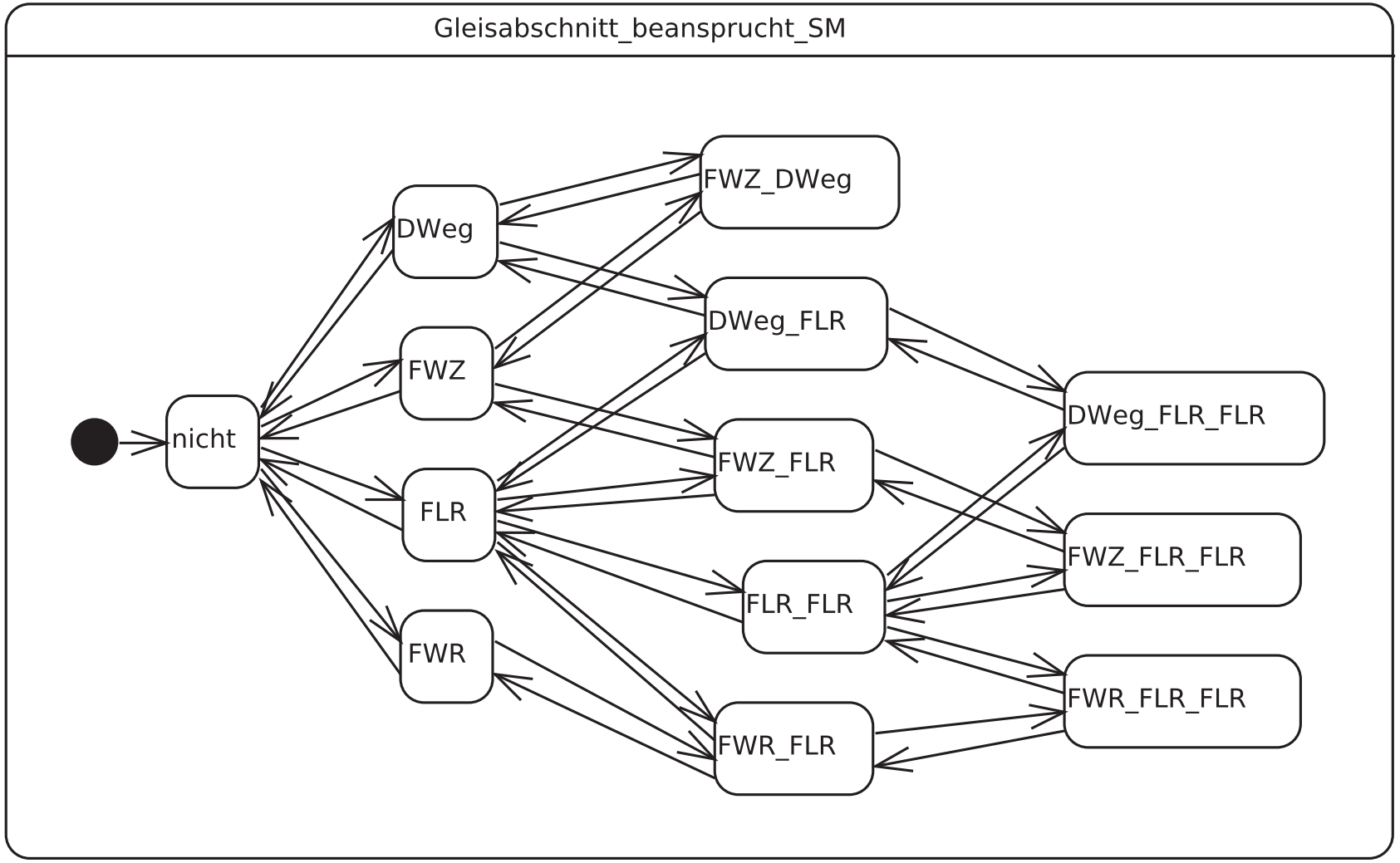
- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easily and fast editable

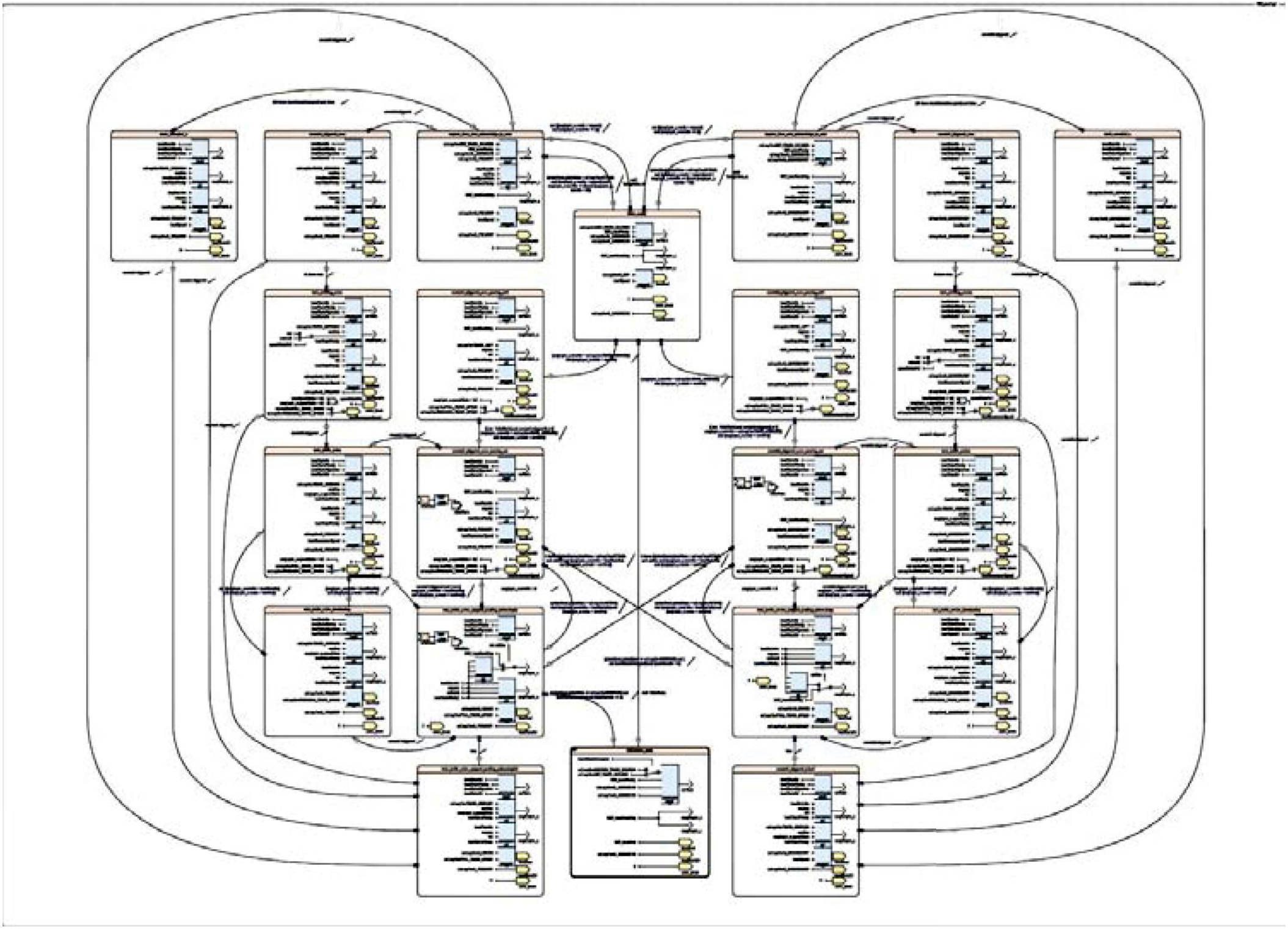
but ...

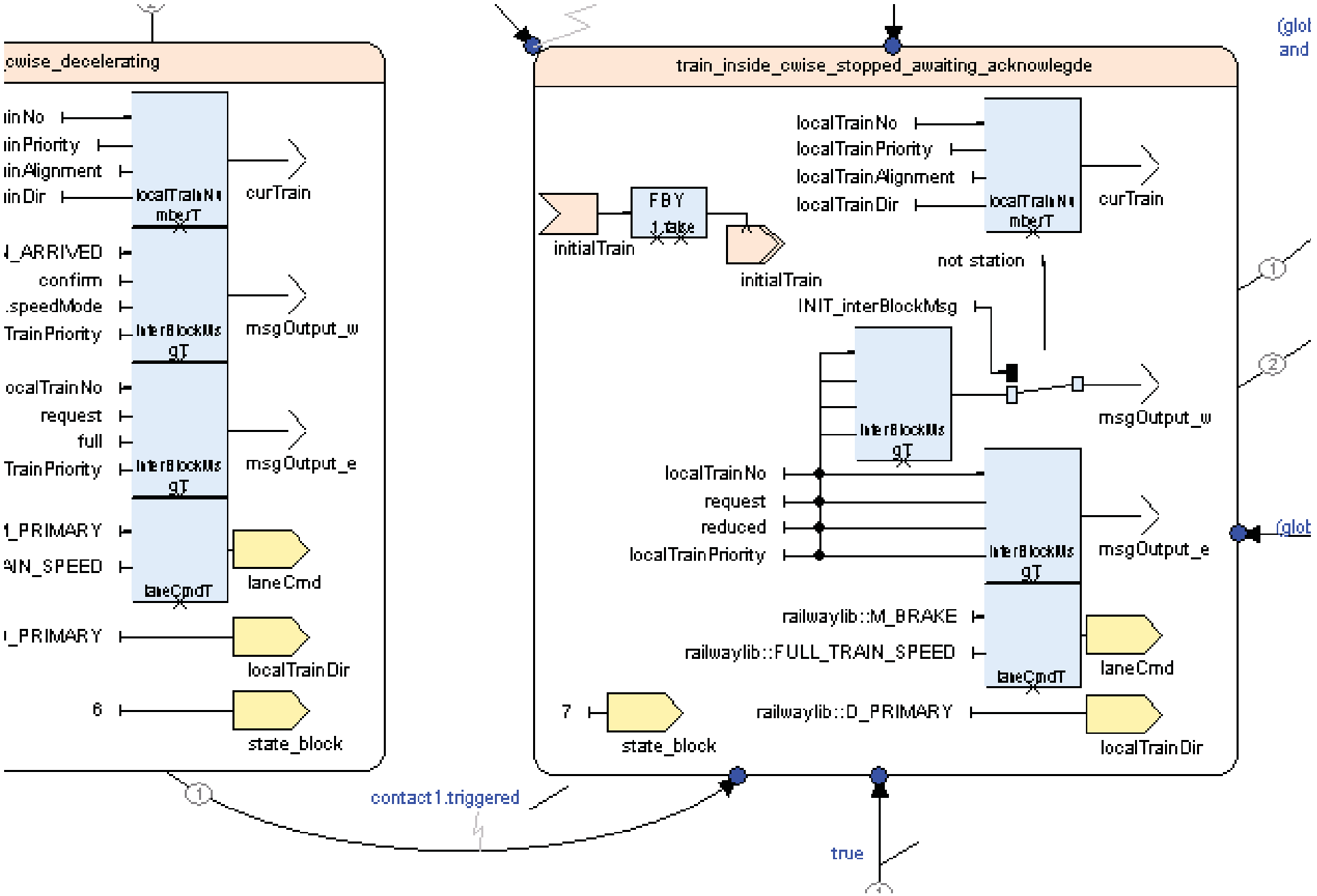
- ☹ content may be hard to conceive
- ☹ they are inflexible:
  mostly text = document
- ☹ exploring is laborious
- ☹ the context is missing / get lost quickly

Question: What about graphical languages?

A graphical notation of state machines






cwise_decelerating
curTrain
msgOutput_w
msgOutput_e
laneCmd
localTrainDir
state_block
train_inside_cwise_stopped_awaiting_acknowlegde
localTrainNo
localTrainPriority
localTrainAlignment
localTrainDir
FBY
1.take
initialTrain
not station
INIT_interBlockMsg
request
reduced
railwaylib::M_BRAKE
railwaylib::FULL_TRAIN_SPEED
railwaylib::D_PRIMARY
contact1.triggered
true

SOS!

Assessment - cont'd

Graphical languages/representations may be ...

- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easier to conceive
- ☺ more flexible

Assessment - cont'd

Graphical languages/representations may be ...

- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easier to conceive
- ☺ more flexible

but ...

- ☹ editing is time-consuming
- ☹ exploring is laborious
- ☹ the context is missing / get lost quickly

Assessment - cont'd

Graphical languages/representations may be ...

- ☺ formal and compact
- ☺ precise in terms of separation of concerns
- ☺ easier to conceive
- ☺ more flexible

but ...

- ☹ editing is time-consuming
- ☹ exploring is laborious
- ☹ the context is missing / get lost quickly

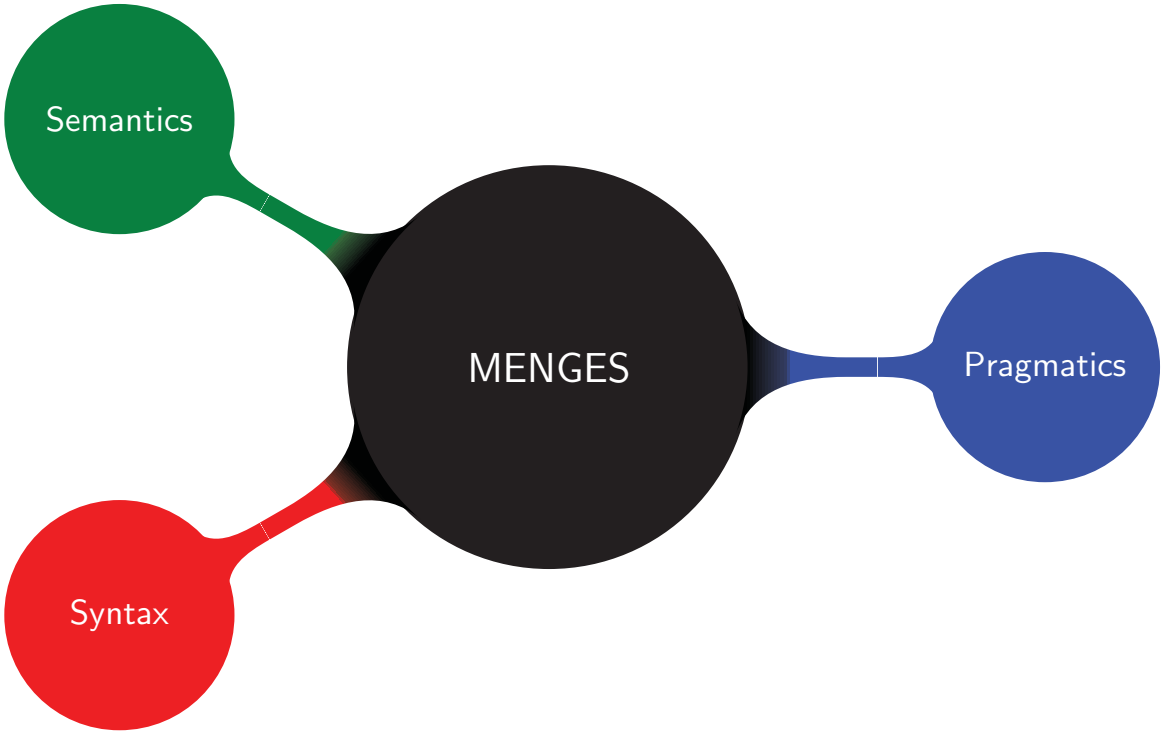
Consequence?

Pragmatics of modeling languages

Pragmatics of modeling languages deserves more attention than it has received so far

Pragmatics of modeling languages

Pragmatics of modeling languages deserves more attention than it has received so far



Pragmatics of modeling languages

Pragmatics of modeling languages deserves more attention than it has received so far.

- Pragmatics usually concentrates on practical aspects of how constructs and features of a language may be used to achieve various objectives (*e. g.*, when to use an assignment).
- Here, focus is on the mechanics of handling a language (editing, maintaining, inspecting).

Pragmatics of modeling languages

Pragmatics of modeling languages deserves more attention than it has received so far.

- Pragmatics usually concentrates on practical aspects of how constructs and features of a language may be used to achieve various objectives (*e. g.*, when to use an assignment).
- Here, focus is on the mechanics of handling a language (editing, maintaining, inspecting).

Pragmatics of modeling languages $=_{def}$
practical aspects of handling a model in a model-based design flows

Our Approach ...

- Get inspiration from successful textual paradigms and tools
- Combine best of graphical and textual worlds
- Provide flexible, alternative views of system under development (SUD) allowing to focus on a certain context

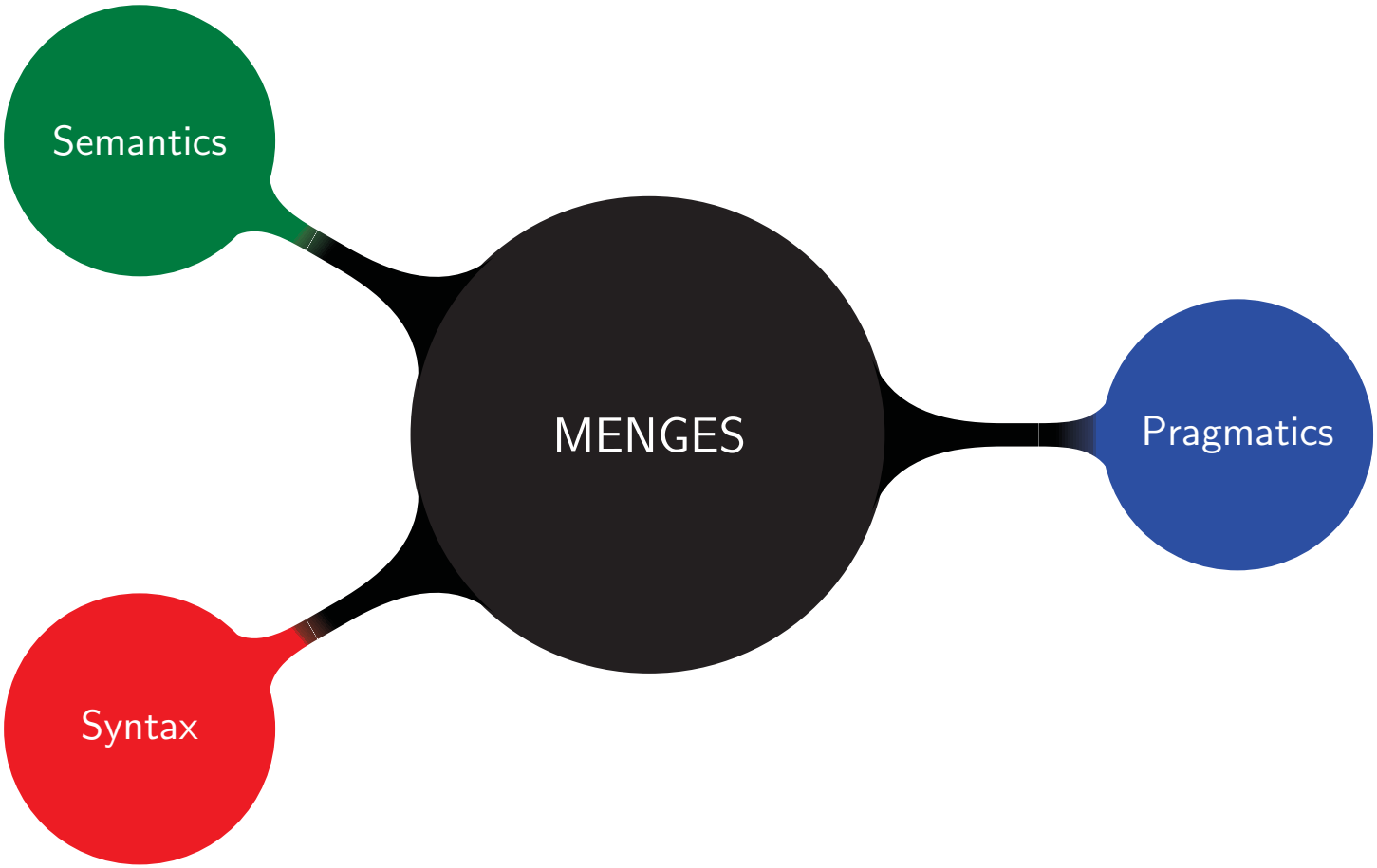
Our Approach ...

- Get inspiration from successful textual paradigms and tools
- Combine best of graphical and textual worlds
- Provide flexible, alternative views of system under development (SUD) allowing to focus on a certain context

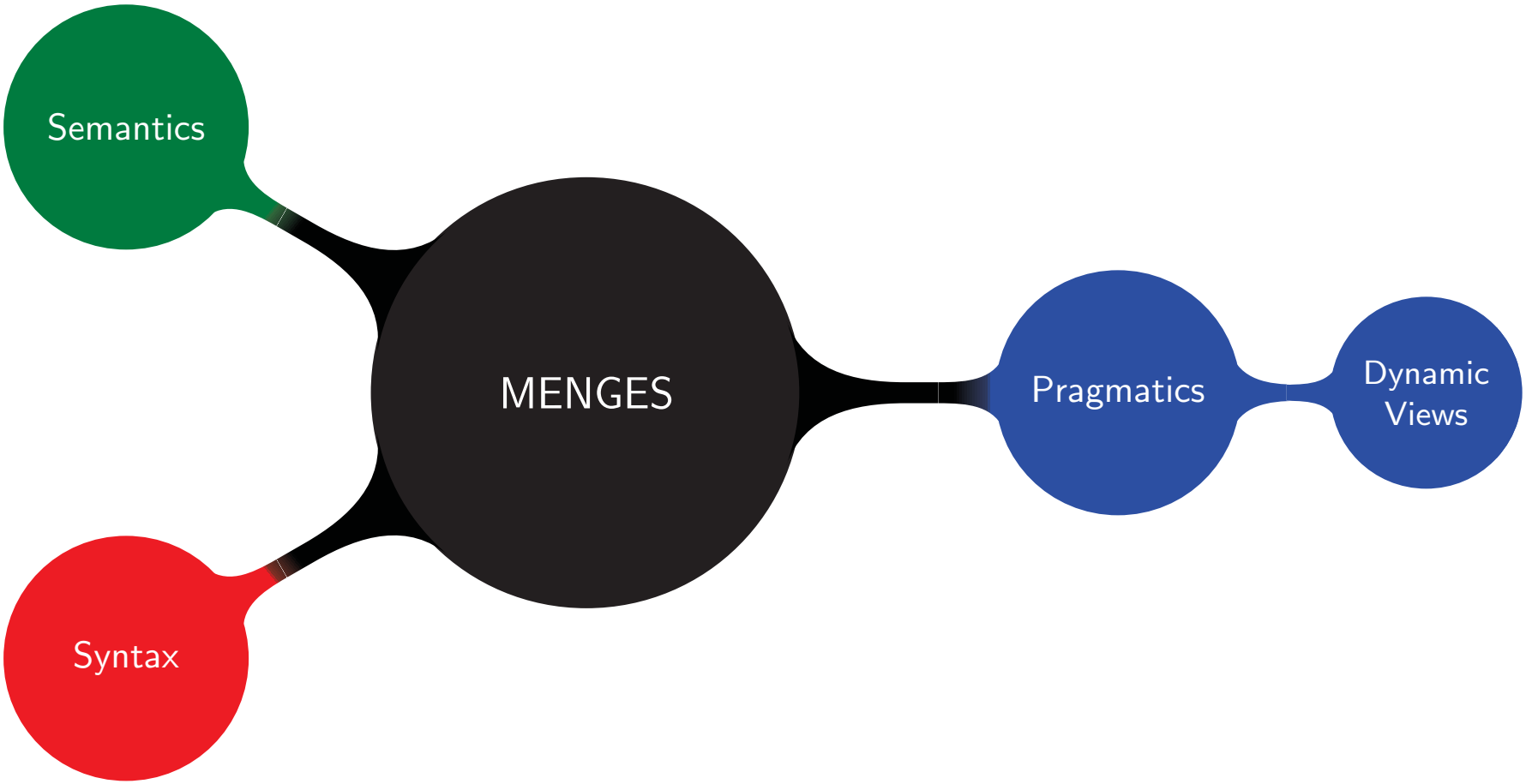
The key enabler:

Automatic, flexible synthesis of graphical & textual views organized by a powerful View Management

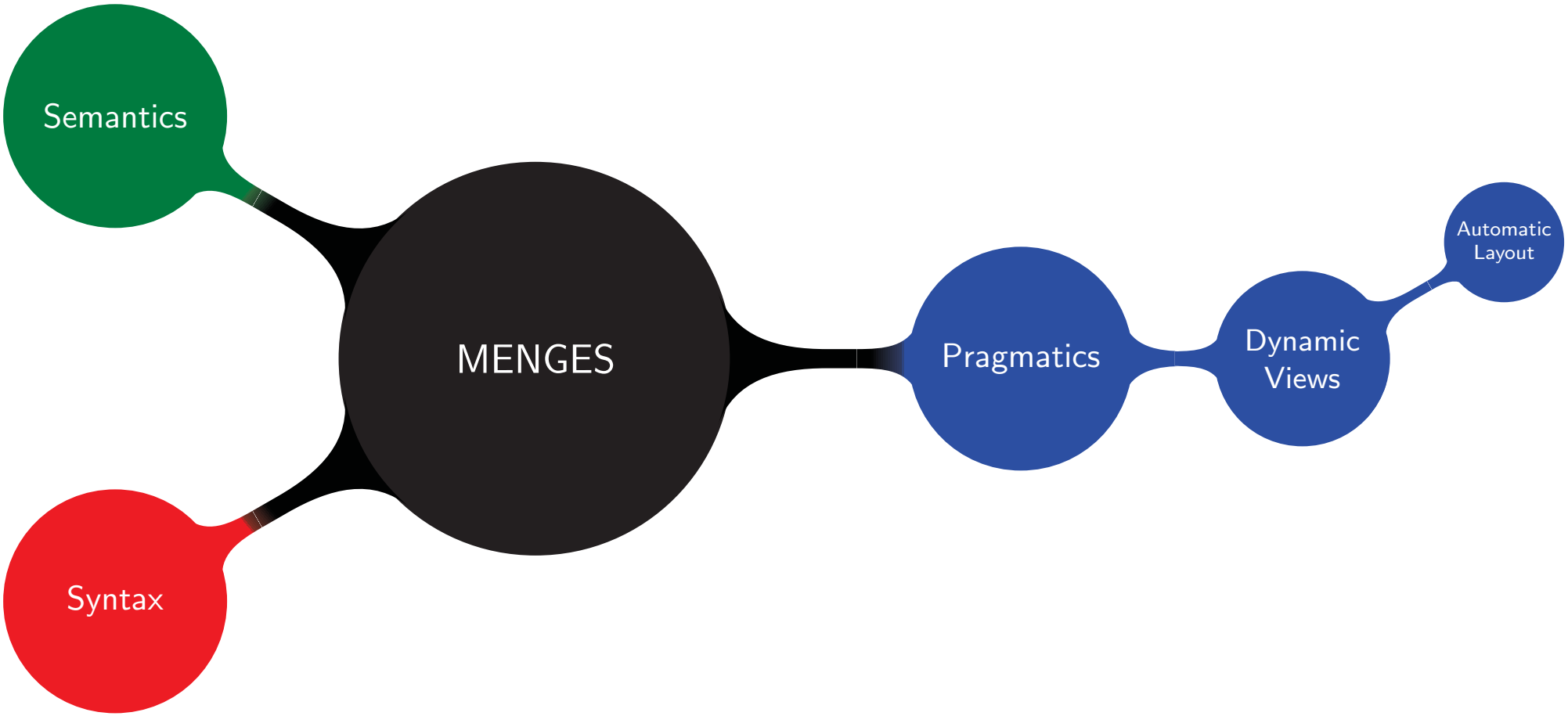
Our Approach ...



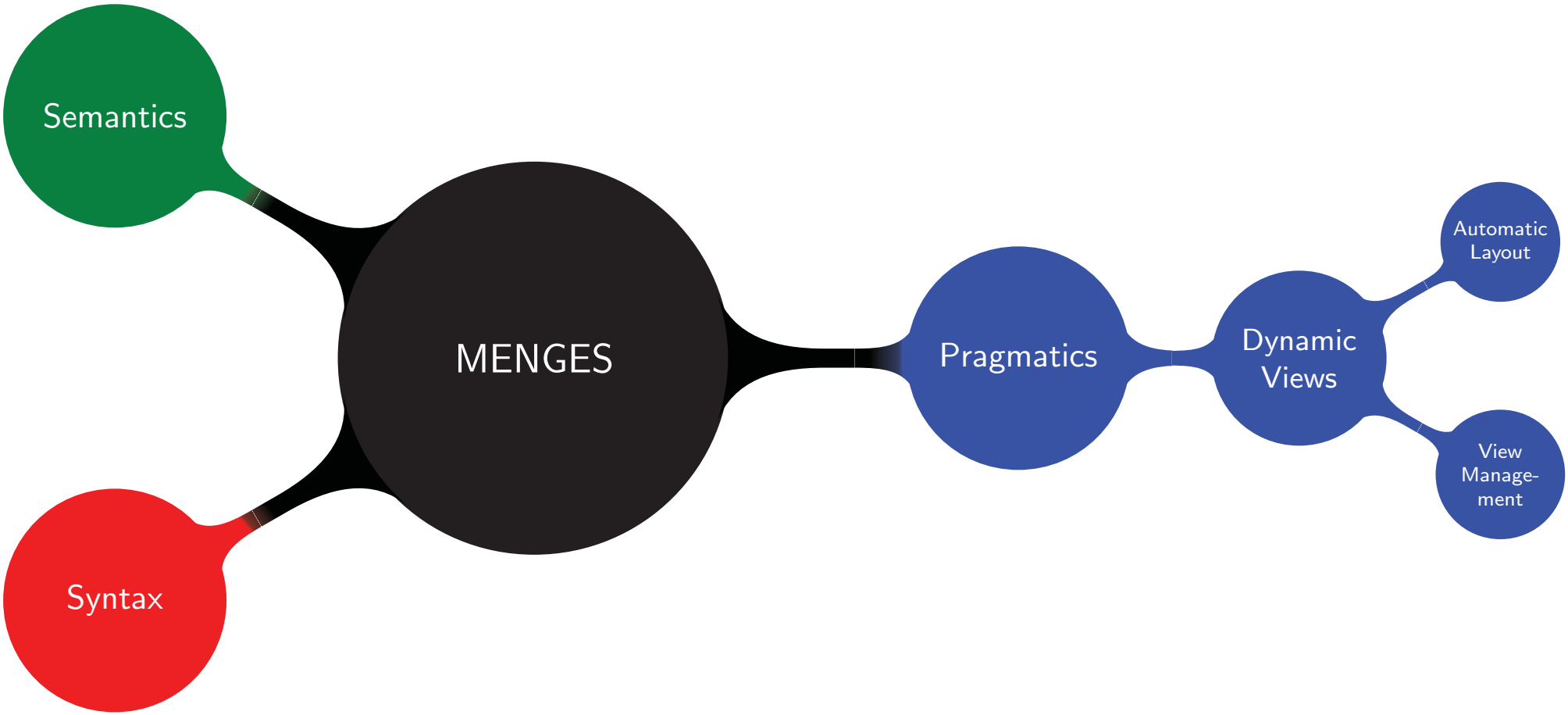
Our Approach ...



Our Approach ...

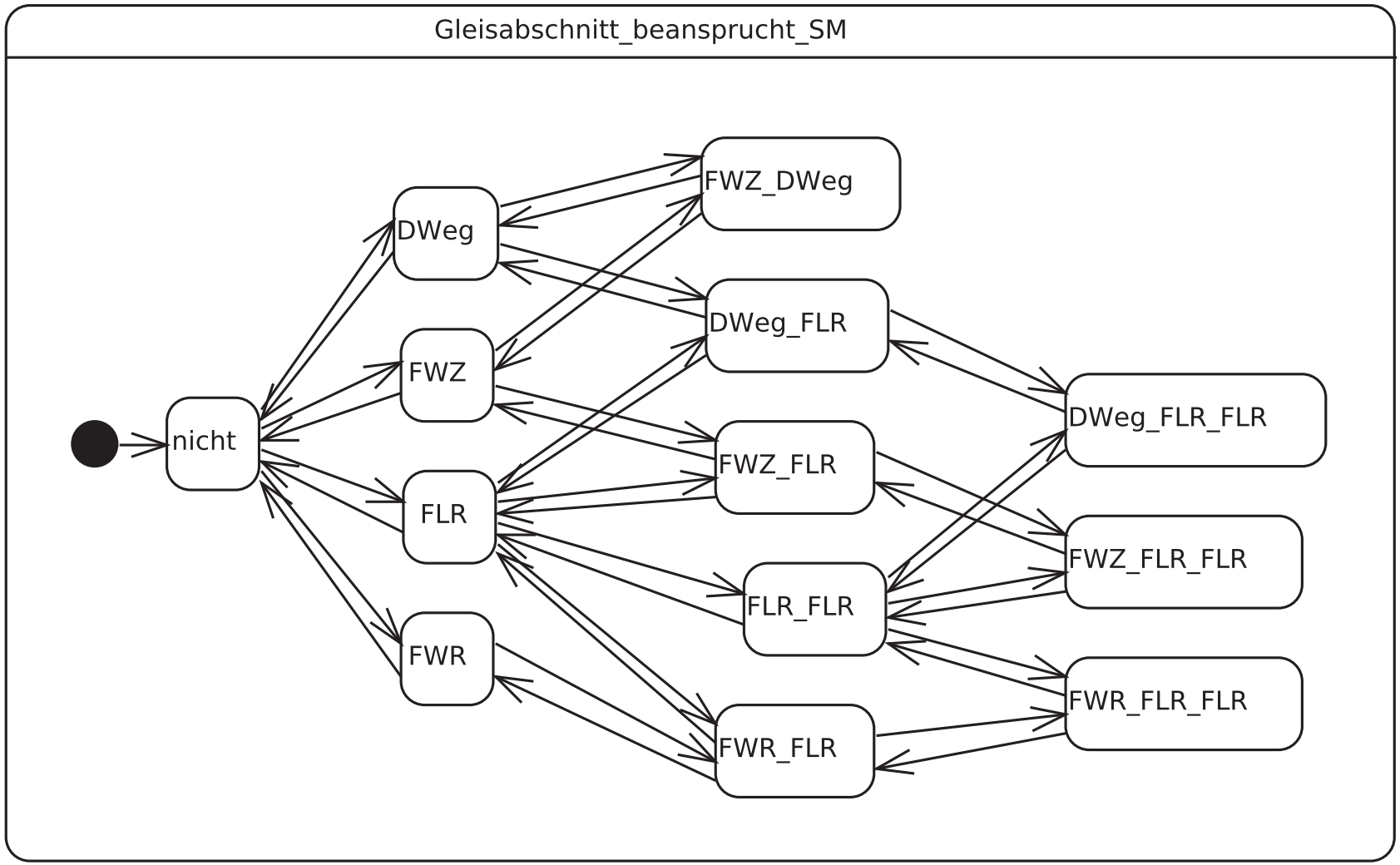


Our Approach ...



Dynamic Views . . .

. . . on state machines

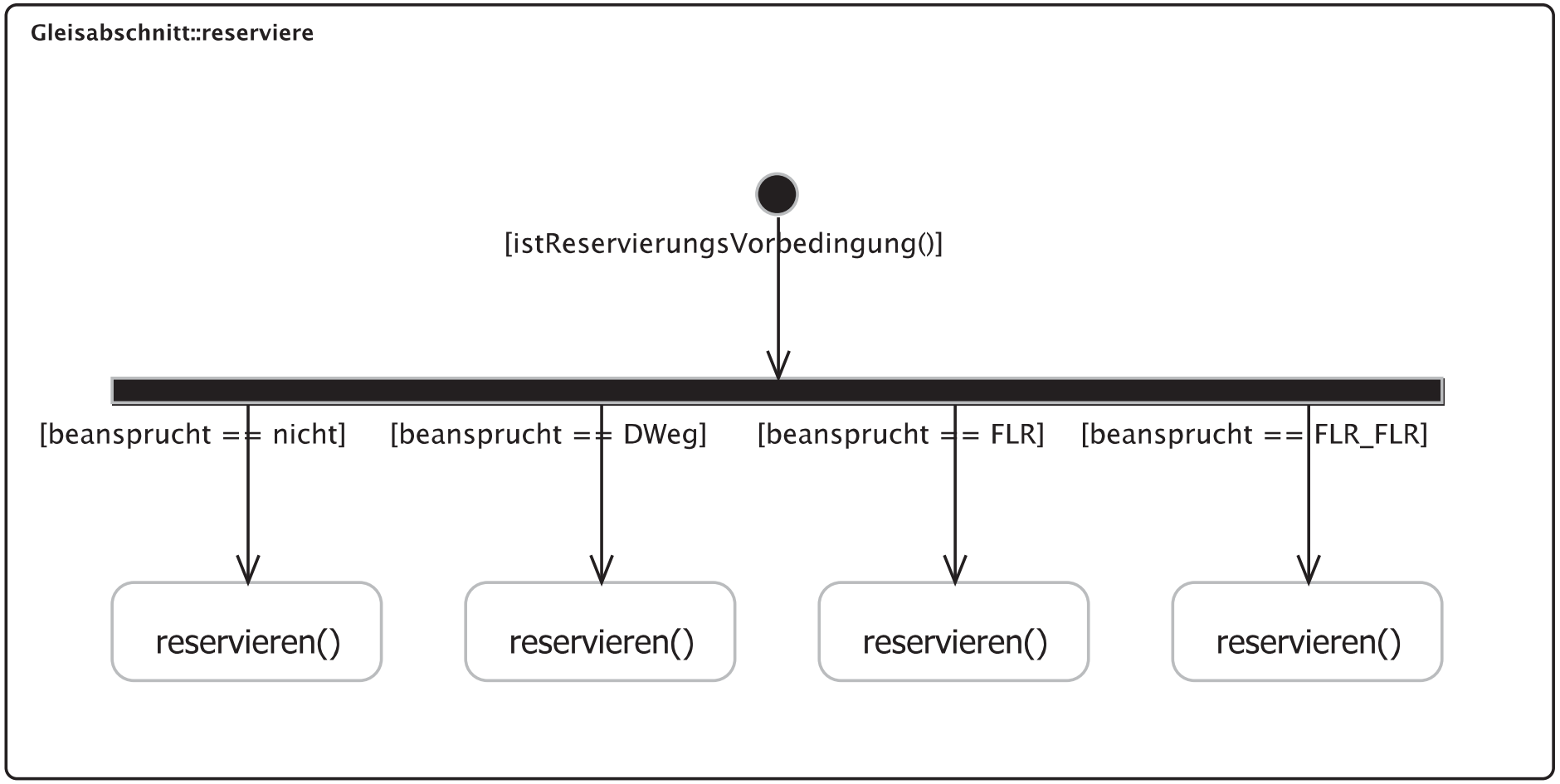


Recall: rule blocks

```
1  rule block reserviere references Gleisabschnitt {
2      rule graph
3      --> [istReservierungsVorbedingung()] {
4          -> [beansprucht == nicht] / {
5              reservieren()
6          };
7          -> [beansprucht == DWeg] / {
8              reservieren()
9          };
10         -> [beansprucht == FLR] / {
11             reservieren()
12         };
13         -> [beansprucht == FLR_FLR] / {
14             reservieren()
15         };
16     }
17 }
```

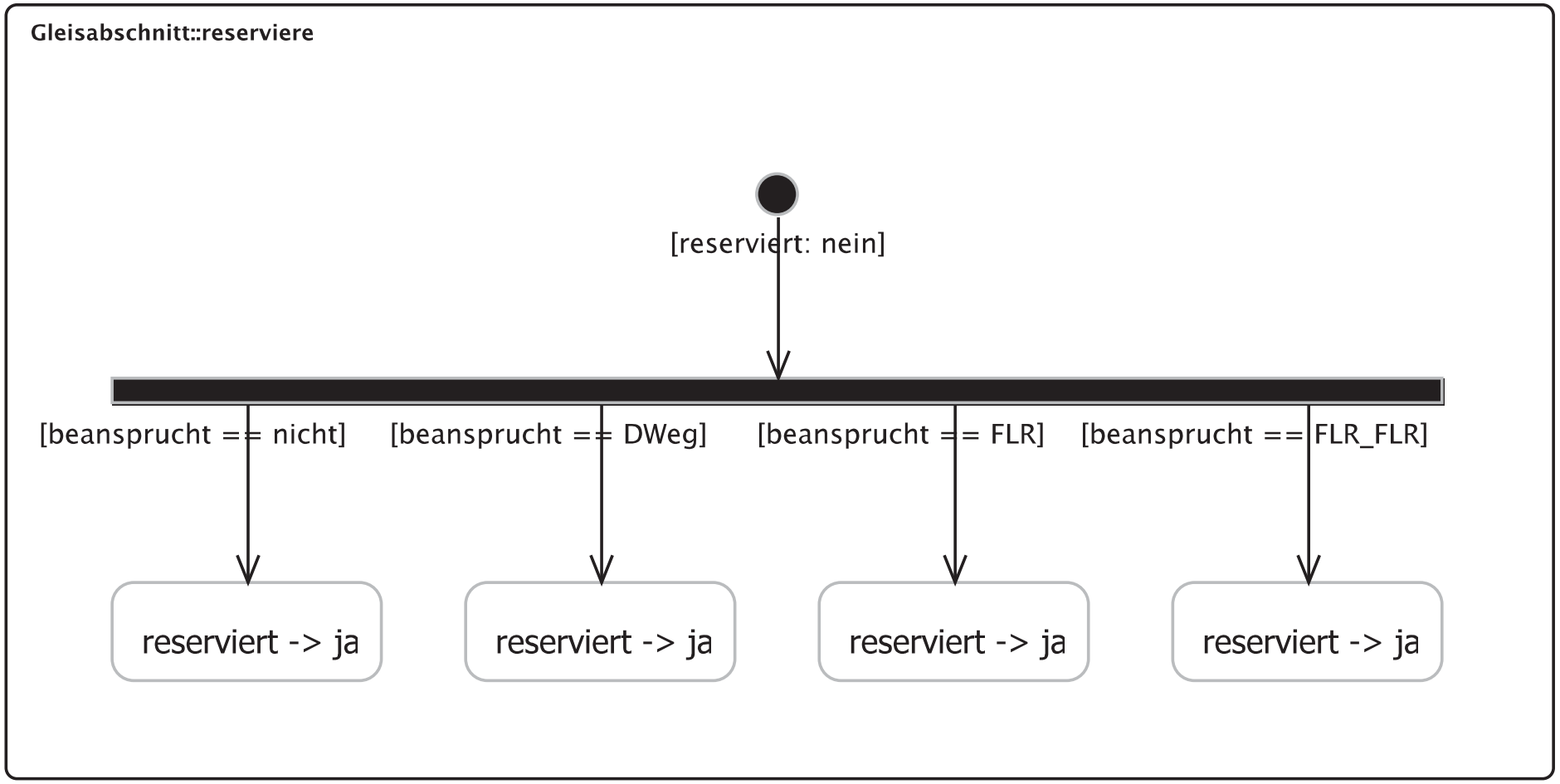
Dynamic Views . . .

. . . on rule blocks



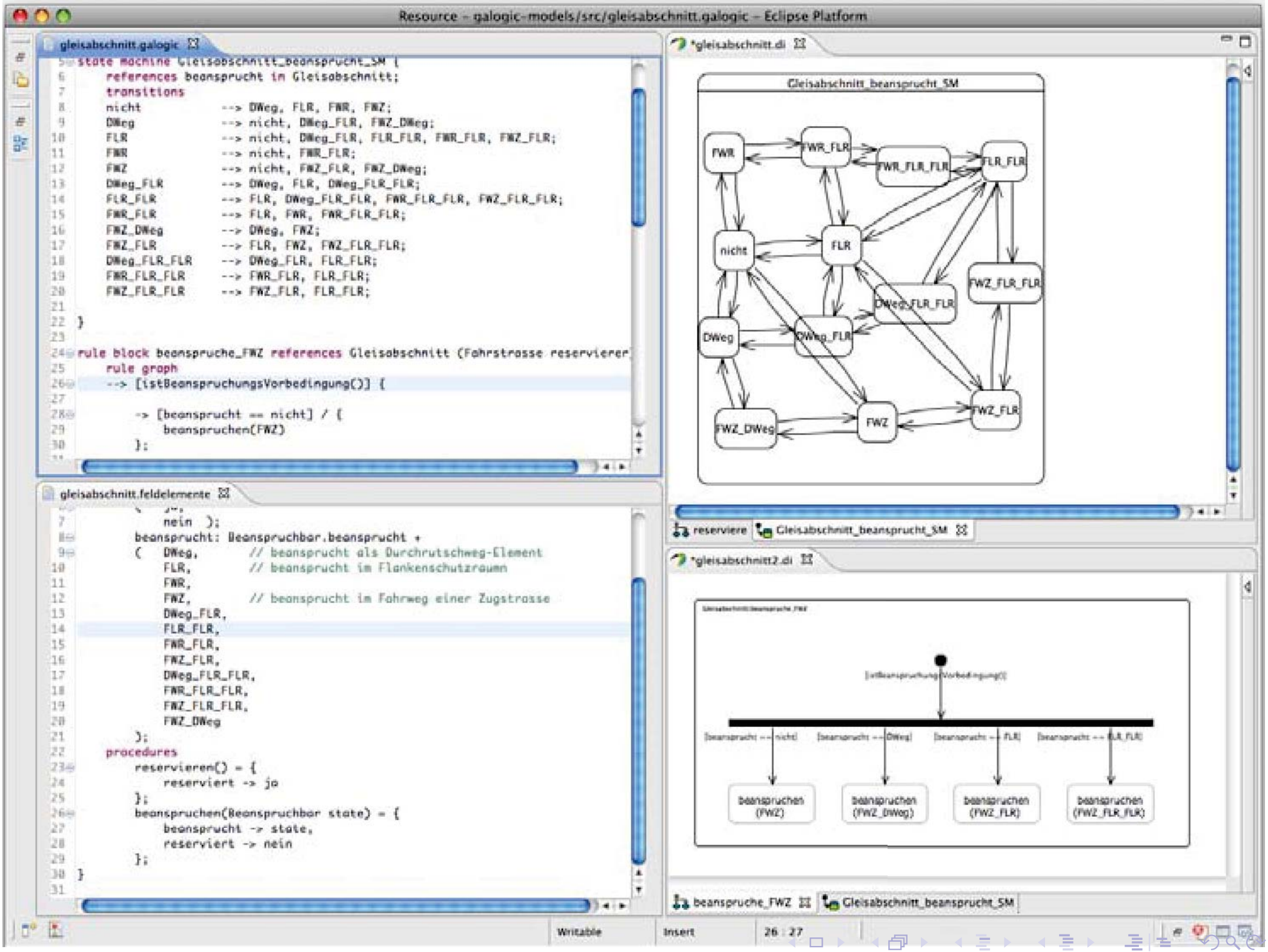
Dynamic Views . . .

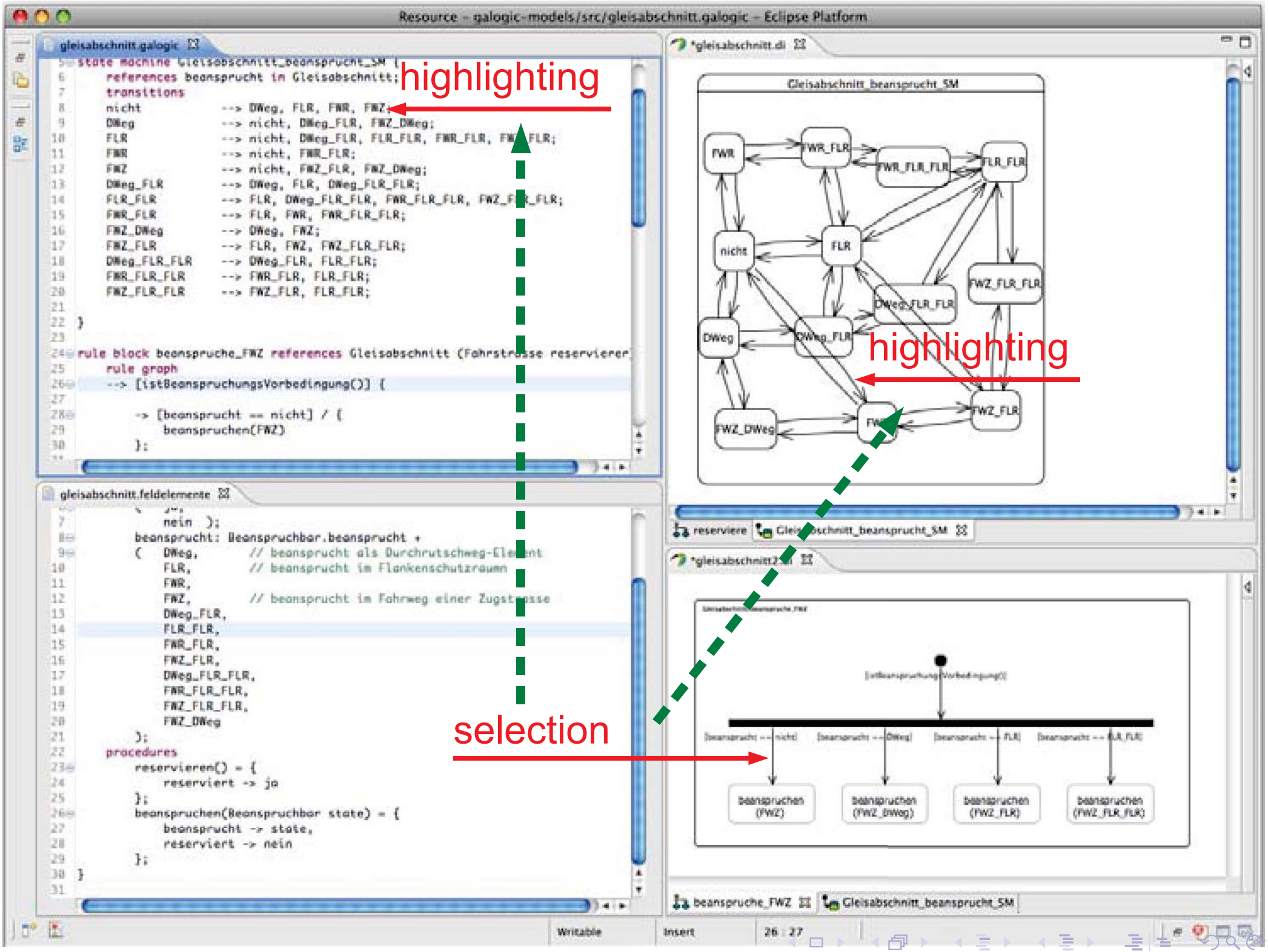
. . . on rule blocks - resolved

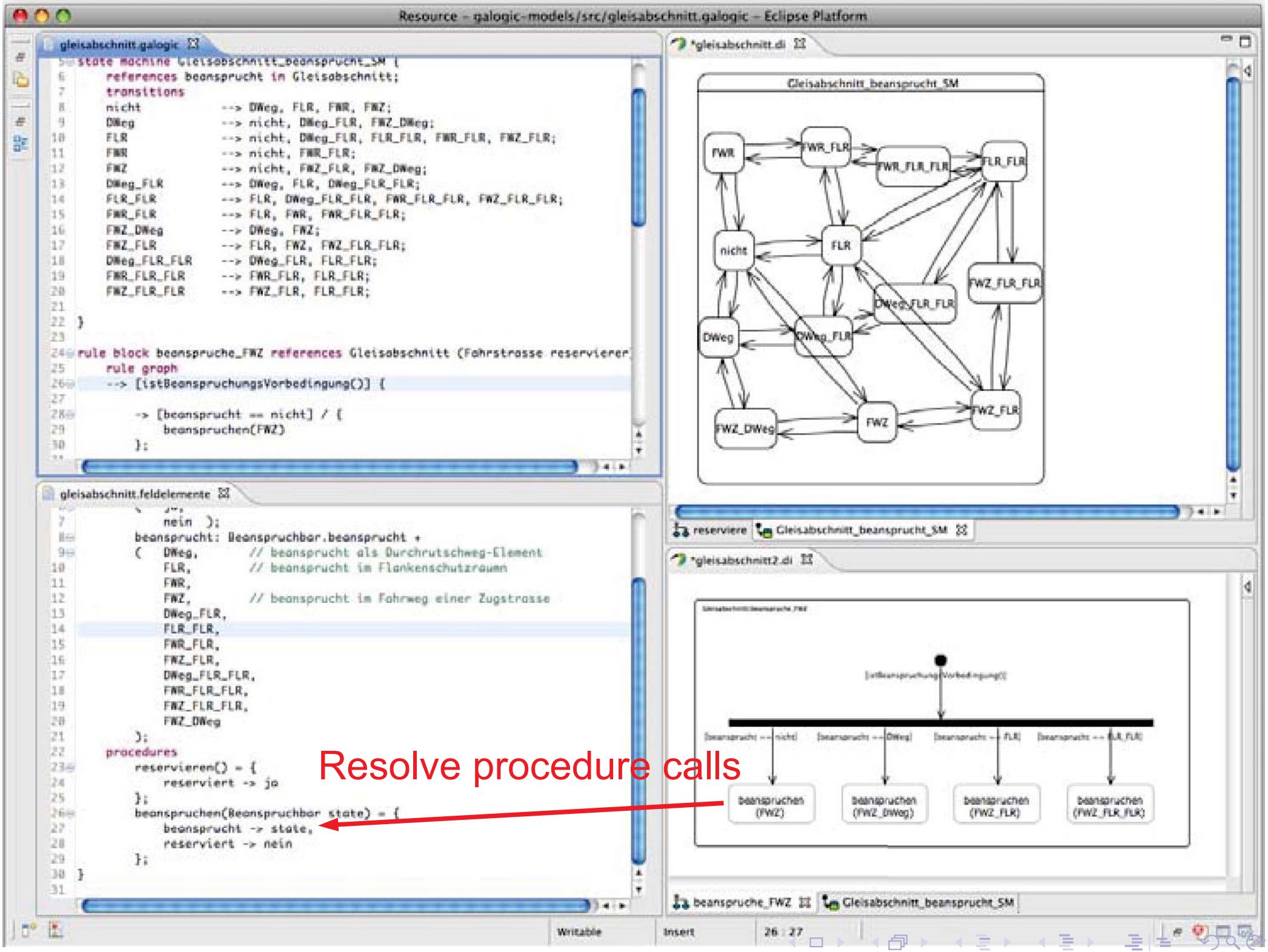


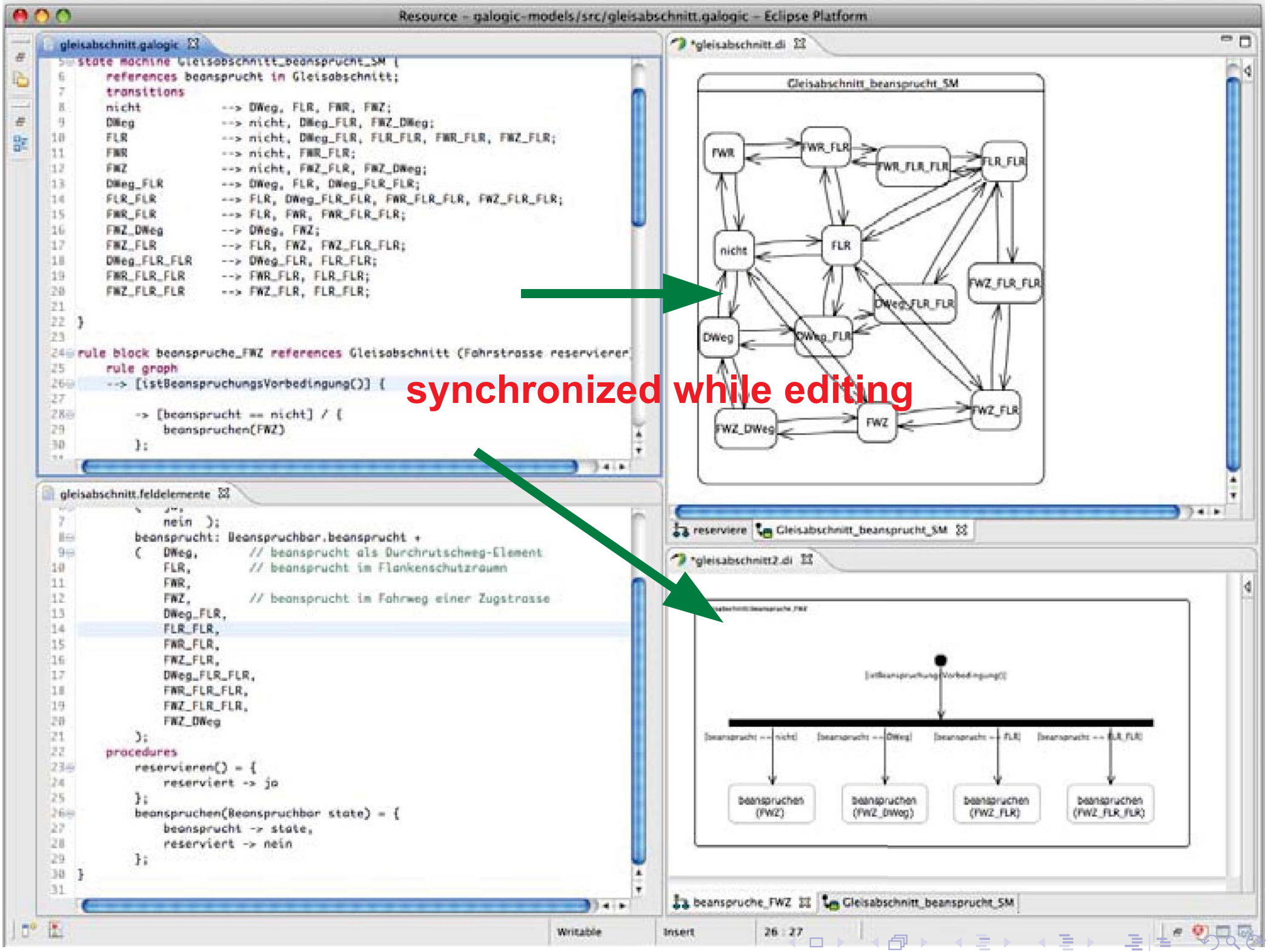
View Management

- Provide a (graphical) view of the part under development
  - allow to resolve called procedures, ...
- Support interactive browsing
  - licking on an element reveals its declaration / origin
- Find mutual references of state transitions and rule blocks
  - compute and highlight (un-) covered transitions
- Synchronize open views if model has been changed
  - without any user request
- Focus on context in simulation and testing taks
- ...


Resource - galogic-models/src/gleisabschnitt.galogic - Eclipse Platform
gleisabschnitt.galogic
state machine Gleisabschnitt_beansprucht_SM {
references beansprucht in Gleisabschnitt;
transitions
nicht --> DWeg, FLR, FWR, FWZ;
DWeg --> nicht, DWeg_FLR, FWZ_DWeg;
FLR --> nicht, DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR;
FWR --> nicht, FWR_FLR;
FWZ --> nicht, FWZ_FLR, FWZ_DWeg;
DWeg_FLR --> DWeg, FLR, DWeg_FLR_FLR;
FLR_FLR --> FLR, DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR;
FWR_FLR --> FLR, FWR, FWR_FLR_FLR;
FWZ_DWeg --> DWeg, FWZ;
FWZ_FLR --> FLR, FWZ, FWZ_FLR_FLR;
DWeg_FLR_FLR --> DWeg_FLR, FLR_FLR;
FWR_FLR_FLR --> FWR_FLR, FLR_FLR;
FWZ_FLR_FLR --> FWZ_FLR, FLR_FLR;
}
rule block beanspruche_FWZ references Gleisabschnitt (Fahrstrasse reservierer
rule graph
--> [istBeanspruchungsVorbedingung()] {
-> [beansprucht == nicht] / {
beanspruchen(FWZ)
};
*gleisabschnitt.di
Gleisabschnitt_beansprucht_SM
FWR
FWR_FLR
FWR_FLR_FLR
FLR_FLR
nicht
FLR
FWZ_FLR_FLR
DWeg_FLR_FLR
DWeg
DWeg_FLR
FWZ_FLR
FWZ_DWeg
FWZ
reserviere
Gleisabschnitt_beansprucht_SM
gleisabschnitt.feldelemente
nein);
beansprucht: Beanspruchbar.beansprucht +
(DWeg, // beansprucht als Durchrutschweg-Element
FLR, // beansprucht im Flankenschutzraum
FWR,
FWZ, // beansprucht im Fahrweg einer Zugstrasse
DWeg_FLR,
FLR_FLR,
FWR_FLR,
FWZ_FLR,
DWeg_FLR_FLR,
FWR_FLR_FLR,
FWZ_FLR_FLR,
FWZ_DWeg
);
procedures
reservieren() = {
reserviert -> ja
};
beanspruchen(Beanspruchbar state) = {
beansprucht -> state,
reserviert -> nein
};
}
*gleisabschnitt2.di
[istBeanspruchungsVorbedingung()]
[beansprucht == nicht]
[beansprucht == DWeg]
[beansprucht == FLR]
[beansprucht == FLR_FLR]
beanspruchen (FWZ)
beanspruchen (FWZ_DWeg)
beanspruchen (FWZ_FLR)
beanspruchen (FWZ_FLR_FLR)
beanspruche_FWZ
Gleisabschnitt_beansprucht_SM
Writable
Insert
26 : 27


Resource - galogic-models/src/gleisabschnitt.galogic - Eclipse Platform
gleisabschnitt.galogic
state machine Gleisabschnitt_beansprucht_SM {
references beansprucht in Gleisabschnitt;
transitions
nicht --> DWeg, FLR, FWR, FWZ;
DWeg --> nicht, DWeg_FLR, FWZ_DWeg;
FLR --> nicht, DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR;
FWR --> nicht, FWR_FLR;
FWZ --> nicht, FWZ_FLR, FWZ_DWeg;
DWeg_FLR --> DWeg, FLR, DWeg_FLR_FLR;
FLR_FLR --> FLR, DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR;
FWR_FLR --> FLR, FWR, FWR_FLR_FLR;
FWZ_DWeg --> DWeg, FWZ;
FWZ_FLR --> FLR, FWZ, FWZ_FLR_FLR;
DWeg_FLR_FLR --> DWeg_FLR, FLR_FLR;
FWR_FLR_FLR --> FWR_FLR, FLR_FLR;
FWZ_FLR_FLR --> FWZ_FLR, FLR_FLR;
}
rule block beanspruche_FWZ references Gleisabschnitt (Fahrstrasse reservierer
rule graph
--> [istBeanspruchungsVorbedingung()] {
-> [beansprucht == nicht] / {
beanspruchen(FWZ)
};
highlighting
gleisabschnitt.feldelemente
nein);
beansprucht: Beanspruchbar.beansprucht +
(DWeg, // beansprucht als Durchrutschweg-Element
FLR, // beansprucht im Flankenschutzraum
FWR,
FWZ, // beansprucht im Fahrweg einer Zugstrasse
DWeg_FLR,
FLR_FLR,
FWR_FLR,
FWZ_FLR,
DWeg_FLR_FLR,
FWR_FLR_FLR,
FWZ_FLR_FLR,
FWZ_DWeg
);
procedures
reservieren() = {
reserviert -> ja
};
beanspruchen(Beanspruchbar state) = {
beansprucht -> state,
reserviert -> nein
};
}
selection
*gleisabschnitt.di
Gleisabschnitt_beansprucht_SM
FWR
FWR_FLR
FWR_FLR_FLR
FLR_FLR
nicht
FLR
FWZ_FLR_FLR
DWeg_FLR_FLR
DWeg
DWeg_FLR
FWZ_DWeg
FWZ
FWZ_FLR
highlighting
reserviere
Gleisabschnitt_beansprucht_SM
*gleisabschnitt2.di
beanspruchen (FWZ)
beanspruchen (FWZ_DWeg)
beanspruchen (FWZ_FLR)
beanspruchen (FWZ_FLR_FLR)
beanspruche_FWZ
Gleisabschnitt_beansprucht_SM
Writable
Insert
26 : 27


Resource - galogic-models/src/gleisabschnitt.galogic - Eclipse Platform
gleisabschnitt.galogic
state machine Gleisabschnitt_beansprucht_SM {
references beansprucht in Gleisabschnitt;
transitions
nicht --> DWeg, FLR, FWR, FWZ;
DWeg --> nicht, DWeg_FLR, FWZ_DWeg;
FLR --> nicht, DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR;
FWR --> nicht, FWR_FLR;
FWZ --> nicht, FWZ_FLR, FWZ_DWeg;
DWeg_FLR --> DWeg, FLR, DWeg_FLR_FLR;
FLR_FLR --> FLR, DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR;
FWR_FLR --> FLR, FWR, FWR_FLR_FLR;
FWZ_DWeg --> DWeg, FWZ;
FWZ_FLR --> FLR, FWZ, FWZ_FLR_FLR;
DWeg_FLR_FLR --> DWeg_FLR, FLR_FLR;
FWR_FLR_FLR --> FWR_FLR, FLR_FLR;
FWZ_FLR_FLR --> FWZ_FLR, FLR_FLR;
}
rule block beanspruche_FWZ references Gleisabschnitt (Fahrstrasse reservierer
rule graph
--> [istBeanspruchungsVorbedingung()] {
-> [beansprucht == nicht] / {
beanspruchen(FWZ)
};
*gleisabschnitt.di
Gleisabschnitt_beansprucht_SM
FWR
FWR_FLR
FWR_FLR_FLR
FLR_FLR
nicht
FLR
FWZ_FLR_FLR
DWeg_FLR_FLR
DWeg
DWeg_FLR
FWZ_FLR
FWZ_DWeg
FWZ
reserviere
Gleisabschnitt_beansprucht_SM
gleisabschnitt.feldelemente
nein);
beansprucht: Beanspruchbar.beansprucht +
(DWeg, // beansprucht als Durchrutschweg-Element
FLR, // beansprucht im Flankenschutzraum
FWR,
FWZ, // beansprucht im Fahrweg einer Zugstrasse
DWeg_FLR,
FLR_FLR,
FWR_FLR,
FWZ_FLR,
DWeg_FLR_FLR,
FWR_FLR_FLR,
FWZ_FLR_FLR,
FWZ_DWeg
);
procedures
reservieren() = {
reserviert -> ja
};
beanspruchen(Beanspruchbar state) = {
beansprucht -> state,
reserviert -> nein
};
}
Resolve procedure calls
*gleisabschnitt2.di
beanspruchen (FWZ)
beanspruchen (FWZ_DWeg)
beanspruchen (FWZ_FLR)
beanspruchen (FWZ_FLR_FLR)
beanspruche_FWZ
Gleisabschnitt_beansprucht_SM
Writable
Insert
26 : 27

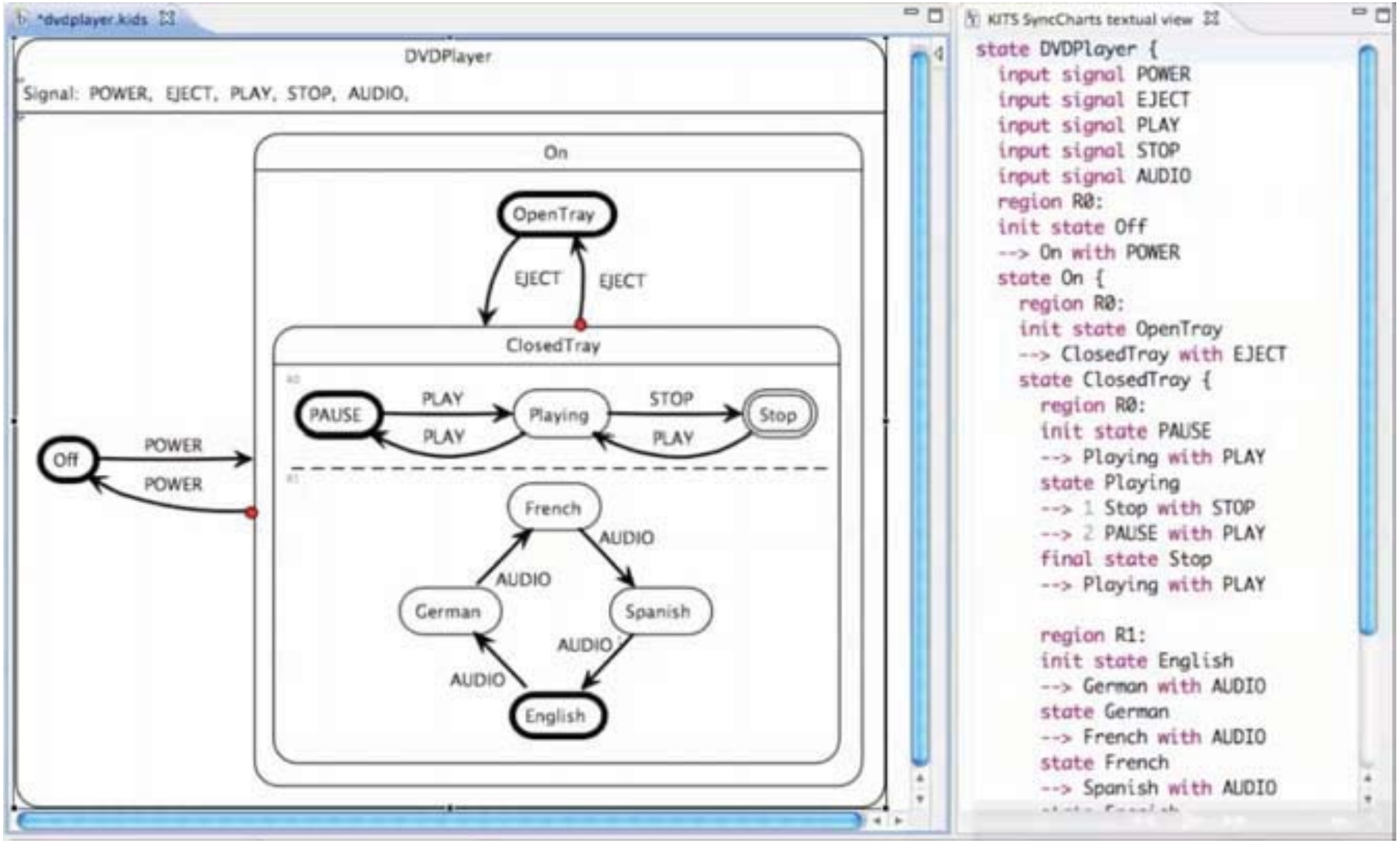

Resource - galogic-models/src/gleisabschnitt.galogic - Eclipse Platform
gleisabschnitt.galogic
state machine Gleisabschnitt_beansprucht_SM {
references beansprucht in Gleisabschnitt;
transitions
nicht --> DWeg, FLR, FWR, FWZ;
DWeg --> nicht, DWeg_FLR, FWZ_DWeg;
FLR --> nicht, DWeg_FLR, FLR_FLR, FWR_FLR, FWZ_FLR;
FWR --> nicht, FWR_FLR;
FWZ --> nicht, FWZ_FLR, FWZ_DWeg;
DWeg_FLR --> DWeg, FLR, DWeg_FLR_FLR;
FLR_FLR --> FLR, DWeg_FLR_FLR, FWR_FLR_FLR, FWZ_FLR_FLR;
FWR_FLR --> FLR, FWR, FWR_FLR_FLR;
FWZ_DWeg --> DWeg, FWZ;
FWZ_FLR --> FLR, FWZ, FWZ_FLR_FLR;
DWeg_FLR_FLR --> DWeg_FLR, FLR_FLR;
FWR_FLR_FLR --> FWR_FLR, FLR_FLR;
FWZ_FLR_FLR --> FWZ_FLR, FLR_FLR;
}
rule block beanspruche_FWZ references Gleisabschnitt (Fahrstrasse reservierer
rule graph
--> [istBeanspruchungsVorbedingung()] {
-> [beansprucht == nicht] / {
beanspruchen(FWZ)
};
*gleisabschnitt.di
Gleisabschnitt_beansprucht_SM
FWR
FWR_FLR
FWR_FLR_FLR
FLR_FLR
nicht
FLR
FWZ_FLR_FLR
DWeg_FLR_FLR
DWeg
DWeg_FLR
FWZ_FLR
FWZ_DWeg
FWZ
reserviere
Gleisabschnitt_beansprucht_SM
synchronized while editing
gleisabschnitt.feldelemente
nein);
beansprucht: Beanspruchbar.beansprucht +
(DWeg, // beansprucht als Durchrutschweg-Element
FLR, // beansprucht im Flankenschutzraum
FWR,
FWZ, // beansprucht im Fahrweg einer Zugstrasse
DWeg_FLR,
FLR_FLR,
FWR_FLR,
FWZ_FLR,
DWeg_FLR_FLR,
FWR_FLR_FLR,
FWZ_FLR_FLR,
FWZ_DWeg
);
procedures
reservieren() = {
reserviert -> ja
};
beanspruchen(Beanspruchbar state) = {
beansprucht -> state,
reserviert -> nein
};
}
*gleisabschnitt2.di
beanspruchen (FWZ)
beanspruchen (FWZ_DWeg)
beanspruchen (FWZ_FLR)
beanspruchen (FWZ_FLR_FLR)
beanspruche_FWZ
Gleisabschnitt_beansprucht_SM
Writable
Insert
26 : 27

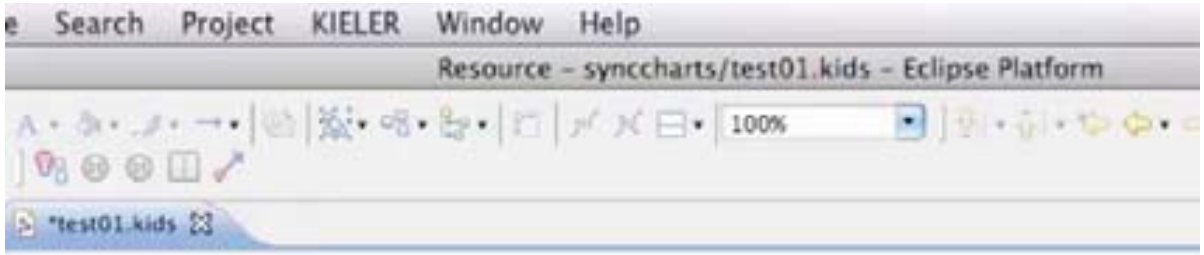
Conclusion

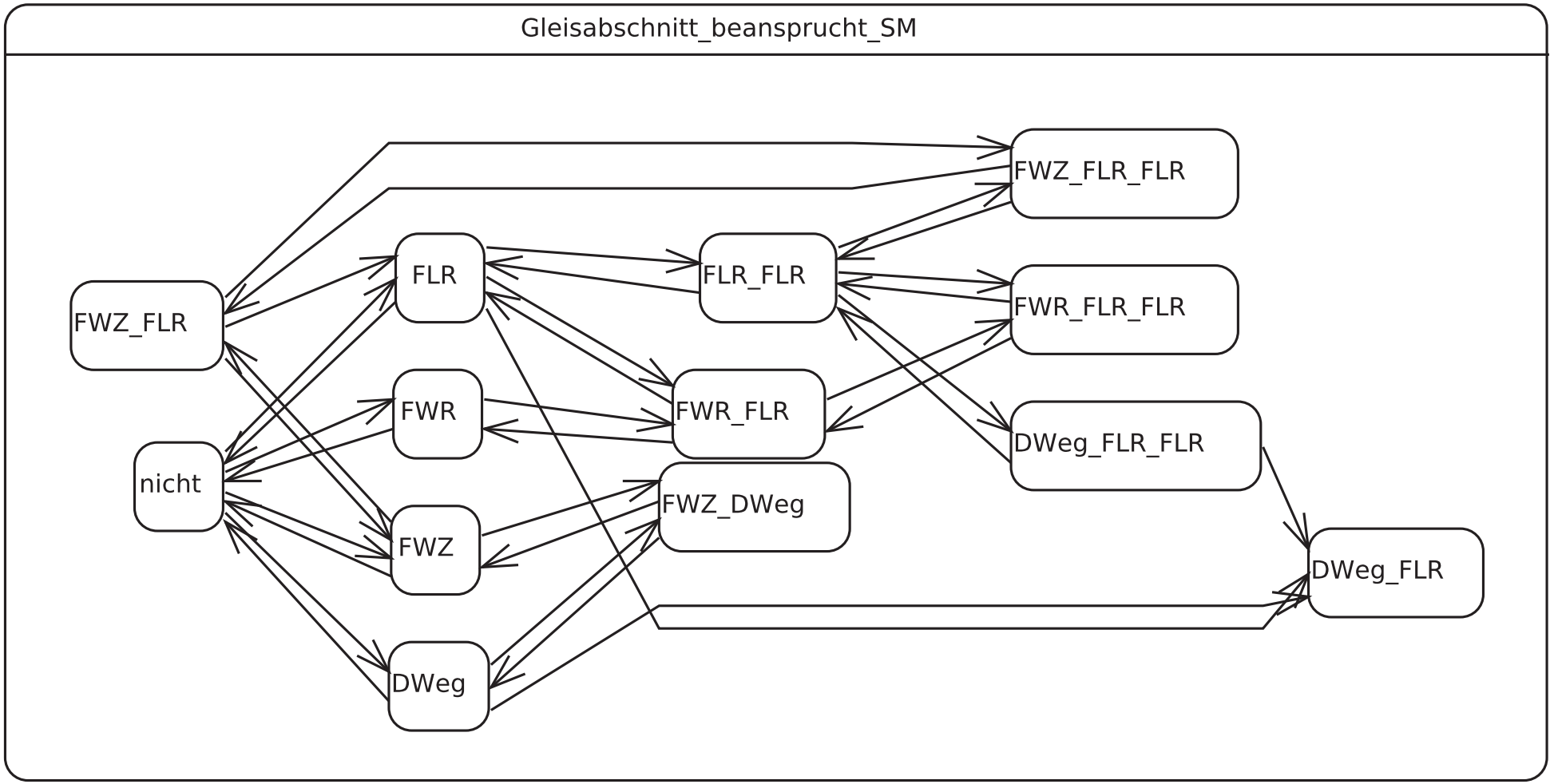
What did I talk about?

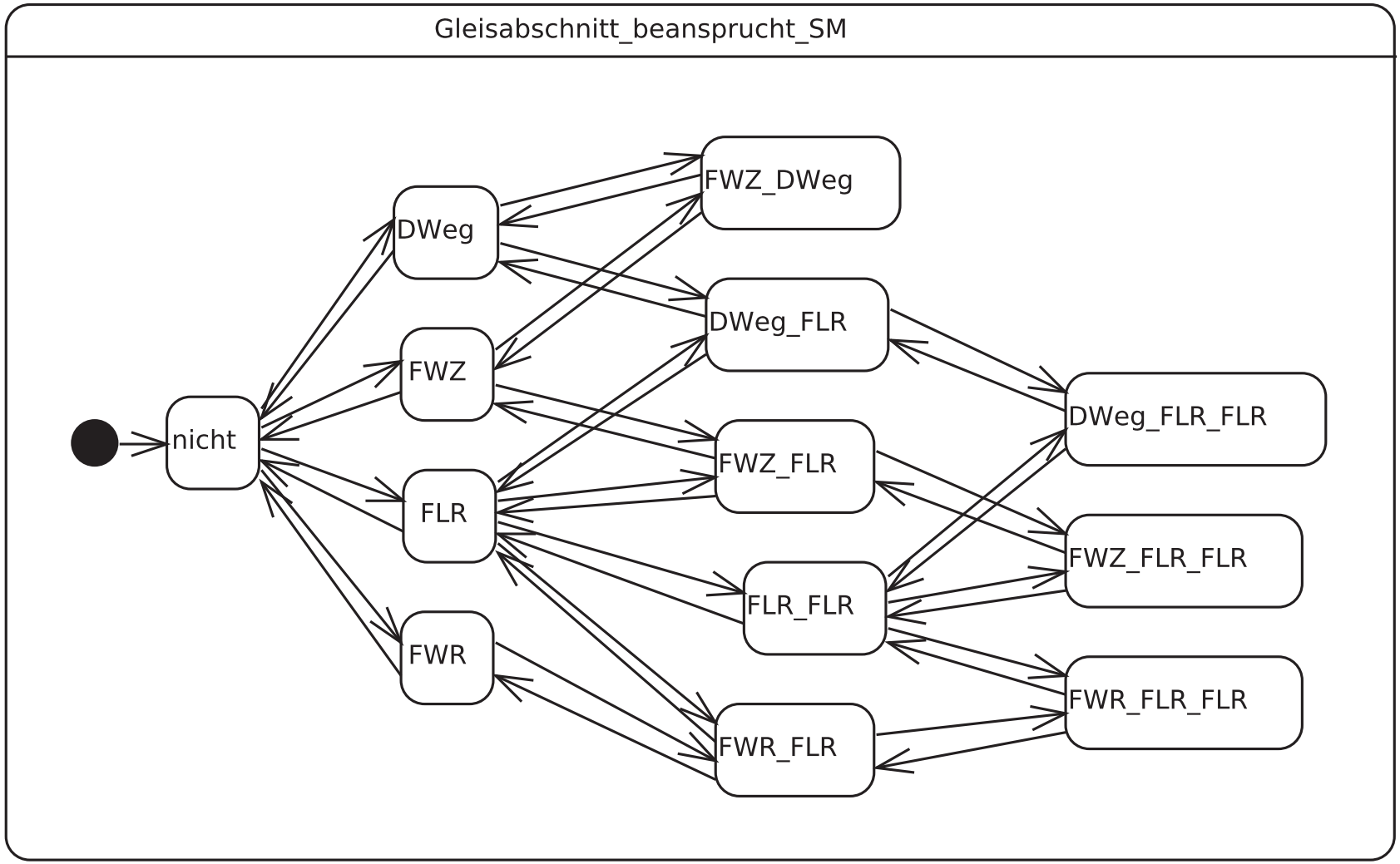
- shortly introduced the MENGES project
- outlined the deployment of textual languages
- motivated benefits/downsides of textual & graphical notations
- sketched an approach on how to integrate them
  resulting in much more abilities of handling models

APPENDIX


*dvdplayer.kids
DVDPlayer
Signal: POWER, EJECT, PLAY, STOP, AUDIO,
On
OpenTray
EJECT
EJECT
ClosedTray
PAUSE
PLAY
Playing
STOP
Stop
PLAY
PLAY
Off
POWER
POWER
French
AUDIO
AUDIO
German
Spanish
AUDIO
AUDIO
English
KITS SyncCharts textual view
state DVDPlayer {
input signal POWER
input signal EJECT
input signal PLAY
input signal STOP
input signal AUDIO
region R0:
init state Off
--> On with POWER
state On {
region R0:
init state OpenTray
--> ClosedTray with EJECT
state ClosedTray {
region R0:
init state PAUSE
--> Playing with PLAY
state Playing
--> 1 Stop with STOP
--> 2 PAUSE with PLAY
final state Stop
--> Playing with PLAY
region R1:
init state English
--> German with AUDIO
state German
--> French with AUDIO
state French
--> Spanish with AUDIO


Search
Project
KIELER
Window
Help
Resource – syncharts/test01.kids – Eclipse Platform
100%
*test01.kids


Gleisabschnitt_beansprucht_SM
FWZ_FLR
nicht
FLR
FWR
FWZ
DWeg
FLR_FLR
FWR_FLR
FWZ_DWeg
FWZ_FLR_FLR
FWR_FLR_FLR
DWeg_FLR_FLR
DWeg_FLR


Gleisabschnitt_beansprucht_SM
nicht
DWeg
FWZ
FLR
FWR
FWZ_DWeg
DWeg_FLR
FWZ_FLR
FLR_FLR
FWR_FLR
DWeg_FLR_FLR
FWZ_FLR_FLR
FWR_FLR_FLR