



Prüfung von „Traceability Links“-Workshop

Darmstadt, 7.12.2007



2

- 10.00 Begrüßung und Vorstellung der Teilnehmer
- 10.30 Erörterung der Entwicklungsmethoden
- 11.30 Mittagspause
- 12.15 Toolnet-Vorstellung (TU Darmstadt)
- 13.30 OOSE/RUP (Matthias Riebisch, TU Ilmenau)
Diskussion von Prüfungen und Bedingungen
- 14.50 Kaffeepause
- 15.00 WOM-Vorstellung (Tobias Zimmermann, EADS)
Diskussion von Prüfungen und Bedingungen
- 16.30 Festlegung der nächsten Schritte
- 17.00 Abschluss des Workshops



Integration and Automatic Maintenance of Traceability Links between System Development Tools

Prof. Dr. rer. nat. Andy Schürr

Darmstadt University of Technology

Dept. of Electrical Engineering & Information Technology

(Associate Member of Department of Computer Science)



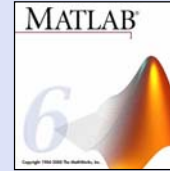
The Running Example



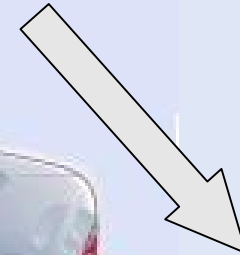
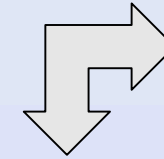
4

www.es.tu-darmstadt.de

System requirements



SW-Functionality



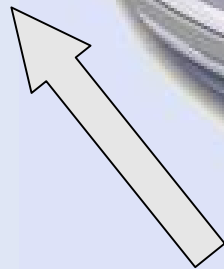
Product Data



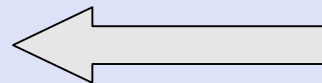
Automotive System Development



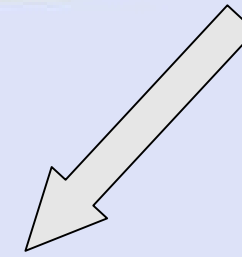
HW-Design



Function Test



ECU-Housing





Outline of Presentation

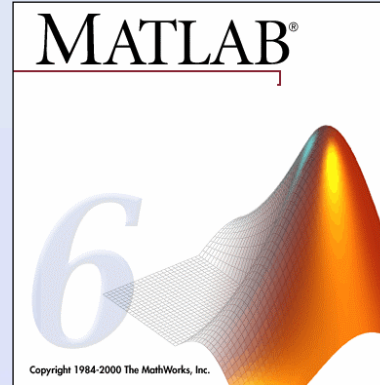
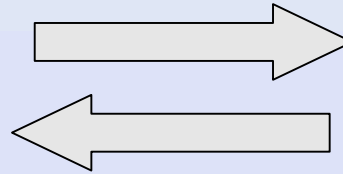


5



DOORS
Requirements Eng.
Tool

3. synchronize

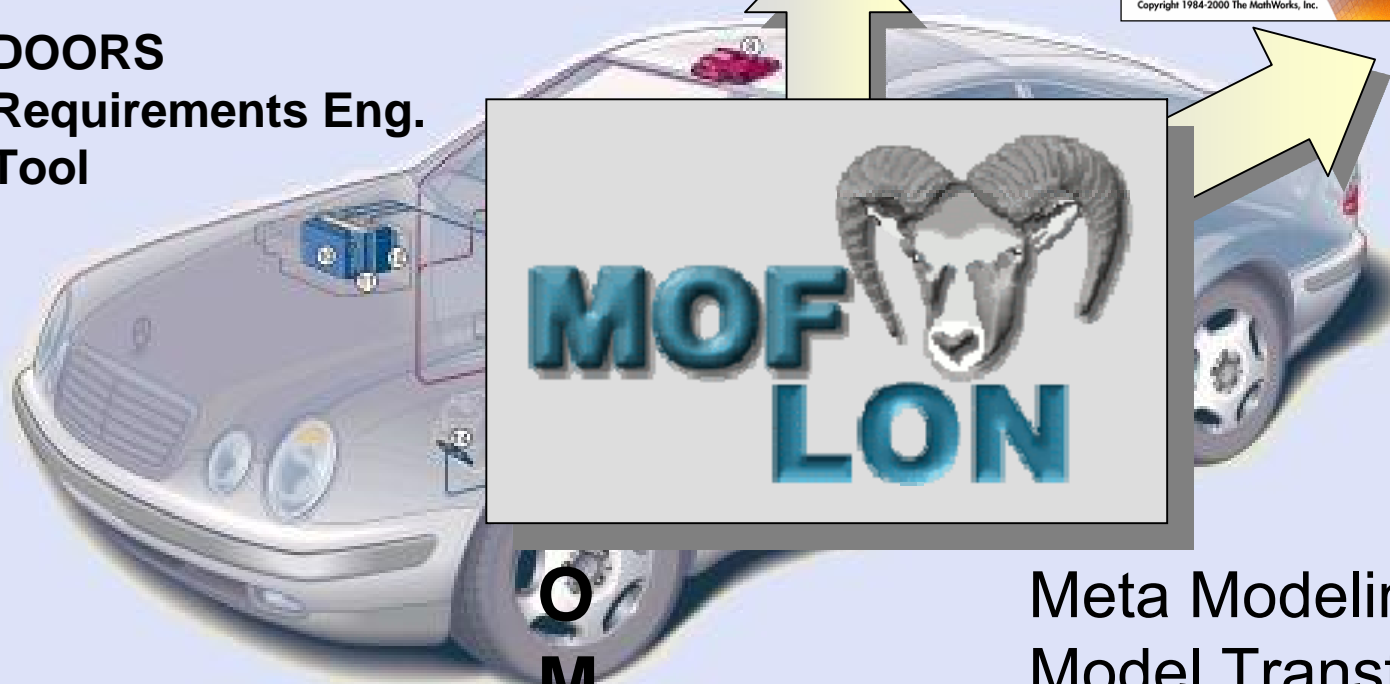


1. analyze



2. repair

Simulink &
Stateflow



**O
M
G**

Meta Modeling &
Model Transformation
Techniques and Tools



1. The MATE Project

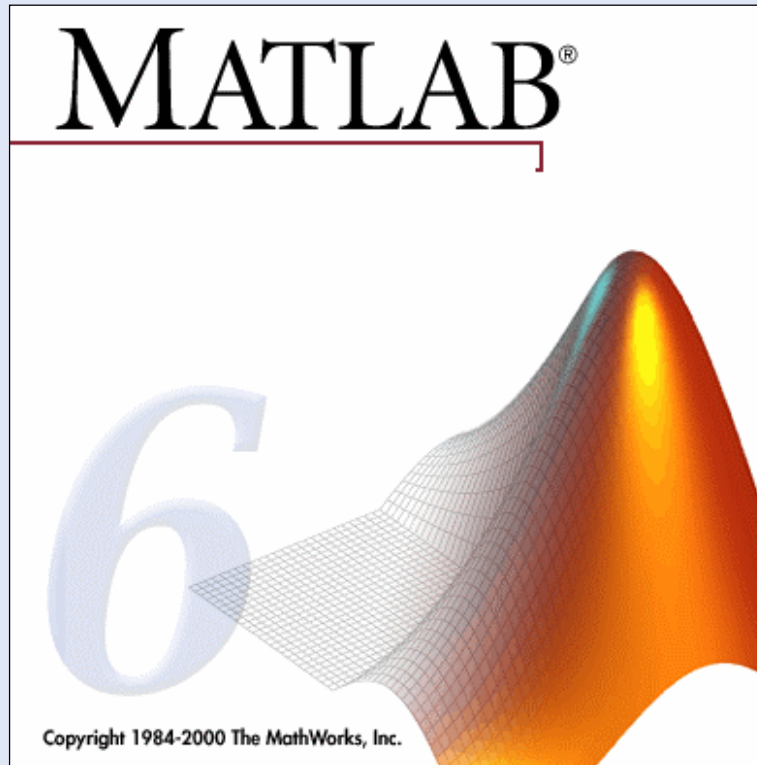
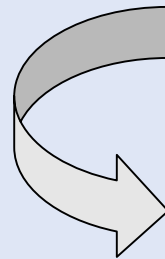
(MATLAB Simulink & Stateflow Analysis and Transformation Environment)



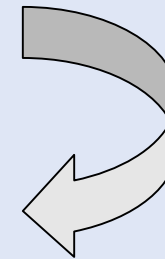
(DaimlerChrysler, Model Engineering Solution,
TU Darmstadt, Univ. Kassel, Paderborn, Siegen)



Analyze



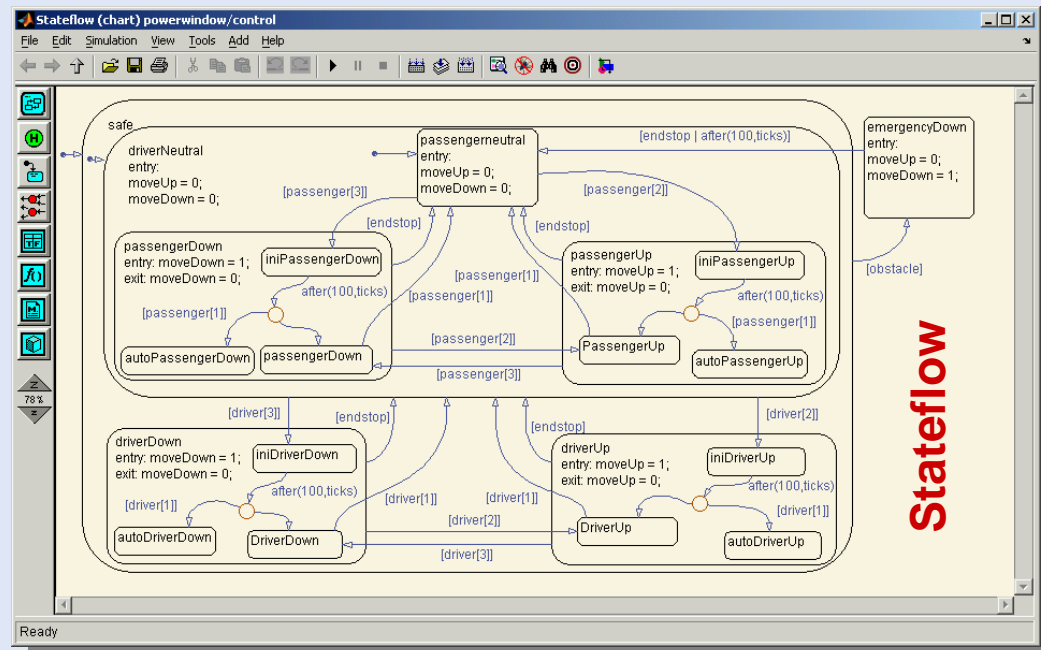
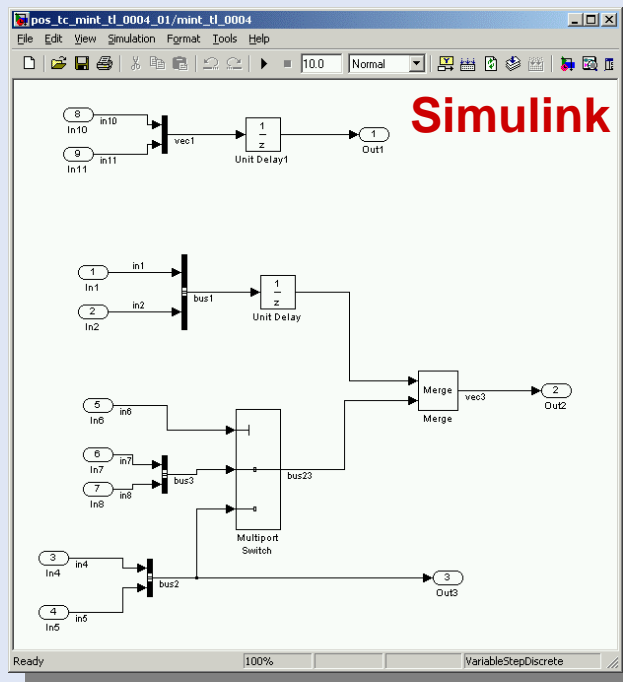
Repair



Simulink & Stateflow



- model-driven development of electronic control unit (ECU) software
- specification of control algorithms
- executable block diagram and statechart models



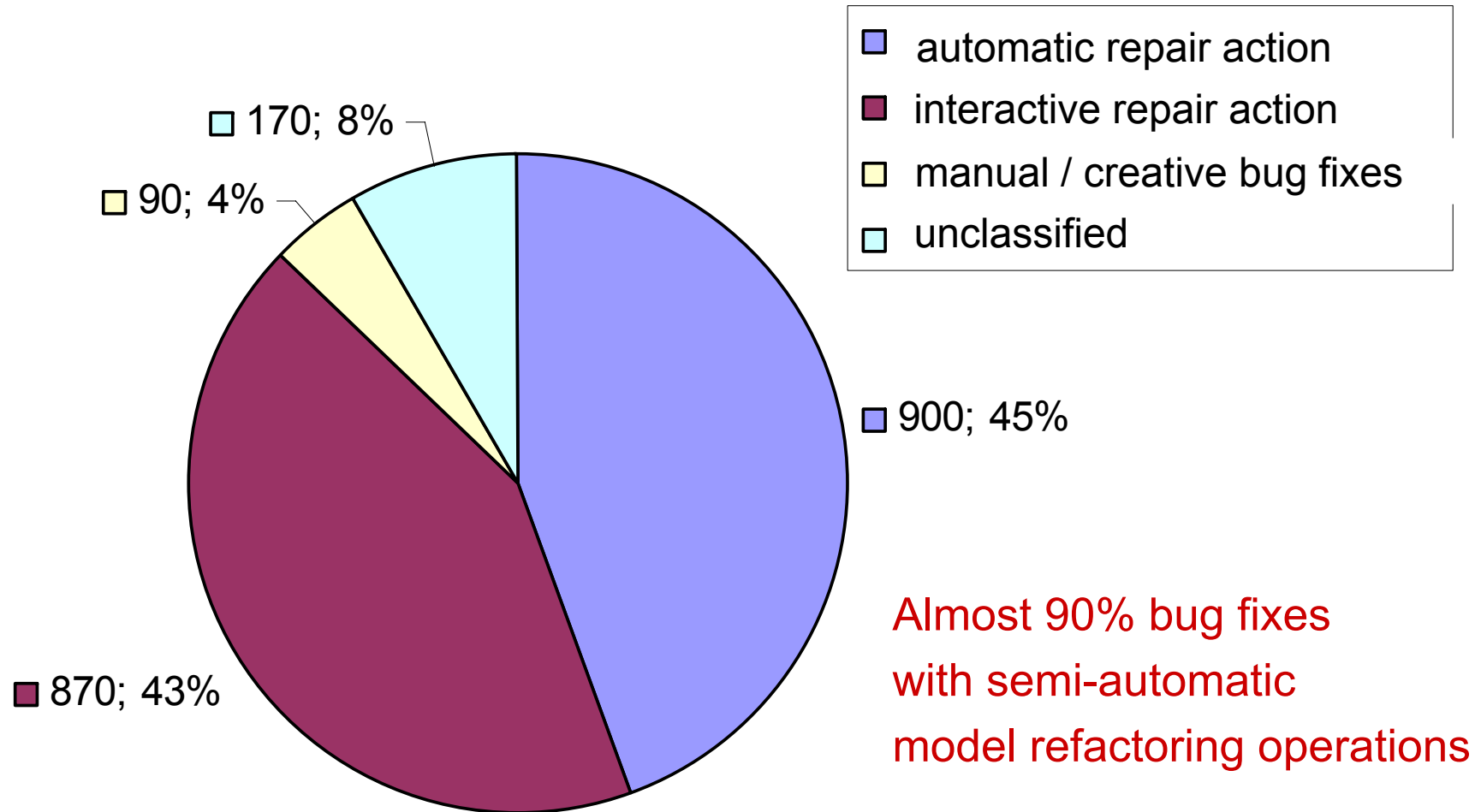


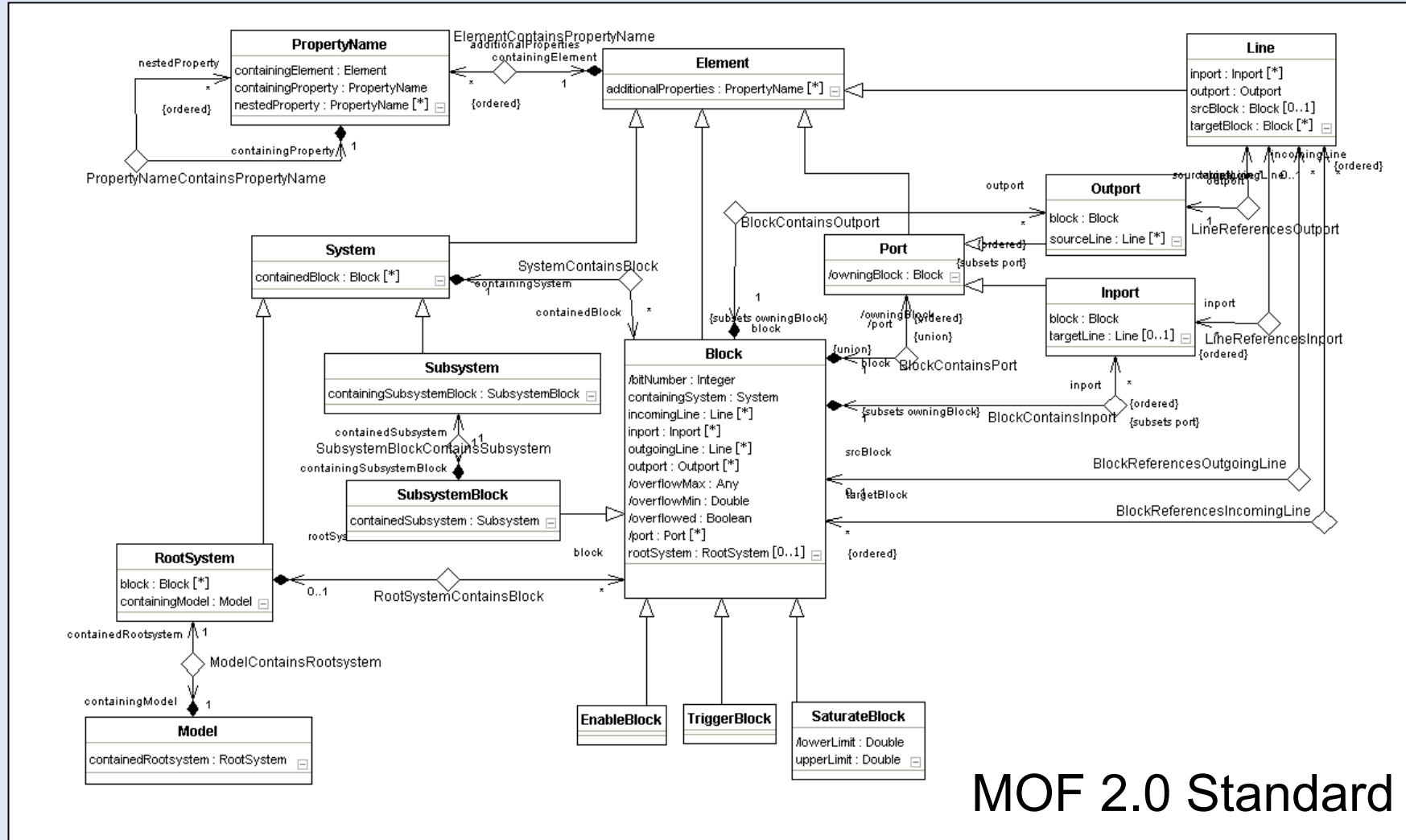
1600 min ≈ 27 h

	M1	M2	M3	M4	M5	M6	M7	M8	...	total
1. Model Metrics										
# blocks	182	24	104	1604	23	379	86	6		9308
# subsystems	17	3	6	189	1	37		1		902
# top level inputs / outputs	17 / 8	4 / 5	11 / 9	43 / 9	12 / 4	10 / 10	19 / 7	2 / 1		80 / 88
# inports / outputs	49 / 27	7 / 6	13 / 12	206 / 236	12 / 4	79 / 60	431 / 194	2 / 1		2061 / 1625
# constants	28	2	7	195	0	49	318	0		969
# Stateflow charts	5	1	5	1	1	0	2	1		18
# states in Stateflow	26	7	33	3	12	0	0	3		123
# transitions in Stateflow	50	30	102	6	21	0	25	53		329
cyclomatic complexity CYC _{MOD}	72	57	175	193	45	47	342	94		1284
subsystem depth	4	3	3	5	2	7	11	2		14
2. Review Duration										
total time for review (min)	186	48	112	67	48	74	472	35		1600
mean time for review (min)	93	24	56	34	24	25	157	12		-
3. Findings and Changes										
# unspecified	0	0	0	0	0	1	1	1		55
# questions	1	0	1	1	3	0	11	0		29
# editorial remarks	5	0	2	13	2	5	22	1		56
# uncritical remarks	5	2	5	0	3	0	5	1		24
# important remarks	1	3	0	0	0	3	28	3		58
# critical remarks	0	0	1	1	0	1	7	0		14
# findings (total)	12	5	9	15	8	10	74	6		236
# changes in model after review	7	4	8	13	7	7	49	2		146

Number of model modifications

Stürmer, I., Conrad, M., Fey, I., and Dörr, H.: **Experiences with Model and Autocode Reviews in Model-based Software Development.** Proc. of 3rd Intl. ICSE Workshop on Software Engineering for Automotive Systems (SEAS 2006), Shanghai, 2006.





MOF 2.0 Standard



Some Guidelines



12

www.es.tu-darmstadt.de

GL1: The name of a subsystem consists of characters, digits, and _ ; it starts with a character ...

GL2: The name of an Enable Block should be the same as the name of the related Enable Signal

GL3: Blocks may not possess unconnected Inports or unconnected Outports

GL4: ...



13

```
function f_block_h = guideline_2(system, cmd_s)
    top_h = get_param(bdroot,'Handle');
    f_block_h = [];
    subsys = get_param(get_param(find_system(top_h, 'BlockType',
        'EnablePort'), 'Parent'), 'Handle');
    for k=1:length(subsys)
        subsys_handle = get_param(subsys{k},'Handle');
        porth = get_param(subsys{k},'PortHandles');
        enable_port_name = get_param(porth.Enable,'Name');
        enableh = find_system(subsys{k},'SearchDepth',1,
            'BlockType','EnablePort');
        enable_block_name = get_param(enableh,'Name');
        if ~(strcmp(enable_port_name, enable_block_name))
            f_block_h = [f_block_h;subsys_handle];
        end
    end % for
end % function
```



14

```
if self.containedBlock
  ->exists(b:Block | b.ocllsTypeOf(EnableBlock) )
then
  self.containingSubsystemBlock.incomingLine
  ->select( line | line.dstPort = "enable" )
  ->collect(qualifiedName)
  ->intersection (self.containedBlock
    ->select(b:Block | b.ocllsTypeOf(EnableBlock))
    ->collect(qualifiedName) )
  ->notEmpty()
endif
```



Enable Signal and EnableBlock must have the same name!

Analyzer::analyseAndRepair_gl2 (enableBlock: EnableBlock): Void

enableSignal :Line

dstPort == "enable"

enableBlock

qualifiedName != enableSignal.qualifiedName

Check and repair action



2. The ToolNet Project

(COTS Tool Integration Framework)

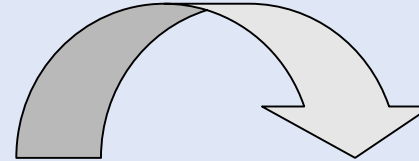
(DaimlerChrysler, EXTESSY AG, TU Darmstadt, ...)



The Running Example – Part 2



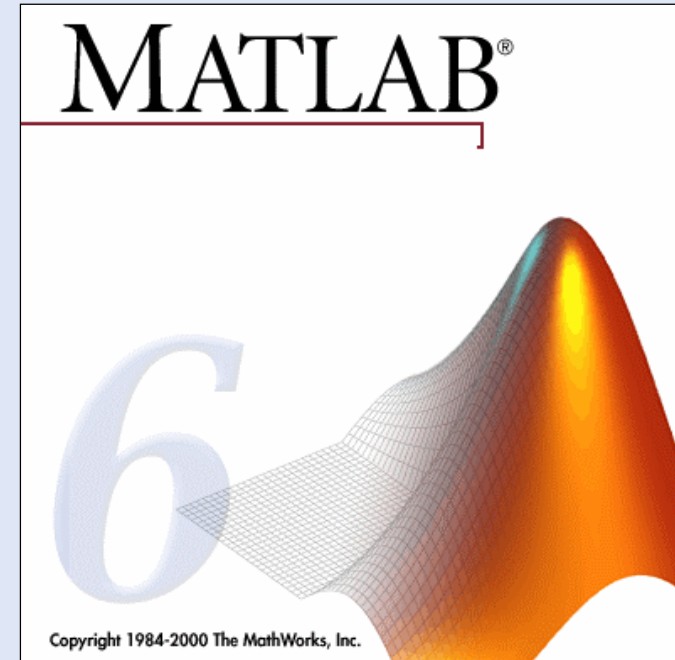
synchronize



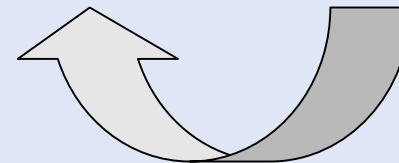
Simulink &
Stateflow



DOORS
Requirements Engineering
Tool



synchronize





DOORS



Traceability Links



MATLAB Simulink

Formal module 'ECMDA/Automobile Comfort System' current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User ToolNet Help

Standard view All levels

- Automobile Comfort System
 - 1 Automobile Comfort System: T
 - 1.1 Ice Warner: The Ice Wa
 - 1.2 Seat Belt Warner: The S
 - 1.3 Headlight Recommendation
 - 1.4 Windscreen Wiper Assist

1 Automobile Comfort System
The Automobile Comfort System is a collection of comfort features that assists a driver in a number of different ways.

1.1 Ice Warner
The Ice Warner informs the driver if the temperature at the outside drops below 3°C.

1.2 Seat Belt Warner
The Seat Belt Warner informs the driver if any passenger has not fasten his or her seat belt.

1.3 Headlight Recommendation
The Headlight Recommendation informs the driver if he or she should switch on the headlights due to lack of brightness at the outside.

1.4 Windscreen Wiper Assistant
If activated the Windscreen Wiper Assistant automatically starts and stops the windscreen wiper and adjusts the speed according to the quantity of rain.

Username: akoenigs Exclusive edit mode

keep consistent

comfortsystem

File Edit View Simulation Format Tools Help

Normal

Ready 100% ode45

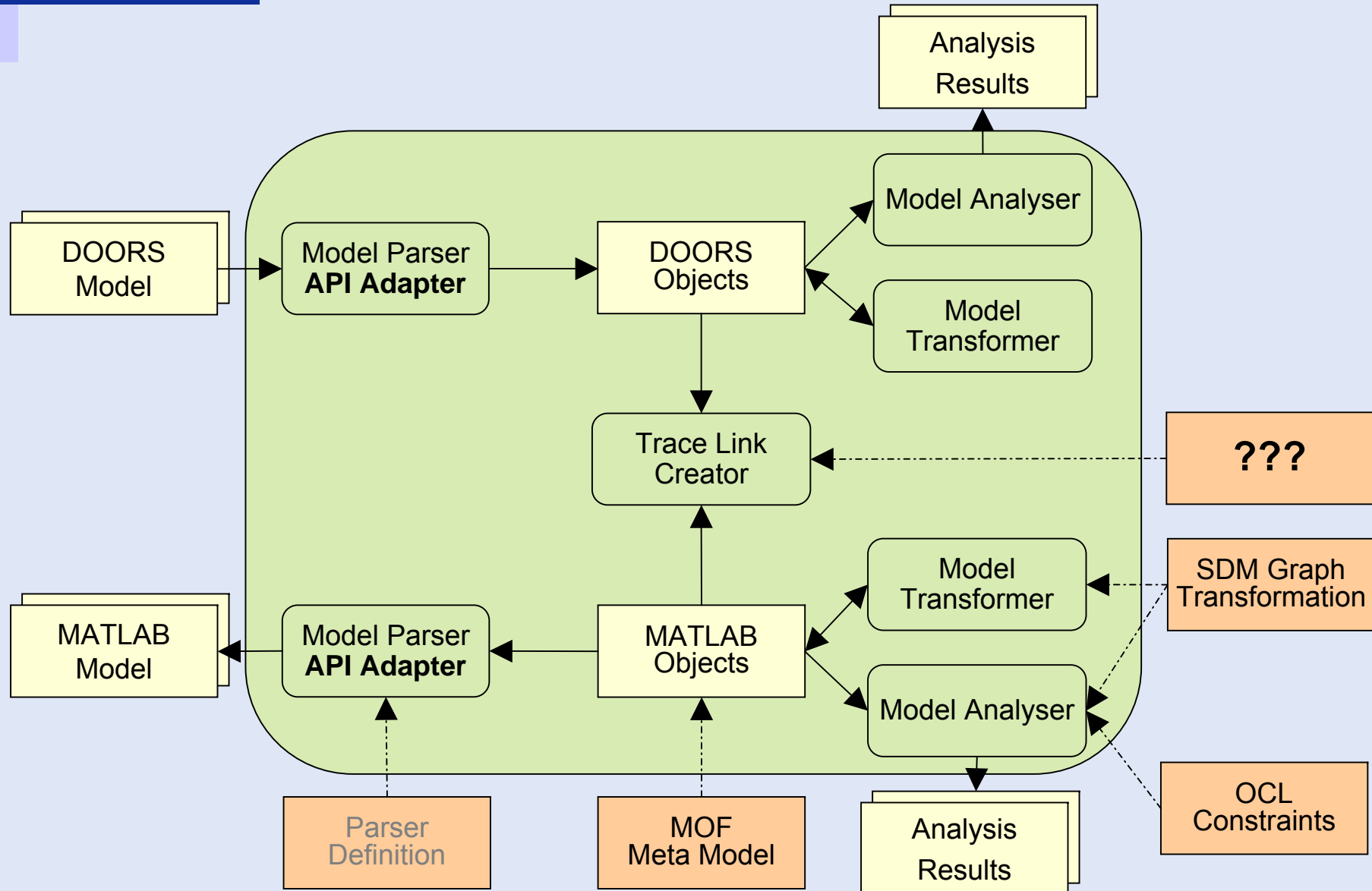


Consistency Checking Scenario



19

www.es.tu-darmstadt.de



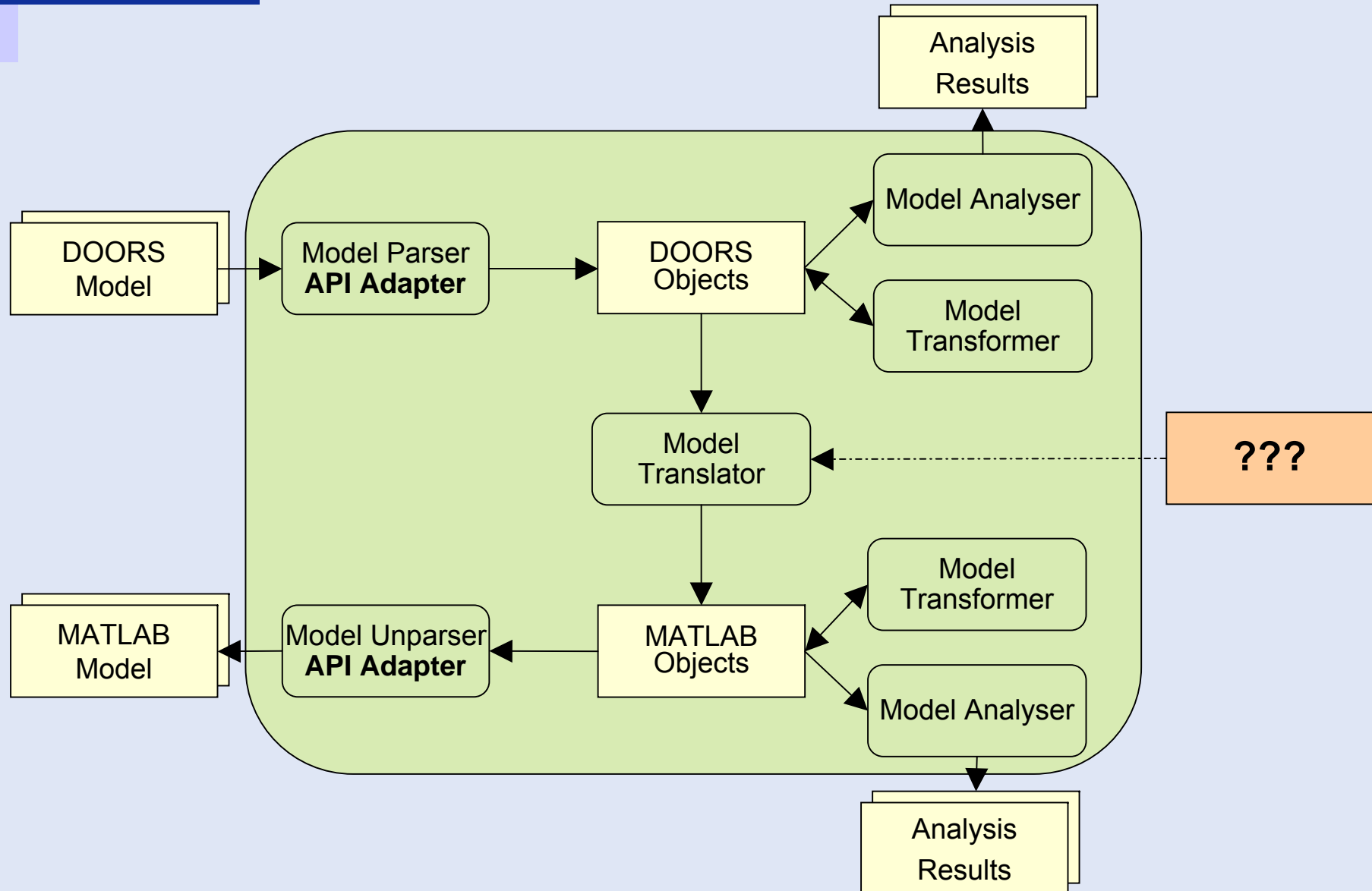


Forward Transformation Scenario



20

www.es.tu-darmstadt.de



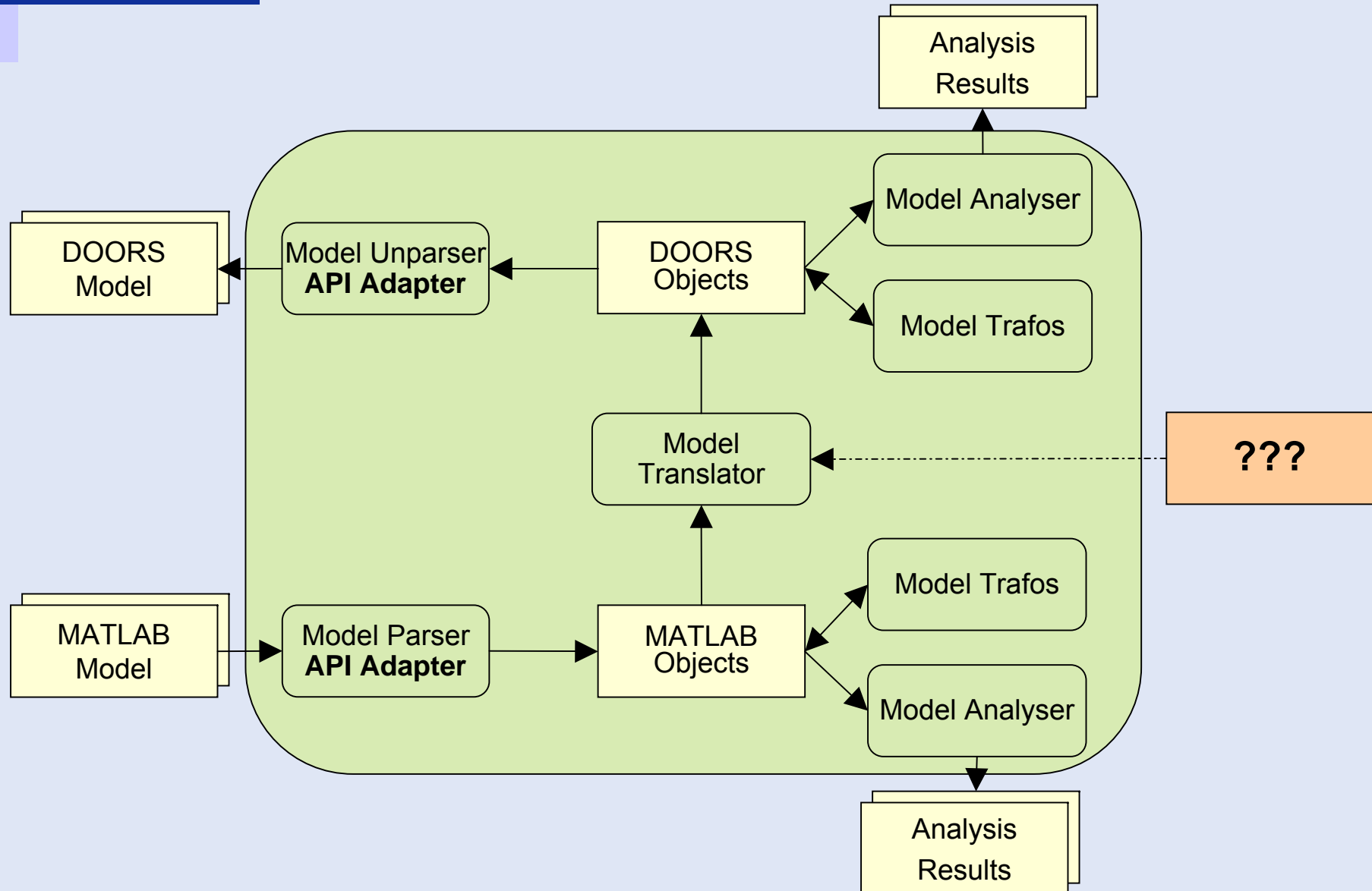


Backward Transformation Scenario



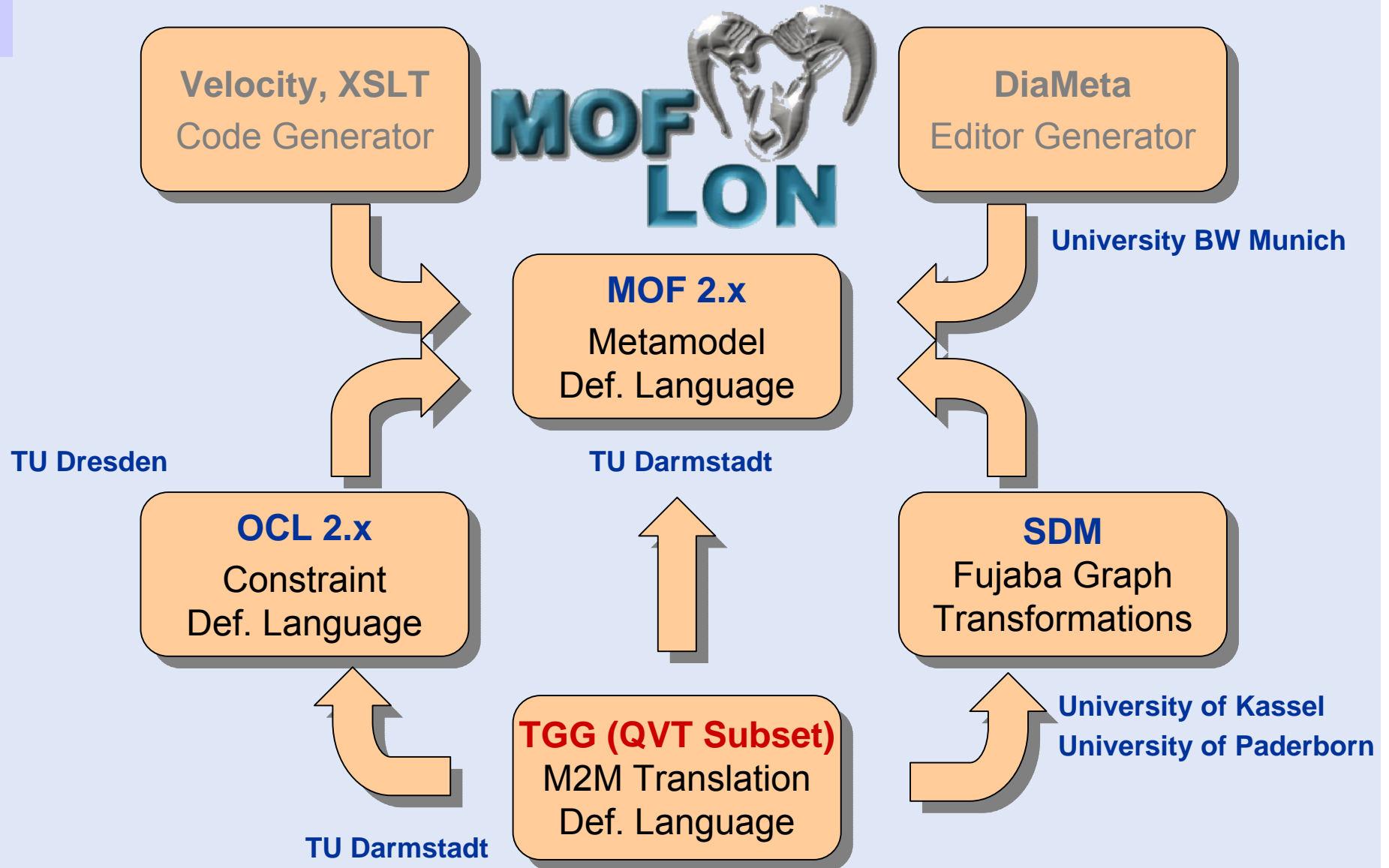
21

www.es.tu-darmstadt.de





- Needed model/document/artefact integration rule sets:
 - create traceability links only
 - check traceability link consistency, completeness, ...
 - remove inconsistent traceability links
 - forward transformation
 - backward transformation
 - forward/backward attribute propagation
 - forward/backward structure modification propagation
 - ...
- Generate all transformation rule sets from single declarative bidirectional model integration rules





Fujaba Tool Suite - MOFLON [DoorsMatlabIntegration]

File MOF 2.0 Diagram Tools Options Window Help

Projects

- DoorsMatlabIntegration
 - Matlab
 - Doors
 - PrimitiveTypes
 - Requirement
 - Module
 - PrimitiveTypes
 - Integration

**+ Matlab
Meta
Model**

DOORS Meta Model

```

classDiagram
    class Requirement {
        heading : String
        module : Module
        nested : Requirement
        nesting : Requirement
        text : String
    }
    class Module {
        name : String
        requirement : Requirement
    }
    Requirement "0..1" --> "1" Requirement : nesting
    Requirement "x" --> "1" Requirement : nested
    Requirement "1" --> "0..1" Module : requirement
    Module "1" --> "0..1" Requirement : module
  
```

Integration

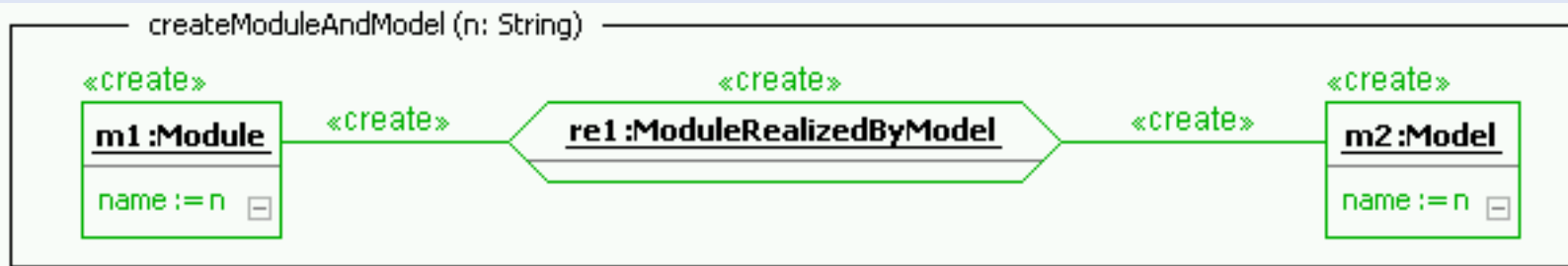
«import»
Module
from Doors

«import»
Requirement
from Doors

Current Project = [null] 10 MByte of 18 MByte allocated

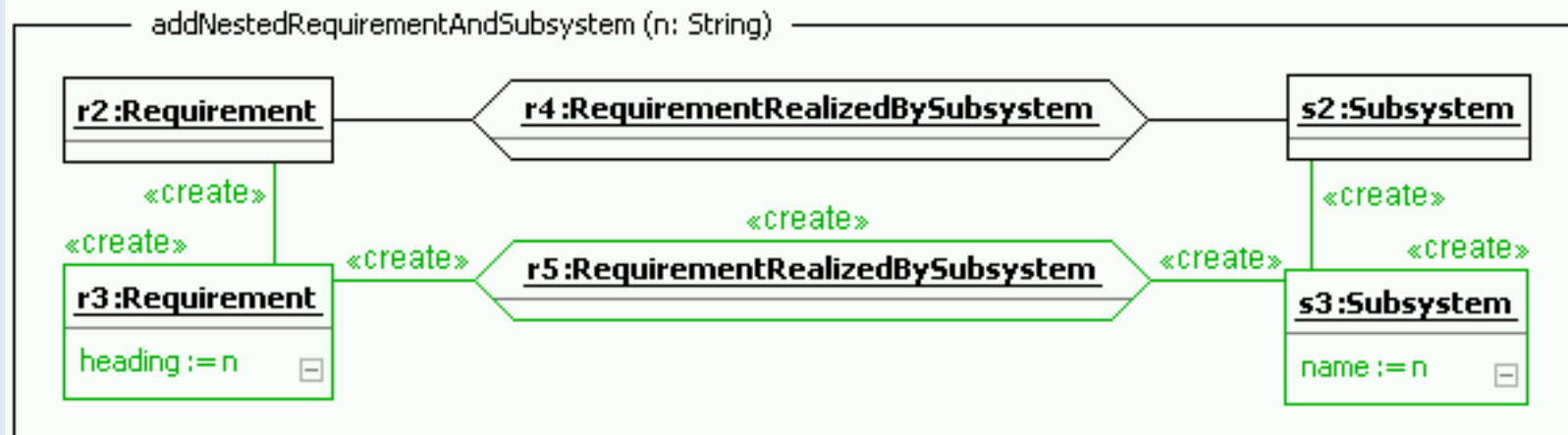


25



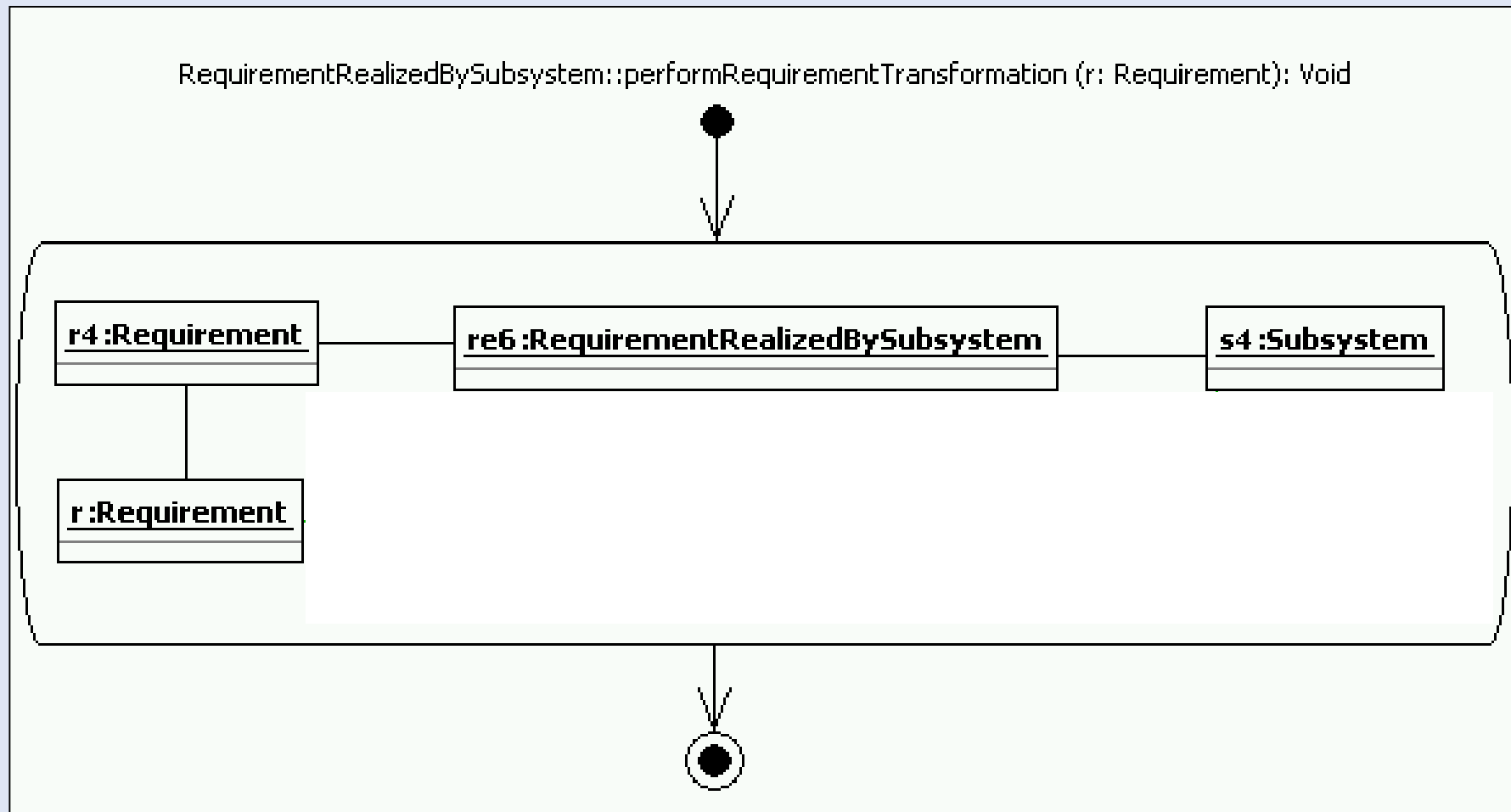
Context

Context





1st step: identify existing context





3. Conclusions



Basic Properties of Traceability Link Types:

- Permitted source and target classes in meta-models
 - Simple consistency checks
- Multiplicities for both link ends
 - Simple completeness checks
- Integrity constraints for link types (OCL, Java, ...)
 - Static correctness checks (static semantics)
- Structural integrity rules, ...
 - Automatic creation and checking of links

Additional properties of traceability link type definitions:

- Inheritance hierarchies of link types
- Composition of rules for link types (subrules)
- Modularization of integration rules (re-use, refinement)



Integrated and Distributed Versioning of Development Artefacts and Links

Representation of Web of Artefacts and Links

Specification of Integration Rules for Automatic Link Creatin and Checking

...