Fault Tree Generation from EMF Models

Christoph Lauer¹, Reinhard German¹ and Jens Pollmer²

¹Department of Computer Science 7 – Computer Networks and Communication Systems,
Friedrich-Alexander University, Erlangen-Nuremberg, Germany

²Department of Safety Electronics,
Audi AG, Ingolstadt, Germany
{christoph.lauer, german}@informatik.uni-erlangen.de, jens.pollmer@audi.de
Outline

- Introduction
- Integrated Safety Architectures
- System Models
- Fault Tree Generation
- Conclusions and Future Work
3
7/6/2009
Lauer, Christoph, Bounding the Number of Relevant Objects in Automotive Environment Perception Systems
(PreVent, 2007)

(PreVent, 2007)
Integrated Safety Architectures in the Automotive Domain (1)
Integrated Safety Architectures in the Automotive Domain (1)
Integrated Safety Architectures in the Automotive Domain (3)

- Integrated ECU
  - Energy Reserve
  - Dual Core Controller
  - Watchdog
  - PSi5 Transceiver

- Flexray
- Safety ASIC
  - Squibs
  - SPI Bus

- x/y
- Φ
- -x
Integrated Safety Architectures in the Automotive Domain (4)

Integrated ECU

Controller 2
- PSI5 Transceiver
- x/y

Watchdog

Controller 1
- Energy Reserve
- φ
- -x

Safety ASIC
- Squibs

SPI Bus

Flexray
Integrated Safety Architectures in the Automotive Domain (5)
Requirements for Modeling & Generation

- Separate modeling of system architecture and functional behavior
- Flexible allocation of functional tasks to system nodes
- Automatic generation of fault trees for further analysis using state-of-the-art tools
- No extensive design space exploration
EMF Model Support (2)
Transformation Rules

1) Start at top-level event
2) Evaluate top-level event
   a. Get faults from allocated system entity
   b. Add faults of entity directly (via OR gate)
3) Evaluate all incoming edges
4) Evaluate node
   a. Get faults from allocated system entity
   b. Traverse graph to top-level event
   c. Add fault directly (via OR gate) if fault propagates, or add guardian (via AND gate) if fault is not propagated
5) Terminate if no incoming edges exist, else go to 3)
Transformation Example (1)
Transformation Example (2)
Transformation Example (3)

S_Pers_1  S_Trans_1
S_Pers_i  S_Trans_i
T_Impl
T_Plaus: (S_Trans_1,2,3,i)
S_Pers_2  S_Trans_2
S_Pers_3  S_Trans_3
Transformation Example (4)
Conclusions

- Modeling of system and behavior using the EMF
- Model transformation from separated system model + behavior model to fault trees
- Just a transformation, the algorithm does not "create knowledge"
- Level-of-detail of the fault trees depends on the level-of-detail of the input models
- Method supports analysis of different architecture options at early design stages
Future Work

- Leave the Ecore path for the sake of UML
  - Modeling of the system and the behavior view using MARTE(+) Depandability profile from Bernardi et al. (2008)) or EAST-ADL2
  - Papyrus plug-in for easy modeling without having to cope with UML

- Implementation (!) of interfaces to FaultTree+ (ISOGraph)
Thanks for your attention!