Theoretical Computer Science – Formal Specification

Research Field	
Overall Research / Technology Area	**Specific Research / Technology Area**
Software Technology	Formal Specification, Visual Languages, Graph Transformation, Petri Nets

Organization

Lead Organization	Partners
TU Berlin, Dept. EE&CS	Academic Partners and Projects, e.g. „Formal Modelling of Mobile Ad-Hoc Networks“, „Appl. of Graph Transformation to Visual Modelling Languages“
Contact @ Lead Organization	**Contact @ Partners**
Prof. Hartmut Ehrig tfs.cs.tu-berlin.de	see tfs.cs.tu-berlin.de/projects

Potential (Future) Applications in Aerospace / Defense

- Visual modelling and formal analysis of processes
- Model transformation from domain-specific models into semantic domains
- Analysis of conflict and dependencies in models
- Restructuring (Optimizing) of model structures
- Model validation by simulation and animation
- Tool *AGG*: graph transformation engine and analysis tool
- Tool *Tiger*: specification-based generation of visual editors in Eclipse

Technology / Research Short Description

Formal Methods and Tools in Software Specification and Development

- **Algebraic Specification Techniques**
  - Integration of modelling techniques and specification languages
  - Component based software architecture and evolution
  - Semantics of modelling and specification languages
- **Graph Transformation and Visual Modelling**
  - Graph transformation systems and tools
  - Formal syntax and semantics of visual languages
  - Model Driven Development and Model Quality
  - Model Transformation by Graph Transformation
- **Petri Net Technology**
  - Abstract Petri Nets: Unification of various Petri net classes by parameterization
  - Modelling and analysis of processes with Higher-Order Nets

Business Value for Boeing

Degree of Strategic Value	1	Degree of Maturity	3

Value Proposition for Boeing

- premature research (still fundamental research)
- use of formal modelling techniques to avoid and detect errors as early as possible in the modelling phase

Potential Collaboration

- possible cooperation via case study from aerospace application to improve applicability of formal techniques

- increasing decentralization of software in technical systems
  - requires correct, safe, and flexibly adjustable software,
  - has to take into account different specification aspects from technical descriptions of physical systems to analysis, design and deployment models of software systems
- Aim: development of integration techniques to
  - compare different specifications,
  - establish semantic correspondences
  - check their consistency
  - define informal or formal semantics of a collection of specifications
- Problems:
  - heterogeneity of languages
  - different, but possibly overlapping views on the system
  - different levels of granularity and abstraction
- Solutions by
  - Meta-Modelling
  - Reference Models based on Algebraic Techniques and Category Theory

Component-based Software Architecture and Evolution

(BMBF Project “Continuous Software Engineering” ContEng)

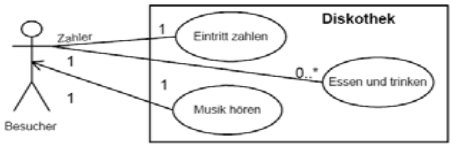
- Aim:

  Development of a component framework for different kinds of software modelling techniques to support software architecture and evolution in the design phase

- Concepts:
  - Algebraic Module Concept
    - H. Ehrig, B. Mahr: *Fundamentals of Algebraic Specification 2: Module Specifications and Constraints*, EATCS Monographs, Springer 1990.
  - Categorical Framework of Institutions and Specification Frames
  - Extensions to graph transformation systems and integrated data type and process modelling techniques
- Applications by Instantiations with
  - partial algebras
  - behavioral specifications including various kinds of Petri nets
  - UML diagram types

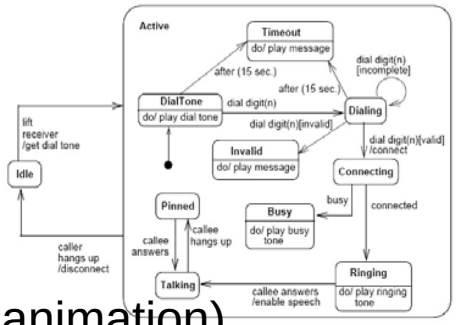
Model Driven Software Development
(DFG Project Application of Graph Transformation to Visual Modelling Languages)

Preparation



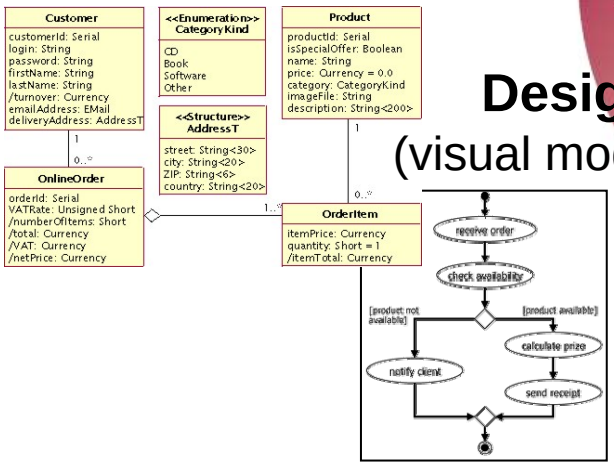
Analysis
(requirements model)

Model based Validation
(simulation und animation)



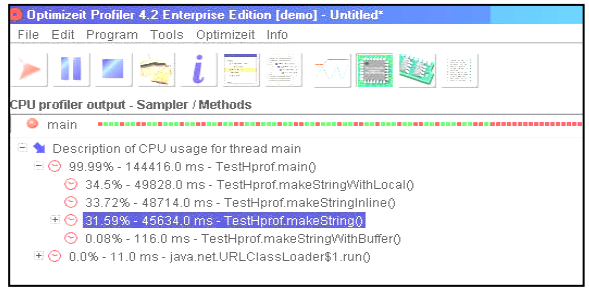
Postprocessing

Design
(visual model)



MDD

Implementation
(partially by code generation)

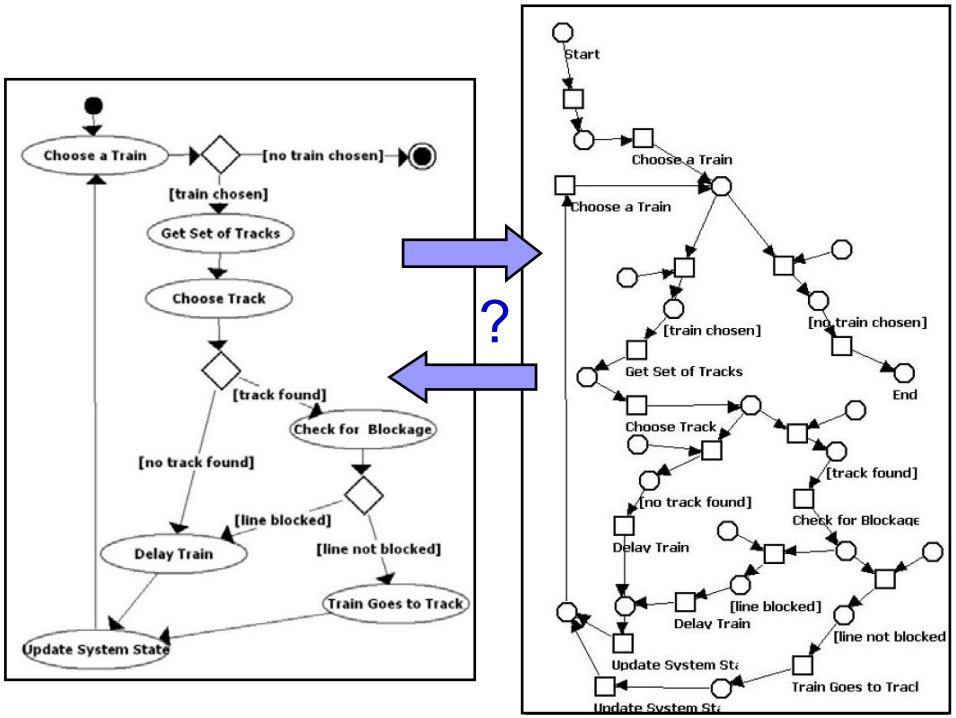


Model Driven Development: Defining Visual Languages
(DFG Project Application of Graph Transformation to Visual Modelling Languages)

- Criteria for Model Quality
  - syntactical and semantical correctness
- Aim: Formal Modelling Technique for Syntax and Semantics of Visual Modelling Languages
  - Syntactical correctness:
    - Does a model belong to a modelling language?
    - Does the model satisfy modelling conventions?
    - Can the model structure be optimized?
  - Semantical correctness:
    - Can the model behaviour be validated?
    - Are semantical model requirements satisfied?
    - Model transformation: is the translated model behaviour-equivalent to the original model?
- Concepts:
  - Definition of visual languages by meta-modelling and syntax graph grammars
  - Validation of model behaviour by graph transformation
  - Definition of model transformation by graph transformation
- Applications:
  - Petri nets, UML (activity diagrams, state diagrams, …), domain-specific languages
  - Model Transformation: activity diagrams to Petri nets

Model Driven Development: Model Transformations
(DFG Project Application of Graph Transformation to Visual Modelling Languages)

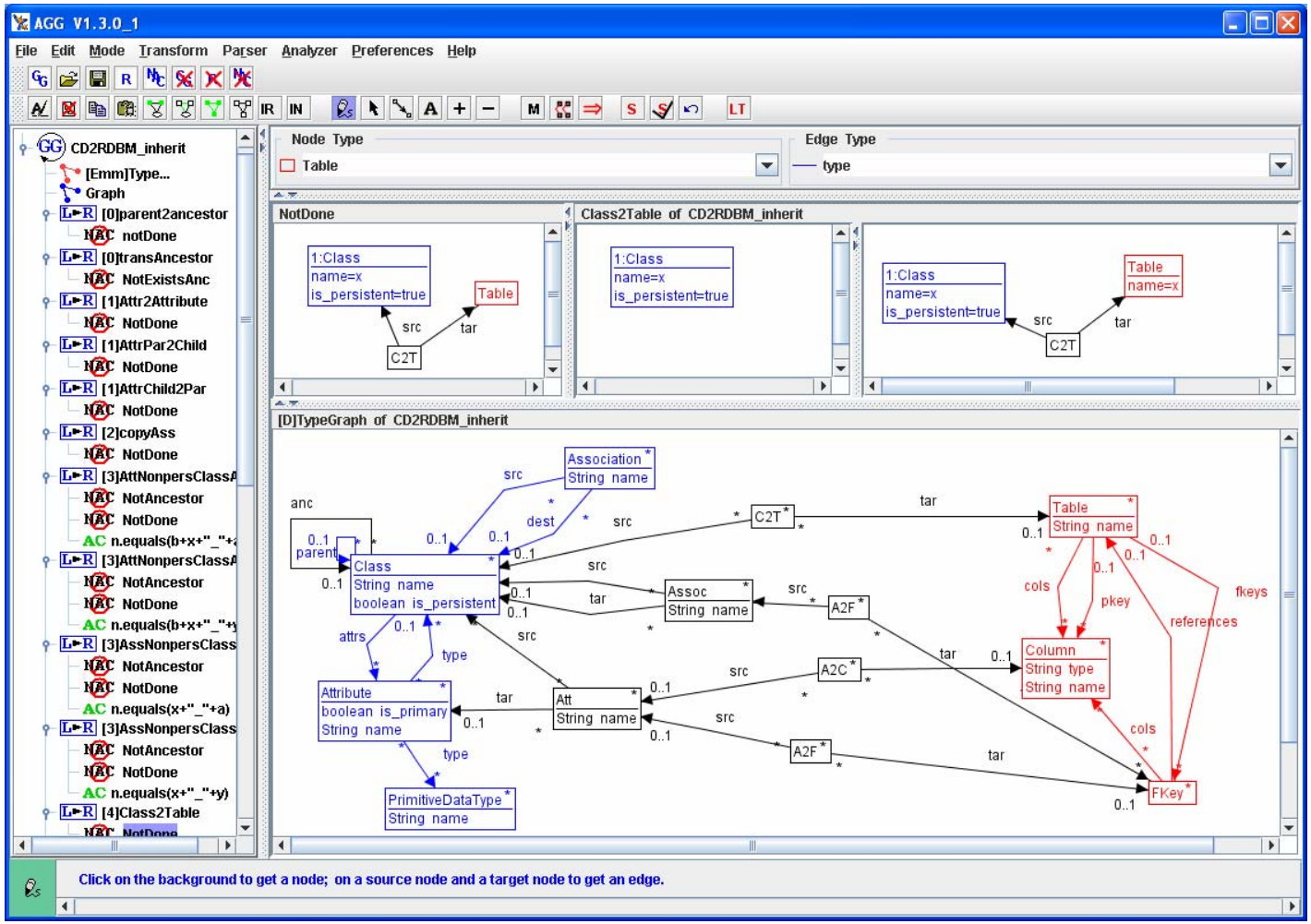
Example:
From activity diagrams to Petri nets



- Model Transformation Languages
  - OMG Standard QVT,
  - Graph Transformation
  - XML Stylesheets
- Aim: Formal Technique for Model Transformation
  - Functional Behaviour:
    - Termination and Confluence
  - Bidirectionality
  - Syntactical Correctness
    - Does the result conform to the target language?
  - Semantical Correctness
    - e.g. behaviour equivalence
- Concepts:
  - Model transformation by graph transformation
- Applications:
  - mapping to semantic domains
  - model refactoring
  - simulation-to-animation transformation

AGG: Attributed Graph Grammars

- Transformation Engine and Analysis Tools

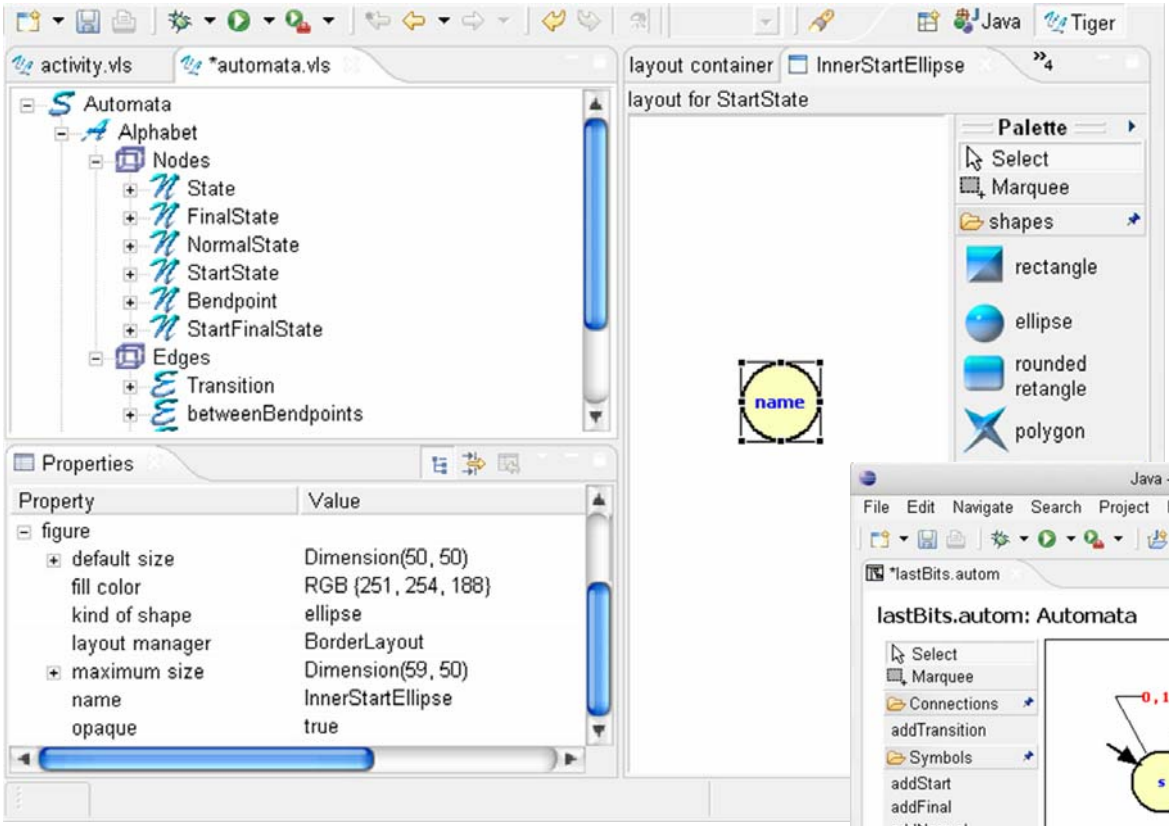


Example:

Model transformation from class diagrams to data base tables

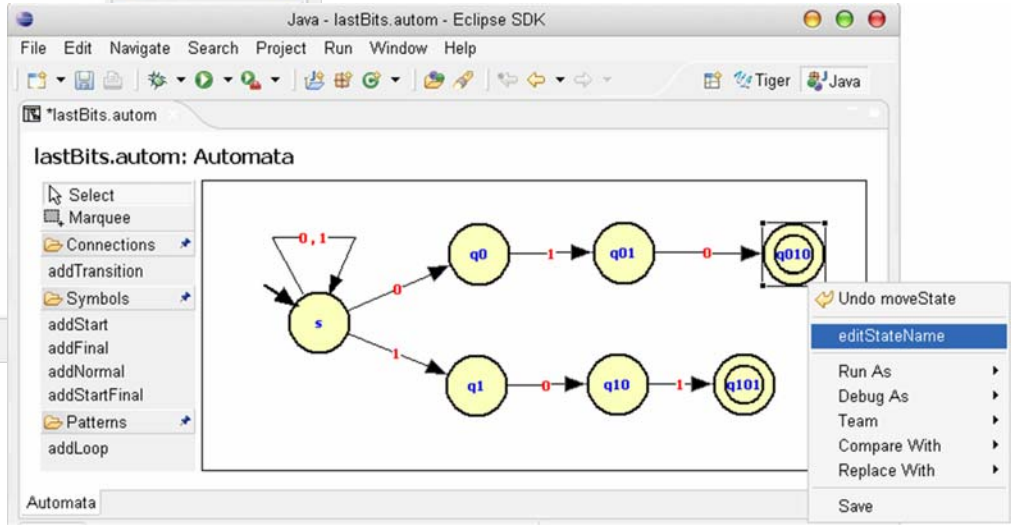
Tiger: Transformation-Induced Generation of Modelling Environments

- Rule Based Visual Language Designer and Editor Generator in Eclipse

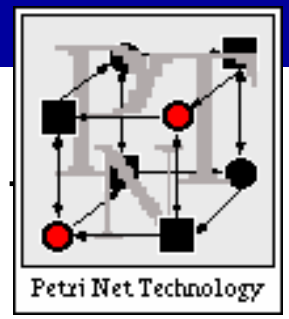


Example:

Generation of a visual editor for automata:



Abstract Petri Nets
(DFG Project Petri Net Technology PNT)



- Petri Nets: formal specification technique f
  process oriented systems
  - many variants of Petri nets for special purposes
- Aim: Unified Theory of Petri nets
  - taking into account the large variety of Petri net types and their different application domains
- Concepts:
  - Classification of Net Types and Techniques
  - Integration of Net Types and Data Types
  - Horizontal and Vertical Structuring
- Applications
  - Open Nets
  - Net-based Modelling of Flexible Workflows
  - Net-based Modelling of European Train Control Systems

Formal Modelling of Mobile Ad-Hoc Networks
(DFG Project ForMANET)

- Mobile Ad-Hoc Networks (MANETs)
  - permanently changing network topology
- Aim: Formal Modelling Technique for flexible Processes in MANETs
  - (re-)structuring of nets, net transformation
  - formal process modelling
  - analysis
  - tool support
- Concepts:
  - Integration of graph transformation, and higher-order Petri nets
- Applications:
  - Flexible Workflow Processes
  - Mobile Policies
  - Disaster Decision Support

Analysis Environment for Reliable Traffic Control Systems
(DFG Project Proposal with TU Braunschweig)