Embedded Control Systems Language for Distributed Processing (ECSL-DP)

A Language and Tools for Constructing Vehicle Control Applications

Sandeep Neema, Gabor Karsai, Attila Vizhanyo

Institute for Software Integrated Systems, Vanderbilt University

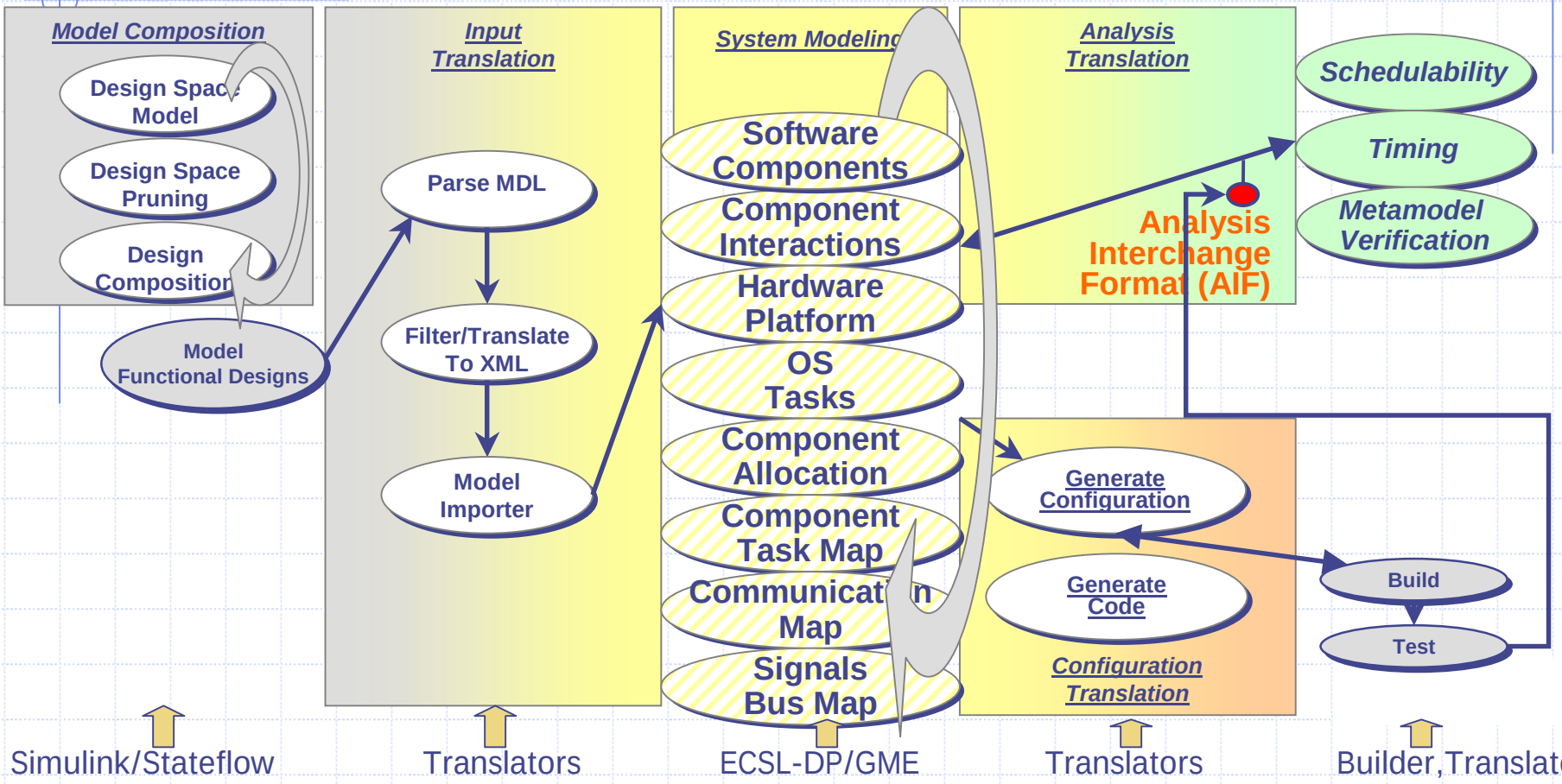
Outline

- Problem domain
- Model-based Development Tool-chain
- ECSL-DP Modeling Language
- Code Generation from ECSL-DP

Problem Domain

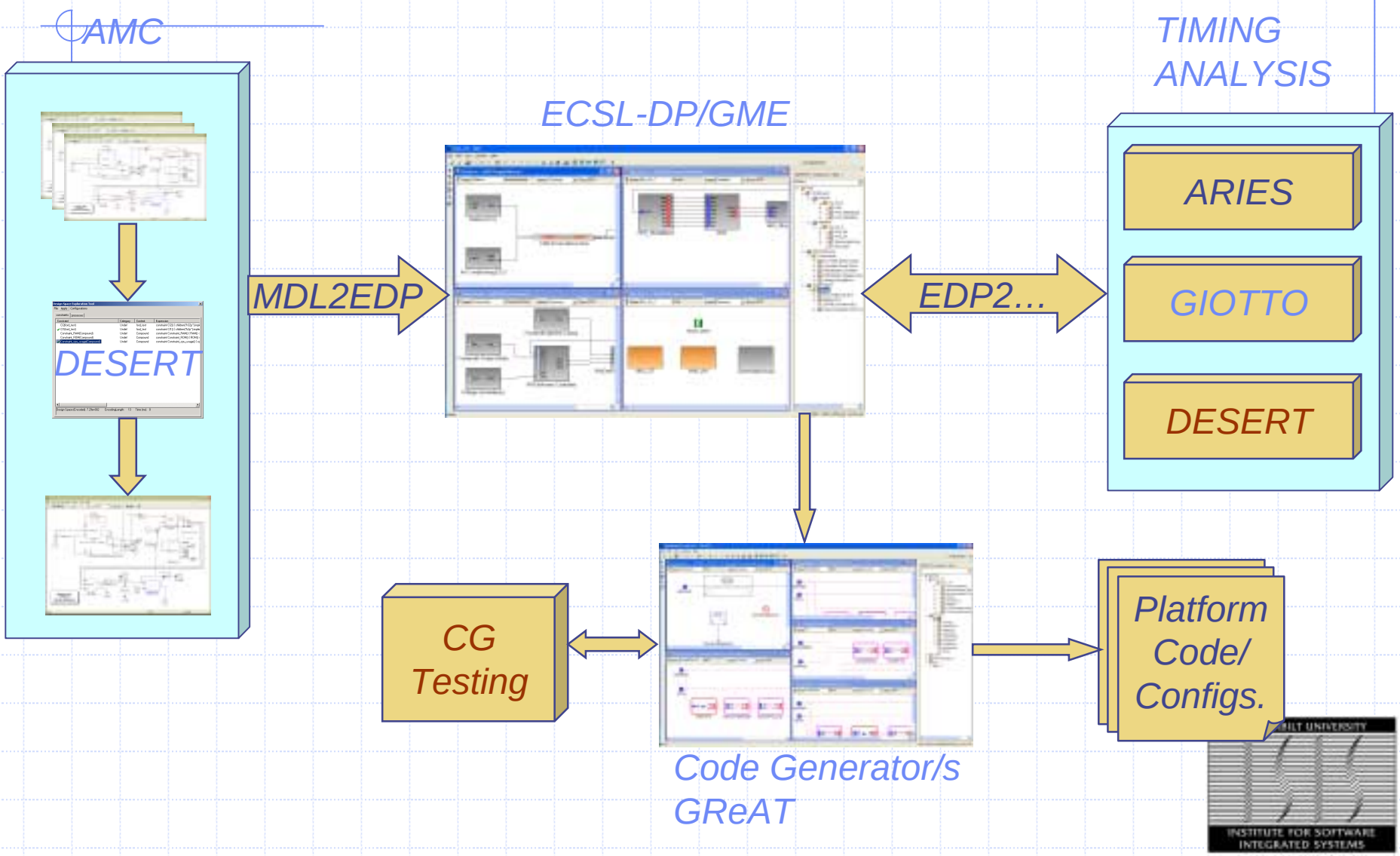
- Vehicle Control Applications
  - Soft real-time, distributed, medium scale (20-s) component-based software
- Time-Triggered Component Model
  - Periodic real-time tasks
- Tool-chain must support:
  - Component Design, System Design, System Composition
  - System Analysis, Platform Code Synthesis

VCP Tool-Chain:
Supported Activities



VCP Tool-Chain:

Tools and Interfaces



VCP Tool-Chain:
Participants

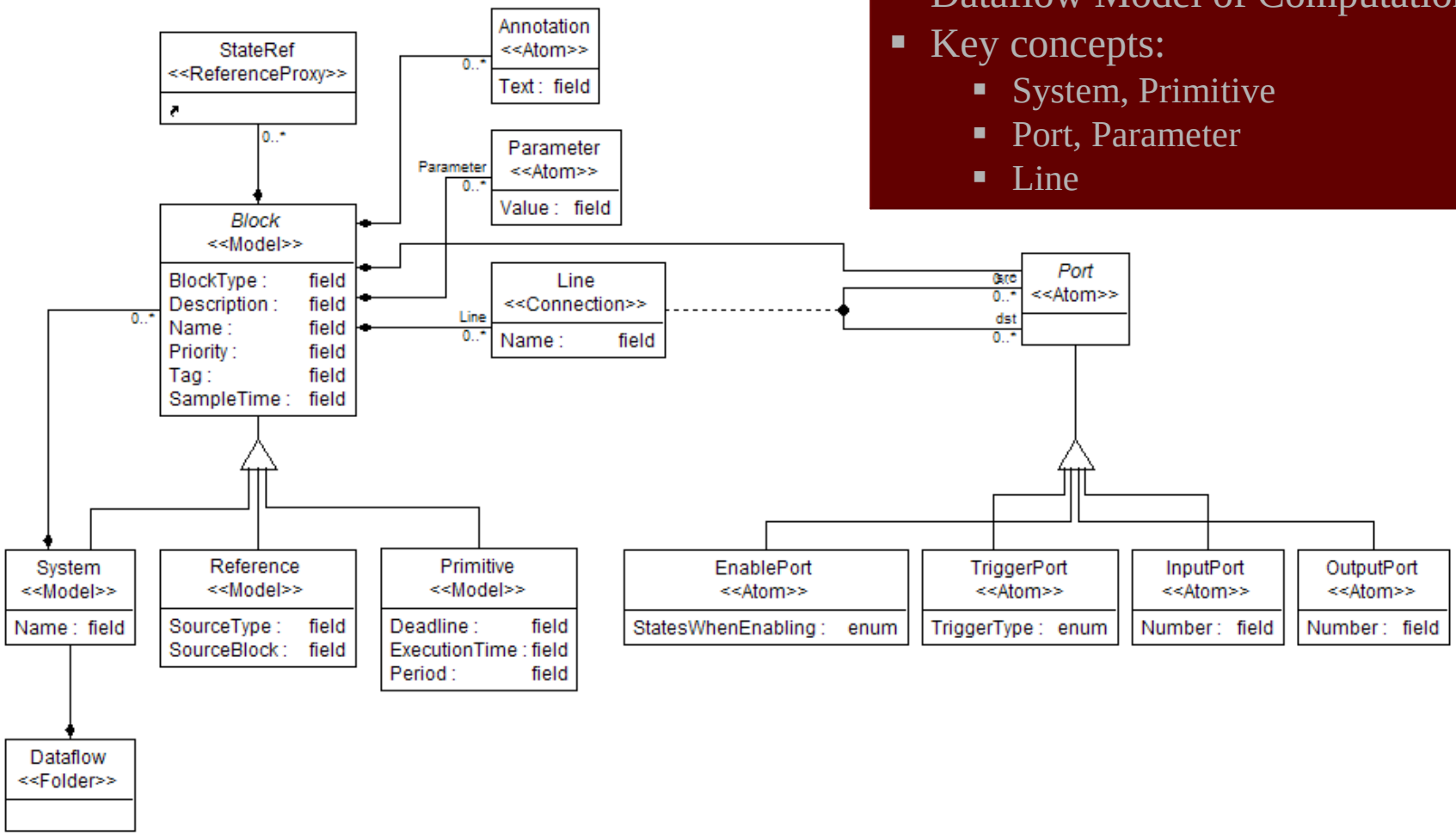
- Tools
  - Simulink/Stateflow
  - DESERT
  - GME(ECSL-DP)
  - AIRES, GIOTTO
- Interfaces
  - DSME
  - DESERT, DESERTBack
  - Matlab
  - AIF
- Translators
  - DSME2DESERT, DESERTBack2DSME
  - DSME2Matlab (DSME2SL;SL2M)
  - Matlab2ECSLDP, ECSLDP2Matlab
  - ECSLDP2AIF, AIF2ECSLDP

ECSL-DP Concepts

- Graphical design language for representing distributed embedded control systems
- Functional Modeling
  - Simulink/Stateflow
  - Dataflow diagram oriented of modeling signal flows
  - Hierarchical State Machine diagrams to model finite state behavior
- Component Modeling
  - Componentize the application
  - Simulink system references
- Platform Modeling – Topology
  - ECU-s, I/O channels, Buses
  - Firmware elements
- Platform Modeling – Deployment
  - Components → OS tasks
  - Component ports → sensors/actuators/bus-messages

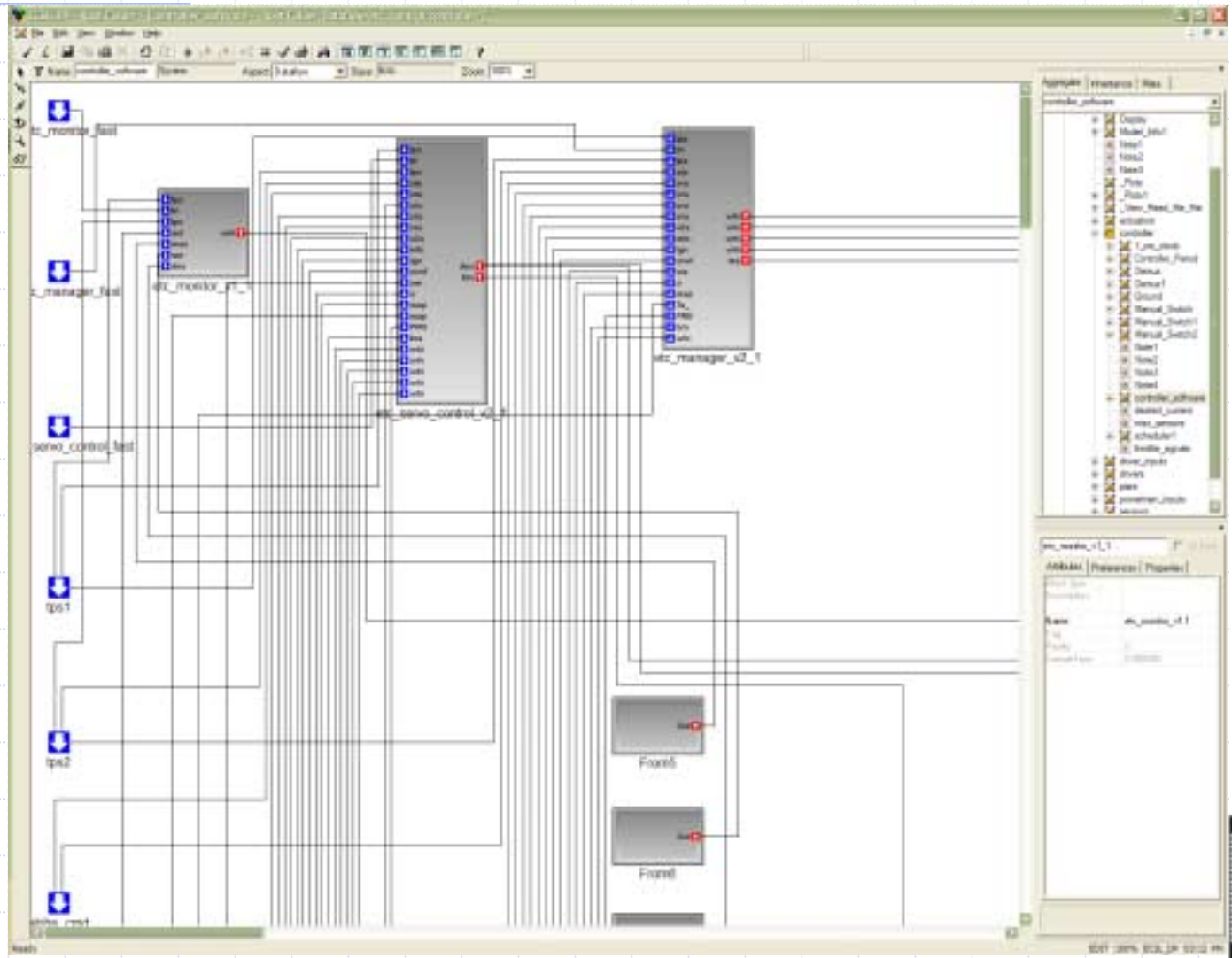
ECSL-DP:
Functional Modeling (Dataflow)

- Hierarchical Parameterized Dataflow Model of Computation
- Key concepts:
  - System, Primitive
  - Port, Parameter
  - Line



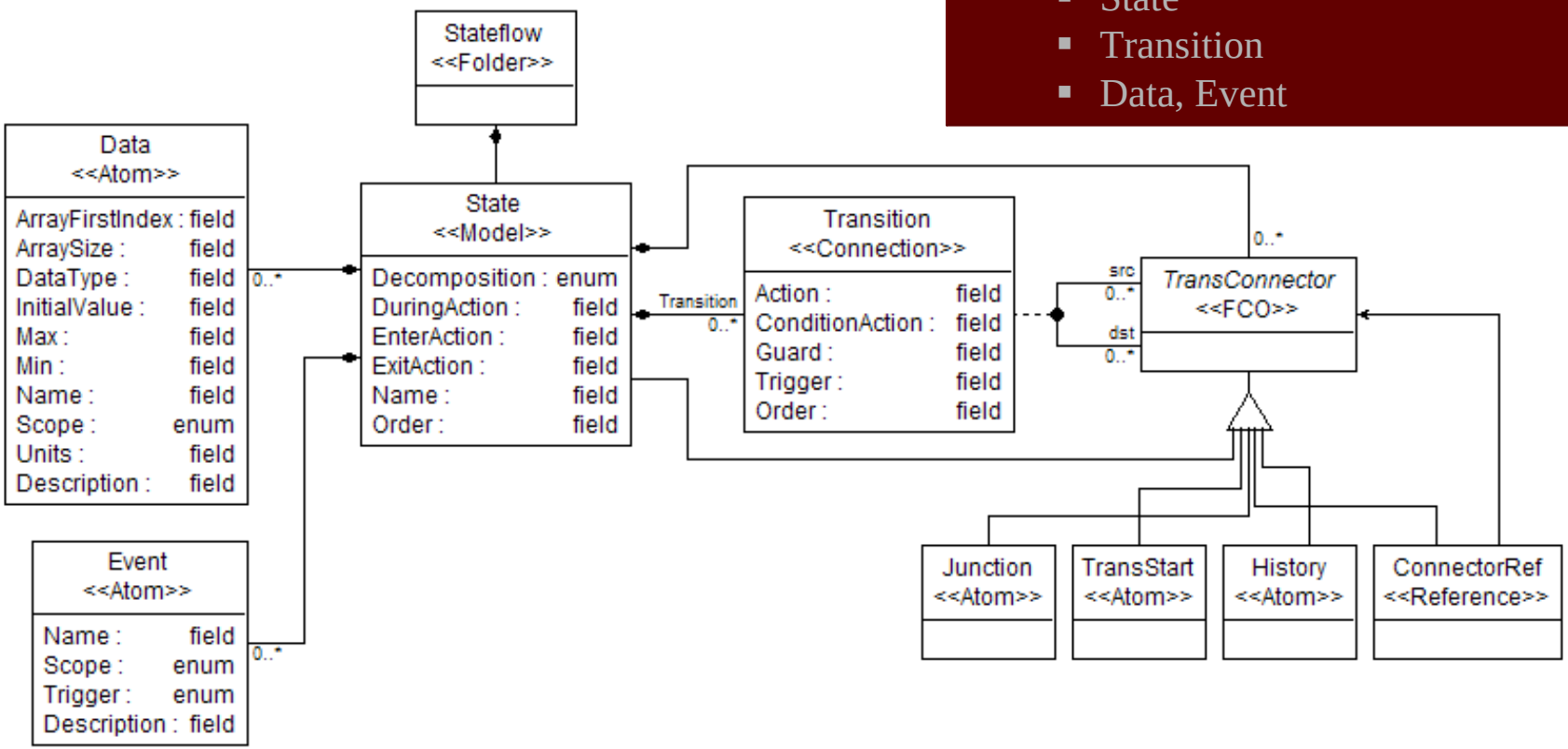
ECSL-DP:

Functional Modeling (Dataflow) Example

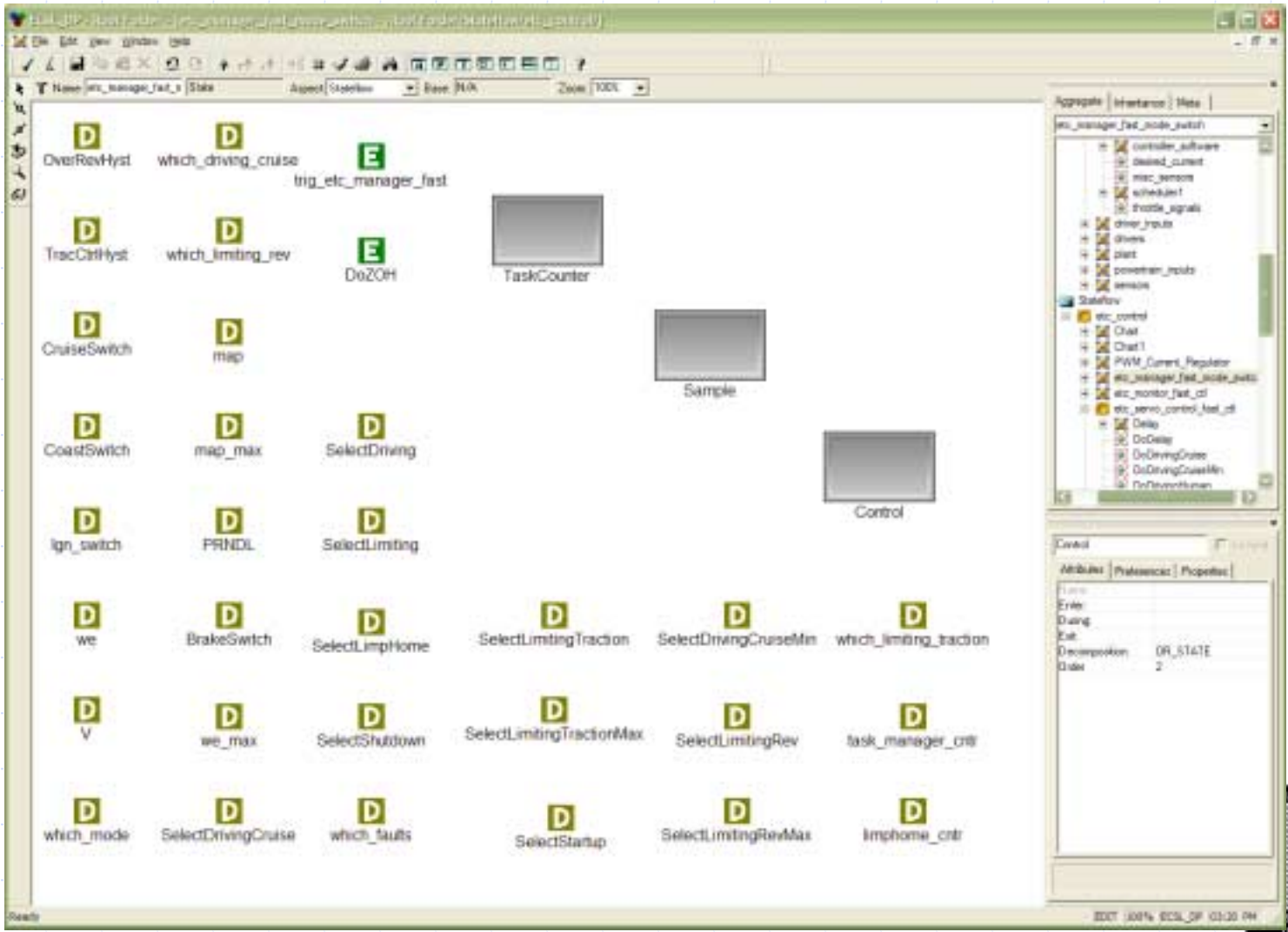


ECSL-DP:
Functional Modeling (Stateflow)

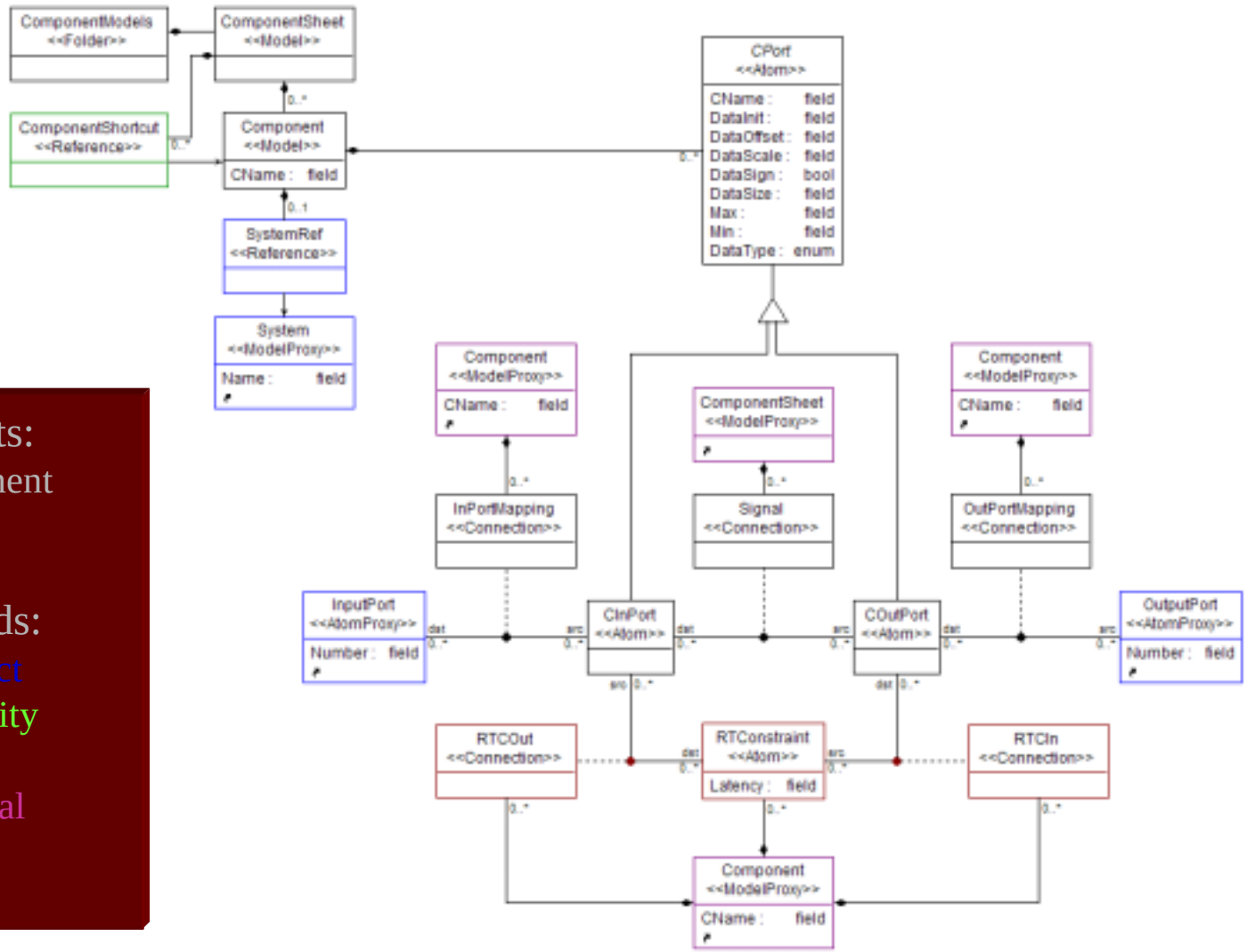
- Hierarchical Finite State Machines Model of Computation
- Key concepts:
  - State
  - Transition
  - Data, Event



ECSL-DP:
Functional Modeling (Stateflow) Example

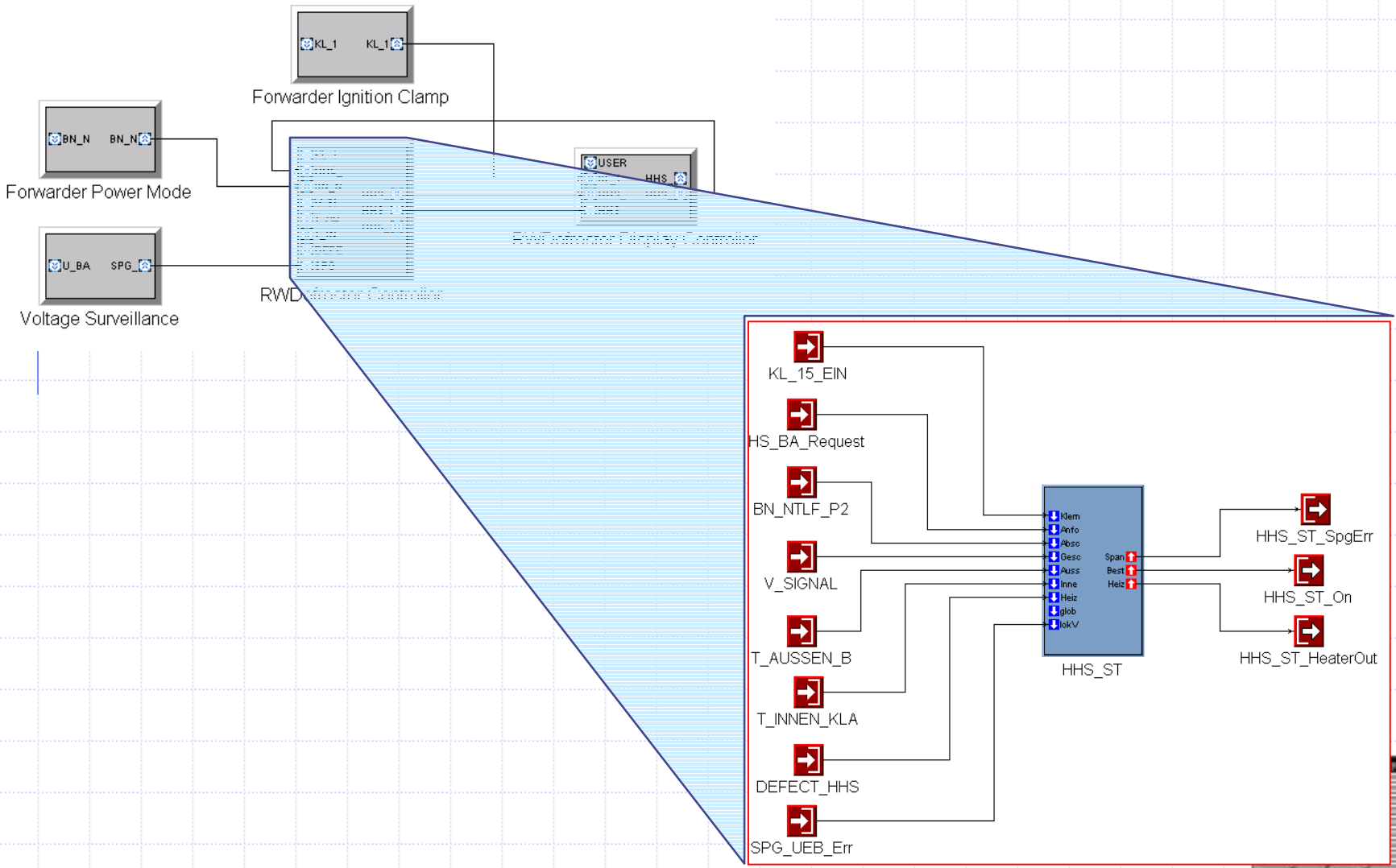


ECSL-DP:
Component Modeling

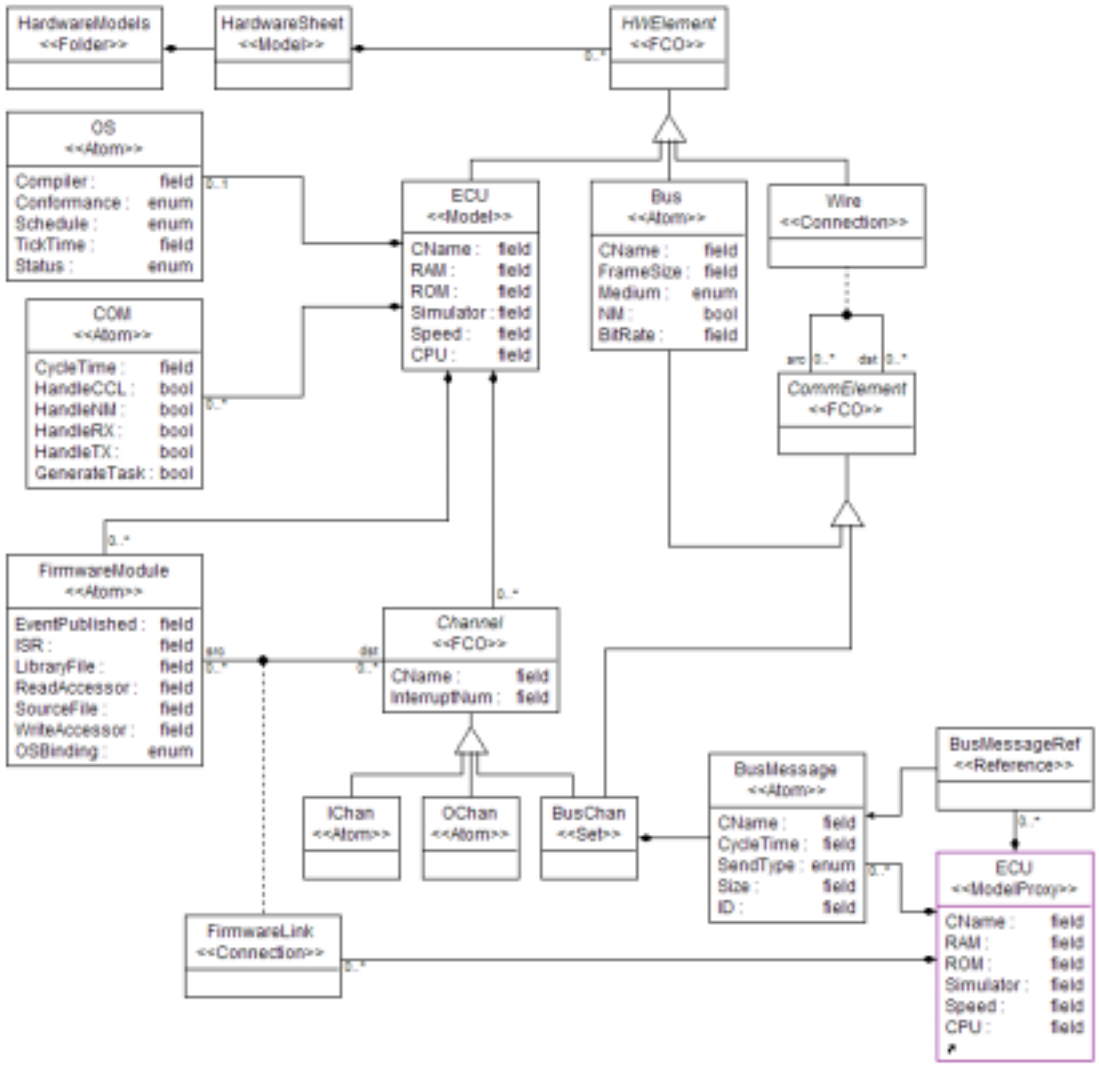


- Key concepts:
  - Component
  - CPort
  - Signal
- Color legends:
  - SL object
  - Scalability concept
  - Graphical layout
  - Timing

ECSL-DP:
Component Modeling Example



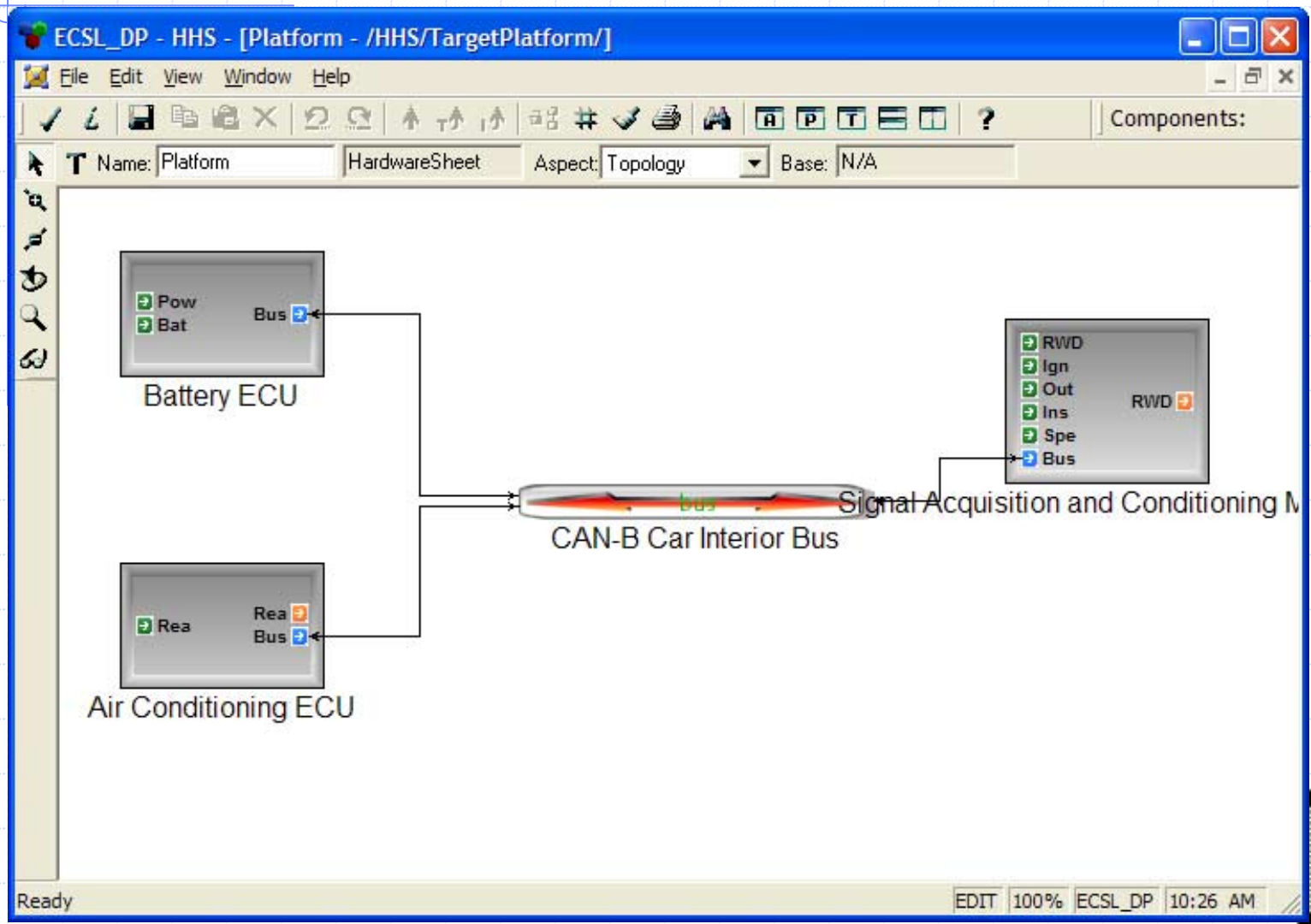
ECSL-DP:
Platform Modeling (Topology)



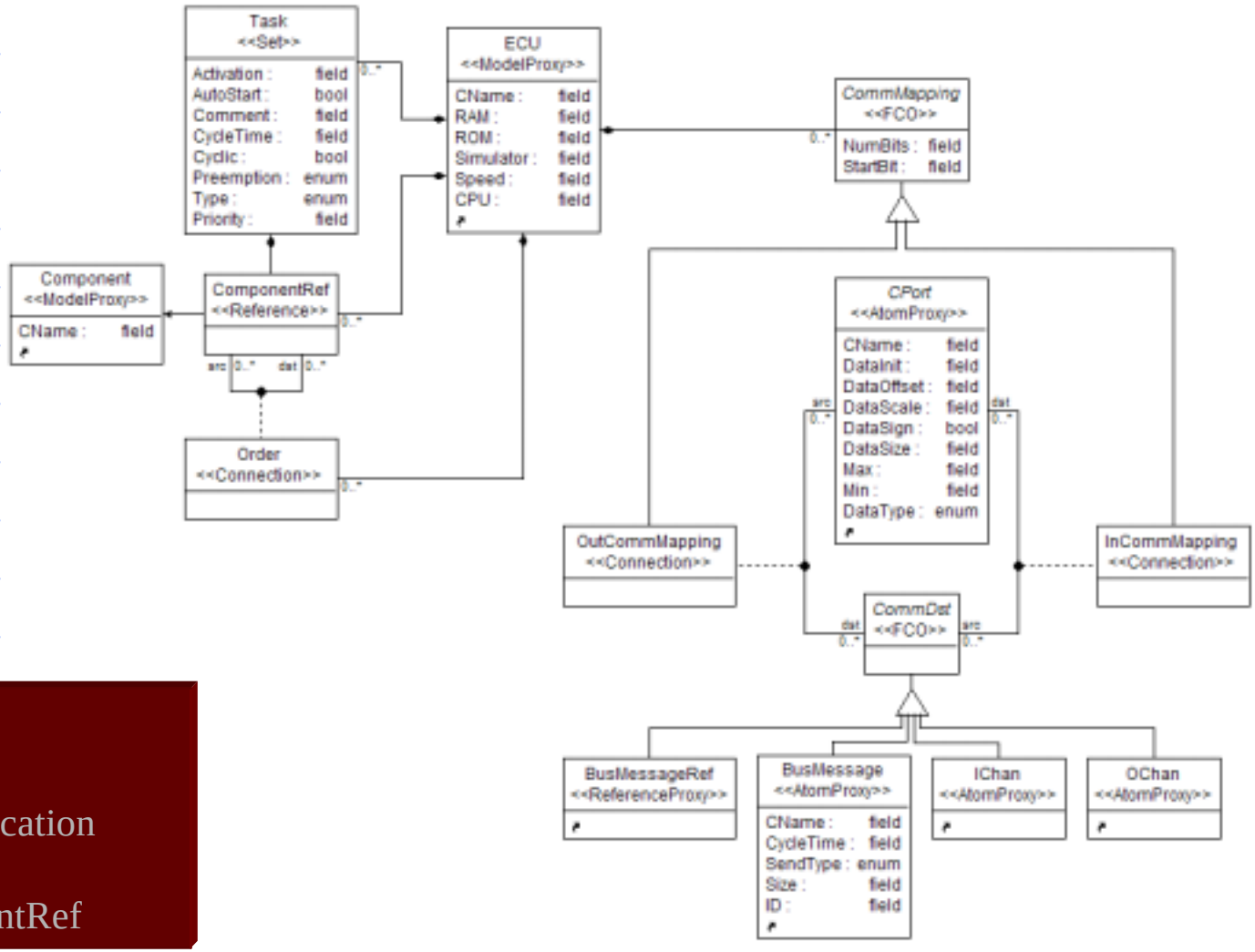
- Hardware concepts
  - ECU, Bus, Channel
- OS concepts
  - OS, COM, Firmware module

ECSL-DP:

Platform Modeling (Topology) Example

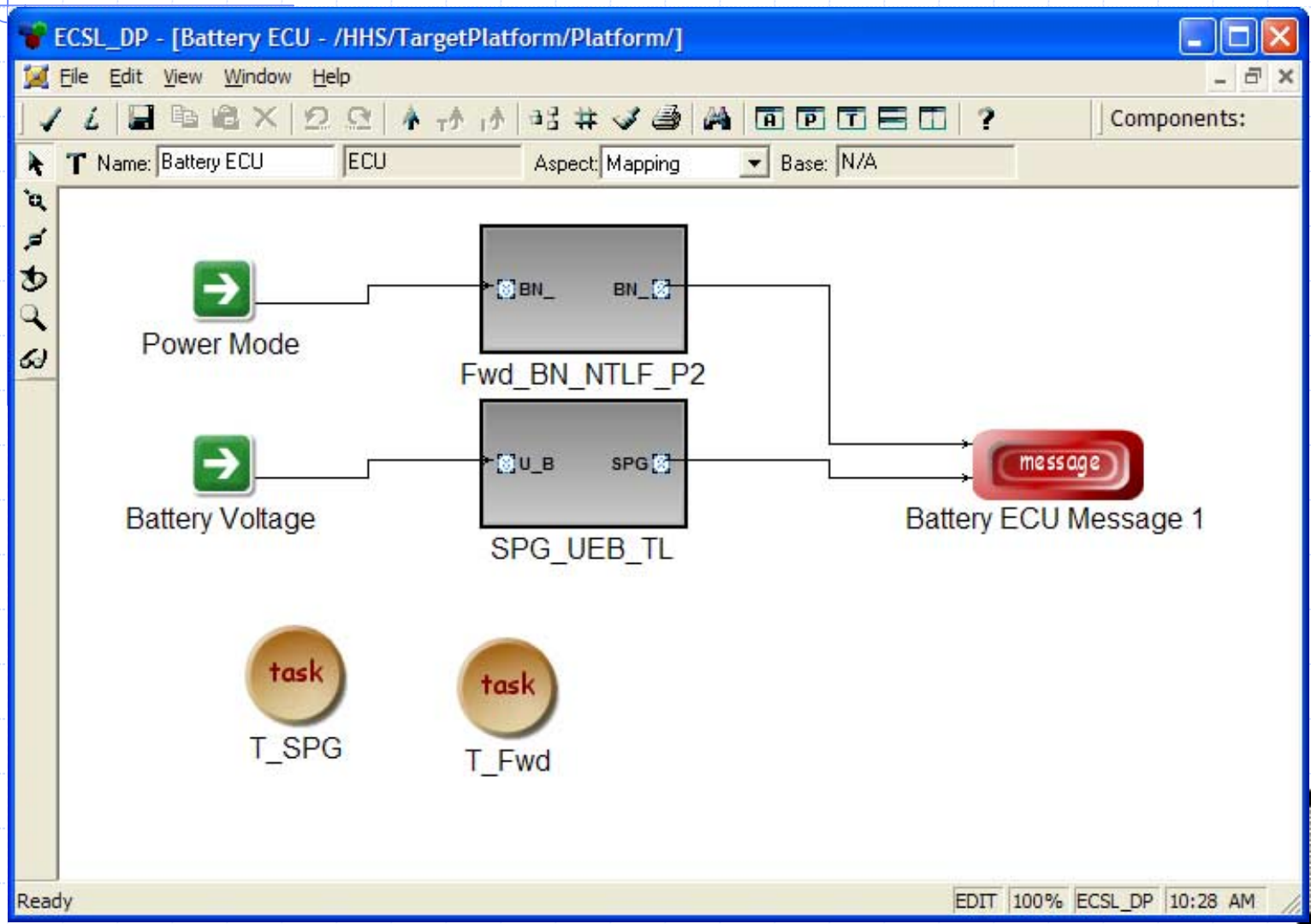


ECSL-DP:
Platform Modeling (Mapping)



- Key concepts
  - Task
  - Communication Mapping
  - ComponentRef

ECSL-DP:
Platform Modeling (Mapping) Example



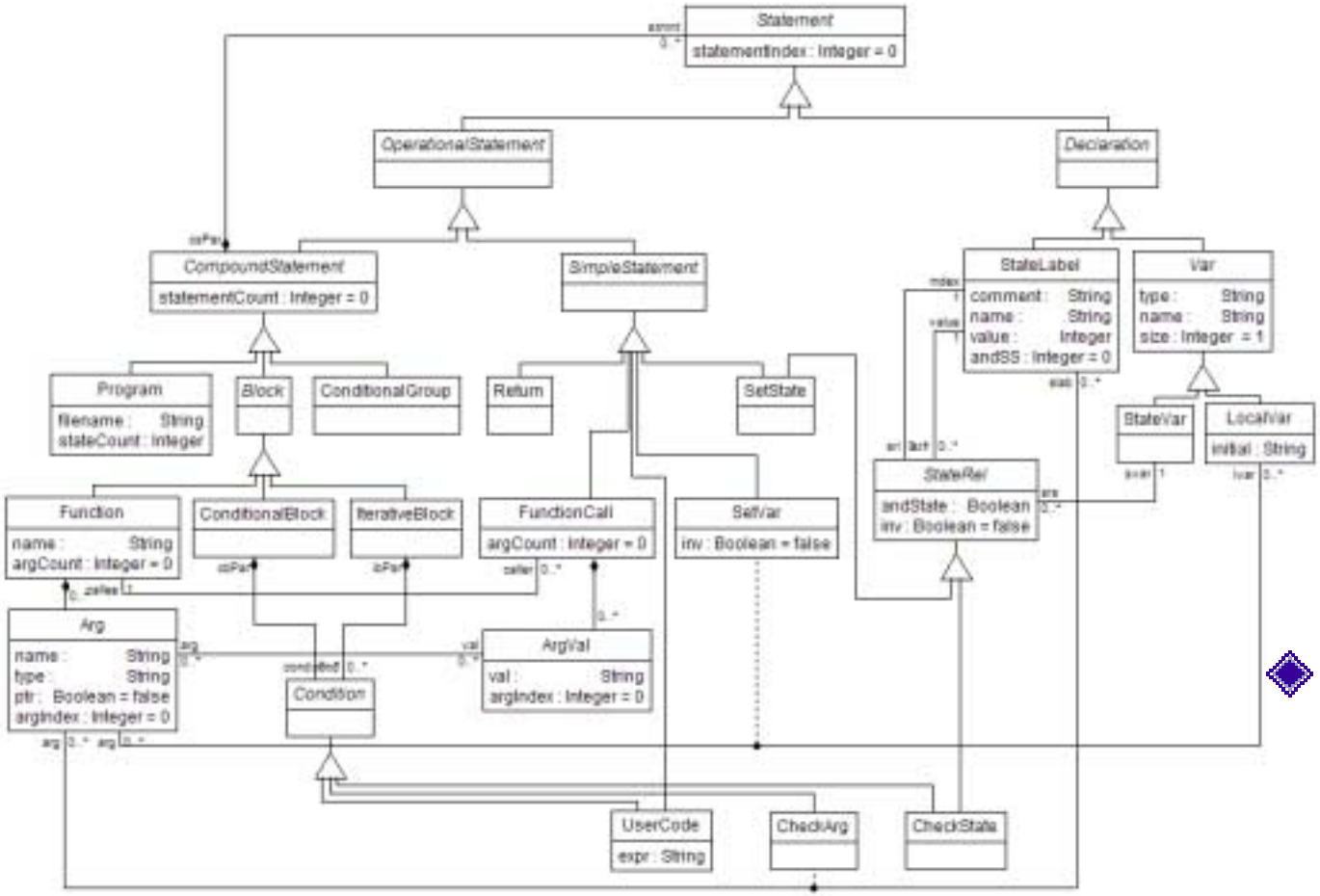
ECSL-DP Code Generator

Stateflow → C

- Transformation specified as a graph-rewrite specification using GReaT
- Transformation target is a stylized subset of C
  - Consistent with the operational semantics of Stateflow defined in Matlab Stateflow documents
  - Stateflow C (SFC) UML meta-model of target
- Transformation output is an object network compliant with the SFC UML meta-model
  - Second stage of transformation performs “anti-parsing”, to print C code text
  - Uses UdmOCLPat

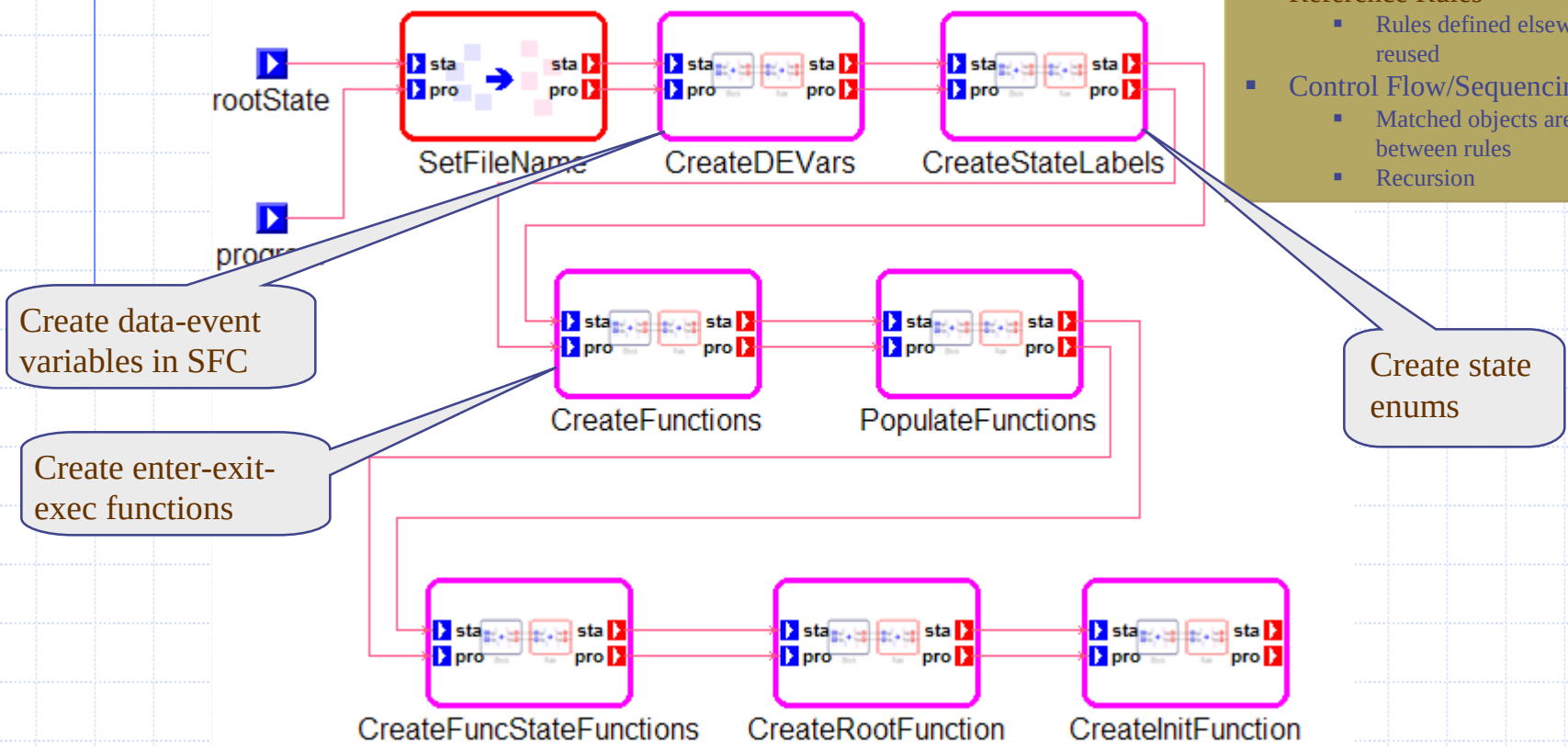
Stateflow → C

SFC UML meta-model



- Key concepts:
  - Compound and Simple Statements
  - Functions, LocalVars, StateVars
  - Conditional, Iterative statements
  - UserCode – user code in models
- Statements are indexed for ordered printing

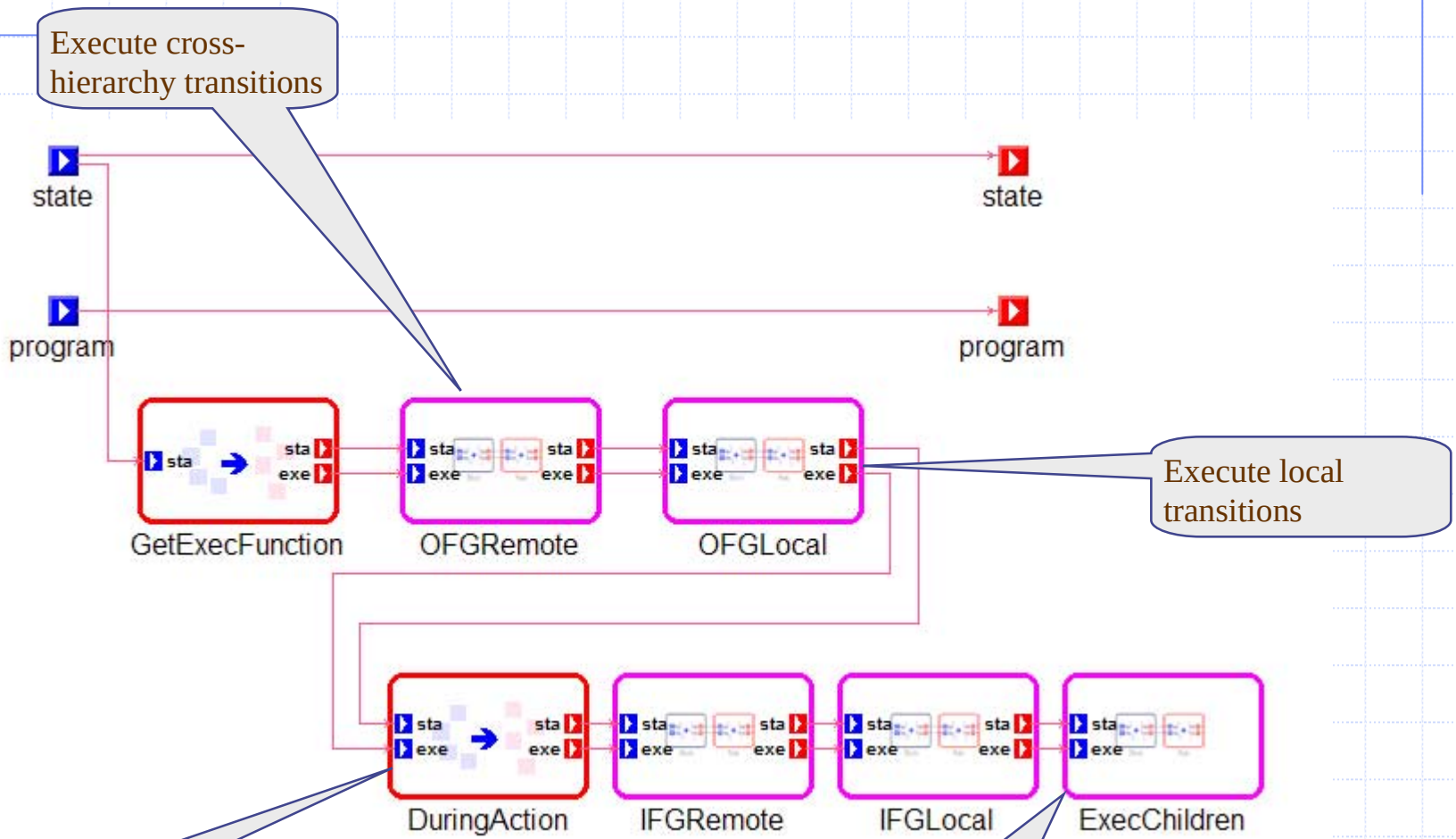
Stateflow → C
Transformation Rules (TL)



- Compound Rules
- Primitive Rules
- Decision Rules
- Reference Rules
  - Rules defined elsewhere can be reused
- Control Flow/Sequencing
  - Matched objects are passed between rules
  - Recursion

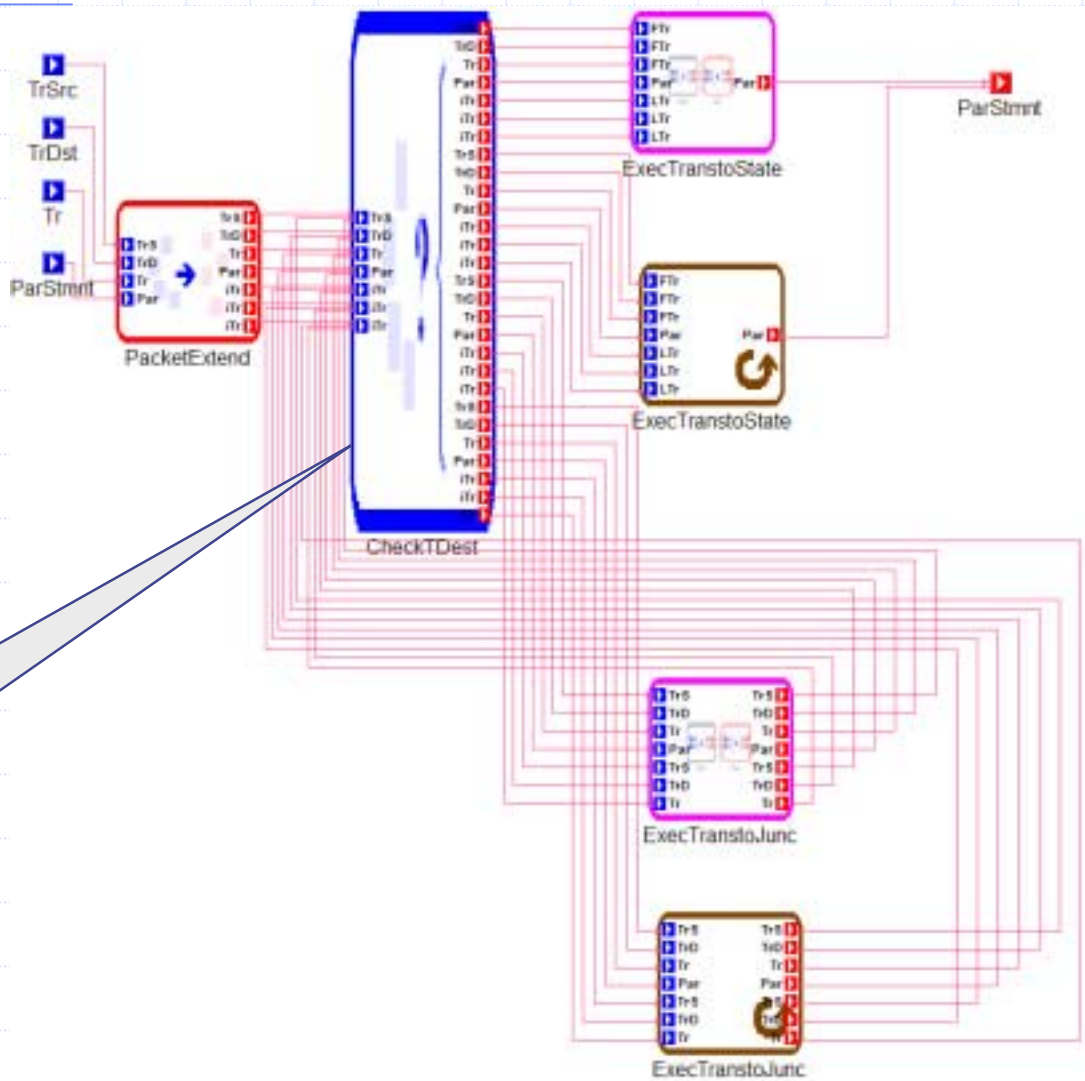
Stateflow → C

PopulateExecFunction Rule



Stateflow → C

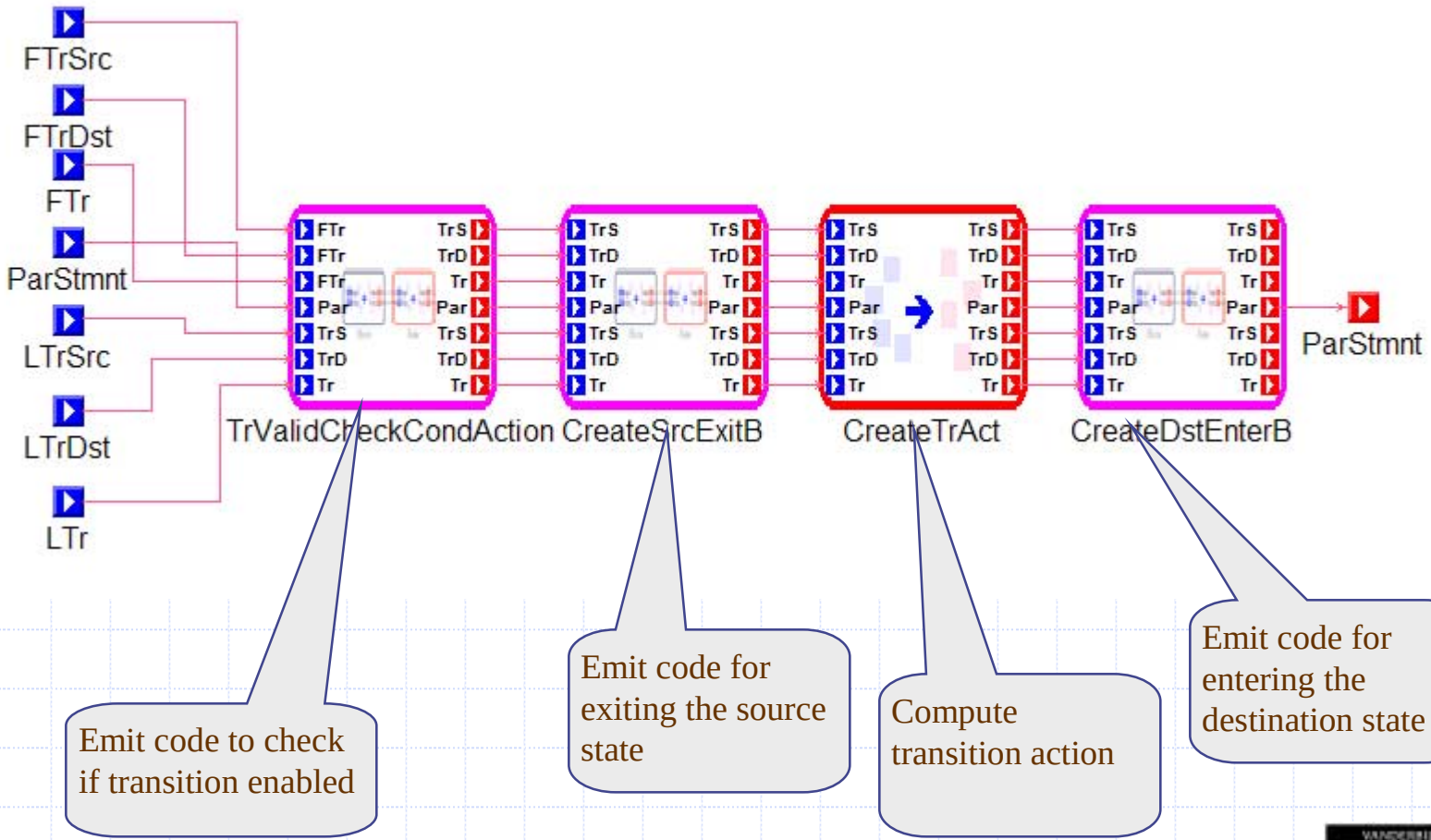
Exec Transition Rule



Is Transition Destination Junction or State?

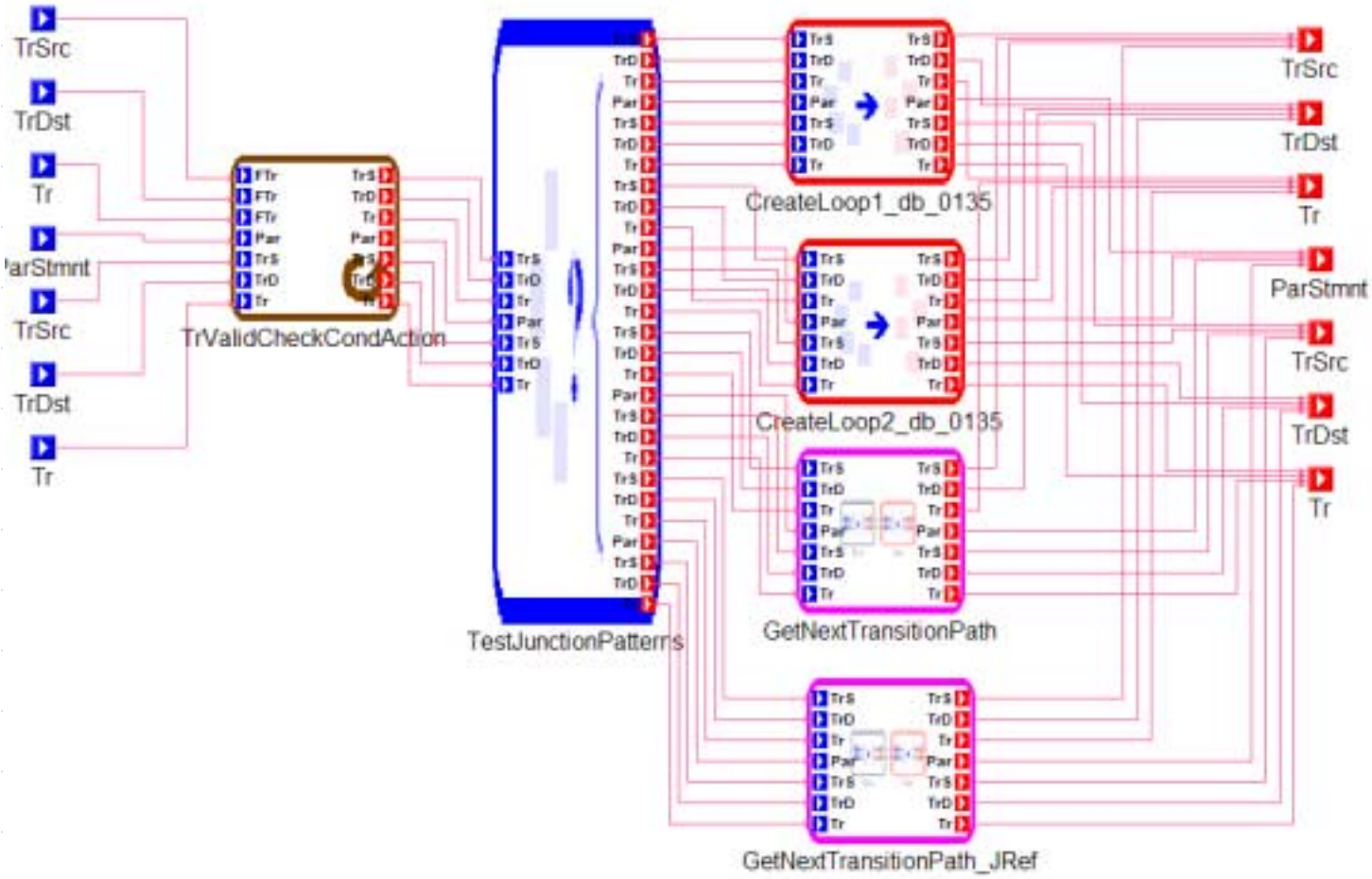
Stateflow → C

Exec Transition to State



Stateflow → C

Exec Transition to Junction



Summary

- Tool-chain
- Model-based Development in VCP
  - Functional modeling
  - Component/System modeling
  - Analysis
  - Generation
- ECSL-DP Capabilities