Visual Coordination Diagrams

FIT VUT Brno, Seminar

David Šafránek
supervised by Luboš Brim

Faculty of Informatics, Masaryk University Brno
Czech Republic

xsafran1@fi.muni.cz

ParaDiSe
Parallel & Distributed Systems Laboratory

Motivation

ParaDiSe
Parallel & Distributed
Systems Laboratory

- architectural description
  - ▷ design vs. implementation of complex systems
  - ▷ primary focus on design issues
  - ▷ component-based structure
    - ○ abstraction, refinement
    - ○ hierarchy
    - ○ component reuse
  - ▷ structure vs. behavior

Motivation

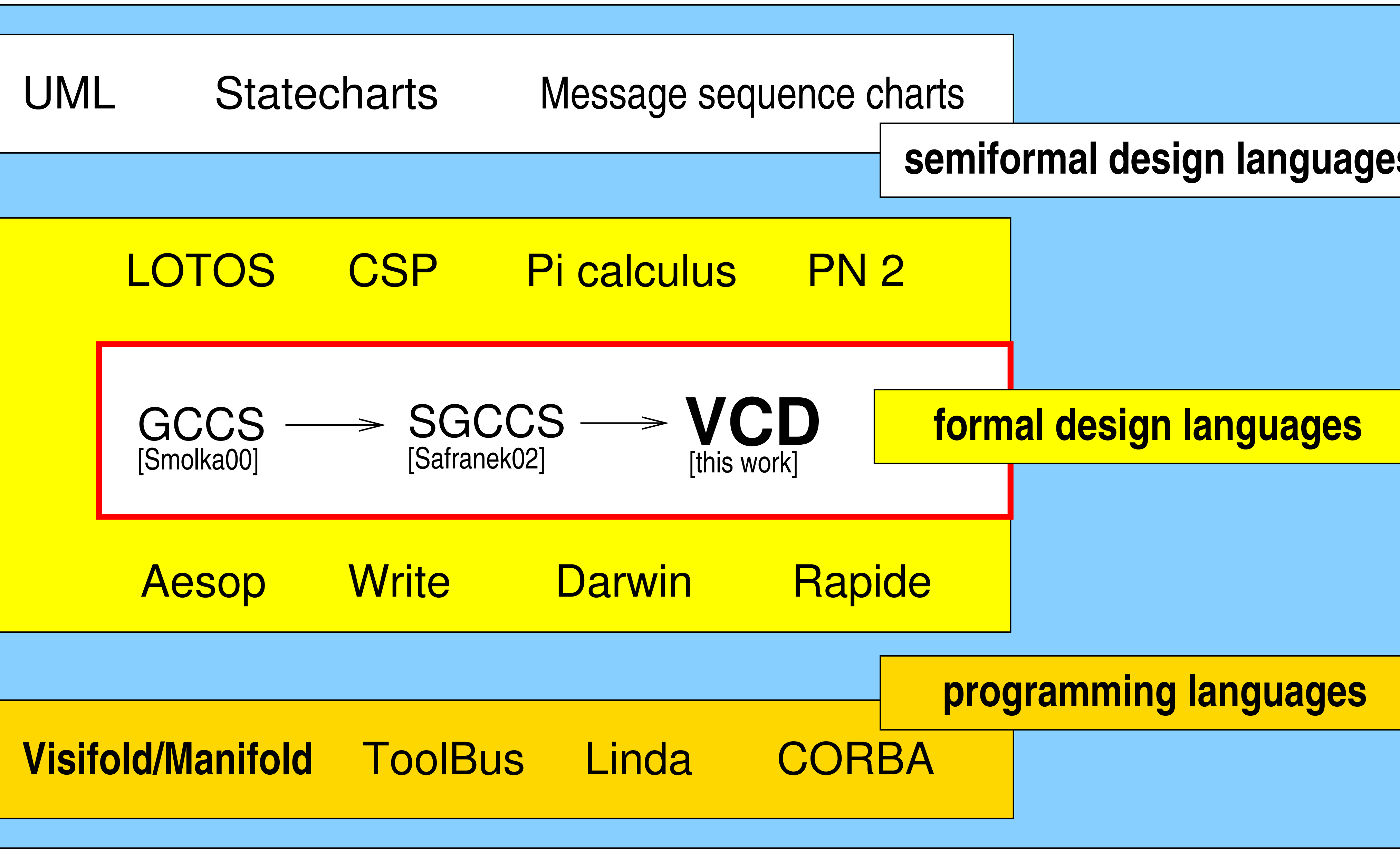
- visual notation
  - ▷ unambiguous rigorous interpretation needed
    - ○ model checking and eqivalence checking
  - ▷ visual notations of UML are not formal
    - ○ structure – communication diagrams
    - ○ behavior – state diagrams
    - ○ coordination – sequence diagrams
  - ▷ some visual formalisms for behavioral description exist
    - ○ architectural formalism can be build above them
    - ○ independency of structural and behavioral aspects
    - ○ call for some heterogeneity

Objectives of this Work

ParaDiSe
Parallel & Distributed
Systems Laboratory

- filling a gap between semi-formal visual notations and programming languages for concurrent systems
- a formal visual design language for concurrent systems
  - ▷ Visual Coordination Diagrams (VCD)
- exogenous coordination model
  - ▷ coordination layer
    - ◦ implicit — VCD
  - ▷ behavioral layer
    - ◦ explicit — Statecharts, Petri-Nets, . . .
- static architecture description

Related Work

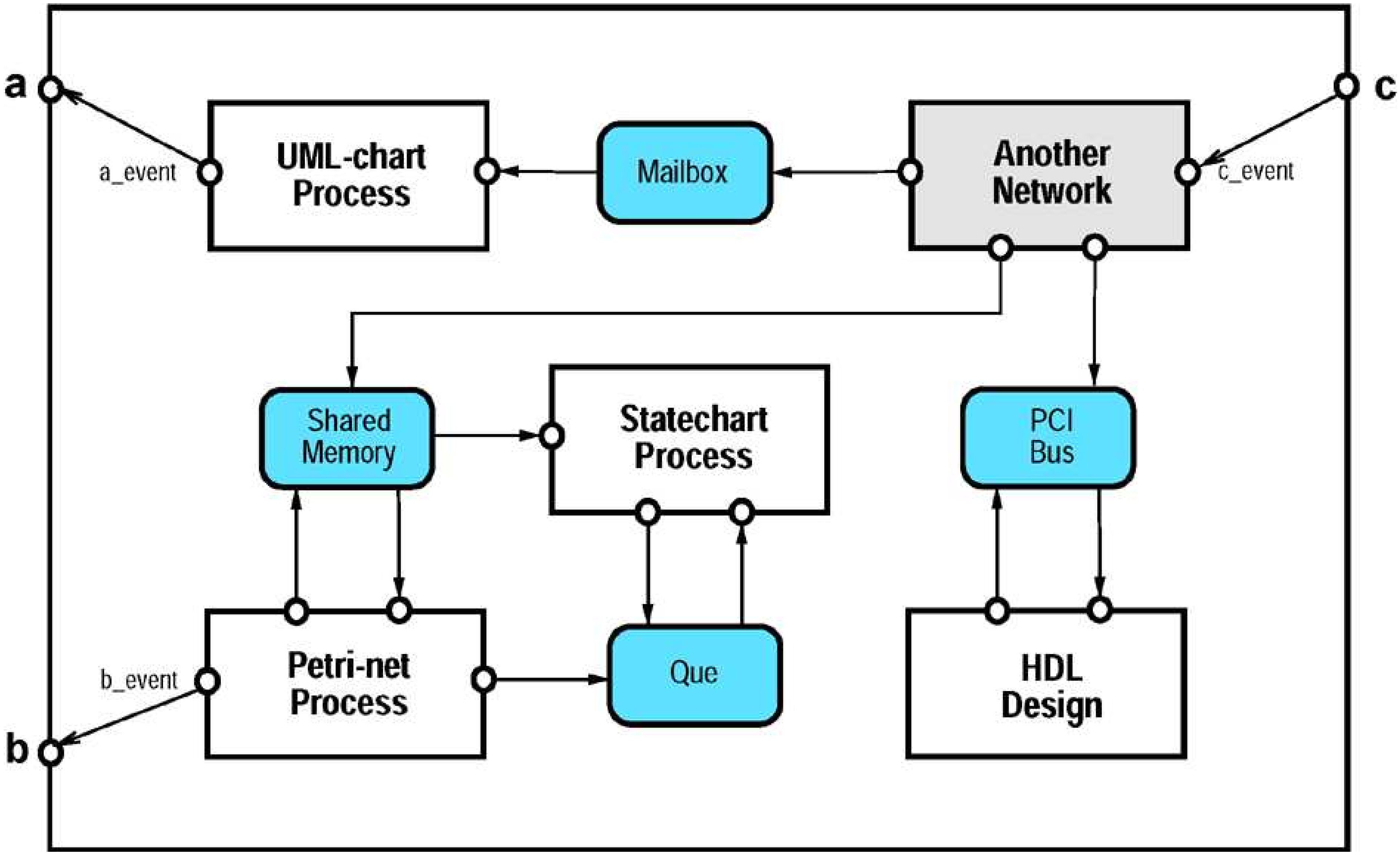


Principal Ideas

- component-based system behavior
  = component behavior + interaction
  [components + connectors]
  - ▷ behavioral (component) specification
    - ○ various models of computation
  - ▷ interaction (connector) specification
    - ○ various communication mechanisms

- component-based systems can be heterogeneous
  - ▷ behavioral-level heterogeneity
  - ▷ interaction-level heterogeneity

Principal Ideas



Principal Ideas

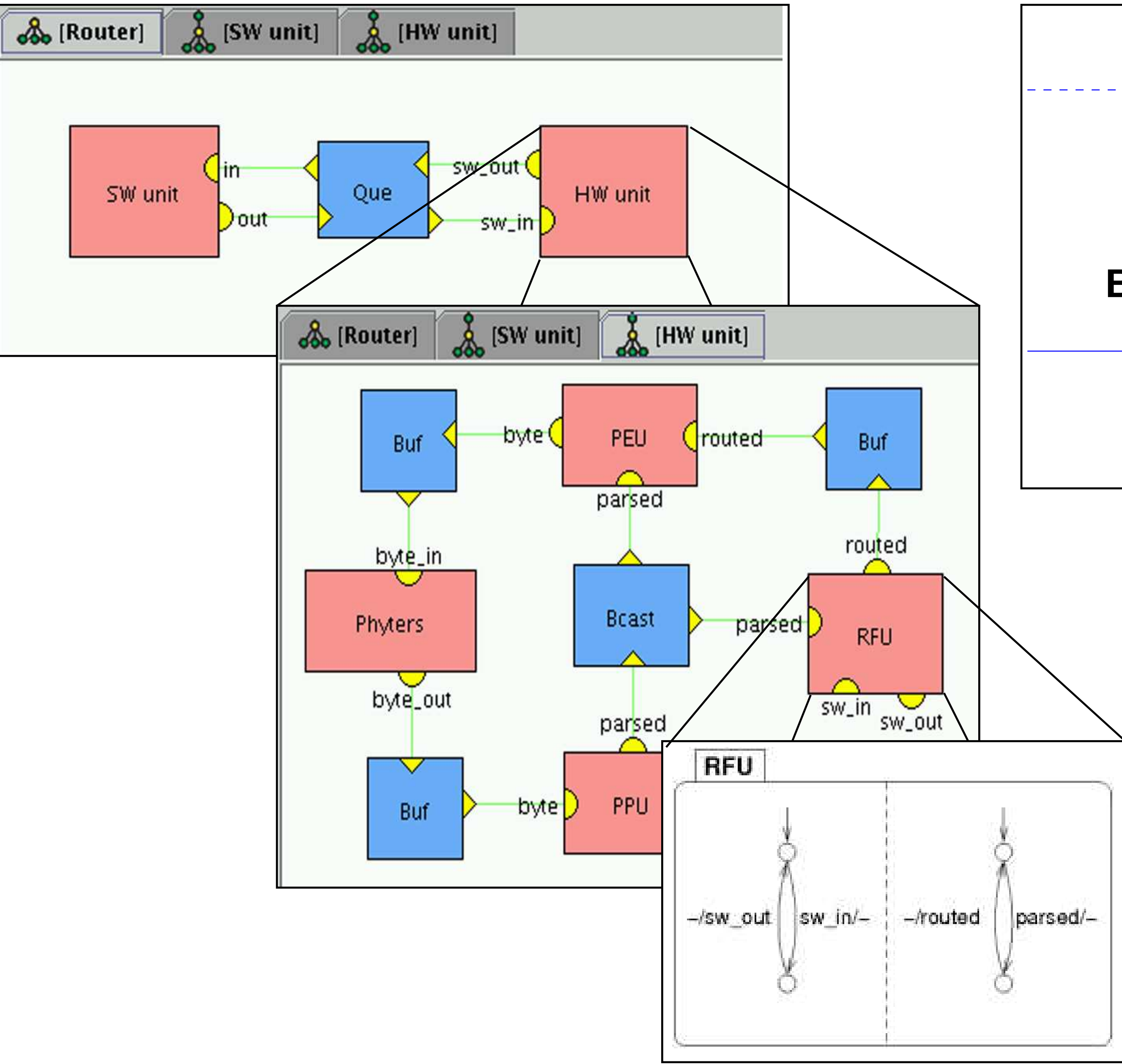
- separation of connectors from components
  - ▷ inspired by Wright [Garlan97]
  - ▷ computational model of a component embedded into interface
  - ▷ interface defined by set of ports
  - ▷ connectors = glue among component interfaces
  - ▷ connectors model protocols of interaction (coordination models)
- hierarchy of components
  - ▷ inspired by SOFA [Plasil98]
  - ▷ recursive structure

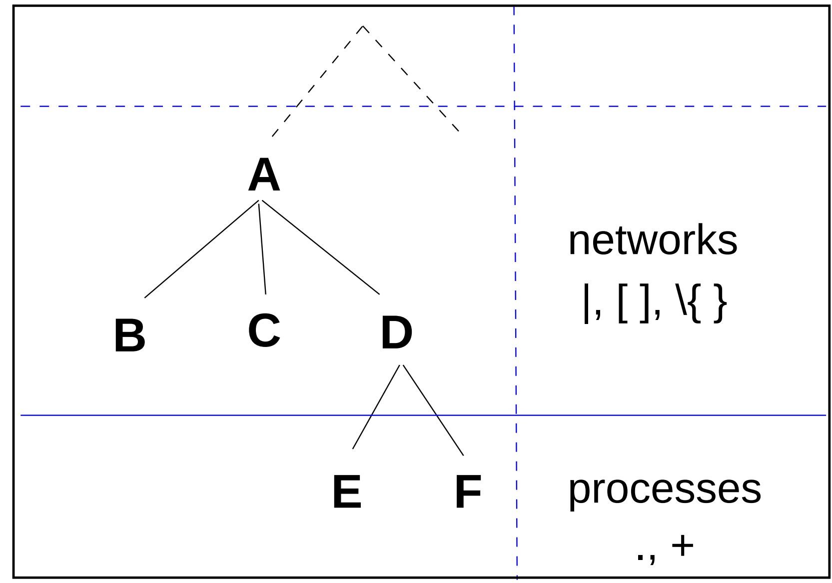
Relation with UML (intuition)

ParaDiSe
Parallel & Distributed
Systems Laboratory

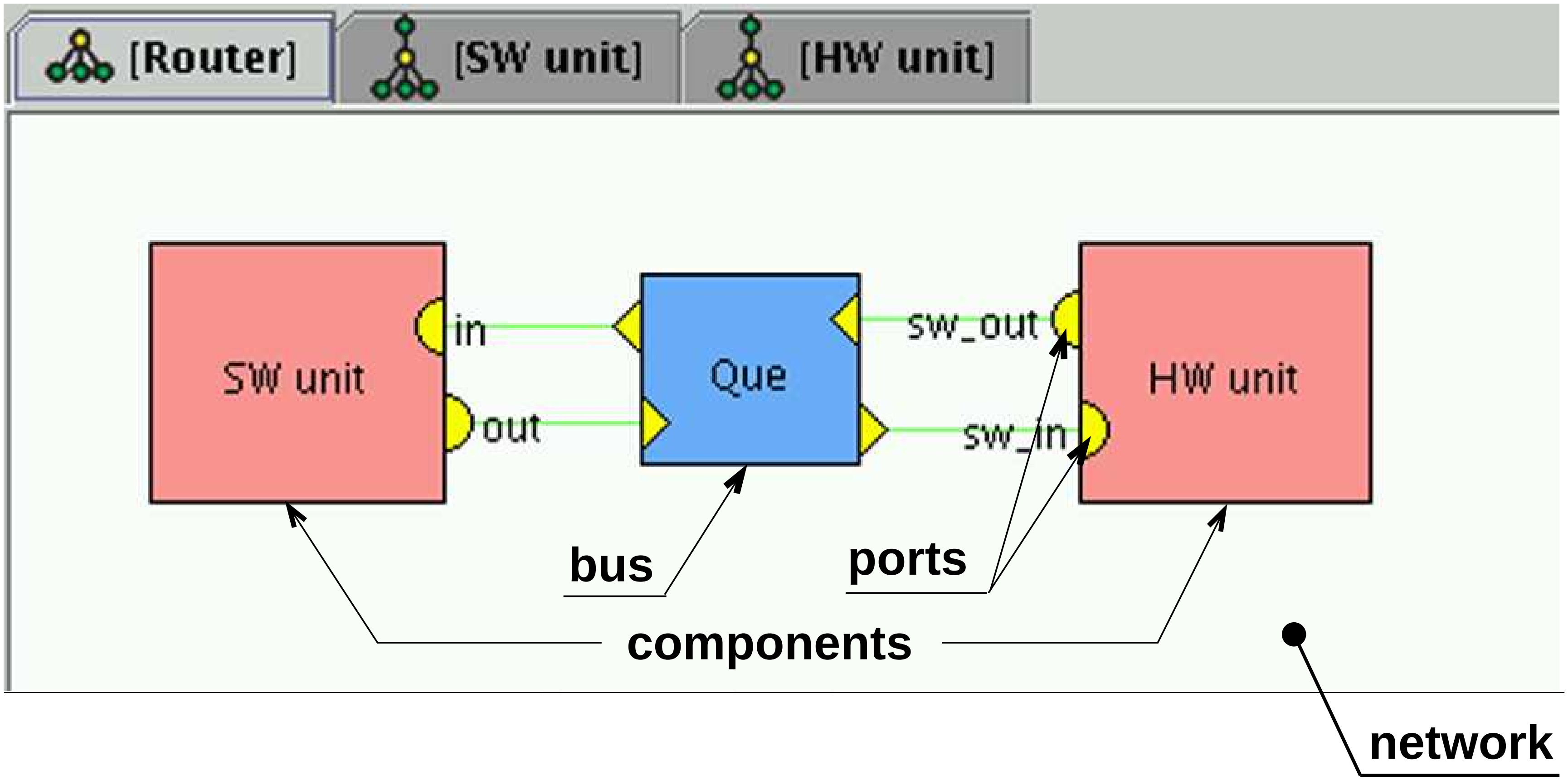
- class diagrams
  - ▷ VCD can be taken as a profile
    - ◦ class of computational components
    - ◦ classes of connectors
- collaboration diagrams and MSCs
  - ▷ static communication infrastructure
- statecharts
  - ▷ UML statecharts can be directly used with VCD

Overview of VCD – Hierarchy



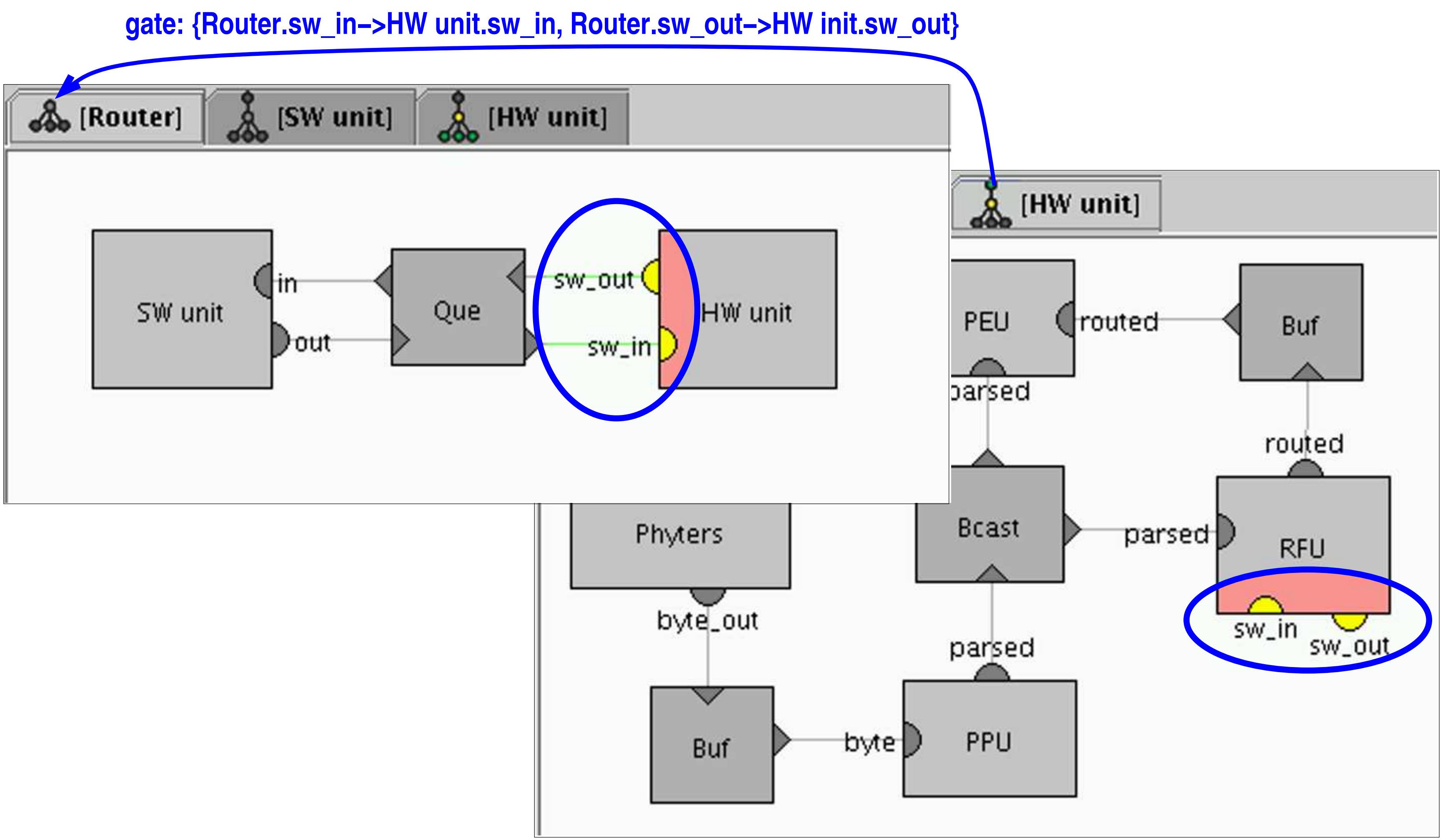


VCD – A Network



- components embedded in networks
- component interfaces contain unidirectional ports
- connectors represented by buses

VCD – Network nesting



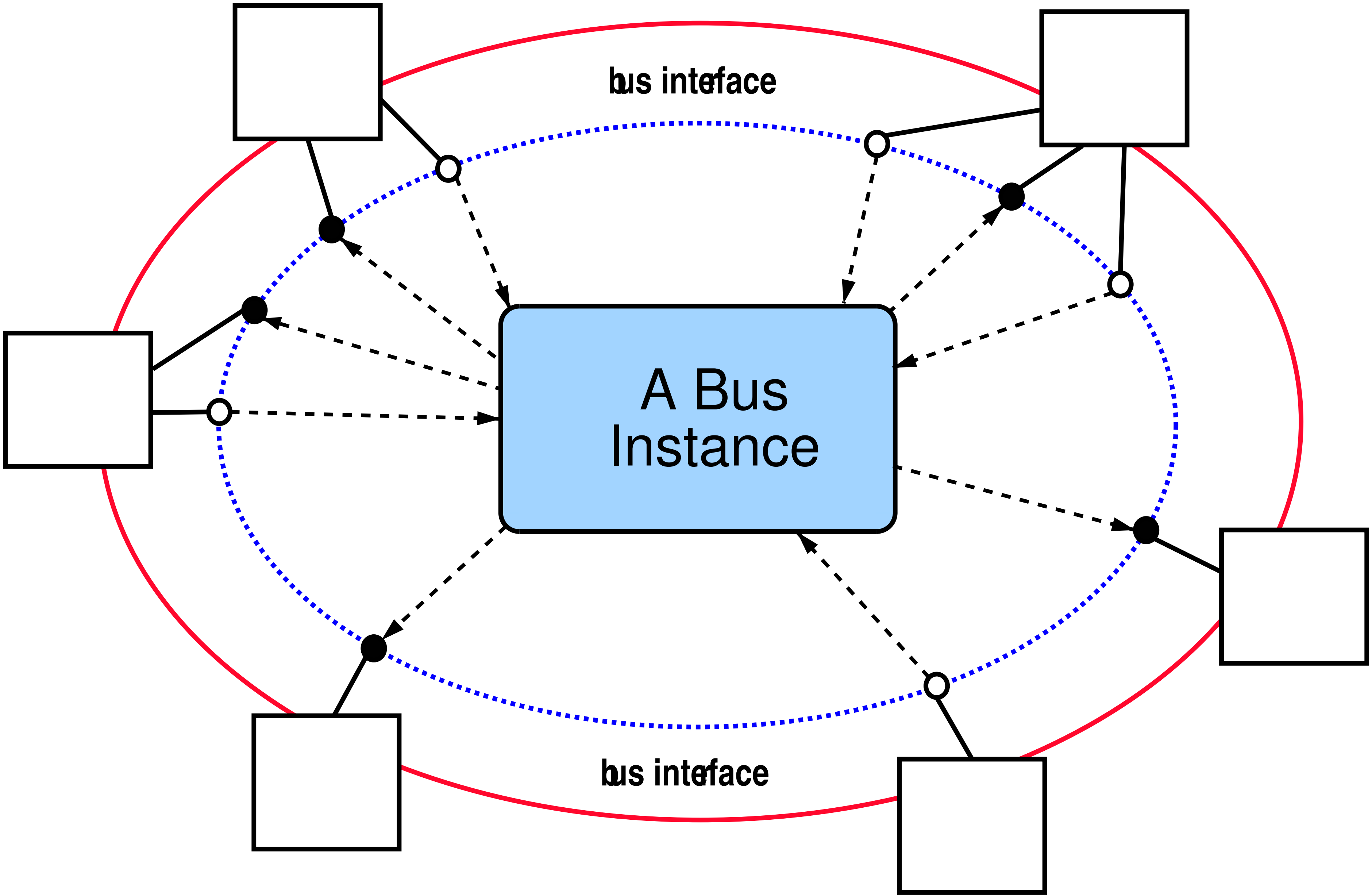
- nested network bound to the encompassing interface by free ports
- binding realized by port names identity or by a network gate

VCD – Buses and Bus Classes

ParaDiSe Parallel & Distributed Systems Laboratory

- atomic network components with reserved semantics
  - ▷ VCD representation of connectors (coordinators)
  - ▷ they cannot be refined with a network
- buses can model various coordination mechanisms
  - ▷ both synchronous and asynchronous types
  - ▷ different buses can be mixed in a particular network
- bus class
  - ▷ a template for a particular coordination mechanism
  - ▷ semantically based on a state-transition logic
- bus instance
  - ▷ an occurrence of a bus in a network

VCD – Buses and Bus Classes



VCD – Buses and Bus Classes

ParaDiSe Parallel & Distributed Systems Laboratory

- $\mathcal{W}$ ... a countable set of output ports
- $\mathcal{R}$ ... a countable set of input ports
- $\mathcal{W} \cap \mathcal{R} = \emptyset$

Bus class $\mathcal{B}$ is a tuple $\langle Q, T, q_0 \rangle$ where

- Q is a (countable) set of states,
- $q_0 \in Q$ an initial state,
- $T \subseteq Q \times 2^{\mathcal{W}} \times 2^{\mathcal{R}} \times Q$ a (countable) transition relation.

Bus instance B of a bus class $\mathcal{B}$ is a tuple $B = \langle I, \mathcal{B} \rangle$ where

- $I = \langle W, R \rangle$, $W \subseteq \mathcal{W}, R \subseteq \mathcal{R}$ (finite) ... a bus interface
- $\mathcal{B}$ ... a bus class.

An Example of a Bus (I)

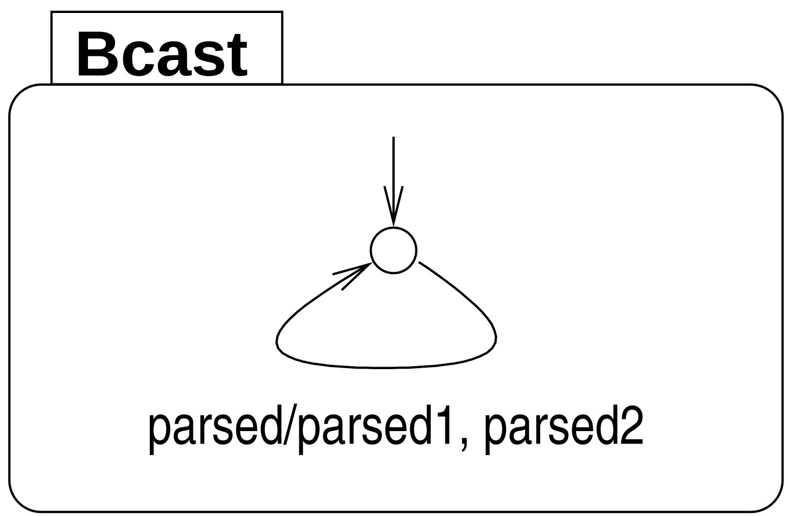
synchronous multicast coordination model

bus class:

$$\mathcal{B}_{mc} = \langle \{q_0\}, T, q_0 \rangle$$

$$\forall w \in \mathcal{W}, \Delta \subseteq \mathcal{R}, \Delta \neq \emptyset.\ \langle q_0, \{w\}, \Delta, q_0 \rangle \in T$$

bus instance:

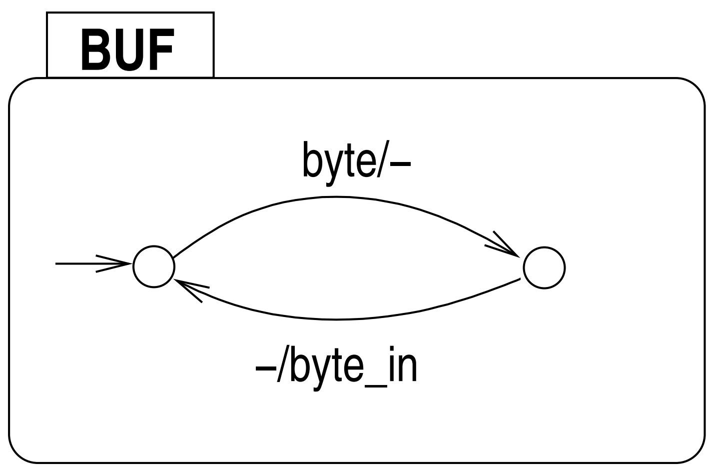


An Example of a Bus (II)

asynchronous message passing coordination model

- $\mathcal{B}_{buf} = \langle Q, T, q_0 \rangle$
- $Q = \{q_w \,|\, w \in \mathcal{W}\} \cup q_0$
- T is defined by the following expression:

$$\forall w \in \mathcal{W}.\ \langle q_0, \{w\}, \emptyset, q_w \rangle \in T$$
$$\wedge \forall q_x \in Q, q_x \neq q_0,\ r \in \mathcal{R}.\ \langle q_x, \emptyset, \{r\}, q_0 \rangle \in T$$



VCD – Behavioral Layer

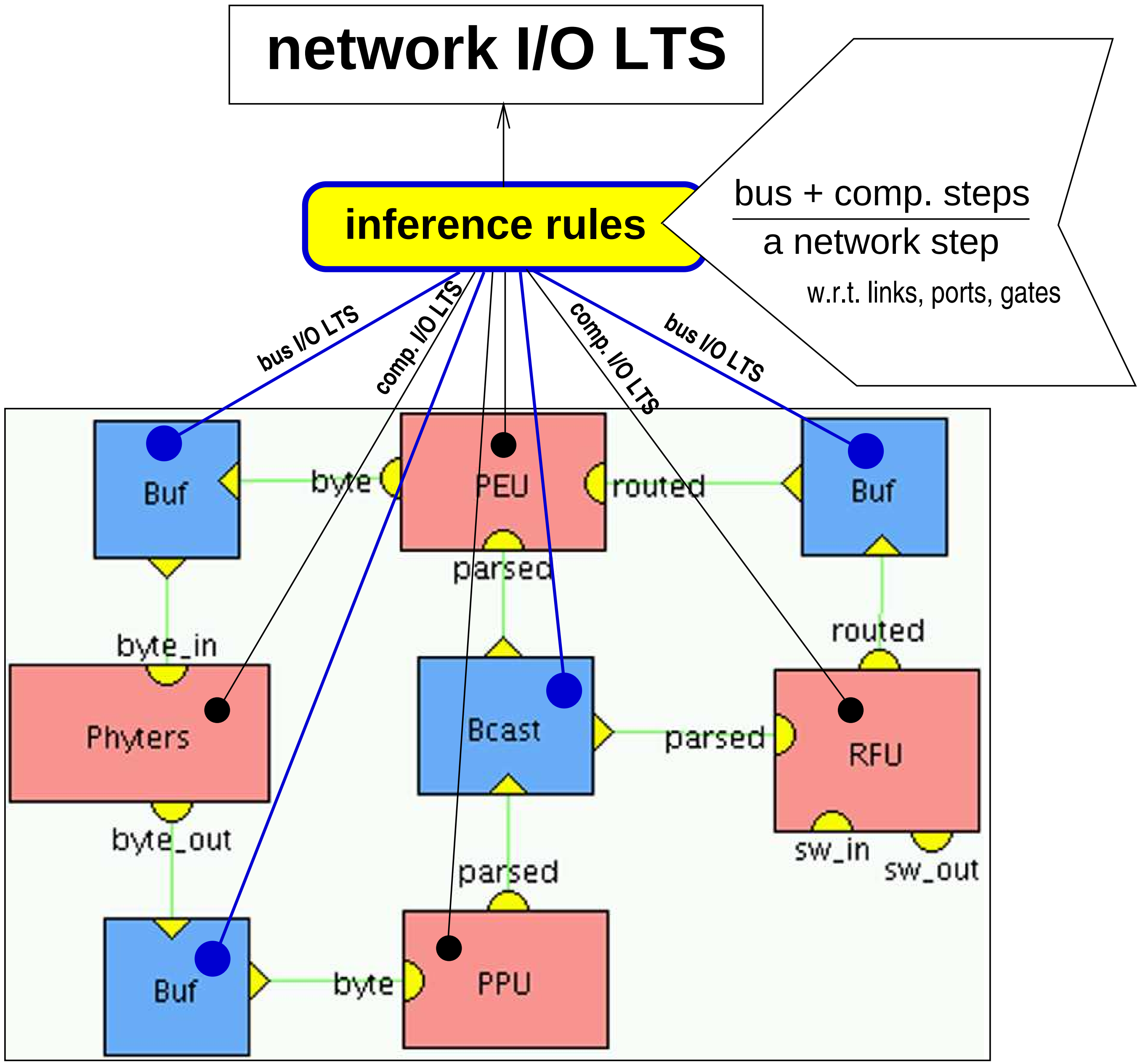
ParaDiSe
Parallel & Distributed
Systems Laboratory

- can be defined using any formalism semantically compatible with the notion of I/O LTS
- multilinguality in the scope of expressiveness of I/O LTS
- we use set-labeled LTSs to capture Statecharts, Petri-Nets, ...

An I/O LTS is a tuple $\langle Q, T, q_0 \rangle$ where

- Q is a set of states (potentially infinite),
- $q_0 \in Q$ an initial state,
- $T \subseteq Q \times 2^{\mathcal{R}} \times 2^{\mathcal{W}} \times Q$ a transition relation.

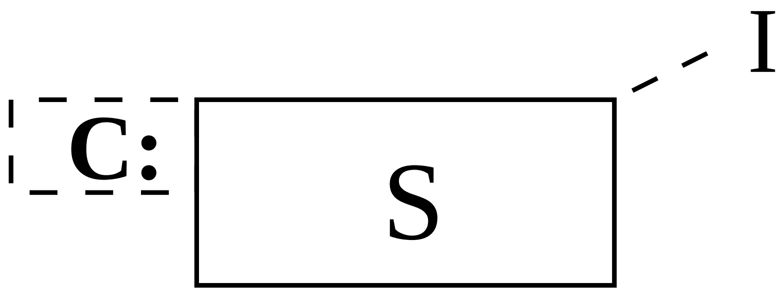
VCD – Semantics



VCD – Semantics (II)

ParaDiSe Parallel & Distributed Systems Laboratory

component body S inserted in interface I makes a VCD component C

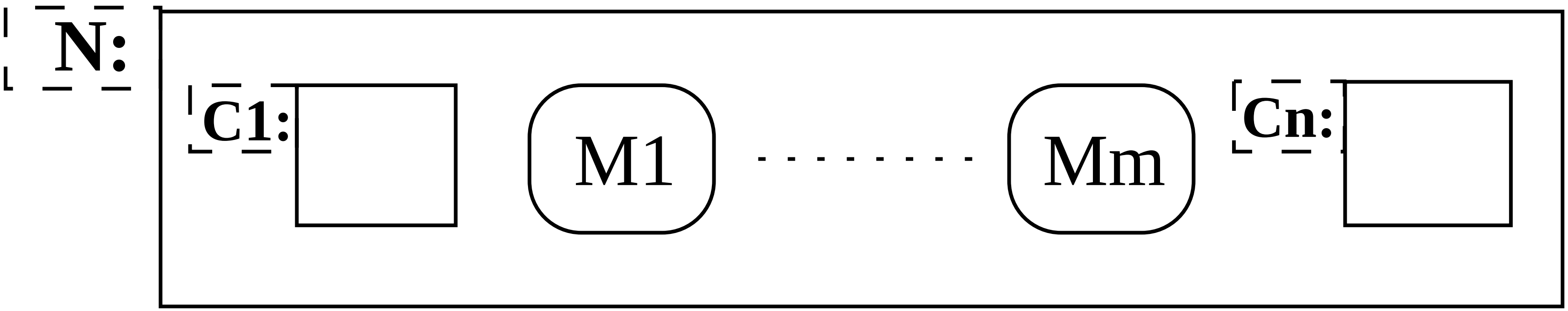


$$\frac{\text{transition of S:} \quad q \xrightarrow[\Delta]{\Gamma} q' \qquad (\Gamma \subseteq ports(S) \cap \mathcal{R} \text{ and } \Delta \subseteq ports(S) \cap \mathcal{W})}{\text{transition of C:} \quad q \xrightarrow[I^W \cap \Delta]{I^R \cap \Gamma} q'}$$

where $ports(S) \subseteq \mathcal{W} \cup \mathcal{R}$ is a set of all actions of S

VCD – Semantics (III)

components $C_1 \ldots C_n$ **and buses** $M_1 \ldots M_m$ **inserted in network** N



- stand-alone components — interleaving
- components connected to buses — interleaving or synchronization

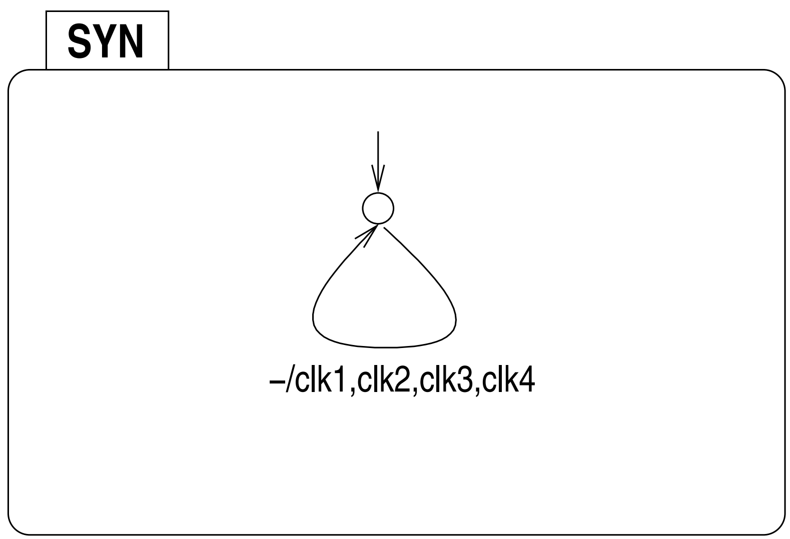
Synchronous Behavior

bus class:

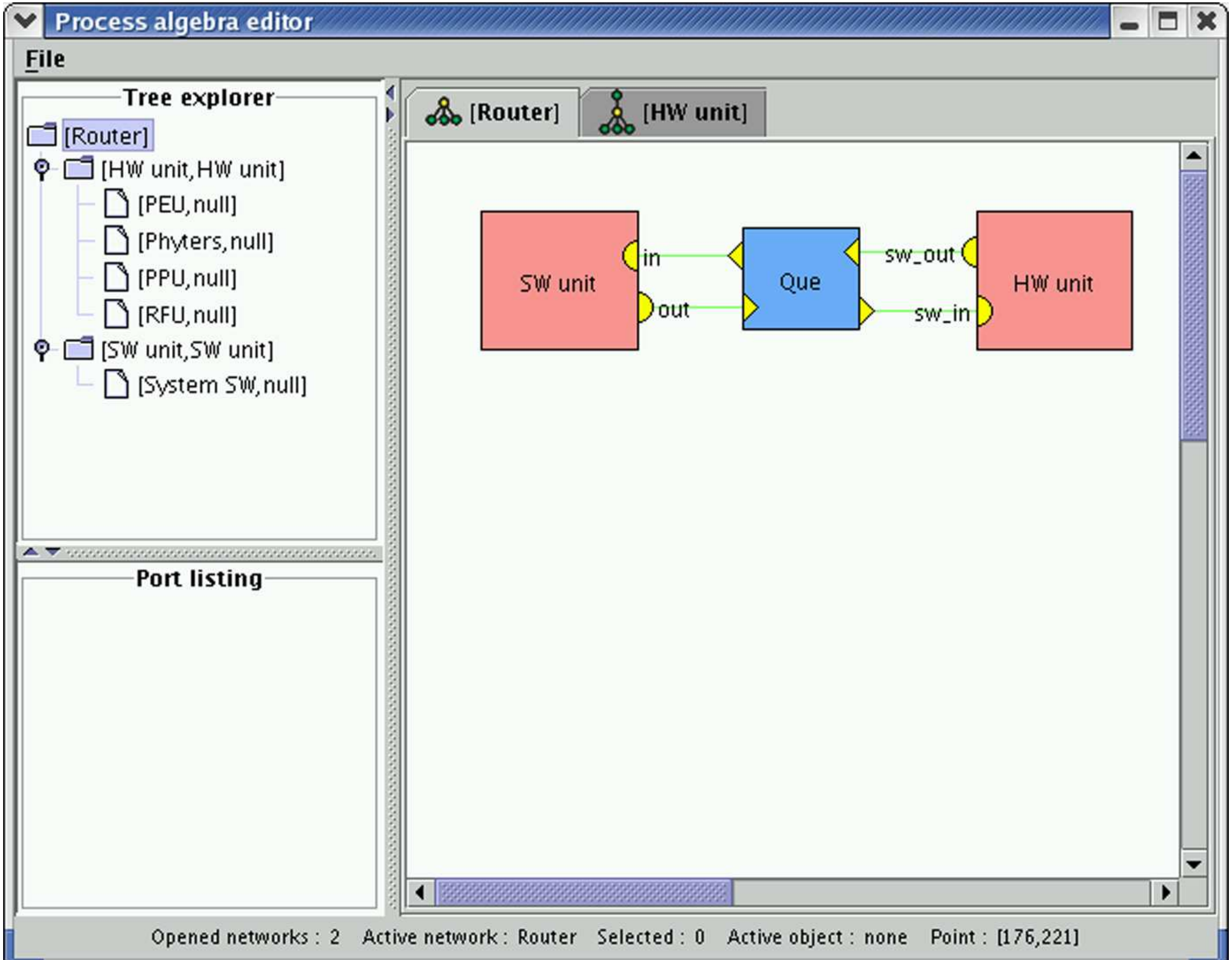
$$\mathcal{B}_{clk} = \langle \{q_0\}, T, q_0 \rangle$$

$$\forall \Gamma \subseteq \mathcal{R}, \Delta \neq \emptyset.\ \langle q_0, \emptyset, \Gamma, q_0 \rangle \in T$$

bus instance:



Tool support (prototype)



Conclusion

ParaDiSe
Parallel & Distributed
Systems Laboratory

- we built a simple design notation with formal semantics
- static hierarchical coordination model
- Statecharts-like models can be used for atomic processes

	VCD	UML	SOFA	Manifold
hierarchy	networks	object aggregation	compound comps.	meta-coordinators
rchitecture	static	static	dynamic (DCUP)	dynamic
terogeneity	various buses state-transition models	connectors = comps. UML objects	gen. connectors diff. paradigms	asynch. channels different paradigms
ltilinguality	LTS with event-sets (Statecharts, Petri-Nets, . . .)	UML statecharts	Java, C++,. . .	C++, Fortran, . . .
application	design of distr. SW, HW, synchronous systems	design of SW async. systems	implement. of distr. SW	design and implementatio of parallel/distr. systems

Future Work

ParaDiSe
Parallel & Distributed
Systems Laboratory

- typed value-passing support
- relation with UML
- extending the network layer (classes of buses)
- extending the behavioral layer (Petri-Nets,. . .)
- improving implementation
- connection with verification tools (DiVinE)

Publications

VCD: A Visual Formalism for Specification of Heterogeneous Software Architectures

with J.Simša, accepted to SOFSEM 2005

Visual Specification of Systems with Heterogeneous Coordination Models

in proceedings of FOCLASA 2004

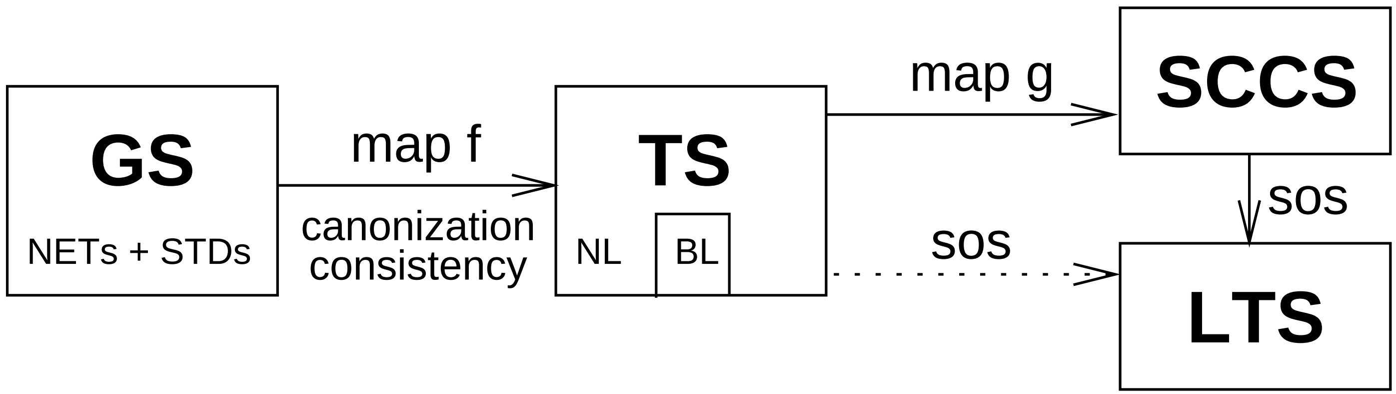
Visual Specification of Concurrent Systems

in proceedings of ASE 2003

SGCCS: A Graphical Language for Real-Time Coordination

in proceedings of FOCLASA 2002

SGCCS Semantics



$$S_i \sim S_i' \;\Rightarrow\; \langle\langle\langle S_1, I_1\rangle, \ldots, \langle S_i, I_i\rangle, \ldots, \langle S_n, I_n\rangle\rangle, B, L\rangle$$
$$\sim \langle\langle\langle S_1, I_1\rangle, \ldots, \langle S_i', I_i\rangle, \ldots, \langle S_n, I_n\rangle\rangle, B, L\rangle$$

$$g(S_i) \sim g(S_i') \;\Rightarrow$$
$$((g(S_1)\backslash R_1)[F_1] \;\times\; \cdots \;(g(S_i)\backslash R_i)[F_i] \;\cdots\; \times\; (g(S_n)\backslash R_n)[F_n])\backslash R[F]$$
$$\sim ((g(S_1)\backslash R_1)[F_1] \;\times\; \cdots \;(g(S_i')\backslash R_i)[F_i] \;\cdots\; \times\; (g(S_n)\backslash R_n)[F_n])\backslash R[F]$$