Discussion for C-based SLDLs: SpecC and SystemC

Weiwei Chen and Rainer Doemer
Center for Embedded Computer Systems
University of California, Irvine
weiwei.chen@uci.edu, doemer@uci.edu

SpecC

- A true language, superset of ANSI-C
- Separate Concerns of Computation and Communication
- Computation
  - Behavioral Hierarchy (***behaviors***)
  - Structural Hierarchy (***behaviors, ports***)
  - Follows the same semantics and syntax as the function in C.
- Communication
  - Data could be transferred through variables via ***ports***
  - Date could be transferred through hierarchical ***channels***.
- Execution Semantics
  - Discrete event (DE) simulation
  - Explicit support for static execution sequence (***pipe***, ***par***, ***fsm*** and ***seq****)*
  - Syntactical support for dynamic execution sequence (***event-wait-notify***)

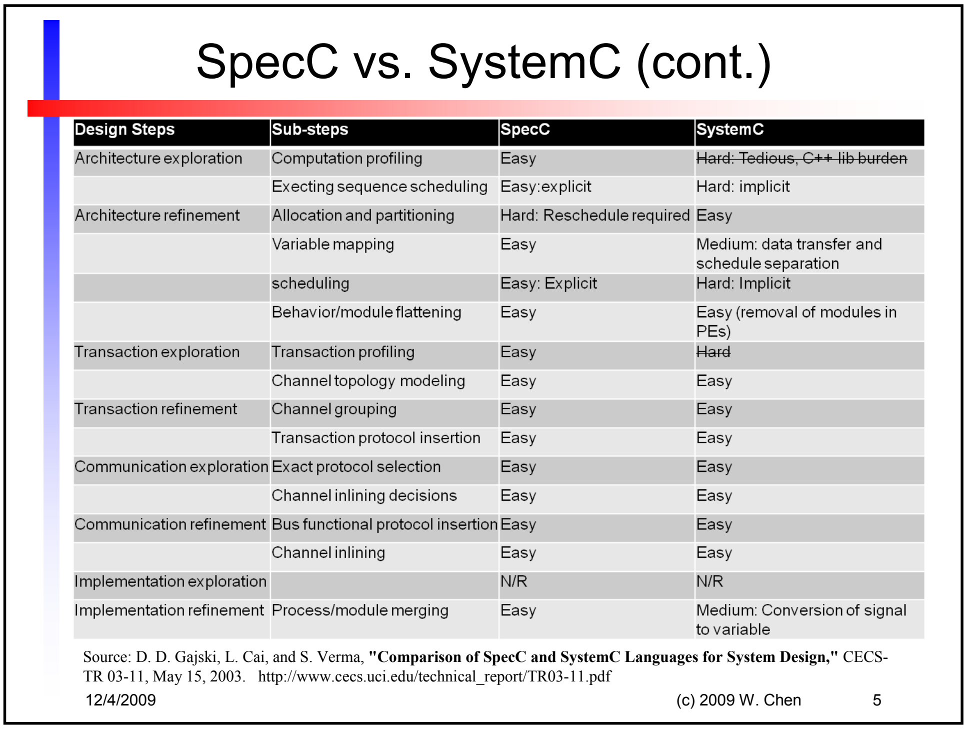
SystemC

- A Class library of C++
- Separate Concerns of Computation and Communication
- Computation
  - Structural entities are modeled by ***Modules*** (consists of other ***modules*** or ***processes***). Structural Hierarchy (***modules, ports***)
  - Hierarchical modeling of ***Processes*** is not possible.
  - Functions are defined the same way as in C.
- Communication
  - Data could be transferred through signals via ports or signals/variables declared in the module.
  - Date could be transferred through channels via variables declared in the module.
- Execution Semantics
  - Discrete event (DE) simulation
  - Support for static sensitivity by using "***sensitivity list***" of a process
  - Support for Dynamic sensitivity by ***event-wait-notify***. (***event-wait-notify*** can be used between processes in one ***module***, or be encapsulated in a ***channel*** for different module synchronization.)

SpecC vs. SystemC

Abstract Models	Model Aspect	SpecC	SystemC
Specification Model	functional block	behavior	module
	schedule	event, definition (par, ...)	event, signal
	data transfer	variable	signal
IP-assembly model	structure blocks	behavior	module
	functional blocks	behavior	process
	schedule inside PEs	event, definition (par, ...)	event, signal
	schedule between PEs	channel	channel
	data transfer inside PEs	variable	signal
	data transfer between PEs	channel	channnel
Implementation Model	fsm	fsmd	switch(SC_THREAD), SC_CTHREAD
	function units	function/behavior	function/module
	storage variable	buffered signal	signal
	Bus	bit	bit
	control signal	signal	signal

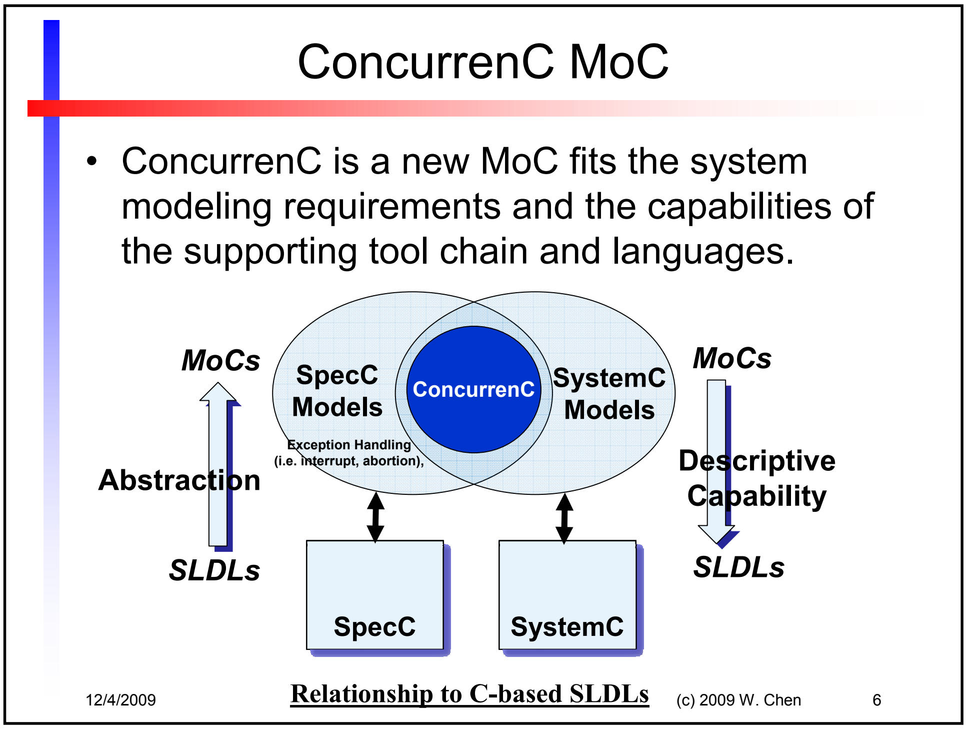
Source: D. D. Gajski, L. Cai, and S. Verma, **"Comparison of SpecC and SystemC Languages for System Design,"** CECS-TR 03-11, May 15, 2003. http://www.cecs.uci.edu/technical_report/TR03-11.pdf



SpecC vs. SystemC (cont.)

Design Steps	Sub-steps	SpecC	SystemC
Architecture exploration	Computation profiling	Easy	~~Hard: Tedious, C++ lib burden~~
	Execting sequence scheduling	Easy:explicit	Hard: implicit
Architecture refinement	Allocation and partitioning	Hard: Reschedule required	Easy
	Variable mapping	Easy	Medium: data transfer and schedule separation
	scheduling	Easy: Explicit	Hard: Implicit
	Behavior/module flattening	Easy	Easy (removal of modules in PEs)
Transaction exploration	Transaction profiling	Easy	~~Hard~~
	Channel topology modeling	Easy	Easy
Transaction refinement	Channel grouping	Easy	Easy
	Transaction protocol insertion	Easy	Easy
Communication exploration	Exact protocol selection	Easy	Easy
	Channel inlining decisions	Easy	Easy
Communication refinement	Bus functional protocol insertion	Easy	Easy
	Channel inlining	Easy	Easy
Implementation exploration		N/R	N/R
Implementation refinement	Process/module merging	Easy	Medium: Conversion of signal to variable

Source: D. D. Gajski, L. Cai, and S. Verma, **"Comparison of SpecC and SystemC Languages for System Design,"** CECS-TR 03-11, May 15, 2003. http://www.cecs.uci.edu/technical_report/TR03-11.pdf



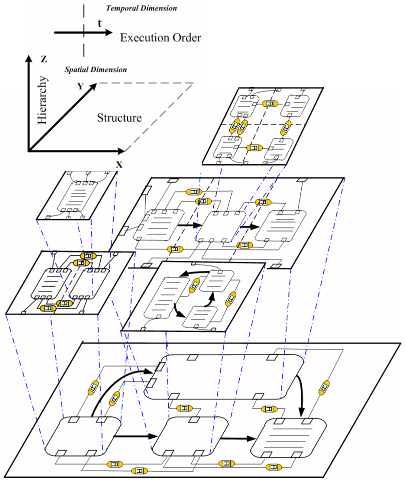
ConcurrenC MoC

- ConcurrenC is a new MoC fits the system modeling requirements and the capabilities of the supporting tool chain and languages.

Relationship to C-based SLDLs

ConcurrenC MoC

- ConcurrenC Features:
  - Clear separation of communication and computation
  - Hierarchical Structural abstraction: blocks (internally c-based), channels and interfaces.
  - Communication abstraction: fix to a set of pre-defined channels
  - Explicit support of Concurrency
  - Timing
  - Execution


Temporal Dimension
t
Execution Order
Z
Spatial Dimension
Hierarchy
Y
Structure
X