SystemC and the future of design languages: Opportunities for users and research

Grant Martin
Fellow, Cadence Berkeley Labs

SBCCI 2003, São Paolo, Brazil, 8-11 Sept 2003
9 September 2003: 10:20-11:10

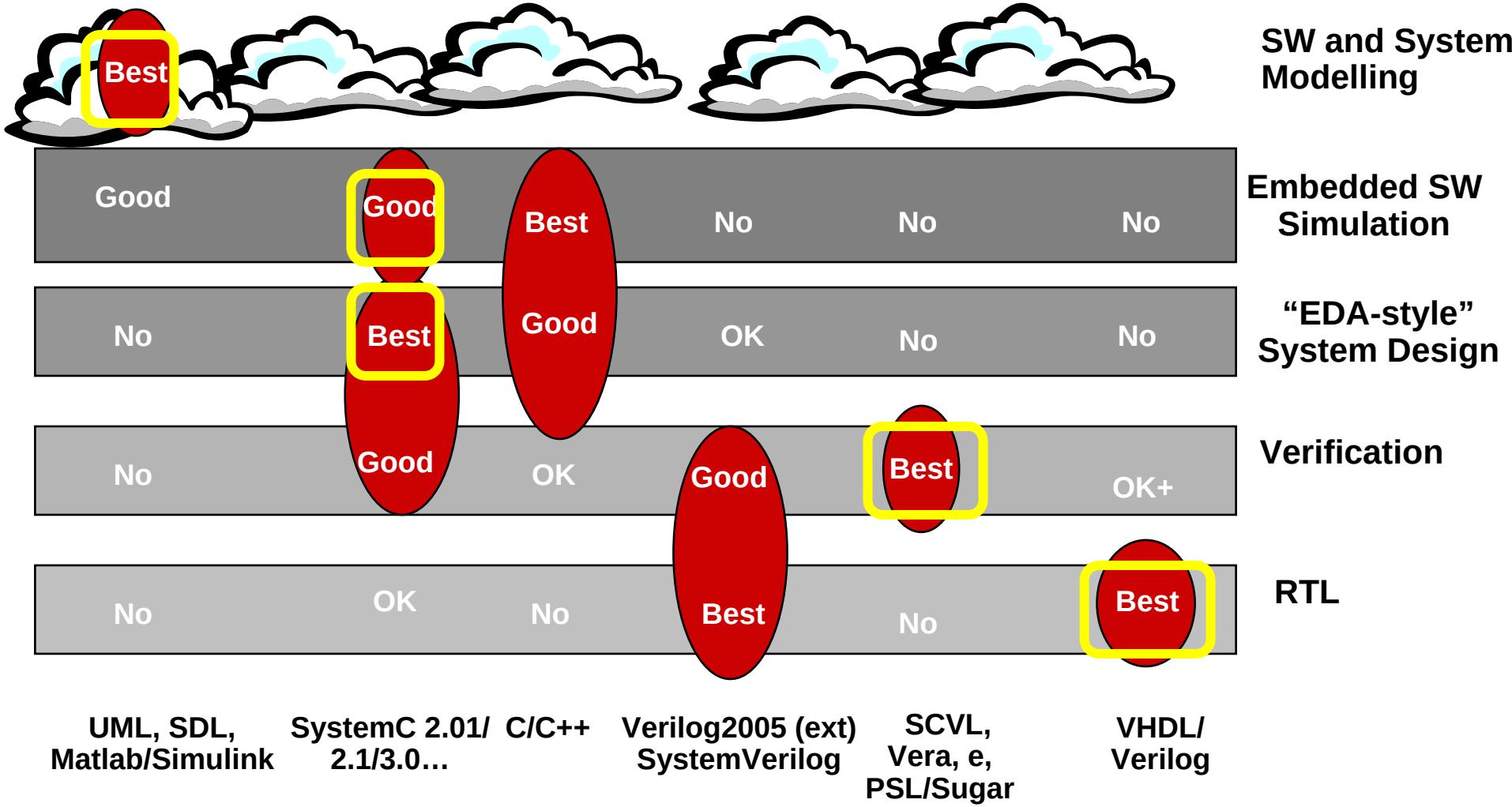
Outline

- Alphabet Soup
- A language war?
- Or peaceful co-existence
- SystemC, HDLs and software modelling
- New Methodologies and Flows
- New Research Opportunities
- Conclusion

Alphabet Soup

VHDL
SDL
Vera
SystemC
PSL
UML
Sugar
OVL
Verilog-A
VHDL-AMS
OpenVera
SystemVerilog
OVA
Verilog-2005
e
Verilog
Matlab/Simulink

Making Sense of the Alphabet Soup



No one language does everything.

A “typical” System-on-Chip Design Project?

- Take a microprocessor and its Instruction Set Simulator (ISS)…modelled in **SystemC**
- Add in some embedded SW – **C** or **C++**
- Add in some digital IP blocks – some modelled in **Matlab** (dataflow) and some created in **Verilog**
- Buy some 3rd party digital IP – from a **VHDL**-based supplier
- Need to have some AMS interfaces – **Spice** models
- Need to validate AMS interfaces in SoC context – **Verilog-A**
- Build some testbenches – **e**
- Create some block assertions – **PSL**

…….is 9 different languages or notations enough?

A Language War?

- **SystemC won't go away quietly**
  By John Cooley, EE Times, Jun 23, 2003
- **Cadence IEEE donation overlaps SystemVerilog**
  By Richard Goering, EE Times, Jun 2, 2003)
- **EDA language dispute erupts in advance of DAC**
  By Richard Goering, EE Times, May 30, 2003
- **New EDA consortium promotes assertion language**
  By Richard Goering, EE Times, May 22, 2003
- **VHDL, the new Latin**
  By John Cooley, EE Times, Apr 7, 2003
- **Verilog 2001 compliance lacking, designer says**
  By Richard Goering, EE Times, Feb 26, 2003
- **DVCon: SystemVerilog key to new design paradigm**
  By Michael Santarini, EE Times, Feb 24, 2003 (7:45 PM)
- **C loves me, C loves me not**
  By Richard Goering, EE Times, Sep 16, 2002
- **Synopsys donation defuses assertion language war**
  By Richard Goering, EE Times, June 11, 2002

Or Peaceful coexistence?

CMP United Business Media

EEdesign
An EE Times Community

Home | Newsletter | About | Feedback | Contact | Media Kit

EETIMES NETWORK

Your resource for design tools and methodologies

e search go

EEdesign Links
- NEWS
- EXCLUSIVE FEATURES
- COMMENTARY
- SILICON ENGINEERING
- ISD MAGAZINE ARCHIVES
- DEEPCHIP (ESNUG)
- EDA GLOSSARY NEW

Buyer's Guides
- EDA TOOLS
- VCX GATEWAY
- IP CATALOG
- EMBEDDED PRODUCTS

Industry Links
- NETSEMINARS
- WHITE PAPERS
- BOOKS
- CONFERENCES
- INDUSTRY GROUPS
- OPEN-SOURCE EDA

MAGMA Fusion
Register

EETIMES New IP/SoC Newsletter
▶ CLICK HERE TO REGISTER

August 12, 2003

Silicon Engineering

Coexistence in a multilingual design world

By Grant Martin
EE Times
June 16, 2003 (11:46 a.m. EST)

PRINT THIS STORY | SEND AS EMAIL

We have heard the commotion over SystemC and an evolved Verilog. The IEEE 1364 Verilog committee has a plan to specify next-generation Verilog: 1364-2005. Verilog 2005 will be based on input from vendors (e.g., Cadence

Latest Headlines

Silicon Engineering
- Facing electromigration issues
- TSMC's Hu sees earlier entry for FinFET
- Mask shop

SIGN UP FOR NEWSLETTERS
CLICK HERE

Advertisement
MAGMA Fusion
Register Today!

Semiconductor manufacturers

Trouble enhancing your yield and reducing time-to-market?

EETIMES' new DESIGN FOR YIELD Newsletter

cadence

SystemC is for *System-Level Design*

(Hugo De Man's "7th. Heaven of Software")

System and SW Modeling: UML, SDL, etc.

System-Level Design and System Level Integration Infrastructure: SystemC

Mere Implementation!! VHDL, Verilog, SystemVerilog, Verilog-2005

(Hugo De Man's "Deep Submicron Hell of Physics")

Important SystemC Milestones

- 1.0
  - Layered on C++
  - Fixed-point modelling
  - RTL (as refinement target …….but weak on system level modelling)
- 1.2
  - Master-Slave library – methodology specific libraries layered on top of kernel
- 2.0
  - True system level abstractions: channels, interfaces, events
  - Basis for multiple Models of Computation and abstraction levels
- SCV 1.0
  - Layered verification library on top of SystemC
- **Current** and future: **2.01 and LRM for IEEE standards transfer**
  - Transaction-level modelling standard for IP interoperability (2.1?)
  - Synthesisable subset work
  - SystemC 3.0 – RTOS, SW tasking and scheduling modelling constructs
  - Continued experiments with Analogue/Mixed-Signal extensions

Role of SystemC

- System-level design at all levels of abstraction:
  - Functional modelling using various models of computation
  - Transaction-level modelling of hardware platforms
  - Refinement of design from untimed functional models through to RTL implementation
  - Creation of HW-SW platform models for use by system designers and embedded software developers
  - System-level testbenches (using the new SCV library) which can be propagated through implementation
- But clearly not:
  - A substitute for HDL's, nor
  - A software modelling language

HDL Evolution

- Despite the appearance of language wars, there is a clear path for HDL evolution:
  - Via IEEE 1364 for Verilog
  - Via IEEE 1076 for VHDL
- The IEEE process provides a mechanism for the reconciliation of various interests:
  - Commercial proprietary interests
  - Industry bodies in language based design such as Accellera
  - Academia
  - users

Verilog Evolution

- IEEE 1364 Verilog working group has begun a process for evolving Verilog beyond Verilog-2001 (“Verilog-2005”…)
- Will accept donations
- Is conducting User Surveys as to needs and priorities
- Opportunity for Accellera to donate SystemVerilog 3.1 or further version as input
- Some companies have already donated technology as inputs
  - E.g. Cadence – June 2003: testbench, IP encryption
- Opportunity for others to contribute
- IEEE 1364 is also a forum for users to influence the standardisation process – as much or perhaps more than tool vendors
- Nothing proposed in this area moves Verilog up to true system-level design abstractions

VHDL Evolution

- IEEE 1076 has begun a process to look at VHDL evolution beyond VHDL-1998: called VHDL-200x
  - Web site URL: http://www.eda.org/vhdl-200x/
- Led by Steve Bailey, now at Mentor Graphics
- Areas of interest in extending:
  - Assertions, testbenches and verification
  - Datatypes and abstraction
  - Environment (simulation control) and interoperability
  - Modeling and productivity
  - Performance
- Hope to have finished by late 2004/early 2005
- Wants strong user input
  - Users will have the strongest impact on the evolution and longevity of VHDL

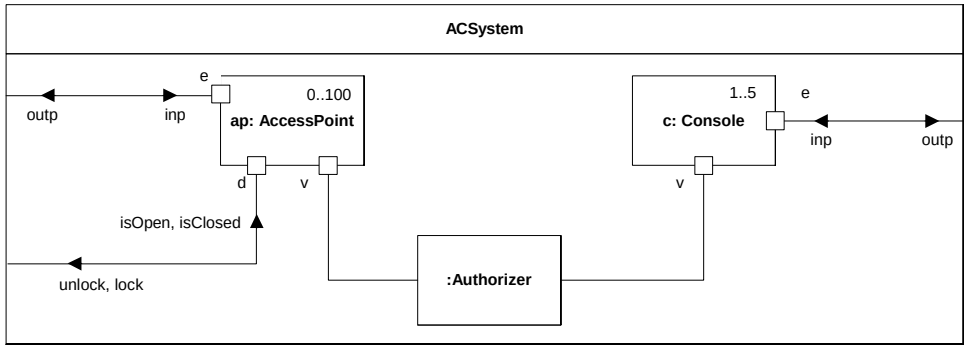
HDL Evolution

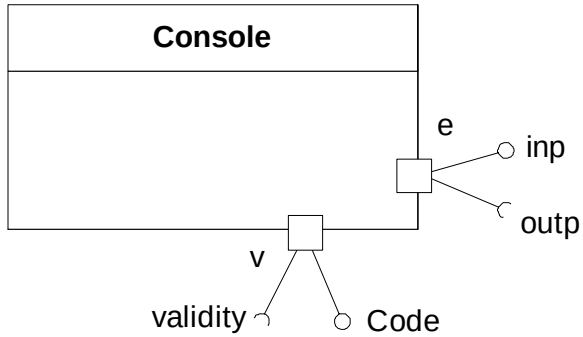
- Despite the proprietary and commercial interests, ultimately users decide on what happens with languages and tools
- Usage attracts tool vendors
- Revenues are more important than ideologies
- Legacy and conversion costs as well as differences of capabilities are significant factors in language longevity
- ……..it’s a multilingual world
- ……and likely to stay that way

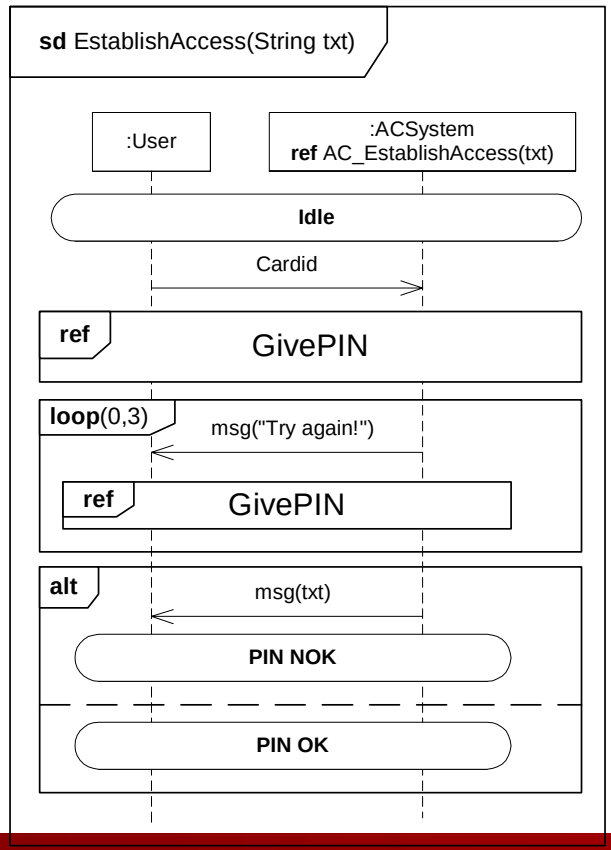
Software Modelling Evolution

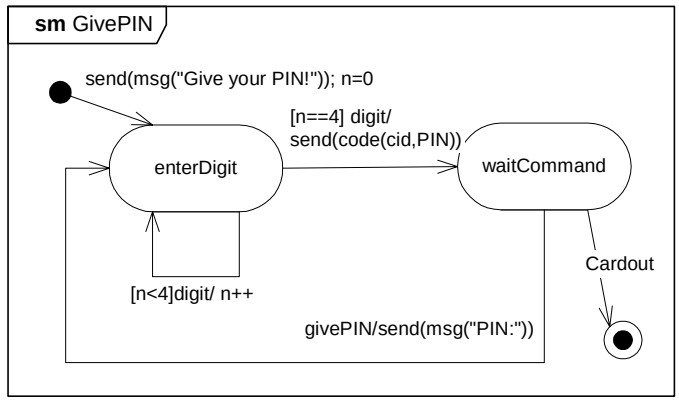
- UML moving towards finalisation in 2003 of UML 2.0
- Adds significant system modelling capabilities to UML's object-oriented software modelling capabilities:
  - Structured classes or components (what in ROOM were called capsules)
  - Ports as interfaces and service specifications
  - Hierarchical sequence diagrams with loops and alternatives
  - Hierarchical state machines
- In addition, UML profile for Scheduling, Performance and Time (SPT), and Action semantics definition, support more complete modelling of real-time aspects of systems, and more optimised code generation
- Finally, the Object Management Group (OMG)'s embrace of MDA – Model Driven Architecture – fits more closely with the EDA/HW design world notions of design methodology and flow

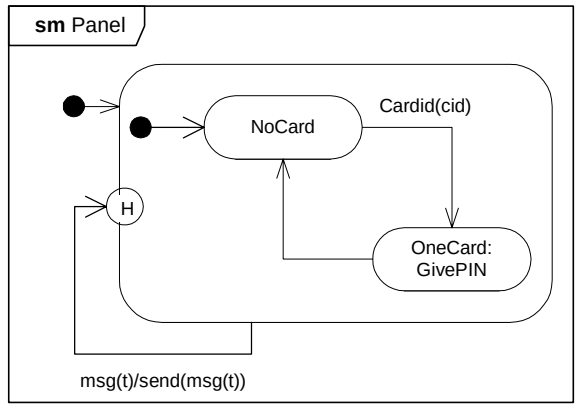
UML 2.0


ACSystem
e
outp
inp
0..100
ap: AccessPoint
d
v
1..5
c: Console
e
inp
outp
v
isOpen, isClosed
unlock, lock
:Authorizer


Console
e
inp
outp
v
validity
Code

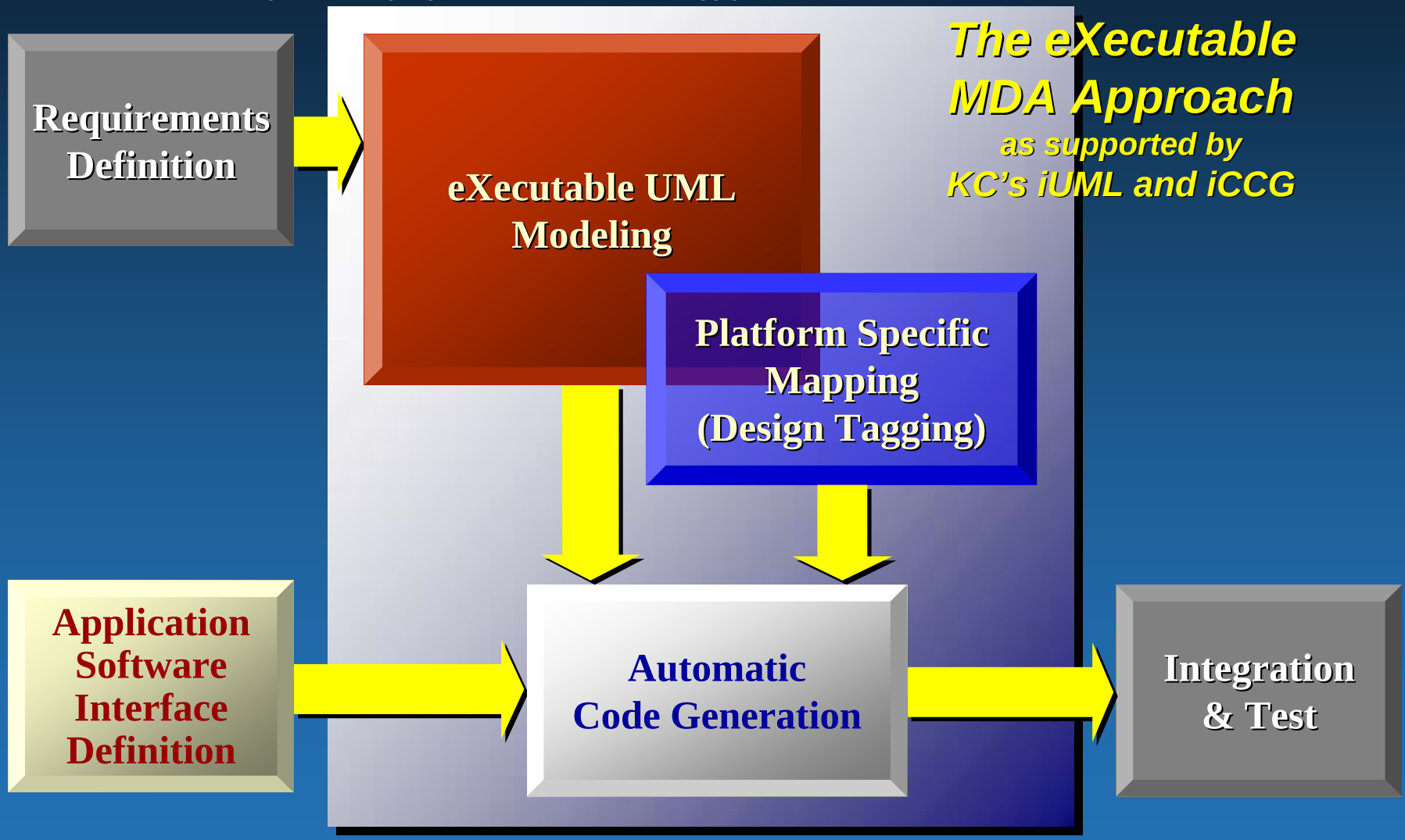

sd EstablishAccess(String txt)
:User
:ACSystem
ref AC_EstablishAccess(txt)
Idle
Cardid
ref
GivePIN
loop(0,3)
msg("Try again!")
ref
GivePIN
alt
msg(txt)
PIN NOK
PIN OK


sm GivePIN
send(msg("Give your PIN!")); n=0
[n==4] digit/
send(code(cid,PIN))
enterDigit
waitCommand
Cardout
[n<4]digit/ n++
givePIN/send(msg("PIN:"))

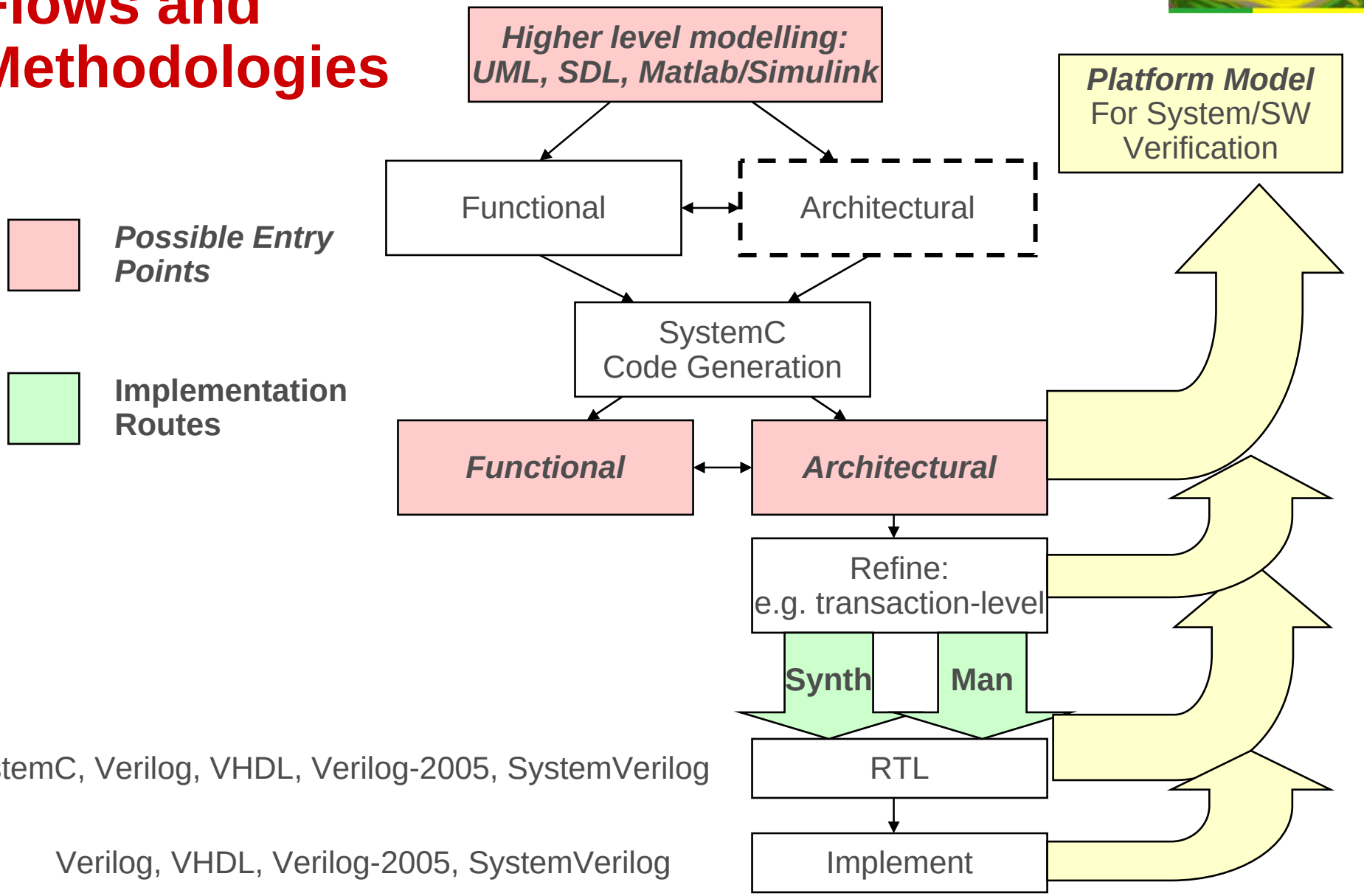

sm Panel
NoCard
Cardid(cid)
H
OneCard:
GivePIN
msg(t)/send(msg(t))

eXecutable MDA: Application Software Development

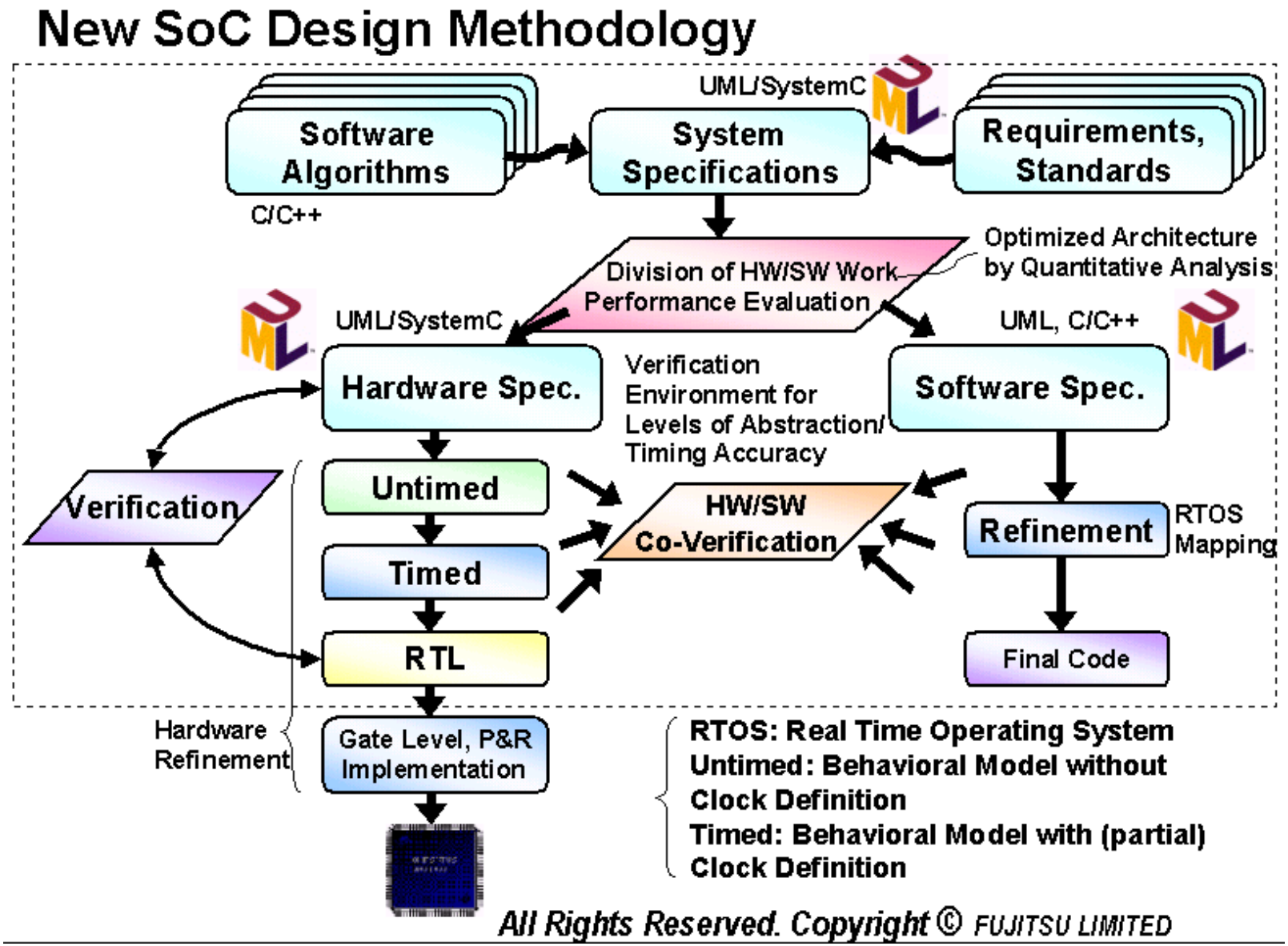
(Source: L. Clark, T. Ruthruff, Bary Hogan and Allan Kennedy, "F-16 Modular Mission Computer Application Software: Achieving Cross-Platform Compatibility with Increased Productivity and Quality Using the OMG's Model-Driven Architecture", OMG WebSite, URL http://www.omg.org/mda/mda_files/New_MDA.ppt)



Flows and Methodologies



Fujitsu – UML, SystemC, flow to RTL implementation



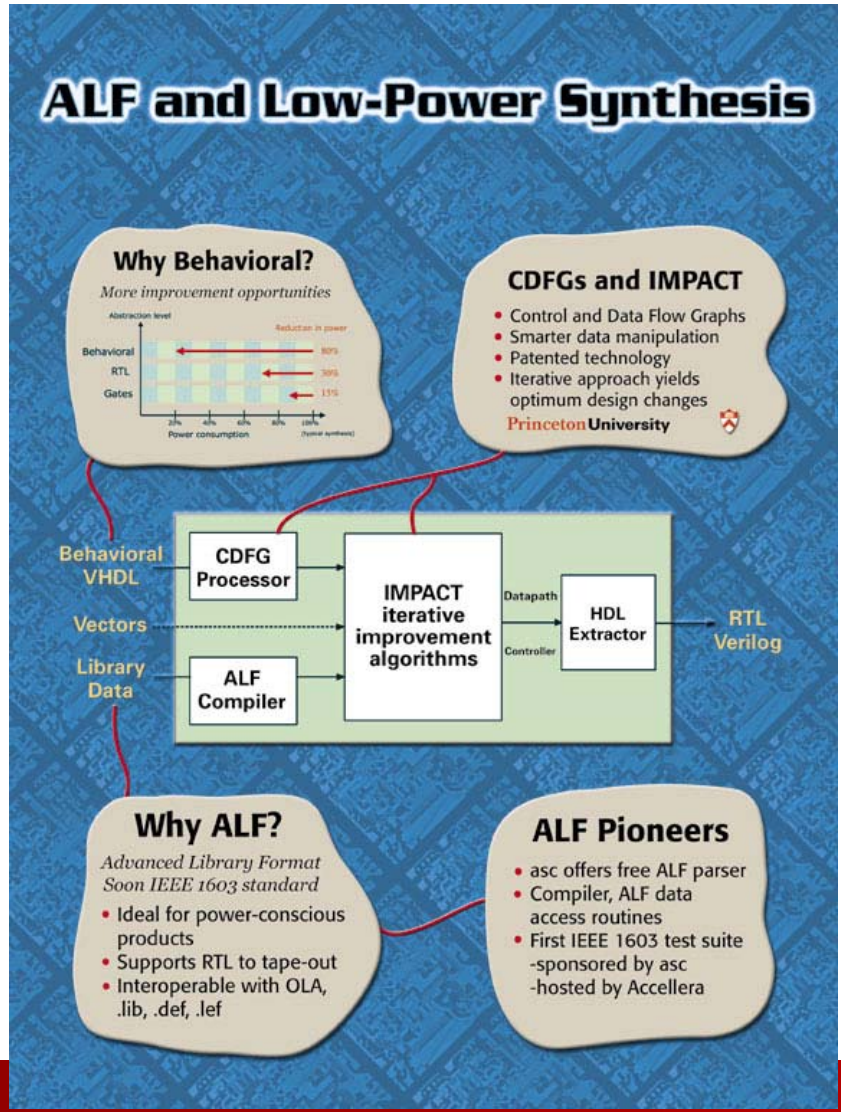
Language-Based Design Research Opportunities

- Paths to implementation:
  - Next generation behavioural synthesis?
  - Co-processor synthesis?
- Models of Computation for systems
  - How many are necessary? How many are enough?
  - How do you link them?
- Transaction-Level Abstractions for Platform modelling
  - Which ones, and how many?
  - Exporting platform models to embedded SW developers
- Verification
  - Assertion-based, Transaction-based, Coverage-based
  - Combined Dynamic (simulation) and Static (formal) verification

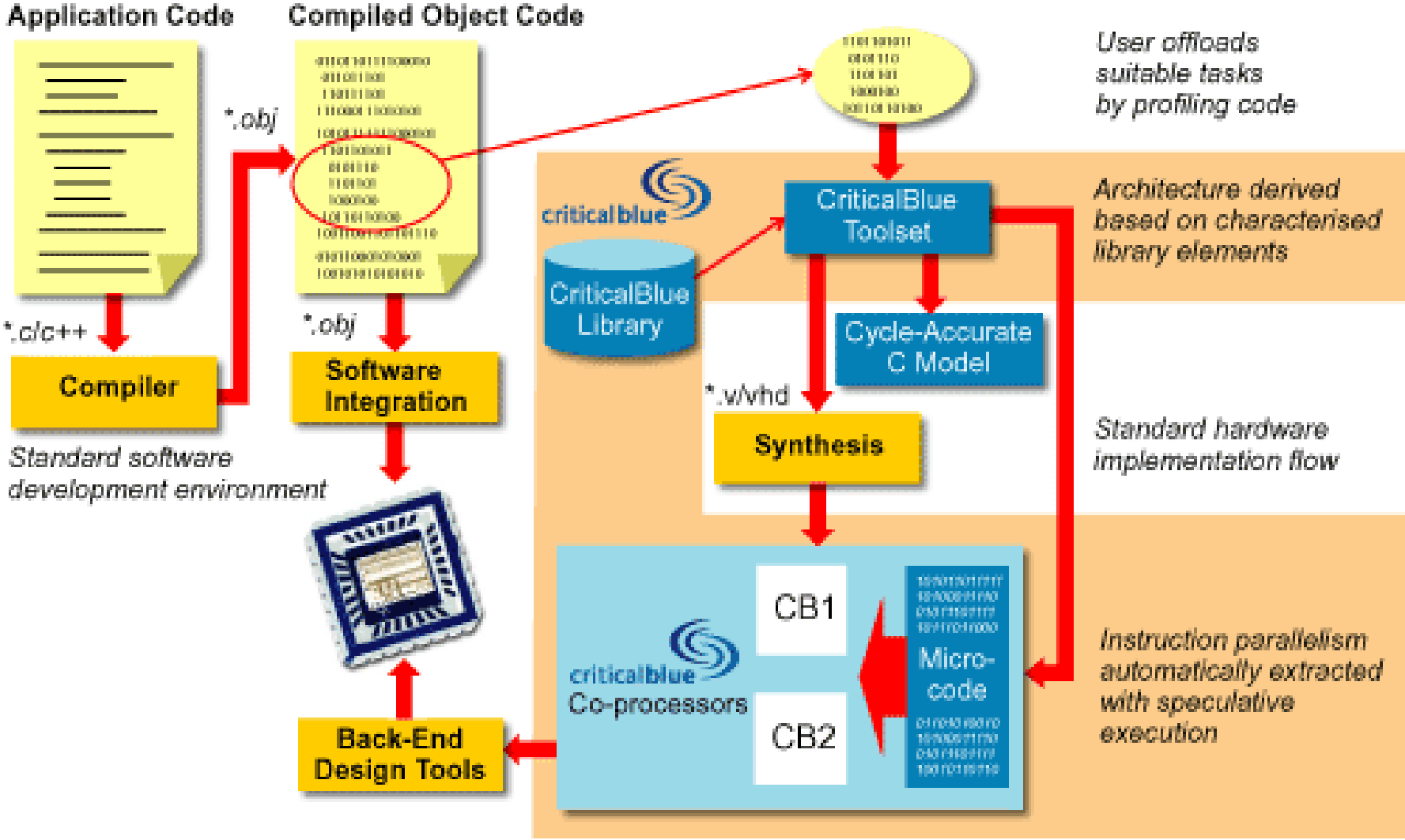
Next-Generation Behavioural Synthesis

- Not your grandfather's or grandmother's behavioural synthesis
  - E.g. Cadence Visual Architect, Synopsys Behavioural Compiler
  - Difficulty of use, quality of results, wrong target architecture, not much above RTL synthesis
- A new generation emerges, with different use models
  - E.g. low power behavioural analysis and synthesis
    - ChipVision ORINOCO, Alternative System Concepts (ASC) "Pacific"
  - E.g. co-processor synthesis
    - Critical Blue
  - E.g. new platform targets
    - Reconfigurable logic – new reconfigurable platforms
- Are all the answers here? Not by a long shot…….

Alternative System Concepts “Pacific” Low-Power Behavioural Synthesis



Co-processor Synthesis Example - CriticalBlue



cadence

(Source: Concurrent Component Patterns, Models of Computation, and Types Edward Lee and Yuhong Xiong, Fourth Annual Workshop on New Directions in Software Technology (NDIST'01), St. John, US Virgin Islands, December 2001).

Example Domains

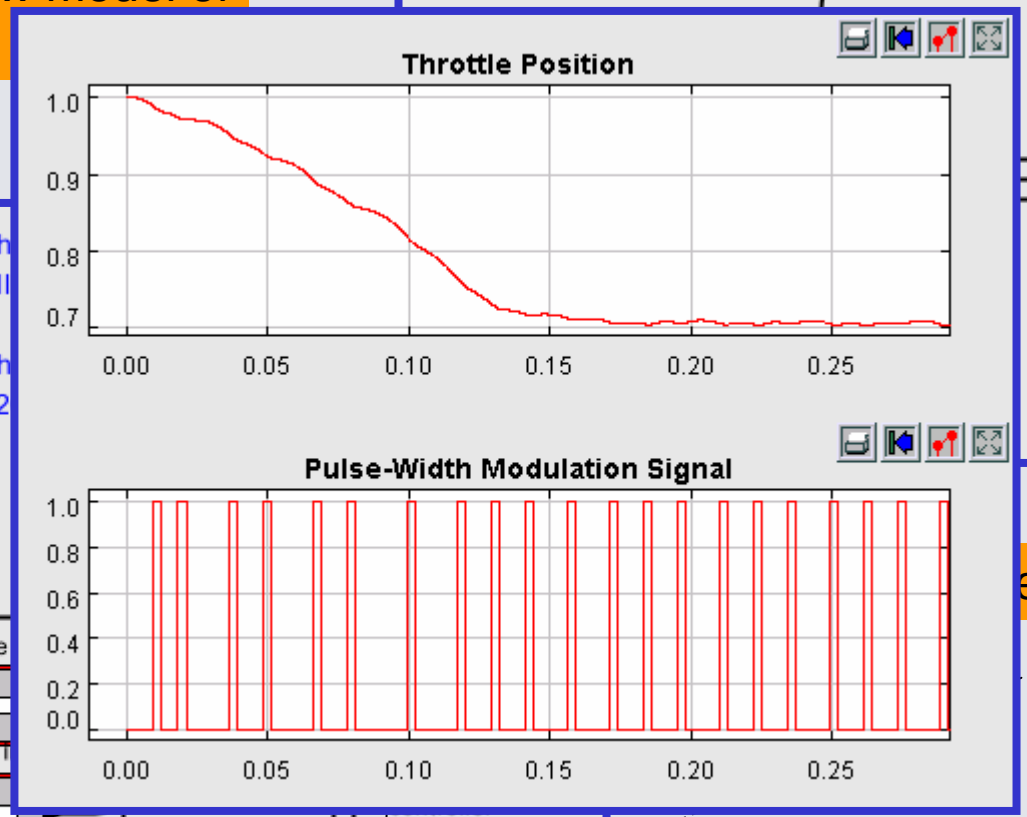
- **Communicating Sequential Processes (CSP):** rendezvous-style communication
- **Process Networks (PN):** asynchronous communication, determinism
- **Synchronous Data Flow (SDF):** stream-based communication, statically scheduled
- **Discrete Event (DE):** event-based communication
- **Synchronous/Reactive (SR):** synchronous, fixed point semantics
- **Time Driven (Giotto):** synchronous, time-driven multitasking
- **Timed Multitasking (TM):** priority-driven multitasking, deterministic communication

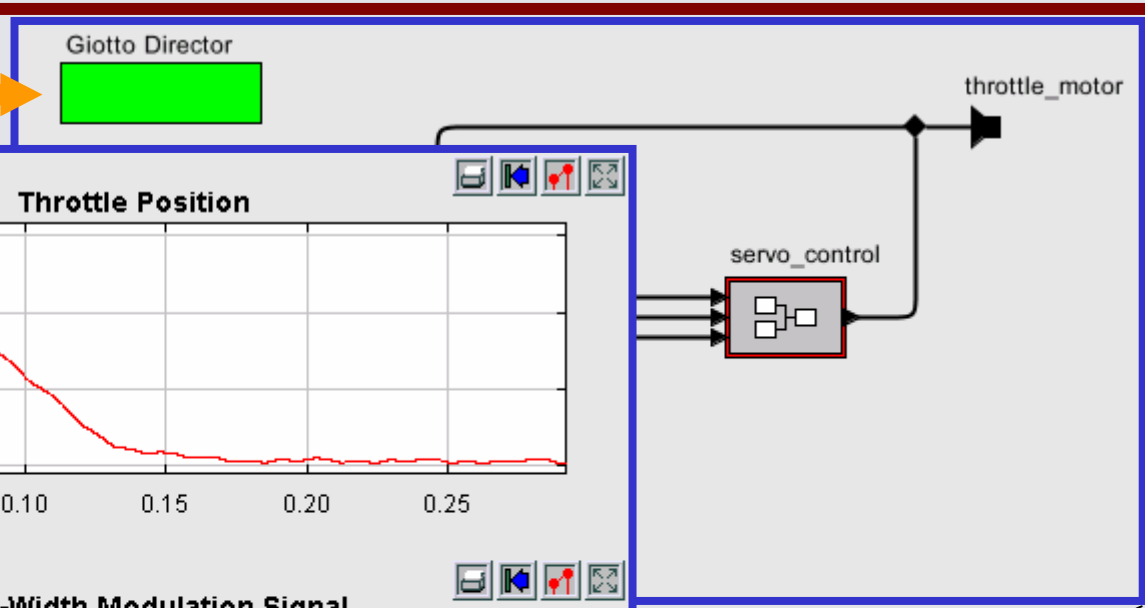
Models of Computation

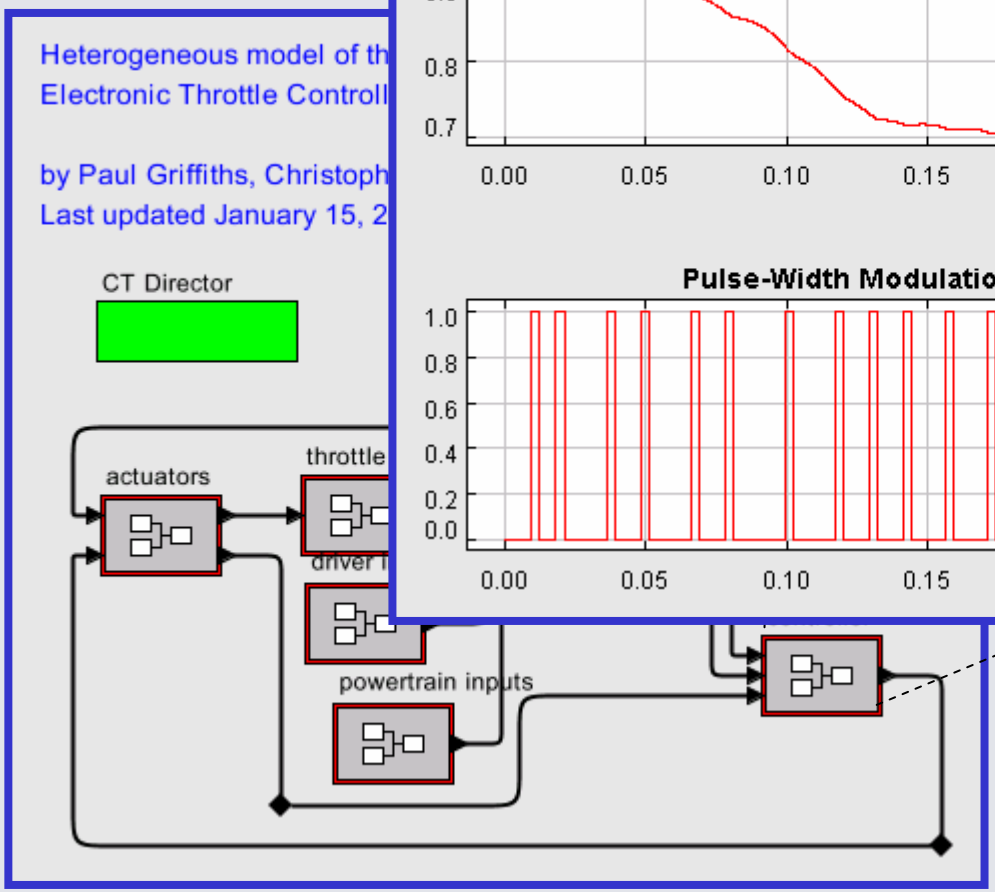
- How many are enough?
  - Another alphabet soup
  - But we can certainly argue for:
    - Dataflow or process network
    - Discrete Event
    - Synchronous/Reactive
  - And for some systems, Continuous time (hybrid systems)

Heterogeneous Models: Periodic/Time-Driven Control Inside Continuous Time

Giotto director indicates a new model of computation.





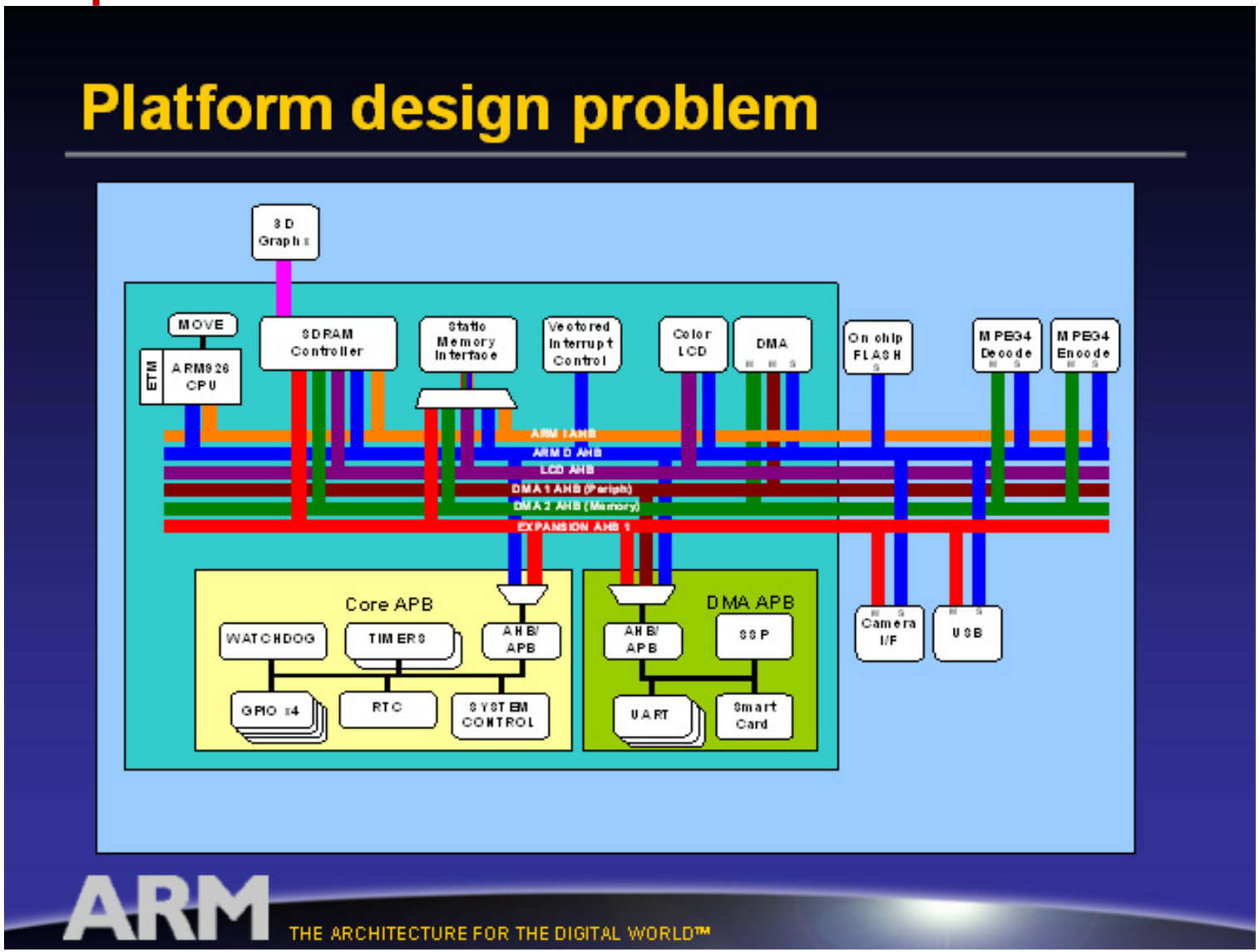


ent.

Domains can be nested and mixed.

Source: "Model-Based Design in the Ptolemy Project" Edward A. Lee, July 31, 2003, A Chess project presented at Boeing, Seattle.

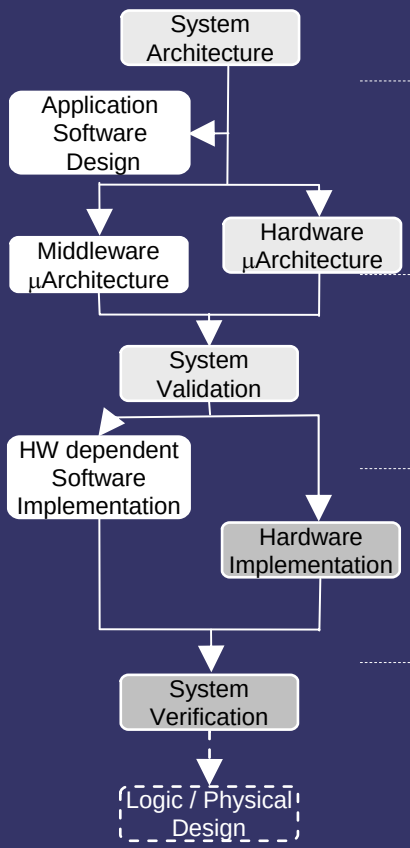
Transaction-Level Models for SoC Design Platforms: example



Source: Jon Connell, ARM: DAC 2002 Open System C Meeting: "Platform Modelling for System Design Using SystemC"

Transaction-Level Abstractions – one proposal

(Source: "Transaction Level Modeling: Overview and Requirements for SystemC Methodology" and "Introduction to TLM" by Mark Burton (ARM), Frank Ghenassia (STMicroelectronics and Stuart Swan (Cadence), May 13, 2003; and "ARM System-Level Modelling" by Jon Connell, June 25, 2003).

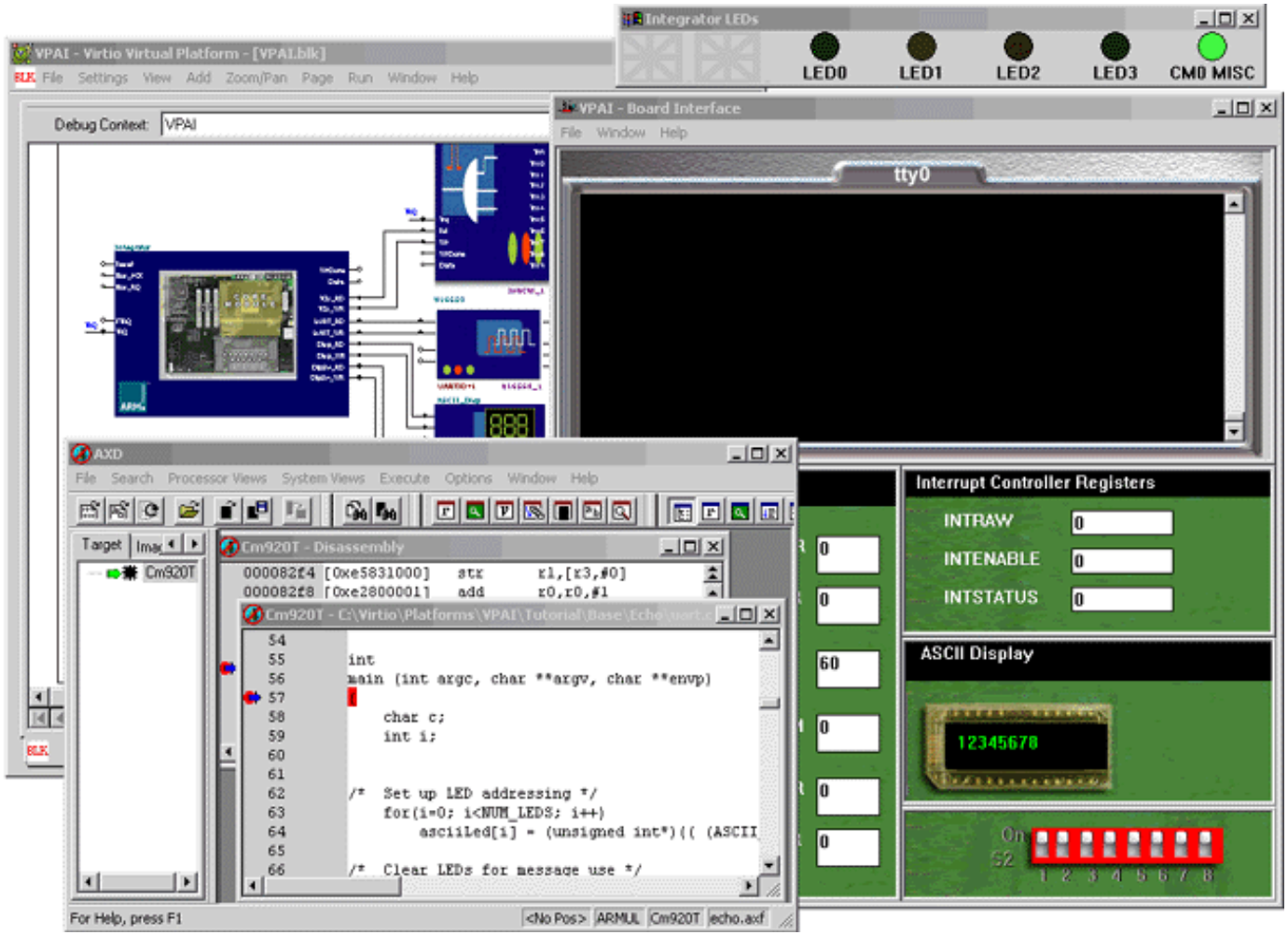


UML – Transaction Level Modeling – HDL

Algorithmic Level (AL) Foundation: Function	Function-calls Functional
Programmer's View (PV) Foundation: Memory Map	Bus generic Architectural
Programmer's View + Timing (PVT) Foundation: Timed Protocol	Bus architecture Timing approx.
Cycle Level (CC) Foundation: Clock Edge	Word transfers Cycle-accurate
RT Level (RT) Foundation: Implementation	Signal/Bit Cycle-accurate

ARM cadence

Example of Virtual Platform Model: Virtio VPAI Arm-based model



Source: http://www.virtio.com/support/platforms/homePage/1,2457,559,00.html

Verification

- Transaction-based verification
  - Beyond the abstraction level decisions – where do the verification stimuli, monitors and checkers come from?
    - From assertions? - Expressed in what form? How can we cover all 'interesting' properties of a protocol?
    - From other notations? Output of system-level simulations?
    - Non-functional properties such as performance?
- Assertion-based verification
  - How do we efficiently generate assertions from specifications?
  - Can a research language such as Rosetta be useful in this context?
  - How do we combine static and dynamic verification methods?
- Coverage
  - The old question – "how much simulation (verification) is enough?"
  - Are we adequately exploiting the years of experience in verifying large systems in building our verification methodologies and tools – and decision making criteria?
  - Can Verification be put on a more sound theoretical footing?

Conclusions

- The role of SystemC is becoming well-defined
- It makes no pretence to cover the full design flow from specification to full physical implementation
- But it covers, reasonably, a set of well-defined use models and requirements in system modelling, refinement, model generation and links to implementation
- We should think of it in the context of a multi-language, multi-disciplinary flow that covers the whole specify-design-implement space
- Notwithstanding the progress that has been made ***– there are many interesting design problems yet to solve, and many interesting research contributions yet to be made.***