Computer Systems, Inc.
MERCURY

UML 2.0 Redux for HPEC

Dr. Jeffrey E. Smith
Mercury Computer Systems, Inc.

Manfred Koethe
88solutions Corp.

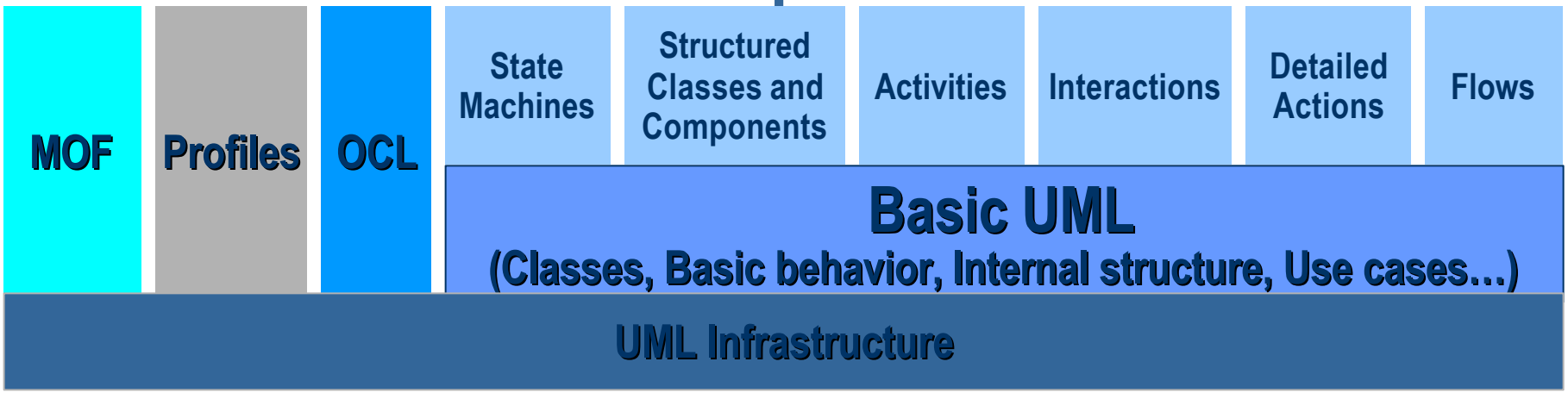
High Performance Embedded Computing (HPEC) Conference
September 25, 2003

The Ultimate Performance Machine

Computer Systems, Inc.
MERCURY
The Ultimate Performance Machine

UML Overview

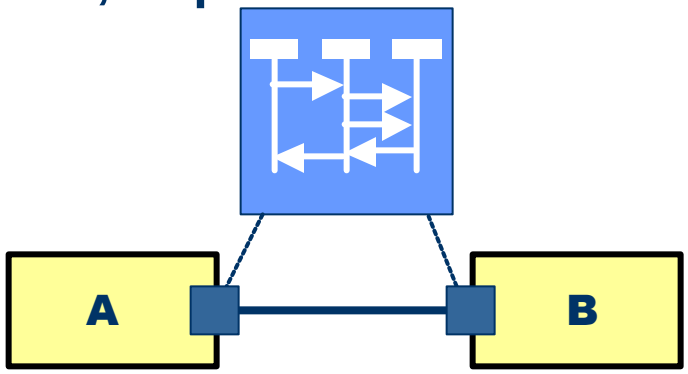
- Visual modeling language
  - Providing controllable levels of abstraction
  - Definition of static and dynamic model features
  - Communicating/predicting application design characteristics in domain-terms
  - Supporting automation of development process
  - Derived from OMT, Use Case and Booch (Component) methodologies
- Dominant modeling language for software architecture “blueprints”



Credit Bran Selic, “An Overview of Model-Driven Development and UML 2.0”

Richer Language Features

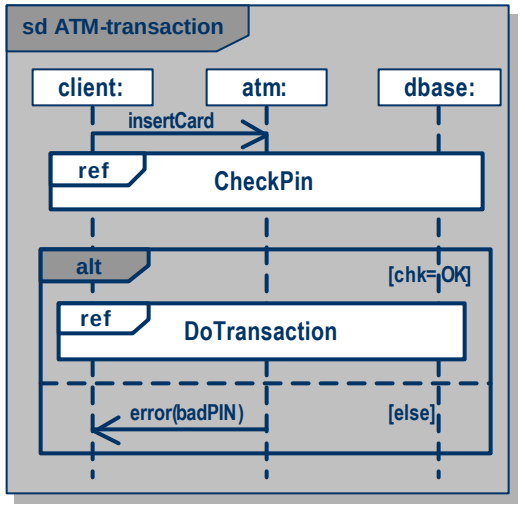
- Architectural modeling: composition and stronger encapsulation via structured classes
  - Components model internal structure, required interfaces and support deployment
  - Ports connect class interfaces to environment
  - Protocol definable on connection
  - Data or control flow



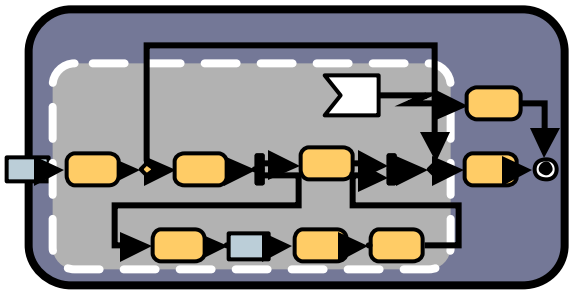
- Deeper profile extension mechanism (than stereotypes, tags and constraints) with UML meta-model extensions
  - Platform-specific terminology, UML symbols and semantics
  - Full integration with MOF providing tool integration

Enhanced Behavioral Modeling Capture Problems at Model Level

- Extended sequence diagrams permit more detailed complex interactions
  - Supports sub-diagrams
  - Decomposition of SDL, MSC and LSC messages
  - Control structures: loop, parallel execution, alternative execution, protected regions, ...

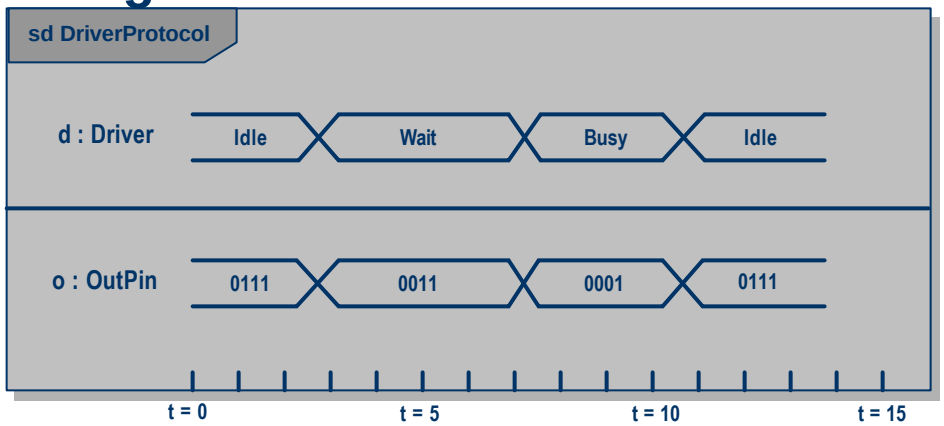


- Activities permit more flexible parallelism, I/O options and data/control flow modeling
  - Petri Net model to derive concurrency
  - Unstructured activities possible
  - Pre/post conditions
  - HPEC features described in HPEC 2001 e.g. interruptible regions and execution ordering



Timing and State Modeling

- Precise modeling of timing via timing sub-diagrams
  - Previous profile for modeling schedulability, performance and time embedded in UML 2.0
  - Enables next level of integrating hardware modeling to platform design



- Statecharts now have sub-statecharts and inheritance

UML 2.0 Provides HPEC "Potential" for Software Design Automation

- Action semantics integrate activities with related low-level actions
- There are many methods of UML-based code generation
  - State translation (I-Logix, Rose RT)
  - Formal translation (NU research, Telelogix)
  - Direct template translation (Pathfinder)
  - MDA-based model execution (Pathfinder, Component-X, 88solutions)
  - Generate/discover components (PCA)
  - Low-level data/state flow import (MathWorks)
  - Informal indirect translation to non-mainstream tools and PGO (HPEC 2000)
  - Model Integrated Computing (MIC, MOBIES)