

# UML 2.0 Redux for HPEC

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

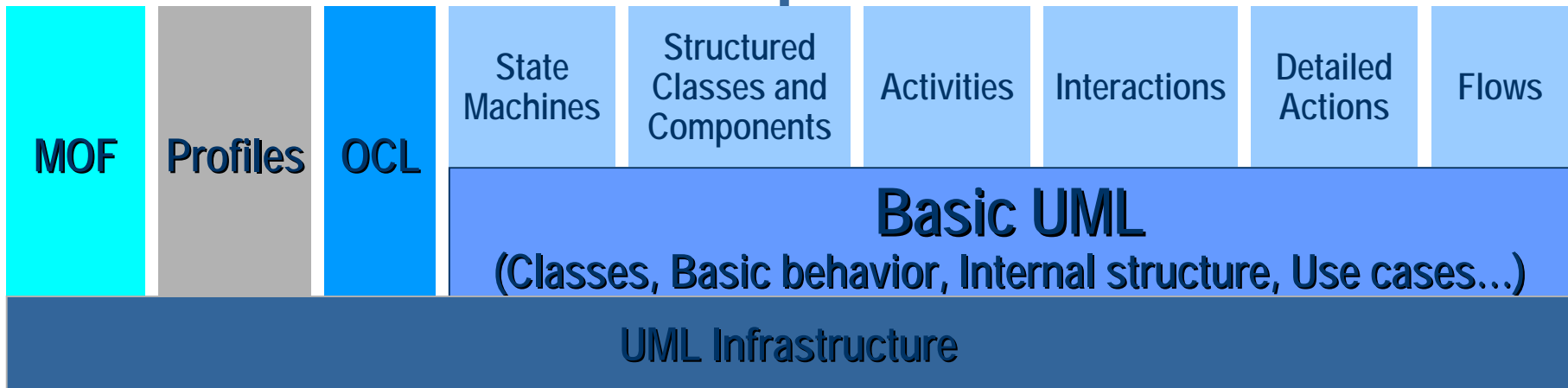
Manfred Koethe  
88solutions Corp.

**HPEC – September 25, 2003**

*The Ultimate Performance Machine*

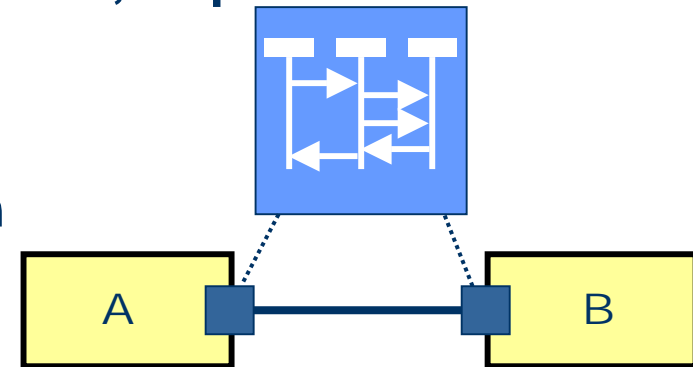
# UML Overview

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



# Richer Language Features

- **Architectural modeling: Composition and stronger encapsulation via Structured Classes**
  - z **Components model internal structure, required interfaces and support deployment**
  - z **Ports connect class interfaces to environment**
  - z **Protocol definable on connection**
  - z **Data or control flow**

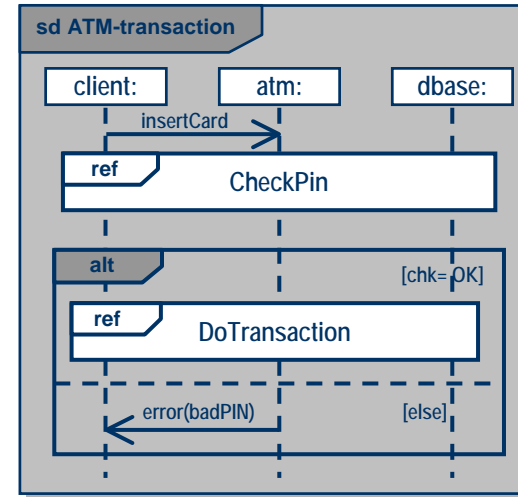


- **Deeper profile extension mechanism (than stereotypes, tags and constraints) with UML meta-model extensions**
  - z **Platform specific terminology, UML symbols and semantics**
  - z **Full integration with MOF providing tool integration**

# Enhanced Behavioral Modeling Capture Problems at Model Level

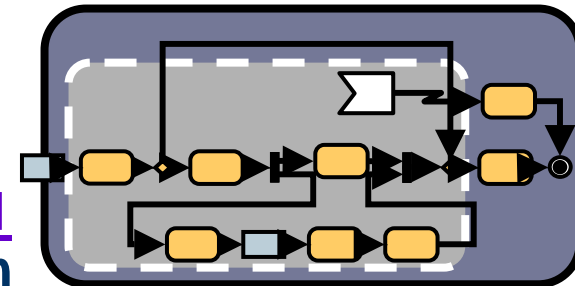
- **Extended Sequence Diagrams permit more detailed complex interactions**

- z Supports sub-diagrams
- z Decomposition of SDL, MSC and LSC messages
- z Control structures: loop, parallel execution, alternative execution, protected regions, ...



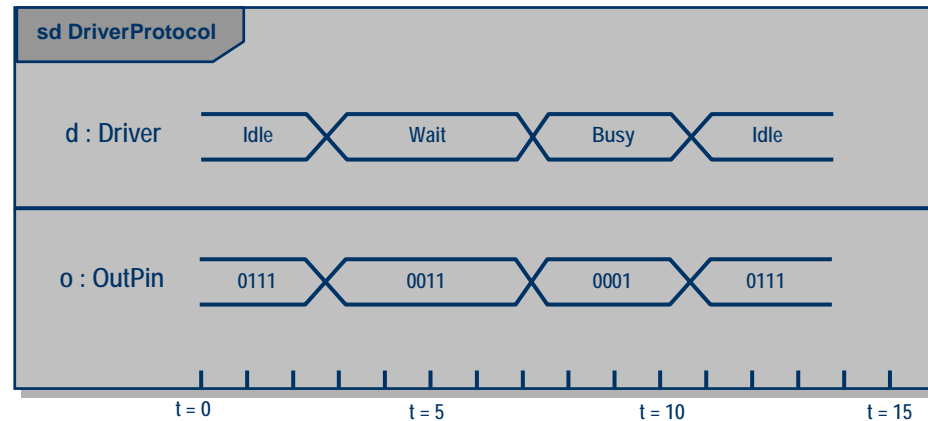
- **Activities permit more flexible parallelism, I/O options and data/control flow modeling**

- z Petri Net model to derive concurrency
- z Unstructured activities possible
- z Pre / post conditions
- z 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**
  - z **Previous Profile for modeling schedulability, performance and time embedded in UML 2.0**
  - z **Enables next level of integrating HW 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**
  - z **State translation (I-Logix, Rose RT)**
  - z **Formal translation (NU research, Telelogix)**
  - z **Direct template translation (Pathfinder)**
  - z **MDA-based model execution (Pathfinder, Component-X, 88solutions)**
  - z **Generate/discover components (PCA)**
  - z **Low-level data/state flow import (MathWorks)**
  - z **Informal indirect translation to non-mainstream tools and PGO (HPEC 2000)**
  - z **Model Integrated Computing (MIC, MOBIES)**