The Road to UML Virtual Machines

Dirk Riehle

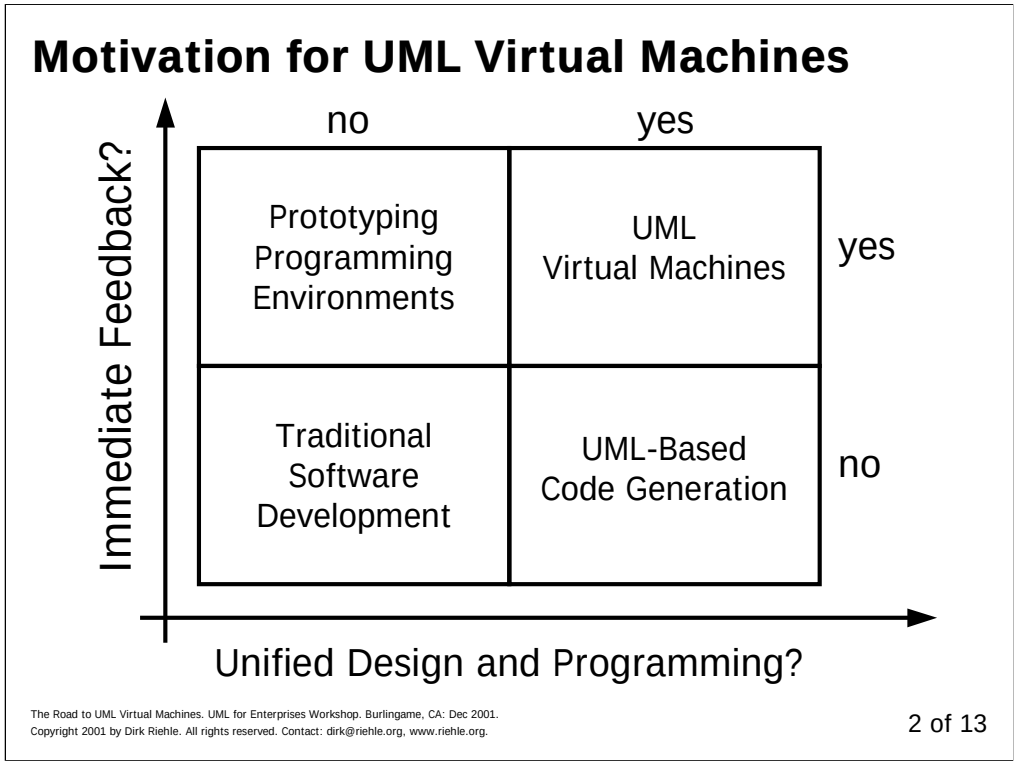
SKYVA International
www.skyva.com

dirk@riehle.org, www.riehle.org

UML for Enterprises Workshop. Burlingame, CA: Dec 2001.
Last updated .

Presented at:

OMG's Second Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise. Sheraton Gateway Hotel, San Francisco International Airport, Burlingame, CA, USA: December 3-6, 2001.



With UML becoming an executable modeling language, there is not much difference between a modeling language and a programming language: UML becomes a programming language.

However, UML will not only provide a programming perspective, it also already provides an analysis and design perspective. All of these perspectives have the same underlying object model.

This unifies design with programming. Developers will work on one common model, on different levels of abstraction. No impedance mismatch anymore between analysis, design, and programming.

Today's tools generate code, and round-trips can take a long time. UML virtual machines interpret a model and provide users with immediate feedback about the functioning of the modeled system.

Code-generation vs. interpretation is a red herring, though: what counts is immediacy of feedback, whatever way you achieve it.

Definition: UML Virtual Machine

- UML virtual machine
  - Is an abstract computing machine (like any VM)
  - Provides an instruction set and a memory model for representing objects
- Instruction set of a UML virtual machine
  - Behavior modeled using UML itself, complemented by Java
  - Elements are persistently represented using XMI
- Memory model of the virtual machine
  - Memory management facilities of implementation language (Java)
  - Dedicated packages, garbage collection

The Road to UML Virtual Machines. UML for Enterprises Workshop. Burlingame, CA: Dec 2001.

Just like every other virtual machine …

High-level Requirements for UML VM

- Model representation
  - Represents models on all levels
- Model execution
  - Interprets models for execution
- Causal connection between models
  - Changes have immediate and defined effects

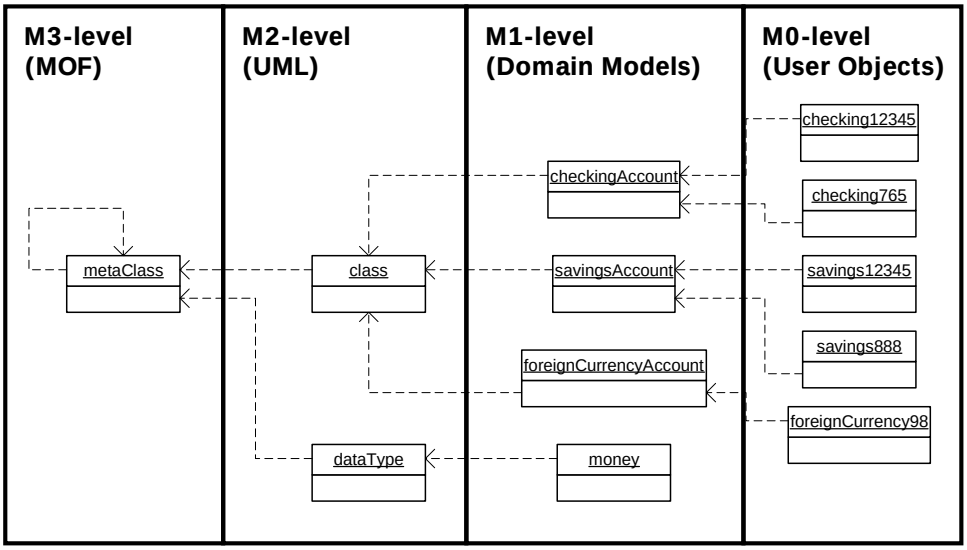
The Road to UML Virtual Machines. UML for Enterprises Workshop. Burlingame, CA: Dec 2001.

We will first discuss some high-level requirements for UML virtual machines, followed by a short architecture review.

Then we will discuss how SKYVA implemented its UML virtual machine.

Finally, we abstract from SKYVA's experience and review alternative concepts and what is necessary to standardize UML VMs.

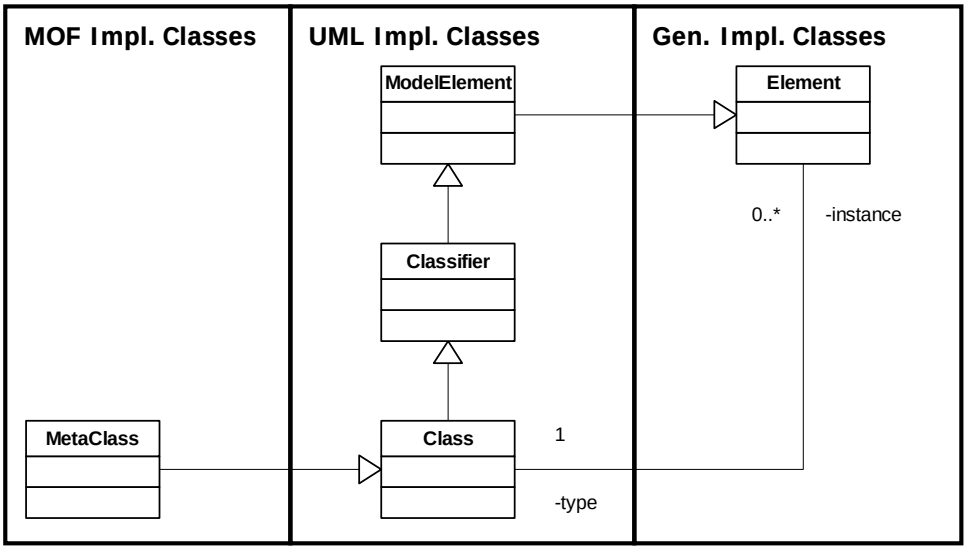
UML 4-Layer Architecture (Logical Arch.)



5 of 13

We need to separate the logical architecture, consisting solely of objects, from the physical architecture, consisting of Java classes that implement the logical architecture.

Java Implementation (Physical Arch.)



For a more in-depth discussion, please see:

Dirk Riehle, Steven Fraleigh, Dirk Bucka-Lassen, and Nosa Omorogbe. "The Architecture of a UML Virtual Machine." In *Proceedings of the 2001 Conference on Object-Oriented Programming Systems, Languages, and Applications* (OOPSLA '01). ACM Press, 2001. Page 327-341.

Element Functionality

- UML-derived functionality
  - Generic attribute value and link access
  - Type checking that can be switched on or off
  - Handling of association objects
  - ...
- VM-needed functionality
  - Shallow/deep cloning, equality
  - Backpointers, garbage collection
  - Serialization, inspection
  - Team collaboration state model, versioning
  - External resource management
  - ...

Element is the root (implementation-)class. It must be available on the implementation level. It may or may not be represented on the logical level.

We favor a single-rooted class hierarchy approach. Therefore, we made Element a logical class as well. It is the superclass of ModelElement.

Behavior Modeling and Execution

- UML state machines
  - State machines are most precisely defined
  - Best understood how to implement
- OCL enhancements
  - Ensure constraints and business rules
  - Lightweight “programming” tasks
- Hand-programming (Java)
  - Customization through policies
  - Requires well-designed extension architecture

The Road to UML Virtual Machines. UML for Enterprises Workshop. Burlingame, CA: Dec 2001.

...

SKYVA Runtime Environment

- Predefined models with predefined implementations
  - Technical as well as business models (some proprietary, some not)
  - Includes configuration models
- Is paramount for efficient execution
  - Transactions, persistence, security!
  - Today: San Francisco; Soon: IBM Websphere

The Road to UML Virtual Machines. UML for Enterprises Workshop. Burlingame, CA: Dec 2001.

...

Bare-Bones UML Virtual Machines?

- Bare-bones UML Virtual Machine
  - Knows only UML as basis for domain models
  - Can only execute behavior of standard types of classes
  - Does not allow for domain-specific UML extensions
- Advantages
  - Easier to implement, better performance
- Disadvantages
  - Forbids use of profiles in modeling

In our opinion, while technically an option, bare-bones UML virtual machines do not make sense. UML extensions like profiles are mandatory and require the existence of a first-class UML model.

Missing Pieces 1/2

- Element class
  - Element/Class collaboration specification
- Behavioral specification of UML itself
  - Time-honored tradition: eating your own dog food
- Operational model of VM
  - Starting point (main)
  - Life-cycle model (incl. garbage collection)
  - Concurrency model
  - Model evolution support
  - Native call interface

...

Missing Pieces 2 / 2

- Standardized technical models (libraries)
  - GUI library
  - Simple persistence

...

Summary

- Expect UML Virtual Machines
- Expect UML IDEs/systems like Smalltalk

- See www.riehle.org/papers/2001/oopsla-2001.html
- Also see: yahoogroups.com/group/uml-virtual-machines
- If you have questions, feel free to email: dirk@riehle.org

- If you (intend to) develop UML VMs...

The Road to UML Virtual Machines. UML for Enterprises Workshop. Burlingame, CA: Dec 2001.

UML virtual machines are in our future.

They may be closer than they appear.