

The Architecture of a UML Virtual Machine

Dirk Riehle

SKYVA International
www.skyva.com

dirk@riehle.org, www.riehle.org

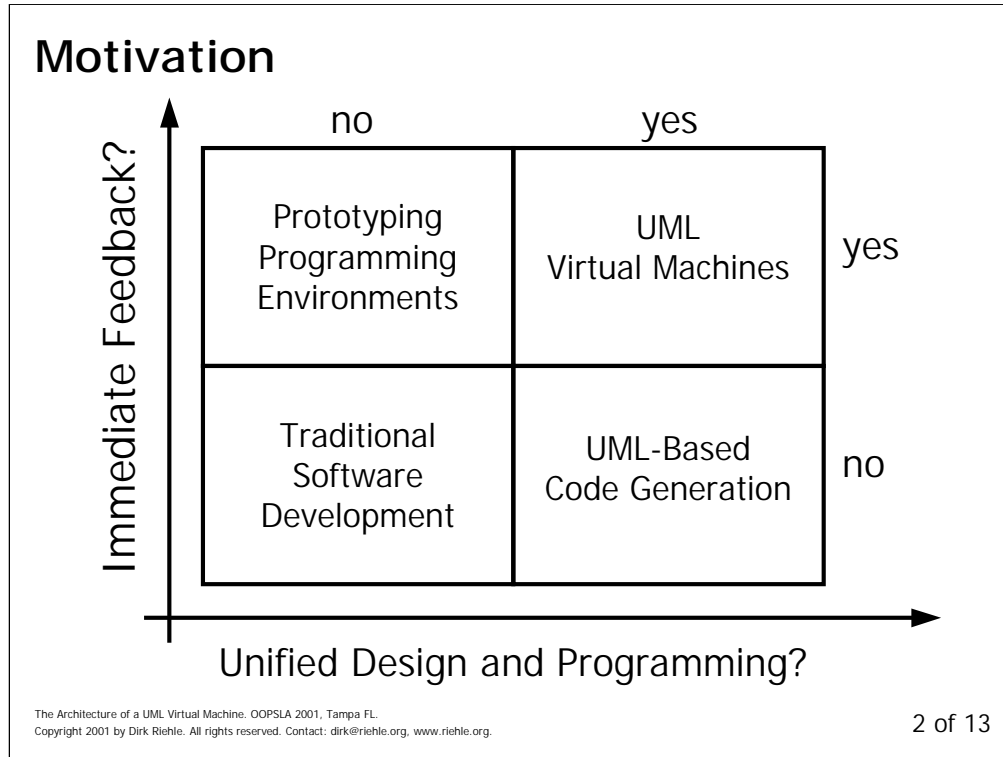
OOPSLA 2001, Tampa, FL.

Last updated .

Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

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.

Presented as a technical paper at OOPSLA 2001 in Tampa, FL, U.S.A.



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 design and programming, no throw-it-over-the-wall attitude.

Today's tools generate code, which 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
 - UML itself. UML provides several behavior modeling capabilities that can be used to describe the behavior of a model (including itself)
 - Models are persistently represented using XMI, the OMG standard for representing UML models using XML
- Memory model of the virtual machine
 - Memory management facilities of implementation language (Java)
 - Dedicated packages, garbage collection

The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

3 of 13

Just like every other virtual machine ...

Requirements: UML Virtual Machine

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

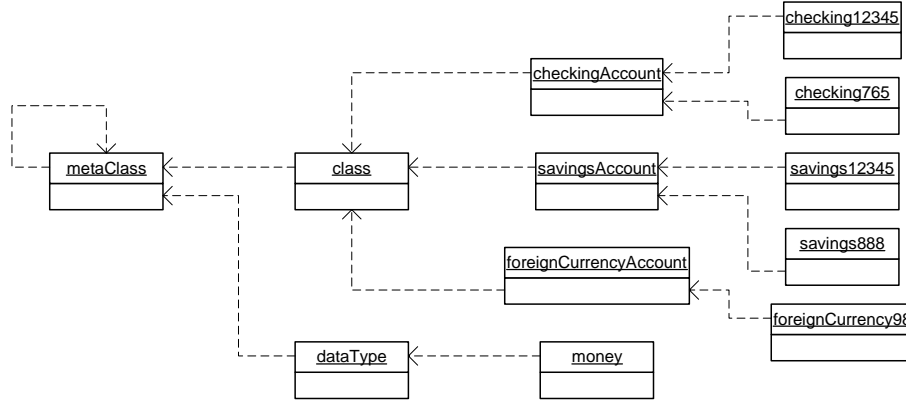
The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

4 of 13

This presentation discusses how to achieve the causal connection requirement, the most fundamental property of model-driven systems, for UML virtual machines.

Single Causally Connected Model

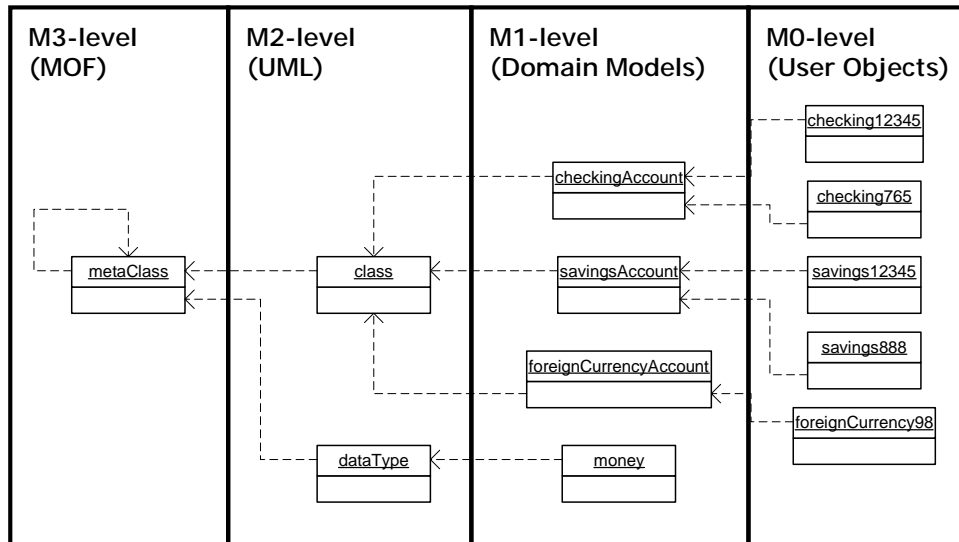
Modeling and Execution Environment



The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

5 of 13

UML 4-Layer Architecture (Logical Arch.)



The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

6 of 13

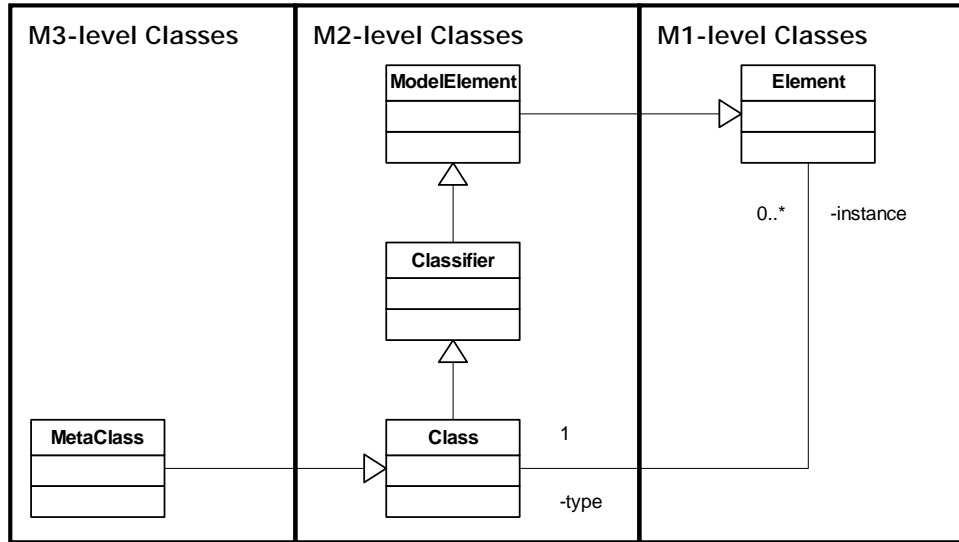
We need to separate the logical architecture, consisting solely of objects, from the physical architecture, consisting of Java classes forming a framework.

Causal Connection

- Definition: Causal Connection
 - A modeling level is causally connected with the next higher modeling level, if the lower level conforms to the higher level and if changes in the higher level lead to according changes in the lower level
- Consequences for UVM
 - Immediate feedback to model changes

This will look familiar to everyone knowledgeable in meta-level architectures.

Java Implementation (Physical Arch.)



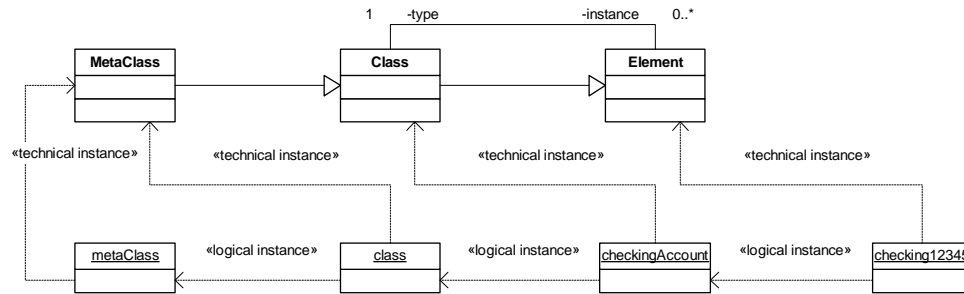
The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

8 of 13

Element Functionality

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

Mapping Logical/Physical Architecture



The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

10 of 13

Ensuring Causal Connection

- Every element has a reference to its class
 - Every access to an element is type-checked by its class
- => Every M0-element is causally connected with its M1-class
- Every class is an element
- => Every Mn-element is causally connected with its Mn+1 class

The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org

11 of 13

By induction, we prove that the presented architecture guarantees the causal connection property.

Every M0-level element, as an instance of Element, has a link to its class. Thus, for every M0-level element, its M1-level class can be found. The class provides all relevant model information for the M0-level element then. Thus, causal connection holds for the M0/M1-level instance/type relationship.

Because every element is an instance of a subclass of Element, it provides the link to its class. This includes Class and every other type of element. Thus, for a given level Mn, every Mn-level element will have its Mn+1-level class.

Therefore, the causal connection property holds.

Open Issues

- Improve behavior modeling capabilities
- Integrate Action Semantics as Java replacement
- Improve execution speed

The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

12 of 13

The presentation did not address behavioral modeling and execution.
However, the paper does, even though not in full detail.

Conclusions

- Expect UML Virtual Machines
- Expect UML IDEs/systems like Smalltalk
- Expect return of the MOP

- The paper covers much more ground than the talk
- If you want to build a company to develop UML VMs...
- If you have questions, feel free to email: dirk@riehle.org
- Also see: yahoogroups.com/group/uml-virtual-machines

The Architecture of a UML Virtual Machine. OOPSLA 2001, Tampa FL.
Copyright 2001 by Dirk Riehle. All rights reserved. Contact: dirk@riehle.org, www.riehle.org.

13 of 13

UML virtual machines are in our future.