

# Model Driven Design and Unified Process

Peter Dolog  
dolog [at] cs [dot] aau [dot] dk  
5.2.47  
Information Systems  
March 3, 2007

# Outline

Model Driven Design

Tutorial on Requirements Eng. and SCRUM reflections (D402a,  
s601c)

Unified Process

# Learning Goal

To apply MDD principles

To reflect on Requirements Engineering and SCRUM

To apply Unified Process

# Outline

## Model Driven Design

- Some Definitions
- Principles
- Examples
- Some suggestions for Agile

Tutorial on Requirements Eng. and SCRUM reflections (D402a, s601c)

Unified Process

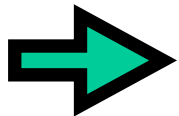
## Definitions of “Architecture”

*... the highest level concept of a system in its environment*

*a shared understanding of the system design ... a social construct*

*things that people perceive as hard to change*

*one of an architect's most important tasks is to remove architecture by  
finding ways to eliminate irreversibility in software designs.*



Fowler, 2003

# Model Driven Architecture

*Model-driven development is simply the notion that we can construct a model of a system that we can then transform into the real thing. (Mellor, Clark & Futagami, 2003)*

# What is a model?

A model is a coherent set of formal elements describing something (for example, a system, bank, phone, or train) built for some purpose that is amenable to a particular form of analysis, such as:

Communication of ideas between people and machines

Completeness checking

Race condition analysis

Test case generation

Viability in terms of indicators such as cost and estimation

Standards

Transformation into an implementation

# Models

Statements about a system under study (SUS)

- A *correct* model makes only true statements
- Often incomplete in concepts and/or details
- Make value judgments about what's important

Characteristics of a useful model

- Abstraction of the SUS
- Understandable
- Accurate
- Predictive
- Inexpensive (relative to the SUS)

Models become primary development artifacts in MDA

Doug Tolbert, 2004



# Notorious Failures: CASE

In the 1980's, CASE technologies promised to marry design and implementation technologies

## Multiple failures

- Model-to-implementation mapping abstractions weak
- Immature enabling technologies
  - Code generators, middleware, deployment
- Vendor hype exceeded capabilities
- Visible product failures (AD/Cycle)

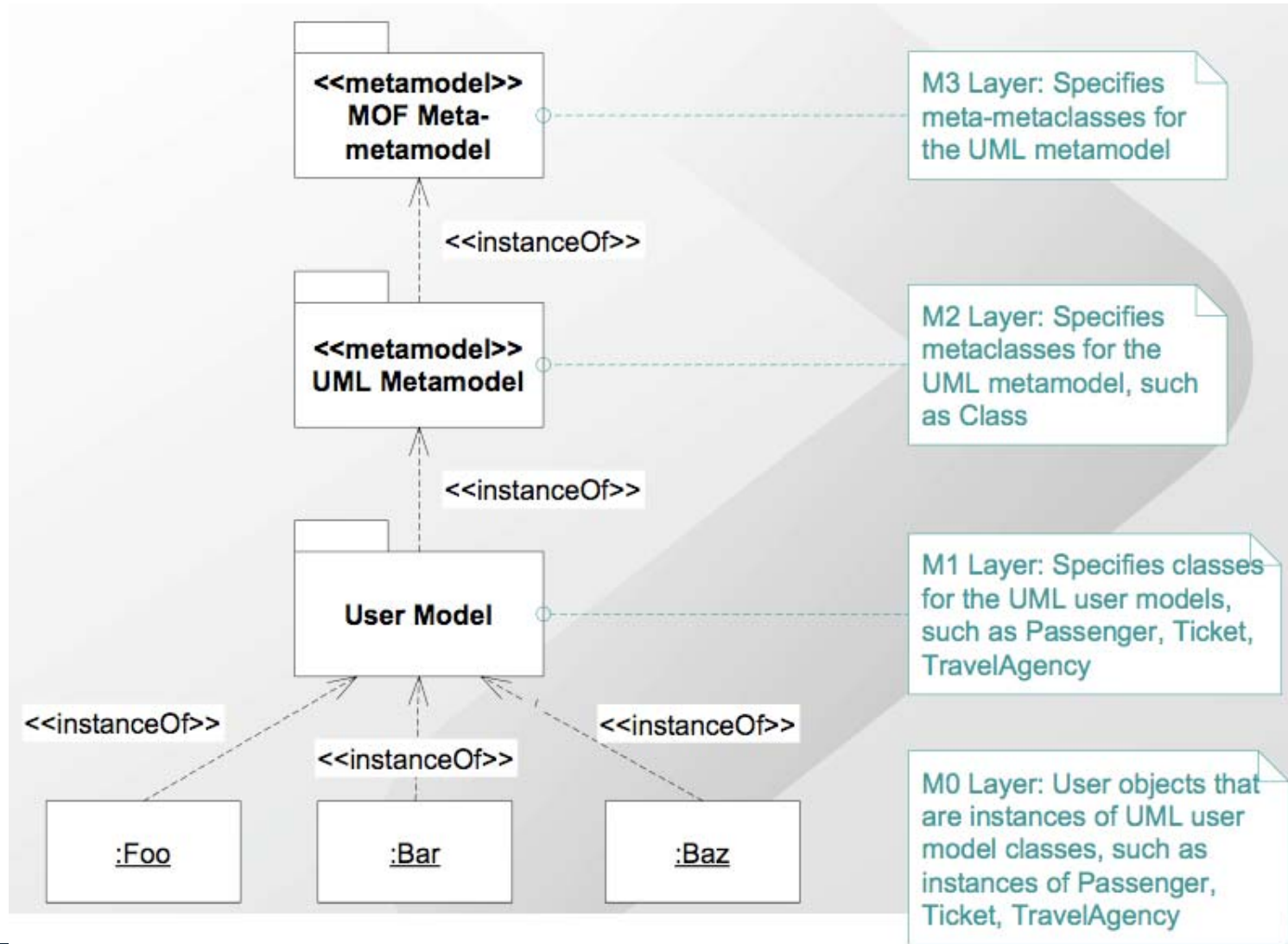
Fueled market skepticism about value of underlying technologies

Doug Tolbert, 2004

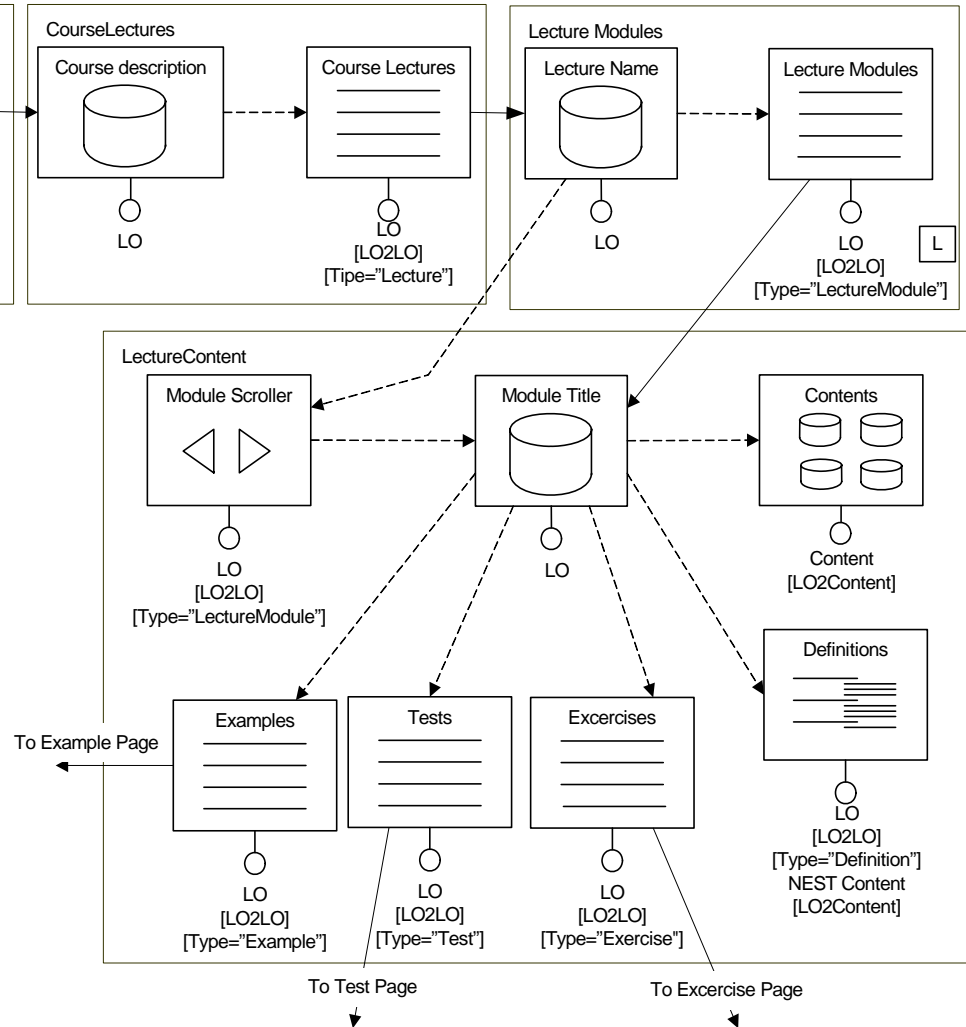
## Mellor et al. 2003

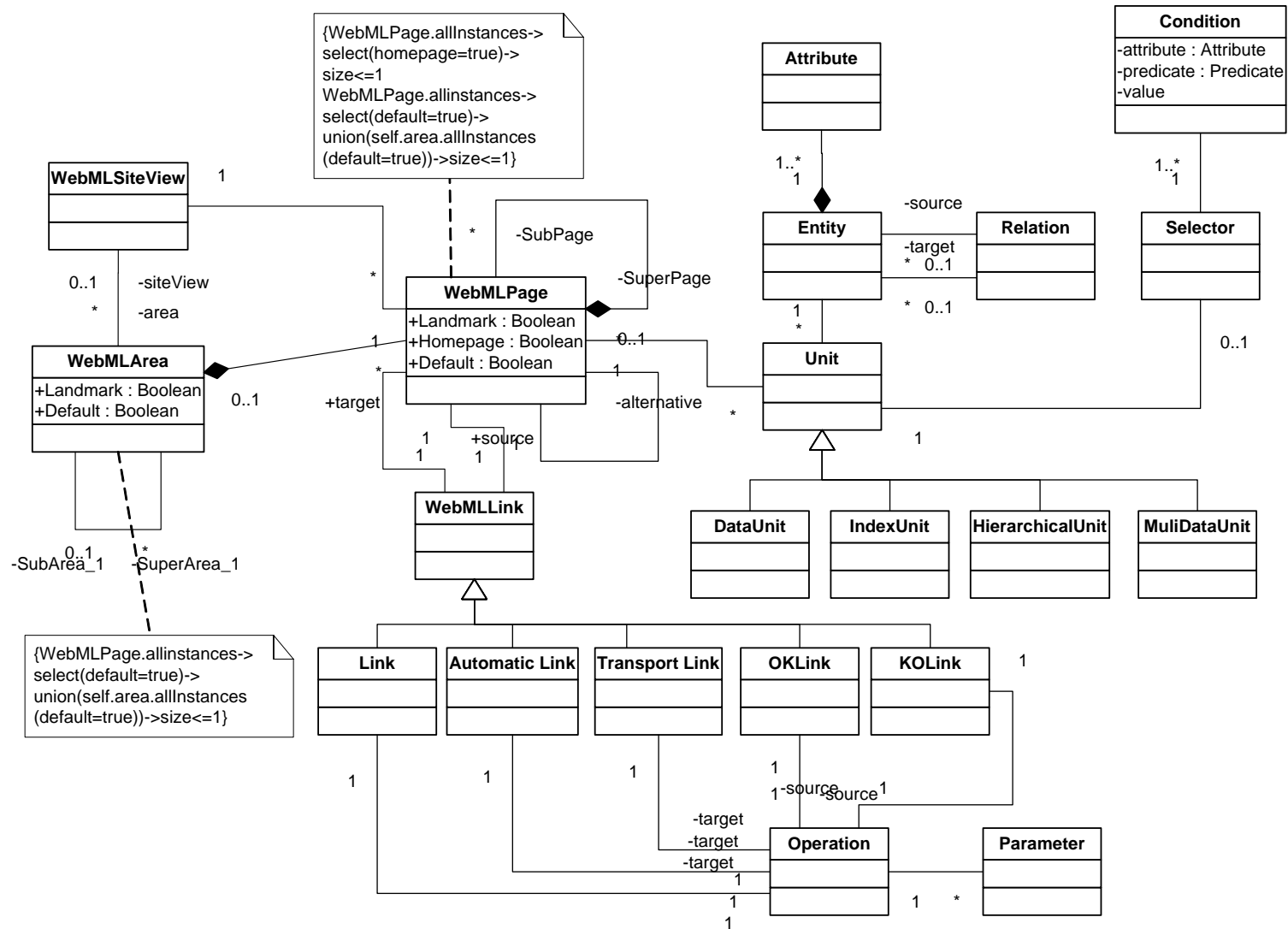
- ... model-driven development offers the potential for automatic transformation of high-level abstract application-subject matter models into running systems
- ... modeling technology has matured to the point where it can offer significant leverage in all aspects of software development
- ... in an increasing number of application areas, you can generate much of the application code directly from models

# OMG –Metamodel Architecture

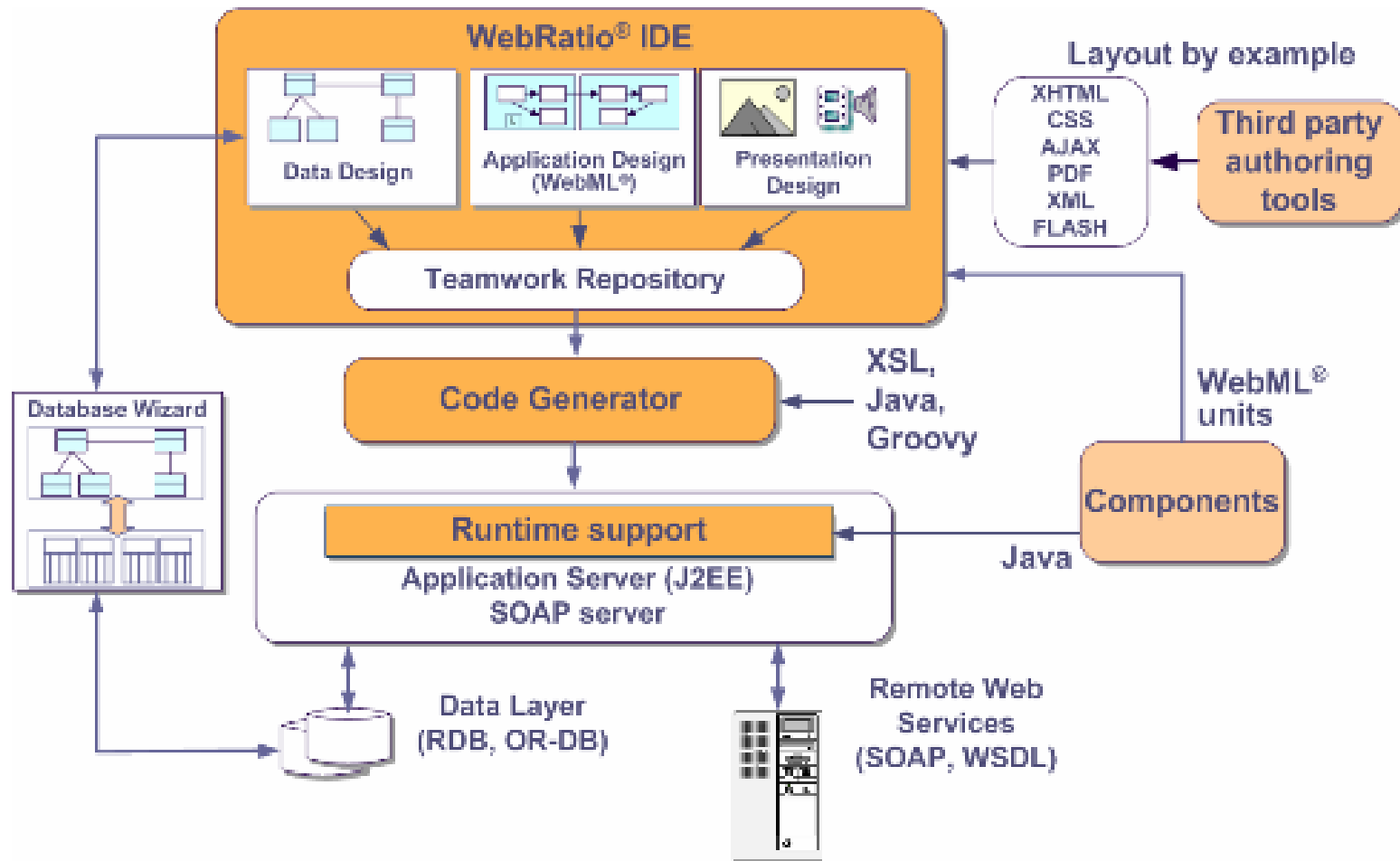


# Webml.org





# Webratio.com



# Point, Counterpoint

*MDA is the next logical evolutionary step to complement 3GLs in the business of software engineering*

*Axel Uhl, 2003*

*Has it been 10 years already? The “uber-modeling tool” vision rears its ugly head yet again*

*Scott Ambler, 2003*

## Ambler, 2003

*Generative MDD, epitomized by the Object Management Group's Model Driven Architecture, is based on the idea that people will use very sophisticated modeling tools to create very sophisticated models that they can automatically "transform" with those tools to reflect the realities of various deployment platforms. Great theory—as was the idea that the world is flat.*

*... I believe that modeling is a way to think issues through before you code because it lets you think at a higher abstraction level.*

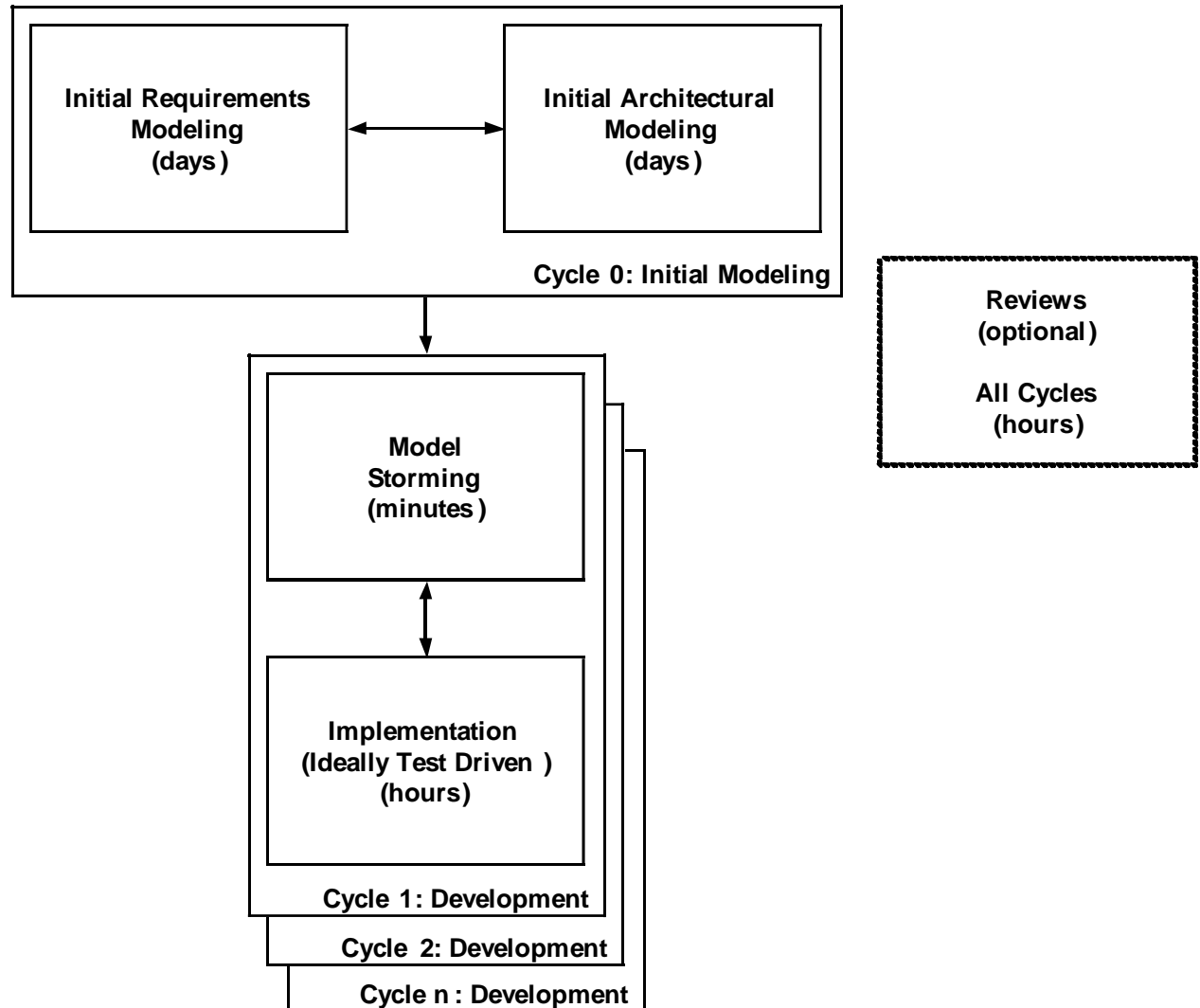


# Agile MDD (AMDD) Project Level

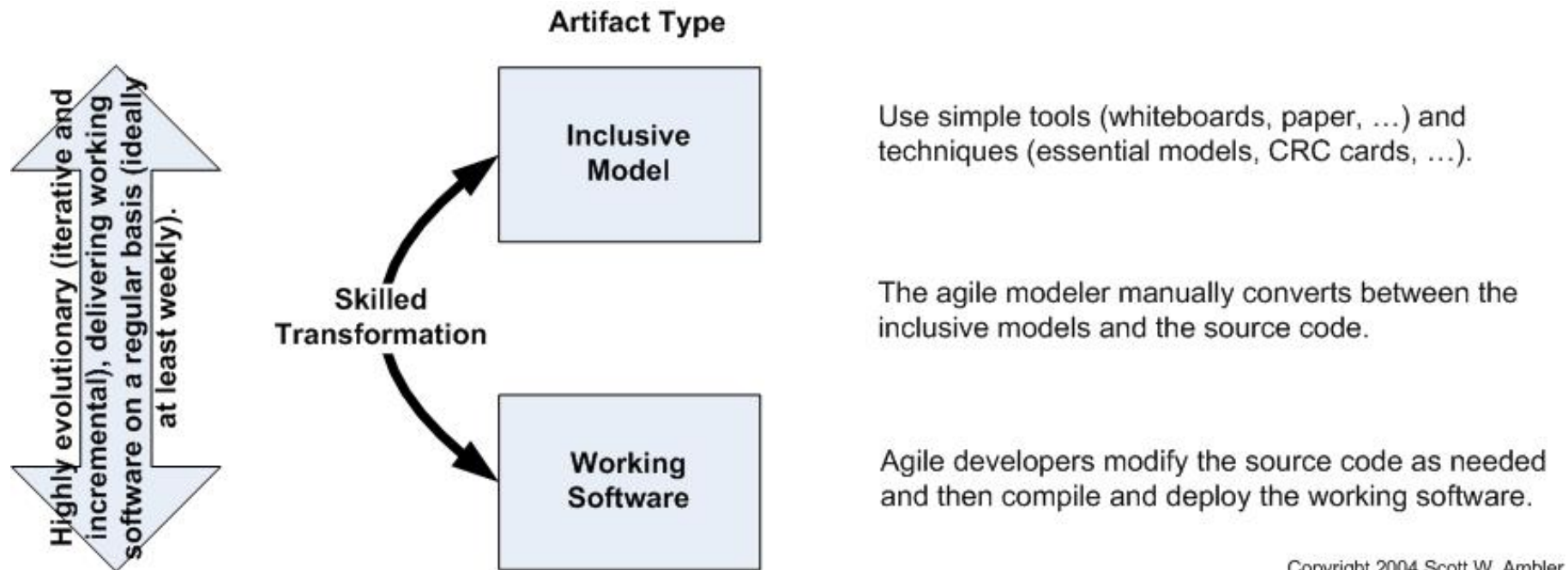
**Goals:** Gain an initial understanding of the scope, the business domain, and your overall approach.

**Goal:** Quickly explore in detail a specific issue before you implement it.

**Goal:** Develop working software in an evolutionary manner.

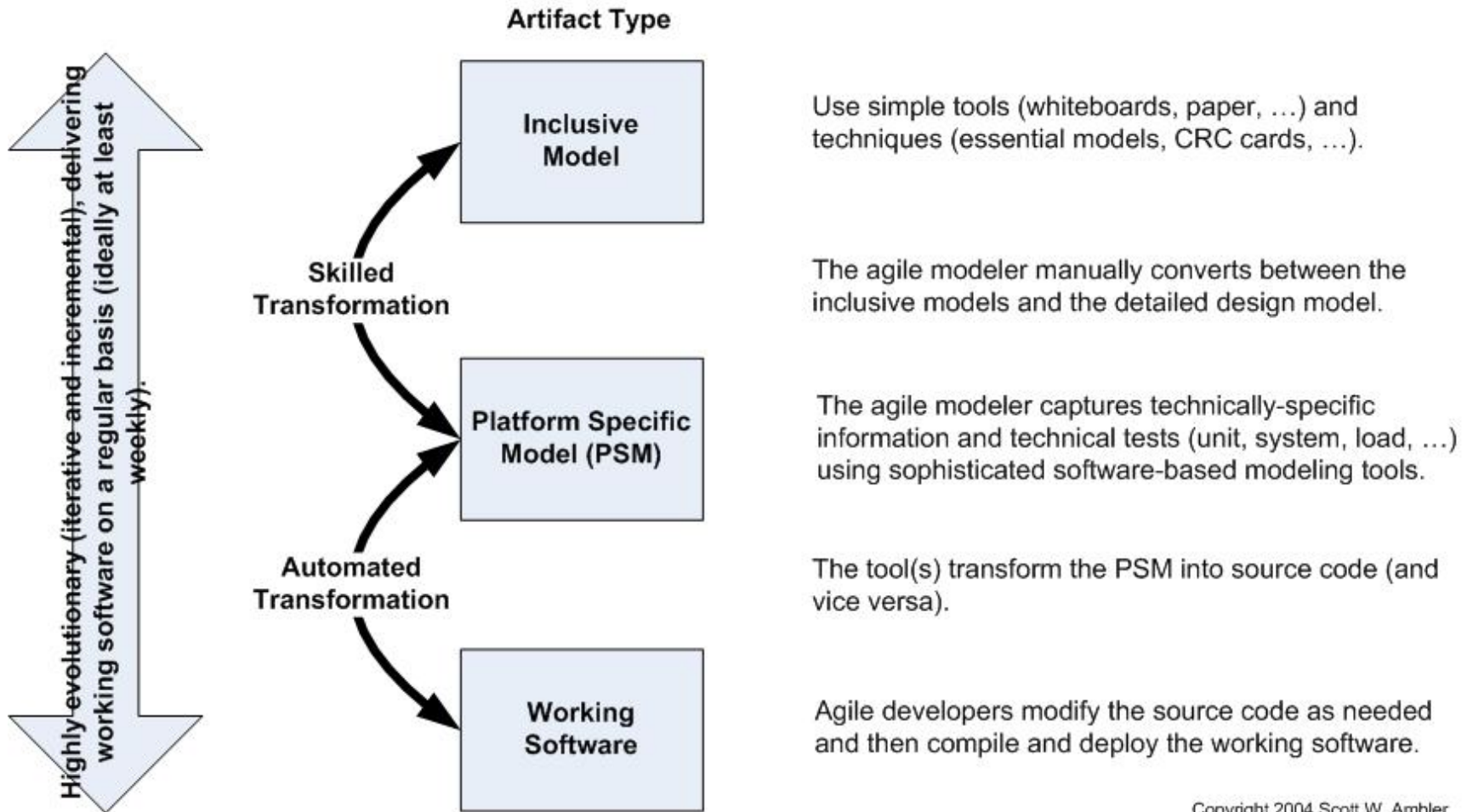


# AMDD – Simple Approach



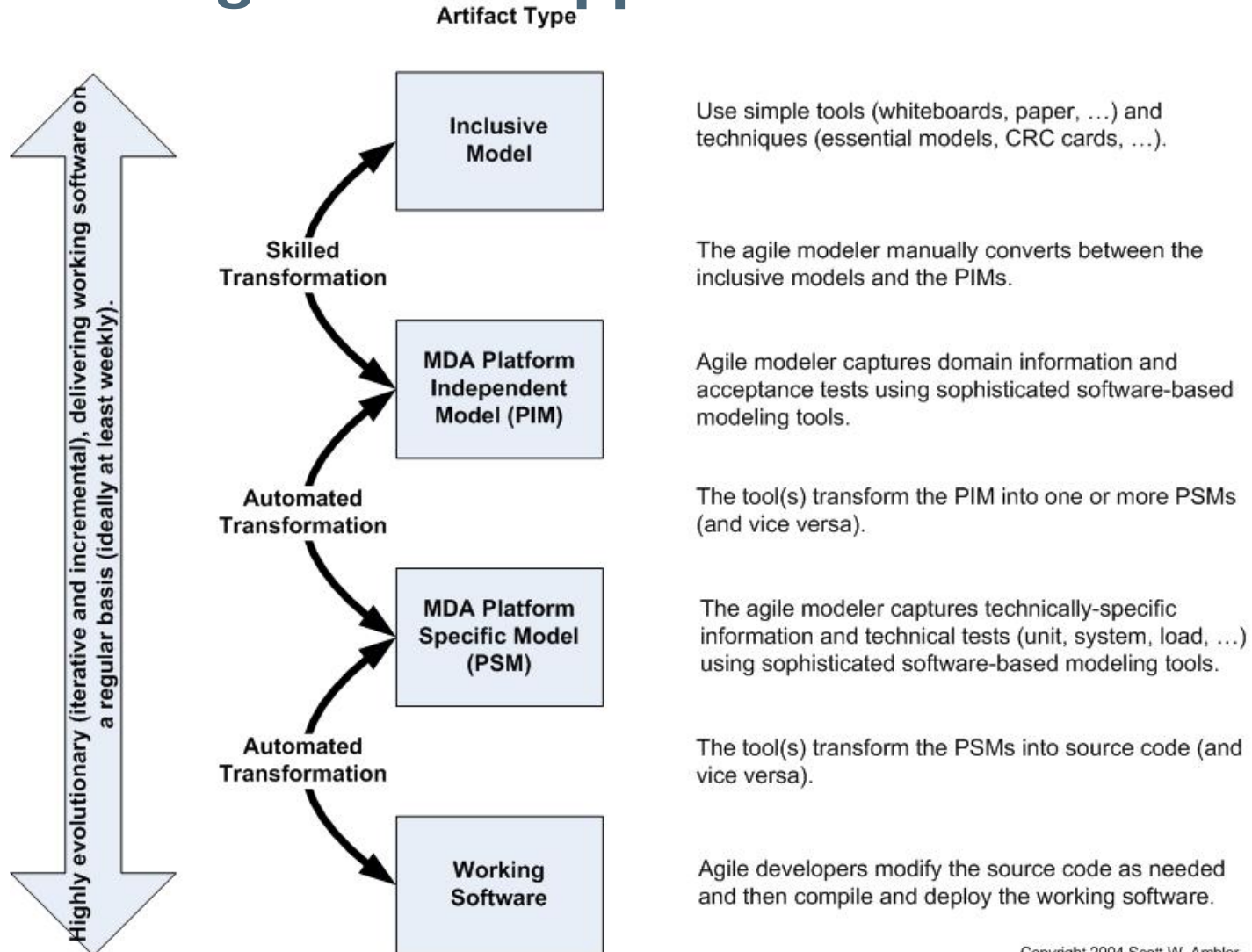
Copyright 2004 Scott W. Ambler

# AMDD – CASE Approach



Copyright 2004 Scott W. Ambler

# AMDD – Agile MDA Approach



# Outline

## Model Driven Design

- Some Definitions
- Principles
- Examples
- Some suggestions for Agile

**Tutorial on Requirements Eng. and SCRUM reflections (D402a, s601c)**

Unified Process