Object Architects

MDA for the “Olde Boys and Girls” and again No Silver Bullet

WHO is serving you this today?

Object Architects

- a guy who is in large scale professional software development since around 1990
- a person who is over forty and who has seen a few hypes before and who should hence be eliminated from the trade in order to get out of the way of MDA marketers ☺
- somebody who thinks that elements of MDA may help – but that the hype is counter productive. MDA is just one discipline of Generative Programming

HOW
Overview

- Silver Bullets in Literature
  - Jerry Weinberg and Fred Brooks revisited
- MDA using COBOL
  - for Object-Oriented Programming on the Mainframe in 1992-1993
- Are Insurance Product Servers a use of MDA?
  - they are a DSSA!
- Summary

1973 – 1998 (25 years)
Gerald M. Weinberg – The Psychology of Computer Programming, Silver Anniversary Edition

1973: over the years, executives have backed their desire to eliminate programmers with staggering funds. (p. 4)

1998: the only thing that has changed here in twenty five years is the fact that the funds dedicated by executives to eliminating programmers from their payrolls have become far more staggering than I imagined back then. And, now I finally recognize in this executive desire a pattern so strong, so strong so emotional, that it has blinded these executives in two facts

1. None of these schemes has succeeded in elimination programmers. **We have now at least 10 times as many as we did then.**
2. **Every one of these schemes has been concocted by the programmers themselves**, the very people the executives passionately want to eliminate.

1975 – 1995 (20 years)

Fred Brooks: The Mythical Man Month; 20th Anniversary Edition, Chapter 16: “No Silver Bullet”

There is no single development in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity

1975 – 1995 (20 years)

Fred Brooks: The Mythical Man Month; 20th Anniversary Edition, Chapter 16: “No Silver Bullet”

Automatic Programming

For almost 40 years, people have been anticipating and writing about „automatic programming“, the generation of a program for solving a problem from a statement of the problem specifications. Some people today write as if they expected this technology to provide the next breakthrough.

Parnas implies that the term is used for glamour and not semantic content, asserting,

> ***in short, automatic programming always has been a euphemism for programming with a higher-level language than was presently available to the programmer***

Did you recognize, what we were talking about ...

Object Architects

4GL?

OOP?

ADA?

Expert Systems?

C++?

Program Verification?

MDA!

Graphical Programming?

Reuse?

CASE?

Automatic Programming?

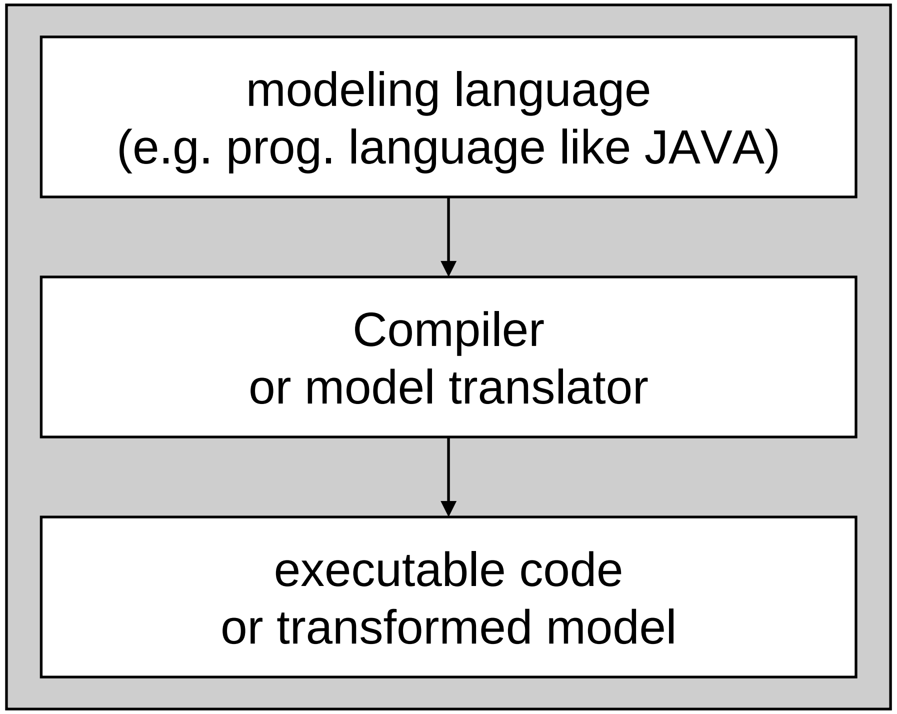
Artificial Intelligence?

HOW
Overview

Object Architects

- Silver Bullets in Literature
  - Jerry Weinberg and Fred Brooks revisited
- MDA using COBOL
  - for Object-Oriented Programming on the Mainframe in 1993
- Are Insurance Product Servers a use of MDA?
  - they are a DSSA!
- Summary

The Use of MDA Terms for this Talk One PStack

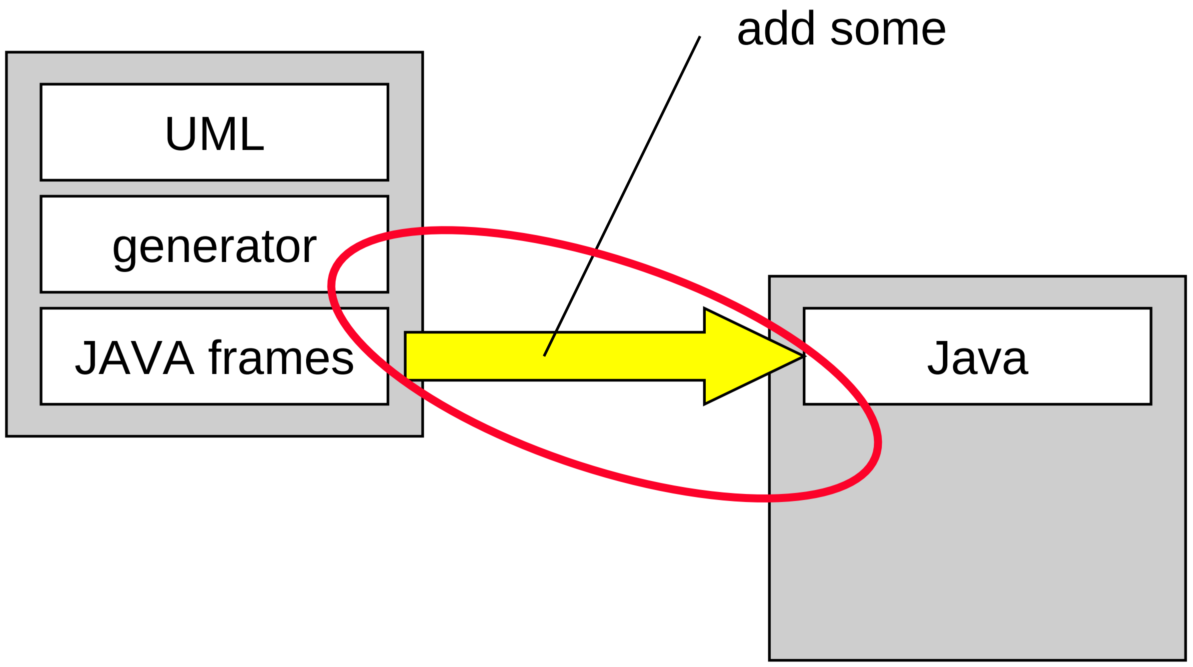


ease of expression, easier to undestand, portable, standardized problem oriented

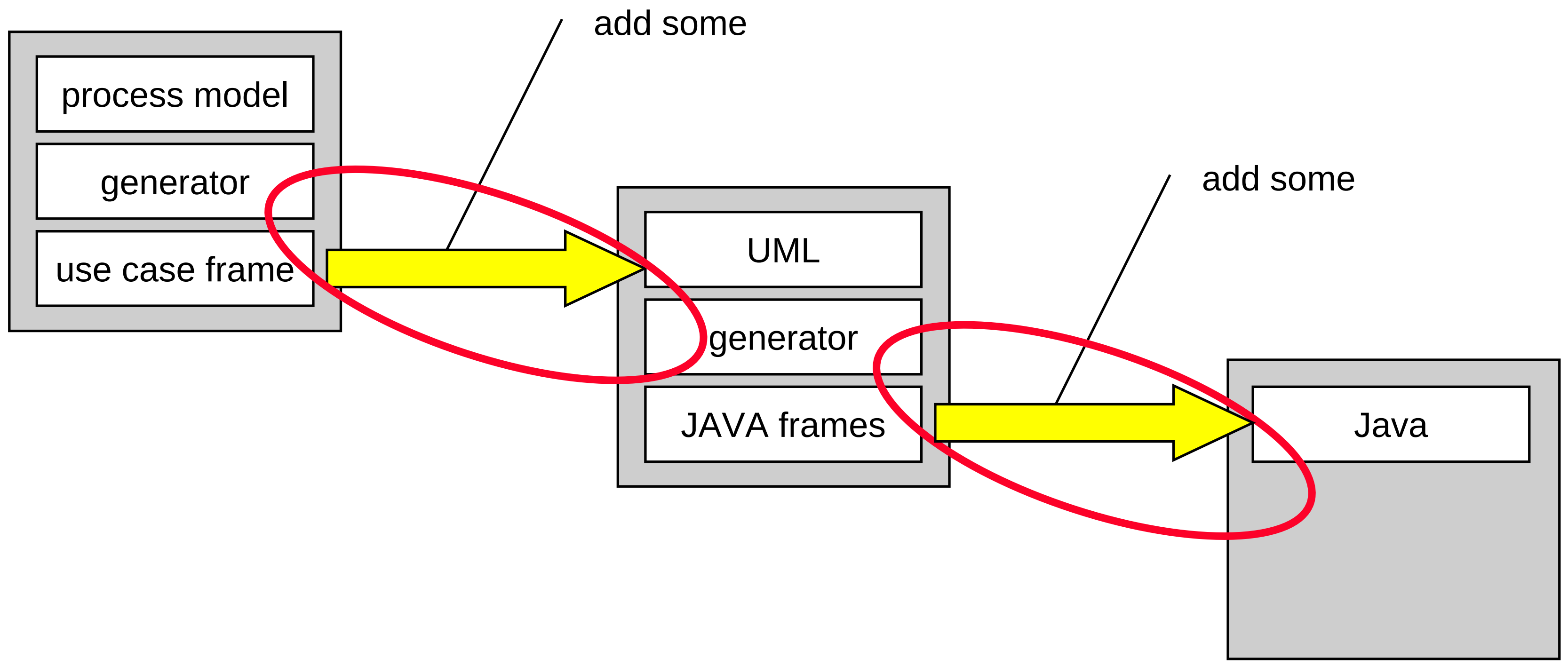
machine dependent, optimized, configurable, flexible

optimized for one special machine

The Use of MDA Terms for this Talk
Two PStacks

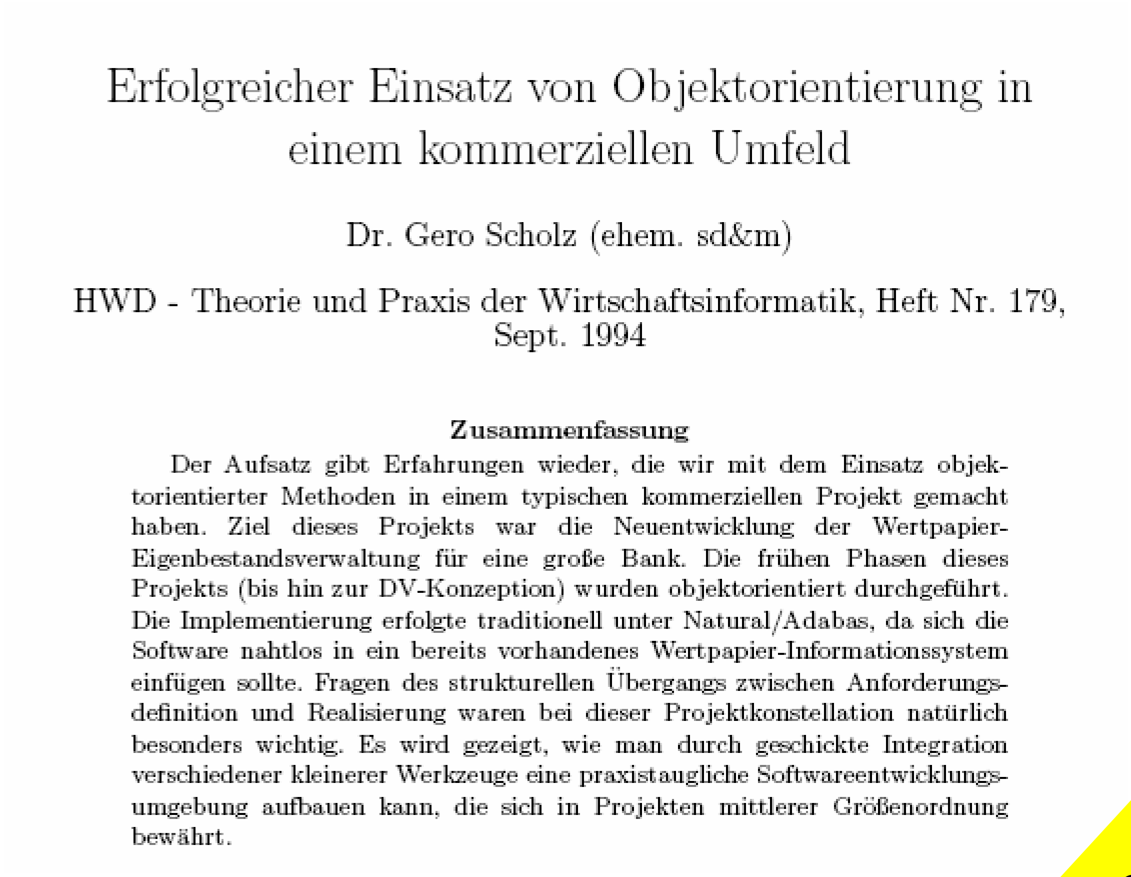


The Use of MDA Terms for this Talk or even three PStacks



and how the „2 PStacks“ model was applied before MDA was a term ...

Object Architects



Erfolgreicher Einsatz von Objektorientierung in einem kommerziellen Umfeld

Dr. Gero Scholz (ehem. sd&m)

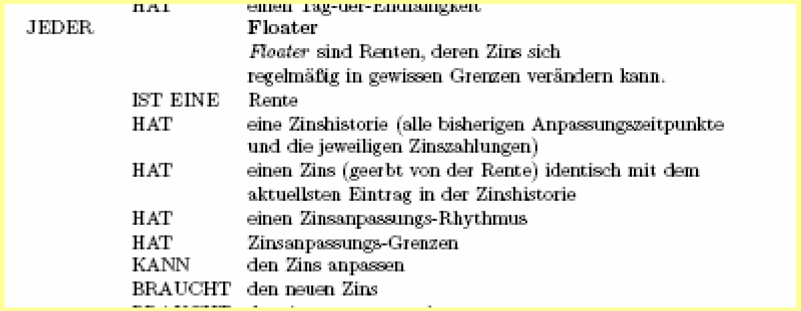
HWD - Theorie und Praxis der Wirtschaftsinformatik, Heft Nr. 179, Sept. 1994

Zusammenfassung

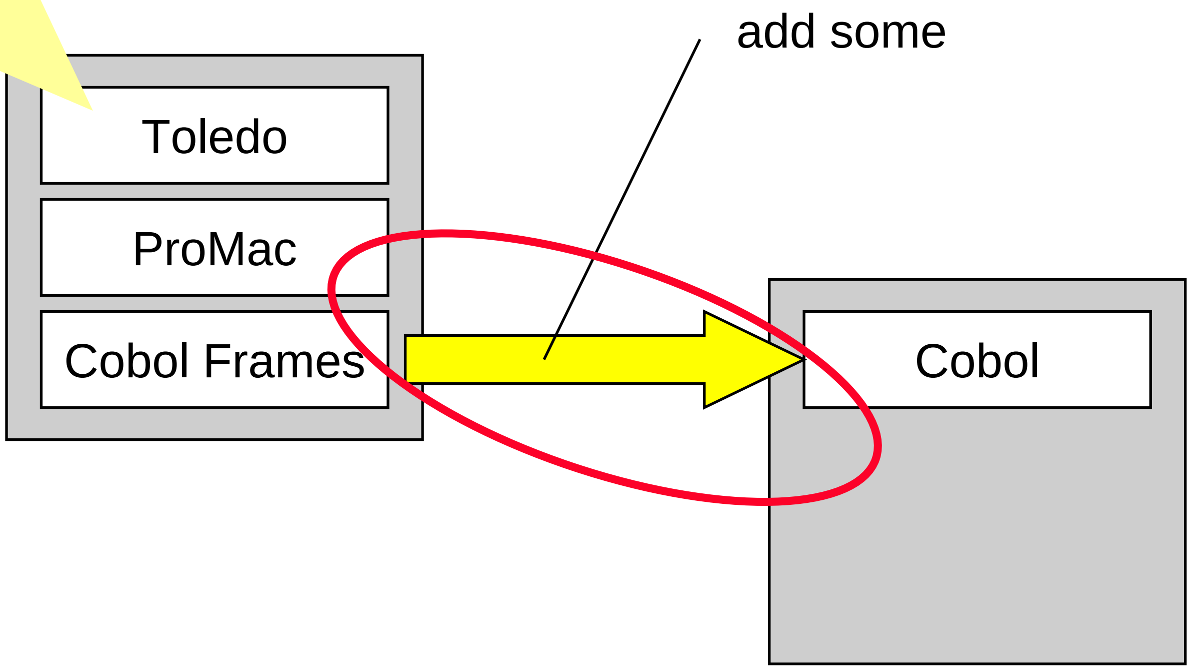
Der Aufsatz gibt Erfahrungen wieder, die wir mit dem Einsatz objektorientierter Methoden in einem typischen kommerziellen Projekt gemacht haben. Ziel dieses Projekts war die Neuentwicklung der Wertpapier-Eigenbestandsverwaltung für eine große Bank. Die frühen Phasen dieses Projekts (bis hin zur DV-Konzeption) wurden objektorientiert durchgeführt. Die Implementierung erfolgte traditionell unter Natural/Adabas, da sich die Software nahtlos in ein bereits vorhandenes Wertpapier-Informationssystem einfügen sollte. Fragen des strukturellen Übergangs zwischen Anforderungsdefinition und Realisierung waren bei dieser Projektkonstellation natürlich besonders wichtig. Es wird gezeigt, wie man durch geschickte Integration verschiedener kleinerer Werkzeuge eine praxistaugliche Softwareentwicklungsumgebung aufbauen kann, die sich in Projekten mittlerer Größenordnung bewährt.

case from 1992-1994

Toledo was a Language similar to the Expressive Power of UML



	HAT	einen Tag-der-Endfälligkeit
JEDER		**Floater**
		Floater sind Renten, deren Zins sich regelmäßig in gewissen Grenzen verändern kann.
	IST EINE	Rente
	HAT	eine Zinshistorie (alle bisherigen Anpassungszeitpunkte und die jeweiligen Zinszahlungen)
	HAT	einen Zins (geerbt von der Rente) identisch mit dem aktuellsten Eintrag in der Zinshistorie
	HAT	einen Zinsanpassungs-Rhythmus
	HAT	Zinsanpassungs-Grenzen
	KANN	den Zins anpassen
	BRAUCHT	den neuen Zins
	[illegible]	[illegible]



MDA – Experiences (I) from the time before MDA

- Round tripping was a crucial factor then and is still a crucial factor today
  - if you have no perfect roundtrip this compensates for a lot of the positive effects of the generator
- Unless the stack is really „mature“, you will have to „add some“ after each generation step – emphasizing the need for good round tripping
  - look at the first compilers and compilers today – today a compiler writes code better than handwritten code
- If you have used Delta or have used a lot of C preprocessor macros you were already using MDA ☺

MDA – Experiences (II) from the time before MDA

Object Architects

- Code Generation and use of templates are around for quite a while and are often very helpful, like in
  - Generation of boring code of any flavour
  - Think of EJB deployment descriptors, database access layer code and the like
- Generative Programming and hence also MDA offer a lot of relief from boring vanilla code writing

MDA Experiences (III) from other people

- think of database schema evolution in an MDA project
  - BTW: Same problem as with an O/R acces layer – this may become an interesting GOTCHA if ignored
- think of how to maintain the templates behind the MDA generators
  - this requires some real experts plus the „refresh“ may become interesting if people have „improved“ the generated code (-> see perfect roundtrip)
- think of debugging the generated code – or will we debug the models? ☺

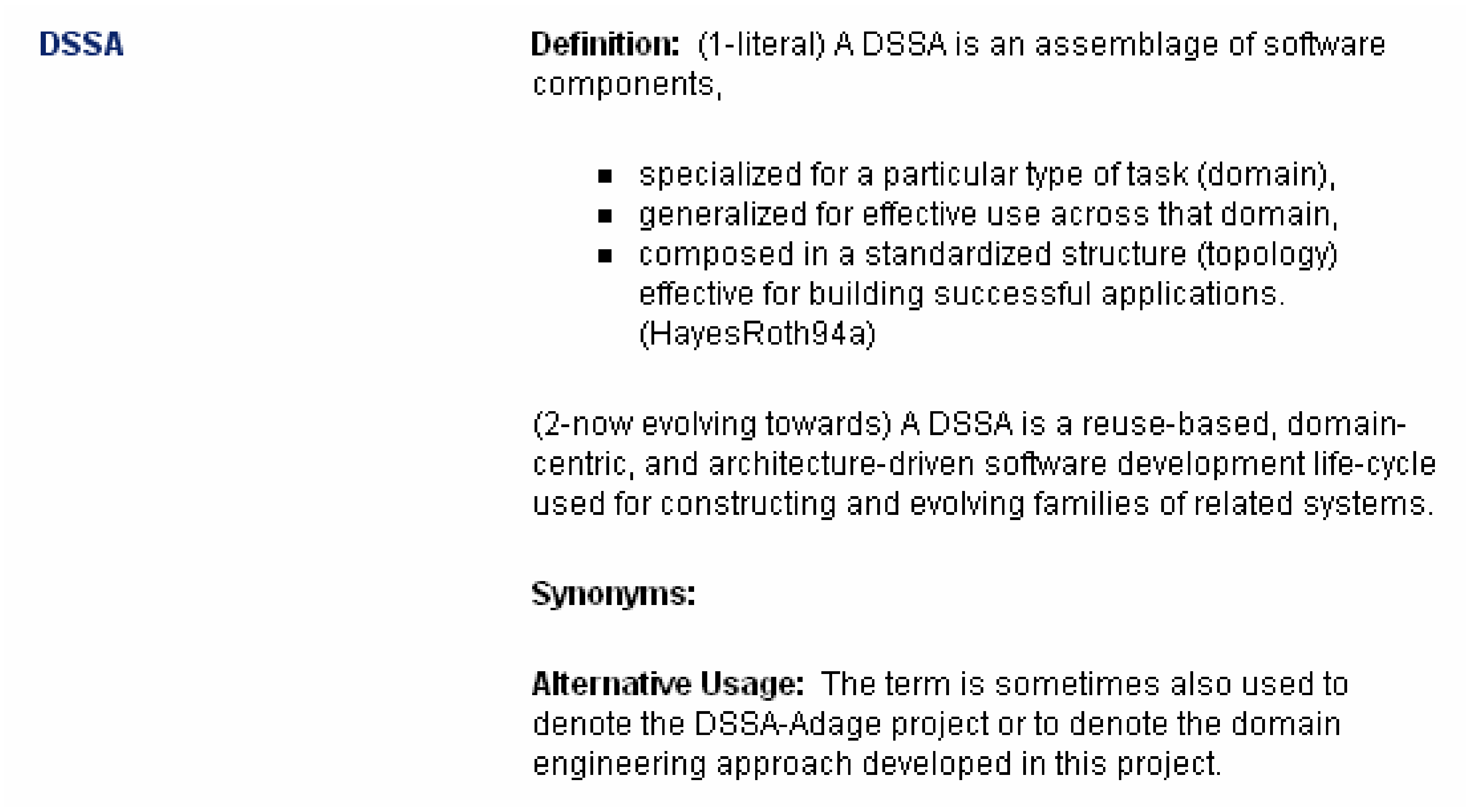
MDA Experiences (IV) from other people

- who would really like to port a Java app to .NET exploiting the platform independence potential of MDA?

HOW
Overview

- Silver Bullets in Literature
  - Jerry Weinberg and Fred Brooks revisited
- MDA using COBOL
  - for Object-Oriented Programming on the Mainframe in 1993
- Are Insurance Product Servers a use of MDA?
  - they are a DSSA!
- Summary

Domain Specific Software Architectures



DSSA

Definition: (1-literal) A DSSA is an assemblage of software components,

- specialized for a particular type of task (domain),
- generalized for effective use across that domain,
- composed in a standardized structure (topology) effective for building successful applications. (HayesRoth94a)

(2-now evolving towards) A DSSA is a reuse-based, domain-centric, and architecture-driven software development life-cycle used for constructing and evolving families of related systems.

Synonyms:

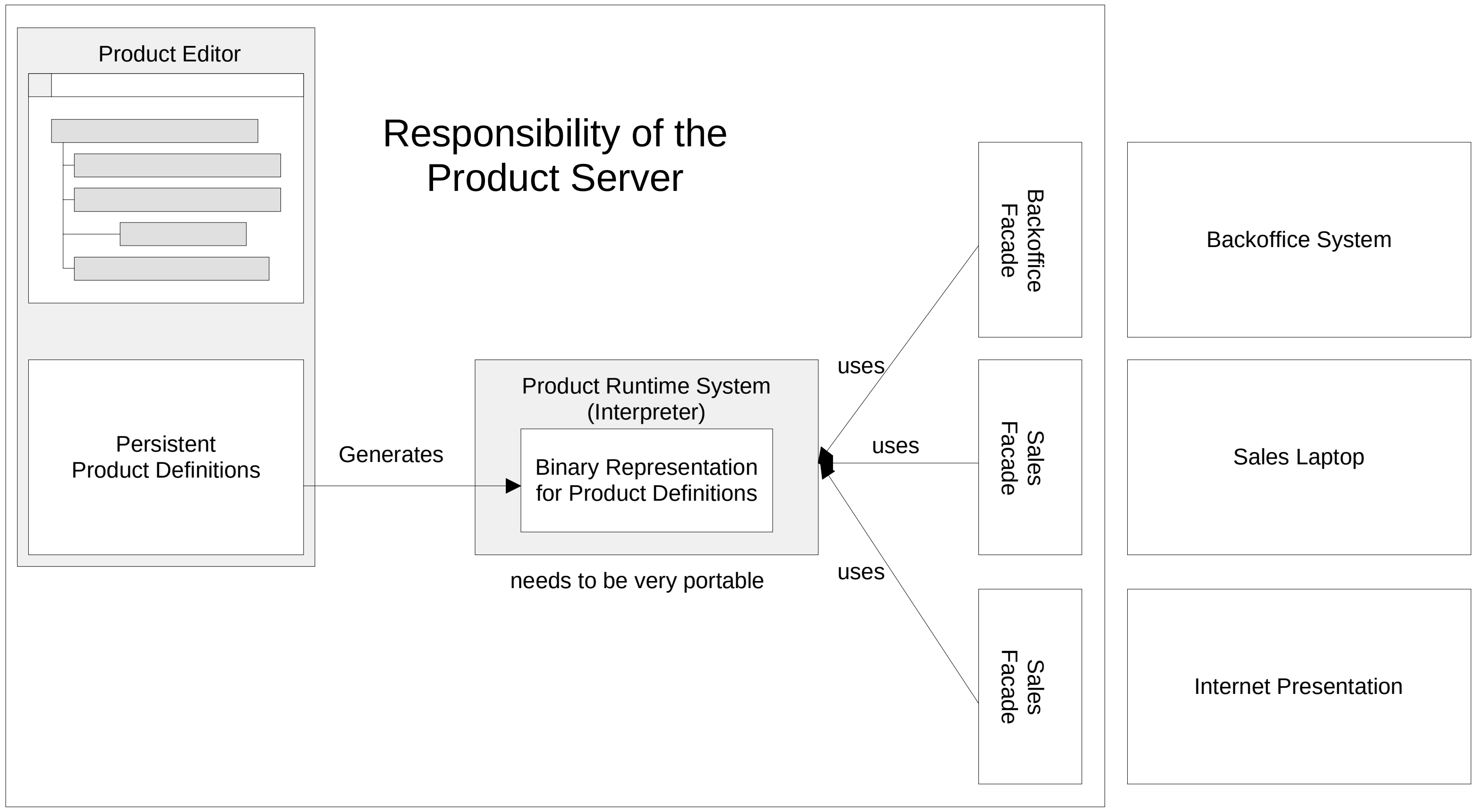
Alternative Usage: The term is sometimes also used to denote the DSSA-Adage project or to denote the domain engineering approach developed in this project.

Quelle: http://www.iese.fhg.de/pubs_and_links/spl/bibliography/definitions/index.html

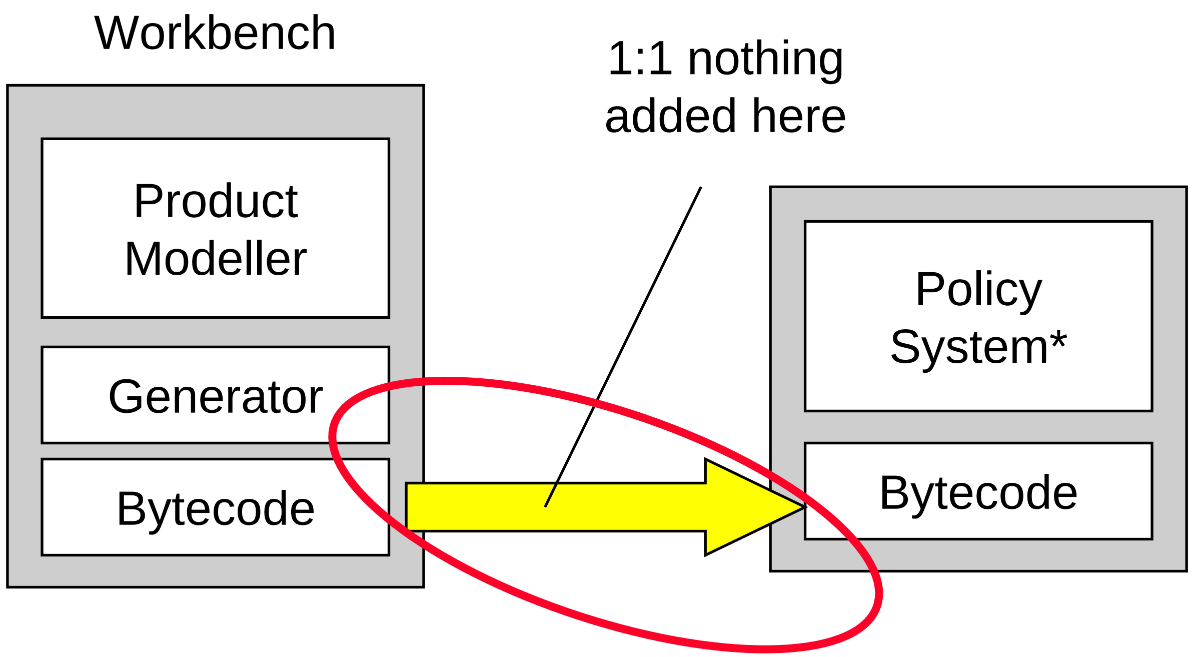
Examples of DSSAs from literature

- Aircraft simulator software suites
- Combat simulators
- ...
- Insurance Product Servers

Insurance Product Server Architectural Pattern

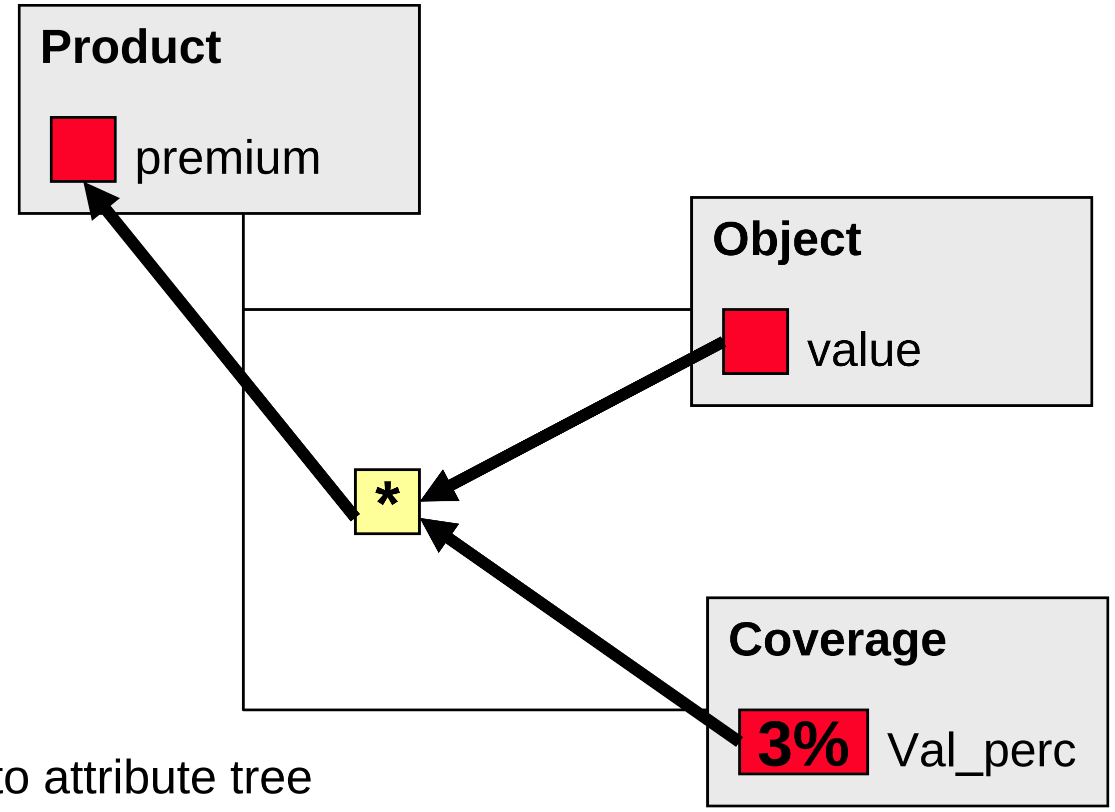

Product Editor
Responsibility of the Product Server
Persistent Product Definitions
Generates
Product Runtime System (Interpreter)
Binary Representation for Product Definitions
needs to be very portable
uses
uses
uses
Backoffice Facade
Sales Facade
Sales Facade
Backoffice System
Sales Laptop
Internet Presentation

The Difference



*or Claims System or Sales System

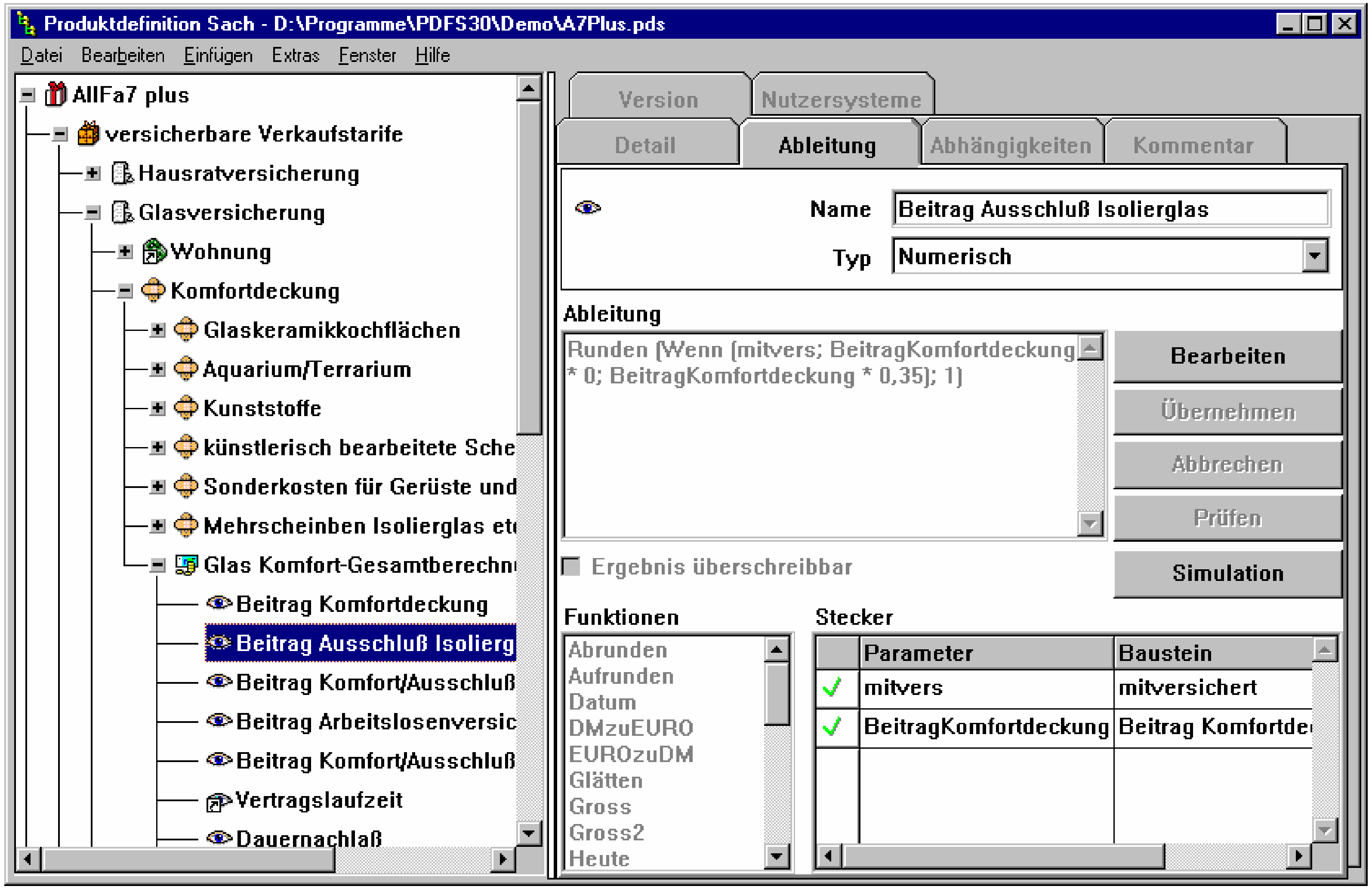
Insurance Product Server Modeling Language Idea



Analogous to attribute tree evaluations in compiler building

Insurance Product Server Modeler‘s User Interface Impression

Object Architects


Produktdefinition Sach - D:\Programme\PDFS30\Demo\A7Plus.pds
Datei Bearbeiten Einfügen Extras Fenster Hilfe
Version
Nutzersysteme
Detail
Ableitung
Abhängigkeiten
Kommentar
Bearbeiten
Übernehmen
Abbrechen
Prüfen
Simulation

- AllFa7 plus
  - versicherbare Verkaufstarife
    - Hausratversicherung
    - Glasversicherung
      - Wohnung
      - Komfortdeckung
        - Glaskeramikkochflächen
        - Aquarium/Terrarium
        - Kunststoffe
        - künstlerisch bearbeitete Sche
        - Sonderkosten für Gerüste und
        - Mehrscheinben Isolierglas etc
        - Glas Komfort-Gesamtberechn
          - Beitrag Komfortdeckung
          - Beitrag Ausschluß Isolierg
          - Beitrag Komfort/Ausschluß
          - Beitrag Arbeitslosenversic
          - Beitrag Komfort/Ausschluß
          - Vertragslaufzeit
          - Dauernachlaß

Name: Beitrag Ausschluß Isolierglas

Typ: Numerisch

Ableitung

Runden (Wenn (mitvers; BeitragKomfortdeckung * 0; BeitragKomfortdeckung * 0,35); 1)

☐ Ergebnis überschreibbar

Funktionen

Abrunden
Aufrunden
Datum
DMzuEURO
EUROzuDM
Glätten
Gross
Gross2
Heute

Stecker

	Parameter	Baustein
✓	mitvers	mitversichert
✓	BeitragKomfortdeckung	Beitrag Komfortde

Summary
The 4 Key Messages Revisited

- MDA is NSB (**No Silver Bullet** ☺).

- MDA will **NOT** automate programming or improve it by an order of magnitude. You will **NOT** get rid of programmers. **MDA technology was around before the term was born**

- For Insurance Product Modeling there are also other technologies (which are not called MDA) – uses of so called **DSSA**s – also known as Product Servers. They come in many flavors such as IP.Suite, VP/MS, Allis PEAP and others

- As with any technology MDA can be useful, once the hype is over or people have a cool attitude towards using it and will exploit its benefits. **Generative Programming** as a whole offers a lot of useful concepts