

CASE-Tools der Welt

Oder:
Softwareentwicklungswerkzeuge
jenseits von emacs, gcc, Visual C++

Themen

- ◆ Geschichtliches
- ◆ Kategorien von CASE-Entwicklern/Anwendern
- ◆ Kategorien von CASE-Werkzeugen
 - Beispiele
- ◆ Was wird?

Die Softwarekrise

- ◆ 60er Jahre bis ?
- ◆ Dijkstra (72): *„Als es noch keine Rechner gab, war auch das Programmieren noch kein Problem, als es dann ein paar leistungsschwache Rechner gab, was das Programmieren ein kleines Problem und nun, wo wir gigantische Rechner haben, ist auch das Programmieren ein gigantisches Problem geworden. In diesem Sinne hat die elektronische Industrie kein einziges Problem gelöst, sondern nur neue geschaffen. Sie hat das Problem geschaffen, ihre Produkte zu nutzen.“*

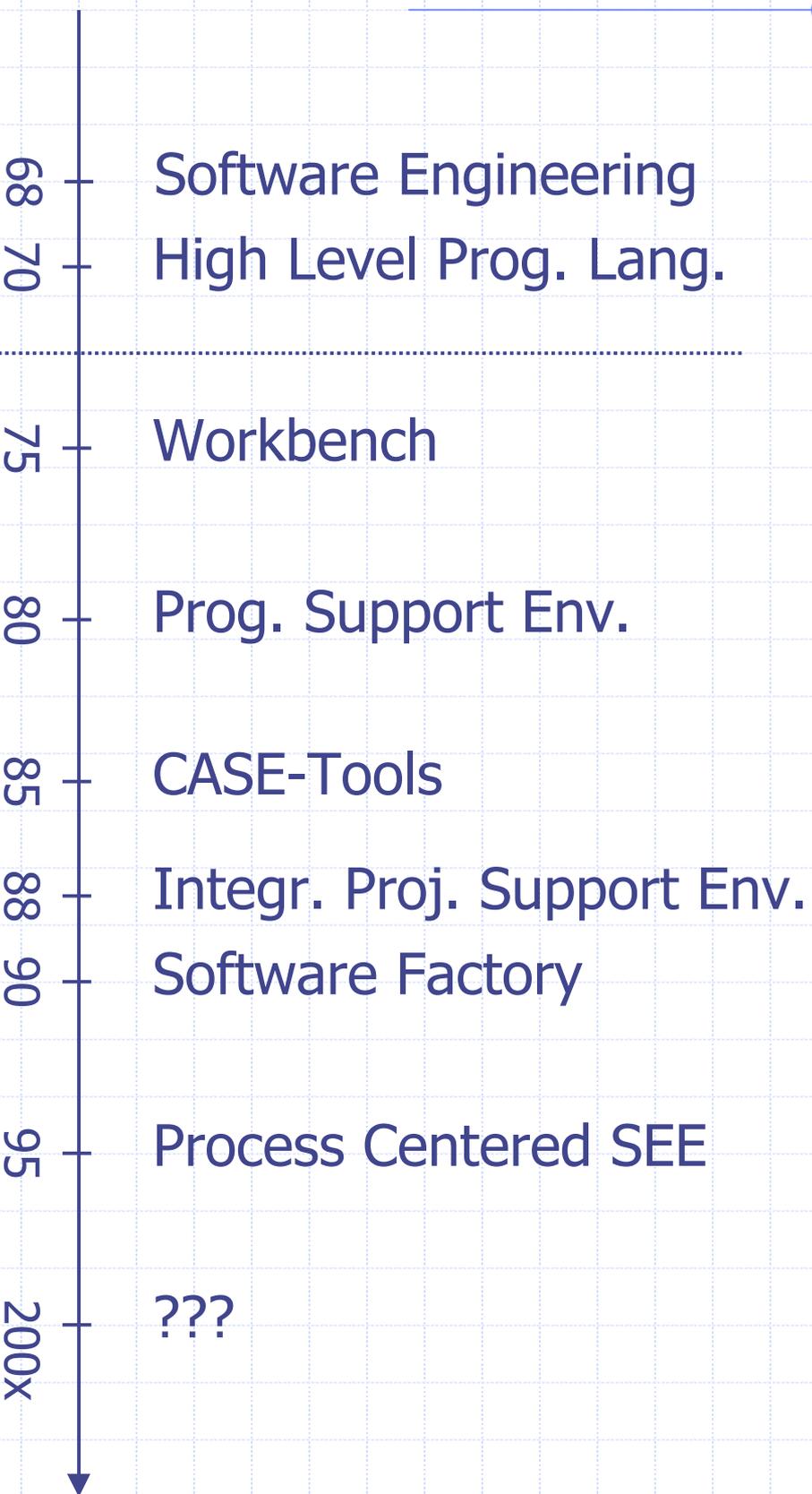
Schwierigkeiten bei der SWE

- ◆ Software ist immateriell – keine physikalischen Grenzen
- ◆ Software lässt sich leicht modifizieren
- ◆ Software altert
- ◆ Software muss auf versch. Plattformen laufen
- ◆ Software wird in vielen Varianten ausgeliefert
- ◆ Software greift in bestehende Arbeitsabläufe ein
- ◆ Mangel an Standards, Methoden, Werkzeuge

Historische Entwicklung

3. Juli 2001 - Perlen

Peter Braun



Kategorien von CASE-Tool-Entwicklern/-Anwendern

- ◆ Framework-Entwickler: ein undankbarer Job
- ◆ Werkzeug-Entwickler: born(?) in the USA
- ◆ Werkzeuganpasser: an der Front
- ◆ Anwendungsentwickler: die arme Sau
- ◆ Softwareentwicklungsmanager: ein riskanter Job
- ◆ CASE-Werkzeugforscher: 😊

Workbench (1)

◆ „Revolutionäres“ Unix:

- „ein“ Dateityp, hierarchisches Dateisystem
- Werkzeuge kommunizieren über stdin/stdout
- Portables Betriebssystem
- Konzepte für Prozesskommunikation
- Pipes!!!

◆ Programmieren = **Wiederverwenden von Komponenten**

Workbench (2)

◆ Populäre Werkzeuge:

- Textverarbeitung: nroff, troff, ...
- Metawerkzeuge: lex, yacc
- Versions- / Konfigurationsverwaltung: sccs, rcs, cvs
- DSEE: cvs + make + vert. Dateisystem + ...

◆ UNIX war Forschungsprodukt!

Programming Support Environment (1)

- ◆ PSE=Eng integrierte Sammlung von Werkzeugen
- ◆ Eigenschaften:
 - Fenster, Maus, Menüs
 - Werkzeuge hochgradig interaktiv; Browsing
 - „Edit/Compile/Debug-Zyklus“ auf die Spitze getrieben

PSE (2)

◆ Historische Beispiele:

- Interlisp: Bolt, Xerox (1966-1980)
- Smaltalk 80: Xerox (1980 – 1983)
- Ada PSE: DoD-Studien

◆ Aktuelle Beispiele:

- Smalltalk: Visual Works, Visual Age
- C++: Visual Age, Visual C++
- Java: Java Studio, Visual Age, Visual J++

PSE (3)

- ◆ Verschränkung Editieren/Ausführen
 - Quelltext **inkrementell** „kompilieren“
 - Zur Laufzeit möglich (-> Just-In-Time-Compiler)
- ◆ Start mit Lisp-Umgebungen
 - Interlisp vs. MacLisp (-> Emacs)
 - Strukturbezogene Editoren
 - interpretativ

Methodenorientierte CASE-Umgebungen

B. Meyer: *„In den 1980ern, mit dem Aufkommen von leistungsfähigen PCs und Workstations, wurde es möglich die Kreise und Pfeile mit Hilfe des Computers zu malen. Die korrespondierenden Werkzeuge bezeichnete man mit CASE, und eine ganze Industrie war drumherum geboren...“*

Merkmale von CASE-Env.

- ◆ Meist Unterstützung genau einer Methode
- ◆ Schwerpunkt: Anforderungsanalyse und Programmieren im Großen
- ◆ Komplementär zu PSE
- ◆ Probleme bei Integration von Arbeitsbereichen

Funktionen von OO-CASE-Env. (1)

◆ Editieren

- Syntaxgesteuerte Programmeditoren
- Texteditoren für Annotationen

◆ Analysieren

- Syntaktische Korrektheit durch Editor
- Einfache Analysen eingebaut
- Batchartige Analysen für komplizierte Konsistenzbedingungen

Funktionen von OO-CASE-Env. (2)

- ◆ Browsing
- ◆ Ausführen/Simulation
- ◆ Generieren
 - Skelette
 - DDL und IDLs
- ◆ Reverse Engineering

Funktionen von OO-CASE-Env. (3)

◆ Anbindung an die „Außenwelt“

- Textformat
- Skriptsprache
- Repository (z.B. Microsoft)

◆ Administrative Dienste

- Mehrbenutzerzugriff
- Merge

DOORS

- ◆ "The world-class requirements management tool" [DOORS Webpage]
- ◆ Requirements management and tracing tool

ODER

- ◆ Texteditor mit der Möglichkeit Links zwischen Textstellen zu ziehen

Formal module '/Simple Car Seat/Simple Car Seat' current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Automotive DOORSConnect UseCases Help

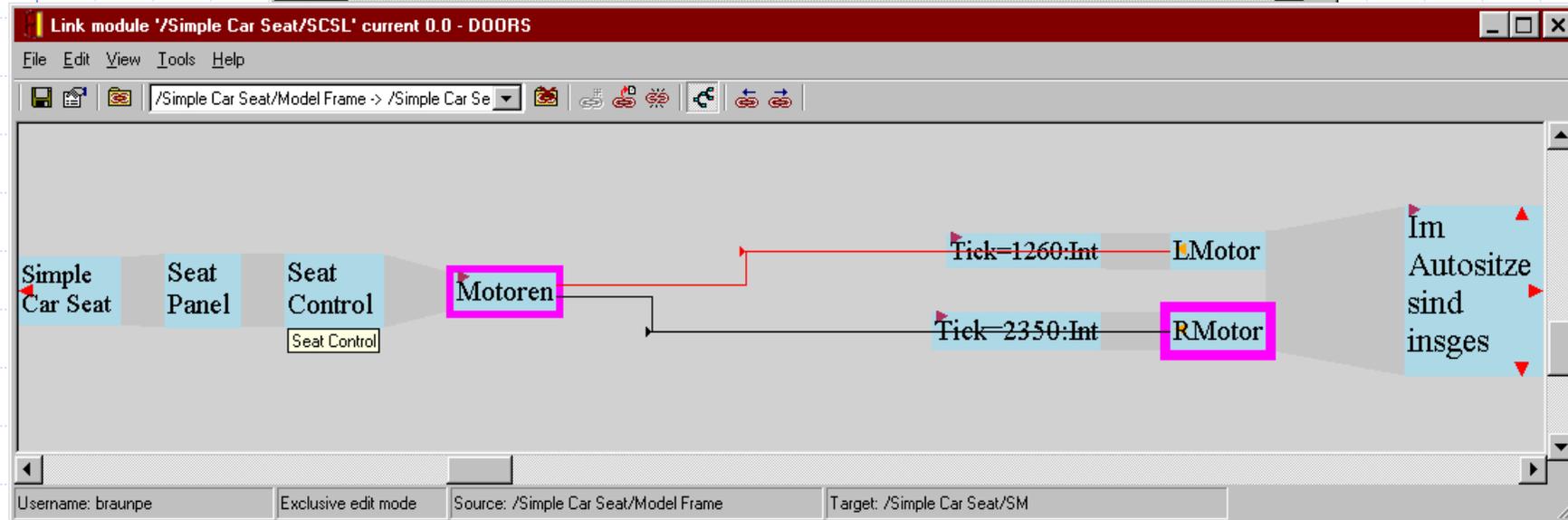
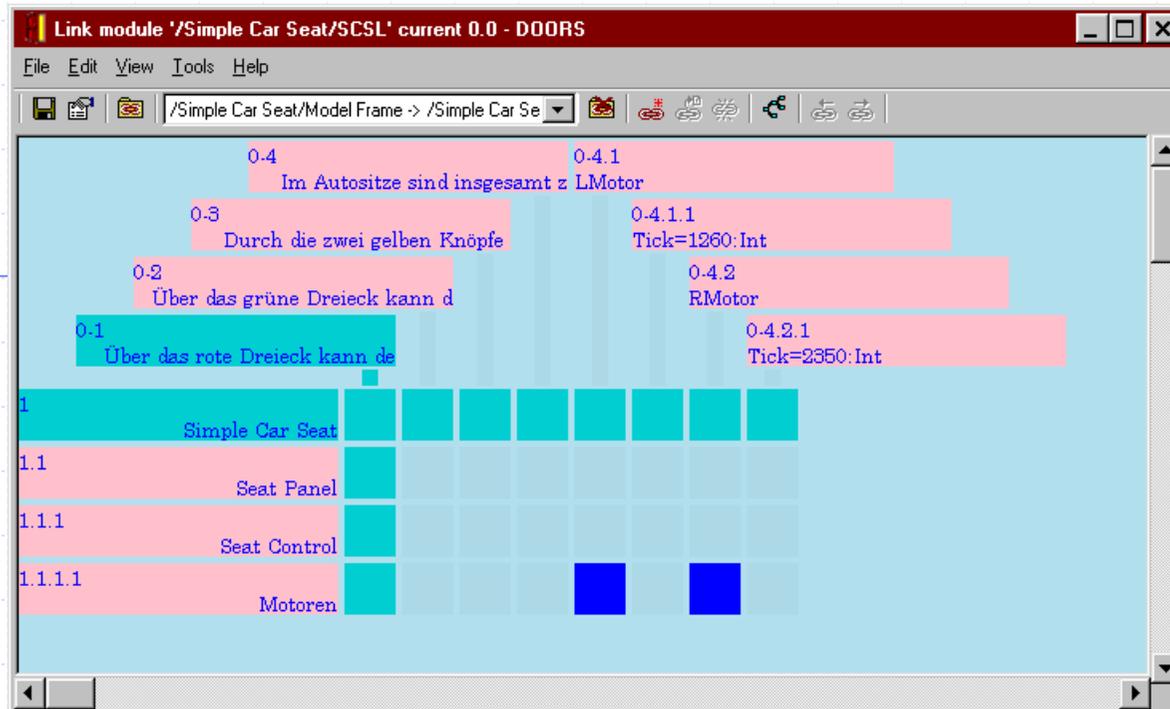
Details All levels

Simple Car Seat

- 1 Simple Seat
 - Der Autositz (Abb. 1) bittet d
 - Der Autositz zeichnet sich au
 - Der Autofahrer kann den Aut
 - Über das rote Dreieck kann
 - Über das grüne Dreieck kan
 - Durch die zwei gelben Knöpl
 - Die blauen Knöpfe (M1 und I
 - Im Autositz sind insgesamt :
 - Die Motoren sind zusätzlic
 - Der Motor LM hält beim Tick
 - 1.1 Memory Funktion

ID	Specification of Simple Car Seat	Type
UR1	1 Simple Seat	
UR2	Der Autositz (Abb. 1) bittet dem Autofahrer Gemütlichkeit und Sicherheit.	
UR3	Der Autositz zeichnet sich aus durch die stabile Formschale, hohem Fahrkomfort und durch die gute Optik. Einen hohem Komfort bietet die hochwertige Polsterung und die starke Seitenführung. Praktisch ist das beliebige Verstellen der Lehne und die Speicherung der Positionen.	
Abbildung 1: Simpel Car Seat		
		
UR7	Der Autofahrer kann den Autositz nach seinen Wünschen verstellen. Der Autositz kann vorwärts oder rückwärts bewegt werden. Die Rückenlehne kann dementsprechend nach vorne oder nach hinten bewegt werden. Um den Autositz auf eine gewünschte Position zu verstellen, ist dem Autofahrer ein Panel (Abbildung 2) angeboten.	HW SW

Username: braunpe Exclusive edit mode



DOORS - Goodies

- ◆ Copy-and-Link
- ◆ DXL
- ◆ Verteilte Bearbeitung
- ◆ Web-Interface
- ◆ Kopplung an andere Werkzeuge möglich

The UML Suite

- ◆ UML 1.3 „konformes“ Werkzeug
- ◆ Multi-User, Multi-Plattform (Datenbank!)
- ◆ Versionsmanagement
- ◆ Stark konfigurierbar (tcl)
 - XML/XMI Im-/Export
 - Codegeneratoren
- ◆ Round-Trip

UML Monitoring Window - C:\Programme\TelelogicTau40\Uml\bin\otsh.exe

File Edit View Options Help

Starting extended checking.
 Loading model for specified classes
 Loading class 'Control' ...
 Loading class 'Emergency' ...
 Loading class 'Activation' ...

Preparing the local model for checking
 Checking the local model.
 Checking class 'Control' ...
 Checking class 'Emergency' ...
 Checking class 'Activation' ...

Errors : 0
 Warnings: 0

Otsh V4.5 finished

Command Ready

Module Availability Editor

File Edit View Options Check Filter Help

modules.modules (user)

Long Name	Type	Select State	Level
Telelogic Tau UML	Product	on	corpo
Import/Export from/to Tcl	ImportExport	on	user
Lib requirement engineering	ReqEngLibrary	on	user
DOORS integration	RequirementEng	on	user
Import/Export from/to XMI	ImportExport	on	user
Export to XMI	ImportExport	on	user

Done

comotive:1/Analysis/I:

Name	Ver:
CDM Activation	Fal
CDM Control	Fal
CDM Emergency	Fal
DD Fensterhebung	Fal

corporate

- Automotive-Metamodel
- BMW-Karosserieelektronik
- Fallstudie-Automotive
 - Analysis
 - Funktionen
 - Interfaces
 - ObjectDesign
- BodyCarElectronics

Andere UML-Werkzeuge

- ◆ Argo UML (frei)
- ◆ Rhapsody (Simulation)
- ◆ Rational Rose (Marktführer)
- ◆ STP (Codegenerator)
- ◆ Together (angenehm)

ASCET-SD

- ◆ Eingebettete System
- ◆ Codegenerator für unterschiedl. Chips
- ◆ Simulation (online/offline)
- ◆ „Objektorientiert“
- ◆ C-Code nah

Database Browser : <tutorial>

Database Item View Options Browse Project Document Compare Database

1 Items

- ETAS-IconLib
- ETAS-SystemLib
- ETAS_Tutorial_Solution
 - Lesson1_2
 - Lesson10
 - EO1_TipEx
 - HO1_TipEx
 - KM_ExpEnv
 - KM_TipEx
 - MO1_DataGenerator
 - MO1_LowPass**
 - PO1_Project
 - SMD1_DataGenerator
 - Lesson3
 - Lesson4
 - Lesson5

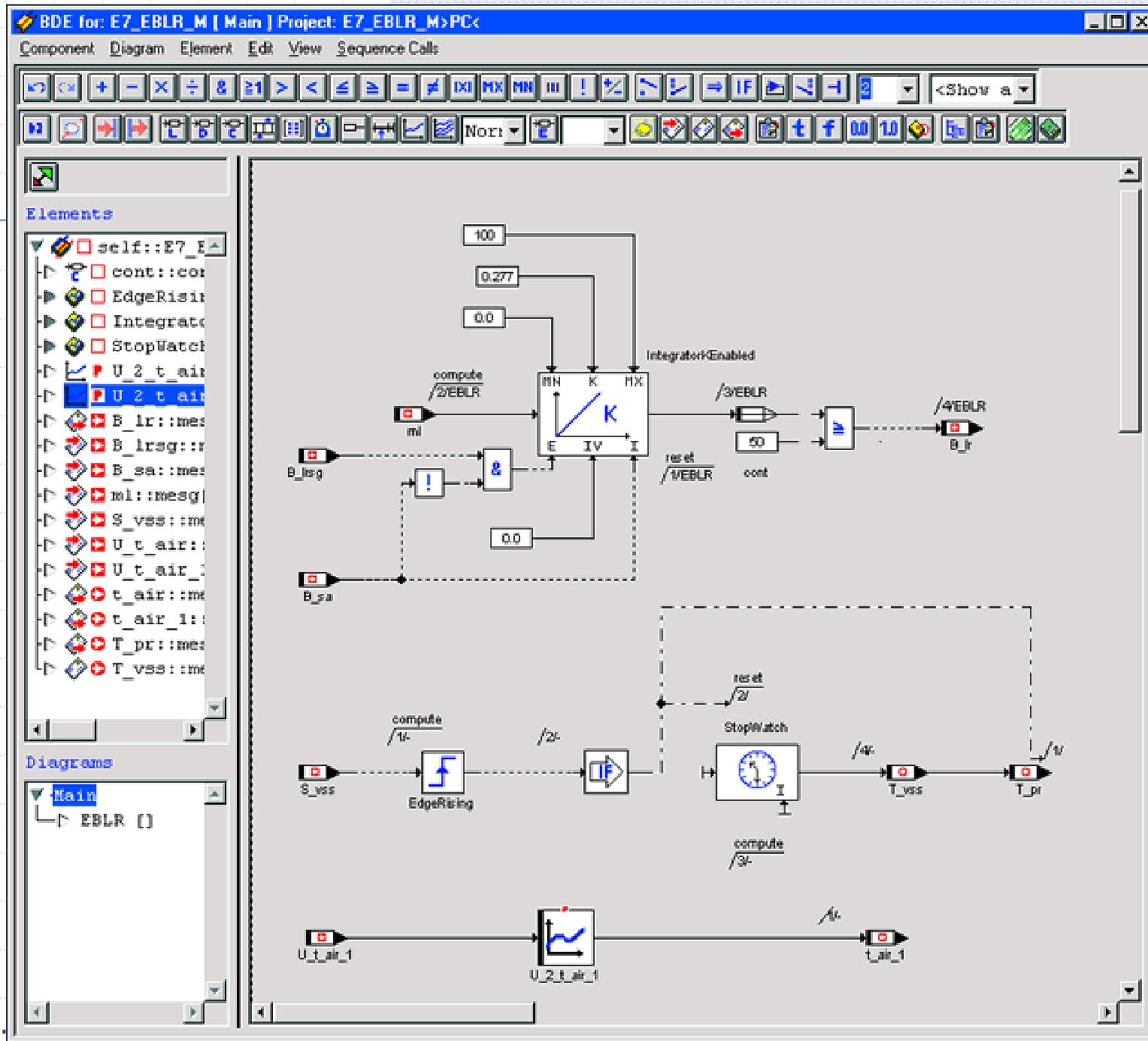
28.01.00 13:00:40 (RW)

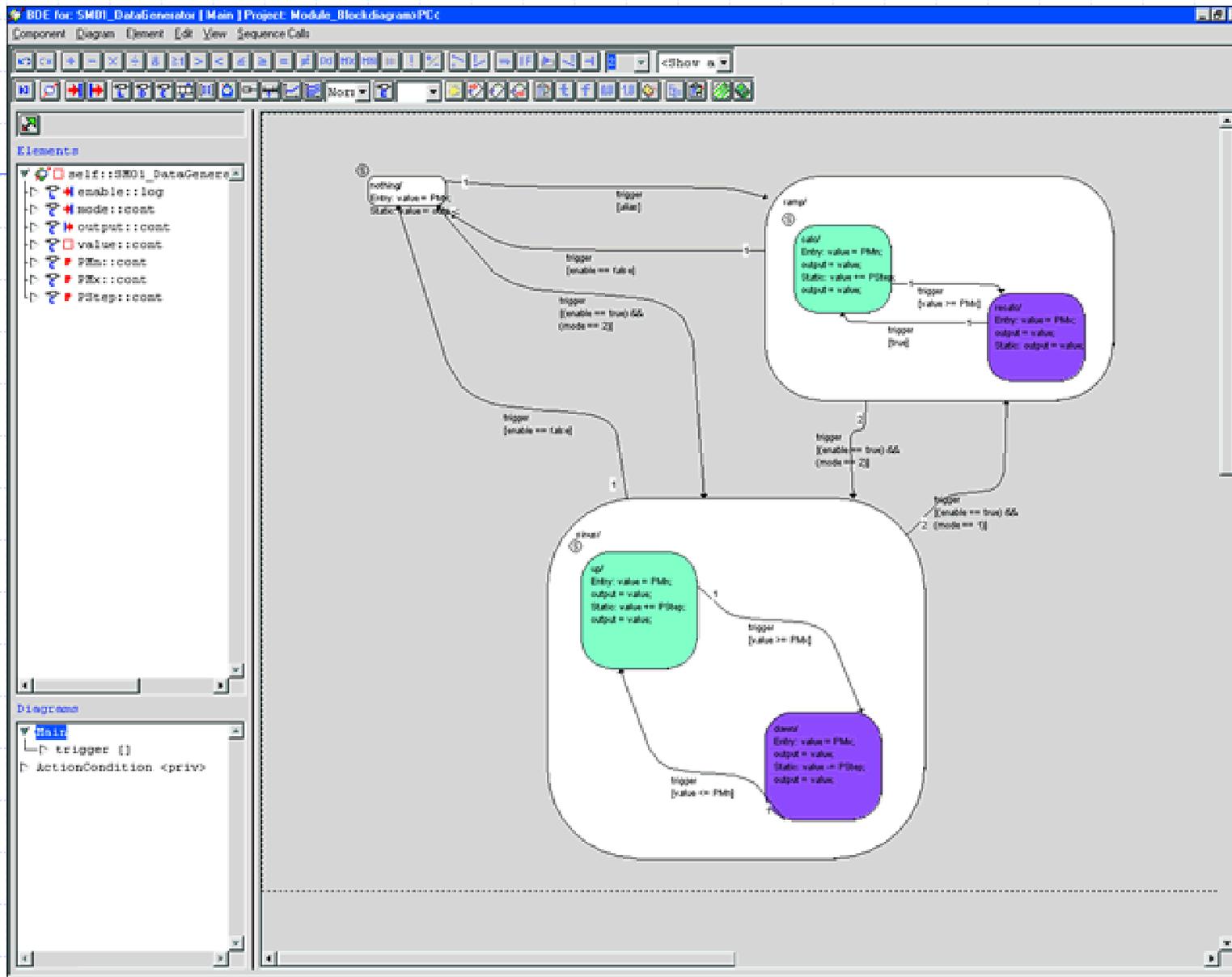
3 Item comment

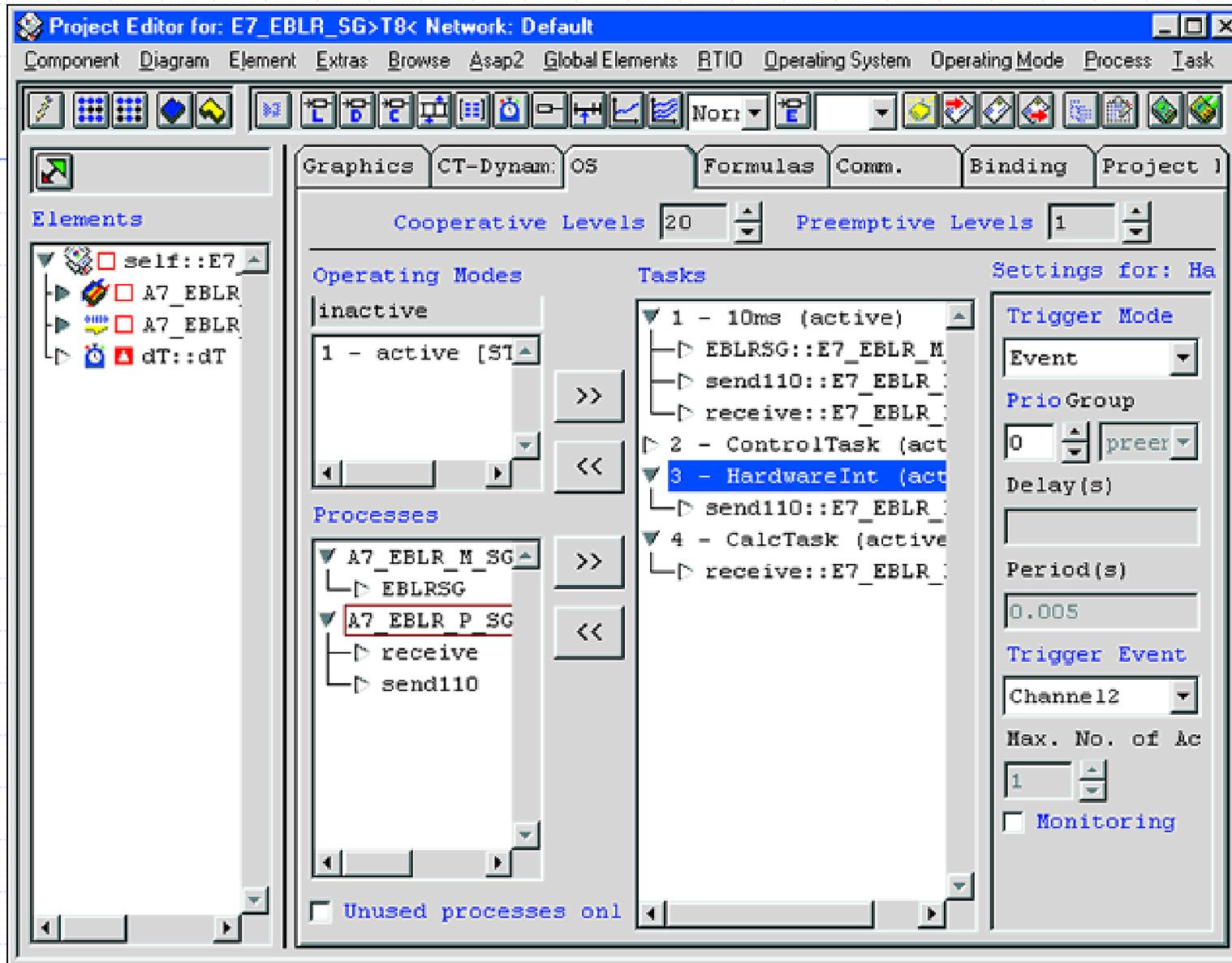
4 Info

Elements Data Implementation Layout

Name	Type	Scope	Kind	Unit
air_low	cont	local	Parameter	
air_nominal	mesg[cont]	exported	Send Message	
buffer	cont	local	Variable	
dT	dT	imported	Variable	
enable	mesg[log]	imported	Receive Message	
in	mesg[cont]	imported	Receive Message	
Integrator	Integrator	local	Variable	
Ki	cont	local	Parameter	
Kp	cont	local	Parameter	
LowpassKEnable	LowpassKE	local	Variable	
LP_IV	cont	local	Parameter	
LP_K	cont	local	Parameter	
n	mesg[cont]	imported	Receive Message	
ndiff	cont	local	Variable	
out	mesg[cont]	exported	Send Message	
pi_value	cont	local	Variable	
reset	mesg[log]	imported	Receive Message	
t	mesg[cont]	imported	Receive Message	
WarmUp	WarmUp	local	Variable	







Experiment: >E01_TipExt Hardware: >H01_TipExt Working base: >M01_Project_WB1C

File Variables Hardware Measurement GUI Window ASAP3 Window Help

E=Target: 1

UP: WORK_PAGE RP: NO_REF_PAGE

demo2: 1

enable
reset
nameText
nameText

in: 76.000
in: 76.000

out: 50.732
out: 50.7

Measure channels

Name	Value
in	90.00000
out	51.88035

Bit channels

Name	Value
1 enable	1
2 reset	0

Calibration window (1)

UP_IV_M01_Low	0.000
UP_IV_M01_High	0.000
Pin_M01_out	0.000
Pin_M01_in	100.000
Pin_M01_D	2.000
variableM01_D	True
Pin_M01_Du	1.000
Pin_M01_Dat	0.000
UP_IV_M01_Low	100

Diagram

process (1)

Max. Buffer Level: 04

VISUALIZATION ON / RECORDING PAUSED

MetaCASE Tools

- ◆ Anwendungsentwickler verwenden eh eigene Methode => generelles Tool zur Unterstützung „beliebiger“ Methoden
- ◆ Meist graphisches „Editorframework“ mit Anbindung an ein textuelles Format

Prozessorientierte SWE (1)

◆ Ziele:

- Einhaltung von Richtlinien
- Dokumentation von Entwicklungsprozess
- Kontrolle des Erfolgs von Vorgehensweisen
- Kommunikation vereinfachen

Prozessorientierte SWE (2)

◆ Schwierigkeiten/Probleme

- Prozessmodelle werden „stur“ eingehalten
- Berge nutzloser Dokumentation
- Entwickler fühlen sich überwacht
- Spontane Kommunikation

Formalismen zur Definition

- ◆ Regelbasiert [Marvel, ...]
- ◆ Netzbasiert [Process Weaver, ...]
- ◆ Programmbasiert
- ◆ ...

Alles aktuelles Forschungsgebiet

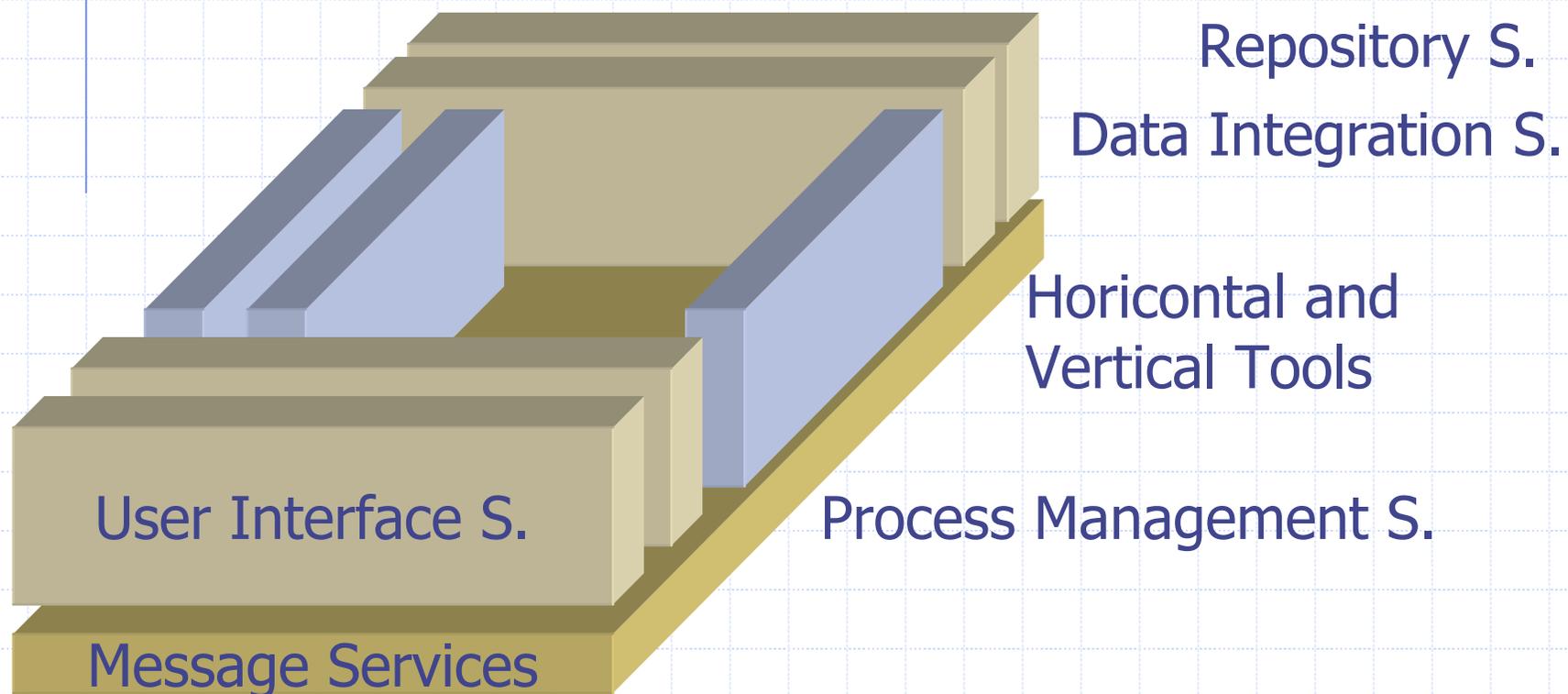
Werkzeugintegration

◆ Nutzen und Probleme:

- Normierung von Benutzeroberflächen
- Datenspeicherung und Datenformate
- Semantik für Daten -> Standards?
- Portabilität
- Kontrolle über „Werkzeugaktivierungen“

Ansätze: ECMA

◆ ECMA-Referenzmodell = Toastermodell



Ansätze: PCTE

◆ Portable Common Tool Environment:

- Unix-Aufsatz
- Objektmanagement (Beziehungen auf Dateien)
- DB-Funktionalitäten
- Prozessverwaltung
- ...

Zukunft

- ◆ Repositories (z.B. Microsoft)
- ◆ OODBs (ODMG)
- ◆ DCE, Corba, DCOM, ...