

# 20. Architektur von Software- Werkzeugen und -Entwicklungsumgebungen

Prof. Dr. Uwe Aßmann

Technische Universität Dresden  
Institut für Software- und  
Multimediatechnik

<http://st.inf.tu-dresden.de>  
Version 11-1.1, 10.11.11

- 1) Grobarchitektur von Werkzeugen
- 2) Werkzeugintegration
  - 1) Datenintegration
- 3) Architektur von SEU
- 4) ECMA-Referenzmodell



SEW, © Prof. Uwe Aßmann

4

## Referenzen

- ▶ ECMA, Reference Model for Frameworks of Software Engineering Environments, Technical Report 55, 3rd Edition, Juni 1993
  - <http://www.ecma-international.org/publications/files/ECMA-TR/TR-055.pdf>
- ▶ Richard C. Holt, Andreas Schürr, Susan Elliot Sim, and Andreas Winter. GXL: A graph-based standard exchange format for reengineering. Science of Computer Programming, 60(2):149-170, April 2006.
  - <http://www.gupro.de/GXL/Publications/publications.html>



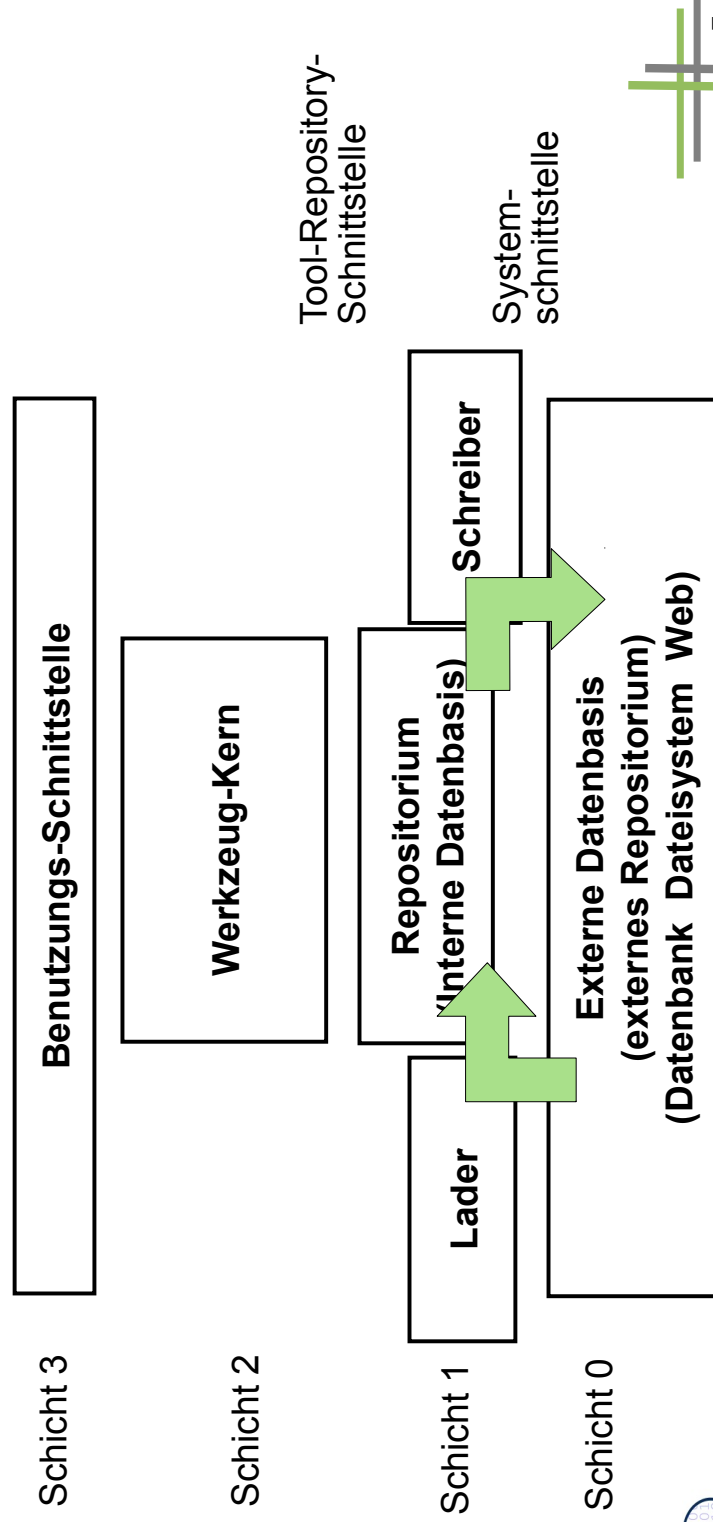
# 20.1 Grobarchitektur von Werkzeugen



SEW, © Prof. Uwe Aßmann

6

## Architektur eines repository-basierten Werkzeugs

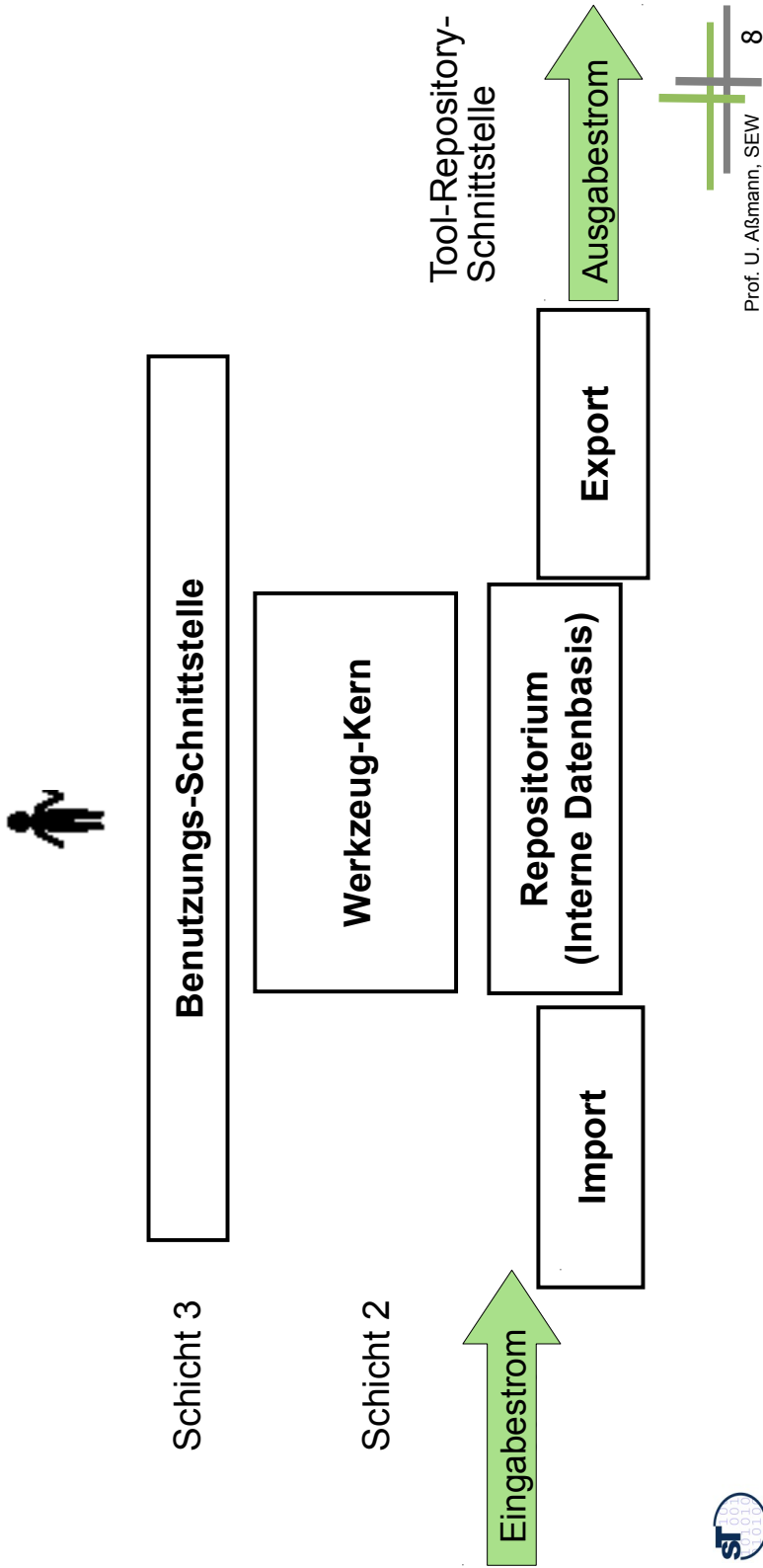


Prof. U. Aßmann, SEW

7

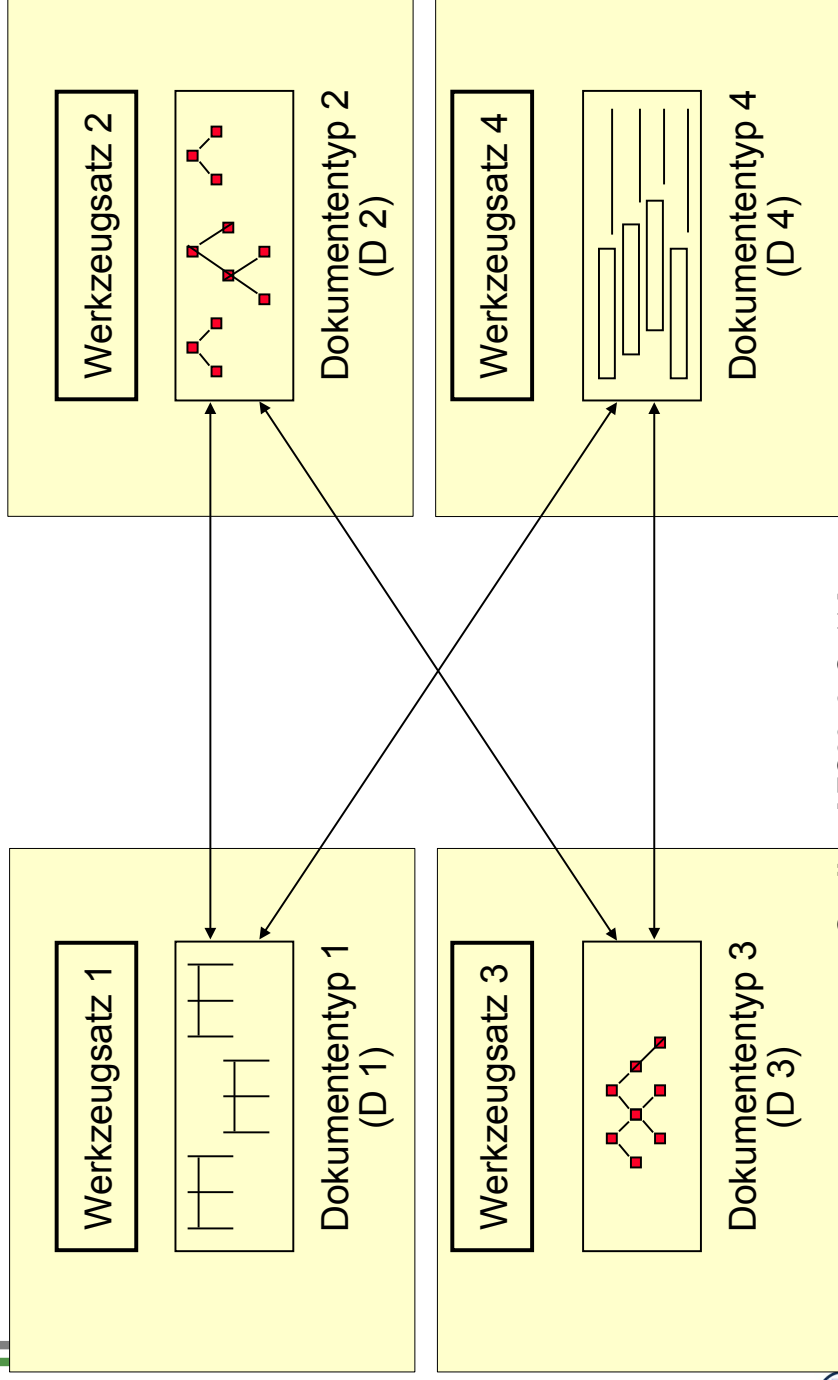
## Architektur eines datenflussgesteuerten, strom- basierten Werkzeugs

- ▶ Arbeit wird stückweise erledigt; meist pro gelesenem Datenpaket



## 20.2 Konzepte der Werkzeug- integration

## Integrierte Werkzeugsätze zur Softwareentwicklung



Quelle : [ ES89, 6, S. 11 ]

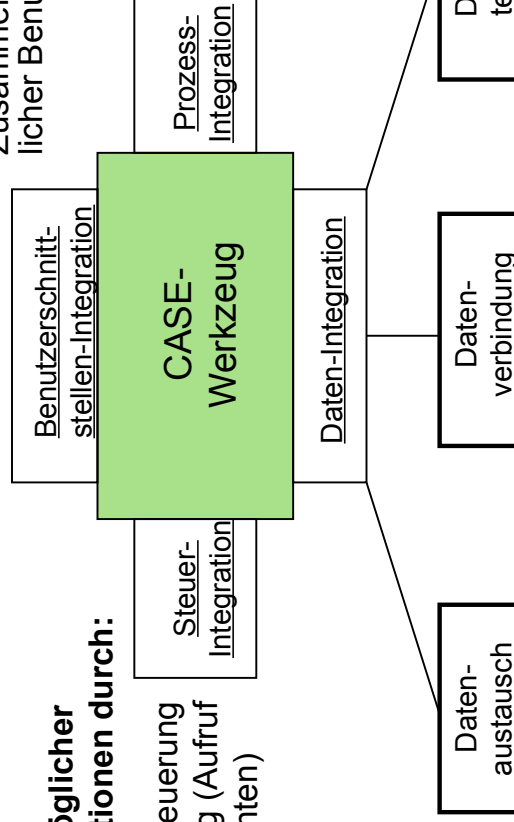
## Werkzeugintegration

### Zusammenarbeit mehrerer Werkzeuge mit gleicher Präsentation, gleichem Verhalten und Interaktionsformen

Zusammenarbeit unter einheitlicher Benutzungsoberfläche

Verbindung möglicher Werkzeugfunktionen durch:

- Interprozesssteuerung
- Metasteuerung (Aufruf über Metaagenten)



Werkzeugintegration zur Unterstützung eines definierten Prozesses/ Vorgehens bestehend aus mehreren Schritten

**Forderungen:** • Interoperabilität mit Formatwandlung

- Redundanzfreiheit - Einmalstorage
- Datenkonsistenz - Persistenz
- Synchronisation - Abstimmung des Werkzeugzugriffs

## 20.2.1 Datenintegration

### Werkzeuge mit Datenaustausch (ad-hoc), ohne Datenverbindung und -integration

- ▶ Keine gemeinsamen Daten, hoher (manueller) Aufwand zum Austausch
- ▶ Austausch mit Datenfluss-Strömen (Datenflussarchitektur)
  - Querysprachen filtern den Datenaustausch
  - Datenformate werden in einer Austauschsprache definiert
- ▶ Aber: unabhängiges, paralleles Arbeiten möglich

WZ1



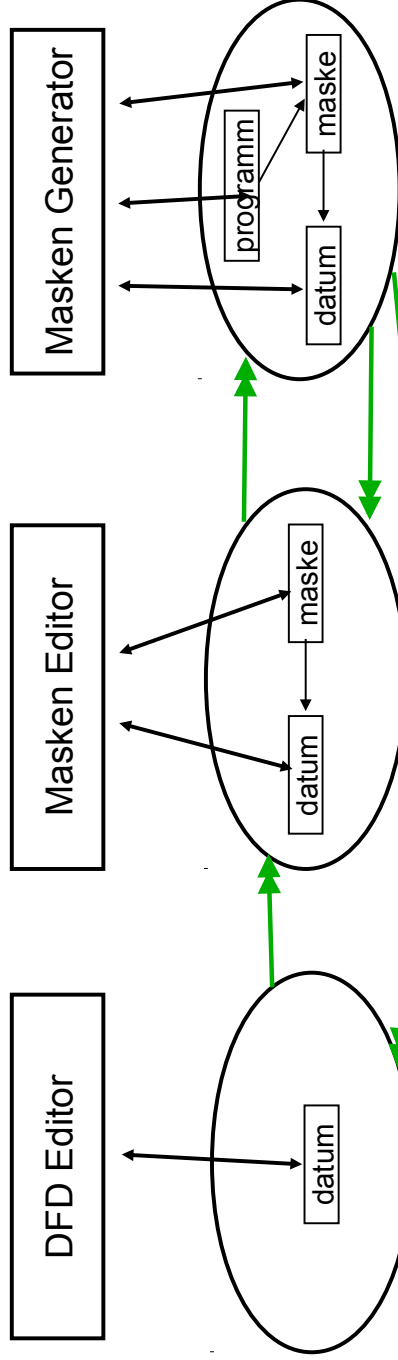
Datenbasis 1

WZ2



Datenbasis 2

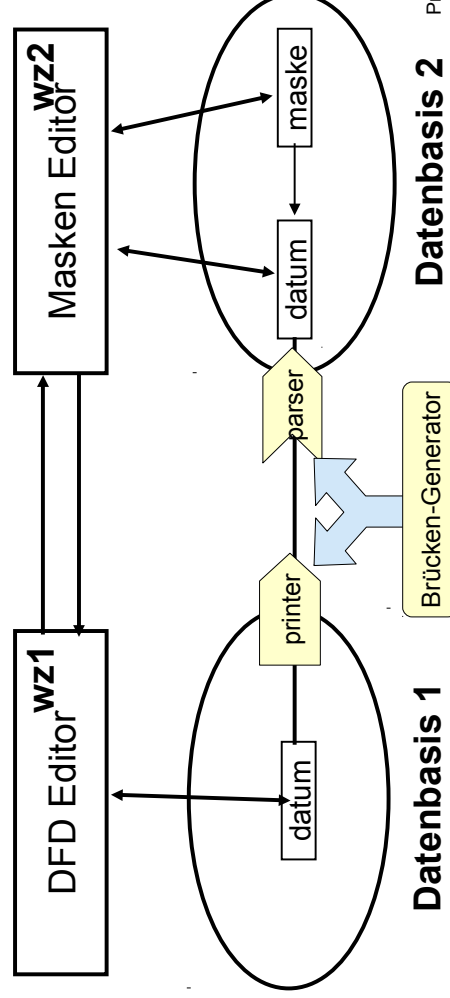
WZ3



Datenbasis 3

# Datenverbindung durch Datenaustausch (Transformationsbrücken)

- ▶ **Datenverbindung** ist die Einführung semantischer Beziehungen zwischen Teilen von Datenbeständen
- ▶ **Automatisierter Datenaustausch** ist die automatisierte Übertragung von *semantisch verbundenen Daten* zwischen Werkzeugen in standardisierten **Austauschformaten** (z. B. ASN, XMI, CDIF, XML)
  - Automatisierung beruht auf Metamodellen
- ▶ **Transformationsbrücke:** Prettyprinter transformieren das interne Format eines Repositoriums in ein externes
  - Parser wandeln es in das interne Format des anderen Werkzeugs
  - Querysprachen filtern den Datenaustausch

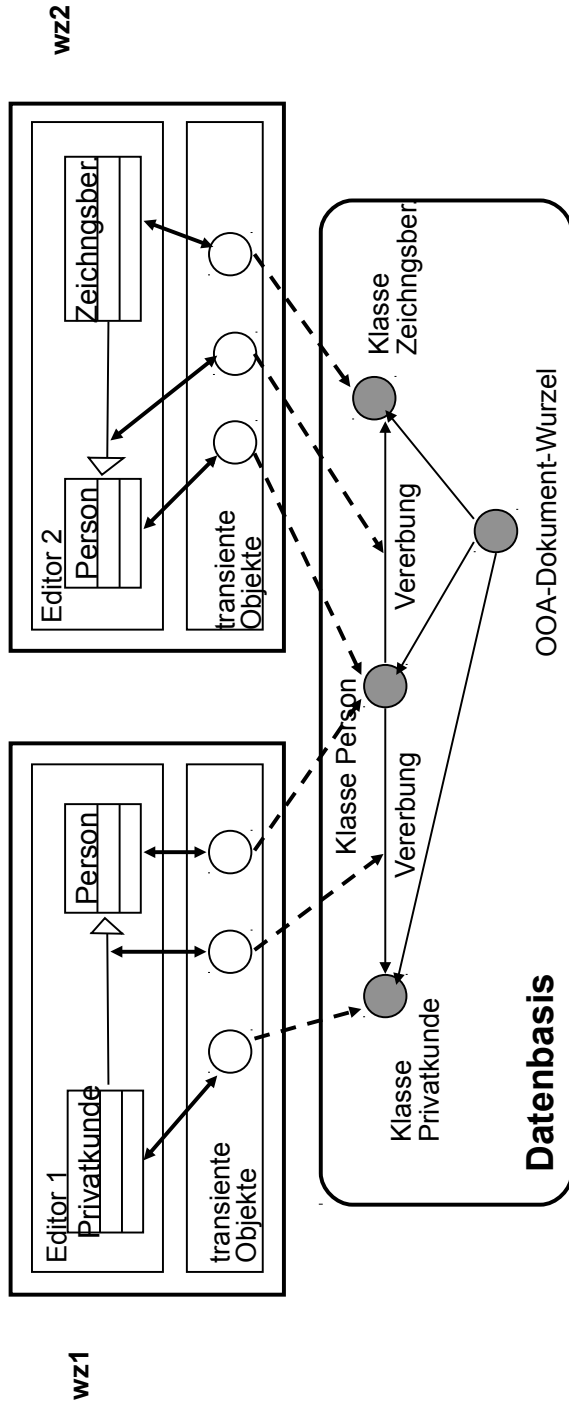


## CASE Austauschformate

- ▶ **Austauschformat:** Hersteller- und methodenunabhängiges Datenkonzept für die Modellierung von Austauschdaten zwischen Werkzeugen
- ▶ **Comma-separated values (CSV):** einfaches text-basiertes Austauschformat für Werkzeuge auf Relationen und Tabellen (Excel, TeX, ...)
  - Keine Metasprache, einfaches Tabellenschema
  - [http://en.wikipedia.org/wiki/Comma-separated\\_values](http://en.wikipedia.org/wiki/Comma-separated_values)
- ▶ **CASE Tool Data Interchange Format (CDIF) - Metasprache ERD für Data Definition, aber auch auf Relationen und Tabellen**
  - Data Flow Model, State Event Model, Object Oriented Analysis and Design
  - <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-270.pdf>
- ▶ **XML Metadata Interchange (XMI) zum Austausch von UML-Diagrammen im XML-Format**
  - Meta Object Facility (MOF) als Metasprache
  - <http://www.omg.com/technology/documents/formal/xmi.htm>
- ▶ **ASN.1 Standard**
  - Eigene Metasprache, an BNF angelehnt
  - [http://de.wikipedia.org/wiki/Abstract\\_Syntax\\_Notation\\_One](http://de.wikipedia.org/wiki/Abstract_Syntax_Notation_One)
- ▶ **RDF/RDFS Resource Description Format - Modelle als Graphen, gespeichert in elementaren Tripeln** <http://www.w3c.org>
- ▶ **GXL Graph Austauschformat**

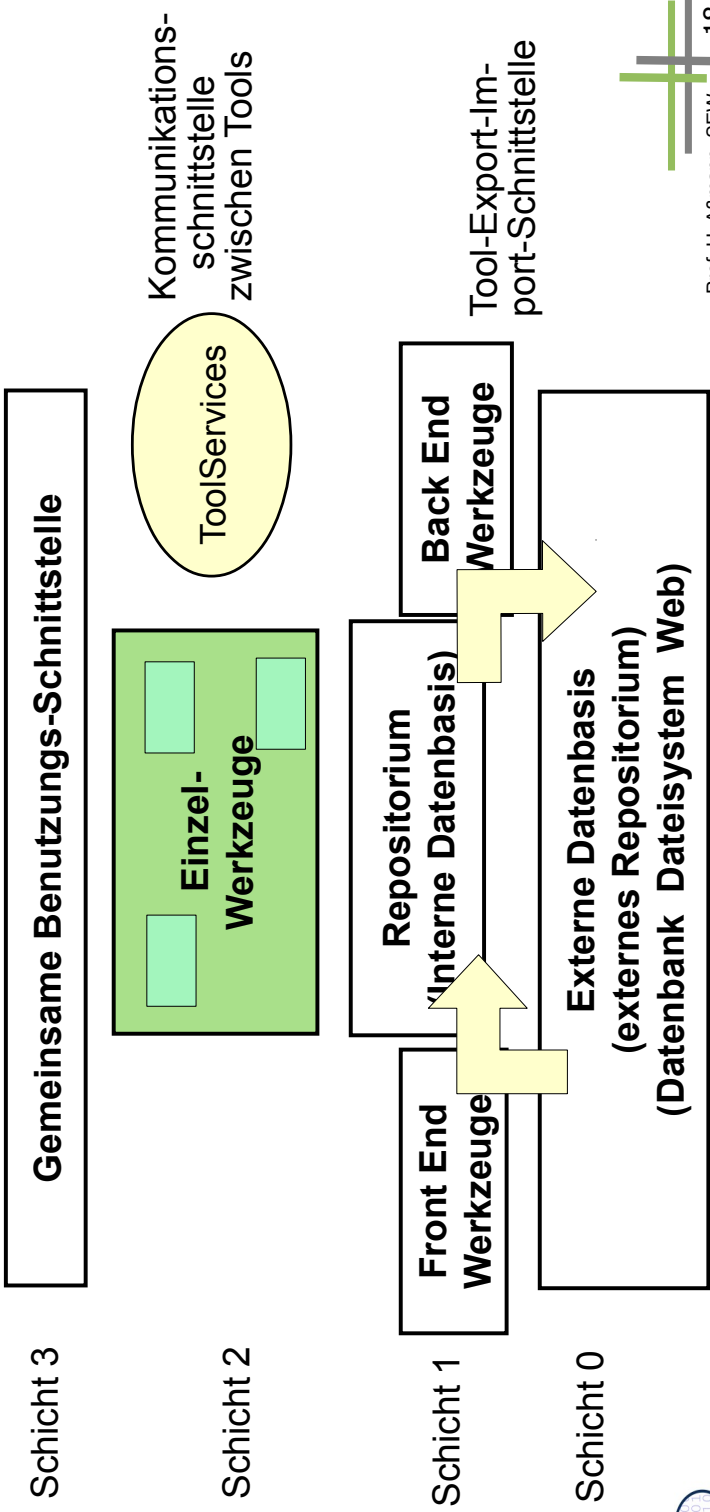
## Datenteilung (Gemeinsames Repository)

- ▶ Werkzeuge greifen direkt auf Daten zu, die in einem gemeinsam verfügbaren Datenbasis mit einheitlichen Schema abgelegt sind.
- ▶ **Repository (Repository, Datenbasis, Objektbank, Artefaktbank):** Datenverwaltungssystem zur Speicherung aller Daten- und Projektinformationen für Werkzeuge



## 20.3 Architektur von SEU

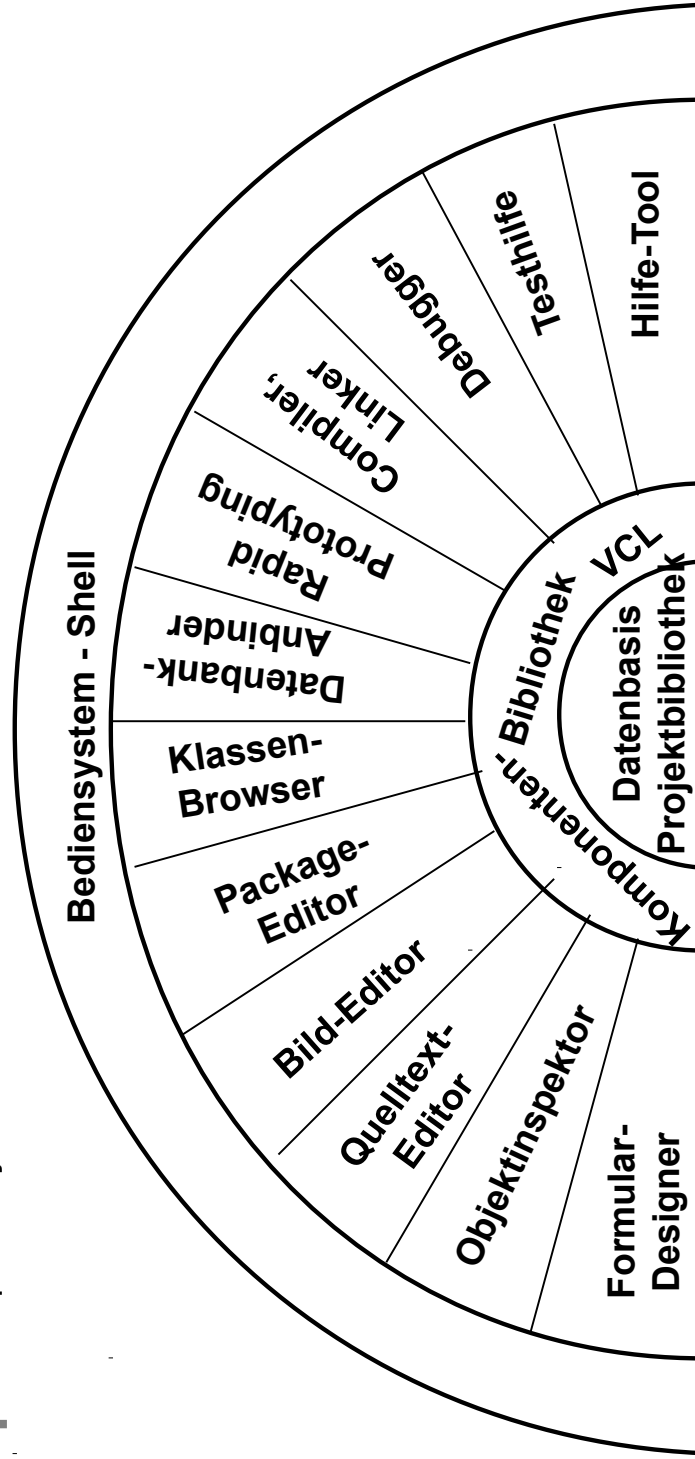
# Architektur einer SEU



Prof. U. Aßmann, SEW 18

# Beispiel: Das Schalenmodell von Delphi

▶ Repository-basiert



9



## 20.4 Beispiel einer Referenzarchitektur für Werkzeug-Umgebungen: Das ECMA Referenzmodell für SEU

.. Der ECMA-Toaster



SEW, © Prof. Uwe Aßmann

20

### Standardisierungsorganisation European Computer Manufacturing Association (ECMA)

- ▶ Weltweite Normierung der Informationstechnologie und Nachrichtentechnik
  - Mehr als 365 ECMA-Standards
  - 2/3 sind als internationale Standards und/oder technische Reports angenommen worden.
- ▶ Ziele:
  - Zusammenarbeit mit nationalen, europäischen und internationalen Normierungsorganisationen über die Standardisierung von Kommunikationstechnologien (ICT) und Verbraucherelektronik (CER).
  - korrekten Gebrauch von Standards anregen und kontrollieren.
  - Veröffentlichung von Standards und technischer Reports, Unterstützung ihrer Verbreitung auch in elektronischer Form



u. a. folgende Technischen Ausschüsse:

TC 32: Kommunikation, Netze und Systemverbindungen  
TC 39: Programmieren und Script-Sprachen  
TC 43: Universal 3D (U3D)  
TC 12: Sicherheit



Quelle: <http://www.ecma-international.org/>



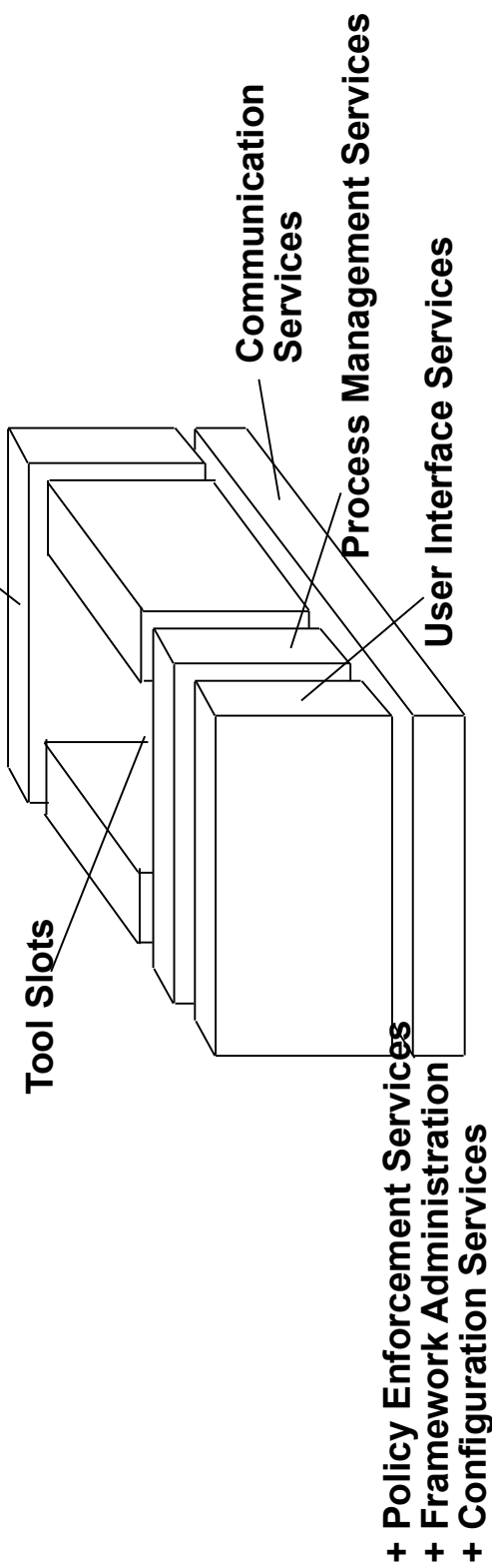
Prof. U. Aßmann, SEW

21

## ECMA-Referenzmodell („ECMA Toaster“)

- ▶ nutzt eine Service-orientierte Architektur (SOA), kann also verteilt sein
- ▶ Diese Dienste sind mehr oder weniger in jeder SEU vorhanden

### Object Management Services (Repository)



- + Policy Enforcement Services
- + Framework Administration
- + Configuration Services

Quelle: ECMA, Reference Model for Frameworks of Software Engineering Environments,

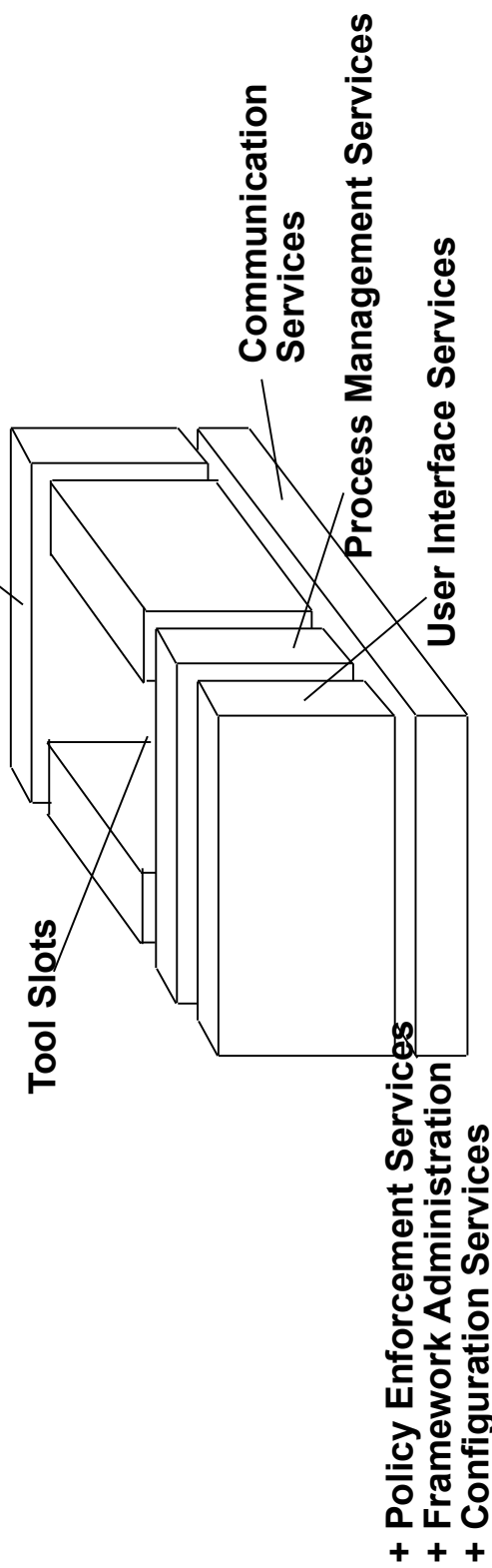
Technical Report 55, 3rd Edition, Juni 1993

<http://www.ecma-international.org/publications/files/ECMA-TR/TR-055.pdf>

## ECMA-Referenzmodell („ECMA Toaster“)

- ▶ nutzt eine Service-orientierte Architektur (SOA), kann also verteilt sein
- ▶ Diese Dienste sind mehr oder weniger in jeder SEU vorhanden

### Object Management Services (Repository)



- + Policy Enforcement Services
- + Framework Administration
- + Configuration Services

Quelle: ECMA, Reference Model for Frameworks of Software Engineering Environments,

Technical Report 55, 3rd Edition, Juni 1993

<http://www.ecma-international.org/publications/files/ECMA-TR/TR-055.pdf>

<b>externe Sicht</b>	beschreibt die externe Nutzung des Dienstes durch andere Dienste bzw. durch Werkzeuge od. den Nutzer
<b>konzeptionelle Sicht</b>	beschreibt die Semantik (Funktionalität), ohne Implementierung oder Verfügbarkeit für den Nutzer zu beachten
<b>interne Sicht</b>	beschreibt die spezifische Implementierung (Betriebssystem, andere Tools) für die Dienstausführung
<b>Sicht auf Operationen</b>	führt die Menge von Operationen eines Dienstes auf, die zur Erreichung der Funktionalität (konzeptionelle Sicht) benötigt wird
<b>Sicht auf Typen</b>	beschreibt das Datenmodell des Dienstes einschließlich der Informationen über dieses Datenmodell (Metamodell)
<b>Sicht auf Regeln</b>	beschreibt Regelmenge, die mögliche Menge der Operationen (Sicht auf Operat.) und annehmbare Zustände der Daten definiert
<b>Sicht auf Dienst-zu-Dienst-Beziehungen</b>	anhand typischer Beispiele wird gezeigt, wie ein Dienst mit einem anderen kommunizieren kann

## ECMA Benutzungsschnittstelle USER INTERFACE SERVICES

ECMA stellt eine Reihe von UI-Diensten (services, Schnittstellen) zur Verfügung, die zur Gewährleistung der Benutzungsschnittstellen-Integration und der konsistenten Bedienung von Anwendungen benötigt werden.

- ▶ **User Interface Metadata Service** dient der Definition, Steuerung und Handhabung von Schemata zur Unterstützung der Benutzungsschnittstelle
- ▶ **Session Service** gewährleistet volle Funktion, unabhängig von Nutzer oder Hardwareumgebung
- ▶ **Security Service** gewährleistet Sicherheitsanforderungen, wie Nutzerauthentifikation, Dunkelsteuerung unbenutzbarer Funktionen u. a.
- ▶ **Profile Service** gestattet mögliche Veränderungen, wie z. B. Systemeinstellungen (Farbe), Menge zu verwendender Werkzeuge u.a.
- ▶ **User Interface Name and Location Service** stellt fest, wer sich wo zum System Zutritt verschafft hat (logging in)
- ▶ **Internationalization Service** stellt nationale Besonderheiten (z. B. Zeichensätze, Datumsformate) zum Zugriff auf das Rechnersystem bereit und gewährleistet ihre Konvertierbarkeit zwischen unterschiedlichen Ländern.

# ECMA Prozessverwaltung PROCESS MANAGEMENT SERVICES

Definieren und organisieren die Ausführung aller Werkzeuge und Aktivitäten für den Softwareentwicklungsprozess von den rein technischen Maßnahmen bis hin zur Steuerung betriebswirtschaftlicher Aspekte:

- ▶ **Process Definition Service** definiert aus den im Repository gespeicherten Projektdaten die Bedingungen zur Ausführung neuer Aktivitäten
- ▶ **Process Control Service** steuert Prozesse im allgemeinen auf dem Niveau eines bestimmten Vorgehensmodells zur Beeinflussung anderer Prozesse, wodurch der Nutzer entsprechend seiner Rolle unterstützt wird
- ▶ **Process Enactment Service** unterstützt und bietet Möglichkeiten der Steuerung vorher definierter Aktivitäten (Analyse-, Hilfe-, Simulationsfunktionen)
- ▶ **Process Visibility and Scoping Service** legt zum Zwecke der Kommunikation und Koordination Sichtbarkeit, Zeitpunkt und Ort von Aktivitätsteilen für andere Aktivitäten fest
- ▶ **Process State Service** sammelt und wertet Ereignisse von Aktivitäten während ihrer Ausführung aus, die für die Koordination und spätere Entscheidungsplanung anderer Projektaktivitäten notwendig sind
- ▶ **Process Resource Management Service** verwaltet das Festlegen von Ressourcen zur Ausführung definierter Prozesse für Werkzeuge und Nutzer

# ECMA Werkzeugdienste (TOOL SERVICES)

- ▶ Werkzeuge können in den ECMA Toaster eingesteckt werden.
  - Die gesamte Toolmenge soll nach außen hin durch eine **einzige Schnittstelle** repräsentiert werden.
  - Die Menge der Tools soll den **Softwareentwicklungsprozess vollständig abdecken**.
- ▶ Die Werkzeuge sind **auswechselbar** und kommunizieren über den *Communication Service* oder *Object Management Service* mit anderen Werkzeugen und Diensten der SEU.
- ▶ Wenn Werkzeuge in die SEU **integriert** werden, ist zu prüfen, ob sie Frameworkdienste bieten.
  - Wenn ja, ist zu entscheiden, diese Dienste weiterhin separat zu ermöglichen oder doch auf die Dienste des SEU-Frameworks überzugehen.
- ▶ Um für alle Werkzeuge ein **gleiches Erscheinungsbild** zu erhalten, müssen Basisdienste und Dienste eines SEU-Framework nach den standardisierten Vorschriften realisiert werden.

## ECMA Datenbasis (Repositorium) OBJECT MANAGEMENT SERVICES

dienen der Definition, der Speicherung, der Handhabung, der Handhabung, der Verwaltung und dem Zugriff auf Objekte/Dokumente (Dateien, Programme, Bibliotheken, Projekte, Geräte usw.) und den Beziehungen zwischen ihnen:

- ▶ **Metadata Service** gestattet die Definition, Steuerung und Handhabung von Schemata und sonstigen Metadaten (Reflektion, Introspektion)
- ▶ **Data Storage and Persistence Service** unterstützt das persistente Anlegen und Speichern von Objekten nach der Metadatenbeschreibung
- ▶ **Relationship Service** erlaubt die Definition und Handhabung von Beziehungen zwischen Objekten und Objekttypen.
- ▶ **Derivation Service** dient dem Baumanagement
  - Legt Wege fest, welche Objekte von anderen abgeleitet sind (z. B. Generierung Objektcode aus Quellcode ähnlich Make-Files).
  - Er schließt Mechanismen zur Vererbung von Datenbeschreibungen ein.
- ▶ **Concurrency Service** sichert den gleichzeitigen synchronisierten Zugriff für Nutzer und Prozesse zum gleichen Objekt der Datenbasis.
- ▶ **Version Service** unterstützt das Anlegen, Zugreifen und Verbinden von Objekt- und Konfigurationsversionen der SEU.

## ECMA: Weitere Services

- ▶ **POLICY ENFORCEMENT SERVICES** sind für Sicherheitsaspekte, Integritätsüberwachung und Verwaltungsfunktionen zuständig:
  - **Mandatory Confidentiality Service** legt auf eigenen Wunsch Zugriffsrechte und Sicherheitsanforderungen (geheim, str. geheim) für Objektinformationen fest.
  - **Mandatory Integrity Service** gestatten den Schutz von SEU-Objekten vor unauthorisierten Änderungen, z. B. Eintragung "read only" usw.
  - **Mandatory Conformity Service** überwacht alle Aktivitäten zur Einhaltung von Konformitätsanforderungen, die z.B. aus der Qualitätssicherung stammen.
- ▶ **COMMUNICATION SERVICES** dienen der Kommunikation zwischen Werkzeugen, zwischen Basisdiensten sowie Diensten verschiedener SEU.
  - Basismechanismen sind Nachrichten (Punkt-zu-Punkt, Broadcast, Multicast), Betriebs-systemaufrufe, Remote Procedure Calls und der Datenaustausch
- ▶ **FRAMEWORK ADMINISTRATION** und **CONFIGURATION SERVICES** übernehmen die sorgfältige Installation der SEU und ihre laufende Pflege u.a.:
  - **Tool Registration Service** übernimmt das An- und Abmelden neuer Tools.
  - **User Administration Service** unterstützt das An- und Abmelden von Nutzern zum System.



