EM/OS Architektur am 15.1.2010 bei der
Regionalgruppe Hamburg der Gesellschaft für Informatik (GI)

Modellbasierte Erzeugung ganzer Anwendungen

Andreas Leue
Sphenon GmbH, Hamburg

SPHENON
The Information Architects

Sphenon

Inhalt

Sphenon: Geschichte

2010 EM/OS 3.0 (Complete Executable Models)

2008 Release EM/OS 2.0
(incl. SirFace 3.0, SWT)
2007 Konsolidierung Desktop-GUI,
100% Java, Processor, Tooling, Cleanup)
2006 Release SirFace 2.0
(OOGenerator, State/DB etc.)
2005 Release SirFace 1.0
(ehemals VUI, mit Java OOGenerator)
2002 - Hypovereinsbank (Generator), HLG (VUI, Generator),
2007 Rekord AG (VUI), Skymaster (EM/OS 1.0),
Bosch Versicherungsportal (EM/OS 2.0)
2000 Onsecure Versicherungsportal
(vollautomatische Produktion, Architektur)
1999 Gründung Sphenon GmbH
1998 Generator in perl, XML, mächtige Templatesprache
1997 POET WebFactory "Database Publishing" & CMS
1995 Prototyp "Medienneutrale Oberfläche"
1994 Architektur-Studie "Cyberspace Architecture Project"
1993 Template-basierter Generator in C

Sphenon: Leistung

- EM/OS Enterprise Model Operation Services
- Maßgeschneiderte individuelle Applikationen
- Modelle für IT und Unternehmen (UML, BPM)
- Beratung, Schulung

SPHENON
The Information Architects

Produkt

Partner - Netzwerk

EM OS
Enterprise Model Operation Services

Frankfurt

cosmik

GWK Informationstechnologie

Berlin

ebuero AG
Das Büro der Zukunft

Hannover

ATHELAS

Hamburg

Vinion

Stuttgart

instinctools

Open Source

www.oomodels.org
Open Operational Business Knowledge

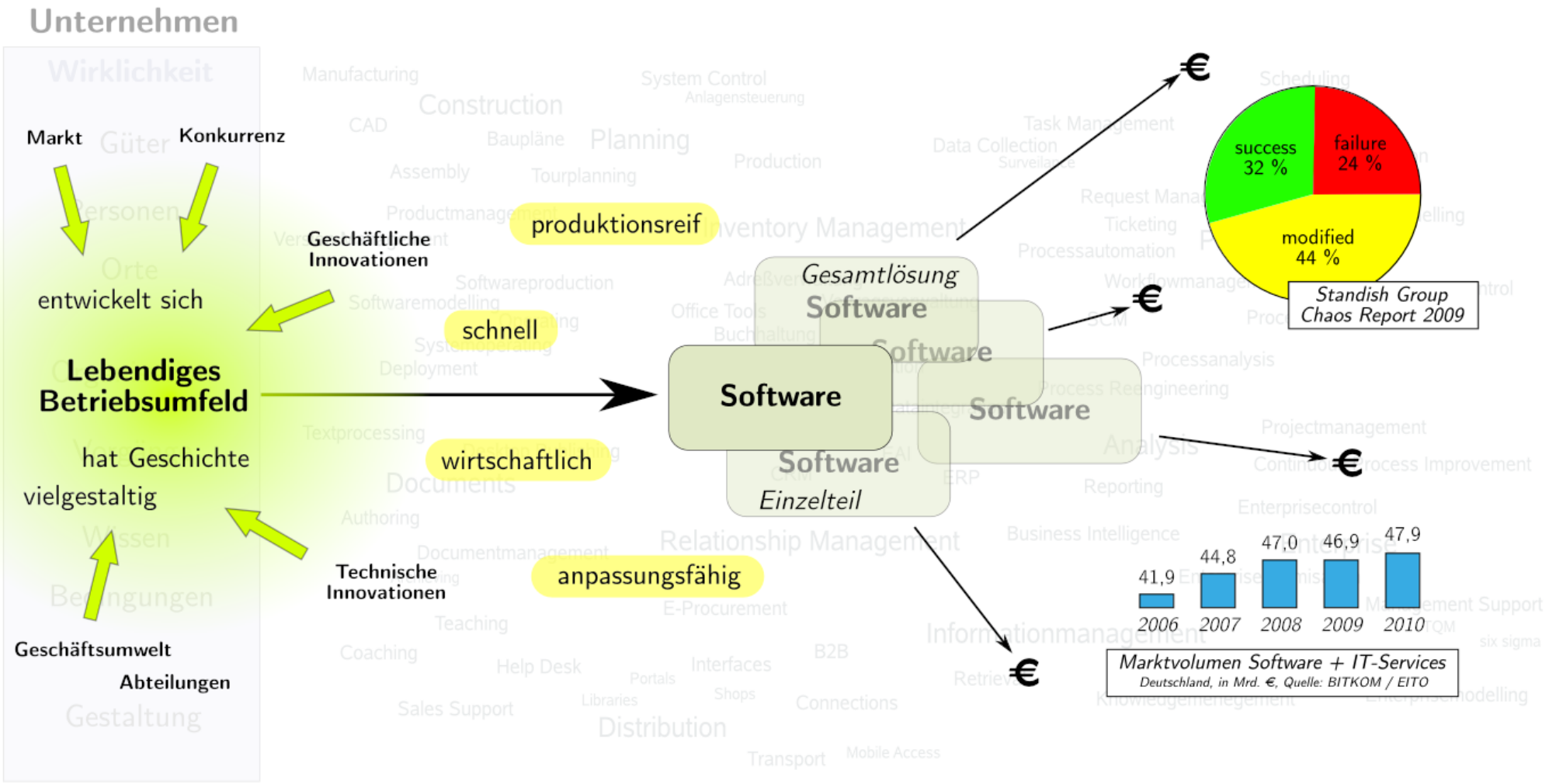
Generator

Object Assembler

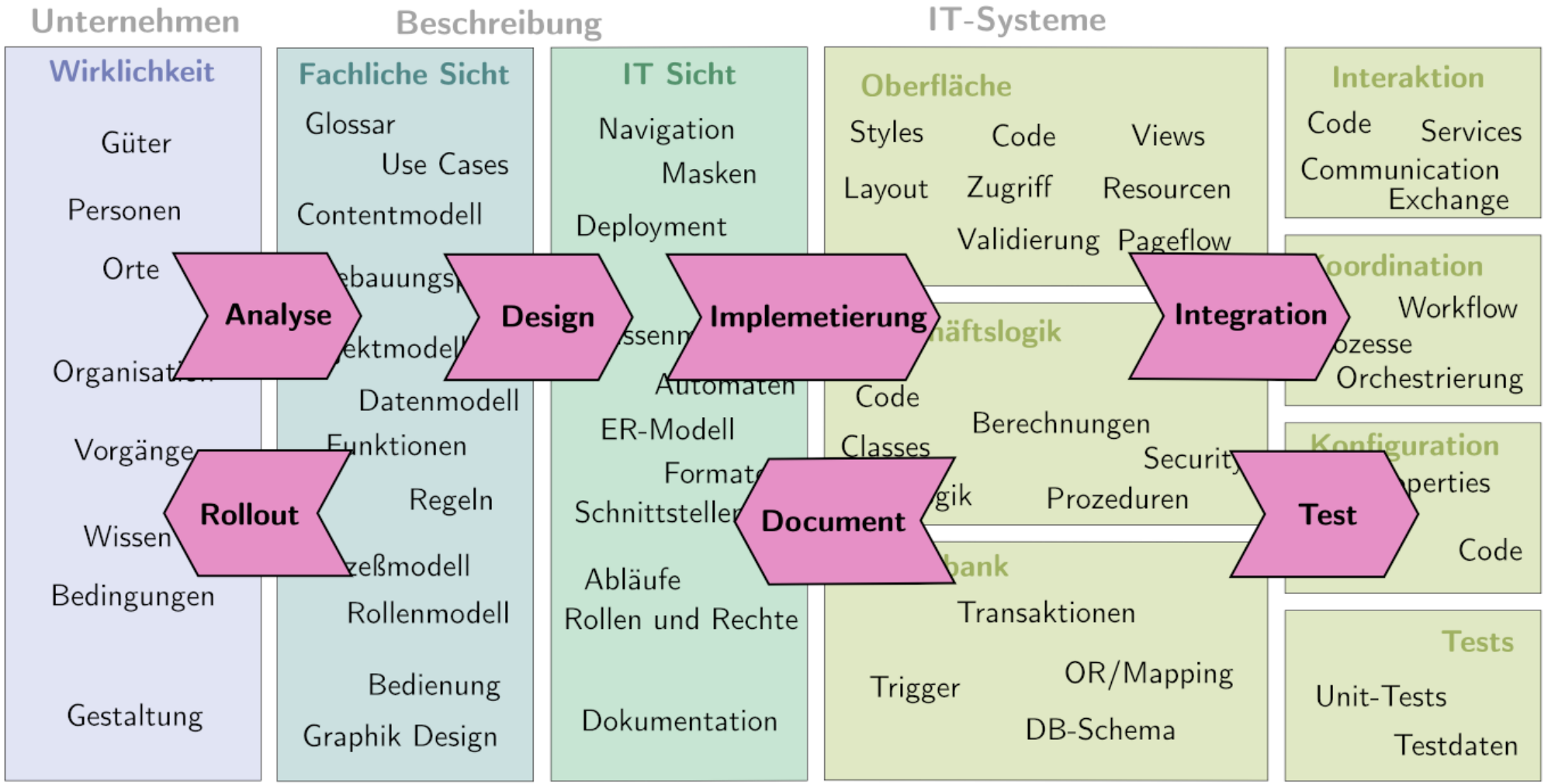
Sphenon

Inhalt

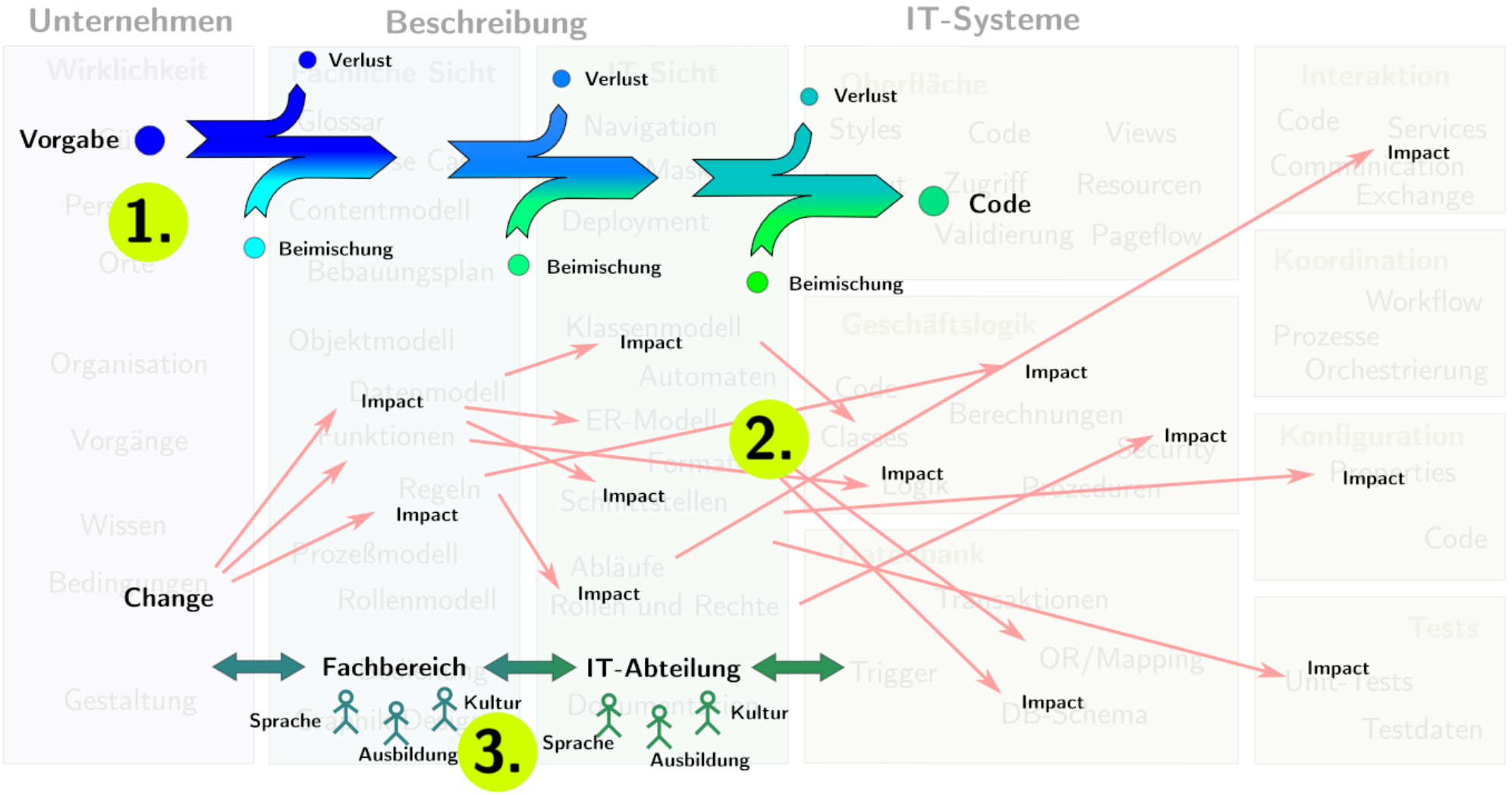
Formbare Software für lebendige Unternehmen



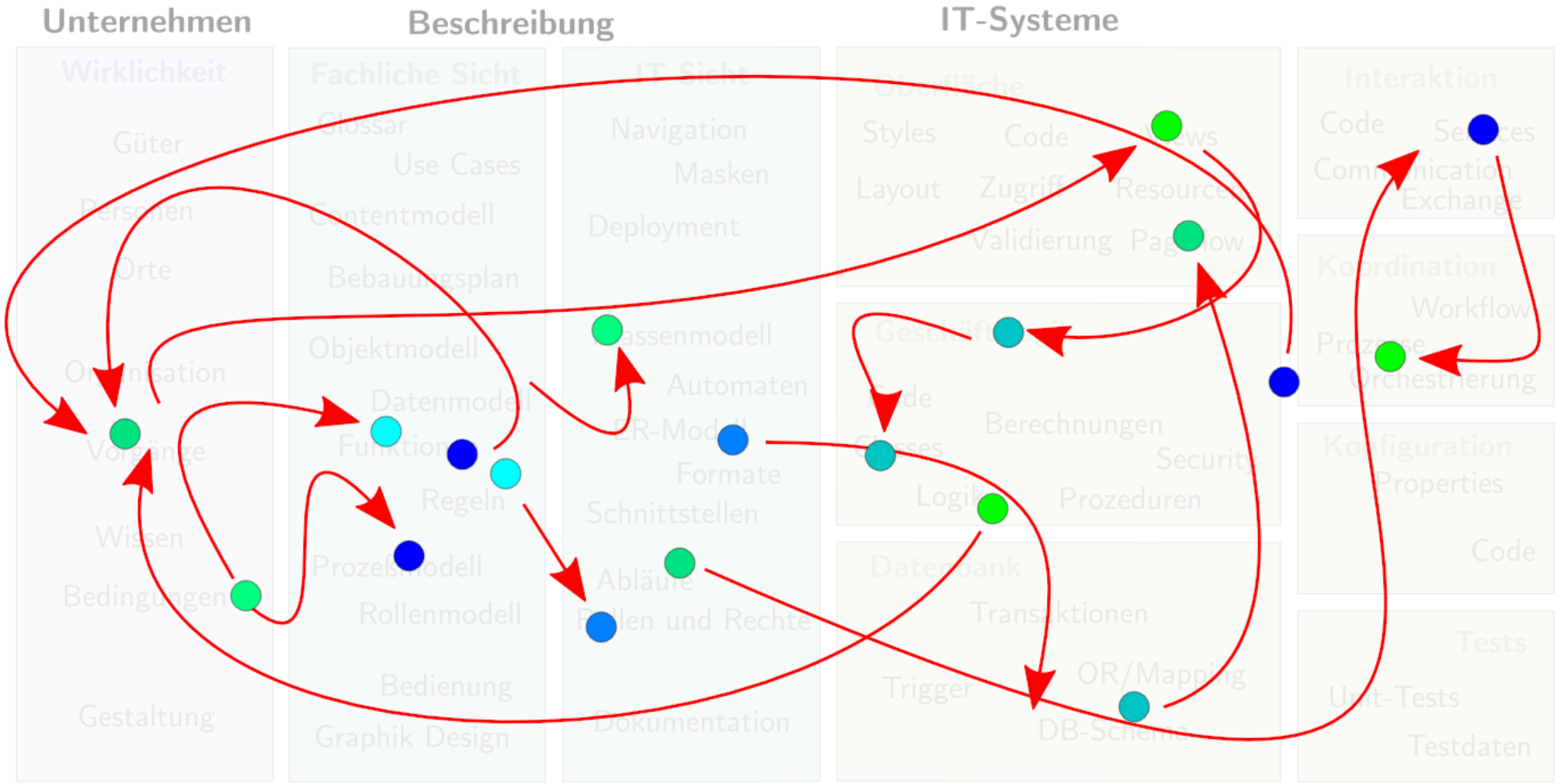
Artefakte & Herstellung



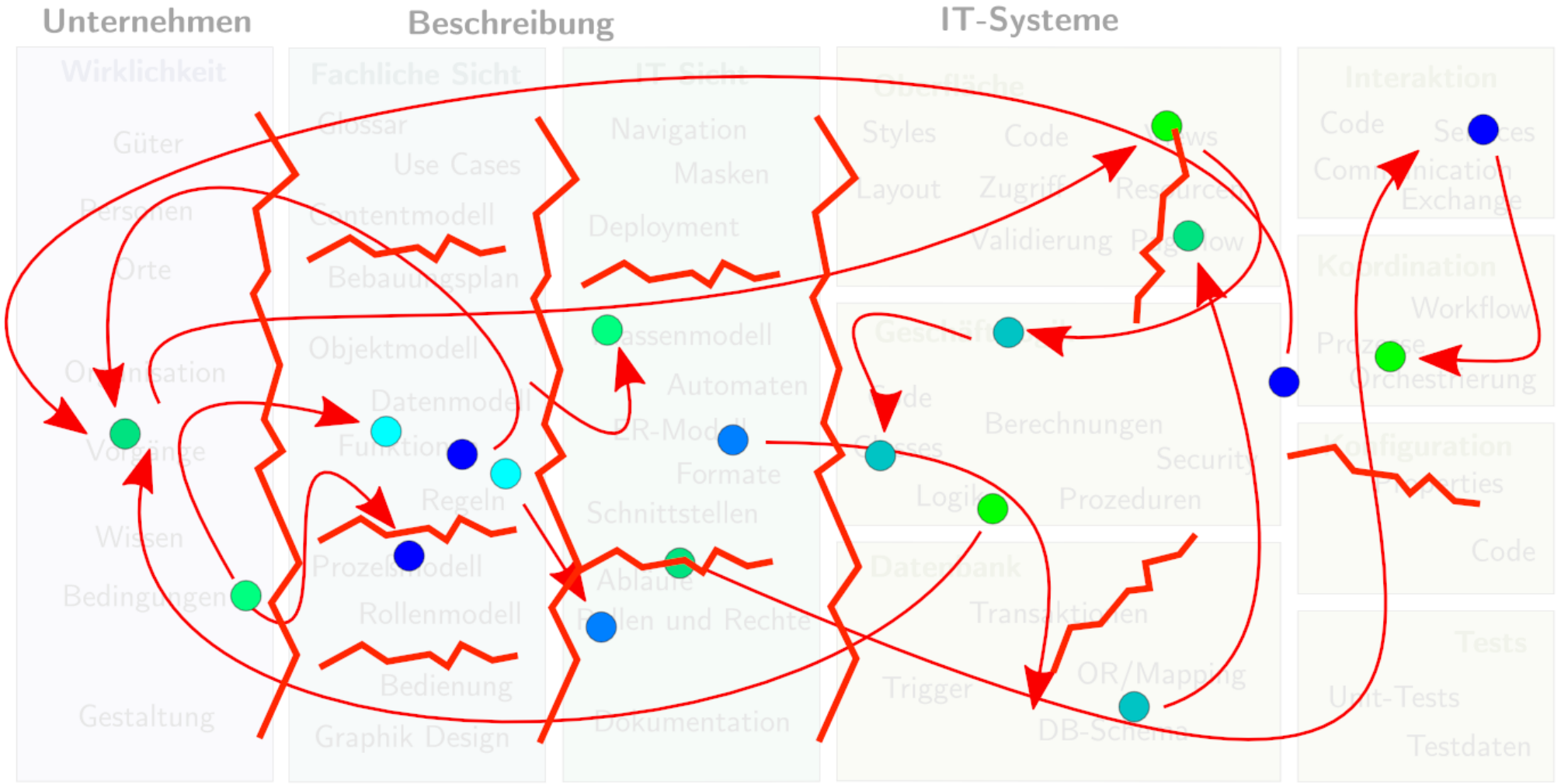
Software-Manufaktur



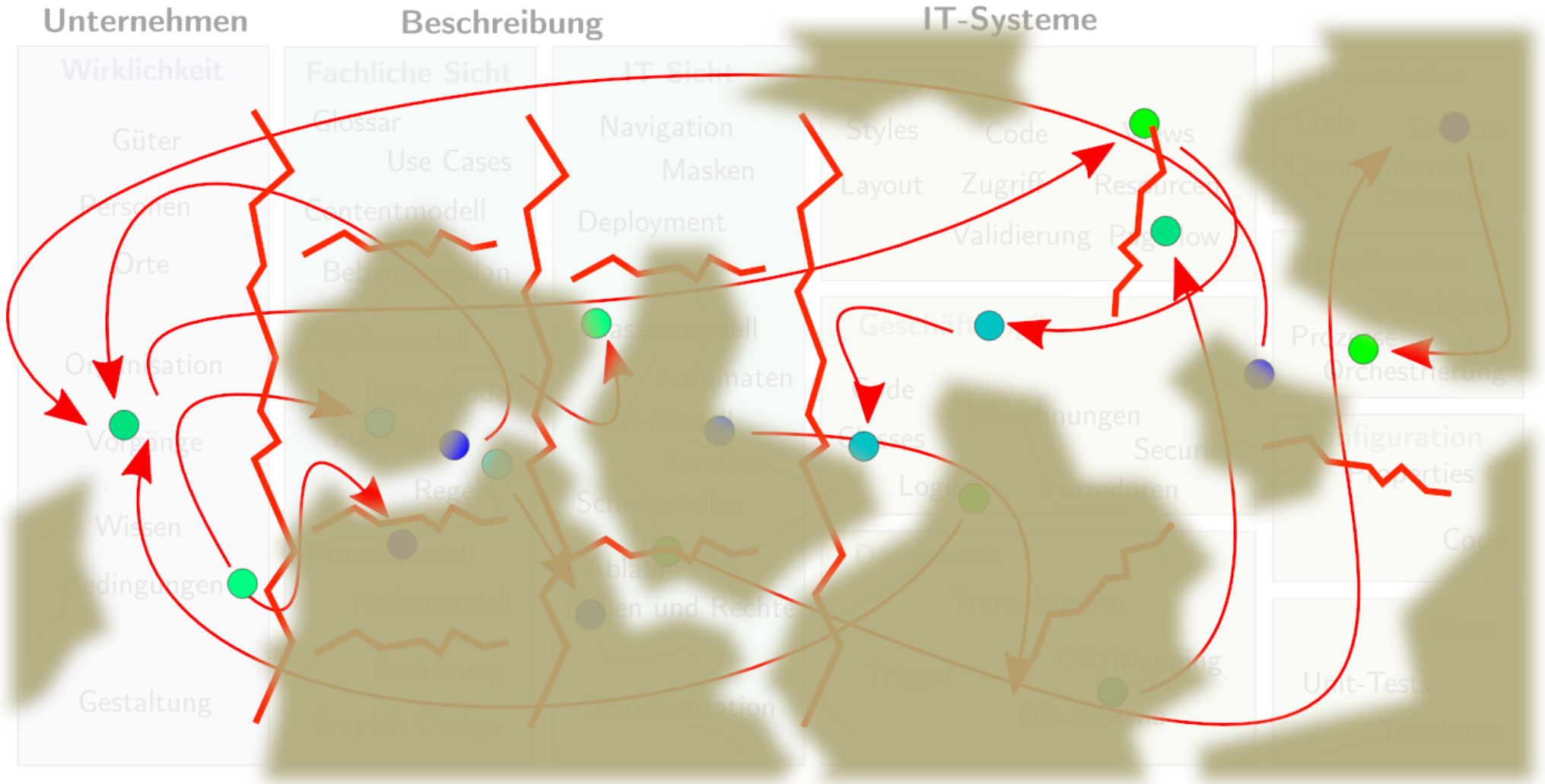
Chaos


Unternehmen
Beschreibung
IT-Systeme
Wirklichkeit
Fachliche Sicht
IT-Sicht
Interaktion
Koordination
Konfiguration
Datenbank
Tests
Güter
Personen
Orte
Organisation
Vorgänge
Wissen
Bedingungen
Gestaltung
Glossar
Use Cases
Contentmodell
Bebauungsplan
Objektmodell
Datenmodell
Funktionen
Regeln
Prozessmodell
Rollenmodell
Bedienung
Graphik Design
Navigation
Masken
Deployment
Klassenmodell
Automaten
ER-Modell
Formate
Schnittstellen
Abläufe
Rollen und Rechte
Dokumentation
Oberfläche
Styles
Code
Views
Layout
Zugriff
Resource
Validierung
Page Flow
Code
Services
Communication
Exchange
Workflow
Prozesse
Orchestrierung
Code
Berechnungen
Classes
Security
Logik
Prozeduren
Properties
Code
Transaktionen
Trigger
OR/Mapping
DB-Schema
Unit-Tests
Testdaten

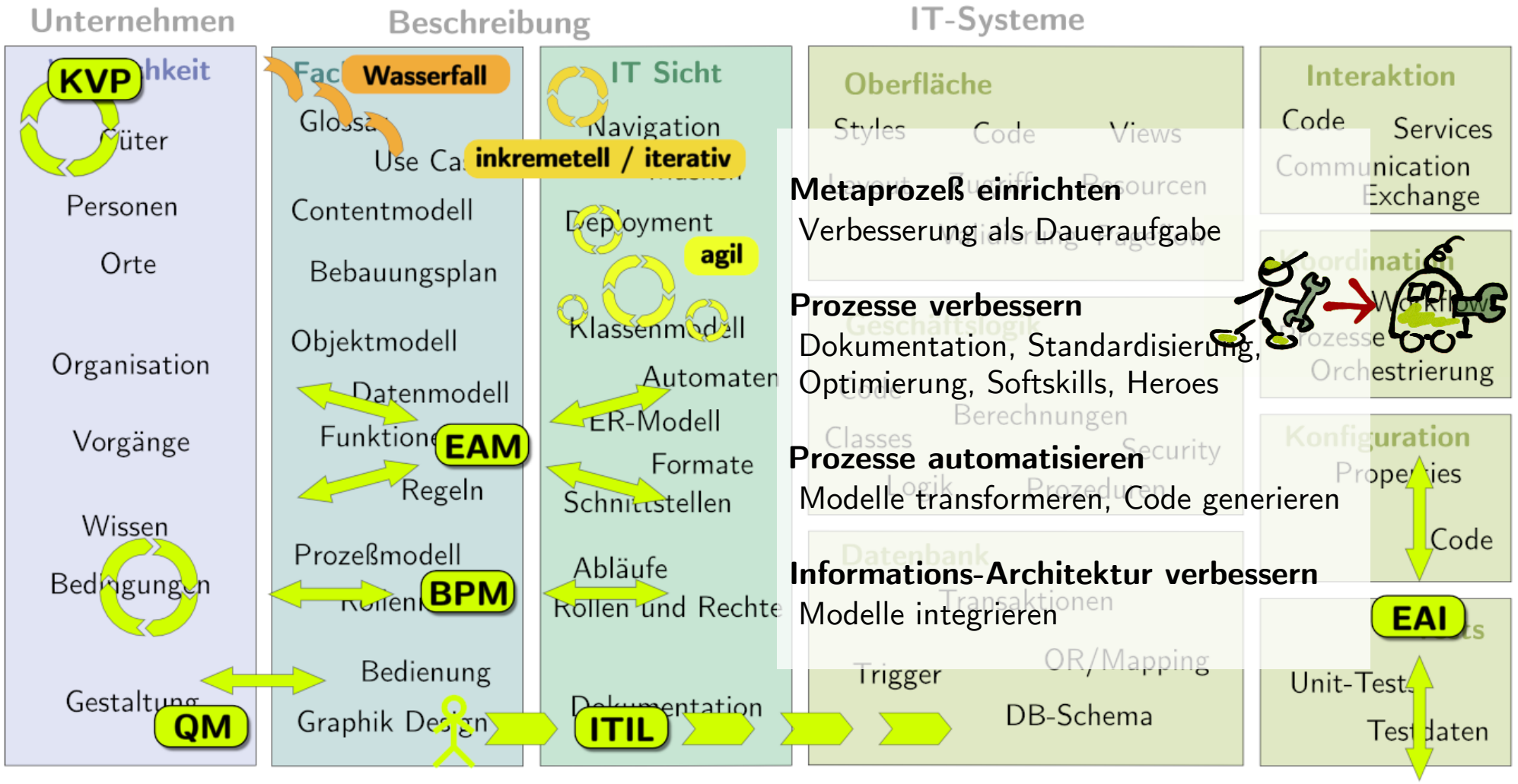
Chaos, Brüche



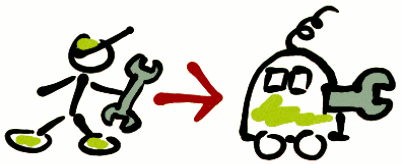
Chaos, Brüche, Erosion

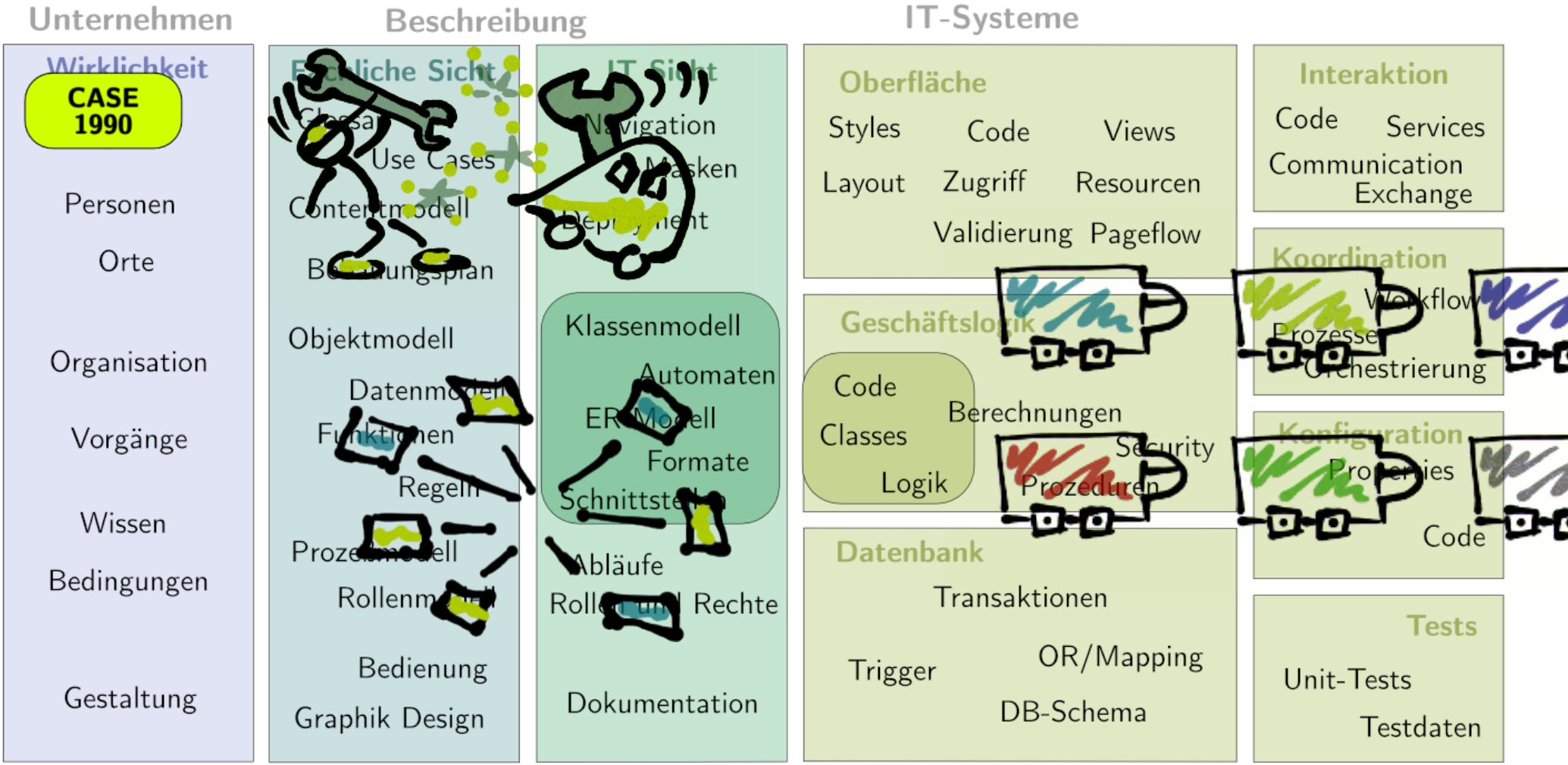


Gegenmaßnahmen

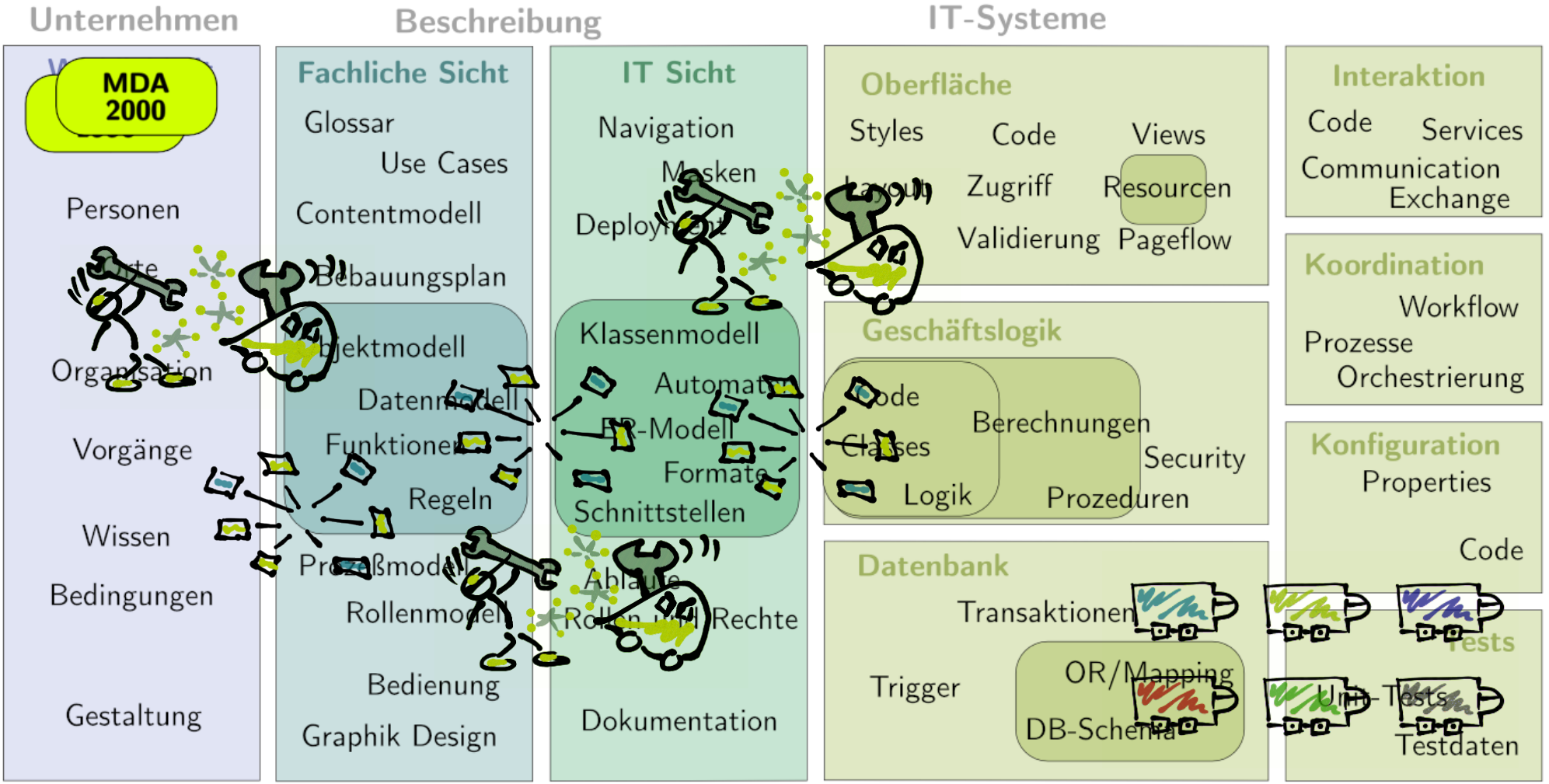


CASE...

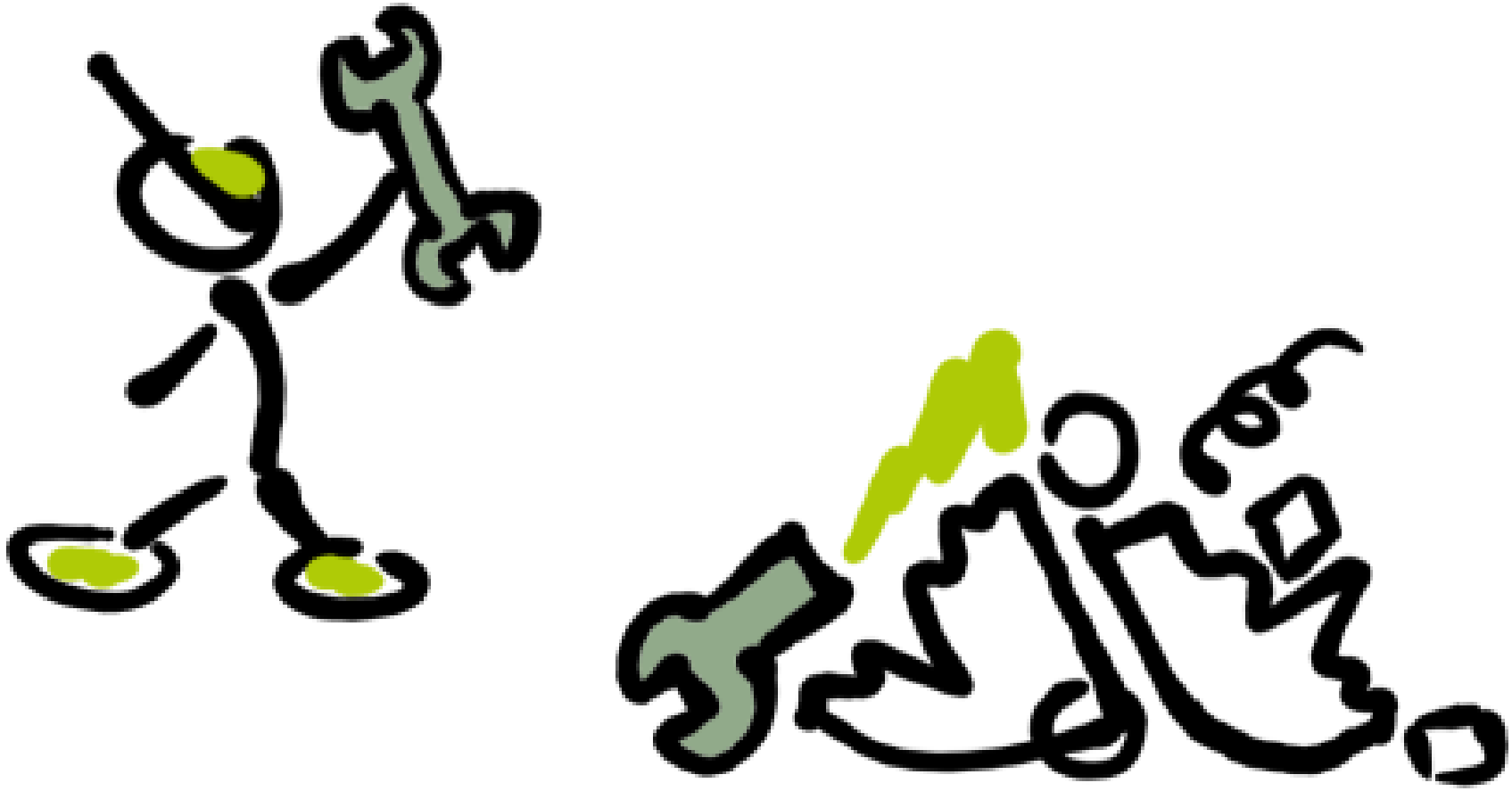



Unternehmen
Beschreibung
IT-Systeme
Wirklichkeit
CASE 1990
Personen
Orte
Organisation
Vorgänge
Wissen
Bedingungen
Gestaltung
Fachliche Sicht
IT Sicht
Use Cases
Navigation
Masken
Objektmodell
Datenmodell
Funktionen
Regeln
Prozessmodell
Rollenmodell
Bedienung
Graphik Design
Klassenmodell
Automaten
ER-Modell
Formate
Schnittstellen
Abläufe
Rollen und Rechte
Dokumentation
Oberfläche
Styles
Code
Views
Layout
Zugriff
Resourcen
Validierung
Pageflow
Geschäftslogik
Code
Classes
Logik
Berechnungen
Prozeduren
Security
Datenbank
Transaktionen
Trigger
OR/Mapping
DB-Schema
Interaktion
Code
Services
Communication
Exchange
Koordination
Workflow
Prozesse
Orchestrierung
Konfiguration
Properties
Code
Tests
Unit-Tests
Testdaten

...MDA...


Unternehmen
Beschreibung
IT-Systeme
MDA 2000
Personen
Organisation
Vorgänge
Wissen
Bedingungen
Gestaltung
Fachliche Sicht
Glossar
Use Cases
Contentmodell
Bebauungsplan
Objektmodell
Datenmodell
Funktionen
Regeln
Prozeßmodell
Rollenmodell
Bedienung
Graphik Design
IT Sicht
Navigation
Masken
Deployment
Klassenmodell
Automaten
DB-Modell
Formate
Schnittstellen
Ablaufe
Rollen und Rechte
Dokumentation
Oberfläche
Styles
Code
Views
Layout
Zugriff
Resourcen
Validierung
Pageflow
Geschäftslogik
Code
Classes
Logik
Berechnungen
Security
Prozeduren
Datenbank
Transaktionen
Trigger
OR/Mapping
DB-Schema
Interaktion
Code
Services
Communication
Exchange
Koordination
Workflow
Prozesse
Orchestrierung
Konfiguration
Properties
Code
Tests
Unit-Tests
Testdaten

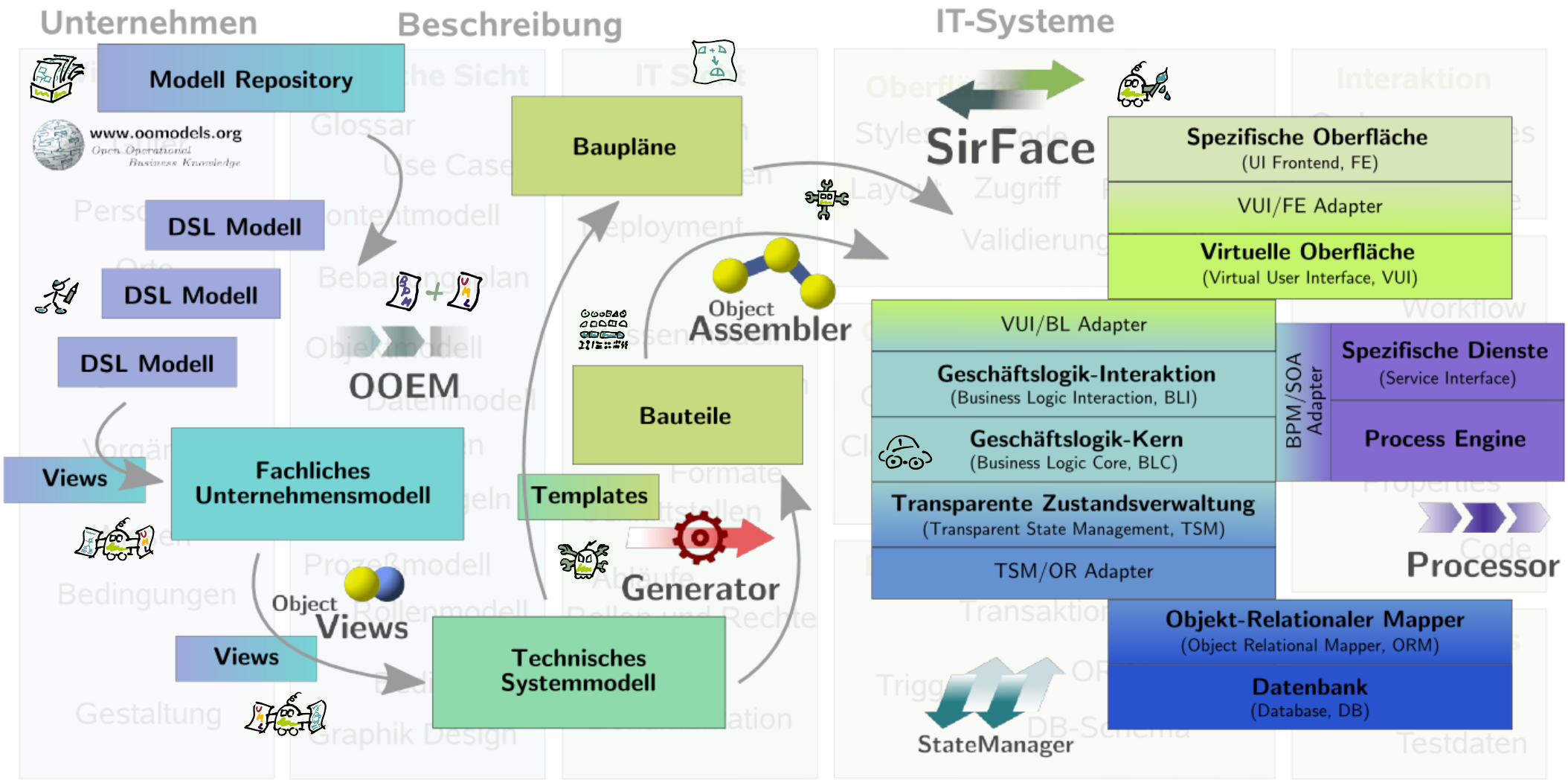
...in die Tonne?



Sphenon

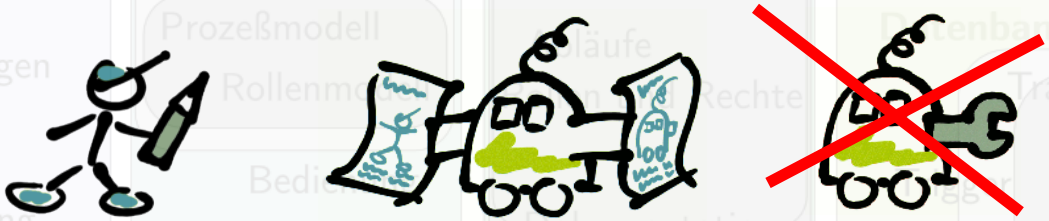
Inhalt

EM/OS Architektur



Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)



Fach-Modell → System-Modell → Hoch-sprache → Assembler → Maschinen-sprache

EM/OS 1.0
1993-2005

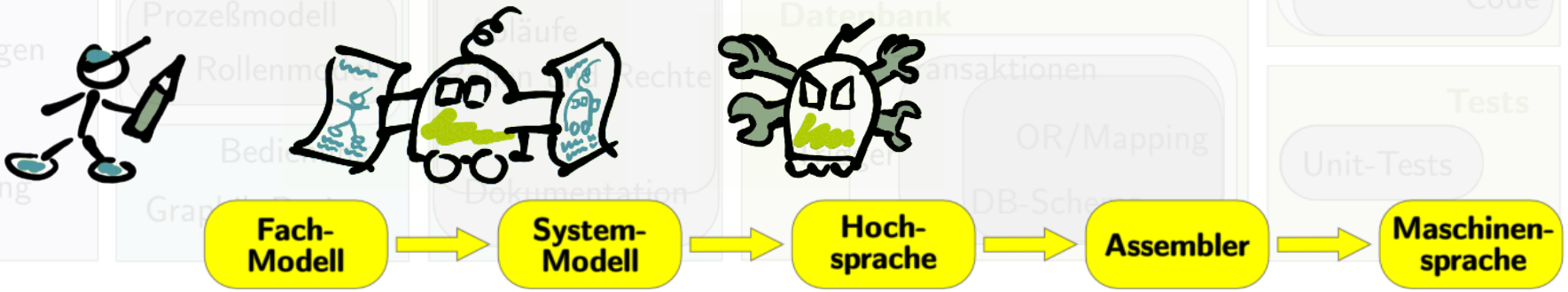
Hochmoderner Generator in 4. Generation

- 2001
- 24322 Zeilen
- 1 Maintainer
- perl, Java, HTML, JS

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
                ⋮ «∀(a:‖"P/Attributes"){»...
    protected <type> ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public <type> get◂‖"P/Name"▸ ();
                ⋮ «}/∀»...
⊏ /☒ ⊐...
```



EM/OS 1.0
1993-2005

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
                ⋮ «∀(a:‖"P/Attributes"){»...
    protected <type> ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public <type> get◂‖"P/Name"▸ ();
                ⋮ «}/∀»...
⊏ /☒ ⊐...
```

«∀(a:‖"Ⓟ/Attributes"){»

- **Sehr gute Lesbarkeit durch Unicode**
- Modular wählbare Syntax
- OO Templates
- Modularisierbarkeit schafft Ordnung
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- Zahlreiche, konfigurierbare Syntaxelemente
- Sehr reife Software

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
        ⋮ «∀(a:‖"P/Attributes"){»...
    protected ≤type≥ ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public ≤type≥ get◂‖"P/Name"▸ ();
        ⋮ «}/∀»...
⊏ /☒ ⊐...
```

```
<% for(Object a:...
```

```
public class MyClass<X, Y<X>> {
    public X getX() { ...
```

- Sehr gute Lesbarkeit durch Unicode
- **Modular wählbare Syntax**
- OO Templates
- Modularisierbarkeit schafft Ordnung
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- Zahlreiche, konfigurierbare Syntaxelemente
- Sehr reife Software

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
        ⋮ «∀(a:‖"P/Attributes"){»...
    protected ≺type≻ ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public ≺type≻ get◂‖"P/Name"▸ ();
        ⋮ «}/∀»...
⊏ /☒ ⊐...
```

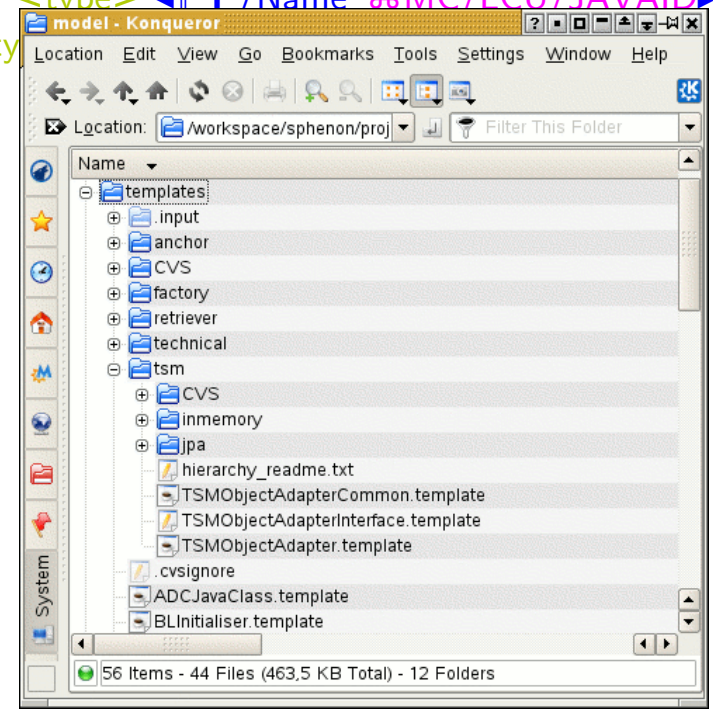
```
G-2.0-uml_state_machine-...
  ⊏ base ClassTemplate ⊐...
  ⊏ ☒ Features(String myarg) ⊐...
  ⊏ ➜ super.Features(myarg) ⊐...
```

- Sehr gute Lesbarkeit durch Unicode
- Modular wählbare Syntax
- **OO Templates**
- Modularisierbarkeit schafft Ordnung
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- Zahlreiche, konfigurierbare Syntaxelemente
- Sehr reife Software

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
                : «∀(a:‖"P/Attributes"){»...
    protected ≤type≥ ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public ≤ty
⊏ /☒ ⊐...
```



- Sehr gute Lesbarkeit durch Unicode
- Modular wählbare Syntax
- OO Templates
- **Modularisierbarkeit schafft Ordnung**
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- Zahlreiche, konfigurierbare Syntaxelemente
- Sehr reife Software

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ⊠ Attributes(Object current) ⊐...
        ⋮ «∀(a:‖"P/Attributes"){»...
    protected ≺type≻ ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public ≺type≻ get◂‖"P/Name"▸ ();
        ⋮ «}/∀»...
⊏ /⊠ ⊐...
```

- Sehr gute Lesbarkeit durch Unicode
- Modular wählbare Syntax
- OO Templates
- Modularisierbarkeit schafft Ordnung
- **Extrem performant, dynamische Übersetzung**
- **Voller Java Sprachumfang verfügbar**
- Zahlreiche, konfigurierbare Syntaxelemente
- Sehr reife Software

Hochmoderner Generator in 4. Generation

www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
                ⋮ «∀(a:‖"P/Attributes"){»...
    protected <type> ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public <type> get◂‖"P/Name"▸ ();
                ⋮ «}/∀»...
⊏ /☒ ⊐...
```

<mytag(myarg,...)>

- Sehr gute Lesbarkeit durch Unicode
- Modular wählbare Syntax
- OO Templates
- Modularisierbarkeit schafft Ordnung
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- **Zahlreiche, konfigurierbare Syntaxelemente**
- Sehr reife Software

Hochmoderner Generator in 4. Generation

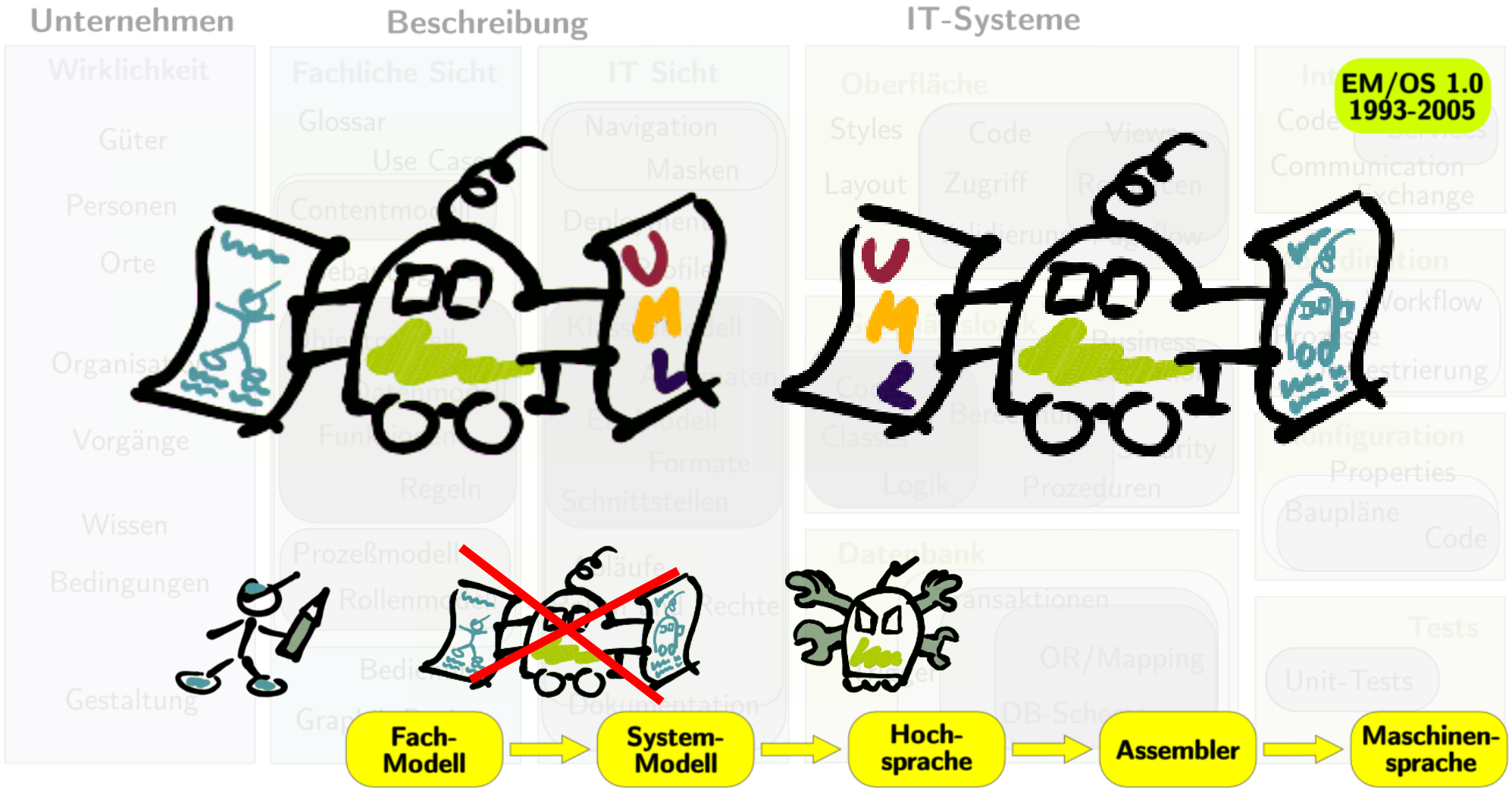
www.oogenerator.org (open source)

```
G-2.0-uml-1.0 -*- coding: utf-8; -*-
≺ Attributes                                    ≻...
⊏ ☒ Attributes(Object current) ⊐...
                ⋮ «∀(a:‖"P/Attributes"){»...
    protected <type> ◂‖"P/Name"⌘MC/LCU/JAVAID▸;
    public <type> get◂‖"P/Name"▸ ();
                ⋮ «}/∀»...
⊏ /☒ ⊐...
```

- Sehr gute Lesbarkeit durch Unicode
- Modular wählbare Syntax
- OO Templates
- Modularisierbarkeit schafft Ordnung
- Extrem performant, dynamische ‹bersetzung
- Voller Java Sprachumfang verfügbar
- Zahlreiche, konfigurierbare Syntaxelemente
- **Sehr reife Software**

- C 1.0 1993...
- C++ 1996...
- perl 1998..., 1.0 2000
- Java 1.0 2002, 2.0 2005, 3.0 2008

Mehrstufige dynamische Modelltransformation



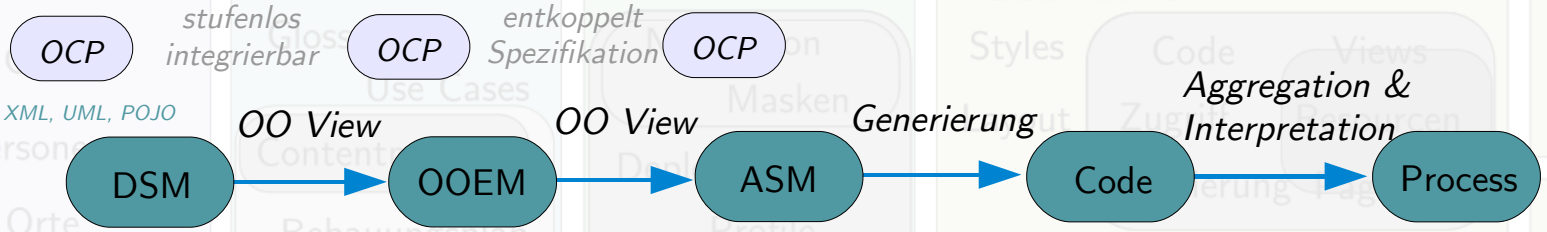
Mehrstufige dynamische Modelltransformation

- Völlige Entkopplung fachlich & technisch
- effektiv wieder-verwendbare Modelle
- handliche Optionen für DSLs u.a.: DSM & OCP-Macros
- ‹berschaubare Themen auf jeder Stufe

Unternehmen

Beschreibung

EM/OS



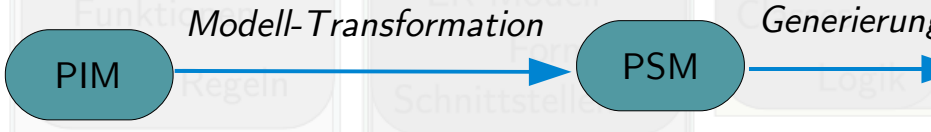
EM/OS 1.0
1993-2005

www.ooem.org
www.xmodel.org
www.ooviews.org

fachlich orientiert (domänen-spezifisch)

technisch orientiert (plattform-spezifisch)

MDA

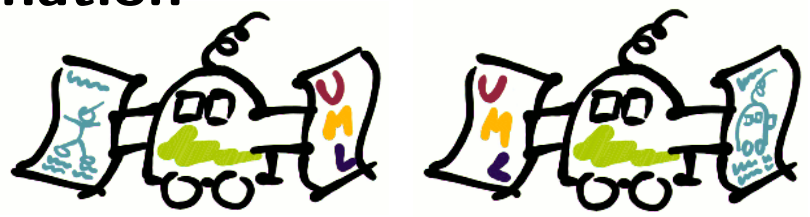


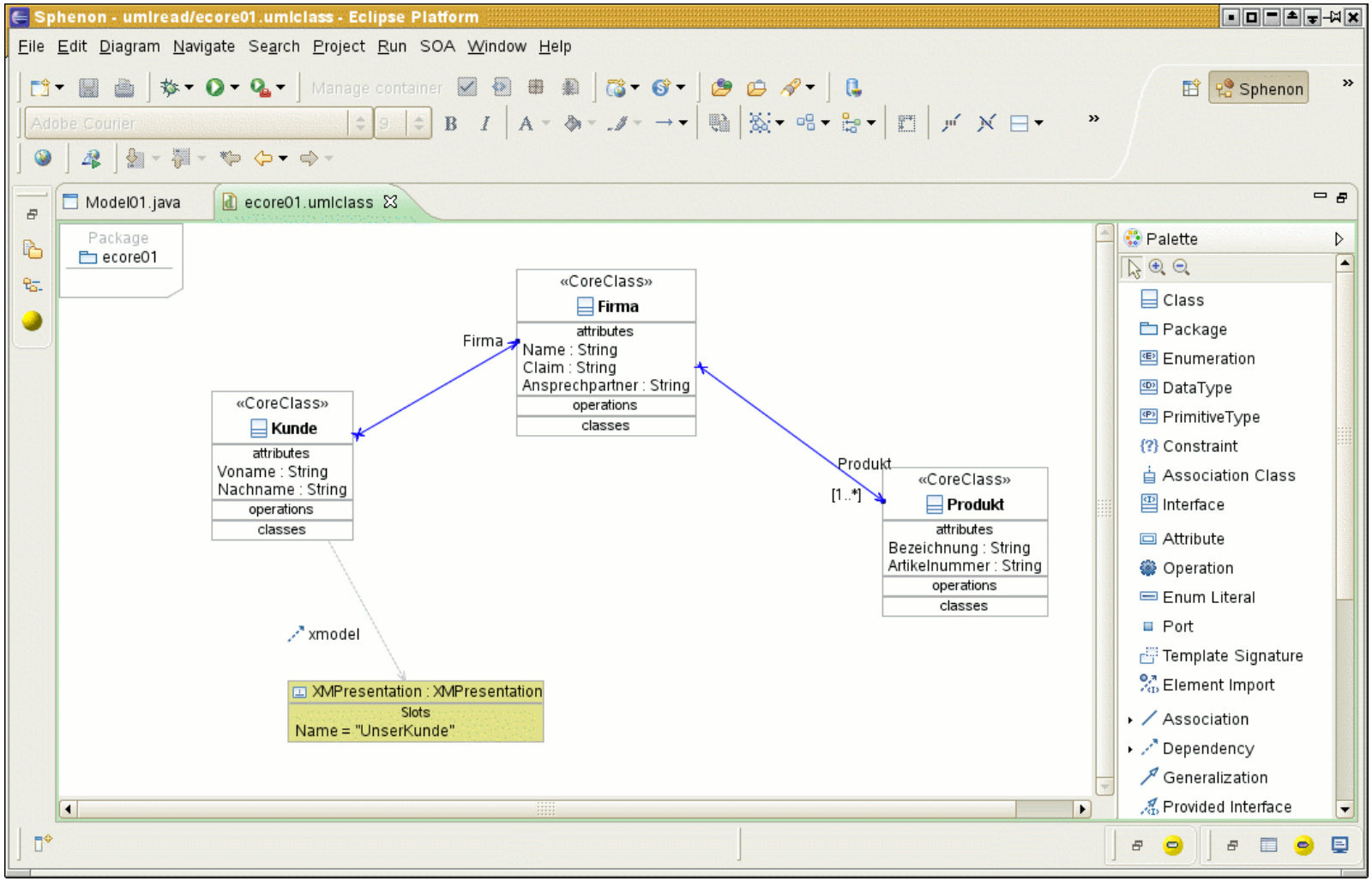
Fach-Modell → System-Modell → Hoch-sprache → Assembler → Maschinen-sprache

fachlich neutral

Mehrstufige dynamische Modelltransformation

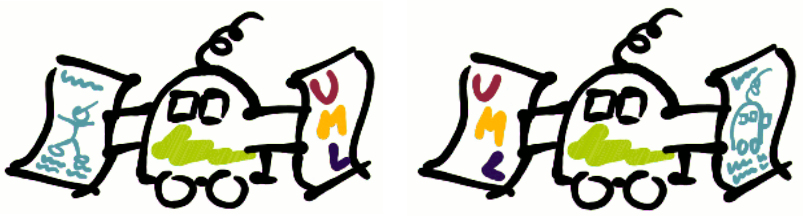
Eclipse UML 2 Editor





Mehrstufige dynamische Modelltransformation

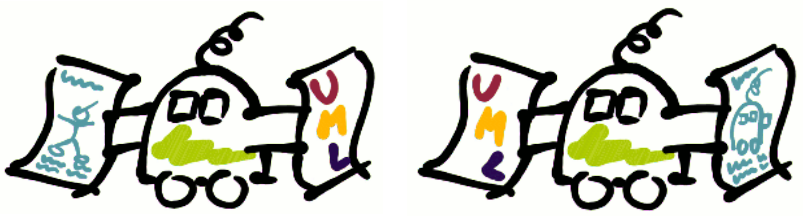
OOEM 2.0 (XML, mit OCP gelesen)



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Class>
  <StereotypeModelImport>org.uml.stereotypes.*</StereotypeModelImport>
  <ModelImport>org.oomodels.business.foundation.entities.*</ModelImport>
  <Name>Person</Name>
  <Base Parent="Entity"/>
  <Stereotype>CoreClass</Stereotype>
  <Attribute Name="Name" Type="NameOfPerson"/>
  <Attribute Name="Sex" Type="Sex"/>
  <Attribute Name="Birthday" Type="Date"/>
</Class>
```

Mehrstufige dynamische Modelltransformation

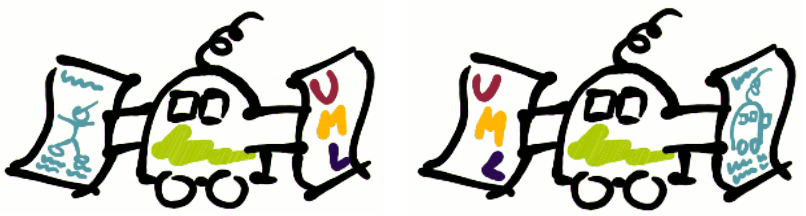
WIML – Wiki Modeling Language

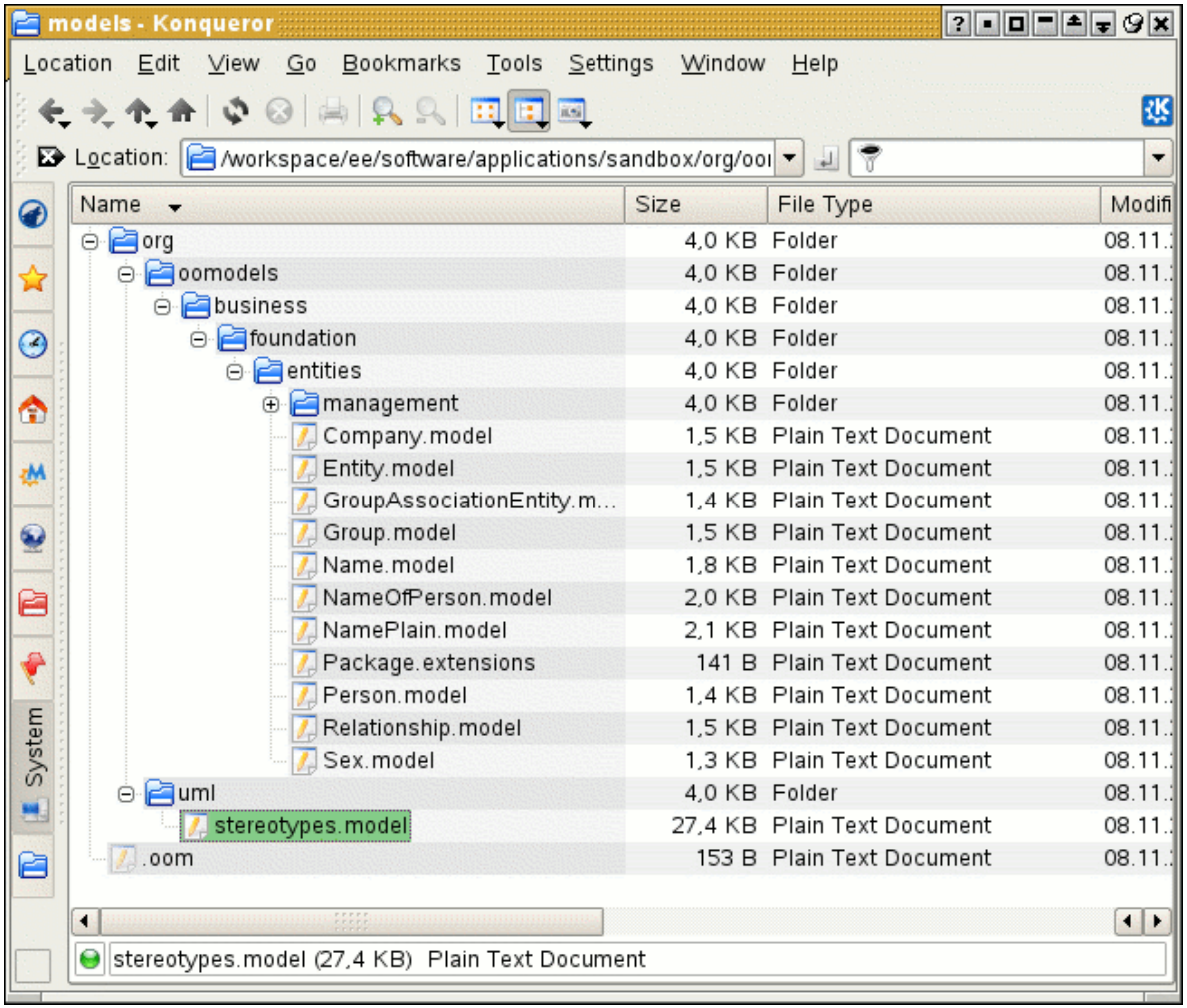


```
<wiml>
* Person CoreClass
  > Entity
  A human person
** Name NameOfPerson
** Sex Sex
** Birthday Date
</wiml>
```

Mehrstufige dynamische Modelltransformation

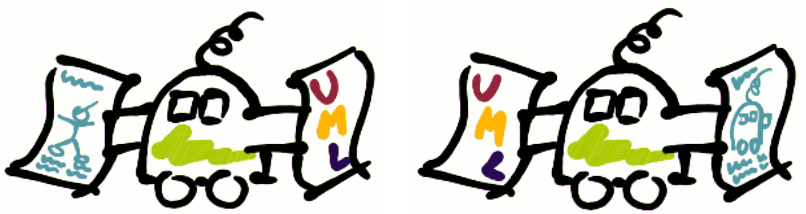
Teil-Modell-Artefakte im Dateisystem





Mehrstufige dynamische Modelltransformation

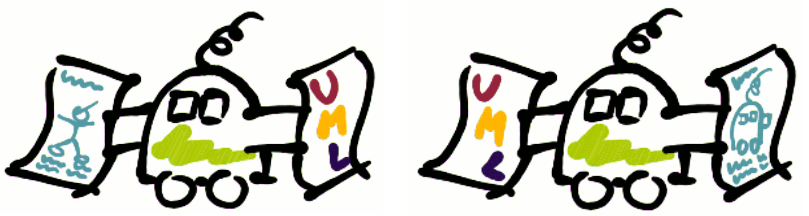
M2M mit OO View mit OCP



```
<?xml version="1.0" encoding="UTF-8"?>
<UMLClass
 SIGNATURE="UMLClass Source, ADCModelOptions Options ..."
 POLYMORPHIC="Class" DEFINE="..." xmlns="...">
 <ModelImport
  FOREACH="m_import : ::*!Source/ModelImports!"
  EXPRESSION="true">m_import</ModelImport>
 <ArtefactOrigin><Track>
    <o1 OID="Origin">com/sphenon/ad/adcore/model/coremodel/UMLClass_Interface</o1>
    <o2 EXPRESSION="true">Source.getPath(context)</o2>
  </Track></ArtefactOrigin>
 <Base FOREACH="base : Source.getBases(context)"
      IF="var ifc = ::[base]'//XModel/Generalization/Interface'; ..."
      >
  <Parent EXPRESSION="true">jspp: prefix + ::[base]'//Property/Parent/Id'</Parent>
 </Base>
 <Name OID="Name" EXPRESSION="true">prefix + Source.getName(context)</Name>
 <Attributes...
```

Mehrstufige dynamische Modelltransformation

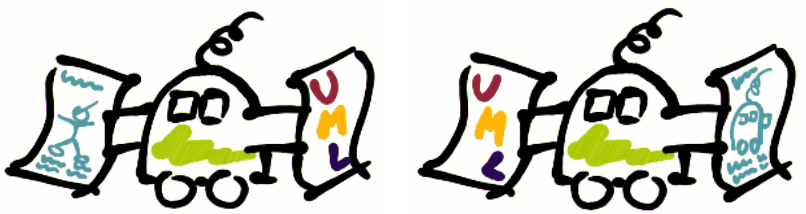
Vermischte Modelle



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Package CLASS="Aggregate::com/sphenon/tools/emfumloov/UMLPackage_EMF">
  <Resource>/DATA/work/eclipse34_ws/umlread/ecore01.uml</Resource>
</Package>
```

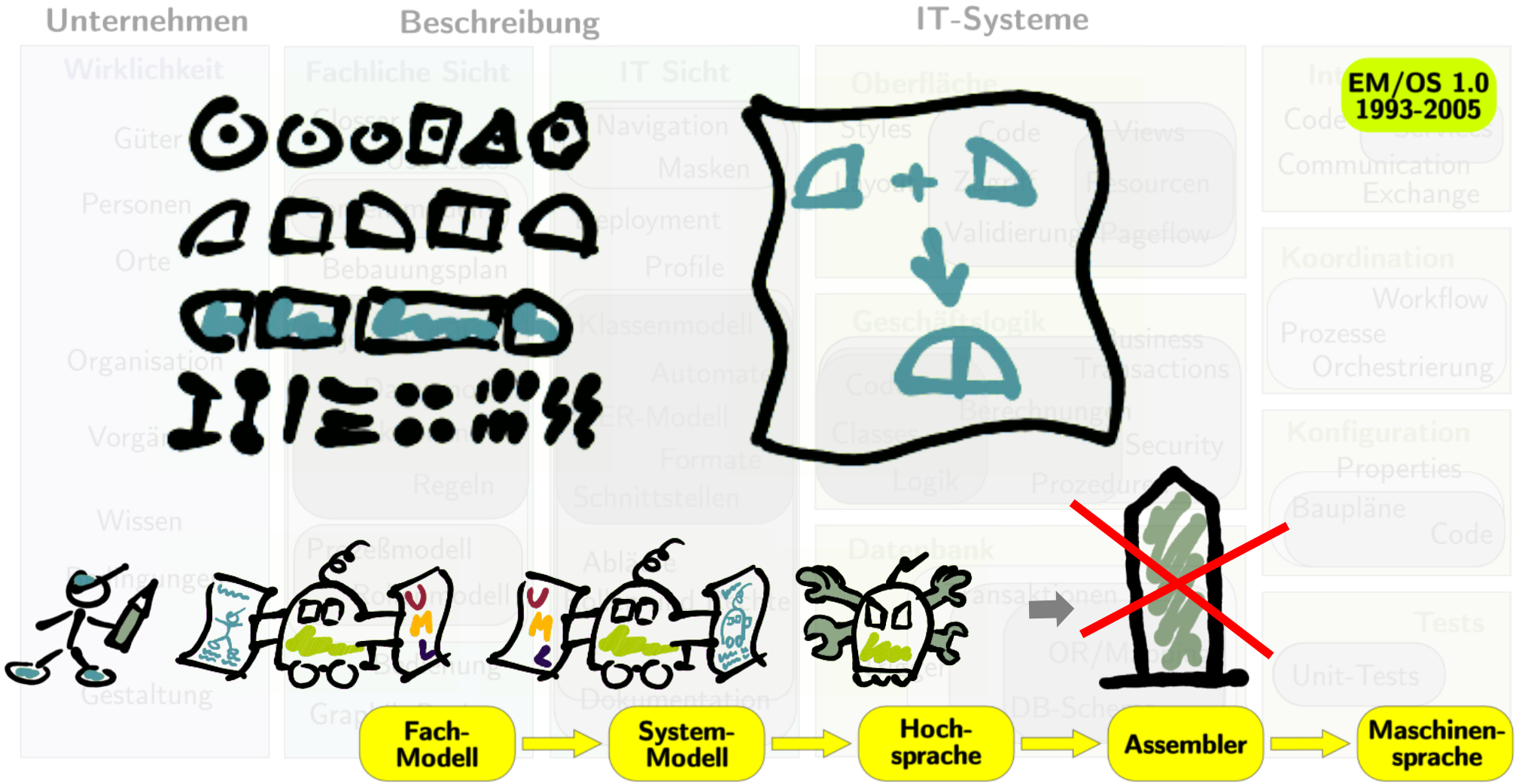
Mehrstufige dynamische Modelltransformation

OOEM View auf Ecore UML



```
<?xml version="1.0" encoding="UTF-8"?>
<UMLPackage
 SIGNATURE="String Resource"
 CLASS="UMLPackage"
 DEFINE="pkg : new Packages.com.sphenon.tools.emfumloov.ECoreAccessor(context, Resource).getPackage(context)"
 xmlns="http://xmlns.sphenon.com/com/sphenon/ad/adcore/model" xmlns:code="code">
 <InternalModelImport>org.uml.stereotypes.*</InternalModelImport>
 <ExternalImport>java.lang.String</ExternalImport>
 <Name EXPRESSION="true">pkg.getName(context)</Name>
 <ArtefactOrigin><Track><o1 EXPRESSION="true">'EMF Resource ' + Resource</o1></Track></ArtefactOrigin>
 <Classes>
  <Class FOREACH="cls : pkg.getClasses(context)">
   <Stereotype FOREACH="sttp : cls.getStereotypes(context)" EXPRESSION="true">sttp.getName(context)</Stereotype>
   <Name EXPRESSION="true">cls.getName(context)</Name>
   <Attributes>
    <Attribute FOREACH="att : cls.getAttributes(context)">
     <Name EXPRESSION="true">att.getName(context)</Name>
     <Type EXPRESSION="true">att.getType(context)</Type>
     <Extensions>
      <XMExtension FOREACH="xme : att.getXModelExtensions(context)"
                   FACTORY="GenericFactory_XModelElement">
       <Class EXPRESSION="true">xme.getClassifiers(context).get(0).getName(context)</Class>
       <Properties EXPRESSION="value">xme.getProperties(context)</Properties>
      </XMExtension>
```

Roboter baut keinen Monolithen, sondern Bauteile und Baupläne

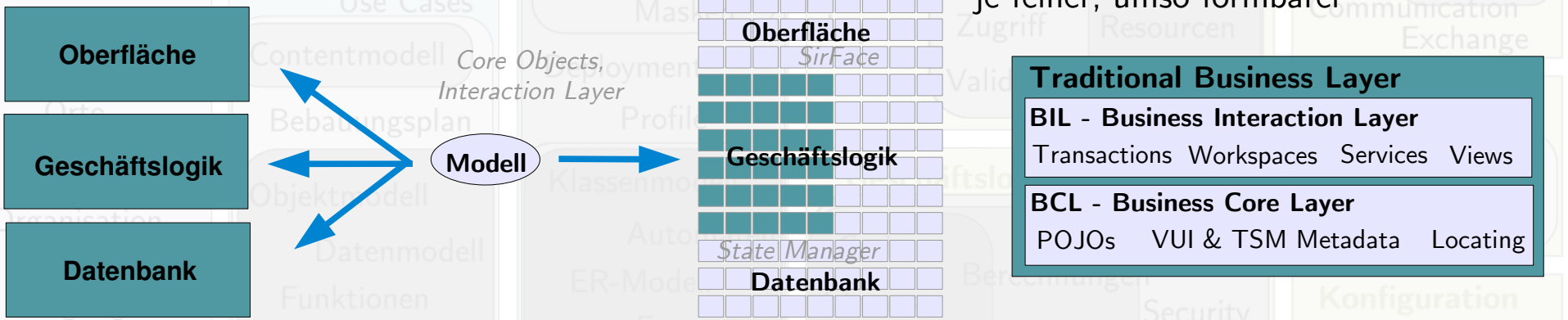


Roboter baut keinen Monolithen, sondern Bauteile und Baupläne

- Handgefertigte Bauteile koexistieren problemlos mit generierten Bauteilen und Fertigteilen
- Mannigfaltige Möglichkeiten (Baupläne, abgeleitete Klassen, Variantenpools, usw.)
- Von Konfektionsware zu Handarbeit mit vielen Auffangnetzen

EM/OS 1.0 1993-2005

- Trennung von Themen: je feiner, umso formbarer



Klassische Generierung

Generierung mit EM/OS

TSM Metadata

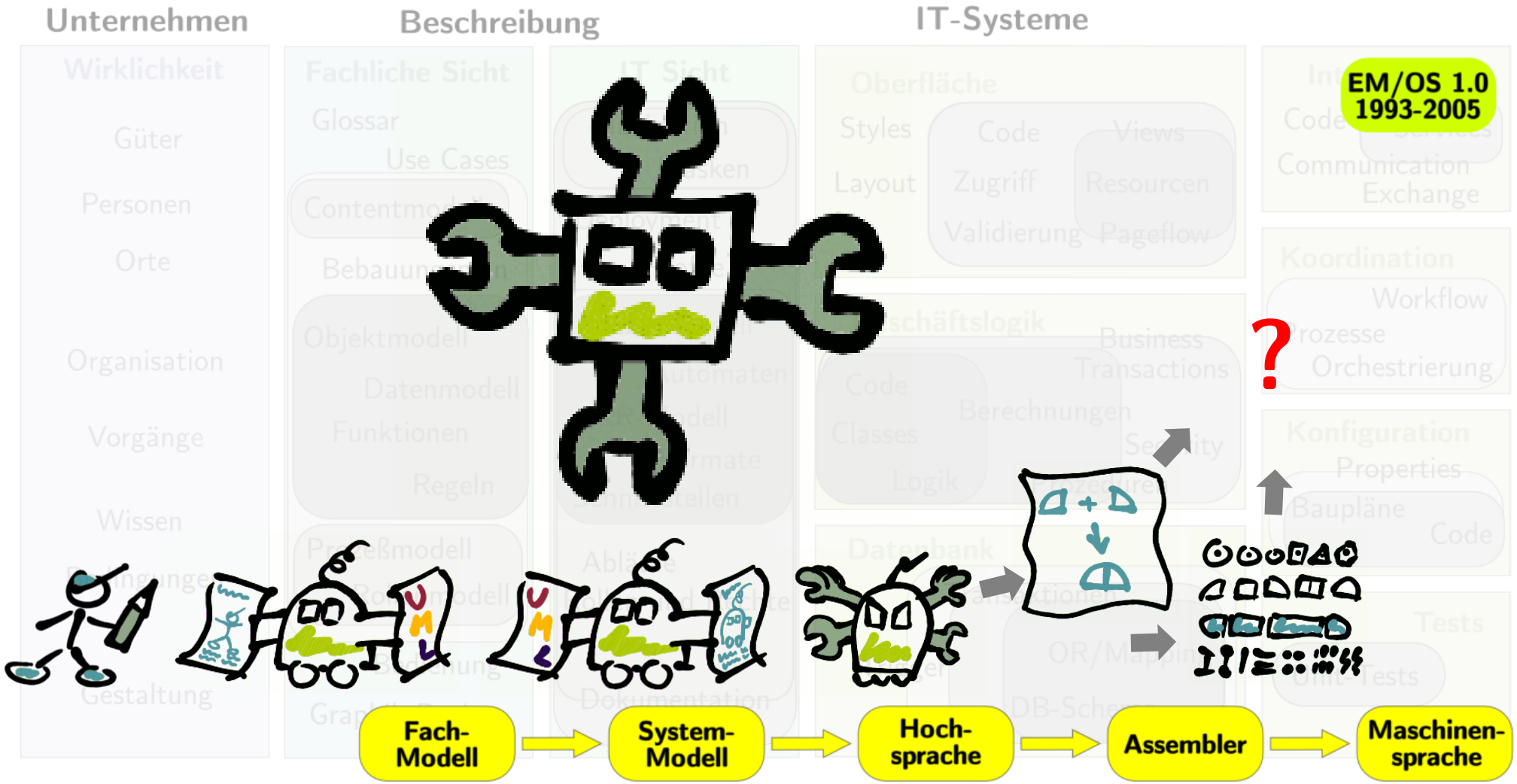
Fach-Modell → System-Modell → Hoch-sprache → Assembler → Maschinen-sprache

Roboter baut keinen Monolithen, sondern Bauteile und Baupläne

Roboter baut keinen Monolithen, sondern Bauteile und Baupläne

```
<SubSpace CLASS="VUIEntityObject_EditorSpace" >
  <ADO CLASS="EditorSpace" OID="space">
    <TargetType>org.oomodels.....Company</TargetType>
    <Operations>
      <Operation CLASS="Aggregate::org/.../CreateTransactionStarter_Company">
        <space OIDREF="space"/>
      </Operation>
      <Operation CLASS="Aggregate::org/.../BrowseTransactionStarter_Company">
        <space OIDREF="space"/>
      </Operation>
    </Operations>
  </ADO>
</SubSpace>
```

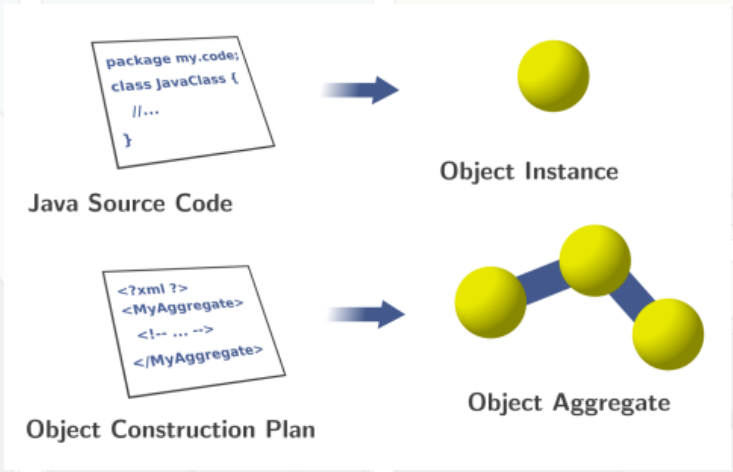
Neuer Roboter versteht Bauanleitungen und baut Teile zusammen



Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

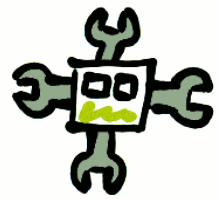
Objektbaupläne und der Object Assembler - www.xocp.org (open source)

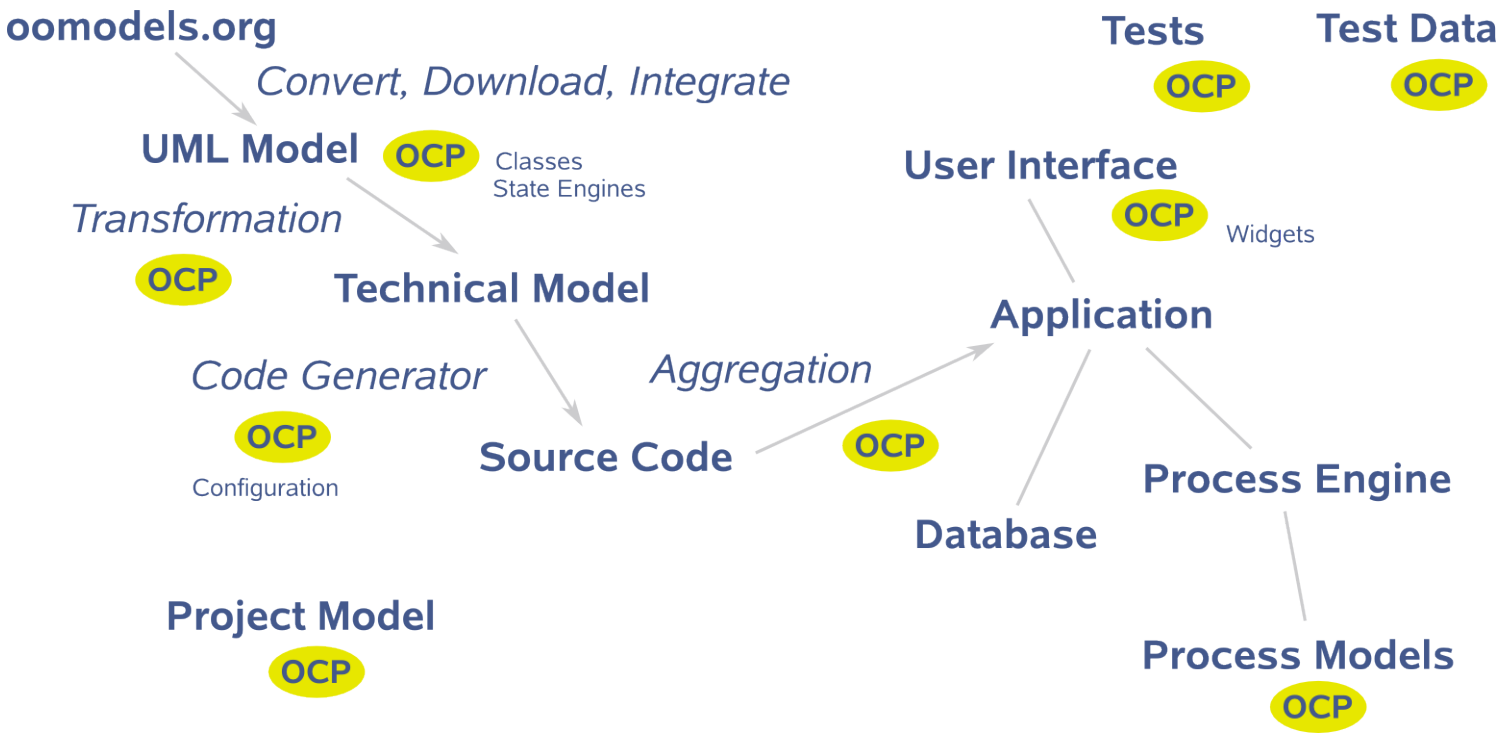
- baut Objektaggregate aus Bauteilen anhand von Bauplänen
- ehemals: gluecode/hardcoded
- breites Anwendungsspektrum
- extrem performant (JIT Compiler)
- dynamisches Laden und Neuübersetzen
- OO Baupläne



Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

Objektbaupläne und der Object Assembler - www.xocp.org (open source)

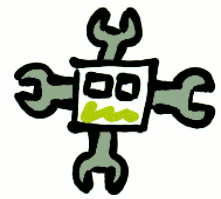


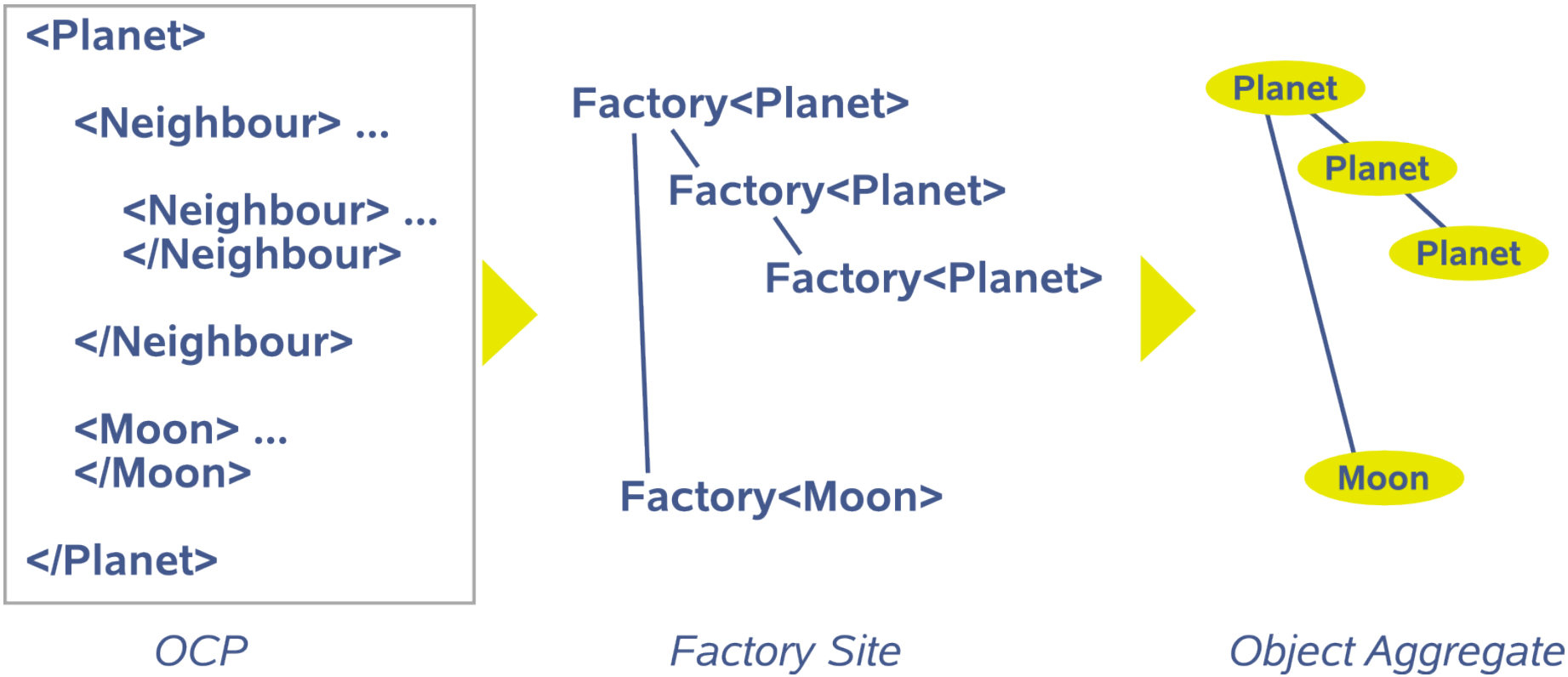


- 1996: Laden von Testdaten
- heute: 15% OCP-Anteil, bezogen auf Java

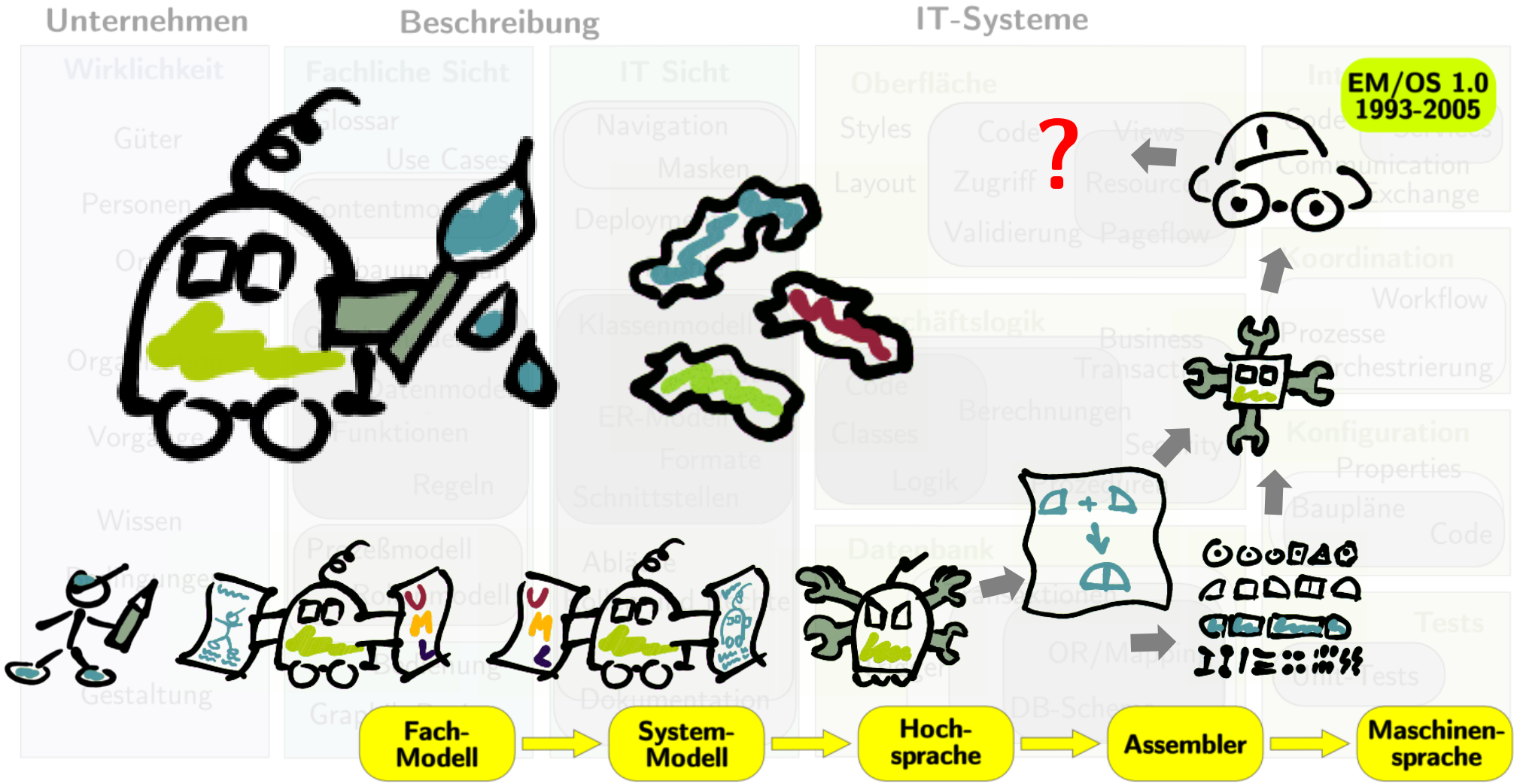
Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

Objektbaupläne und der Object Assembler - www.xocp.org (open source)



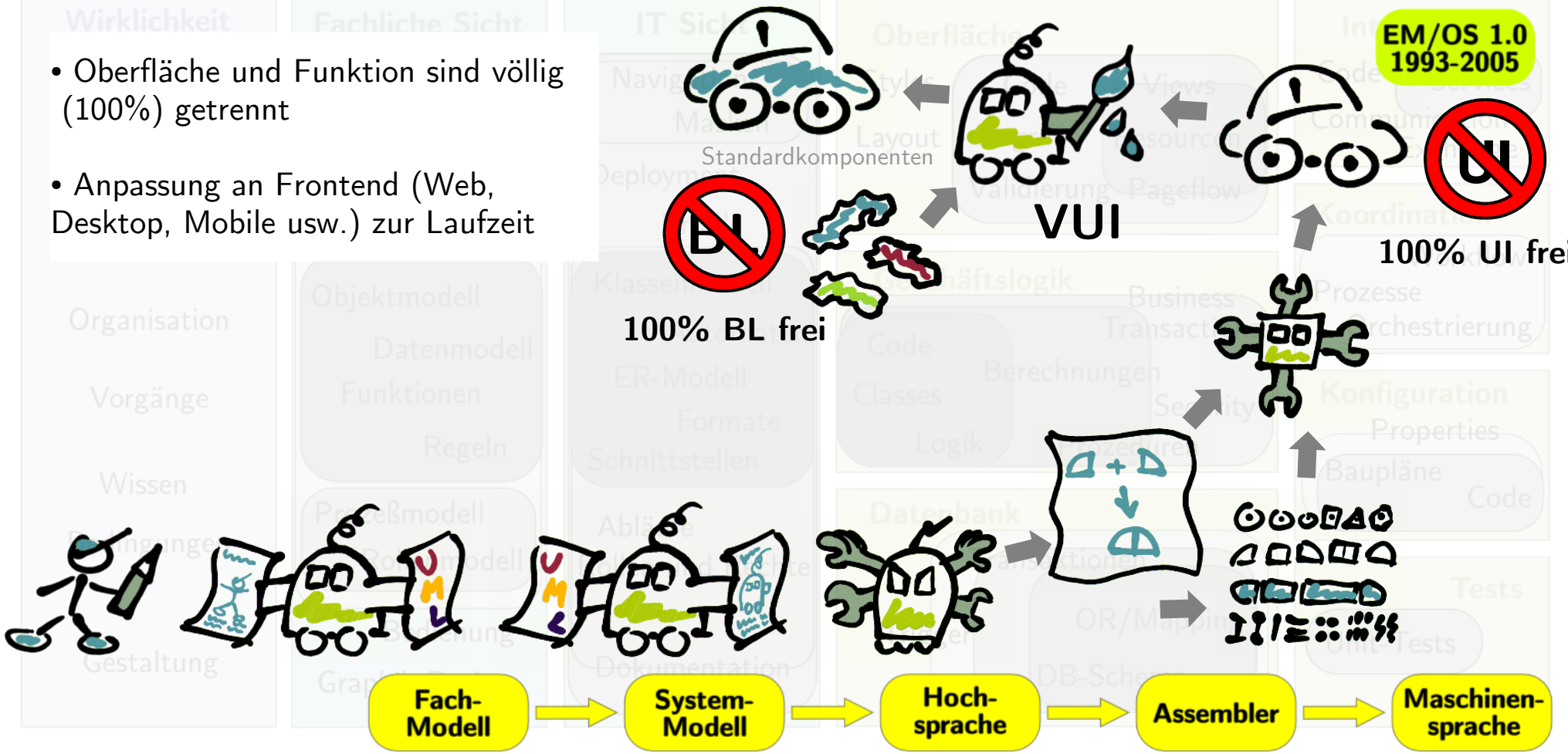


Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org



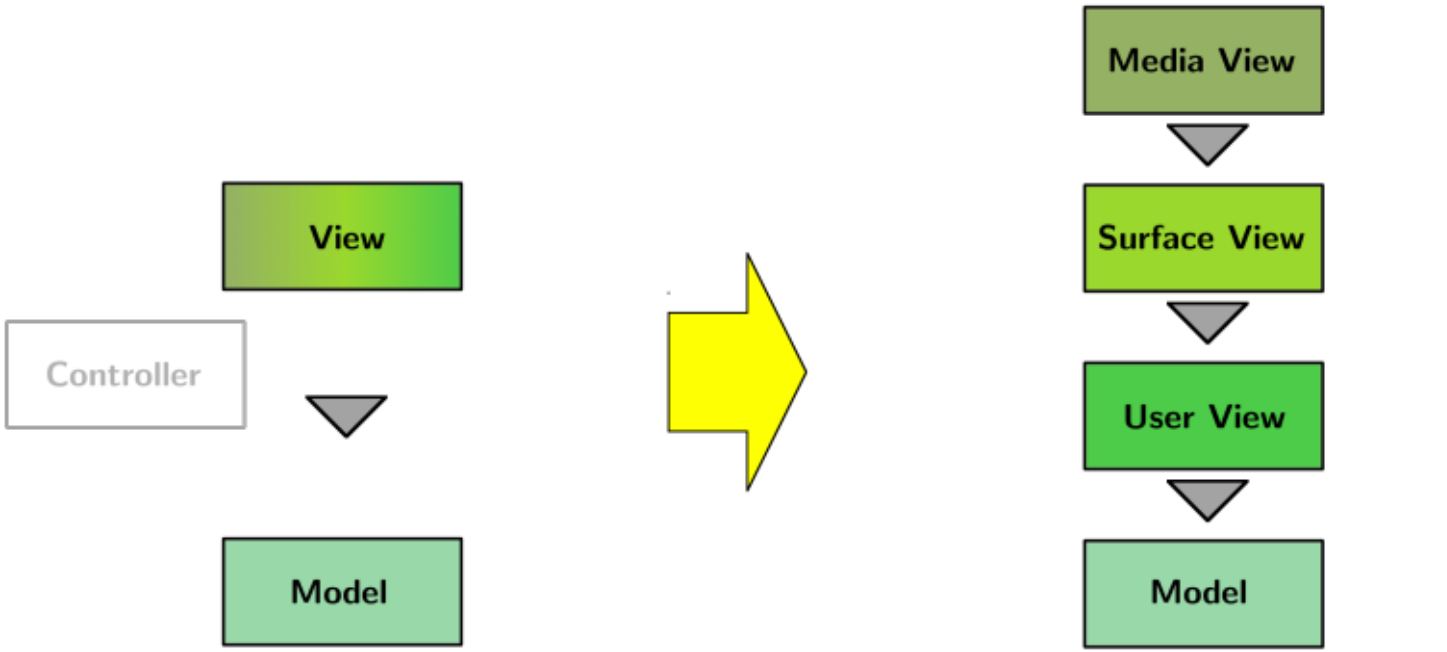
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

- Oberfläche und Funktion sind völlig (100%) getrennt
- Anpassung an Frontend (Web, Desktop, Mobile usw.) zur Laufzeit



Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

M3V Muster – 1 Model, 3 Views

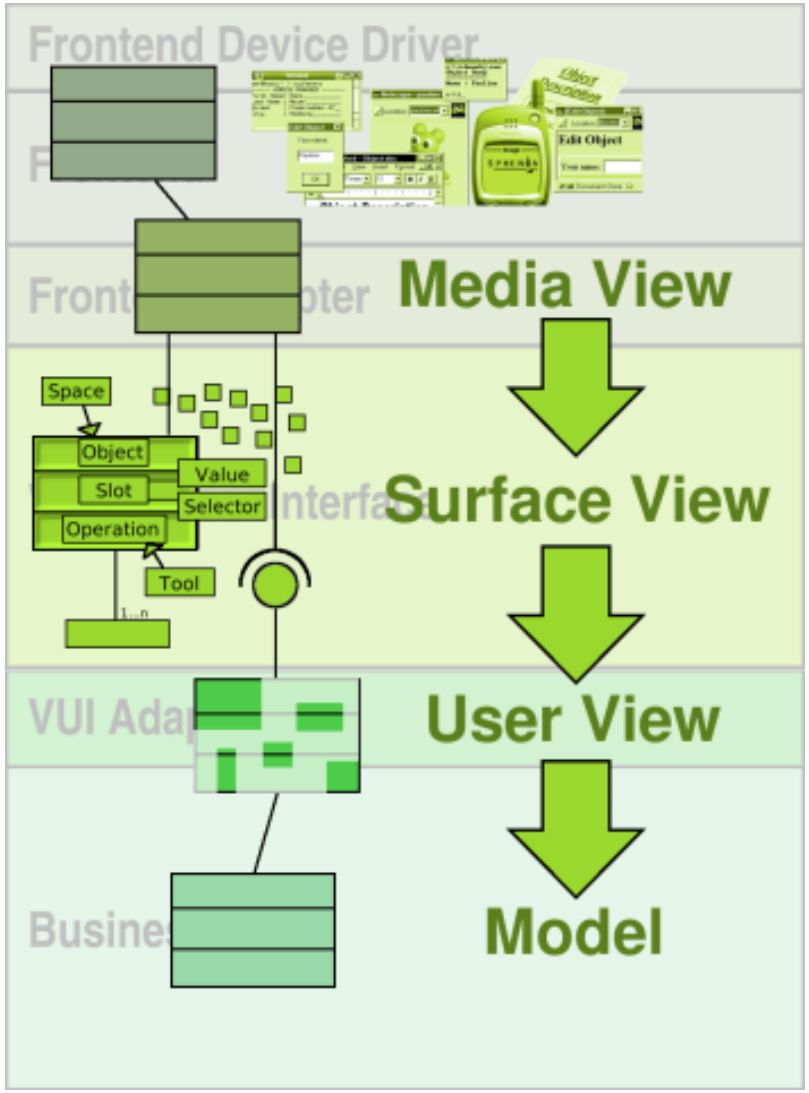


MVC
Model-View-Controller

M3V
Model-View-View-View

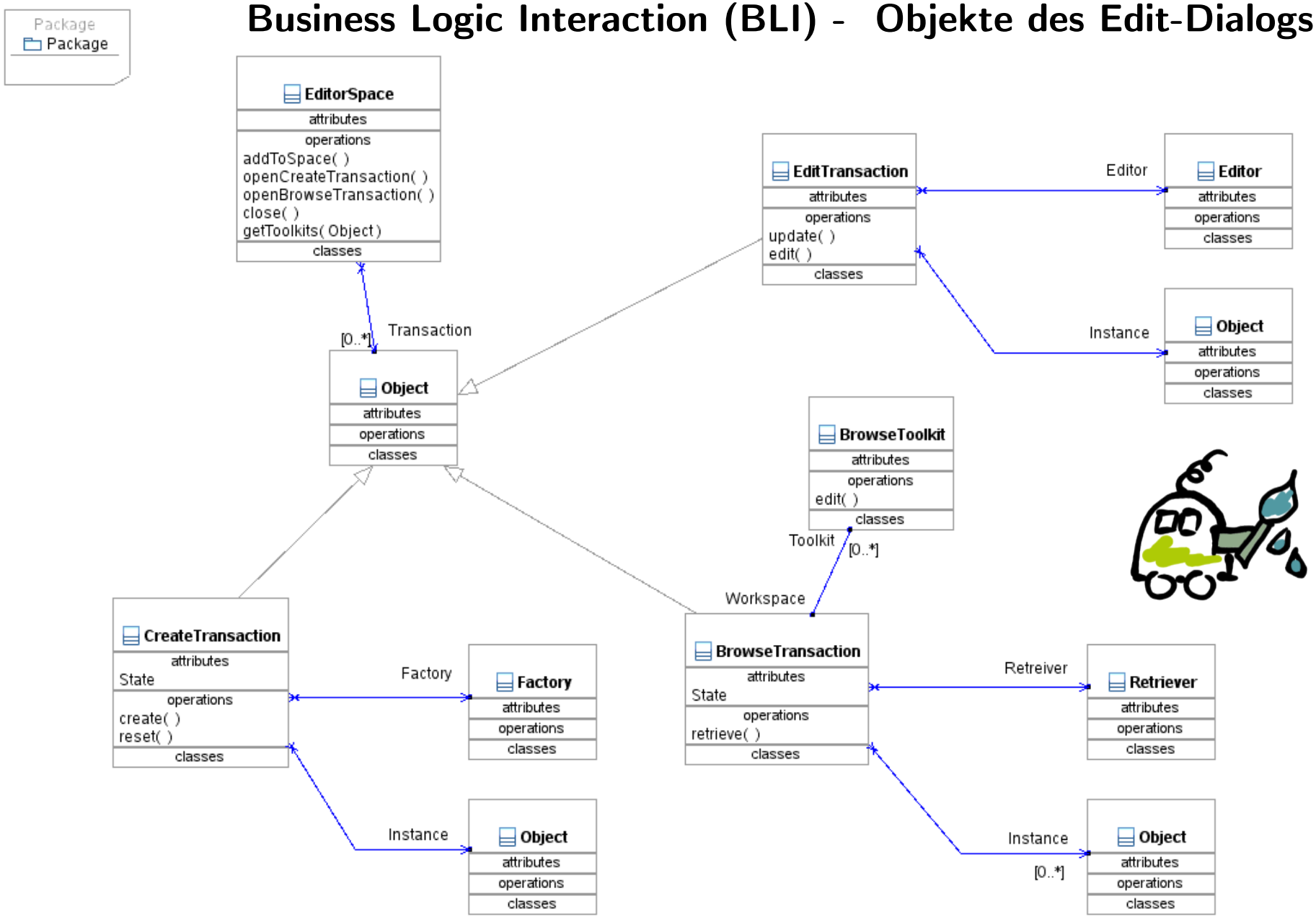
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

M3V Muster – 1 Model, 3 Views



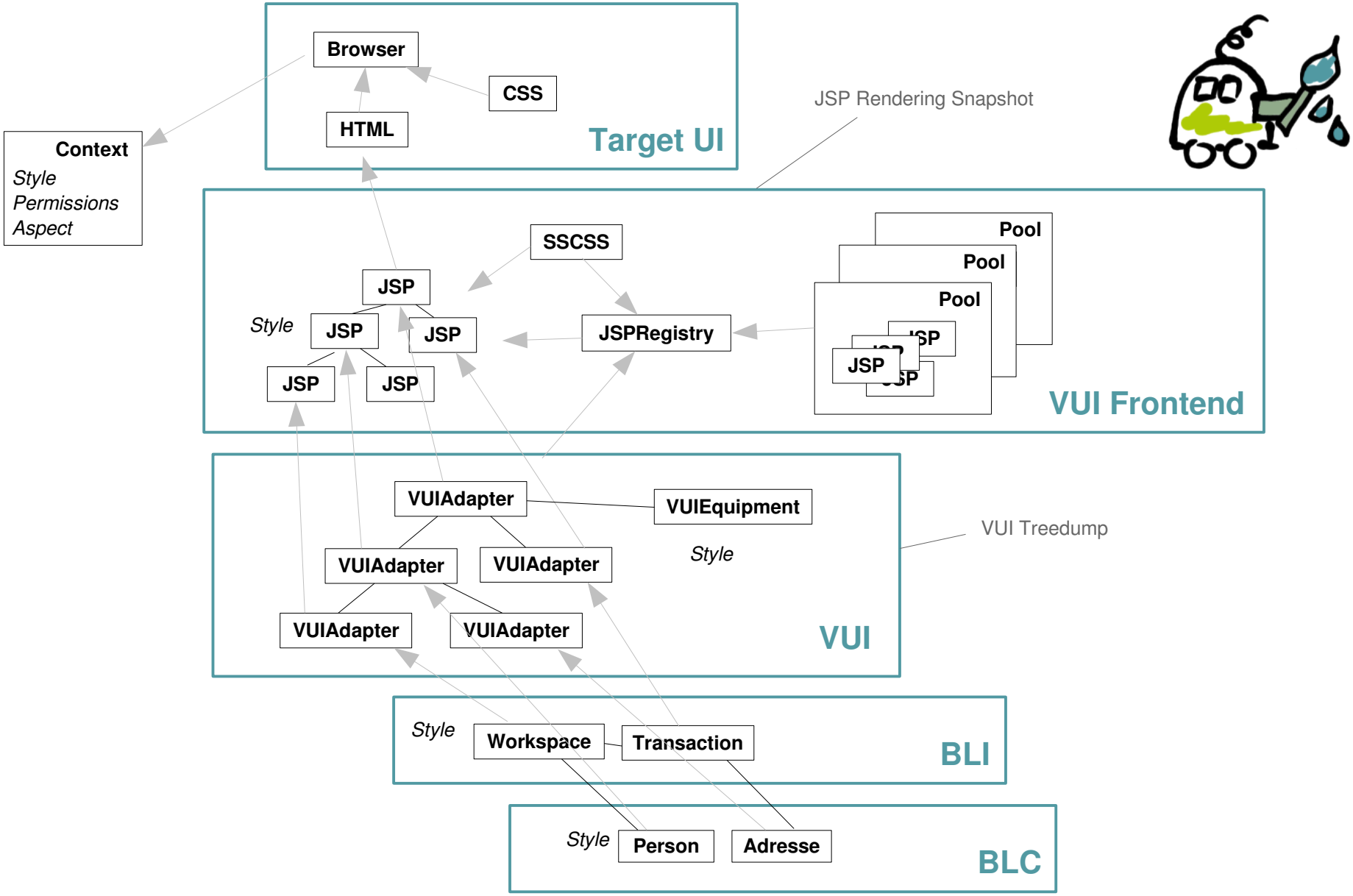
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Business Logic Interaction (BLI) - Objekte des Edit-Dialogs



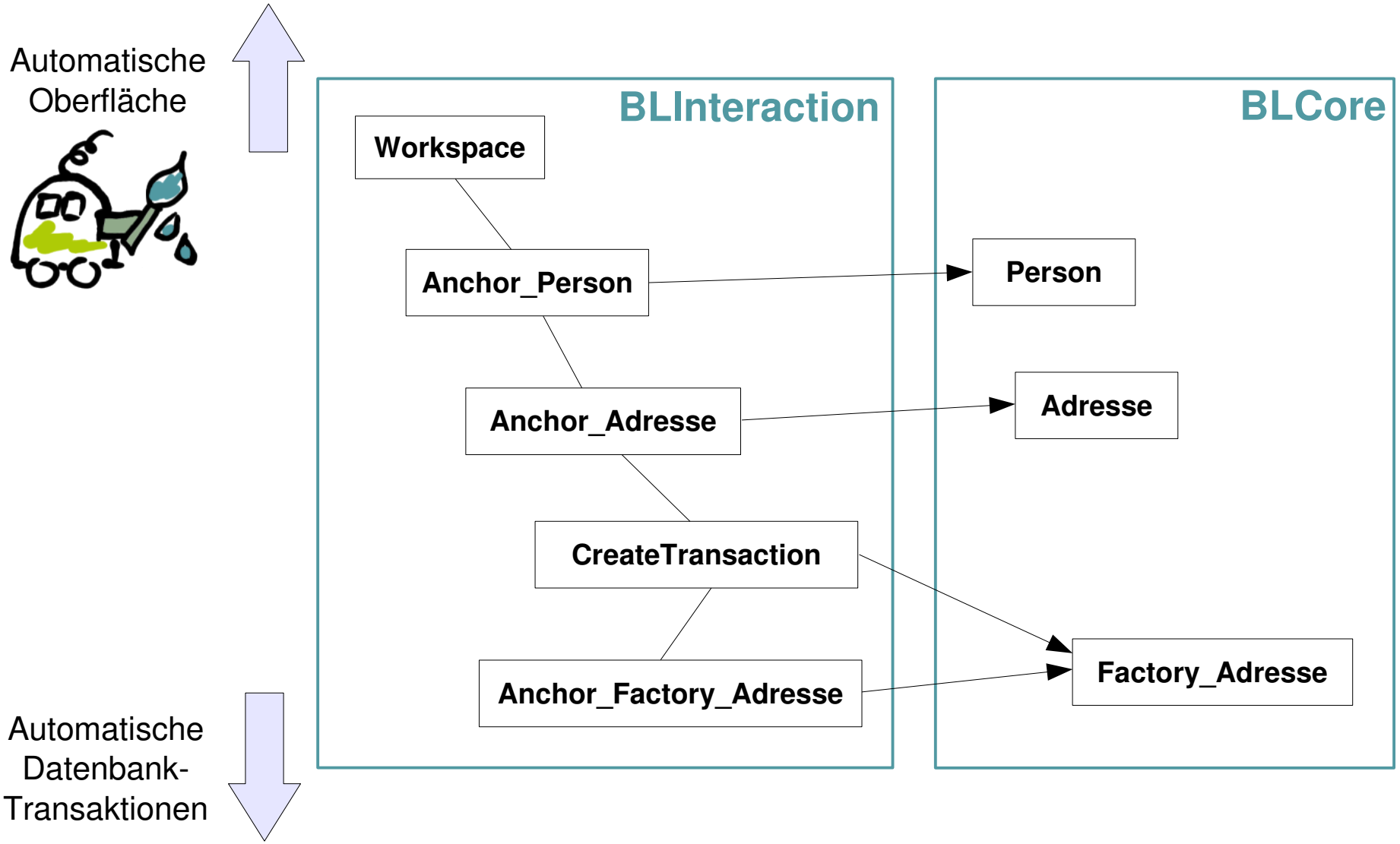
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Erzeugung der Oberfläche am Beispiel Web

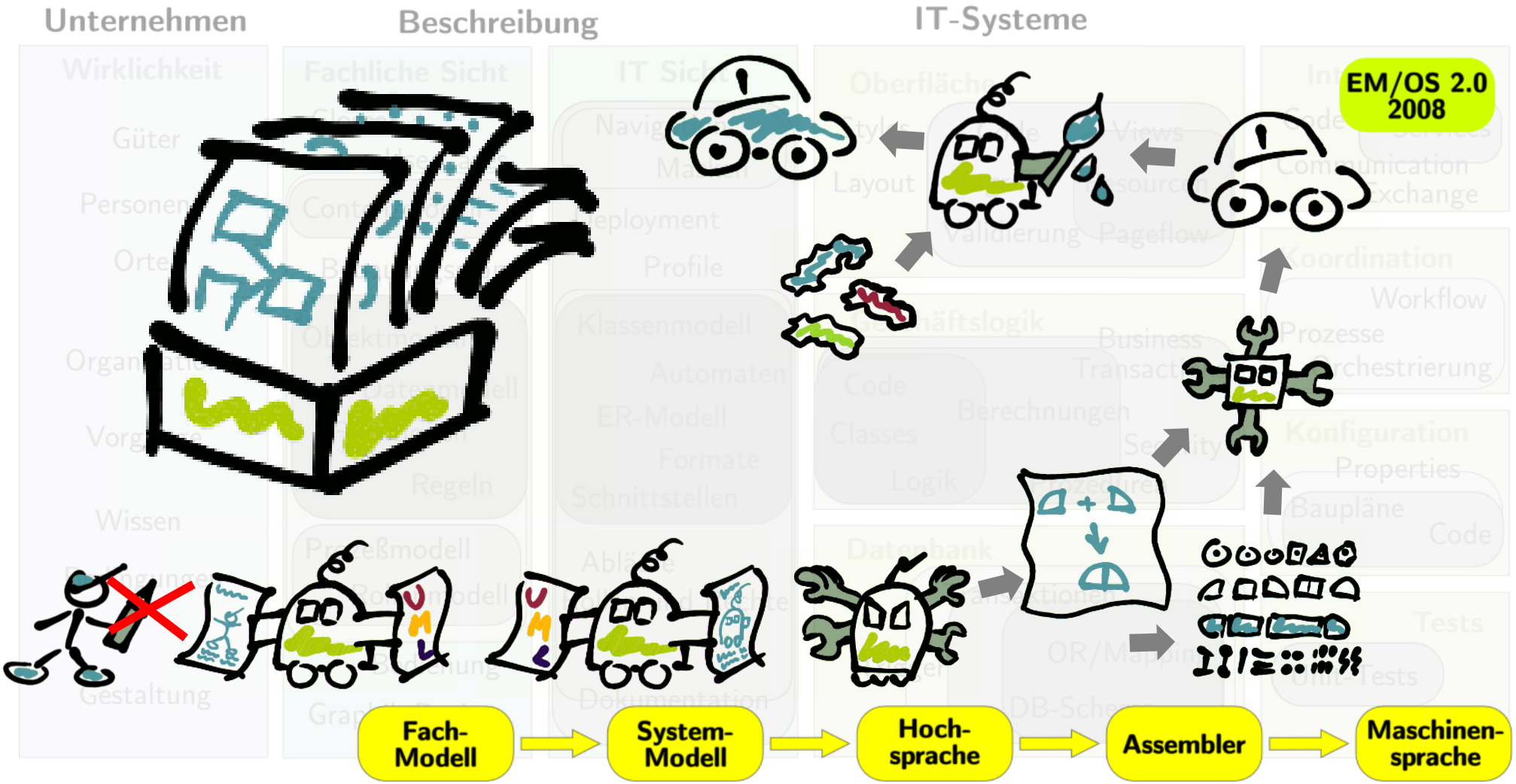


Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Business Logic Interaction (BLI) - Arbeits- und Transaktionsgerüst



Modell Repository - www.oomodels.org



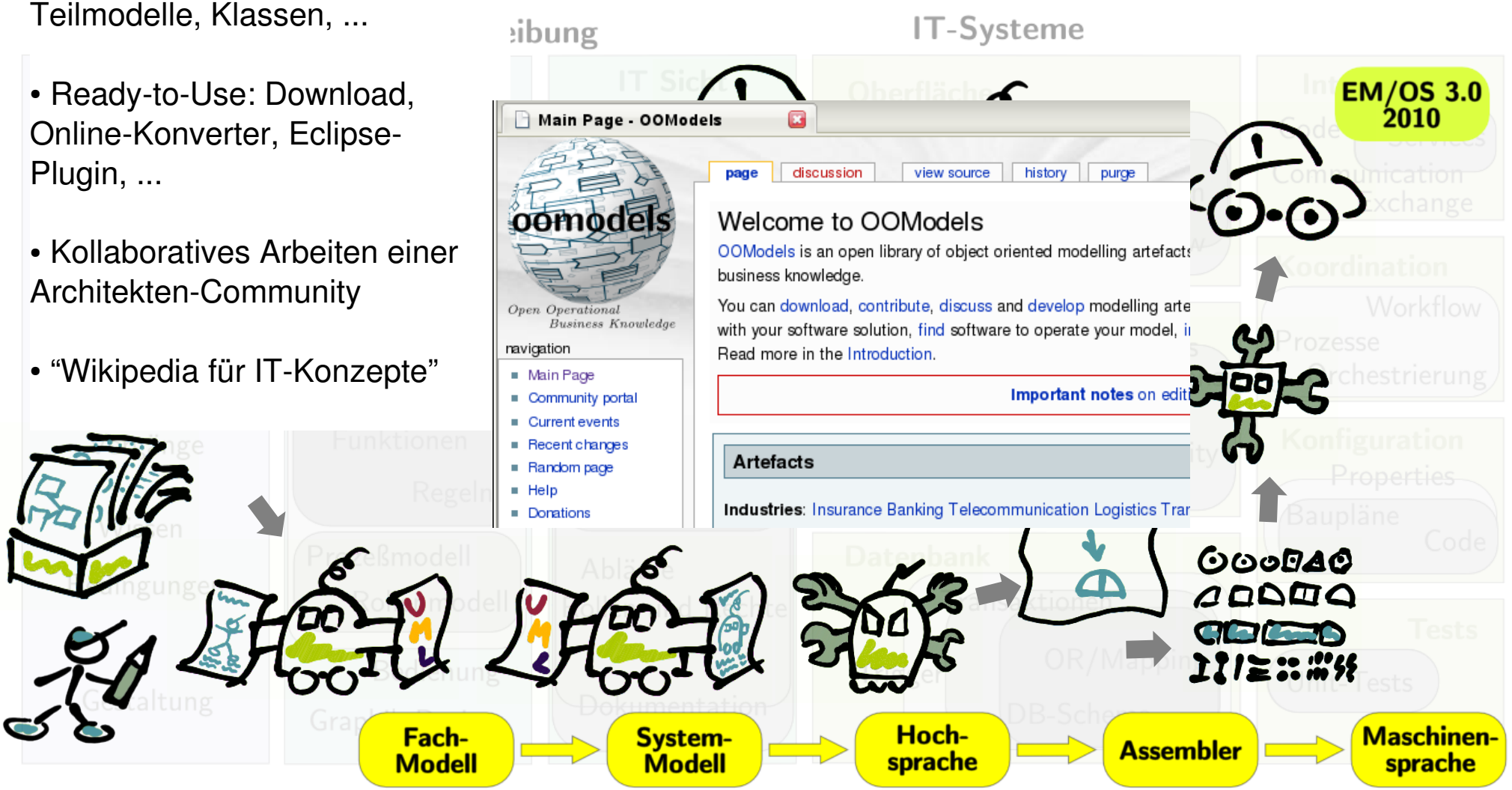
Modell Repository - www.oomodels.org

- Frei verfügbares Basis-Fachwissen: Modelle, Teilmodelle, Klassen, ...

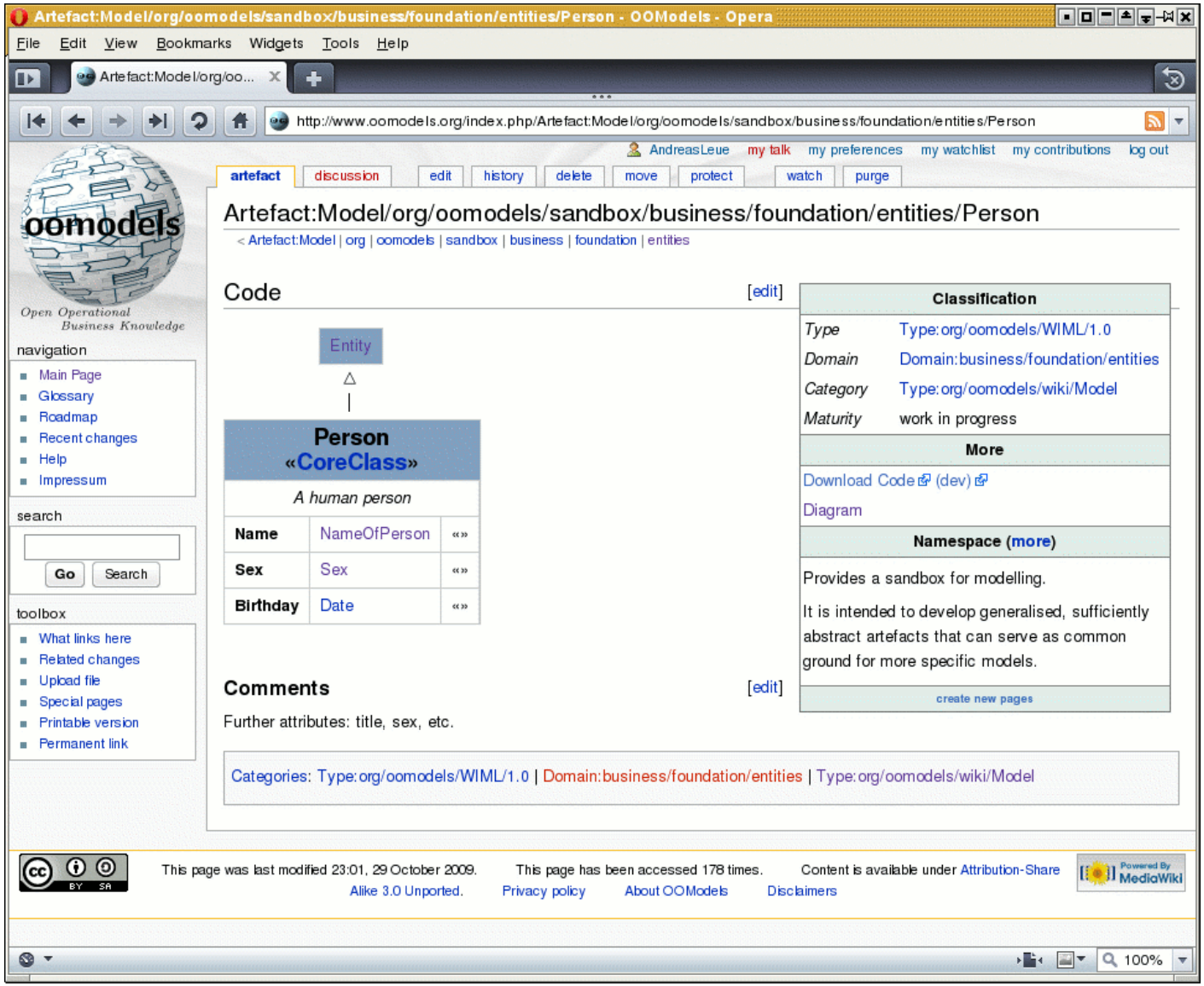
- Ready-to-Use: Download, Online-Konverter, Eclipse-Plugin, ...

- Kollaboratives Arbeiten einer Architekten-Community

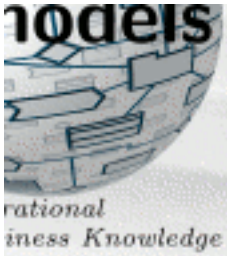
- “Wikipedia für IT-Konzepte”



Modell Repository - www.oomodels.org



Modell Repository - www.oomodels.org



age
ry
ap
changes
sum

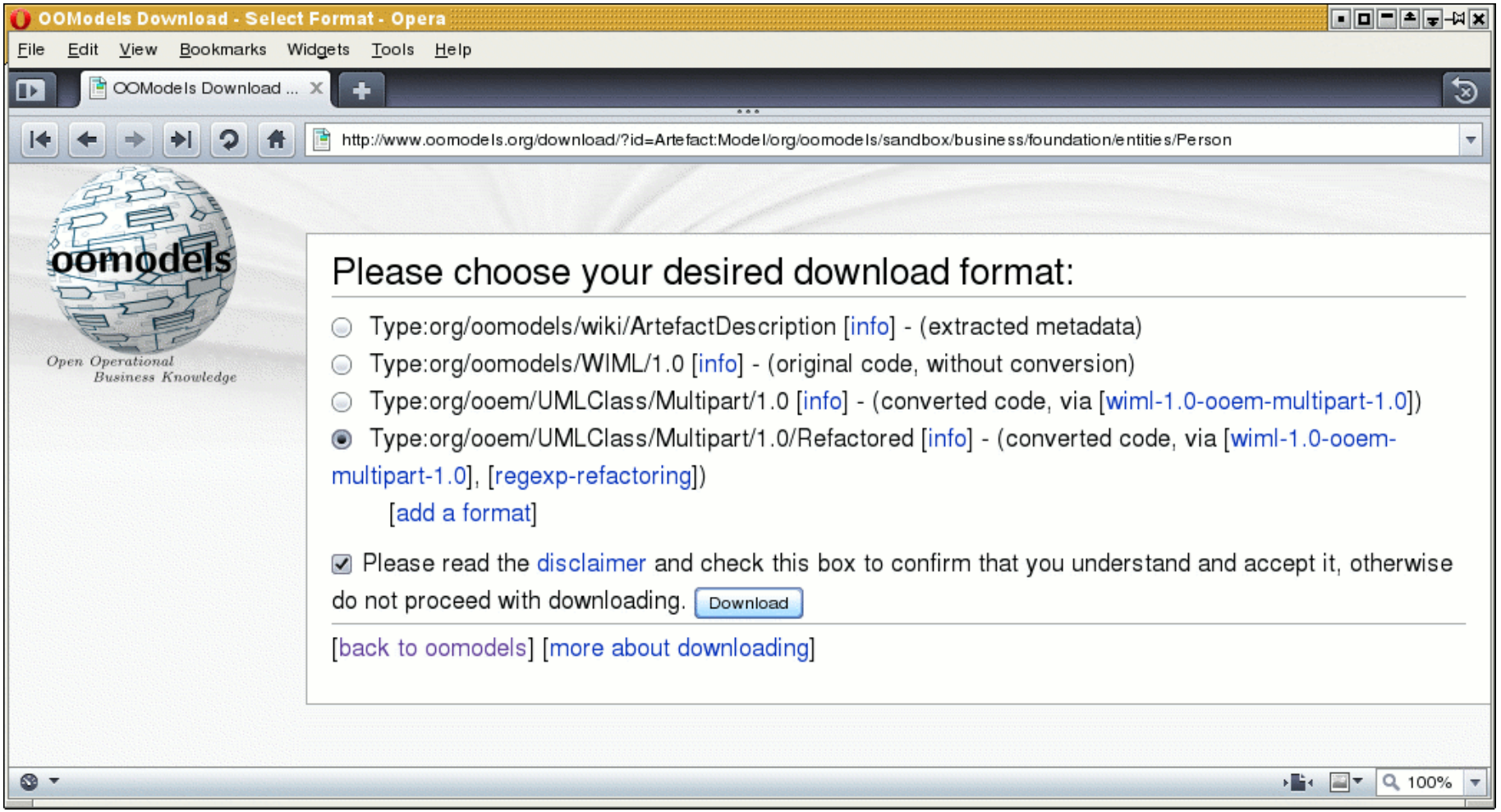
Search

ks here
changes
file
pages

Editing Artefact:Model/org/oomodels/sandbox/business/foundation/entities/Entities

```
{{Metabox
| Meta_Type     = Type:org/oomodels/WIML/1.0
| Meta_Domain   = Domain:business/foundation/entities
| Meta_Category = Type:org/oomodels/wiki/Model
| Meta_Download = true
| Meta_Diagram = Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities
| Meta_AddTypeToCategory     = true
| Meta_AddCategoryToCategory = true
| Meta_Maturity = work in progress
}}

== Code ==

<wiml>
* Person CoreClass
> org/oomodels/sandbox/business/foundation/entities/Entity
  A human person
** Name org/oomodels/sandbox/business/foundation/entities/NameOfPerson
** Sex org/oomodels/sandbox/business/foundation/entities/Sex
** Birthday Date
</wiml>
```

Modell Repository - www.oomodels.org



Modell Repository - www.oomodels.org

Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities - OOModels - Opera

File Edit View Bookmarks Widgets Tools Help

http://www.oomodels.org/index.php/Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities

AndreasLeue my talk my preferences my watchlist my contributions log out

artefact | discussion | edit | history | delete | move | protect | watch | purge

oomodels — Open Operational Business Knowledge

navigation
- Main Page
- Glossary
- Roadmap
- Recent changes
- Help
- Impressum

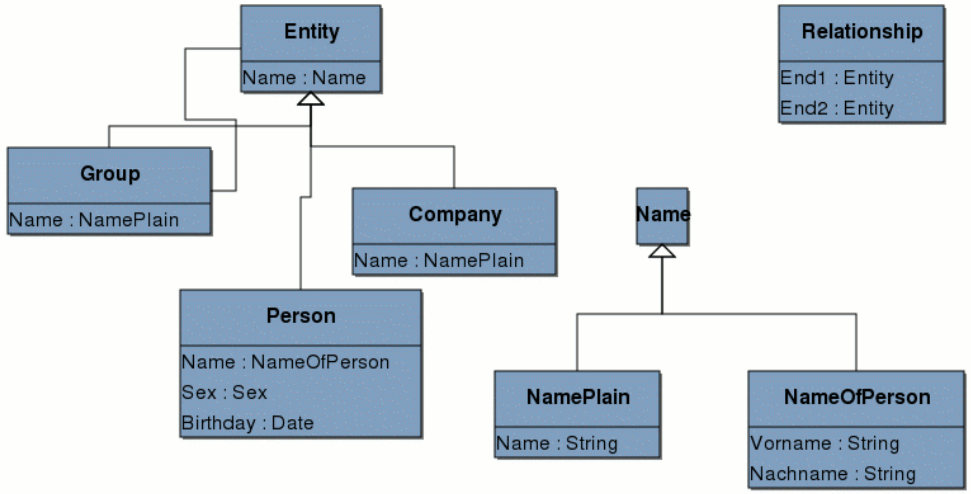
search: Go | Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities

< Artefact:Diagram | org | oomodels | sandbox | business | foundation | entities

Code [edit]



Classification	
Type	Type:org/oomodels/WIDL/1.0
Domain	Domain:it/test
Category	Type:org/oomodels/wiki/Diagram

More

Download Code (dev)

Namespace (more)

Provides a sandbox for modelling.

It is intended to develop generalised, sufficiently abstract artefacts that can serve as common ground for more specific models.

create new pages

Categories: Type:org/oomodels/WIDL/1.0 | Domain:it/test | Type:org/oomodels/wiki/Diagram

This page was last modified 20:31, 8 November 2009. This page has been accessed 40 times. Content is available under Attribution-Share Alike 3.0 Unported. Privacy policy About OOModels Disclaimers Powered By MediaWiki

100%

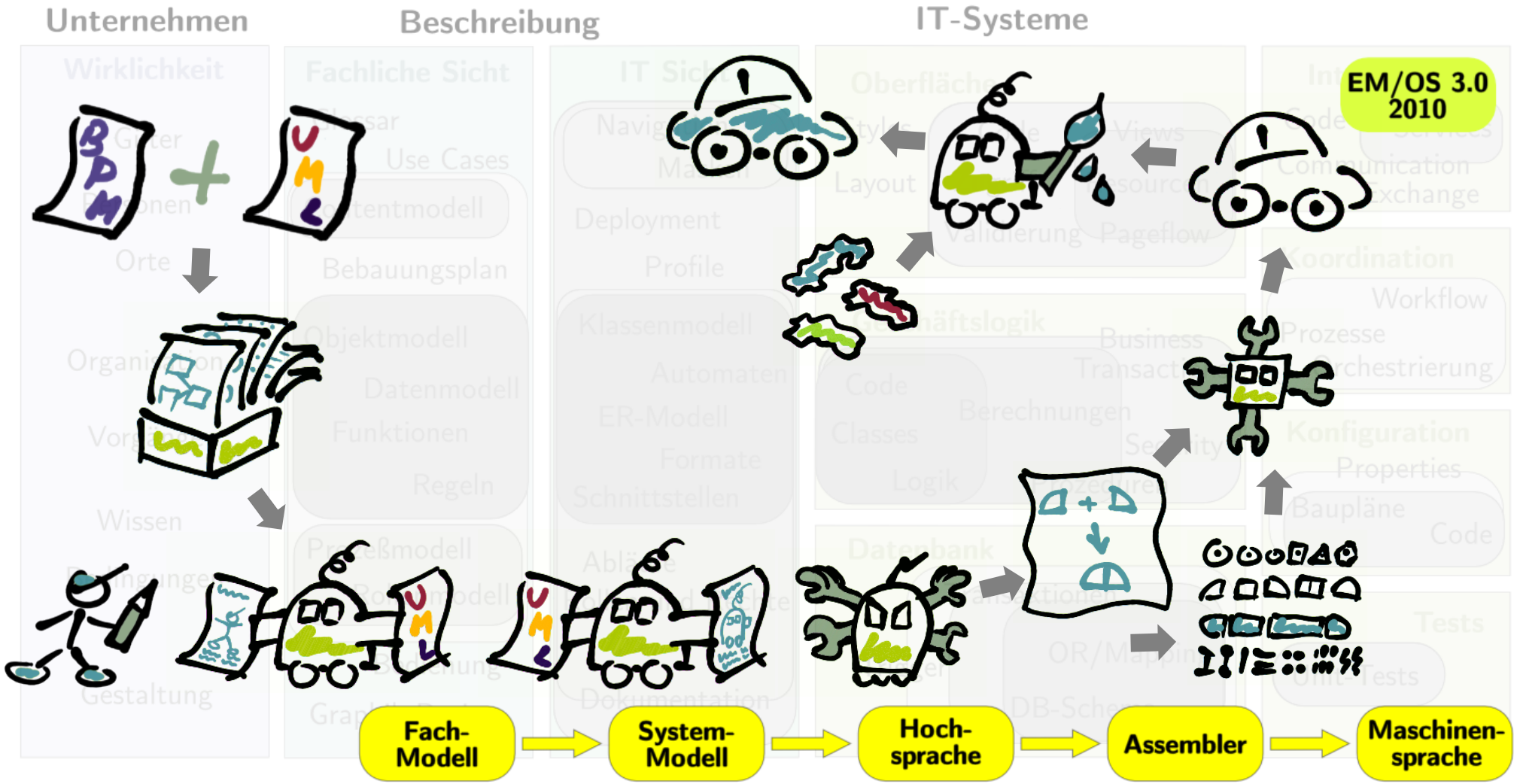
Modell Repository - www.oomodels.org

Editing Artefact:Diagram/org/oomodels/sandbox/business/foundat

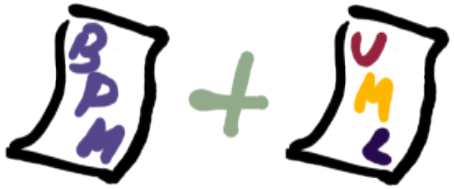
```
| Meta_Category          = Type:org/oomodels/wiki/Diagram
| Meta_Download          = true
| Meta_AddTypeToCategory    = true
| Meta_AddCategoryToCategory = true
}}

== Code ==

<widl>

                ./Entity                        ./Relationship


 ./Group

                        ./Company           ./Name


            ./Person

                                ./NamePlain         ./NameOfPerson


</widl>
```

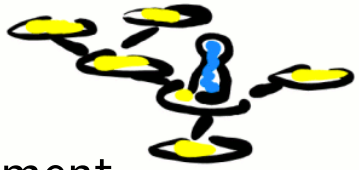
Please note that all contributions to OOModels are considered to be released under the Attribution-S

Unifizierte BPM/IT-Modelle - www.ooem.org / www.ubpml.org



Unifizierte BPM/IT-Modelle - www.ooem.org / www.ubpml.org





Strategie

Management

Organisation

Zielvorgaben

Planung

zu schwammig

zu realitätsfern

Management Model

zu technisch

Hohe Präzision

Automatisierung

System Model

System Architect

zu ineffektiv

Process Model

zu kompliziert

Sämtliche Eventualitäten

Koordination

Checklisten

Mitarbeiter

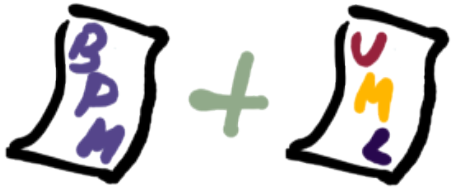
Arbeitspensum

zu ungenau, zu lückenhaft

Durchführung

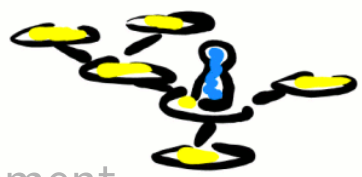
Konkrete Probleme lösen

Unifizierte BPM/IT-Modelle - www.ooem.org / www.ubpml.org

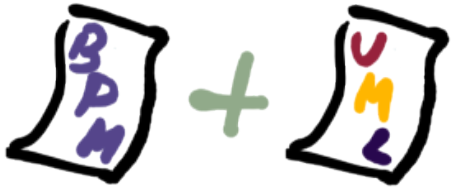


- Klassen in's Modell
  (Zustandbehaftete Klassen und Objekte)
- Weg mit dem Kontrollfluß
  (Konstellationen als kausale Zusammenhänge)
- Entkopplung der Modellebenen durch OO
  (Steps als Objekte, Prozeduren assoziiert)

Unifizierte BPM/IT-Modelle - www.ooem.org / www.ubpml.org

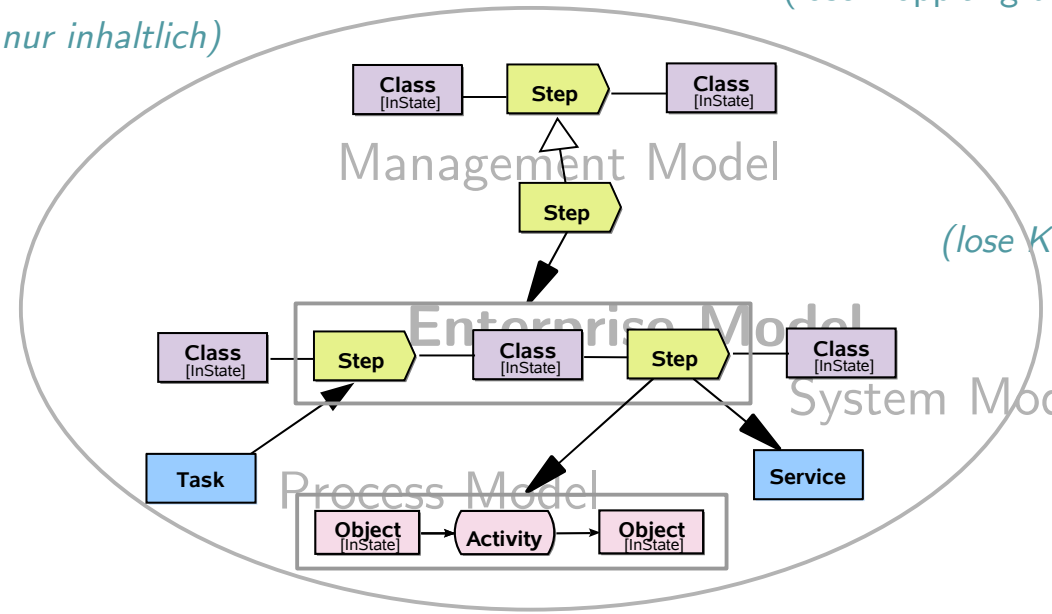


Management



Kein Kontrollfluß
(lose Kopplung: nur inhaltlich)

Vererbung
(lose Kopplung durch Abstraktion)



Verfeinerung
(lose Kopplung durch dynamische Zuordnung)

Mitarbeiter

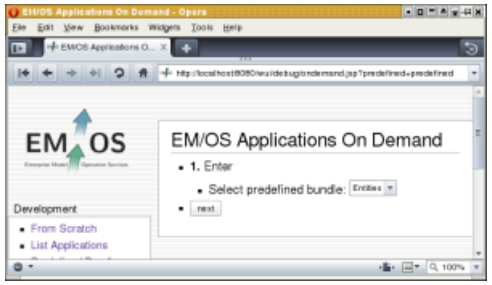
System Architect

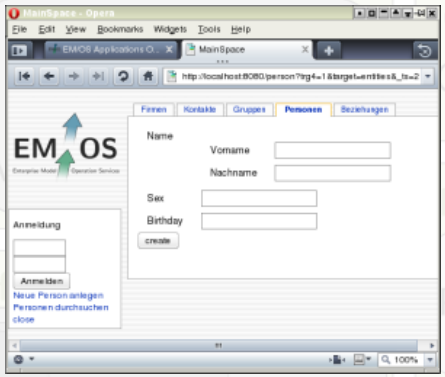
IT Services
(lose Kopplung durch dynamische Zuordnung)

Offen für Erweiterungen
(lose Kopplung durch Prozedur-Abstraktion)

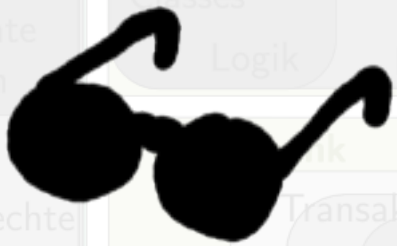
EM/OS 2.0 - Hausaufgaben

- Refactoring & Cleanup
- Usability & First Experience
- 100% Java
- Automation



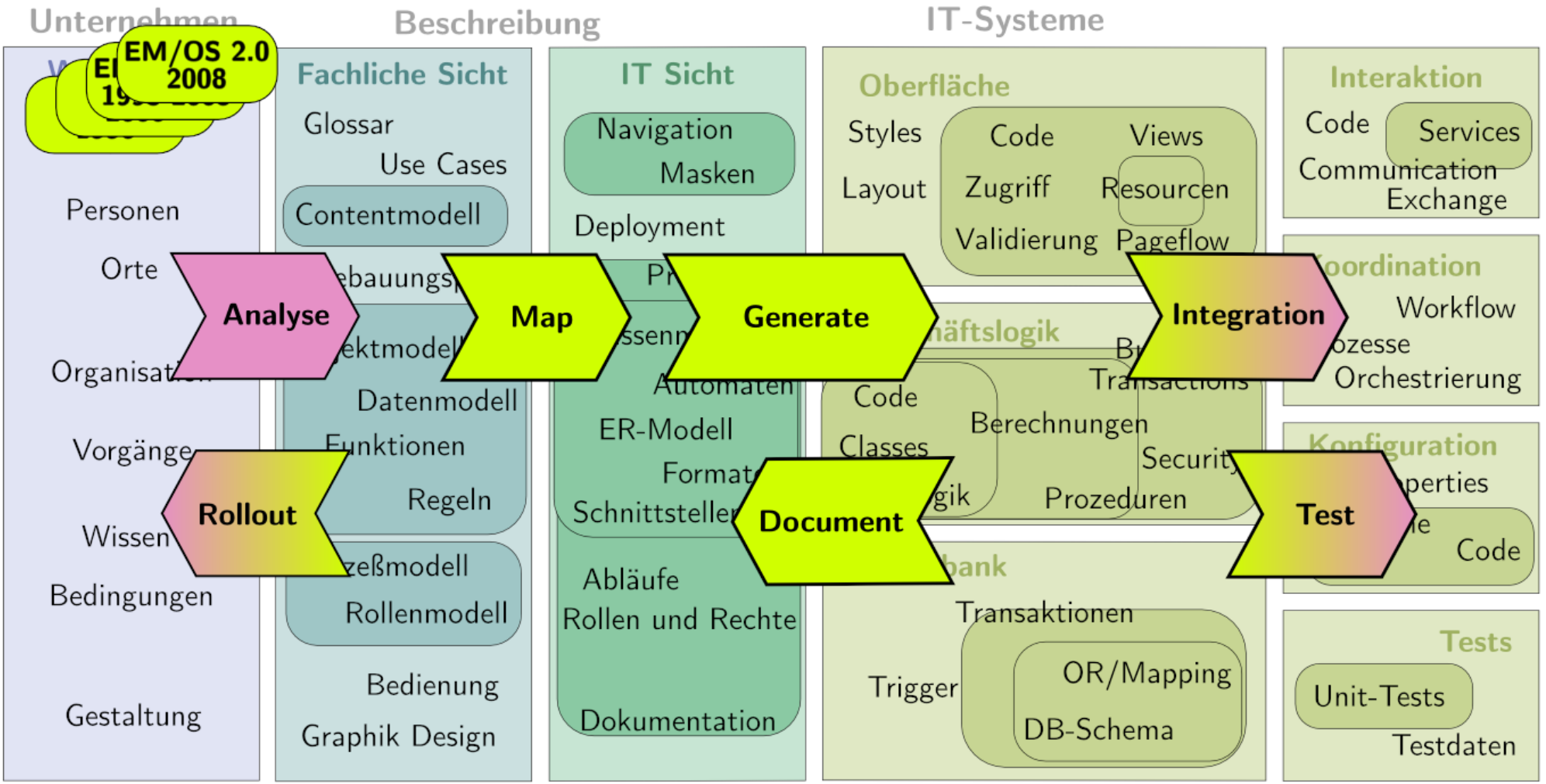


EM/OS 2.0
2008

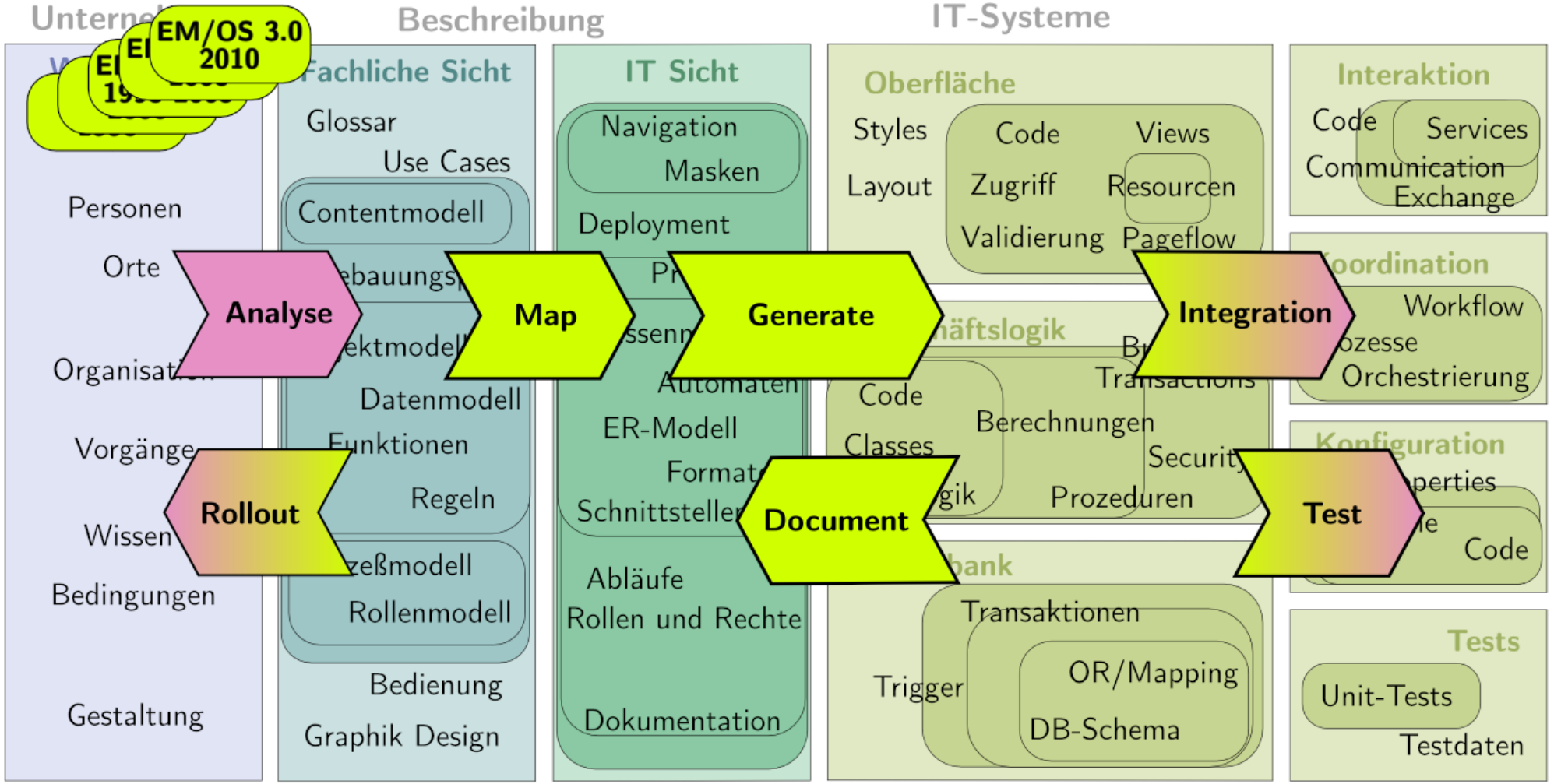


Fach-Modell → System-Modell → Hoch-sprache → Assembler → Maschinen-sprache

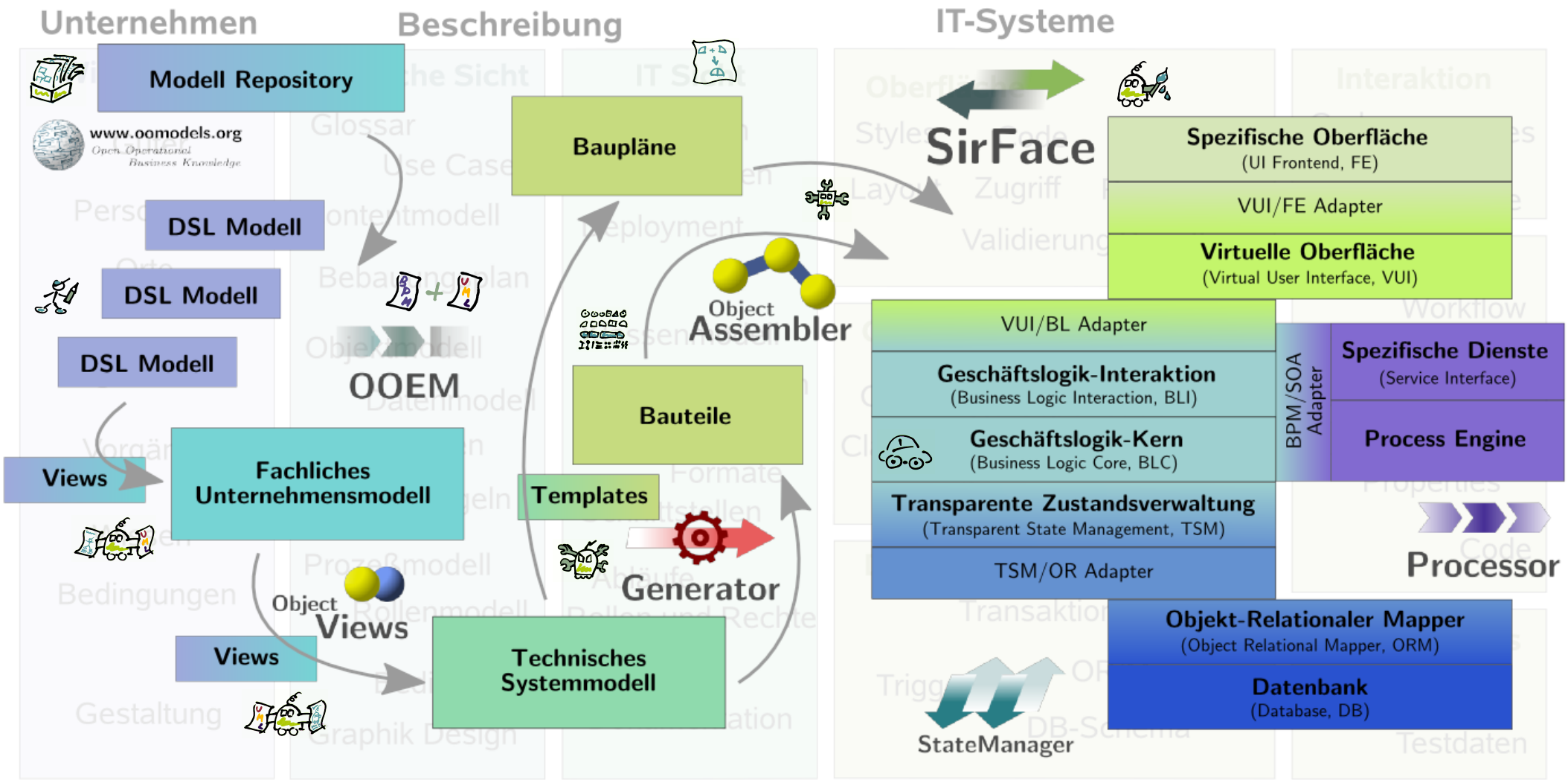
EM/OS 2008


Unternehmen
Beschreibung
IT-Systeme
EM/OS 2.0
2008
Fachliche Sicht
Glossar
Use Cases
Contentmodell
Datenmodell
Funktionen
Regeln
Rollenmodell
Bedienung
Graphik Design
IT Sicht
Navigation
Masken
Deployment
Automaten
ER-Modell
Schnittstellen
Abläufe
Rollen und Rechte
Dokumentation
Oberfläche
Styles
Code
Views
Layout
Zugriff
Resourcen
Validierung
Pageflow
Code
Berechnungen
Classes
Prozeduren
Transaktionen
Trigger
OR/Mapping
DB-Schema
Interaktion
Code
Services
Communication
Exchange
Workflow
Orchestrierung
Konfiguration
Code
Tests
Unit-Tests
Testdaten
Personen
Orte
Organisation
Vorgänge
Wissen
Bedingungen
Gestaltung
Analyse
Map
Generate
Integration
Rollout
Document
Test

EM/OS 2010


EM/OS 3.0
2010
Beschreibung
IT-Systeme
Fachliche Sicht
IT Sicht
Oberfläche
Interaktion
Glossar
Use Cases
Contentmodell
Personen
Orte
Organisation
Vorgänge
Wissen
Bedingungen
Gestaltung
Datenmodell
Funktionen
Regeln
Rollenmodell
Bedienung
Graphik Design
Navigation
Masken
Deployment
Automaten
ER-Modell
Schnittstellen
Abläufe
Rollen und Rechte
Dokumentation
Styles
Layout
Code
Zugriff
Validierung
Views
Resourcen
Pageflow
Code
Classes
Berechnungen
Prozeduren
Transaktionen
Trigger
OR/Mapping
DB-Schema
Code
Communication
Services
Exchange
Workflow
Orchestrierung
Code
Unit-Tests
Testdaten
Analyse
Map
Generate
Integration
Rollout
Document
Test

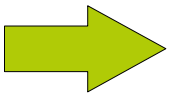
EM/OS Architektur



Fazit

EM/OS ist

- mehr als MDA und kein Vergleich mehr zu “Generatoren”
- praxiserprobt und gereift

 industrielle Software-Produktion

Sphenon

Inhalt

Ende

Interesse an EM/OS?

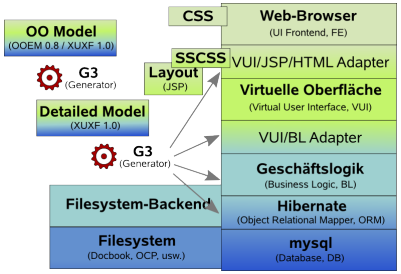
- Einfach ausprobieren:
  www.oomodels.org
  www.xocp.org
  www.oogenerator.org
- Produktvorführung
  "OnDemand Applications für Web & Desktop"
- usw.

Projekte mit EM/OS

Produktportal

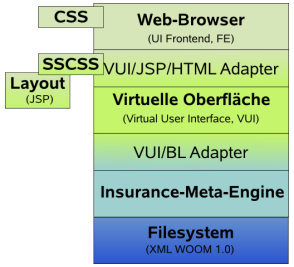
SKYMASTER

- Umfangreicher Katalog
- Online CMS
- Produkt-Assistent



Online Versicherungsmakler Plattform

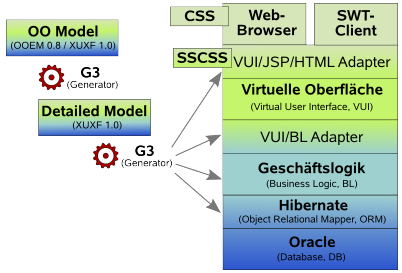
BOSCH

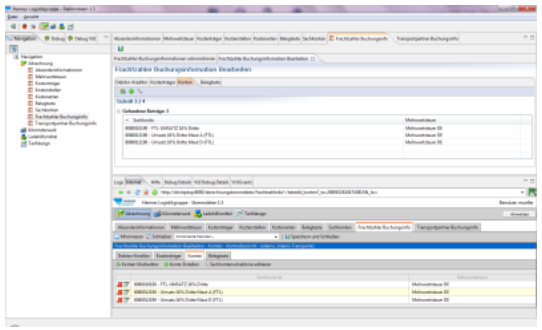


- Interpretiertes Modell
- incl. Content
- pixelgenaue Layout Vorgabe

Stammdaten-Administration

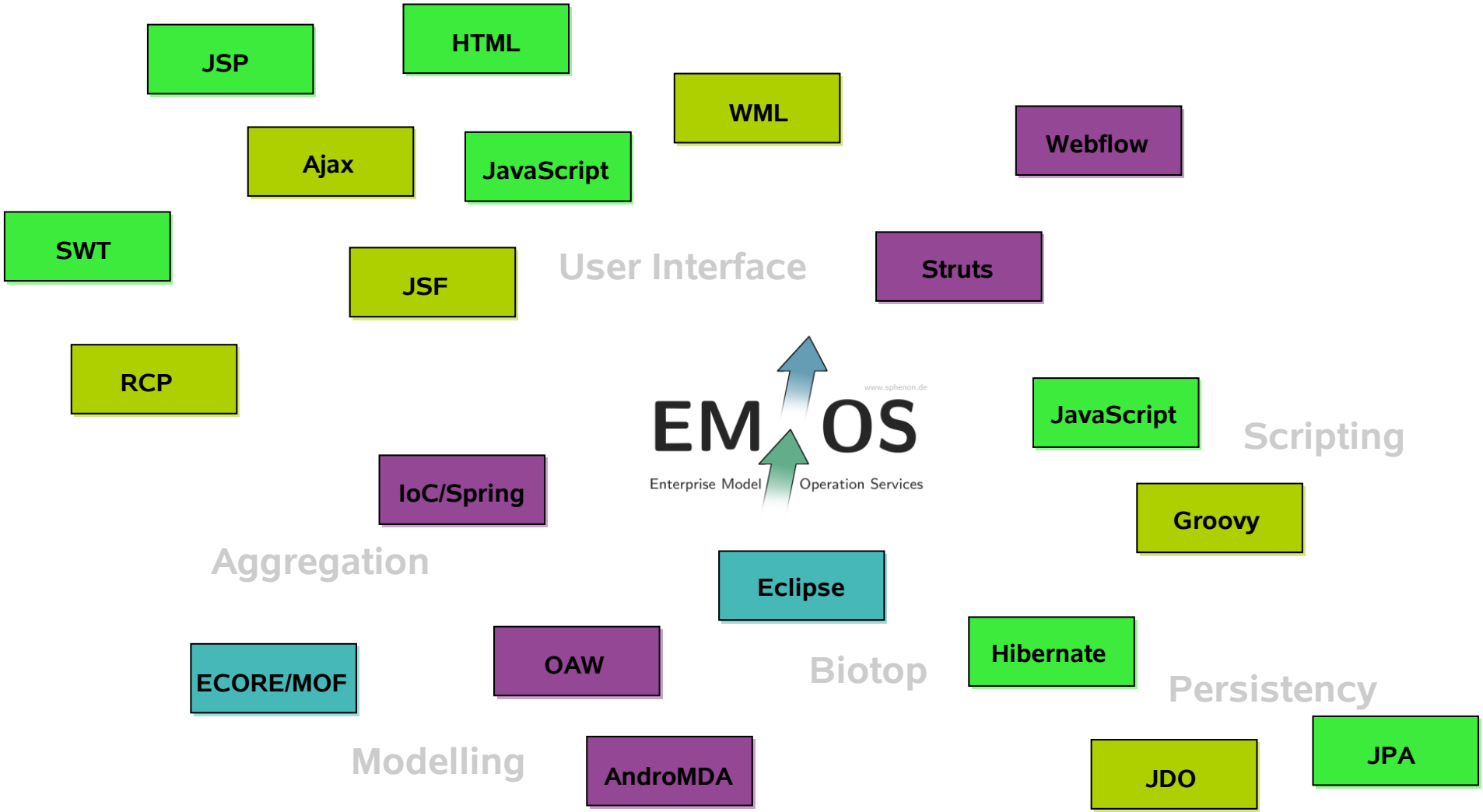
Hermes





- Legacy Datenbanken
- pixelgenaue Layout Vorgabe
- Web-Client & SWT-Client

EM/OS & OSS


JSP
HTML
WML
Webflow
Ajax
JavaScript
SWT
JSF
User Interface
Struts
RCP
www.sphenon.de
EM OS
Enterprise Model Operation Services
JavaScript
Scripting
IoC/Spring
Groovy
Aggregation
Eclipse
OAW
Biotop
Hibernate
ECORE/MOF
Persistency
JPA
Modelling
AndroMDA
JDO