buschmais

Beratung . Technologie . Innovation

Java User Group Saxony

# Modellierung statischer Domänenmodelle mit Xtext

Frank Schwarz

buschmais GbR

Inhaber

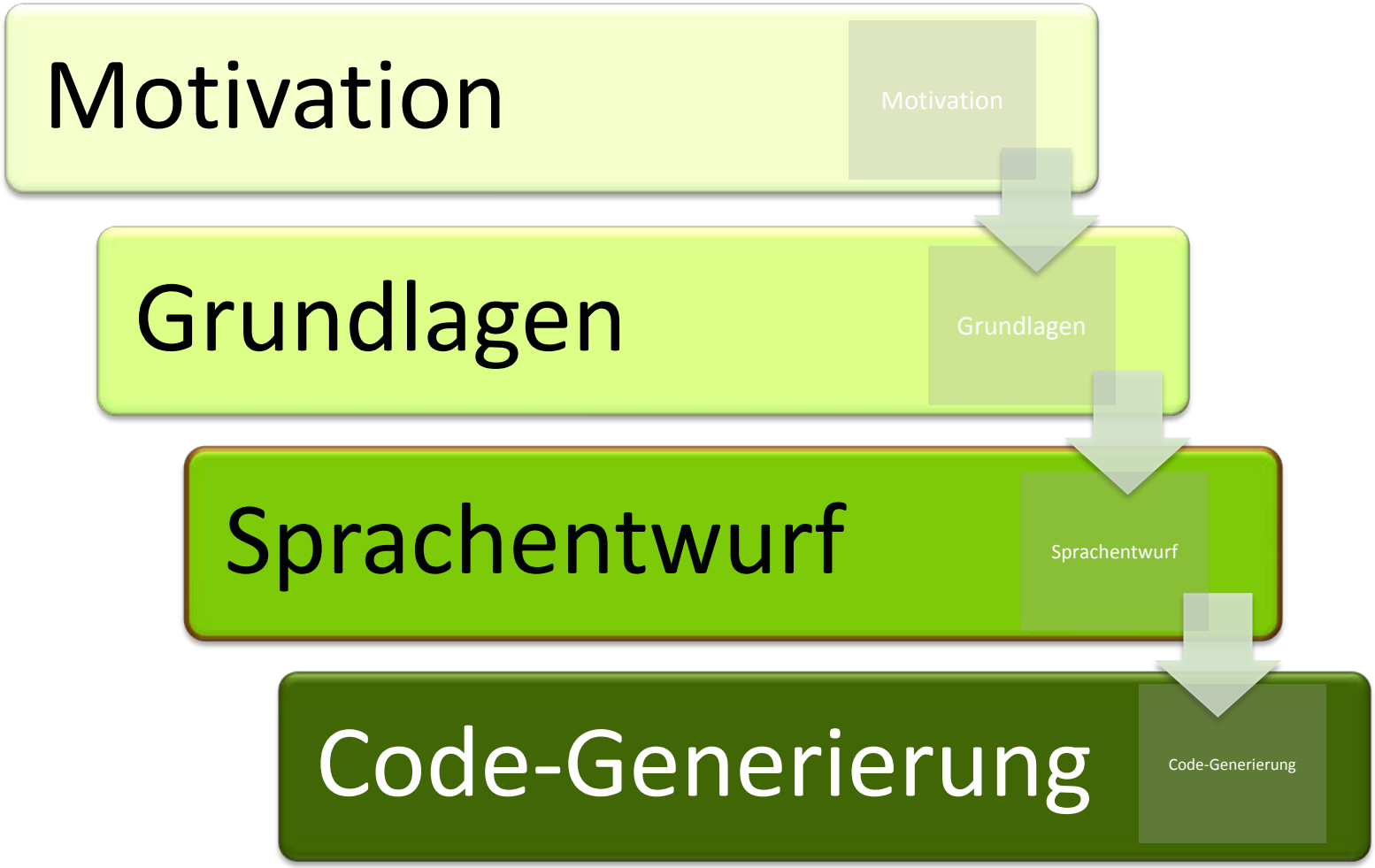
Torsten Busch, Frank Schwarz,
Dirk Mahler und Tobias Israel

frank.schwarz@buschmais.de
http://www.buschmais.de/author/frank

Leipzig, 15. Januar 2009

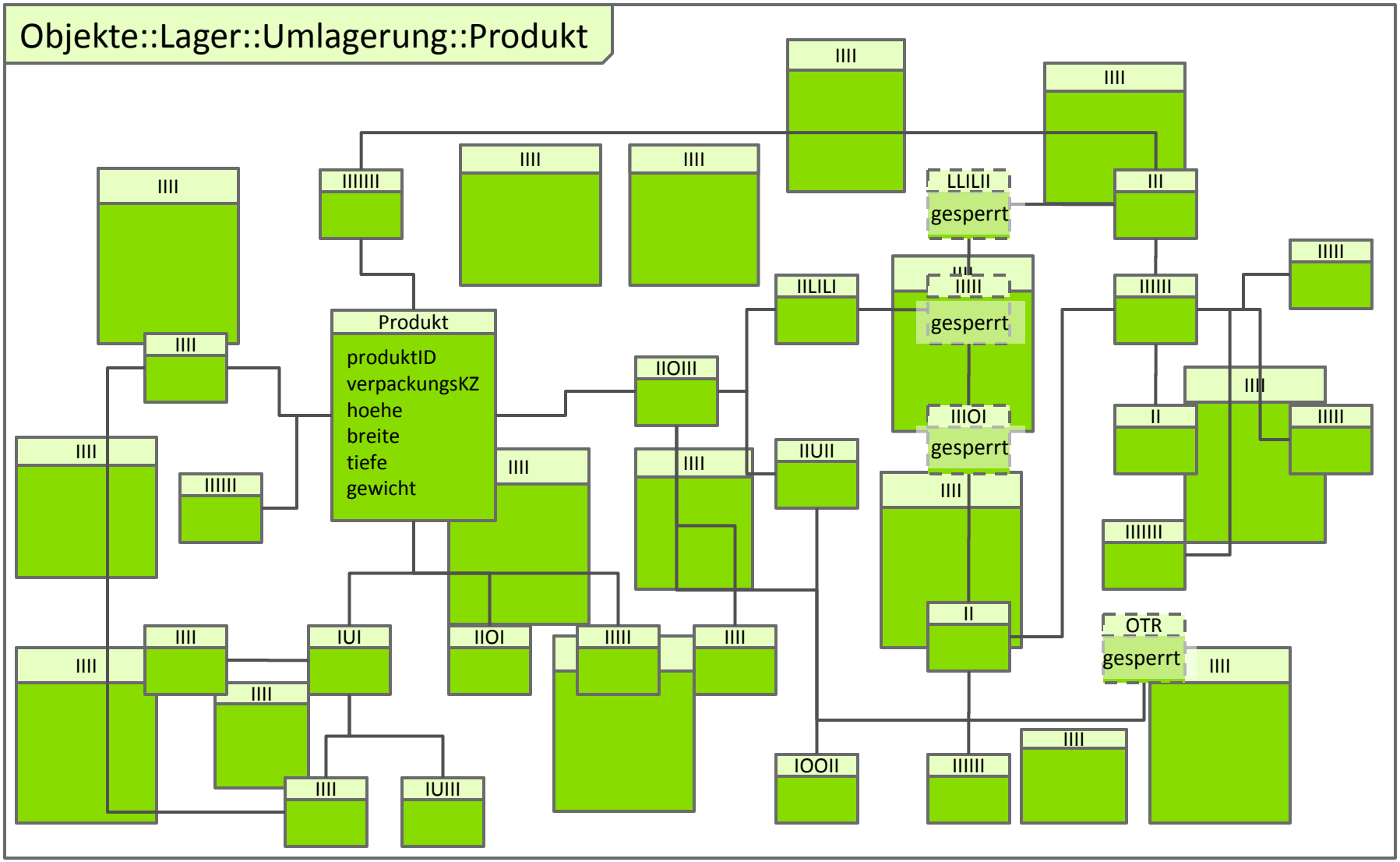
# Modellierung statischer Domänenmodelle mit Xtext

buschmais
Beratung . Technologie . Innovation



# Die Hölle des Modellierers

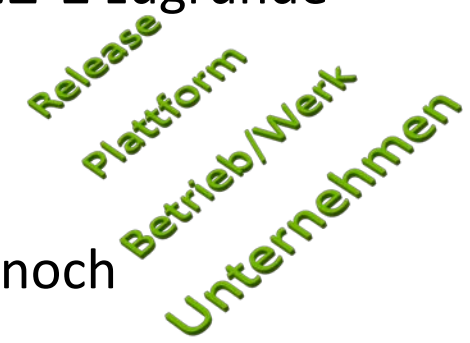
buschmais
Beratung . Technologie . Innovation



# Die Hölle des Konfigurationsmanagers

buschmais
Beratung . Technologie . Innovation

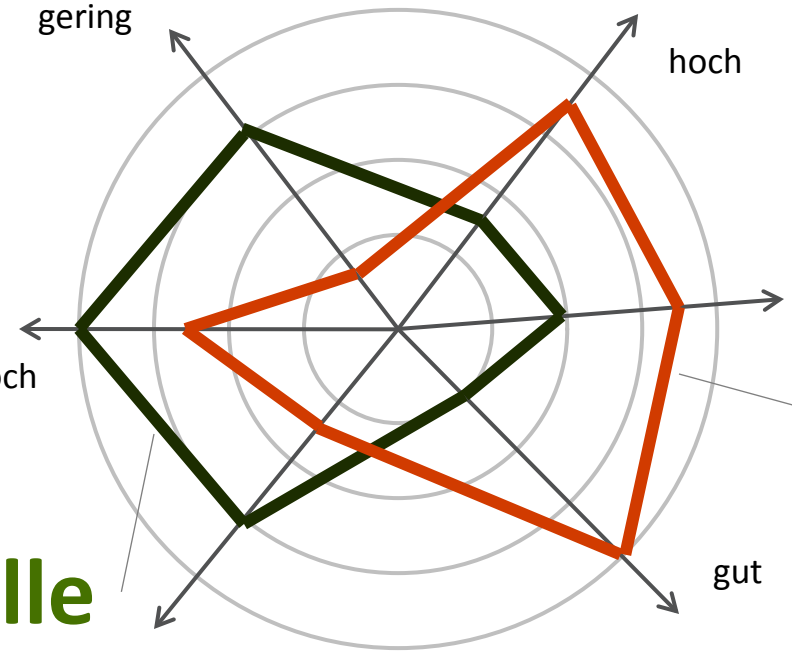

- Ist in **5 Jahren** noch das Modell **zugänglich**, welches dem Anwendungssystem in **Version 2.12.2-1** zugrunde liegt?




- Kann der Entwicklungsprozess später noch vollständig ausgeführt werden.

# Zielkonflikt modellgetriebener Softwareentwicklung

buschmais
Beratung . Technologie . Innovation




**XML-Modelle**

**UML-Modelle**

# Ein kleines Beispiel zum Einstieg

buschmais
Beratung . Technologie . Innovation

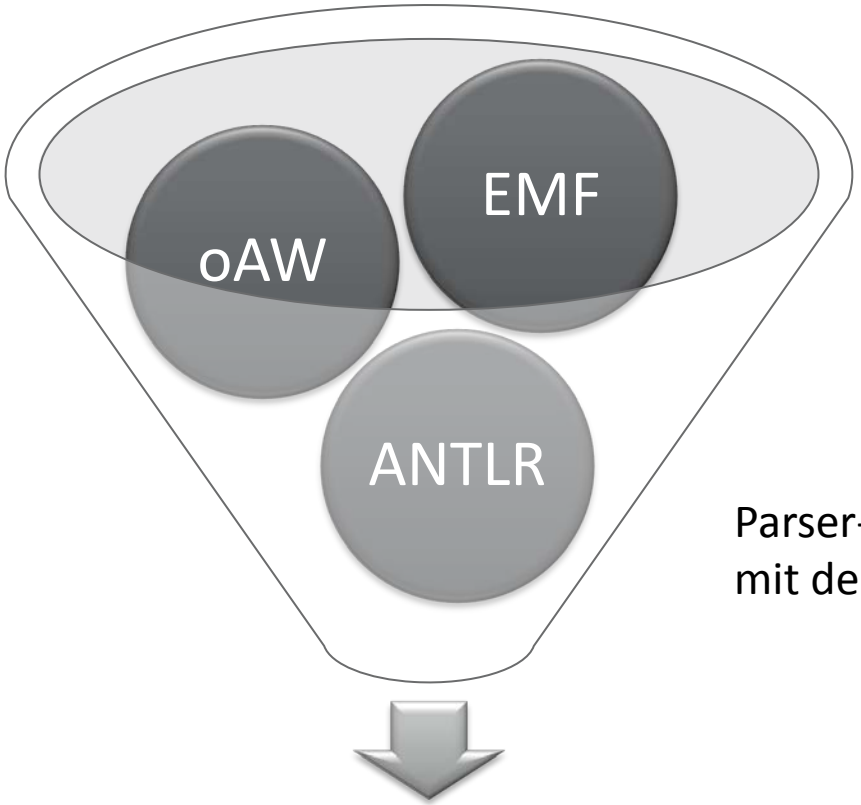
```
entitytype Person {
    attribute name : String;
    attribute birthday : Date;
    attribute salary : Currency;
}

entitytype Invoice {
    attribute invoiceNumber : String;
    attribute paymentTarget : Date;
    attribute total : Currency;
}
```

# Xtext

buschmais
Beratung . Technologie . Innovation

Ecore/EMF-Modelle

Workflows
Xpand
Xtend

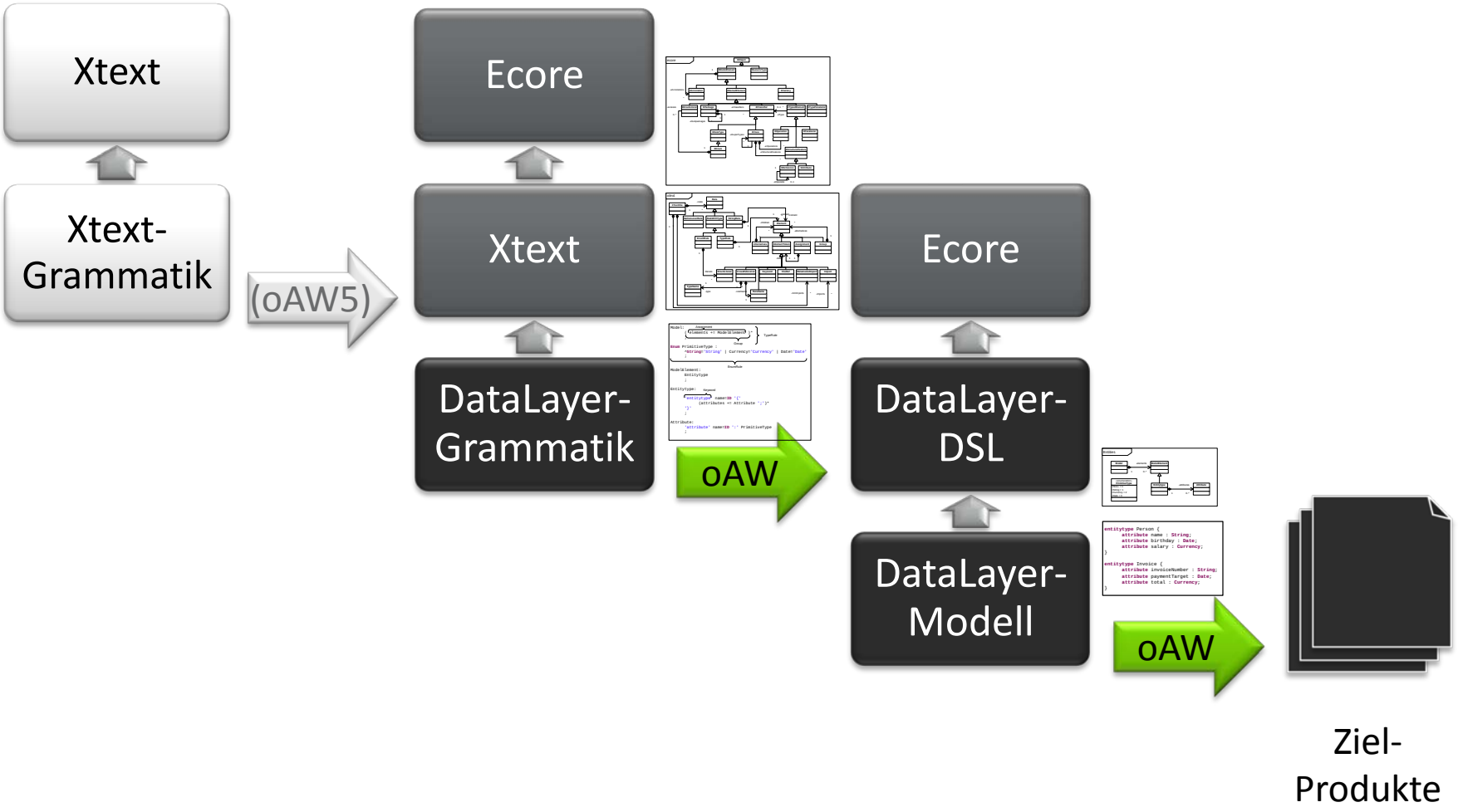



Parser-Generator
mit der Mächtigkeit LL(*)

domänenspezifische Sprache
auf Basis von Xtext

# Modellbeziehungen

buschmais
Beratung . Technologie . Innovation



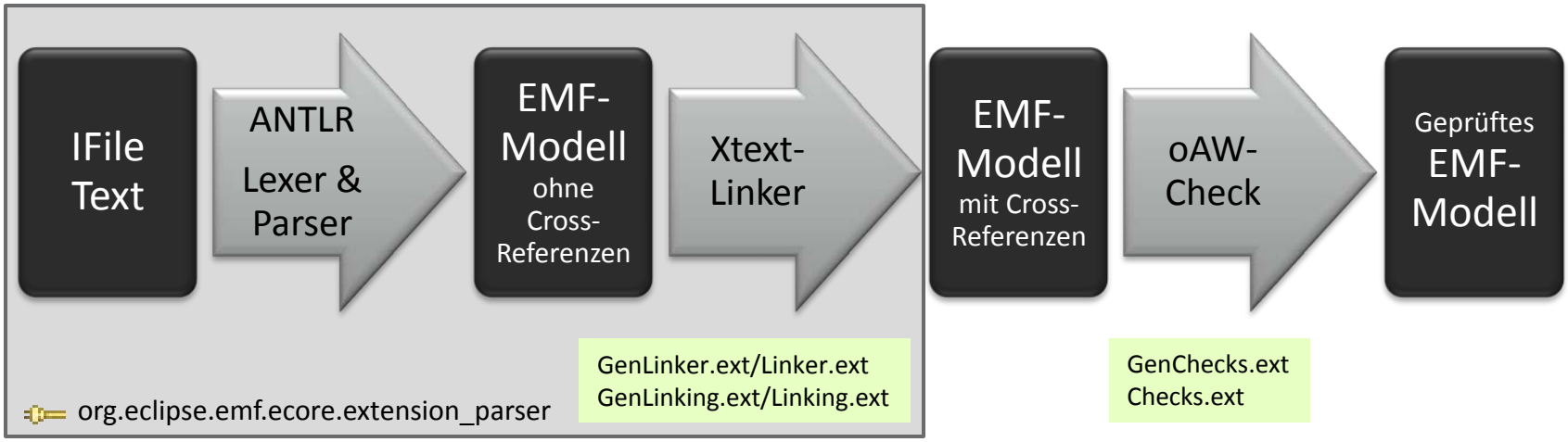
# Transformationen

buschmais
Beratung . Technologie . Innovation

1. DataLayer-Grammatik → Entities-DSL (Erzeugung der Werkzeuge)

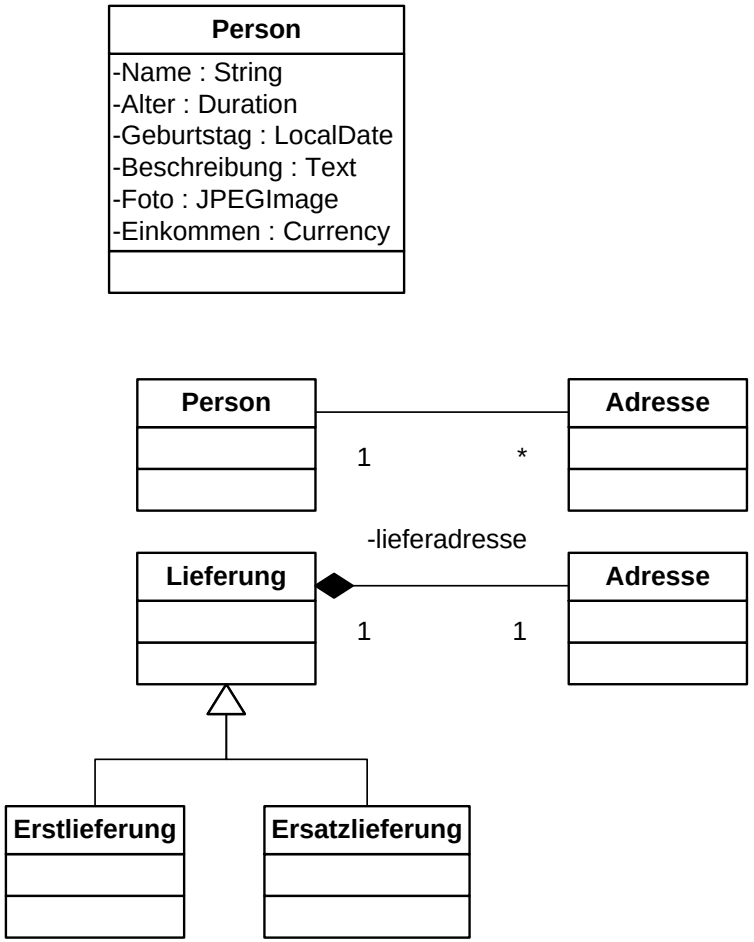
| EMF | ANTLR | Editor-Plug-in | oAW-Workflow (Parser.oaw) |
|---|---|---|---|
| • MM Ecore<br>• MM XMI<br>• Factory und Implementierung für Modell-Parser (und Dummy-Serialisierer) | • Grammatik<br>• Lexer<br>• Parser | • DataLayer-Editor (Linker, Checks, Style, Content-Assist, Outline, Navigation)<br>• New-Model-Wizard<br>• New-Project-Wizard | • Parser<br>• Xtext-Linker<br>• oAW-Checker |

2. DataLayer-Modell → Entities-DSL (Anwendung der Werkzeuge)



# Beschreibung der Domäne

buschmais
Beratung . Technologie . Innovation




- Konzepte
  - Entitätstypen
  - Attribute
  - Entitäts-“Ergänzungen“
  - Primitive Datentypen
  - Enumerationen
  - Referenzen: Existenz-Semantik, Änderungssemantik, Ordnung/Sortierung
  - Vererbung: Delegation, Mehrfachvererbung
- Datenbank-Mapping
  - Spalten-/Tabellen-Namen
  - Spaltengrößen
  - Indizes und Constraints
- O/R-Mapper-Konfiguration
  - Caching
  - Fetch-Strategien
  - Historisierung

# Datenbank-Mapping von Relationstypen

buschmais
Beratung . Technologie . Innovation

- Problemfall Java Persistence API 1.0
  - Join-Tabelle vs. Join-Spalte bei 1-1- oder 1-N-Beziehungen

| JPA1 | Unidirektionale 1-N-Beziehung | Bidirektionale 1-N-Beziehung |
|---|---|---|
| Join-Table |  |  |
| Join-Column |  |  |

  - Reihenfolgeerhalt bei Listen
  - Collections von primitiven Datentypen
  - ToOne-Beziehungen als Teil der Objektidentität
  - Geschachtelte Einbettung von Embeddable-Typen

# Sprach-Syntax

buschmais
Beratung . Technologie . Innovation

- Kriterien
  - Konsistenz der abstrakten Grammatik (Konzept-Konsistenz)
  - Parsebarkeit (Xtext-Freundlichkeit)
  - Kompaktheit vs. Lesbarkeit/Verständlichkeit
  - Zweckmäßigkeit („When in doubt, leave it out!“)

- Beispiel:

```
entitytype Person {
  superclass : Akteur;
  complement : Änderungsverfolgung;
  attribute name : String is-indexed;
  reference adresse : Adresse is-historized;
  reference projektHistorie : List<Projekt> # mitglieder is-ordered-by(endeDatum desc);
  reference kompetenzen : Set<Kompetenz> # many-to-many;
}
```

# Generierbare Artefakte

buschmais
Beratung . Technologie . Innovation

- Java-Beans
- Mapping-Deskriptoren (orm.xml vs. hbm.xml)
- Meta-Klassen
  - ähnlich dem Java-Reflection-API bzw. EMF-Modellen
  - Spalten-/Tabellen-Namen, Spalten-Präzision, direkte und geerbte Klassen-Attribute, direkte und indirekte Subklassen, …
  - typsichere Criteria-Queries

```
"SELECT p FROM Person p WHERE p.name = :name"

DomainObject fromPerson = queryDefinition.addRoot(Person.class);
fromPerson.where(fromPerson.get("name").equal(queryDefinition.parse("name")));

fromPerson = PersonEntity.newQueryRoot()
fromPerson.where(PersonEntity.name.equal(myQueryParameter));
```

- SQL-Schema
  - Create-Table-Skripte
  - Indizes / Constraints
  - Purge-Table-Skripte
  - Unterstützung der Schema-Migration

- Template-Sprache Xpand

```
«DEFINE main FOR Model»
  «EXPAND JavaBean FOREACH this.eContents.typeSelect(EntityType)»
«ENDDEFINE»

«DEFINE JavaBean FOR EntityType»
  «FILE this.name+'.java'»
    «EXPAND attribute FOREACH this.attributes()»
  «ENDFILE»
«ENDDEFINE»
```

- Modell-Prüfung mit Check

```
context EntityType
  ERROR "Name of Entity '" + this.name + "' is not unique" :
  this.siblings().select(e|e.name == this.name).size == 0
;
```

- Funktionale Erweiterungen über Xtend

```
List[EntityType] siblings(EntityType type):
     type.eContainer.eContents.typeSelect(EntityType).remove(this)
;
```

# Generierungsstrategien

buschmais
Beratung . Technologie . Innovation

- Redundanz in den Templates ist OK
- IF-Blöcke in Templates sind nur für „Ein-Zeiler“ empfehlenswert
- verschachtelte IF-Blöcke machen Template unwartbar
- besser sind selbstgeschriebene Funktionen als komplizierte Operationen in den Templates
  «`name.toFirstUpper()+'Type'`» → «`name.entityName()`»
- Es ist besser Model-zu-Model-Transformationen zur Auflösung von Konventionen nutzen (z.B. Namenskonventionen) als If-Not-Null-Abfragen in den Templates
- Neben der üblichen Typ-Selektion in den Templates kann eine semantische Selektionen IF-Abfragen reduzieren

```
«EXPAND oneToManyReference FOREACH this.oneToManyReferences()»
«EXPAND manyToManyReference FOREACH this.manyToManyReferences()»
```

buschmais

Beratung . Technologie . Innovation

# Fragen & Antworten

nächster Xtext Workshop: 19.03.2009 in Dresden