



J-Spring

16 april 2008 Spant! - Bussum



# Pragmatic MDA: Domain Specific Languages with Eclipse

Jos Warmer  
Partner





## Who, why



- **Expert in Model Driven Development**
- **One of the authors of the UML standard**
  - Main architect of OCL
- **Written some books**
  - Praktisch UML
  - MDA Explained
  - Object Constraint Language
- **Partner at Ordina**
  - Responsible for MDD
  - J-Technologies → SMART-Java
  - Ordina Microsoft Solutions → SMART-Microsoft

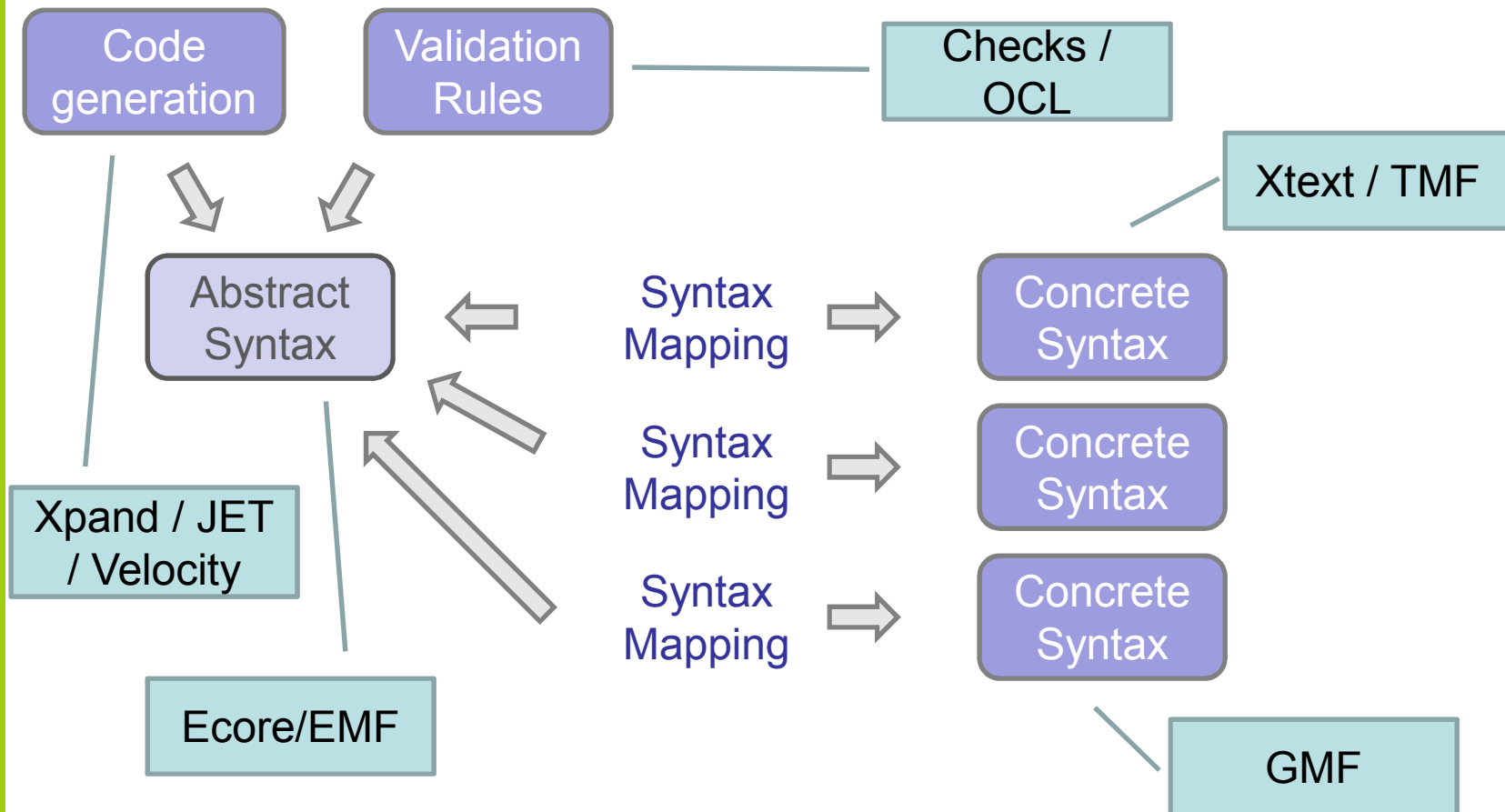


# Anatomy of Domain Specific Languages

- **The abstract syntax**
  - The domain model or meta model
- **Validation rules**
  - What constitutes a correct model
- **The concrete syntax or notation**
  - What does the language look like
- **Code generation**
  - Making the DSL executable
  - Alternative: interpretation



# DSL Definition





J-Spring

16 april 2008 Spant! - Bussum



# Abstract Syntax

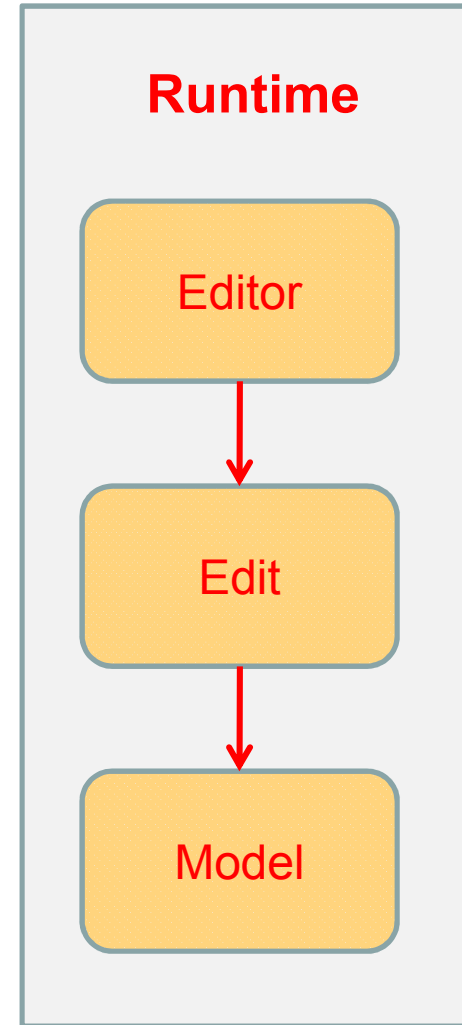
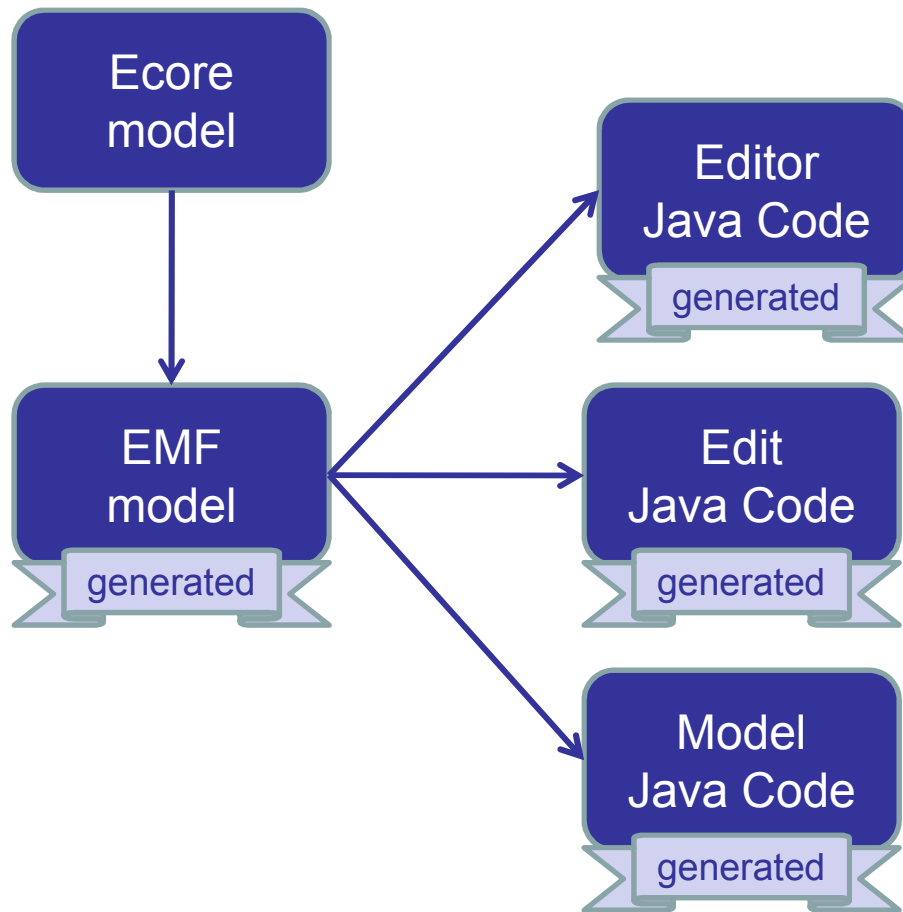


## Abstract Syntax

- **The abstract syntax describes the concepts of the DSL.**
  - These concepts come directly from the domain
- **This is the vocabulary used in the domain**
  - Often exists already with the domain experts
- **Usually modeled by a class diagram like notation**
  - E.g. UML Class models
- **Also called**
  - Meta model by the OMG
  - Domain model by Microsoft



# Eclipse: Ecore





J-Spring

16 april 2008 Spant! - Bussum



# Validation Rules

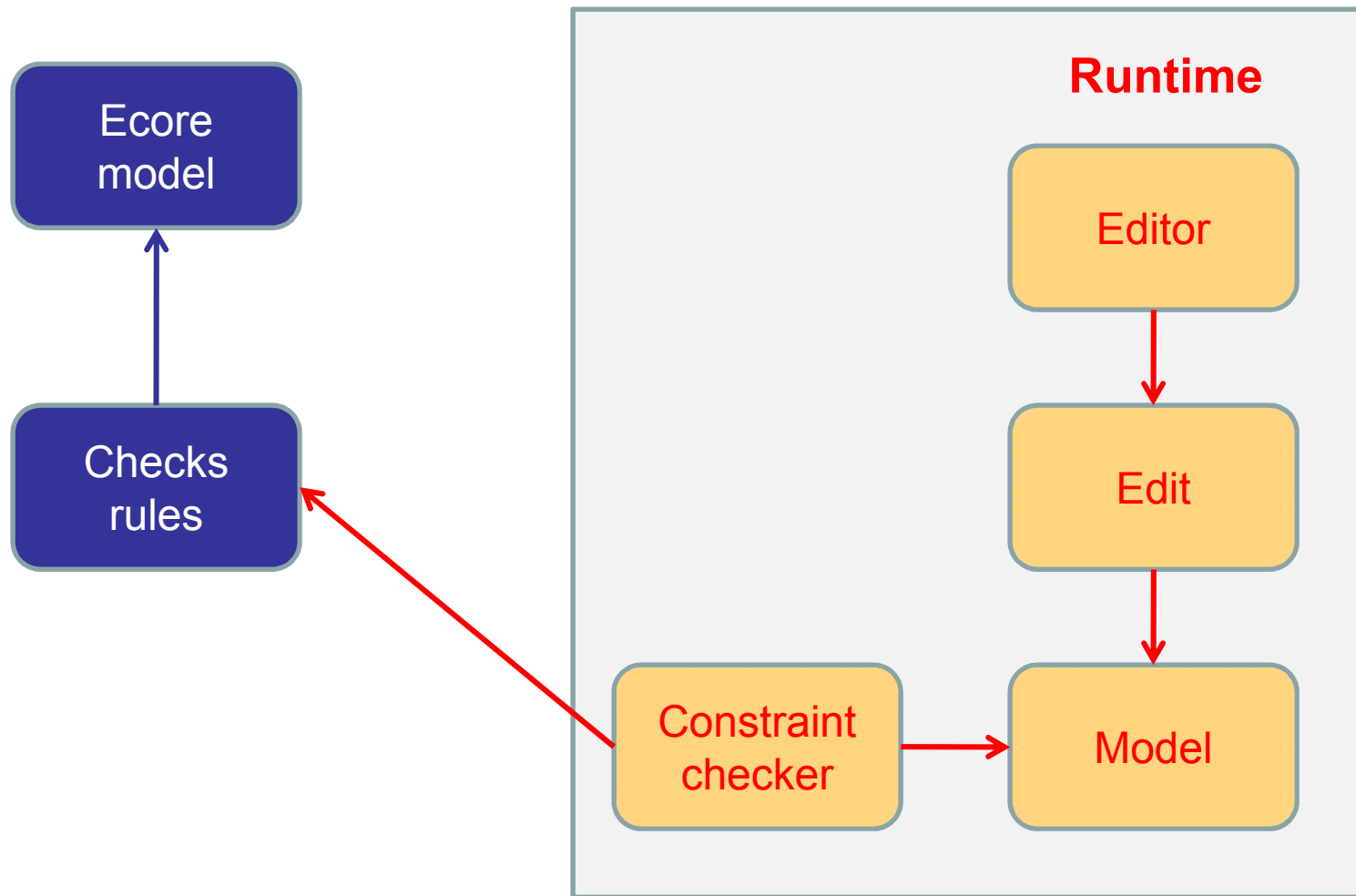


## Validation rules

- **The abstract syntax defines the structure**
- **Which instantiations of the abstract syntax are valid?**
- **Specified by validation rules (or well-formedness rules)**
- **E.g.**
  - **Class names must be unique**
  - **A class may not be a generalization of itself**



# Validation Rules in oAW / Checks





J-Spring

16 april 2008 Spant! - Bussum



# Code Generation

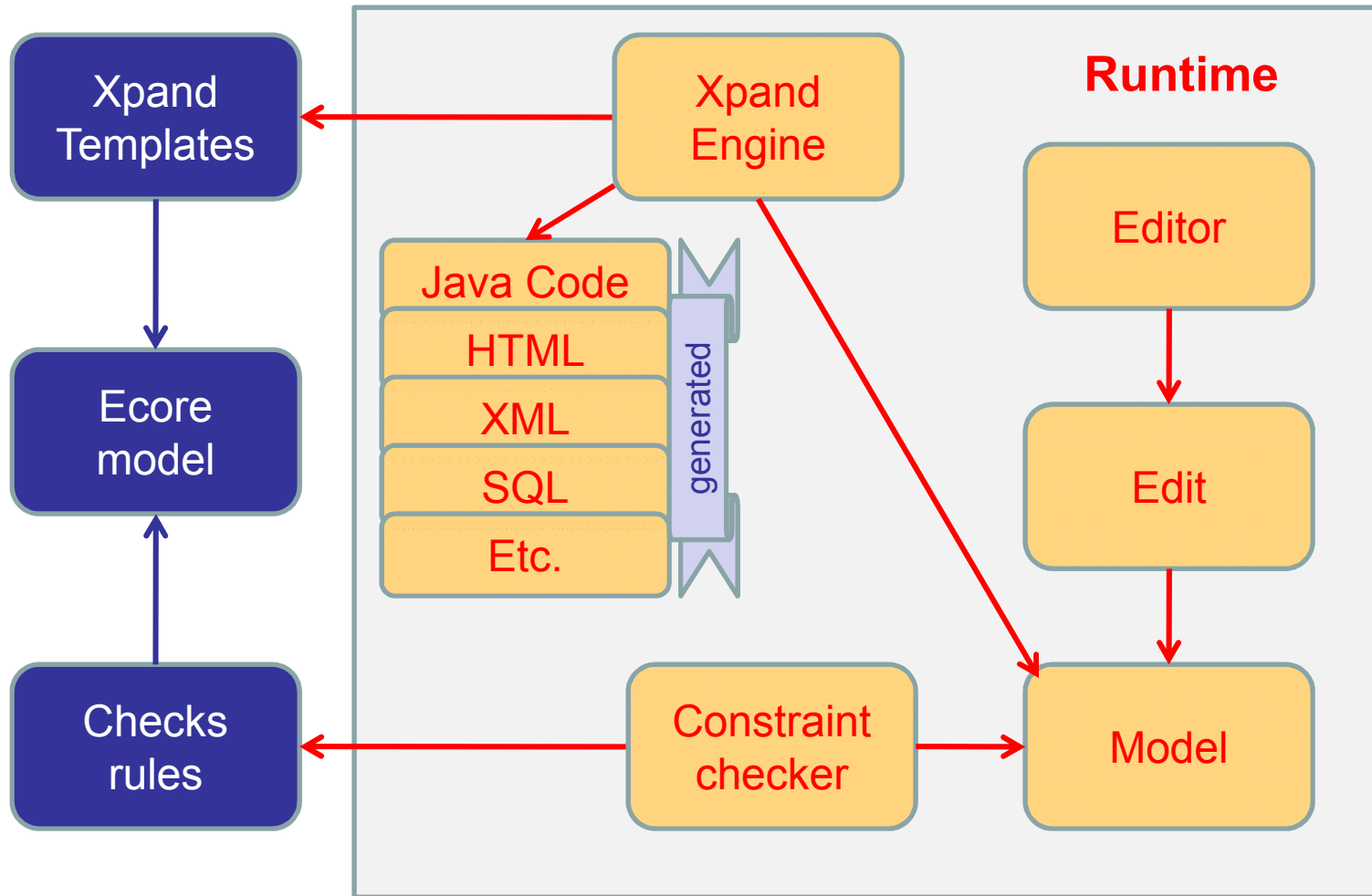


## Code Generation

- **Makes a DSL “executable”**
  - Can generate anything



# Code Generation with oAW Xpand





J-Spring

16 april 2008 Spant! - Bussum



# Concrete Syntax



## Concrete Syntax

- **Concrete Syntax** defines the “look” of the DSL
  - This is the only thing that users of the DSL actually see
- **Concrete syntax** is also called
  - Notation
  - Visualization
  - Surface syntax
- **A DSL may have more than one concrete syntax**
  - Different types of users may require different concrete syntaxes
  - Concrete syntax should be designed with the user type in mind



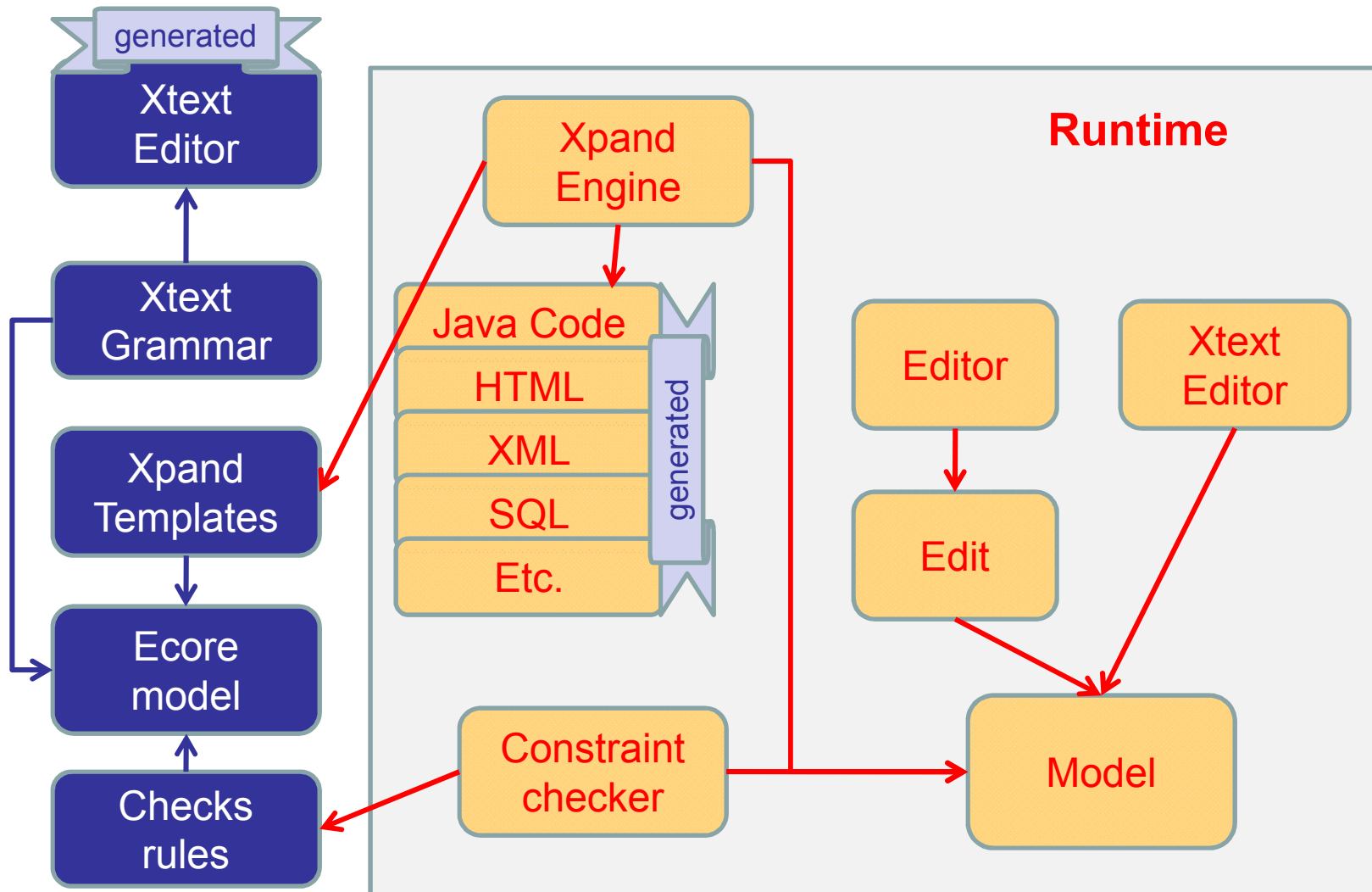
## Concrete syntax

- **Textual versus graphical**
  - ... or both !
- **Standalone or embedded**
  - This presentation is about standalone only.
- **Textual: oAW Xtext**
  - Alternative: TCS
  - Both will move (end merge) into Eclipse Textual Modeling Framework (TMF)
- **Visual: Graphical Modeling Framework (GMF)**
  - No real alternatives

xText: DSL's made easy  
*Jeroen Benckhuijsen,  
Meinte Boersma*



# Textual syntax: oAW Xtext





## Textual Concrete Syntax

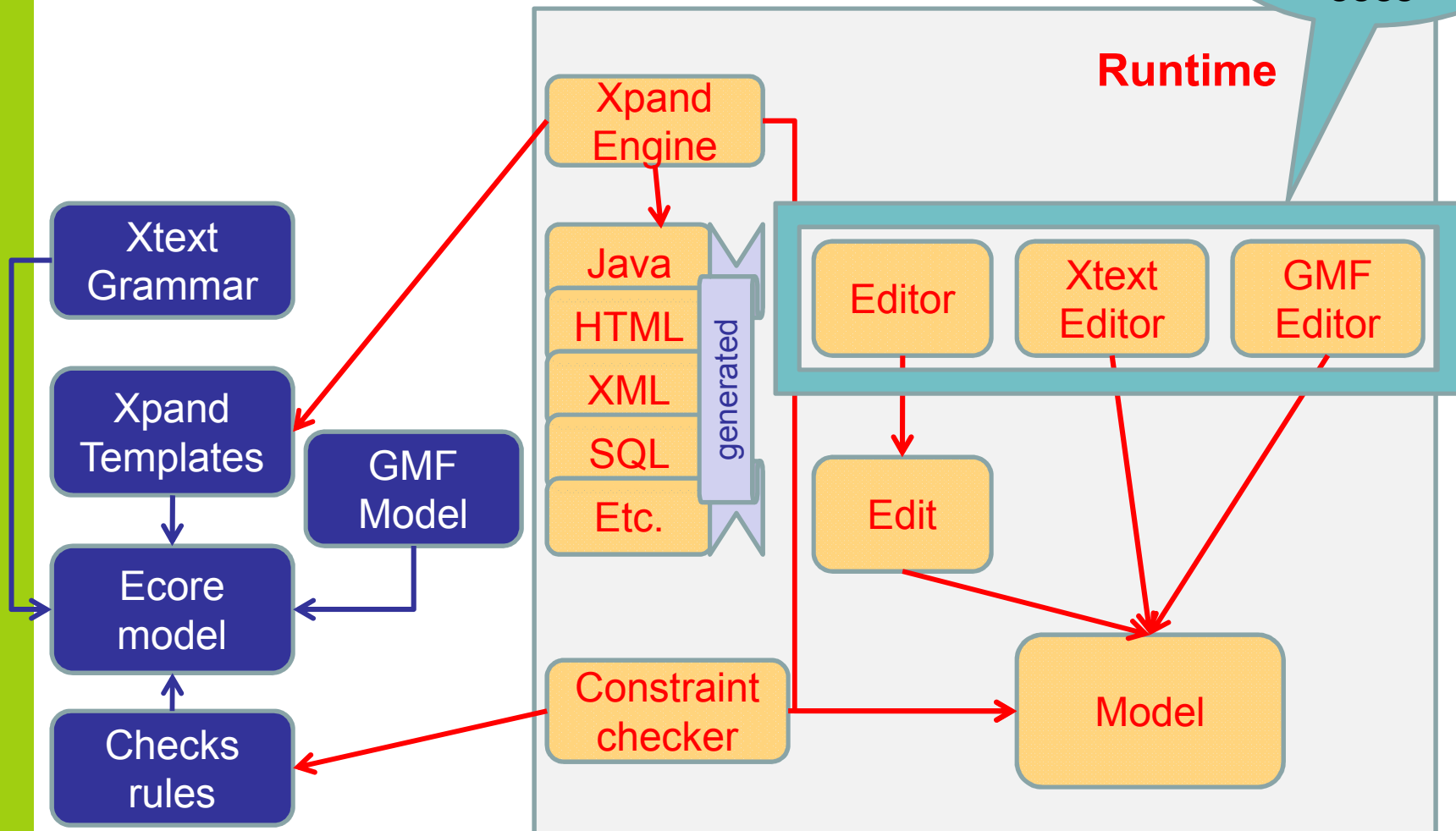
- **Traditional parser technology**
  - Uses parser generator tools
    - ANTRL
    - JavaCC
    - Lex & Yacc
    - ...
  - Grammar based (EBNF)

Building a Domain  
Specific Language  
with ANTLR v3 –  
Jeroen Leenarts



# Visual syntax: GMF

What the developer sees





J-Spring

16 april 2008 Spant! - Bussum



**Thanks and have fun !**

Jos Warmer

[jos.warmer@ordina.nl](mailto:jos.warmer@ordina.nl)