HOCHSCHULE PFORZHEIM UNIVERSITY

MERSES

Zentrum für Angewandte Forschung

Architekturen
2012

Domain Specific Languages for Modeling and Realization of Embedded Image Processing Architectures

Introduction – ZAFH MERSES

- Zentrum für Angewandte Forschung, Modellgestützte Entwurfs- und Realisierungsmuster für Signalverarbeitende Eingebettete Systeme
- Cooperative project in Baden-Württemberg
  - HS Pforzheim
  - HS Karlsruhe
  - HS Mannheim
  - HS Albstadt-Sigmaringen
  - Uni Tübingen,
  - Karlsruher Institut für Technologie (Uni Karlsruhe)
- Homepage www.merses.de

Introduction - Outline

1. Introduction
2. Embedded Systems
3. MDSD/MDA for Embedded Systems
4. Xtext and EMF
5. Image Processing SES
6. Example

Embedded Systems

- What is an Embedded System?
  - CPU, Coprocessor, Sensors/Actors
  - Limited Ressources
- Signal Processing Embedded Systems (SES)
  - DSPs, FPGAs, ASICs
  - Control Flow vs. Data Flow
  - Real-Time
- Design Flow for SES: Design Space Exploration

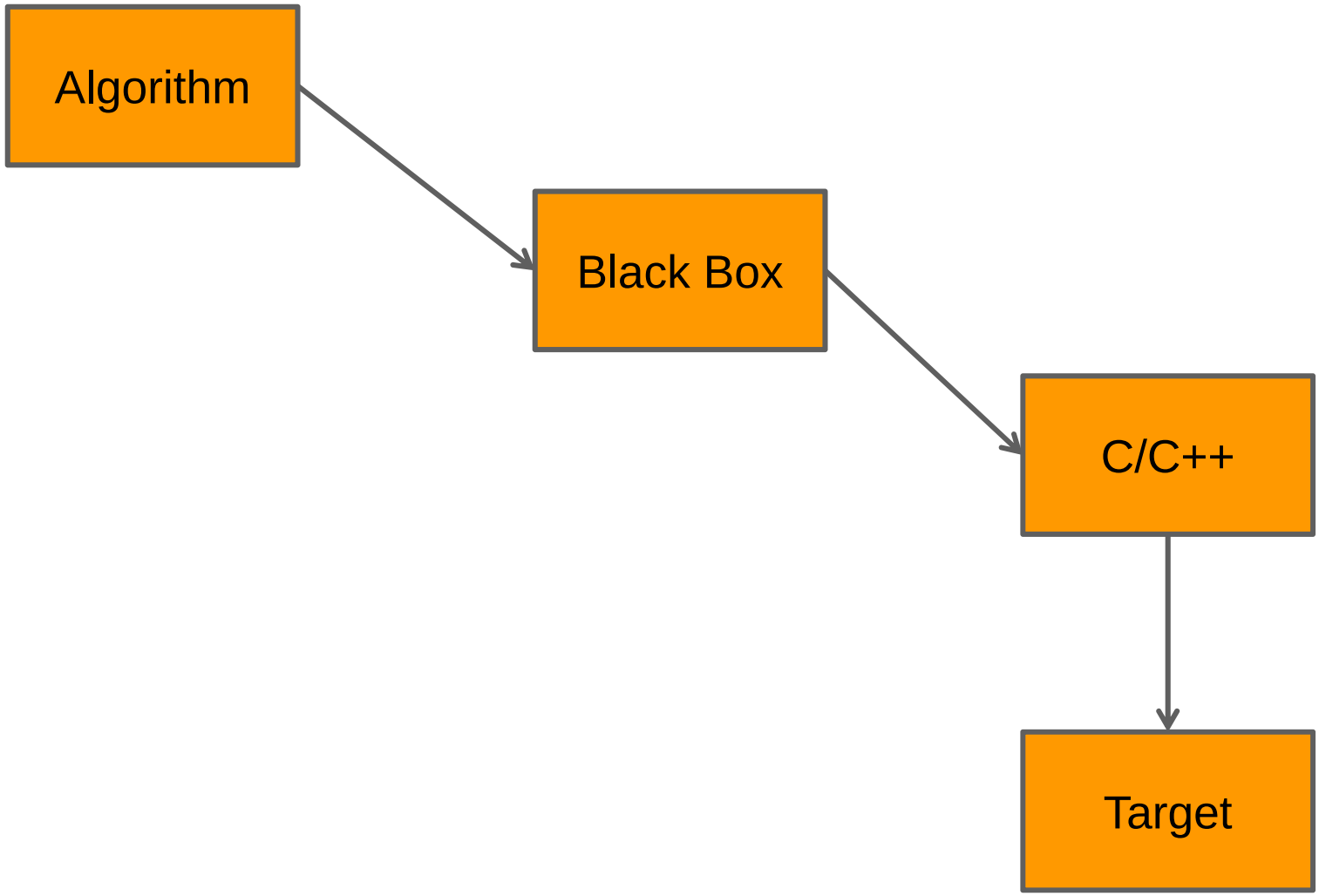
Embedded Systems: „Traditional“ Design Flow

- Algorithmic (Functional) Design
- Constraints
- Design Space Exploration
- (Manual) Code Development

Model Based Design of SES - Approaches

- The Matlab/Simulink way and IP/IDE based
- Domain Specific Approaches

The Matlab/Simulink methodology



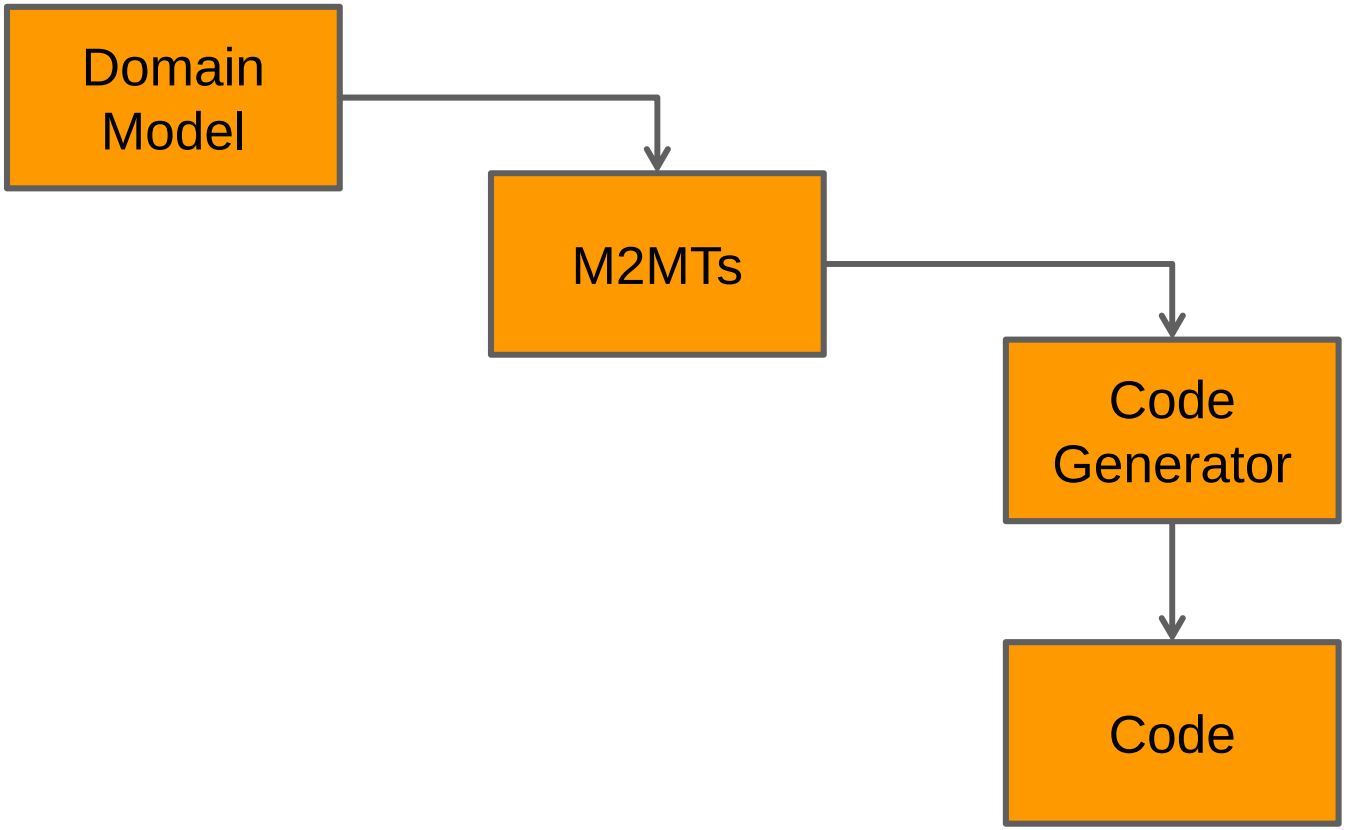
The Matlab/Simulink methodology

- Pros:
  - High Level Algorithmic Design
  - Powerful functions and libraries (“OTS”)
  - Industrial Standard

- Cons:
  - Expensive
  - Vendor Lock-In
  - IP Black Box

Domain Specific Languages

- Goal: Close gap between Domain Experts and „Programmers“



Domain Specific Languages

- Pros
  - Rich Tool Ecosystem
  - Can be cheap
  - High degree of freedom
- Cons
  - High degree of freedom
  - Not reached professional status (yet)
  - High effort at the beginning

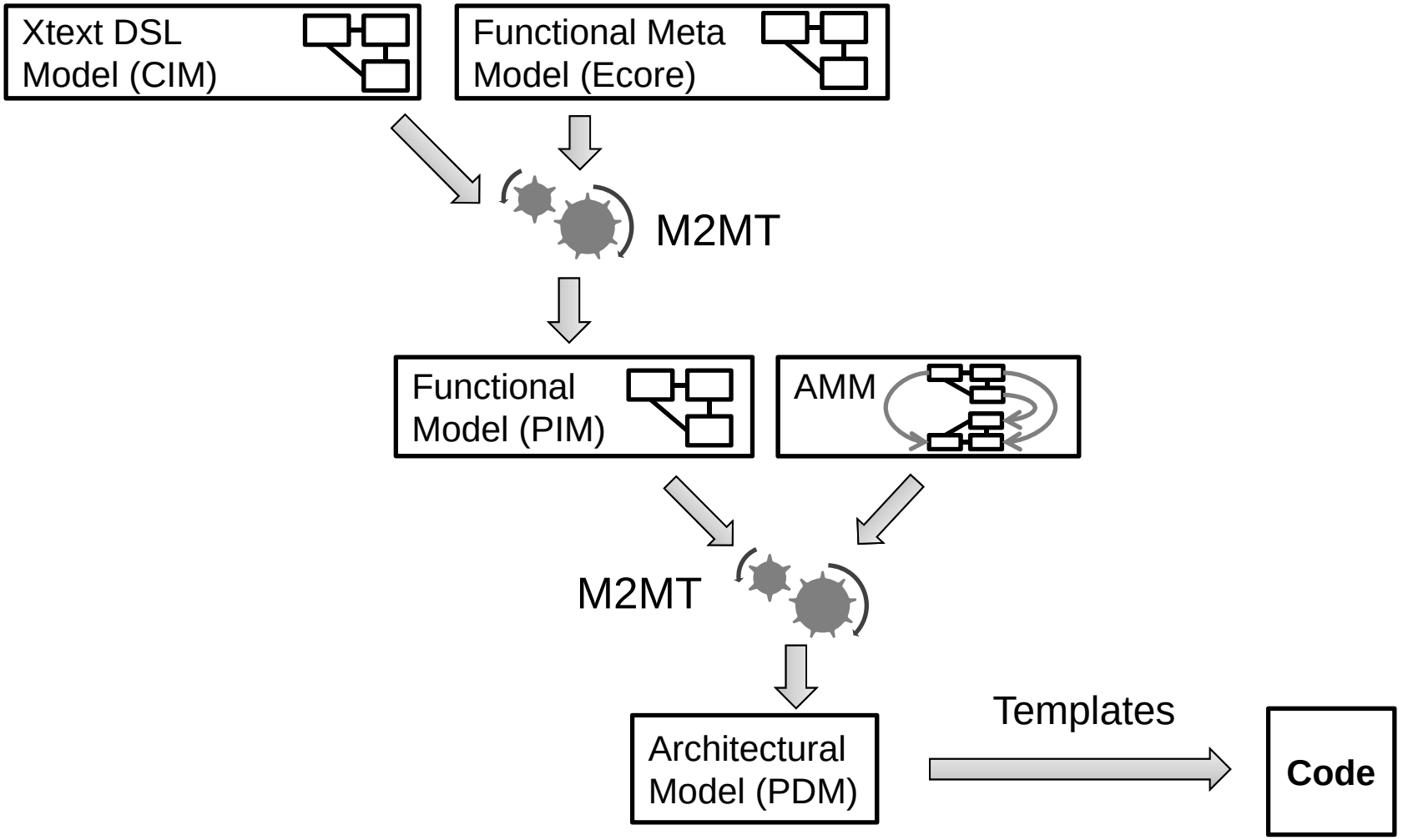
EMF/Xtext Framework

- Eclipse Modeling Framework EMF
- Xtext Framework
- Integrates with MWE, Xtend, Xpand

Xtext

- EBNF like Definition of LL(*) Grammars
- Parser generator
- AST class model
- Additional “Helpers”, e.g. Editor

MDA Design Flow with Xtext



SES Domain: Image Processing

- Low Level vs. High Level
- Volume vs. Complexity of data/operations
- Dedicated hardware architectures

→ Sophisticated AMMs

→ Efficient Code Generators

Image Processing Architectures

- High Data Rates
- Real-Time
- Complex Memory Patterns
- SIMD and MIMD

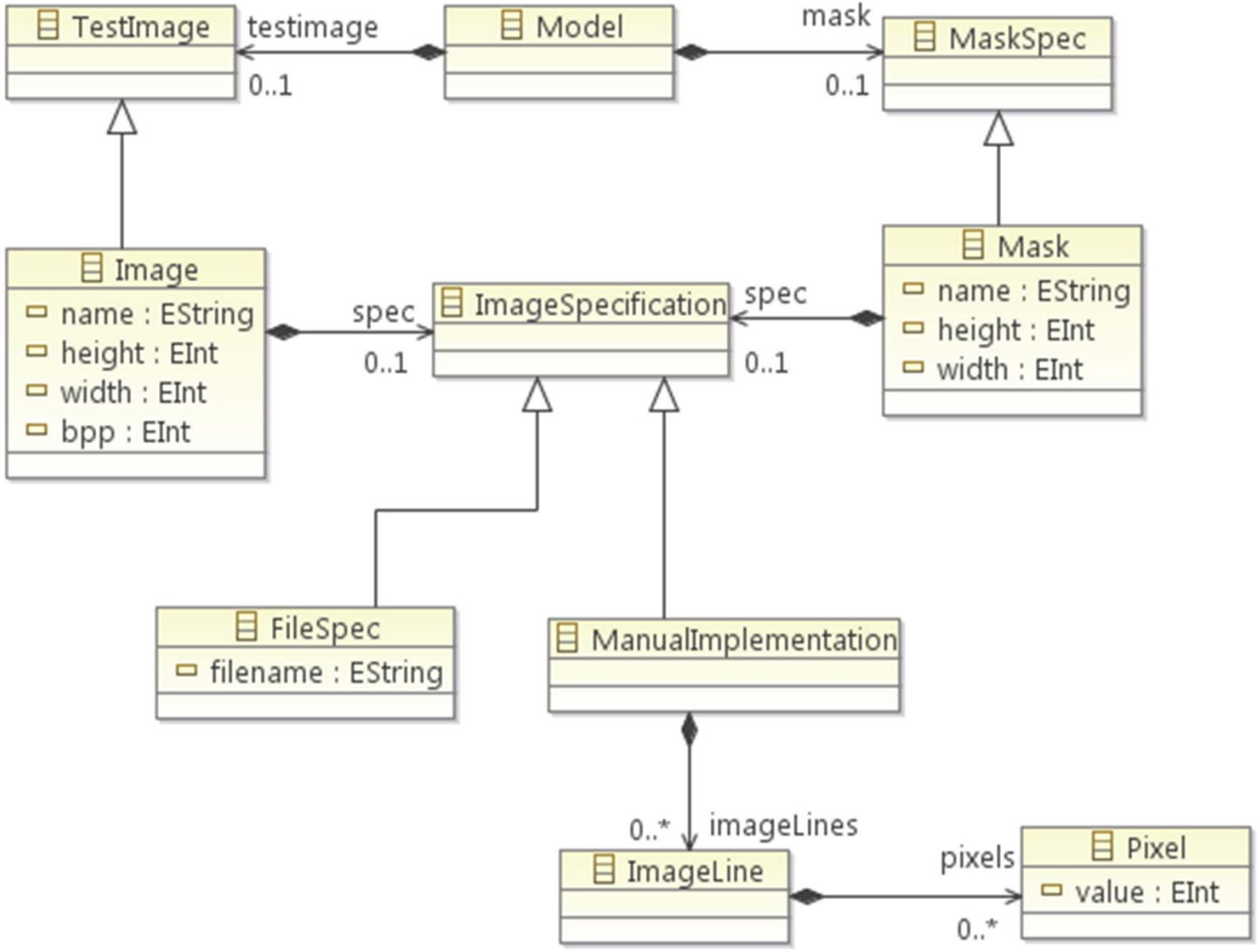
DSL for Image Processing Architectures

- Specification of Image structures (matrices)
- Processing: Blocks and Nodes
- Quantization (Floating Point -> Fixed Point -> Integer)
- Well-known functions and „IP“
- Test Definitions and Reference data

Example - DSL

```
1
2  grammar merses.my.first.hough.HoughDsl with org.eclipse.xtext.common.Terminals
3  import "http://www.eclipse.org/emf/2002/Ecore" as ecore
4
5  generate houghDsl "http://www.my.merses/first/hough/HoughDsl"
6
7  //The model has exactly one input image and one mask
8  Model:
9      testimage=TestImage
10     mask=MaskSpec
11 ;
12
13 TestImage: {Image} "image" name=ID "(" height=INT "," width=INT "," bpp=INT ")" spec=ImageSpecification;
14
15 ImageSpecification: ManualImplementation | FileSpec;
16
17 FileSpec: "file=(" filename=FILESTRING ");";
18
19 ManualImplementation: "("imageLines+=ImageLine ("," imageLines+=ImageLine)*");";
20
21 ImageLine: "("pixels+=Pixel ("," pixels+=Pixel)* ")";
22
23 Pixel: value = INT;
24
25 MaskSpec: {Mask} "mask" name=ID "(" height=INT "," width=INT")" spec=ImageSpecification;
26
27 terminal FILESTRING returns ecore::EString: ('a'..'z'|'A'..'Z'|':'|"\\"|'.'|'%'|('0'..'9')+ID|'_')*;
28 terminal INT returns ecore::EInt: ('0'..'9')+;
```

Example – DSL as Ecore



MERSES

Example – Input Model

```
1
2  //image i1 (768,1024,1) file=(d:\temp\testbilder\768x1024_kreis_mit_kreuz_10_stellen_10%Rauschen.png);
3  image image1 (512,512,1) file=(d:\\temp\\Testbilder\\512x512_40noise_32x32ref10.png);
4
5  mask m1 (32,32) file=(d:\temp\testbilder\32x32kreis_mit_kreuz.png);
6
7  /*mask m1 (5,5) (
8      (1,0,0,0,1),
9      (0,1,0,1,0),
10     (0,0,1,0,0),
11     (0,1,0,1,0),
12     (1,0,0,0,1)
13 );*/
```

Example – Generated Code and Files

- VHDL code files
- Test- and reference data
- Additional Java helpers

Conclusion

- DSL Approach for the Design of SES
- Design Space Exploration
- Code Generation
- Integration with EDA Tools

Q & A

Thank you for your attention

Merses on the web **www.merses.de**