

# Domain-Specific Languages and Xtext

Jean-Marc Jézéquel / Mathieu Acher

[jezequel@irisa.fr](mailto:jezequel@irisa.fr), [mathieu.acher@irisa.fr](mailto:mathieu.acher@irisa.fr)

# Plan

- Domain-Specific Languages (DSLs)
  - Languages and abstraction gap
  - Examples and rationale
  - DSLs vs General purpose languages, taxonomy
- External DSLs
  - Grammar and parsing
  - Xtext
- DSLs, DSMLs, and (meta-)modeling

# BIBTEX

```
[Event "F/S Return Match"]
[Site "Belgrade, Serbia Yugoslavia|JUG"]
[Date "1992.11.04"]
[Round "29"]
[White "Fischer, Robert J."]
[Black "Spassky, Boris V."]
[Result "1/2-1/2"]

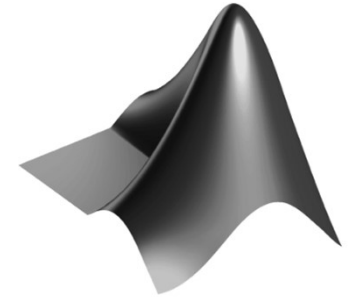
1. e4 e5 2. Nf3 Nf6 3. Bb5 Bb5 4. Bb5 Nf6 5. O-O Be7 6. Bb5 Bb5 7. Bb5 Nf6 8. Bb5 Nf6 9. Bb5 Nf6 10. Bb5 Nf6 11. c4 c6 12. c5 Nc3 13. Nc3 Nc3 14. Nc3 Nc3 15. Nc3 Nc3 16. Nc3 Nc3 17. Nc3 Nc3 18. Bxe7 Qx7 19. exd6 Qf6 20. Nf3 Nf6 21. Nf3 Nf6 22. Nf3 Nf6 23. Ne5 Rae8 24. Rxf7+ Rxf7 25. Rxf7 Rxf7 26. Rxf7 Rxf7 27. Rxf7 Rxf7 28. Rxf7 Rxf7 29. h4 h5 29. b3 Ke6 30. a3 Kd6 31. axb4 cxb4 32. Ra5 Nd7 33. Ra5 Nd7 34. Ra5 Nd7 35. Ra7 g6 36. Ra6+ Kc5 37. Kd1 Nf4 38. g3 Nxf3 39. Nf2 42. g4 Bd3 43. Re6 1/2-1/2
```



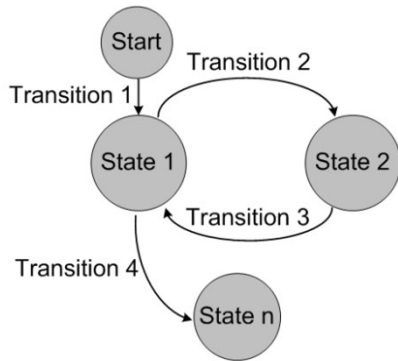
## Graphviz



## Make



## Matlab



## Finite State Machine

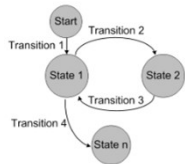


# Domain-Specific Languages (DSLs)

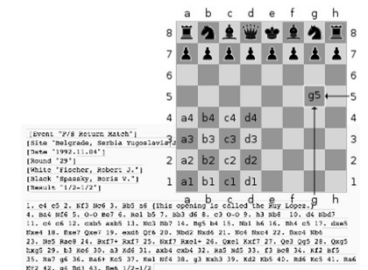
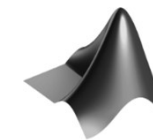
# DSL = Syntax + Services

## Specialized notation:

Textual or Graphical  
Specific Vocabulary  
Idiomatic constructs



BIBTEX



# xtext

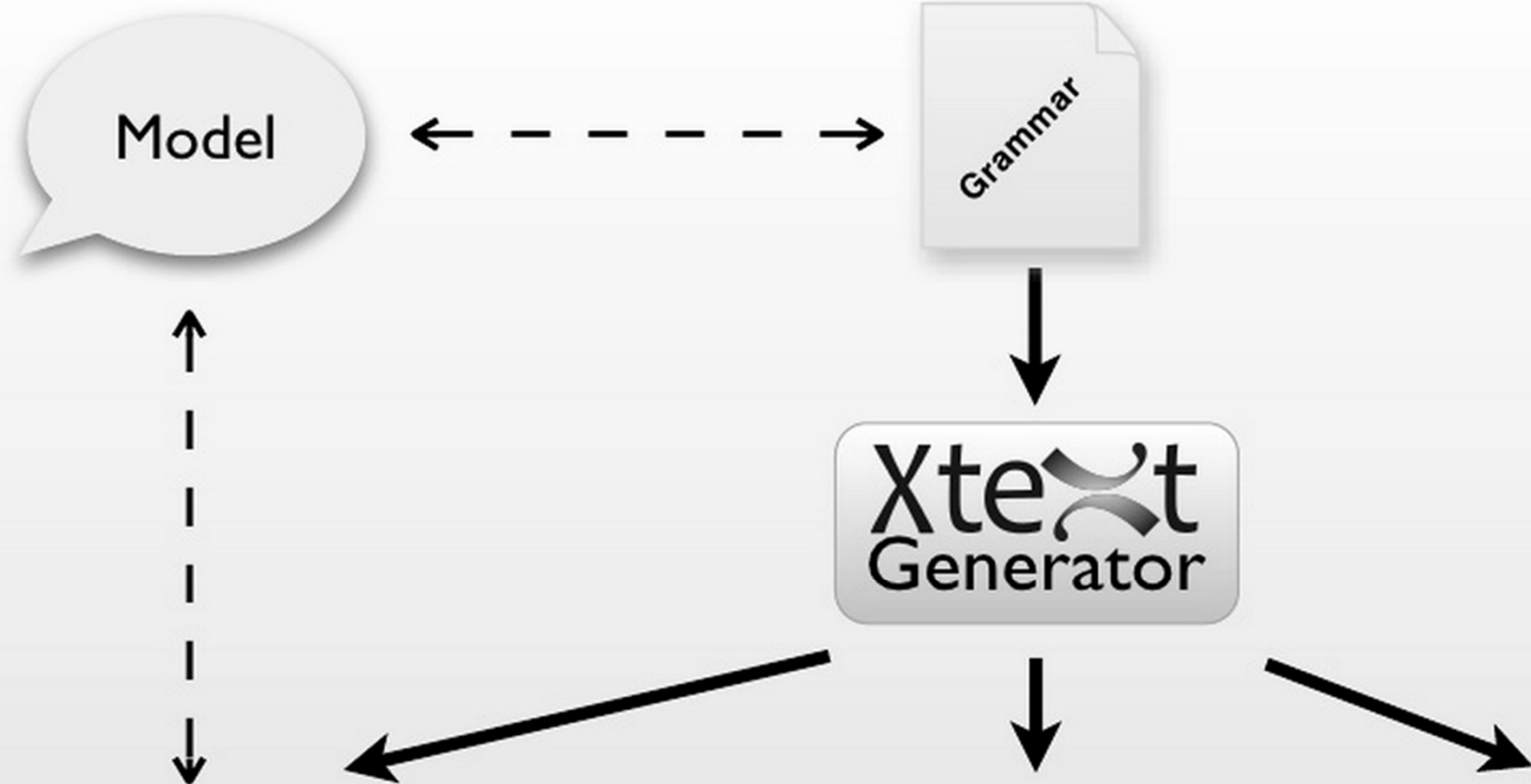
Give me a **grammar**,

I'll give you (for free)

- \* a comprehensive editor (auto-completion, syntax highlighting, etc.) in Eclipse

- \* an Ecore metamodel and facilities to load/serialize/visit conformant models (Java ecosystem)

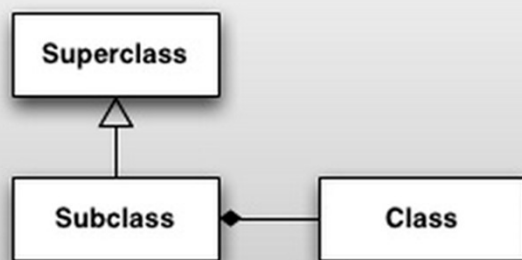
- \* extension to override/extend « default » facilities (e.g., checker)



# Xtext Runtime



LL(\*) Parser

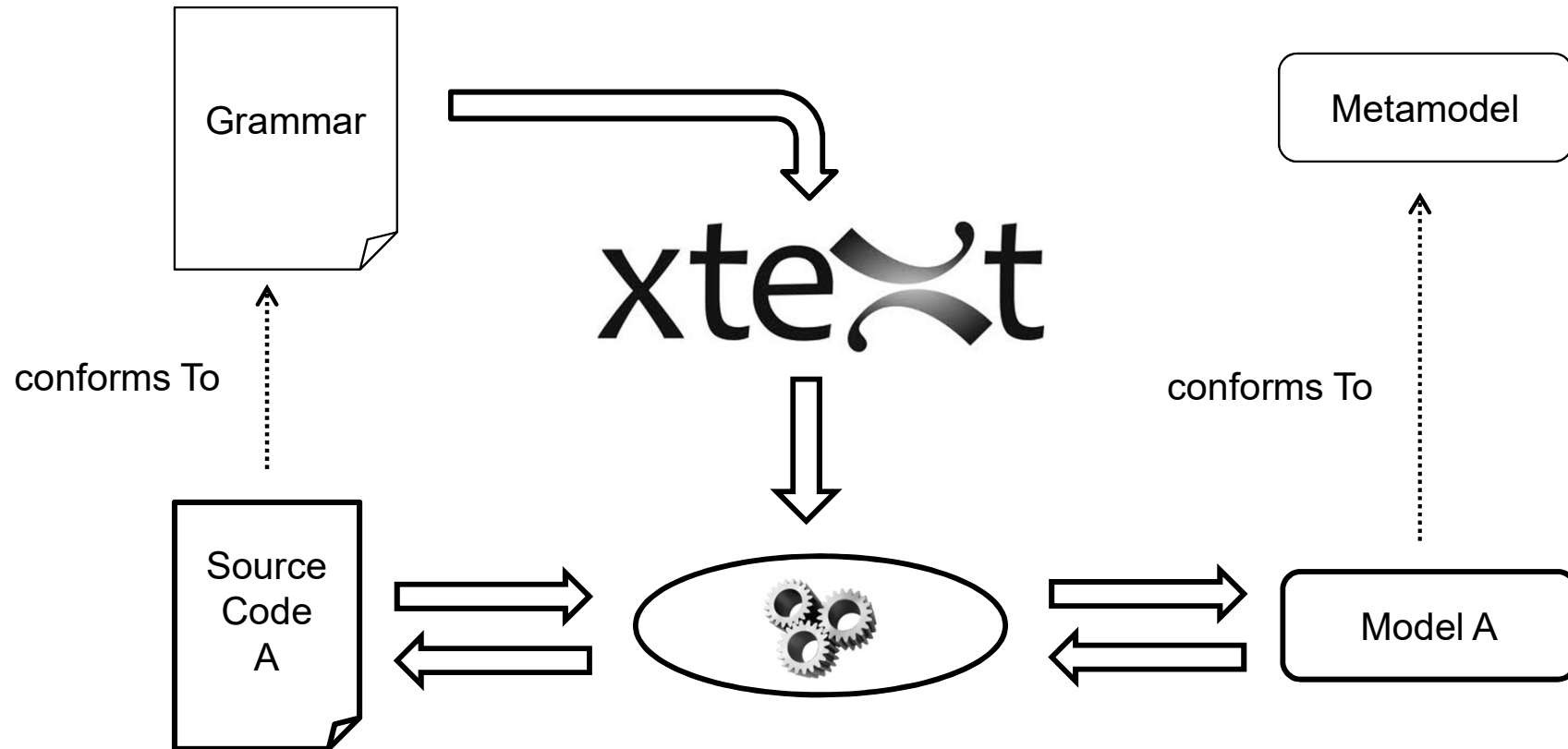


ecore meta model



editor

# Xtext, Grammar, Metamodel



# Xtext Project

- Eclipse Project
  - Part of Eclipse Modeling
  - Part of Open Architecture Ware
- Model-driven development of Textual DSLs
- Part of a family of languages
  - **Xtext**
  - Xtend
  - Xbase
  - Xpand
  - Xcore

The logo for the Xtext project, featuring the word "xtext" in a lowercase, sans-serif font. The letter "x" is stylized with a horizontal bar that curves upwards and then downwards, resembling a stylized "x" or a pair of wings.



# Eclipse Modeling Project



# The Grammar Language of Xtext

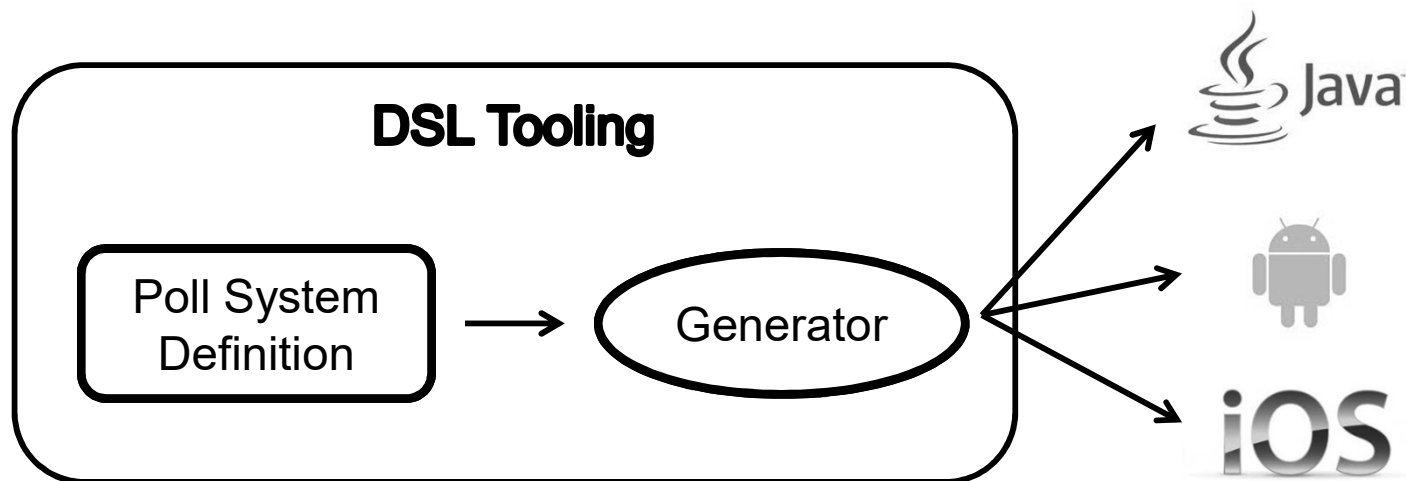
- Corner-stone of Xtext
- A... DSL to define textual languages
  - Describe the concrete syntax
  - Specify the mapping between concrete syntax and domain model
- From the grammar, it is generated:
  - The domain model
  - The parser
  - The tooling

# Main Advantages

- Consistent look and feel
- Textual DSLs are a resource in Eclipse
- Open editors can be extended
- Complete framework to develop DSLs
- Easy to connect to any Java-based language

# Example DSL

- Poll System application
  - Define a Poll with the corresponding questions
  - Each question has a text and a set of options
  - Each option has a text
- Generate the application in different platforms



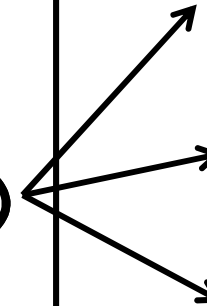
# Something like...

## DSL Tooling

```
PollSystem {  
  Poll Quality {  
    Question q1 {  
      "Value the user experience"  
      options {  
        A : "Bad"  
        B : "Fair"  
        C : "Good"  
      }  
    }  
    Question q2 {  
      "Value the layout"  
      options {  
        A : "It was not easy to locate elements"  
        B : "I didn't realize"  
        C : "It was easy to locate elements"  
      }  
    }  
  }  
  Poll Performance {  
    Question q1 {  
      "Value the time response"  
      options {  
        A : "Bad"  
        B : "Fair"  
        C : "Good"  
      }  
    }  
  }  
}
```



Generator



# Xtext Grammar

Grammar  
definition

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

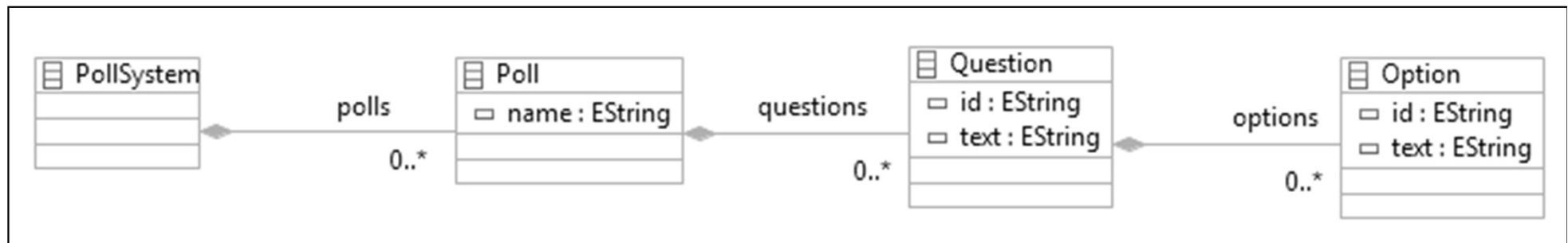
generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
    'PollSystem' '{' polls+=Poll+ '}' ;

Poll:
    'Poll' name=ID '{' questions+=Question+'}';

Question:
    'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
    id=ID ':' text=STRING;
```



# Xtext Grammar

Grammar  
reuse

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

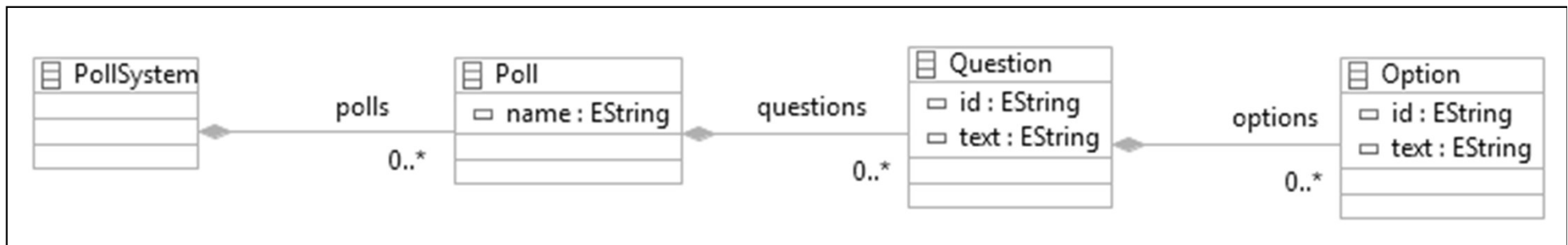
generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
    'PollSystem' '{' polls+=Poll+ '}' ;

Poll:
    'Poll' name=ID '{' questions+=Question+'}';

Question:
    'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
    id=ID ':' text=STRING;
```



# Xtext Grammar

Derived  
metamodel

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

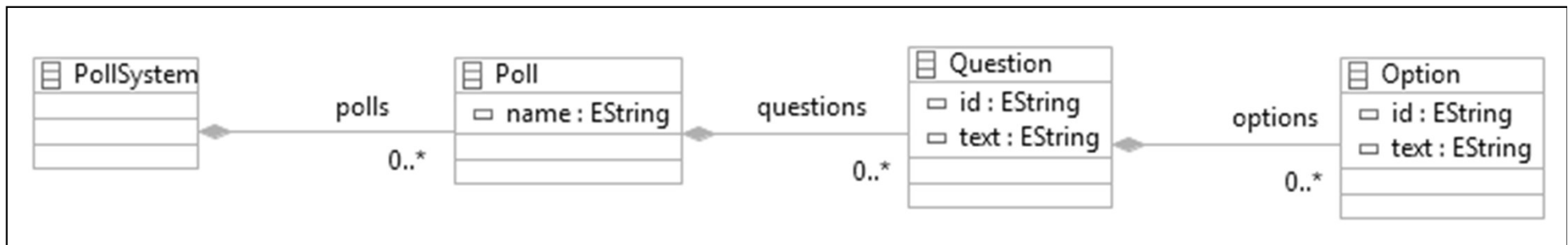
generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
    'PollSystem' '{' polls+=Poll+ '}' ;

Poll:
    'Poll' name=ID '{' questions+=Question+'}';

Question:
    'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
    id=ID ':' text=STRING;
```





# Xtext Grammar

Parser Rules

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

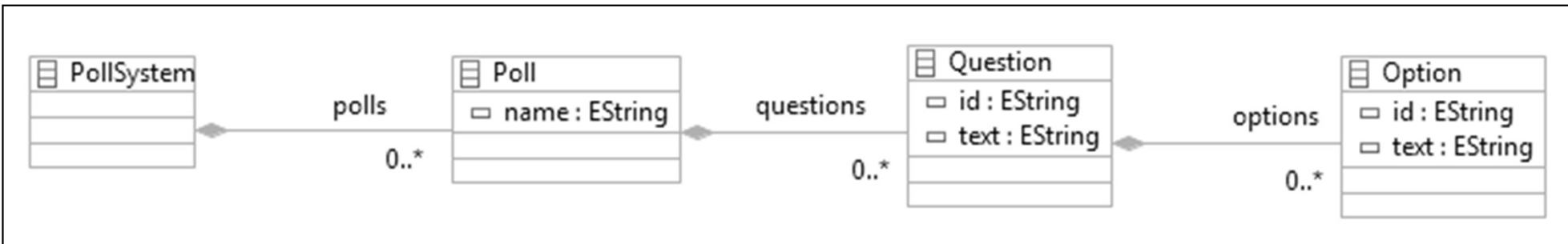
generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
    'PollSystem' '{' polls+=Poll+ '}' ;

Poll:
    'Poll' name=ID '{' questions+=Question+'}';

Question:
    'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
    id=ID ':' text=STRING;
```



# Xtext Grammar

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
```

```
generate poll "http://www.miage.fr/xtext/Poll"
```

```
PollSystem:
```

```
→ 'PollSystem' '{' polls+=Poll+ '}' ;
```

```
Poll:
```

```
→ 'Poll' name=ID '{' questions+=Question+'}';
```

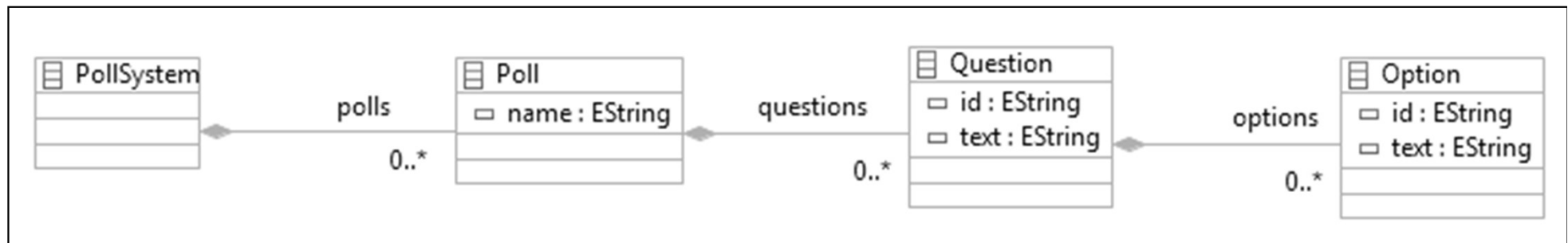
```
Question:
```

```
→ 'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;
```

```
Option:
```

```
id=ID ':' text=STRING;
```

Keywords



# Xtext Grammar

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
```

```
generate poll "http://www.miage.fr/xtext/Poll"
```

```
PollSystem:  
    'PollSystem' '{' polls+=Poll+ '}' ;
```

← Multivalue assignment

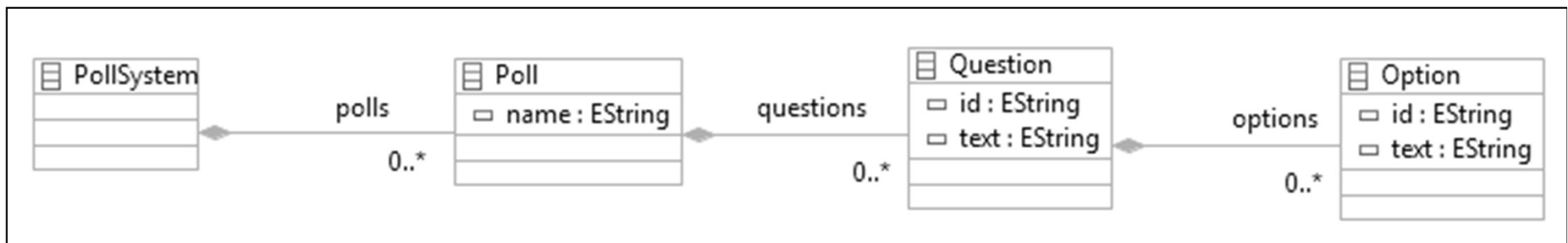
```
Poll:  
    'Poll' name=ID '{' questions+=Question+'}';
```

```
Question:  
    'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;
```

```
Option:  
    id=ID ':' text=STRING;
```

← Simple assignment

(not here → ?= Boolean assignment)



# Xtext Grammar

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
```

```
generate poll "http://www.miage.fr/xtext/Poll"
```

```
PollSystem:
```

```
  'PollSystem' '{' polls+=Poll+ '}' ;
```

← Cardinality (others: \* ?)

```
Poll:
```

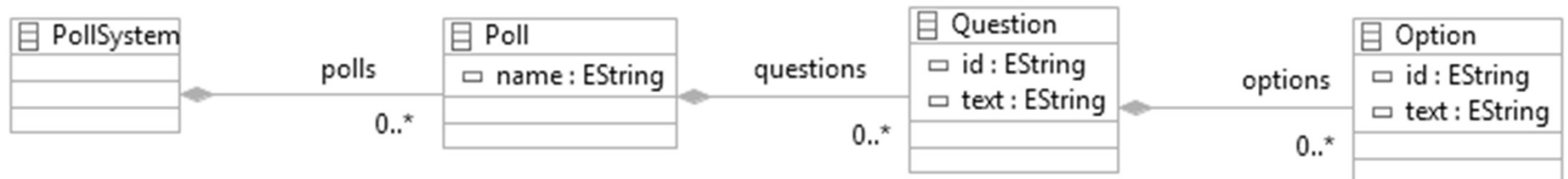
```
  'Poll' name=ID '{' questions+=Question+'}' ;
```

```
Question:
```

```
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;
```

```
Option:
```

```
  id=ID ':' text=STRING;
```



# Xtext Grammar

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
```

```
generate poll "http://www.miage.fr/xtext/Poll"
```

```
PollSystem:
```

```
  'PollSystem' '{' polls+=Poll+ '}' ;
```

```
Poll:
```

```
  'Poll' name=ID '{' questions+=Question+'}';
```

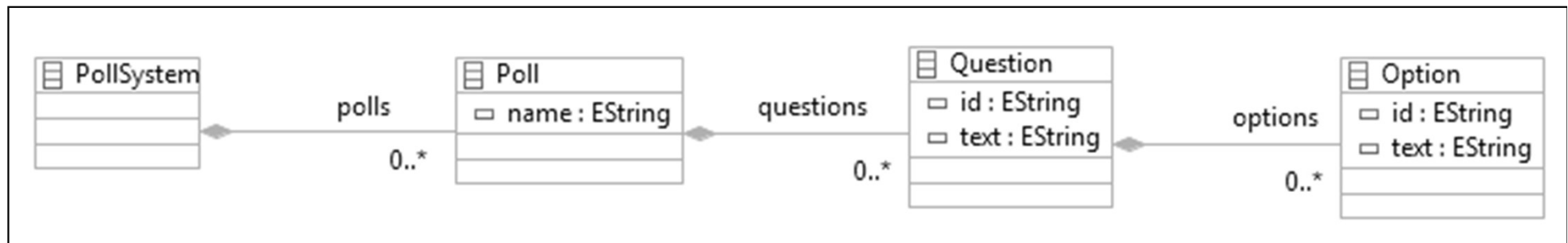
```
Question:
```

```
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;
```

```
Option:
```

```
  id=ID ':' text=STRING;
```

Containment



# Grammar and Programs/Specifications/Models

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
generate poll "http://www.miage.fr/xtext/Poll"

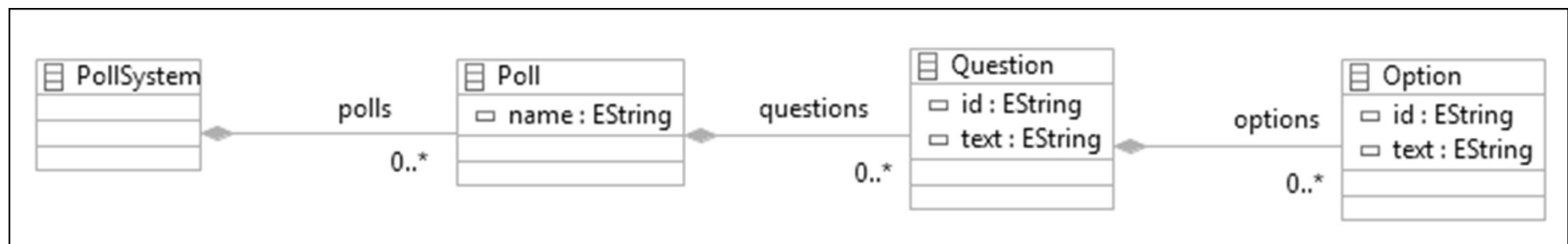
PollSystem:
  'PollSystem' '{' polls+=Poll+ '}' ;

Poll:
  'Poll' name=ID '{' questions+=Question+'}';

Question:
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
  id=ID ':' text=STRING;
```

```
PollSystem {
  Poll Quality {
    Question q1 {
      "Value the user experience"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
    Question q2 {
      "Value the layout"
      options {
        A : "It was not easy to locate elements"
        B : "I didn't realize"
        C : "It was easy to locate elements"
      }
    }
  }
  Poll Performance {
    Question q1 {
      "Value the time response"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
  }
}
```



# Grammar and Programs/Specifications/Models

```
grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals
generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
  'PollSystem' '{' polls+=Poll+ '}' ;

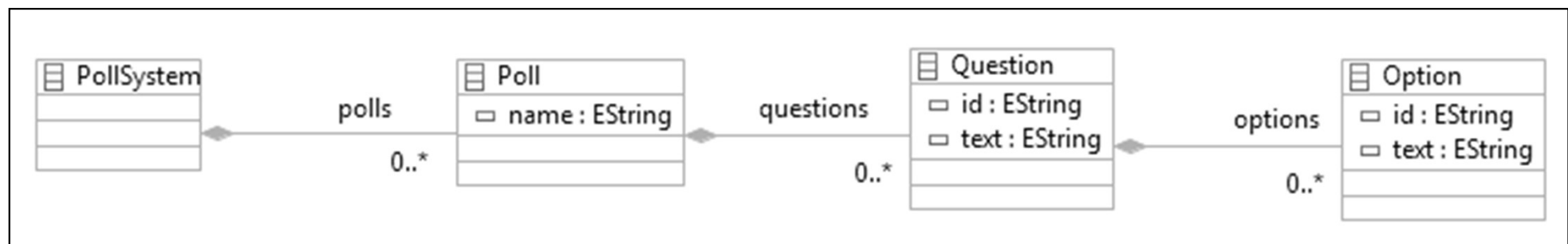
Poll:
  'Poll' name=ID '{' questions+=Question+'}';

Question:
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
  id=ID ':' text=STRING;
```

```
PollSystem {
  Poll Quality {
    Question q1 {
      "Value the user experience"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
    Question q2 {
      "Value the layout"
      options {
        A : "It was not easy to locate elements"
        B : "I didn't realize"
        C : "It was easy to locate elements"
      }
    }
  }
}

Poll Performance {
  Question q1 {
    "Value the time response"
    options {
      A : "Bad"
      B : "Fair"
      C : "Good"
    }
  }
}
```



# Grammar and Programs/Specifications/Models

```

grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
  'PollSystem' '{' polls+=Poll+ '}' ;

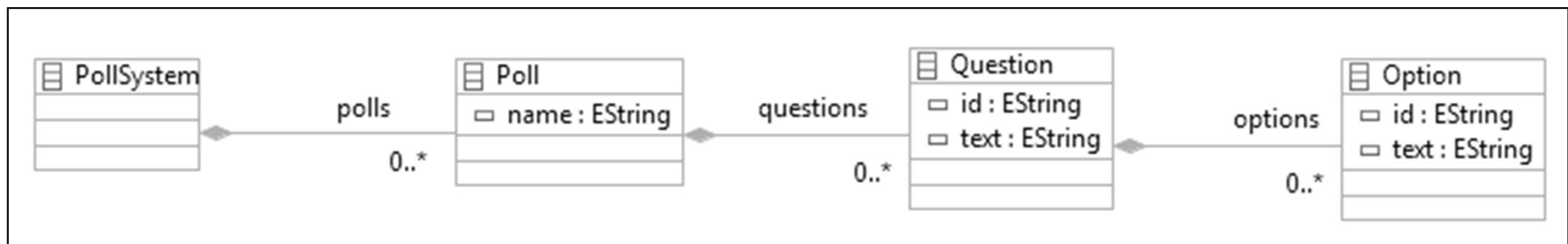
Poll:
  'Poll' name=ID '{' questions+=Question+'}';

Question:
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+'}' '}' ;

Option:
  id=ID ':' text=STRING;
  
```

```

PollSystem {
  Poll Quality {
    Question q1 {
      "Value the user experience"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
    Question q2 {
      "Value the layout"
      options {
        A : "It was not easy to locate elements"
        B : "I didn't realize"
        C : "It was easy to locate elements"
      }
    }
  }
  Poll Performance {
    Question q1 {
      "Value the time response"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
  }
}
  
```





# Grammar and Programs/Specifications/Models

```

grammar fr.miage.xtext.Poll with org.eclipse.xtext.common.Terminals

generate poll "http://www.miage.fr/xtext/Poll"

PollSystem:
  'PollSystem' '{' polls+=Poll+ '}' ;

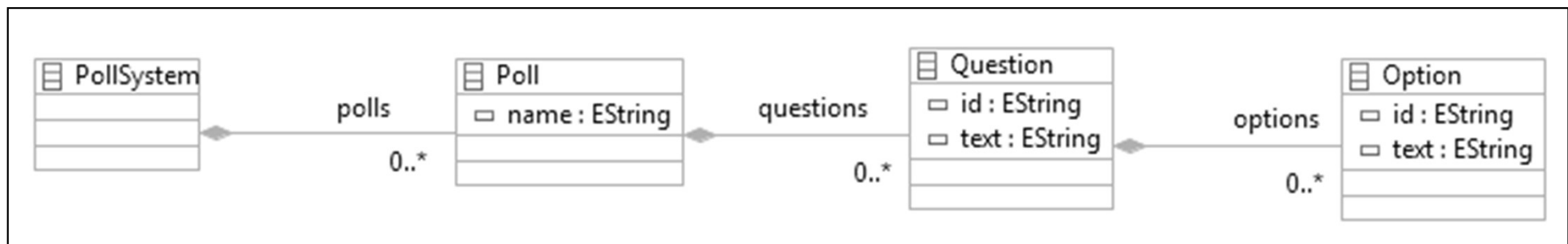
Poll:
  'Poll' name=ID '{' questions+=Question+'}';

Question:
  'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;

Option:
  id=ID ':' text=STRING;
  
```

```

PollSystem {
  Poll Quality {
    Question q1 {
      "Value the user experience"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
    Question q2 {
      "Value the layout"
      options {
        A : "It was not easy to locate elements"
        B : "I didn't realize"
        C : "It was easy to locate elements"
      }
    }
  }
  Poll Performance {
    Question q1 {
      "Value the time response"
      options {
        A : "Bad"
        B : "Fair"
        C : "Good"
      }
    }
  }
}
  
```



**Xtext, your DSL in 5'  
(incl. editors and  
serializers)**

**Live Demonstration**

Package Explorer

- org.xtext.example.questionnaire
  - src
    - org.xtext.example.mydsl
      - GenerateQuestionnaire.mwe2
      - Questionnaire.xtext
      - src-gen
      - xtend-gen
      - JRE System Library [JavaSE-1.8]
      - Plug-in Dependencies
      - META-INF
      - build.properties
    - org.xtext.example.questionnaire.sdk
    - org.xtext.example.questionnaire.tests
    - org.xtext.example.questionnaire.ui

Questionnaire.xtext

```
1 grammar org.xtext.example.mydsl.Questionnaire with org.eclipse.xtext.common.Terminals
2
3 generate questionnaire "http://www.xtext.org/example/mydsl/Questionnaire"
4
5 PollSystem:
6     'PollSystem' '{' polls+=Poll+ '}';
7
8 Poll:
9     'Poll' name=ID '{' questions+=Question+ '}';
10
11 Question : 'Question' id=ID '{' text=STRING 'options' '{' options+=Option+ '}' '}' ;
12
13 Option : id=ID ':' text=STRING ;
14
15
```

Package Explorer

- org.xtext.example.questionnaire
  - src
    - org.xtext.example.mydsl
      - GenerateQuestionnaire.mwe2**
      - Questionnaire.xtext
        - src-gen
        - xtend-gen
        - JRE System Library [JavaSE-1.8]
        - Plug-in Dependencies
        - META-INF
        - build.properties
        - org.xtext.example.questionnaire.sdk
        - org.xtext.example.questionnaire.tests
        - org.xtext.example.questionnaire.ui
        - org.xtext.example.videogenerator
        - org.xtext.example.videogenerator.sdk
        - org.xtext.example.videogenerator.tests
        - org.xtext.example.videogenerator.ui

Questionnaire.xtext

```
1 grammar org.xtext.example.mydsl.Questionnaire {
2
3 generate questionnaire "http://www.xtext.org/example/questionnaire/Questionnaire.xtext"
4
```

- New
- Open F3
- Open With
- Show In ⌘W
- Copy ⌘C
- Copy Qualified Name
- Paste ⌘V
- Delete ⌘X
- Build Path
- Refactor ⌘T
- Import...
- Export...
- Refresh F5
- Assign Working Sets...
- Validate
- Run As**
- Debug As
- Replace With
- Team
- Compare With
- Properties ⌘I

```
system '{' polls+=Poll+ '}';
name=ID '{' questions+=Quest
Question' id=ID '{' text=ST
-ID ':' text=STRING ;
```

- 1 MWE2 Workflow
- Run Configurations...



```
<terminated> Generate Language Infrastructure (org.xtext.example.questionnaire) [Mwe2 Launch] /Library/Java/JavaVirtualMachines/jdk1.8.0_31.jdk/Contents/Home/bin/java (28 sept. 2014)
0 [main] INFOlipse.emf.mwe.utils.StandaloneSetup - Registering platform uri '/Users/macher1/Documents/workspaceIDM1516'
127 [main] INFOlipse.emf.mwe.utils.StandaloneSetup - Adding generated EPackage 'org.eclipse.xtext.xbase.XbasePackage'
408 [main] INFOlipse.emf.mwe.utils.GenModelHelper - Registered GenModel 'http://www.eclipse.org/Xtext/Xbase/XAnnotations' from 'platform:/resources/org.eclipse.xtext.xbase.XAnnotations.ecore'
413 [main] INFOlipse.emf.mwe.utils.GenModelHelper - Registered GenModel 'http://www.eclipse.org/xtext/xbase/Xtype' from 'platform:/resources/org.eclipse.xtext.xbase.Xtype.ecore'
436 [main] INFOlipse.emf.mwe.utils.GenModelHelper - Registered GenModel 'http://www.eclipse.org/xtext/xbase/Xbase' from 'platform:/resources/org.eclipse.xtext.xbase.Xbase.ecore'
436 [main] INFOlipse.emf.mwe.utils.GenModelHelper - Registered GenModel 'http://www.eclipse.org/xtext/common/JavaVMTypes' from 'platform:/resources/org.eclipse.xtext.common.JavaVMTypes.ecore'
1005 [main] INFOlipse.emf.mwe.utils.StandaloneSetup - Adding generated EPackage 'org.eclipse.xtext.common.types.TypesPackage'
```

**\*ATTENTION\***

```
It is recommended to use the ANTLR 3 parser generator (BSD licence - http://www.antlr.org/license.html).
Do you agree to download it (size 1MB) from 'http://download.itemis.com/antlr-generator-3.2.0-patch.jar'? (type 'y' or 'n' and hit enter)y
11812 [main] INFOgenerator.parser.antlr.AntlrToolFacade - downloading file from 'http://download.itemis.com/antlr-generator-3.2.0-patch.jar'
108842 [main] INFOgenerator.parser.antlr.AntlrToolFacade - finished downloading.
108848 [main] INFOlipse.emf.mwe.utils.DirectoryCleaner - Cleaning /Users/macher1/Documents/workspaceIDM1516/org.xtext.example.questionnaire,
108849 [main] INFOlipse.emf.mwe.utils.DirectoryCleaner - Cleaning /Users/macher1/Documents/workspaceIDM1516/org.xtext.example.questionnaire,
108849 [main] INFOlipse.emf.mwe.utils.DirectoryCleaner - Cleaning /Users/macher1/Documents/workspaceIDM1516/org.xtext.example.questionnaire,
110353 [main] INFOlipse.emf.mwe.utils.GenModelHelper - Registered GenModel 'http://www.xtext.org/example/mydsl/Questionnaire' from 'platform:/resources/org.xtext.example.mydsl.Questionnaire.ecore'
113410 [main] INFOtext.generator.junit.Junit4Fragment - generating Junit4 Test support classes
113428 [main] INFOtext.generator.junit.Junit4Fragment - generating Compare Framework infrastructure
113584 [main] INFOlipse.emf.mwe2.runtime.workflow.Workflow - Done.
```

- org.xtext.example.questionnaire
  - src
    - org.xtext.example.mydsl
      - QuestionnaireRuntimeMod
      - QuestionnaireStandaloneS
      - GenerateQuestionnaire.mv
      - Questionnaire.xtext
    - org.xtext.example.mydsl.form
    - org.xtext.example.mydsl.gen
    - org.xtext.example.mydsl.scop
    - org.xtext.example.mydsl.valid
  - src-gen
  - xtend-gen
  - JRE System Library [JavaSE-1.8
  - Plug-in Dependencies
  - META-INF
  - model
  - build.properties
  - plugin.xml
  - org.xtext.example.questionnaire.sd
  - org.xtext.example.questionnaire.tes
  - org.xtext.example.questionnaire.ui
  - org.xtext.example.videogenerator
  - org.xtext.example.videogenerator.s
  - org.xtext.example.videogenerator.te
  - org.xtext.example.videogenerator.u

- New
- Go Into
- Open in New Window
- Open Type Hierarchy F4
- Show In  $\backslash$   $\mathbb{W}$
- Copy  $\mathbb{C}$
- Copy Qualified Name
- Paste  $\mathbb{V}$
- Delete  $\mathbb{X}$
- Build Path
- Source  $\backslash$   $\mathbb{S}$
- Refactor  $\backslash$   $\mathbb{T}$
- Import...
- Export...
- Refresh F5
- Close Project
- Close Unrelated Projects
- Assign Working Sets...
- Run As
- Debug As
- Validate
- Restore from Local History...
- Team
- Compare With
- Plug-in Tools
- Configure
- Properties  $\mathbb{I}$

```

org.xtext.example.mydsl.Questionnaire
e questionnaire "http://www.xtext.org/
tem:
llSystem' '{' polls+=Poll+ '}'';
ll' name=ID '{' questions+=Question+ '
n : 'Question' id=ID '{' text=STRING '
: id=ID ':' text=STRING ;
1 Eclipse Application  $\backslash$   $\mathbb{X}$  E
2 Java Applet  $\backslash$   $\mathbb{X}$  A
3 Java Application  $\backslash$   $\mathbb{X}$  J
4 OSGi Framework  $\backslash$   $\mathbb{X}$  O
Run Configurations...
INFO eclipse.emf.mwe.utils.GenModelHel
INFO eclipse.emf.mwe.utils.GenModelHel
INFO eclipse.emf.mwe.utils.GenModelHel
INFO lipse.emf.mwe.utils.StandaloneSe

```

## File

Create a new file resource.



Enter or select the parent folder:

FooQuestionnaire



Folder FooQuestionnaire

Folder VideoGen1

File name: foo2.q

Advanced >>

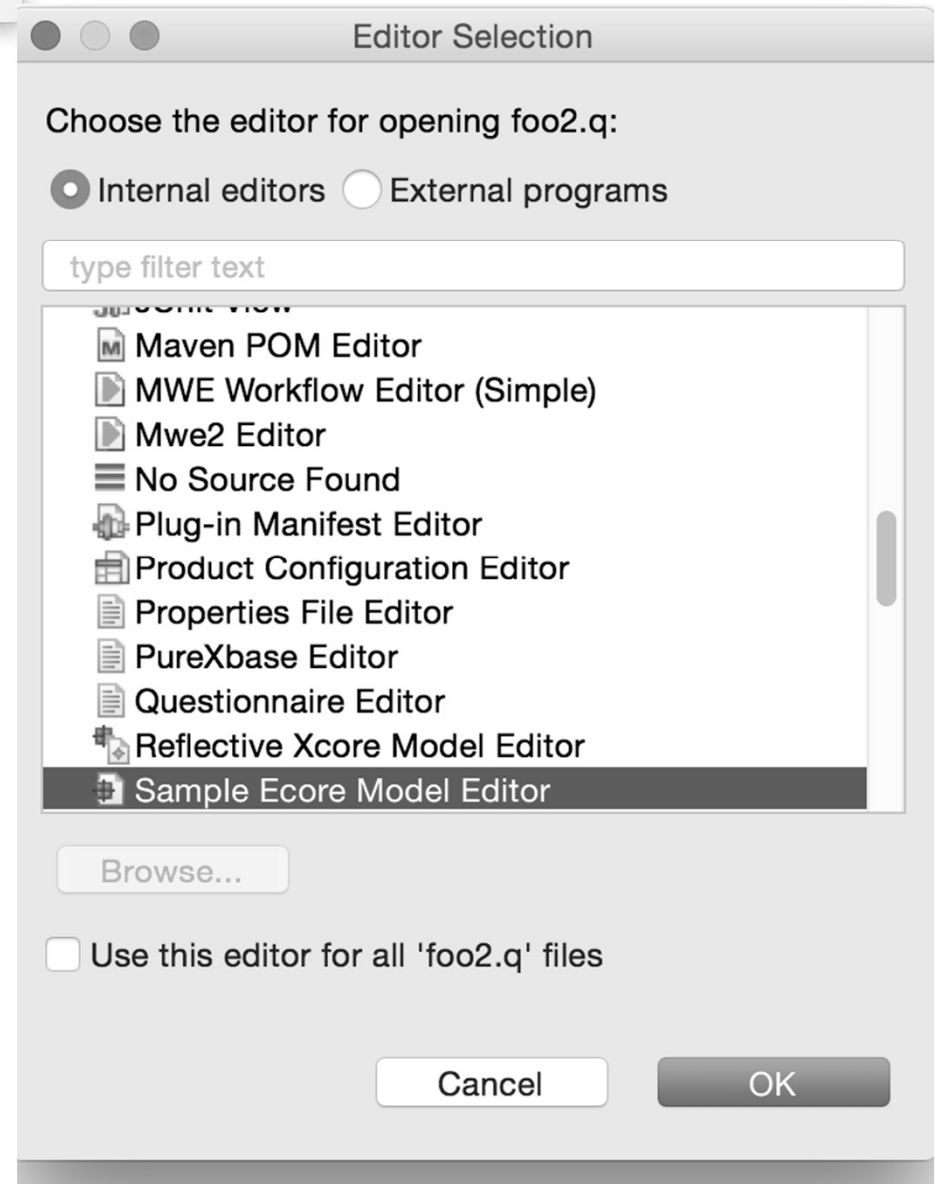
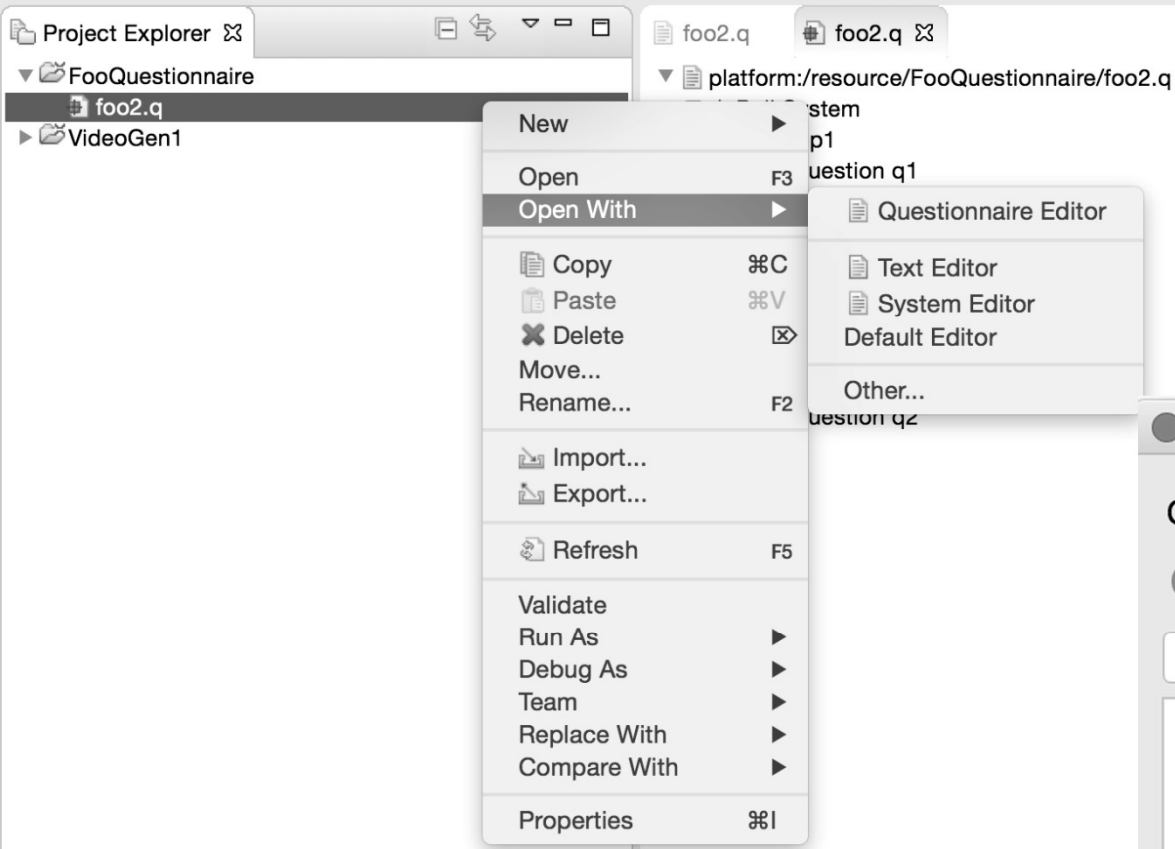


Cancel

Finish

```
PollSystem {  
  Poll p1 {  
    Question q1 {  
      "What is the best JavaScript framework for testing?"  
      options {  
        A1: "PhantomJS"  
        A2: "Jasmine"  
        A3: "Mocha"  
        A4: "I prefer to develop my own framework"  
      }  
    }  
    Question q2 {  
      "What is the best CSS preprocessor?"  
      options {  
        A1: "Less.js"  
        A2: "Sass"  
        A3: "Stylus"  
        A4: "I don't care about preprocessing CSS"  
      }  
    }  
  }  
  Poll p2 {  
    Question q1 {  
      "What is the best Java framework for testing?"  
      options {  
        A1: "JUnit"  
        A2: "Jasmine"  
        A3: "I prefer to develop my own framework"  
      }  
    }  
    Question q2 {  
      "What is the best Java library for logging?"  
      options {  
        A1: "Log4J"  
        A2: "java.util.logging"  
        A3: "I don't care about logging"  
      }  
    }  
  }  
}
```





2.q ✖

```
ollSystem {  
  
  Poll p1 {  
    Question q1 {  
      "What is the best JavaScript framework for testing?"  
      options {  
        A1: "PhantomJS"  
        A2: "Jasmine"  
        A3: "Mocha"  
        A4: "I prefer to develop my own framework"  
      }  
    }  
  
    Question q2 {  
      "What is the best CSS preprocessor?"  
      options {  
        A1: "Less.js"  
        A2: "Sass"  
        A3: "Stylus"  
        A4: "I don't care about preprocessing CSS"  
      }  
    }  
  }  
  
  Poll p2 {  
    Question q1 {  
      "What is the best Java framework for testing?"  
      options {  
        A1: "JUnit"  
        A2: "Jasmine"  
        A3: "I prefer to develop my own framework"  
      }  
    }  
  
    Question q2 {  
      "What is the best Java library for logging?"  
      options {  
        A1: "Log4J"  
        A2: "java.util.logging"  
        A3: "I don't care about logging"  
      }  
    }  
  }  
}
```

foo2.q

foo2.q ✖

platform:/resource/FooQuestionnaire/foo2.q

- ▼ ◆ Poll System
  - ▼ ◆ Poll p1
    - ▼ ◆ Question q1
      - ◆ Option A1
      - ◆ Option A2
      - ◆ Option A3
      - ◆ Option A4
    - ▶ ◆ Question q2
  - ▼ ◆ Poll p2
    - ▶ ◆ Question q1
    - ▶ ◆ Question q2



- ▼ org.xtext.example.questionnaire
  - ▶ src
  - ▶ src-gen
  - ▶ xtend-gen
  - ▶ JRE System Library [JavaSE-1.8]
  - ▶ Plug-in Dependencies
  - ▶ META-INF
  - ▼ model
    - ▼ generated
      - Questionnaire.ecore
      - Questionnaire.genmodel

Questionnaire.xtext

Questionnaire.ecore

platform:/resource/org.xtext.example.questionnaire/model/generated/Questionnaire.ecore

- ▼ questionnaire
  - ▼ PollSystem
    - ▶ polls : Poll
  - ▼ Poll
    - ▶ name : EString
    - ▶ questions : Question
  - ▼ Question
    - ▶ id : EString
    - ▶ text : EString
    - ▶ options : Option
  - ▼ Option
    - ▶ id : EString
    - ▶ text : EString

# My 3 take away messages

**#1 DSLs are important (as intuited for a long time - it will become more and more apparent)**

**#2 DSL technology is here (no excuse)**

**#3 MDE meets language engineering**

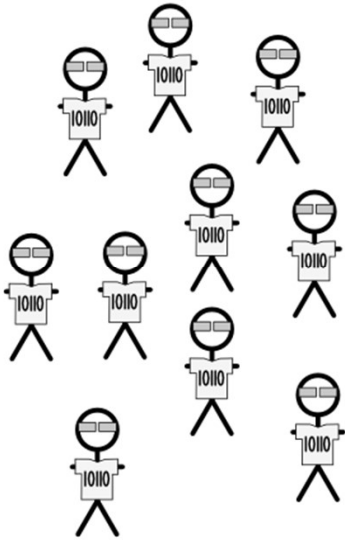
**But my take away  
message is NOT**

**That DSLs should be used  
systematically, in every  
situations**

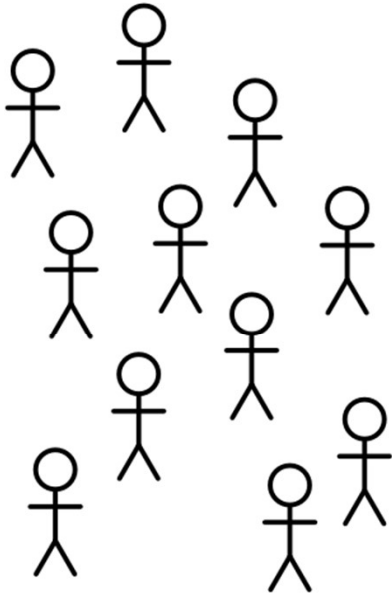
# When Developing DSLs?

- Tradeoff cost/time of development versus productivity gained for solving problems
  - If you use your DSL for resolving one problem, just one time, hum...
  - DSL: reusable, systematic means to resolve a specific task in a given domain
- DSL development can pay off quickly
  - 5' you can get a DSL
- But DSL development can be time-consuming and numerous worst practices exists

# Actors



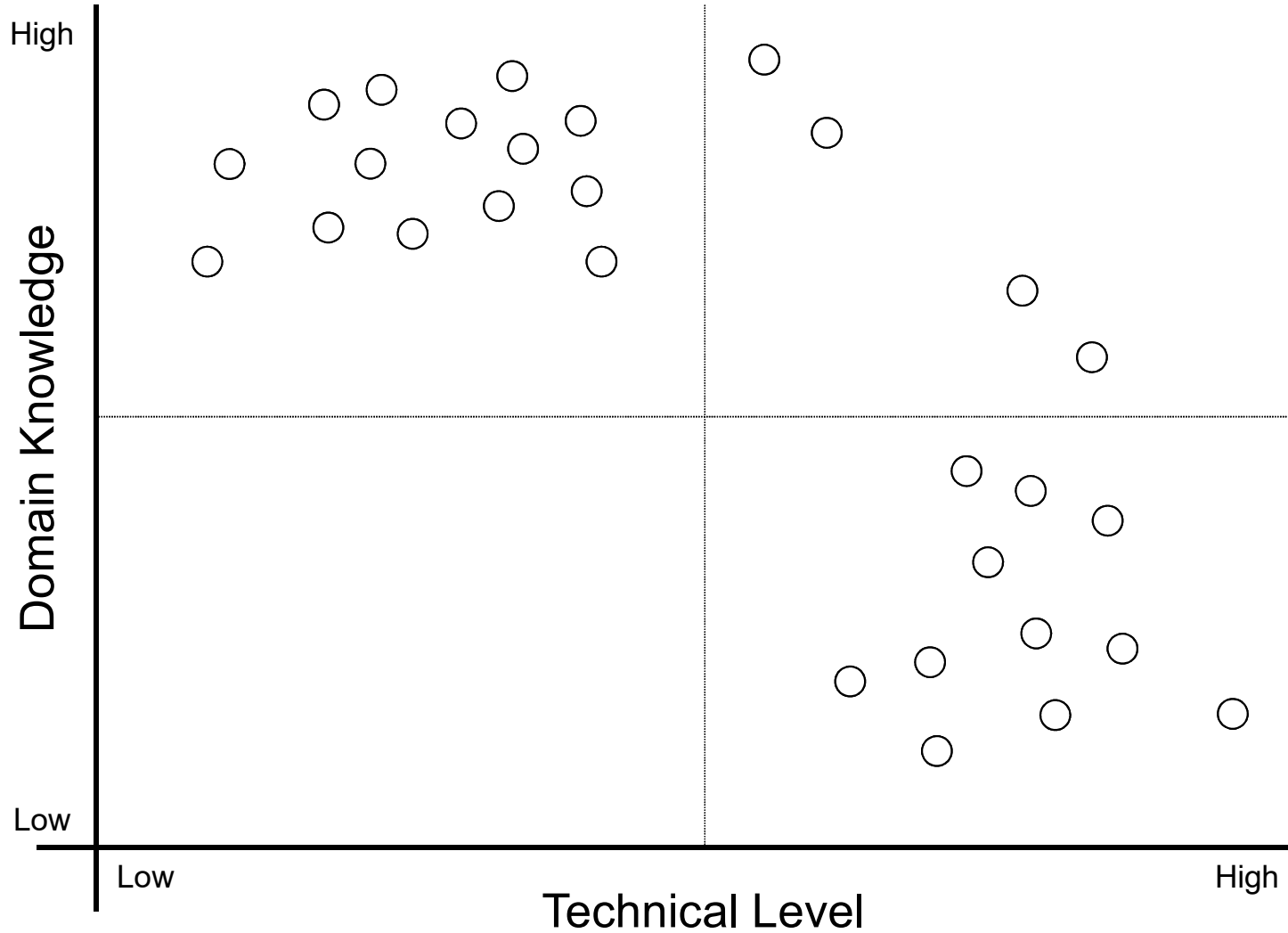
Developers



End-Users



# Actors Actors



# Best Practices

Limit  
Expressiveness

Viewpoints

Evolution

Learn from  
GPLs

Support

Tooling

# Worst Practices

- Initial conditions
  - Only Gurus allowed
    - Believe that only gurus can build languages or that “I’m smart and don’t need help”
  - Lack of Domain Understanding
    - Insufficiently understanding the problem domain or the solution domain
  - Analysis paralysis
    - Wanting the language to be theoretically complete, with its implementation assured

# Worst Practices

- The source for Language Concepts
  - UML: New Wine in Old Wineskins
    - Extending a large, general-purpose modeling language
  - 3GL Visual Programming
    - Duplicating the concepts and semantics of traditional programming languages
  - Code: The Library is the Language
    - Focusing the language on the current code's technical details
  - Tool: if you have a hammer
    - Letting the tool's technical limitations dictate language development

# Worst Practices

- The resulting language
  - Too Generic / Too Specific
    - Creating a language with a few generic concepts or too many specific concepts, or a language that can create only a few models
  - Misplaced Emphasis
    - Too strongly emphasizing a particular domain feature
  - Sacred at Birth
    - Viewing the initial language version as unalterable

# Worst Practices

- Language Notation
  - Predetermined Paradigm
    - Choosing the wrong representational paradigm or the basis of a blinkered view
  - Simplistic Symbols
    - Using symbols that are too simple or similar or downright ugly

# Worst Practices

- Language Use
  - Ignoring the use process
    - Failing to consider the language's real-life usage
  - No training
    - Assuming everyone understands the language like its creator
  - Pre-adoption Stagnation
    - Letting the language stagnate after successful adoption

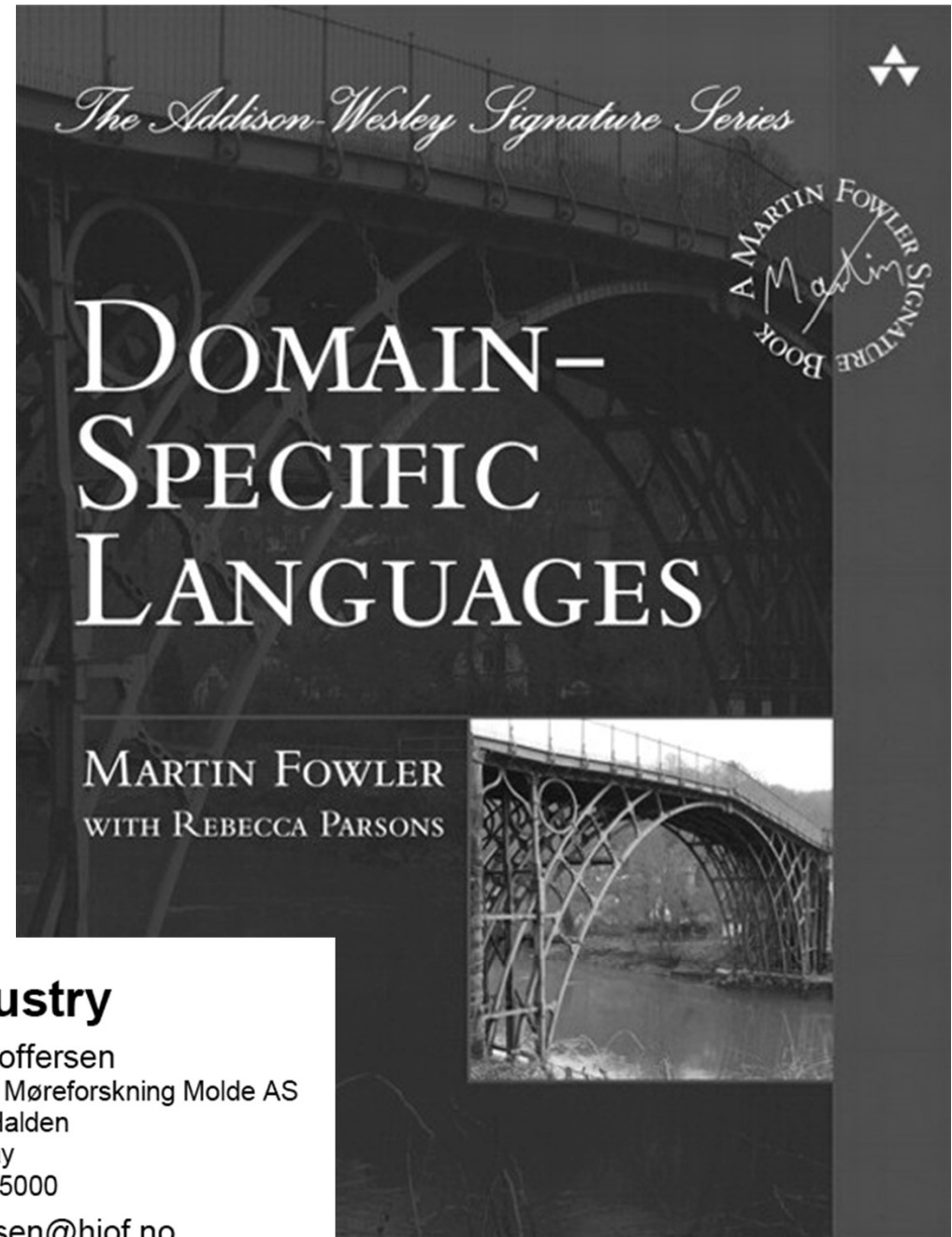
**Questions ?**

**(see also resources and  
lab sessions)**



<http://martinfowler.com/bliki/DomainSpecificLanguage.html>

# xtext



## Empirical Assessment of MDE in Industry

Jon Hutchinson, Jon Whittle, Mark Rouncefield  
School of Computing and Communications  
Lancaster University, UK  
+44 1524 510492

{j.hutchinson, j.n.whittle,  
m.rouncefield}@lancaster.ac.uk

Steinar Kristoffersen  
Østfold University College and Møreforskning Molde AS  
NO-1757 Halden  
Norway  
+47 6921 5000

steinar.kristoffersen@hiof.no